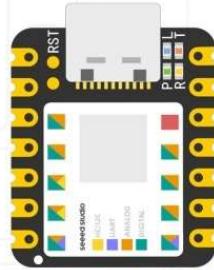


# XIAO: Big Power, Small Board

Mastering Arduino and TinyML

GPL-3.0 license



Lei Feng, Marcelo Rovai

trad. Italiana: Baldassarre Cesarano

seeed studio



EDGE  
IMPULSE

2024.01



# XIAO: Big Power, Small Board

Mastering Arduino e TinyML

Lei Feng, Marcelo Rovai

28 settembre 2024

## Indice

Prefazione .....	7
Ringraziamenti .....	8
Introduzione .....	8
Informazioni su Questo Libro.....	8
Il Pubblico .....	8
Cosa impareranno i lettori.....	9
Dipendenze software .....	9
Schema del libro.....	10
Dichiarazione di Copyright .....	11
Download della Versione PDF dell'eBook .....	11
Xiao: Big Power, Small Board - Edizione PDF .....	11
Come Eseguire il Download .....	11
Anteprima eBook (versione PDF inglese) .....	11
<b>1.1 Primo Programma Arduino con Seeed Studio XIAO: Blink .....</b>	<b>15</b>
1.1.1 Arduino IDE - l'Editor di Testo .....	16
1.1.2 Aggiungere Seeed Studio XIAO all'IDE Arduino .....	17
Collegamento di Seeed Studio XIAO ad Arduino IDE.....	21
⚠ Reset di Seeed Studio XIAO.....	24
Struttura dei Programmi Arduino .....	26
Segnali Digitali e Impostazioni di I/O .....	27
1.1.3 Attività 1: Eseguire Blink per far lampeggiare il LED di XIAO .....	28
Analisi del Codice.....	31
Caricamento del Programma.....	31
1.1.4 Attività 2: Completare l'esempio Blink collegando un LED esterno a Seeed XIAO ESP32C3 senza LED ..	34
Analisi del Codice.....	35
1.1.5 Esercizio Esteso .....	35
<b>1.2 Utilizzo del pulsante sulla scheda di espansione XIAO per controllare una luce LED.....</b>	<b>36</b>
1.2.1 Conoscenza di Base .....	36
1.2.1.1 Seeed Studio XIAO Expansion Board .....	36
1.2.1.2 Tre Strutture di Base dei Programmi .....	39
1.2.2 Attività 1: Controllare il LED sullo XIAO tramite il pulsante sulla "XIAO expansion board".....	42
Analisi.....	42
Variabile.....	42
Scrittura del Programma:.....	42
Caricamento del Programma.....	43
1.2.3 Attività 2: Utilizzare il pulsante sulla scheda di espansione XIAO per controllare il LED esterno su XIAO ESP32C3 .....	44
Alimentazione di XIAO con una batteria esterna .....	46
1.2.4 Esercizio Esteso .....	47
Flow Chart .....	47
<b>1.3 Trasformare XIAO e la sua Expansion Board in un Trasmettitore di Codice Morse .....</b>	<b>48</b>
1.3.1 Conoscenza di Base .....	48
1.3.1.1 Il Buzzer .....	48
1.3.1.2 Le Funzioni <code>tone()</code> e <code>noTone()</code> .....	49
1.3.1.3 Operatori Comuni .....	49
1.3.1.4 Codice Morse .....	50
1.3.2 Attività 1: Trasmissione Automatica di "SOS" .....	52
Analisi.....	52
1.3.3 Attività 2: Controllare il buzzer con un pulsante .....	54
1.3.4 Esercizio Esteso .....	55
<b>1.4 Monitorare i Cambiamenti di Valore della Manopola con il Serial Monitor .....</b>	<b>55</b>
1.4.1 Conoscenza di Base .....	56
1.4.1.1 Potenziometro Rotativo.....	56

1.4.1.2 I/O Analogico.....	56
PWM Pulse Width Modulation .....	57
1.4.1.3 Comunicazione Seriale.....	58
1.4.2 Attività 1: Utilizzare il Serial Monitor per Verificare se il pulsante è Premuto .....	59
1.4.3 Attività 2: Utilizzo del Serial Monitor per Visualizzare i Cambiamenti di Valore della Manopola .....	61
#define Definizione della Macro.....	61
1.4.4 Esercizio Esteso .....	63
<b>1.5 Controllo del LED e del Servo con una Manopola .....</b>	<b>64</b>
1.5.1 Conoscenza di Base .....	64
1.5.1.1 Servo e Libreria Servo .....	64
1.5.1.2 Funzione map () .....	65
1.5.2 Attività 1: Utilizzo di un potenziometro a manopola per controllare la luminosità del LED a bordo della scheda XIAO .....	66
Controllo di un LED esterno con una manopola su XIAO ESP32C3 .....	69
1.5.3 Attività 2: Controllo di un Servomotore con un Potenziometro Rotativo.....	70
1.5.4 Esercizio Esteso .....	73
<b>1.6 Visualizzazione di “Hello World” su OLED .....</b>	<b>73</b>
1.6.1 Conoscenza di Base .....	74
1.6.1.1 Il Display OLED.....	74
1.6.1.2 Come Scaricare e Installare la Libreria U8g2_Arduino .....	74
1.6.1.3 Libreria U8g2 per OLED.....	75
1.6.2 Attività 1: Visualizzare Hello World! sull’OLED della scheda di espansione XIAO .....	77
1.6.3 Attività 2: Disegnare un cerchio sul display OLED .....	78
1.6.4 Esercizio Esteso .....	80
<b>2.1 Introduzione alla Progettazione di Prototipi di Prodotti.....</b>	<b>81</b>
2.1.1 Coltivare la Mentalità del Maker .....	81
Mantieni un atteggiamento giocoso .....	81
Sii Curioso .....	81
Sii fisico .....	82
Trova uno strumento preferito .....	82
Fai qualcosa che non hai mai fatto prima .....	82
Crea qualcosa .....	82
Quali sono i vantaggi del coinvolgimento dei giovani nella creazione?.....	82
Quali sono gli attributi di un maker? Cos’è una mentalità da maker?.....	82
2.1.2 Illuminazioni sulla Progettazione di Prototipi di Prodotto .....	83
2.1.2.1 Processo di Base della Progettazione di Prototipi di Prodotto .....	83
2.1.2.2 Pratica di Prototipo di Prodotto - Prototipo “Allarme a Distanza di un Metro” .....	86
<b>2.2 Igrometro e Termometro Intelligenti .....</b>	<b>92</b>
2.2.1 Conoscenza di Base .....	92
2.2.1.1 Temperatura .....	92
2.2.1.2 Termometro .....	93
2.2.1.3 Umidità .....	94
2.2.1.4 Sensore di Temperatura e Umidità — Sensore di Temperatura e Umidità Grove V2.0 (DHT20) .....	94
2.2.2 Attività 1: Lettura dei Valori di Temperatura e Umidità nel Monitor Seriale (Basato sul modello DHT20) .....	94
2.2.3 Creazione del Progetto: Misuratore Intelligente di Temperatura e Umidità .....	99
Descrizione del Progetto .....	99
Scrittura del Programma .....	99
Operatori Booleani.....	101
2.2.4 Design dell’Aspetto .....	103
Caso di riferimento .....	104
<b>2.3 Scatola Regalo a Sorpresa Basata sul Sensore di Luce .....</b>	<b>104</b>
2.3.1 Conoscenza di Base .....	104
2.3.1.1 Sensore di Luce.....	104
2.3.1.2 Striscia di LED RGB.....	106
2.3.2 Attività 1: Accendere la Striscia LED RGB Per Iniziare a Usare le Strisce LED RGB, Iniziare installando e comprendendone la libreria. .....	106
2.3.3 Progetto: Scatola Regalo a Sorpresa.....	108
Descrizione del Progetto .....	108
Scrittura del Programma .....	109
2.3.4 Design Esteriore.....	112
<b>2.4 Danza Ritmica con un Accelerometro Triassiale .....</b>	<b>113</b>
2.4.1 Conoscenza di Base .....	113
2.4.1.1 Accelerometro Triassiale .....	113
2.4.2 Attività 1: Lettura dei Valori dagli Assi XYZ dell’Accelerometro a Tre Assi.....	119
2.4.3 Progetto di Produzione: Danza Ritmica .....	122

Descrizione del Progetto.....	122
Scrittura del Programma.....	122
Attività: Controllare la Striscia LED RGB per Modificare gli Effetti di Luce Tramite Accelerometro a Tre Assi .....	122
2.4.4 Design Esteriore .....	125
<b>3.1 Porta con Telecomando Intelligente .....</b>	<b>126</b>
3.1.1 Conoscenza di Base .....	126
3.1.1.1 Ricevitori e Trasmettitori a Infrarossi .....	126
3.1.2 Attività 1: Lettura dei Codici Chiave del Telecomando.....	127
3.1.3 Creazione del Progetto: Smart Remote Door .....	129
Descrizione del Progetto.....	129
Scrittura del Programma.....	129
Attività 2: Controllare la Rotazione del Servo con un Telecomando a Infrarossi.....	129
3.1.4 Design Esteriore .....	132
<b>3.2 Orologio Intelligente .....</b>	<b>133</b>
3.2.1 Conoscenza di Base .....	134
3.2.1.1 RTC Clock.....	134
3.2.2 Attività 1: Visualizzazione dell’Orologio RTC nel Monitor Seriale .....	134
3.2.3 Project Making: Orologio Smart .....	136
Descrizione del Progetto.....	136
Programmazione .....	136
3.2.4 Design Esteriore .....	141
<b>3.3 Pianoforte ad Aria .....</b>	<b>143</b>
3.3.1 Conoscenza di Base .....	143
3.3.1.1 Sensore di Distanza a Ultrasuoni Grove .....	143
3.3.2 Attività 1: Lettura del Valore del Grove Ultrasonic Distance Sensor.....	143
3.3.3 Progetto di Produzione: Arpa ad Aria Ultrasonica.....	146
Descrizione del Progetto.....	146
Scrittura del Programma.....	146
Utilizzo della Funzione tone() per Riprodurre una Melodia .....	147
Attività 2: Arpa ad Aria Ultrasonica.....	148
3.3.4 Design Esteriore .....	150
<b>3.4 Implementazione di connessioni Wi-Fi e applicazioni con XIAO ESP32C3.....</b>	<b>151</b>
3.4.1 Conoscenza di Base .....	151
3.4.1.1 Modello di Riferimento OSI (Modello di Rete a Sette Livelli) .....	151
3.4.1.2 ICMP (Internet Control Message Protocol) e Comando ping .....	152
3.4.2 Attività 1: Utilizzo della rete Wi-Fi su XIAO ESP32C3.....	153
3.4.3 Attività 2: Ping di un Sito Web Specifico.....	158
3.4.4 Creazione del Progetto: Utilizzo di XIAO ESP32C3 per Effettuare Richieste HTTP GET e HTTP POST .....	161
3.4.4.1 Introduzione al Protocollo HTTP .....	161
3.4.4.2 Introduzione al Protocollo HTTPS .....	161
3.4.4.3 Metodi di richiesta HTTP.....	162
3.4.4.4 Attività 3: Utilizzo di XIAO ESP32C3 per Inviare una Richiesta HTTP GET .....	162
3.4.4.5 Attività 4: Utilizzo di XIAO ESP32C3 per Inviare una Richiesta HTTP POST .....	165
<b>3.5 Telemetria e Comandi Tramite Protocollo MQTT con XIAO ESP32C3.....</b>	<b>167</b>
3.5.1 Conoscenza di Base .....	168
3.5.1.1 IoT (Internet of Things).....	168
3.5.1.2 Protocolli di Comunicazione .....	168
3.5.1.3 Message Queuing Telemetry Transport (MQTT) .....	168
3.5.2 Attività 1: Collegare XIAO ESP32C3 al Broker MQTT .....	171
3.5.3 Approfondimento su MQTT.....	175
3.5.4 Telemetria.....	175
3.5.5 Attività 2: Invio di Informazioni di Telemetria da XIAO al Broker MQTT.....	176
Con quale frequenza deve essere inviata la telemetria? .....	180
Perdita di connessione .....	181
3.5.6 Comandi.....	182
3.5.7 Attività 3: Inviare Comandi a XIAO Tramite Broker MQTT.....	183
Connessione persa .....	187
Utilizzo della funzione Bluetooth di XIAO .....	187
<b>4.1 Informazioni su TinyML ed Edge Impulse Studio .....</b>	<b>188</b>
4.1.1 Termini Comuni.....	188
4.1.1.1 Sistemi Embedded.....	188
4.1.1.2 Edge Computing e Internet of Things (IoT) .....	189
4.1.1.3 Intelligenza artificiale (IA) .....	190
4.1.1.4 Machine Learning (ML) .....	191
4.1.1.5 Edge AI .....	192

4.1.1.6 Apprendimento Automatico Embedded .....	192
4.1.1.7 Tiny Machine Learning (TinyML).....	192
4.1.2 Vantaggi e Funzionamento dell'IA Edge .....	193
4.1.3 Applicazioni di Edge AI.....	195
4.1.4 Introduzione a Edge Impulse Studio .....	197
<b>4.2 Rilevamento di Anomalie e Classificazione del Movimento .....</b>	<b>197</b>
4.2.1 Cose utilizzate in questo progetto .....	198
Componenti hardware .....	198
App software e servizi online .....	198
4.2.2 Introduzione .....	198
4.2.3 XIAO nRF52840 Sense .....	199
4.2.3.1 Collegamento di XIAO nRF52840 Sense con Arduino IDE.....	199
4.2.3.2 Test del Microfono.....	201
4.2.3.3 Test dell'IMU .....	203
4.2.4 Il Modello di Classificazione del Movimento TinyML.....	207
4.2.4.1 Collegamento di un dispositivo a Edge Impulse Studio.....	208
4.2.4.2 Raccolta Dati .....	211
4.2.4.3 Pre-elaborazione dei Dati .....	212
4.2.4.4 Progettazione del Modello .....	214
4.2.4.5 Impulse Design .....	215
4.2.4.6 Generazione di feature .....	220
4.2.4.7 Training .....	221
4.2.4.8 Test .....	224
4.2.4.9 Classificazione Live .....	225
4.2.4.10 Deployment .....	226
4.2.4.11 Inferenza .....	229
Post-elaborazione.....	231
4.2.4.12 Conclusione.....	231
4.2.4.13 Applicazioni di Casi .....	232
<b>4.3 Classificazione del Suono (KWS).....</b>	<b>233</b>
4.3.1 Cose utilizzate in questo progetto .....	233
Componenti hardware .....	233
App software e servizi online .....	233
4.3.2 Introduzione .....	233
4.3.3 Il progetto KWS .....	235
4.3.3.1 Il Flusso di Lavoro del Machine Learning .....	236
4.3.3.2 Il Dataset.....	236
4.3.3.3 Acquisizione di dati audio online con Edge Impulse e uno smartphone.....	237
4.3.3.4 Acquisizione di Dati Audio con XIAO nRF52840 Sense .....	237
4.3.3.5 Acquisizione (offline) di Dati Audio con uno martphone o un PC .....	242
4.3.3.6 Modello di training con Edge Impulse Studio .....	242
4.3.3.7 Creazione di Impulse (Pre-Process / Definizione del Modello) .....	246
4.3.3.8 Pre-elaborazione (MFCC) .....	248
4.3.3.9 Progettazione e Addestramento del Modello .....	249
4.3.3.10 Test .....	251
4.3.3.11 Distribuzione e Inferenza .....	252
4.3.3.12 Post-elaborazione .....	256
4.3.3.13 Conclusione.....	258
<b>4.4 Classificazione delle Immagini .....</b>	<b>259</b>
4.4.1 Cose utilizzate in questo progetto .....	259
4.4.1.1 Componenti hardware.....	259
4.4.2 App software e servizi online.....	259
4.4.2 Introduzione .....	260
4.4.3 Installazione di XIAO ESP32S3 Sense su Arduino IDE.....	262
4.4.4 Test della scheda con BLINK .....	263
4.4.5 Collegamento del Modulo Sense (Scheda di Espansione) .....	264
4.4.6 Test del Microfono.....	265
4.4.7 Test della Fotocamera .....	267
4.4.8 Test del WiFi .....	269
4.4.9 Frutta contro Verdura - Un progetto di Classificazione delle Immagini TinyML.....	274
4.4.10 Addestramento del modello con Edge Impulse Studio .....	276
4.4.11 Test del modello (inferenza) .....	289
4.4.12 Test con un Modello Più Grande .....	290
4.4.13 Esecuzione dell'inferenza su SenseCraft-Web-Toolkit .....	292
4.4.14 Conclusione .....	297

<b>4.5 Rilevamento degli Oggetti .....</b>	<b>297</b>
4.5.1 Cose utilizzate in questo progetto.....	297
4.5.1.1 Componenti hardware .....	297
4.5.2 App software e servizi online .....	297
4.5.2 Introduzione.....	298
4.5.2.1 Object Detection e Image Classification.....	298
4.5.2.2 Una soluzione innovativa per il Rilevamento di Oggetti: FOMO .....	300
4.5.3 Obiettivo del Progetto di Object Detection .....	300
4.5.4 Raccolta Dati .....	302
4.5.4.1 Raccolta di Dataset con XIAO ESP32S3 .....	302
4.5.4.2 Edge Impulse Studio .....	304
4.5.4.3 Caricamento dei dati non etichettati .....	305
4.5.4.4 Etichettatura del Dataset .....	306
4.5.4.5 Bilanciamento del dataset e suddivisione Train/Test .....	308
4.5.5 Impulse Design .....	308
4.5.5.1 Pre-elaborazione di tutti i dataset.....	309
4.5.6 Progettazione, Addestramento e Test del Modello .....	311
4.5.6.1 Modello di test con “Live Classification” .....	313
4.5.7 Deploying del Modello (Arduino IDE) .....	315
4.5.8 Distribuzione del Modello (SenseCraft-Web-Toolkit).....	319
4.5.9 Conclusione.....	323
<b>4.6 Per saperne di più.....</b>	<b>323</b>
Corsi Online .....	323
Libri.....	323
Repository di Progetti .....	323
<b>5.1 Progetti XIAO creativi e utili.....</b>	<b>323</b>
5.1.1 Salt Water Tracker (SWT) Trasportato da Drone .....	324
5.1.2 Progetto SAJAC: Giacca Intelligente per Avventure Speleologiche .....	325
5.1.3 Computer da Bicicletta su Spresense .....	326
5.1.4 Elaborazione dello Yogurt Basata su IoT AI e Previsione della Consistenza con Blynk.....	328
5.1.5 Robot azionato da browser Web per il rilevamento di perdite di gas .....	329
5.1.6 Controller del Treno con Seeed Studio XIAO ESP32C3 .....	329
5.1.7 Auto Radio Comandata (Stampata in resina 3D Basata su Arduino) RC_Car_RP .....	331
5.1.8 Tracker per Attività di Animali Domestici con XIAO BLE Sense e Edge Impulse.....	332
5.1.9 H.E.D.S. Al polso, la nuova scheda Seeeduino XIAO .....	333
5.1.10 Monitor del Battito Cardiaco con XIAO NRF52840.....	334
5.1.11 Controller Multi MIDI, Filtro, Router e Generatore di Suoni .....	338
5.1.12 Sintetizzatore Modulare eurorack fai da te Raspberry Pi VCO con Seeed XIAO .....	338
5.1.13 Sequencer Xiao CV .....	339
5.1.14 ANAVI Macro Pad 10 e Manopole .....	340
5.1.15 Lampada da Scrivania Death Stranding.....	341
5.1.16 HackerBox 0077 : Veritas .....	342
5.1.17 DISCIPLINE - Un timer per l’allenamento.....	343
5.1.18 Seeed Fusion DIY XIAO Mecha.....	346
1° premio: TOTEM   una minuscola tastiera divisa con splay .....	347
2° premio: Beyblock20   un MacroPad magnetico e modulare .....	347
2° premio: Purple Owl   una tastiera al 60% con tecnologia Seeed XIAO RP2040 .....	348
3° premio: KLEIN   una tastiera ergonomica wireless .....	348
3° premio: GRIN Quern   una tastiera ergonomica sul trackpad centrale.....	348
3° premio: Kidoairaku Swallowtail   una simpatica tastiera a forma di farfalla .....	350
<b>Riferimenti .....</b>	<b>350</b>
<b>Informazioni sugli autori .....</b>	<b>350</b>

## Prefazione

Dalle grandi board del passato, Arduino ha fatto molta strada ed è entrato nella serie Seeed Studio XIAO: grande quanto un pollice ma potente, aprendo un vasto orizzonte all’innovazione. “XIAO: Big Power, Small Board” si tuffa in queste capacità, guidando i lettori dalle basi di Arduino a progetti miniaturizzati intricati. Che i lettori vogliano illuminare un LED o approfondire lo “Embedded Machine Learning (TinyML)” con schede XIAO ed Edge Impulse Studio, questo libro li copre. Si devono avere conoscenze pregresse? Nessun problema!

Questo libro adotta un approccio pratico basato sui progetti, assicurando che i lettori comprendano i concetti durante l'implementazione. Alla fine, saranno esperti di XIAO e ispirati da molti progetti creati dagli utenti che mostrano le infinite possibilità offerte da questa piccola scheda.

## Ringraziamenti

Vogliamo esprimere la nostra sincera gratitudine a Jiamou Yang, Yanming Wen, Mengdu Li, Chunchun Tian, Haixu Liu, Tianrui Wang e Jianjing Huang per il loro prezioso supporto tecnico e le revisioni del manoscritto. Questo libro non sarebbe stato possibile senza il loro contributo.

Estendiamo la nostra più profonda gratitudine all'intera TinyML4D Academic Network, composta da illustri professori, ricercatori e professionisti. I notevoli contributi di Marco Zennaro, Brian Plancher, José Alberto Ferreira, Jesus Lopez, Diego Mendez, Shawn Hymel, Dan Situnayake, Pete Warden e Laurence Moroney sono stati determinanti nel far progredire la nostra comprensione dell'Embedded Machine Learning (TinyML).

Un elogio speciale è riservato al Professor Vijay Janapa Reddi dell'Università di Harvard. La sua incrollabile convinzione nel potenziale trasformativo delle comunità open source, unita alla sua inestimabile guida e ai suoi insegnamenti, è stata un faro per il nostro lavoro fin dall'inizio.

Riconoscendo queste persone, rendiamo omaggio alla saggezza e alla dedizione collettive che hanno arricchito questo campo e il nostro lavoro.

---

Immagini illustrate sulle copertine degli e-book e dei capitoli generate da DALL-E di OpenAI tramite ChatGPT

## Introduzione

La serie Seeed Studio XIAO rappresenta un'evoluzione rivoluzionaria nell'ecosistema Arduino, che unisce compattezza e prestazioni potenti. Comprendere e sfruttare le sue capacità è essenziale per qualsiasi appassionato o professionista di elettronica e "machine learning" [apprendimento automatico]. Con la rapida progressione della tecnologia e la crescente domanda di dispositivi più piccoli ed efficienti, padroneggiare XIAO e la sua integrazione con TinyML è fondamentale. Rappresenta una nuova frontiera per l'innovazione, consentendo la creazione di progetti sofisticati in spazi precedentemente ritenuti impossibili. Questo argomento è fondamentale in quanto si allinea con la futura traiettoria dell'elettronica, dell'IoT e dell'apprendimento automatico, rendendolo indispensabile per coloro che mirano a rimanere all'avanguardia dei progressi tecnologici.

## Informazioni su Questo Libro

### Il Pubblico

Il pubblico principale di "XIAO: Big Power, Small Board" comprende hobbisti, studenti, educatori e professionisti dell'elettronica e dell'apprendimento automatico che desiderano esplorare e massimizzare il potenziale delle piattaforme hardware compatte. In genere, questi lettori potrebbero ricoprire posizioni come appassionati di elettronica, creatori di progetti fai da te, educatori di elettronica o persino sviluppatori junior di sistemi embedded. Man mano che avanzano nella loro carriera, potrebbero puntare a ruoli come ingegneri di progettazione elettronica, sviluppatori IoT o integratori di hardware per l'apprendimento automatico.

Il nostro pubblico possiede una conoscenza di base dei concetti di elettronica, ma potrebbe non aver ancora approfondito la programmazione Arduino o la progettazione di hardware compatto. Probabilmente hanno incontrato libri per principianti standard su Arduino o elettronica generale, ma potrebbero non essersi ancora avventurati in hardware specializzato o TinyML. Per quanto riguarda le competenze, hanno una certa esperienza pratica con l'elettronica di base o la programmazione, ma non hanno padroneggiato le complessità di TinyML o le funzionalità avanzate del microcontrollore.

## Cosa impareranno i lettori

Alla fine di questo libro, il lettore comprenderà:

- I fondamenti dell'hardware open source, concentrandosi sulle capacità della serie Seeed Studio XIAO.
- Come passare da progetti elettronici di base ad avanzati, iniziando con semplici controlli LED e avanzando verso applicazioni complesse come telemetria e rilevamento di parole chiave vocali.
- I concetti alla base della progettazione di prototipi e le sue implicazioni pratiche nello sviluppo del prodotto.
- Le complessità dell'integrazione di vari moduli come il ricevitore a infrarossi, il sensore di distanza a ultrasuoni e l'orologio RTC con la piattaforma XIAO.
- L'importanza e l'applicazione di Tiny Machine Learning (TinyML), sottolineando il suo potere trasformativo in hardware come XIAO nRF52840 Sense ed ESP32S3 Sense.
- Tecniche per utilizzare strumenti avanzati come Edge Impulse Studio per applicazioni del mondo reale come rilevamento di anomalie e oggetti e classificazione di video o suoni.

Il lettore sarà in grado di:

- Impostare, programmare e risolvere i problemi dei progetti su tutte le schede della serie XIAO, passando dalle interazioni hardware di base a progetti complessi.
- Convertire idee astratte in prototipi di prodotti elettronici tangibili, sfruttando le intuizioni del corso.
- Progettare e implementare progetti di livello intermedio come uno Smart Watch e un Air Piano utilizzando sensori e moduli specializzati.
- Sfruttare la potenza dei protocolli Wi-Fi e MQTT con XIAO ESP32C3 per comunicazioni cloud e scambio dati.
- Distribuire TinyML su diverse schede XIAO, eseguendo attività come classificazione di immagini, movimento e suono oltre al rilevamento di anomalie e oggetti.
- Innovare ed estendere le idee di progetto, traendo ispirazione da una raccolta curata di progetti XIAO e adattandoli alle proprie esigenze.

## Dipendenze software

- **Arduino IDE:** Aggiornamenti o modifiche importanti all'Arduino IDE potrebbero influire sui contenuti correlati allo sviluppo e alla programmazione di Arduino nel libro.
- **Librerie Seeed Studio XIAO:** Gli aggiornamenti alle librerie specifiche della serie XIAO possono influenzare i progetti o i codici di esempio forniti.
- **Edge Impulse Studio:** Aggiornamenti significativi o modifiche alle funzionalità su questa piattaforma richiederebbero modifiche nei capitoli TinyML.
- **Librerie/Protocolli MQTT:** Eventuali modifiche relative alle librerie MQTT o al protocollo stesso potrebbero influenzare il contenuto della telemetria e dei comandi.
- **Librerie ESP32:** Gli aggiornamenti alle librerie utilizzate dalla scheda XIAO ESP32C3 ed ESP32S3 potrebbero avere un impatto sui progetti o sugli esempi associati.

## Schema del libro

- **Capitolo 1: Introduzione all'Hardware e alla Programmazione**

In questo capitolo, i lettori iniziano con la programmazione di base su XIAO utilizzando Arduino IDE. Attraverso semplici programmi di esempio, impareranno a controllare luci LED, pulsanti, buzzer e altri componenti elettronici, padroneggiando i concetti di programmazione di base come I/O digitale, I/O analogico, generazione di toni e valori di mappatura. Digitando manualmente esempi di codice riga per riga, svilupperanno forti abitudini di codifica e comprenderanno la sintassi della programmazione.

- **Capitolo 2: Progetto Pratico per Principianti - Introduzione alla Progettazione di Prototipi**

In questo capitolo, i lettori apprenderanno le basi della progettazione di prototipi con XIAO attraverso progetti adatti ai principianti. Partiranno da un'idea e creeranno rapidamente un prototipo di verifica, concentrandosi maggiormente sull'applicazione pratica del codice piuttosto che sull'analisi riga per riga. Sfruttando le librerie Arduino, le risorse della community e i programmi di esempio, impareranno come trovare e adattare frammenti di codice per ottenere gli effetti desiderati in modo efficiente. Inoltre, esplorano come progettare l'aspetto fisico dei prototipi combinando in modo creativo hardware elettronico con oggetti di uso quotidiano. I risultati chiave sono l'acquisizione di un approccio basato su progetti e lo sviluppo di competenze per costruire semplici prototipi interattivi.

- **Capitolo 3: Progetto Pratico Intermedio - Progetti Complessi**

In questo capitolo, i lettori miglioreranno le proprie competenze di prototipazione creando progetti IoT sofisticati con XIAO. Implementeranno funzionalità come connettività Wi-Fi, telemetria MQTT e comandi di controllo remoto utilizzando XIAO ESP32C3. Attraverso build complesse come una porta remota intelligente, uno smartwatch e una pianola ad aria, affineranno le tecniche di programmazione per la comunicazione wireless, l'integrazione cloud e il controllo incorporato. Saranno forniti progetti opzionali, ma i lettori sono incoraggiati a esplorare progetti di contenitori creativi con materiali alternativi. I risultati chiave sono la padronanza della prototipazione IoT intermedia e la preparazione per applicazioni TinyML avanzate.

- **Capitolo 4: Progetto Pratico Avanzato - Applicazione TinyML**

Tra i prodotti della serie XIAO, "Seeed Studio XIAO nRF52840 Sense" ha connettività wireless Bluetooth 5.0, basso consumo energetico, microfono sensori PDM e IMU a 6 assi integrati. XIAO ESP32S3 Sense integra inoltre una fotocamera, un microfono digitale e supporto per schede SD. Queste caratteristiche li rendono strumenti potenti per progetti TinyML (Embedded Machine Learning). TinyML risolve i problemi in un modo completamente diverso dai metodi di programmazione tradizionali. Questo capitolo introdurrà i lettori a questo campo all'avanguardia esaminando l'intero flusso di lavoro di apprendimento automatico dalla raccolta dati, formazione e test alla distribuzione e inferenza utilizzando lo strumento Edge Impulse Studio.

- **Capitolo 5: Esperimenti Creativi**

Sin dal suo lancio, la serie Seeed Studio XIAO è stata ampiamente acclamata per le sue dimensioni compatte, le prestazioni potenti e la versatile gamma di prodotti. La comunità dei maker ha prodotto un gran numero di progetti creati con XIAO. A causa delle limitazioni di spazio, abbiamo selezionato alcuni progetti eccezionali realizzati con XIAO dai nostri maker. Questi progetti dimostrano appieno le potenti funzioni e le ampie applicazioni di XIAO. Seguiamo i passi dei maker, stimoliamo la creatività ed esploriamo le infinite possibilità di XIAO. I lettori possono trarre ispirazione da questi progetti, usare l'immaginazione ed esplorare nuovi territori con XIAO.

## Dichiarazione di Copyright

Questo libro, "XIAO: Big Power, Small Board", è pubblicato con licenza GNU General Public License versione 3.0 (GPL-3.0), garantendo che rimanga libero e aperto a tutti per l'uso, la modifica e la distribuzione. La licenza GPL-3.0 è una licenza di software libero ampiamente rispettata e utilizzata, che garantisce agli utenti la libertà di eseguire, studiare, condividere e modificare il software. Gli autori, Lei Feng e Marcelo Rovai, insieme a tutti i collaboratori, abbracciano i principi della conoscenza e dell'istruzione aperte, assicurando che questa preziosa risorsa rimanga accessibile a coloro che cercano di migliorare le proprie competenze in Arduino, TinyML e oltre. Scegliendo questa licenza, ci impegniamo a promuovere un approccio all'apprendimento e all'innovazione guidato dalla comunità. Per maggiori dettagli sui diritti e sulle limitazioni di questa licenza, fare riferimento alla documentazione ufficiale [GPL-3.0](#).

## Download della Versione PDF dell'eBook

### Xiao: Big Power, Small Board - Edizione PDF

Siamo entusiasti di offrire la versione PDF di "Xiao: Big Power, Small Board", un eBook approfondito dedicato alla scheda di sviluppo Xiao. Questa edizione è pensata per la lettura digitale, offrendo un formato accessibile per coloro che preferiscono i dispositivi elettronici. Che si sia hobbisti, studenti o un professionisti, questa guida è progettata per aiutare a esplorare tutte le capacità della scheda Xiao in un comodo formato PDF.

### Come Eseguire il Download

Per scaricare l'edizione PDF dell'eBook, cliccare semplicemente sul link qui sotto:

[Download dell'Edizione PDF \(in inglese\)- Xiao: Big Power, Small Board](#)

[Download dell'Edizione PDF \(in Italiano\)- Xiao: Big Power, Small Board](#)

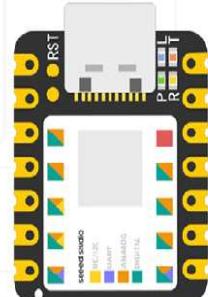
### Anteprima eBook (versione PDF inglese)

Di seguito sono riportati alcuni screenshot della versione PDF dell'eBook, che offrono un'anteprima del contenuto:

# XIAO: Big Power, Small Board

Mastering Arduino and TinyML

GPL-3.0 license



Lei Feng, Marcelo Rovai

seeed studio



2024.01

## 1.2 Using the Button Switch on the XIAO Expansion Board to Control an LED Light

In the previous section, we learned how to control an LED light to blink using only the Seed Studio XIAO and the onboard LED light. However, there was no interaction with the external environment, such as controlling the LED light through light or sound. In this section, we will introduce a simple sensor - the button switch, to form an automatic control system of sensor-controller-actuator. Before starting the task, we need to learn some basic knowledge, like what variables are and the common program structures, so that we can better understand and run the program.

### 1.2.1 Background Knowledge

In the last section, we only used the onboard LED light of the Seed Studio XIAO without connecting other modules. It could take quite some effort for beginners to use Dupont wires to connect external sensors to a board the size of a thumb and also involve a breadboard. Is there a simpler method?

#### Seed Studio XIAO Expansion Board

The Seed Studio XIAO Expansion Board, only half the size of Raspberry Pi 4, is powerful and can quickly and easily build prototypes and projects. The board has a variety of peripherals such as OLED, RTC, expandable memory, passive buzzer, RESET/User button, 5V servo/sensor connector, various data interfaces... You can explore the infinite possibilities of Seed Studio XIAO. The board also supports [CircuitPython](#).

All models in the Seed Studio XIAO series have uniform specifications and support the Seed Studio XIAO Grove Shield and Seed Studio XIAO Expansion Board. The series includes XIAO SAMD21, XIAO RP2040, XIAO nRF52840, XIAO nRF52840 Sense, XIAO ESP32C3 and XIAO ESP32S3. The front and back function interfaces of the XIAO expansion board are shown in the following figure:

24 | XIAO: Big Power, Small Board - Mastering Arduino and TinyML

To make it easier and quicker to build projects with Seed Studio XIAO, we equipped it with a powerful expansion board. This board has a wealth of onboard peripherals and can quickly connect to more electronic modules to implement various functions. The expansion board brings out all the pins of XIAO, as shown in the pin diagram below.

### Pinout

In most cases, the XIAO expansion board is suitable for all Seed Studio XIAO series products. When we need to use the XIAO expansion board, we need to connect the XIAO development board to the corresponding position on the expansion board, as shown in the figure below. Connect the pin headers on the XIAO main board to the position circled in yellow on the expansion board. Be sure to align it before pressing down to avoid damaging the pins. After that, we can start working on projects in combination with the expansion board.

**Attention:**

Please first plug the Seed Studio XIAO into the two female headers on the expansion board, and then plug in the Type-C power supply, otherwise it will damage the Seed Studio XIAO and the expansion board.

Chapter 1: introduction to hardware and programming | 25

## Indice - Edizione PDF

## 4.2 Anomaly Detection & Motion Classification

**4.2.1 Things used in this project**

**Hardware components**

Seed Studio XIAO nRF52840 Sense x 1

**Software apps and online services**

- Arduino IDE
- Edge Impulse Studio

**4.2.2 Introduction**

As you learned in the previous section, microcontrollers (MCUs) are very cheap electronic components, usually with just a few kilobytes of RAM, designed to use tiny amounts of energy. They can be found in almost any consumer, medical, automotive, and industrial device. Over 40 billion microcontrollers will be sold this year, and there are probably hundreds of billions in service nowadays. However, these devices get little attention because they're often only used to replace the functionality of older electro-mechanical systems in cars, washing machines, or remote controls. More recently, with the Internet of Things (IoT) era, a significant part of those MCUs is generating "quintillions" of data that, in its majority, is not used due to the high cost and complexity (bandwidth and latency) of data transmission.

On the other hand, in recent decades, we have seen a lot of development in Machine Learning models trained with vast amounts of data in very powerful and power-hungry mainframes. And what is happening today is that due to those developments, it is now possible to take noisy signals like images, audio, or accelerometers and extract meaning from them by using Machine Learning algorithms such as Neural Networks.

And what is more important is that we can run these algorithms on microcontrollers and sensors themselves using very little power, interpreting much more of those sensor data that we are currently ignoring. This is TinyML, a new technology that enables machine intelligence right next to the physical world.

178 | XIAO: Big Power, Small Board - Mastering Arduino and TinyML

TinyML can have many exciting applications for the benefit of society at large. This section will explore TinyML, running on a robust and tiny device, the [Seed XIAO nRF52840 Sense](#) (also called XIAO BLE Sense).

### 4.2.3 XIAO nRF52840 Sense

**MainFeatures**

- Bluetooth 5.0 with onboard antenna
- CPU: Nordic nRF52840, ARM® Cortex®-M4 32-bit processor with FPU, 64 MHz
- Ultra-Low Power: Standby power consumption is less than 5µA
- Battery charging chip: Supports lithium battery charge and discharge management
- 2 MB flash
- 256 KB RAM
- PDM microphone
- 6-axis LSM6DS3TR-C IMU
- Ultra Small Size: 20 x 17.5mm, XIAO series classic form-factor for wearable devices
- Rich interfaces: 1xUART, 1xI2C, 1xSPI, 1xNFC, 1xSWD, 1xGPIO(PWM), 6xADC
- Single-sided components, surface mounting design

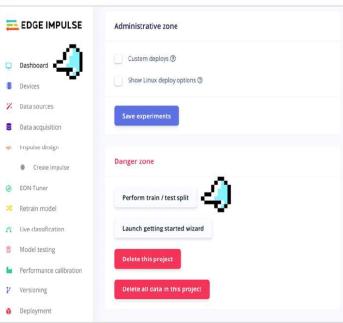
**Connecting the XIAO nRF52840 Sense with Arduino IDE**

The easiest way to test and use this device is using the [Arduino IDE](#). Once you have the IDE installed on your machine, navigate to [File > Preferences](#), and fill in "Additional Boards Manager URLs" with the URL below: [https://files.seedstudio.com/arduino/package\\_seeduno\\_boards\\_index.json](https://files.seedstudio.com/arduino/package_seeduno_boards_index.json)

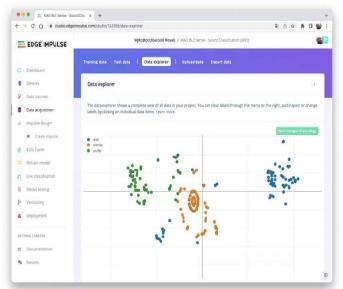
Chapter 4: project practice advanced - tinyml applications | 179

## Capitolo Introduttivo - Edizione PDF

Now, you should split that dataset into Train/Test. You can do it manually (using the three dots menu, moving samples individually) or using **Perform Train / Test Split** on Dashboard – Danger Zone.

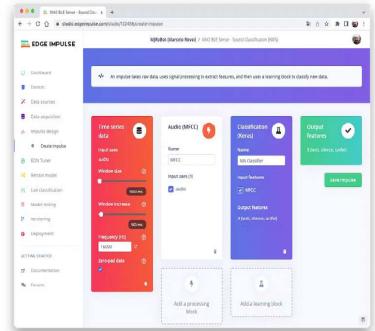


We can optionally check all datasets using the tab Data Explorer. The data points seem apart, which means that the classification model should work:



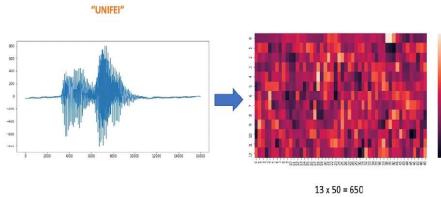
**Creating Impulse (Pre-Process / Model definition)**

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.



First, we will take the data points with a 1-second window, augmenting the data, sliding that window each 500ms. Note that the option zero-point pad is set. It is important to fill with zeros samples smaller than 1 second in some cases, I reduced the 1000 ms window on the split tool to avoid noises and spikes.

Each 1-second audio sample should be pre-processed and converted to an image (for example,  $13 \times 50 \times 1$ ). We will use **Audio (MFCC)**, which extracts features from audio signals using Mel Frequency Cepstral Coefficients, which are well suited for the human voice, which is our case here.



Next, we select the **Classification** block to build our model from scratch using a Convolution Neural Network (CNN).

**Pre-Processing (MFCC)**

The next step is to create the images to be trained in the next phase:

## Capitolo Introduttivo - Edizione PDF

That's it! You can save the images directly on your computer to be used on projects.

### 4.4.9 Fruits versus Veggies - A TinyML Image Classification Project

Now that we have an embedded camera running, it is time to try image classification. For comparative motive, we will replicate the same image classification project developed to be used with an old ESP2-CAM:

ESP32-CAM: TinyML Image Classification - Fruits vs Veggies



The whole idea of our project will be to train a model and proceed with inference on the XIAO ESP32S3 Sense. For training, we should find some data ([in fact, tons of data!](#)).

But first of all, we need a goal! What do we want to classify?

With TinyML, a set of techniques associated with machine learning inference on embedded devices, we should limit the classification to three or four categories due to limitations (mainly memory). We will differentiate **apples** from **bananas** and **potatoes** (you can try other categories).

So, let's find a specific dataset that includes images from those categories. Kaggle is a good start:

<https://www.kaggle.com/kritikseth/fruit-and-vegetable-image-recognition>

This dataset contains images of the following food items:

- **Fruits** - banana, apple, pear, grapes, orange, kiwi, watermelon, pomegranate, pineapple, mango.
- **Vegetables** - cucumber, carrot, capsicum, onion, potato, lemon, tomato, radish, beetroot, cabbage, lettuce, spinach, soybean, cauliflower, bell pepper, chili pepper, turnip, corn, sweetcorn, sweet potato, paprika, jalapeño, ginger, garlic, peas, eggplant.

Each category is split into the **train** (100 images), **test** (10 images), and **validation** (10 images).

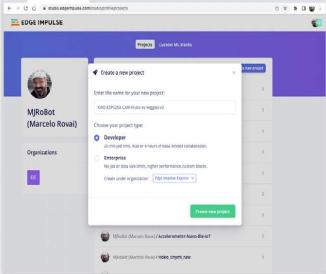
- Download the dataset from the Kaggle website to your computer.

*Optionalmente, puoi aggiungere alcune foto fresche di banane, mele, e patate dalla tua cucina domestica, utilizzando, per esempio, i codici discusso nella sezione precedente.*

### 4.4.10 Training the model with Edge Impulse Studio

We will use the Edge Impulse Studio to train our model. As you know, [Edge Impulse](#) is a leading development platform for machine learning on edge devices.

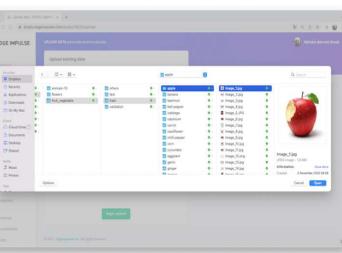
Enter your account credentials (or create a free account) at Edge Impulse. Next, create a new project:



**Data Acquisition**

Next, on the UPLOAD DATA section, upload from your computer the files from chosen categories:

It would be best if you now had your training dataset split into three classes of data:



It would be best if you now had your training dataset split into three classes of data:

240 | XIAO: Big Power, Small Board - Mastering Arduino and TinyML | 241

## Capitolo Introduttivo - Edizione PDF

Abbiamo creato questa edizione PDF per garantire un'esperienza di lettura fluida, ricca di spunti e informazioni preziose. Se si hanno domande o si necessita di ulteriori informazioni, non esitare a contattarci.

Buona lettura e speriamo che si trovi l'eBook informativo e stimolante!

---

Per risorse aggiuntive e maggiori informazioni sulla scheda di sviluppo Xiao, visitare il nostro sito Web ufficiale: [https://mjrovai.github.io/XIAO\\_Big\\_Power\\_Small\\_Board-ebook/](https://mjrovai.github.io/XIAO_Big_Power_Small_Board-ebook/)

## 1.1 Primo Programma Arduino con Seeed Studio XIAO: Blink

Arduino è una piattaforma di prototipazione elettronica open source popolare a livello mondiale, che include vari modelli di schede di sviluppo Arduino e la piattaforma software Arduino IDE. Grazie alle sue caratteristiche aperte, comode e facili da avviare, è diventata la prima scelta per molti principianti di software e hardware.

Con essa, si può completare rapidamente lo sviluppo del progetto e implementare le proprie idee. Ad oggi, Arduino ha introdotto vari modelli di controller e numerosi moduli periferici, come sensori, attuatori, schede di espansione, ecc. Questi moduli possono implementare vari progetti entusiasmanti e pratici se utilizzati con Arduino.

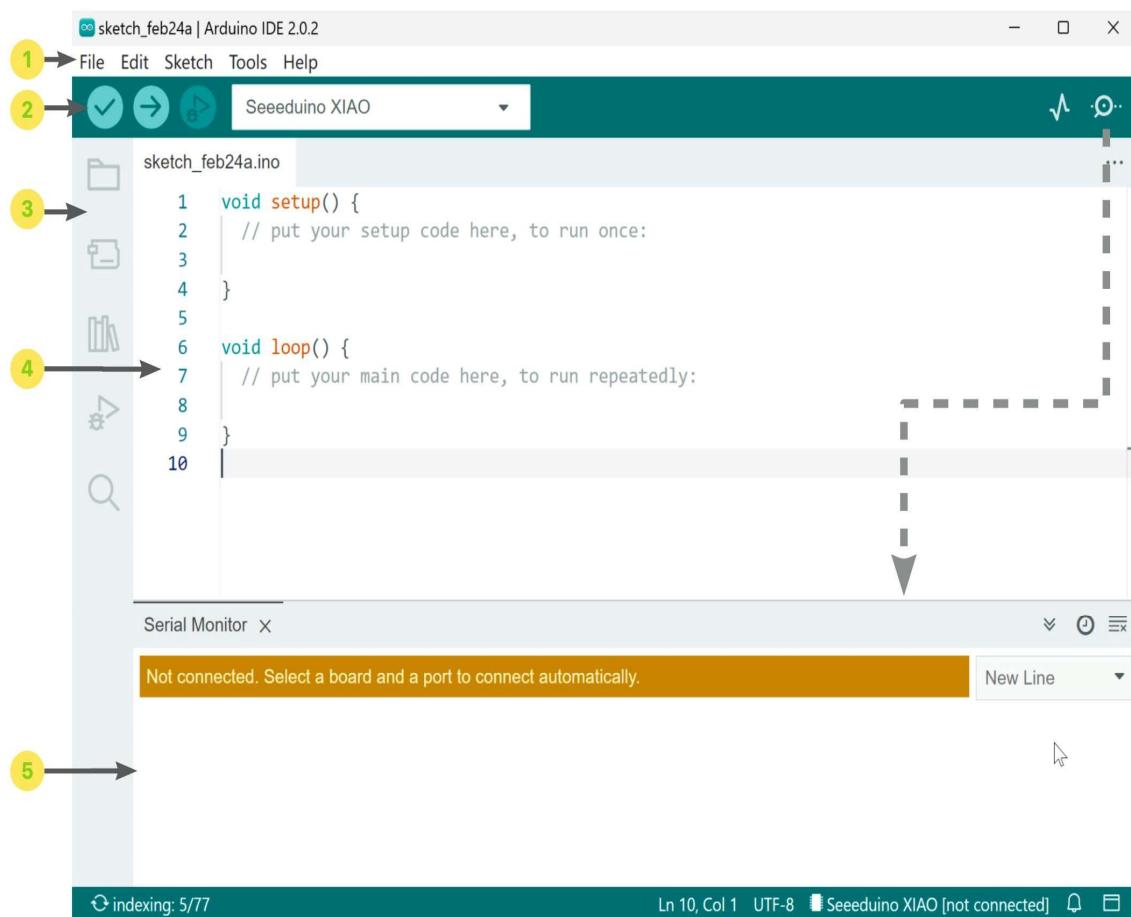
I prodotti della serie Seeed Studio XIAO di cui stiamo parlando oggi sono schede di sviluppo derivate da Arduino. Appartengono alla serie Seeeduino e sono i membri più piccoli di questa serie.

## 1.1.1 Arduino IDE - l'Editor di Testo

Dobbiamo programmare l'hardware tramite l'editor di testo Arduino IDE. Se non è già installato Arduino IDE, si va alla pagina di download per installarlo: [Software](#). Arduino IDE (Integrated Development Environment) è un software di programmazione progettato esplicitamente per Arduino. Tramite esso, possiamo scrivere e caricare diversi programmi per l'hardware Arduino. Quando apriamo il software Arduino IDE, verrà creato un nuovo file denominato Sketch, che possiamo rinominare.

### Per gli Utenti Windows

L'interfaccia di Arduino IDE è impeccabile e può essere divisa in quattro parti: barra dei menu, barra degli strumenti, area di modifica e finestra di debug.



**Barra dei Menù:** Include files, edit, sketch, tools e help, come new, save, example programs, select serial port, ecc.

**Barra degli Strumenti Orizzontale:** Contiene diversi pulsanti funzione comunemente usati: verify, upload, debug, selezione della board, serial plotter e serial monitor.

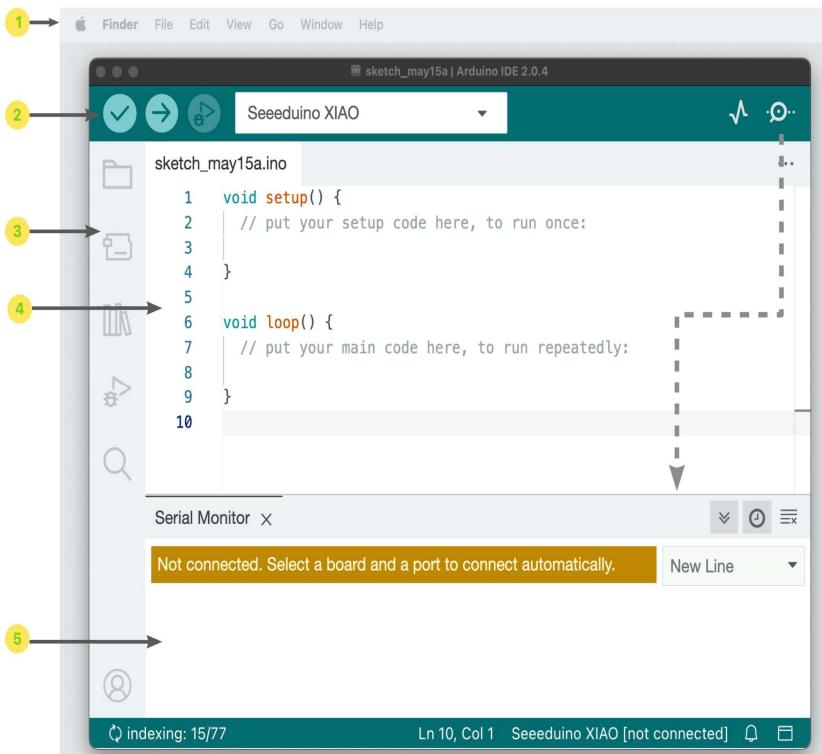
**Barra degli Strumenti Verticale:** Contiene collegamenti alla cartella del progetto, gestore scheda, gestore libreria, debug e ricerca.

**Area di Modifica del Codice:** Qui è dove si scrive il codice del programma, proprio come di solito digitiamo il testo in una finestra di Word. Scrivere il codice del programma in quest'area.

**Monitor seriale, finestra di output:** sul lato destro della barra degli strumenti orizzontale, si può aprire o chiudere la finestra del monitor seriale.

### Per gli Utenti MAC

Ad eccezione della posizione della barra dei menù (in alto), che è leggermente diversa rispetto agli utenti Windows, tutti gli altri strumenti e le esperienze sono gli stessi.



## 1.1.2 Aggiungere Seeed Studio XIAO all’IDE Arduino

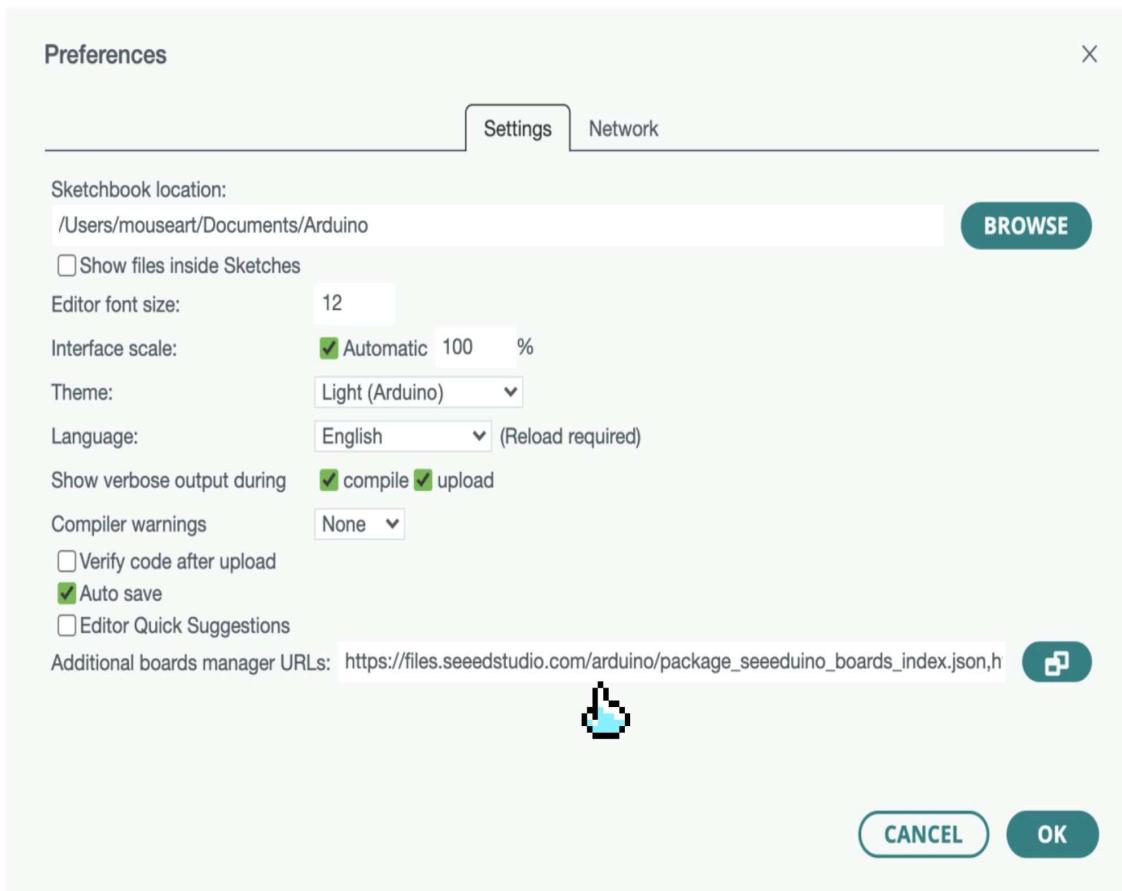
### Attenzione

> A causa delle limitazioni di spazio, tutte le parti del codice del programma e della connessione hardware di questo corso si basano su Seeed Studio **XIAO SAMD21**. La maggior parte del codice nel libro può essere applicata a tutti i prodotti della serie Seeed Studio XIAO. Se ci sono eccezioni, saranno ulteriormente contrassegnate o spiegate per l’hardware applicabile. Se non contrassegnate, si applicano a più prodotti.

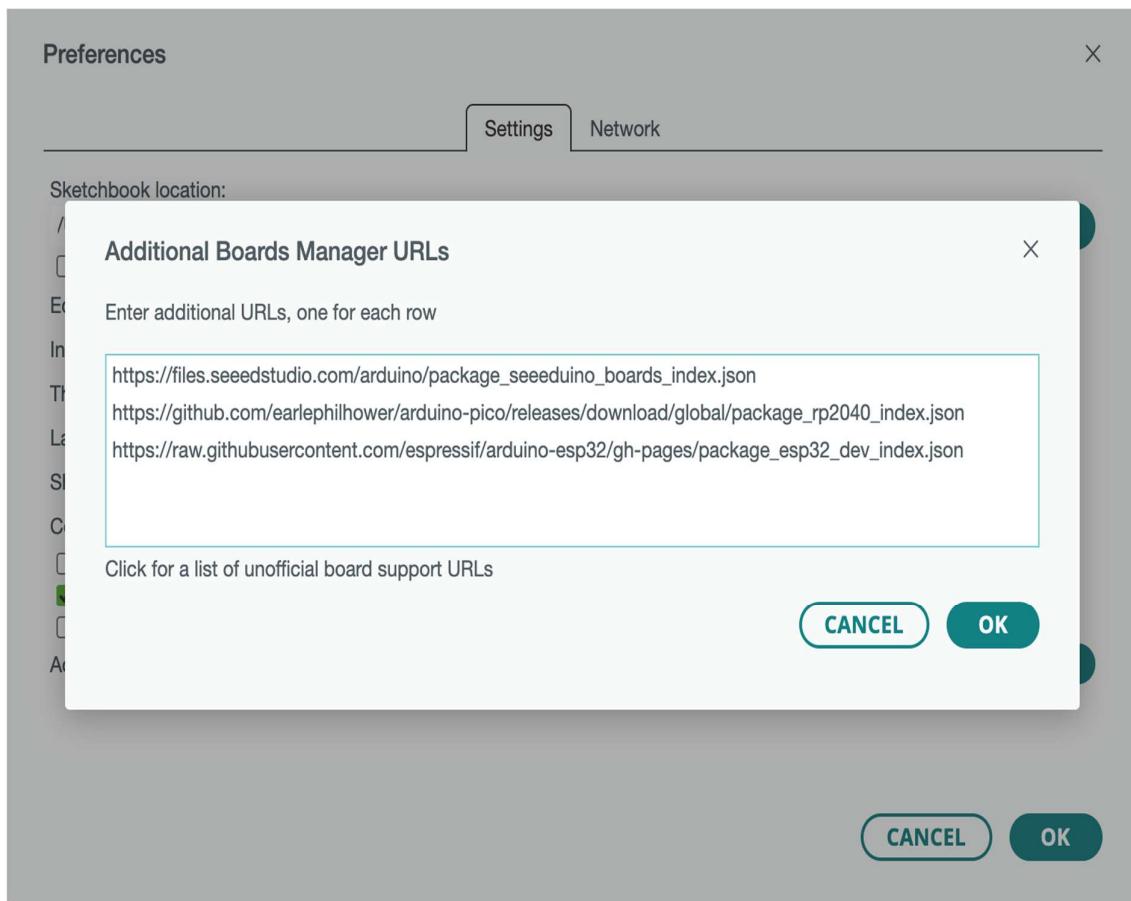
Dobbiamo aggiungere i prodotti della serie Seeed Studio XIAO all’IDE Arduino per iniziare il nostro percorso di apprendimento.

- Per gli utenti Windows, per prima cosa, si apre l’IDE Arduino, si clicca su “File→Preferences” nella barra dei menù in alto, come mostrato nella figura, e si copia il seguente URL in “Additional Boards Manager URLs.”
- Per gli utenti Mac, per prima cosa, si apre l’IDE Arduino, si clicca su “Arduino IDE→Preferences” nella barra dei menù in alto, come mostrato nella figura, e si copia il seguente URL in “Additional Boards Manager URLs.”
- Per Seeed Studio XIAO SAMD21, XIAO nRF52840 e XIAO nRF52840 Sense, copiare l’indirizzo del link qui sotto:  
[https://files.seeedstudio.com/arduino/package\\_seeeduino\\_boards\\_index.json](https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json)
- Per Seeed Studio XIAO RP2040, copiare l’indirizzo del link qui sotto:  
[https://github.com/earlephilhower/arduino-pico/releases/download/global/package\\_rp2040\\_index.json](https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json)

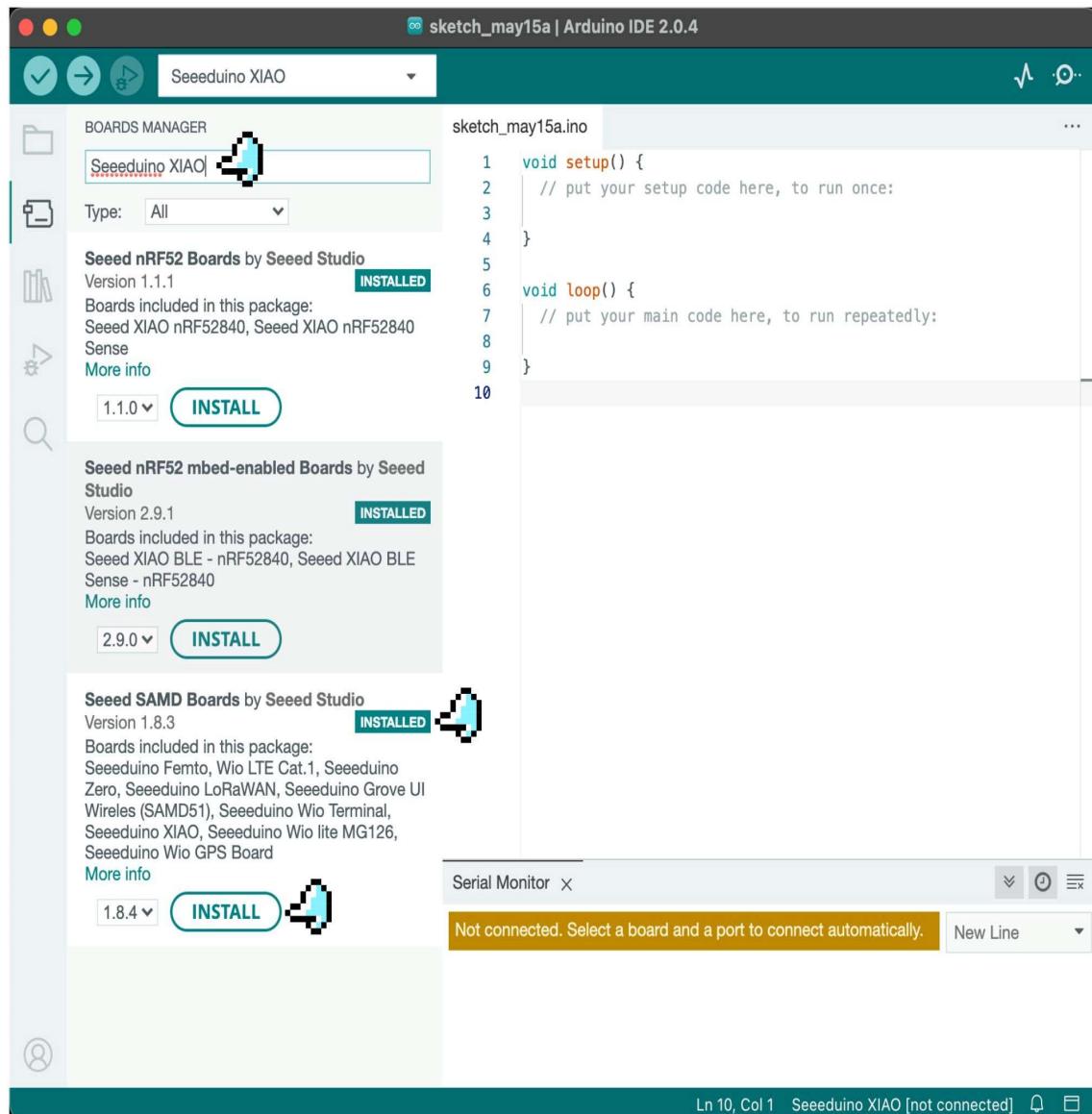
- Per Seeed Studio XIAO ESP32C3, XIAO ESP32S3, copiare l'indirizzo del link qui sotto:  
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)



Se si usano spesso più modelli diversi di XIAO contemporaneamente, si può cliccare sull'icona sul lato destro della barra degli indirizzi e aggiungere tutti e tre gli indirizzi sopra al gestore della scheda, come mostrato nella figura qui sotto.



Quindi, si clicca su “Tools→Board→Board Manager”, si inserisce la parola chiave `seeeduino Xiao` nella barra di ricerca, si trova `seeed SAMD Boards` nelle voci visualizzate e si clicca su `INSTALL`.



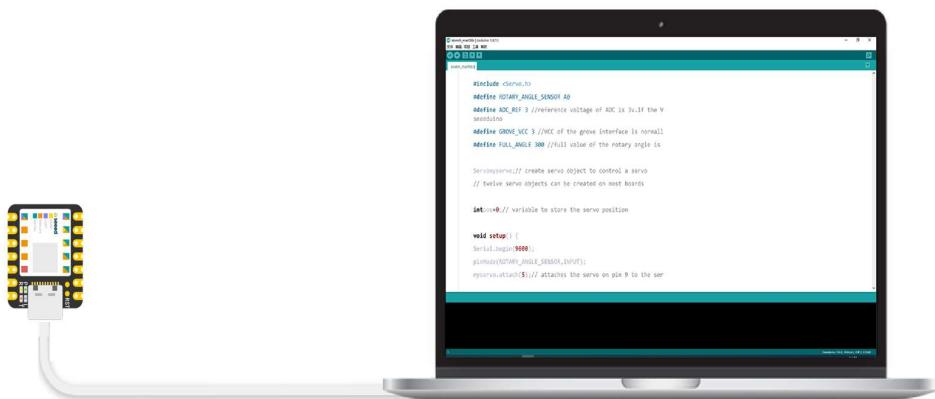
Quando l'installazione inizia, si vedrà una finestra pop-up di output. Dopo aver completato l'installazione, apparirà un'opzione "INSTALLED".

### Attenzione

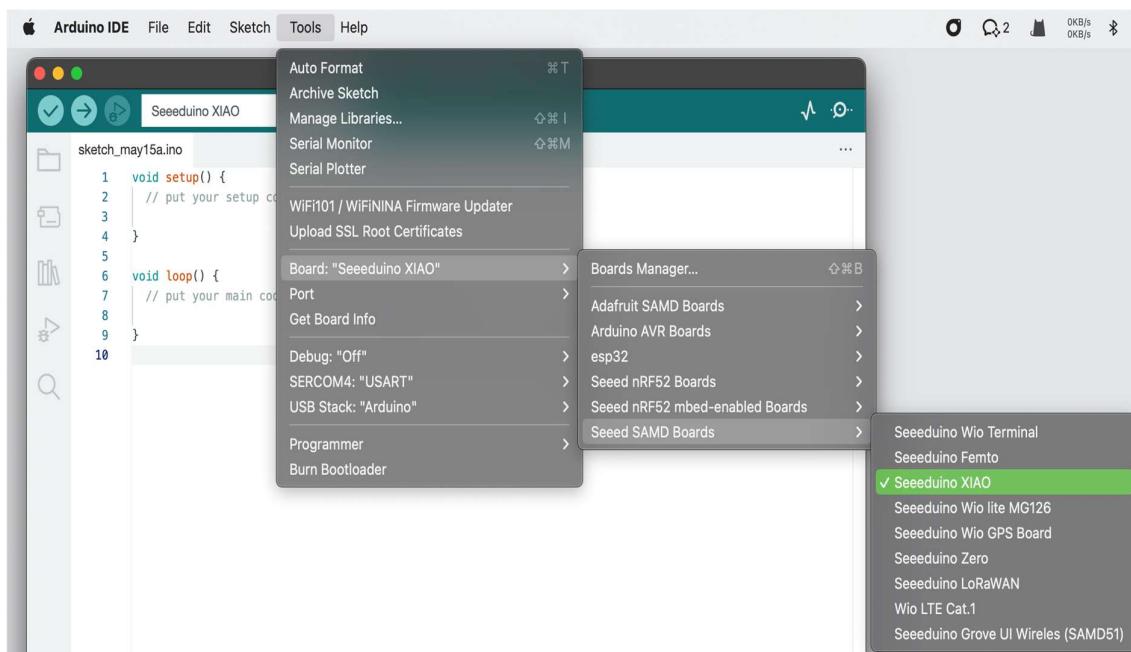
- Si inserisce "RP2040" nella barra di ricerca per trovare il pacchetto di installazione per Seeeduino XIAO RP2040.
- Si inserisce "XIAO nrf52840" per trovare due pacchetti di installazione: Seeed nRF52 Boards (per progetti a bassa potenza) e Seeed nRF52 mbed-enabled Boards (per progetti TinyML ad alta potenza).
- Si inserisce "ESP32" per trovare il pacchetto di installazione per ESP32 di Espressif Systems.

## Collegamento di Seeed Studio XIAO ad Arduino IDE

Collegare XIAO al computer con un cavo dati, come mostrato nella figura seguente:



Quindi, fai clic su “Tools→Board”, si cerca “Seeeduino XIAO,” e lo si seleziona, come mostrato nella figura seguente.



### ⚠ Attenzione

Se la scheda di sviluppo in uso è XIAO nRF52840, selezionare Seeed XIAO nrf52840.  
Se la tua scheda di sviluppo è XIAO nRF52840 Sense, selezionare Seeed XIAO nrf52840 Sense. Se la tua scheda di sviluppo è XIAO RP2040, selezionare Seeed XIAO RP2040. Se la tua scheda di sviluppo è XIAO ESP32C3, selezionare XIAO\_ESP32C3. Se la tua scheda di sviluppo è XIAO ESP32S3, selezionare XIAO\_ESP32S3.

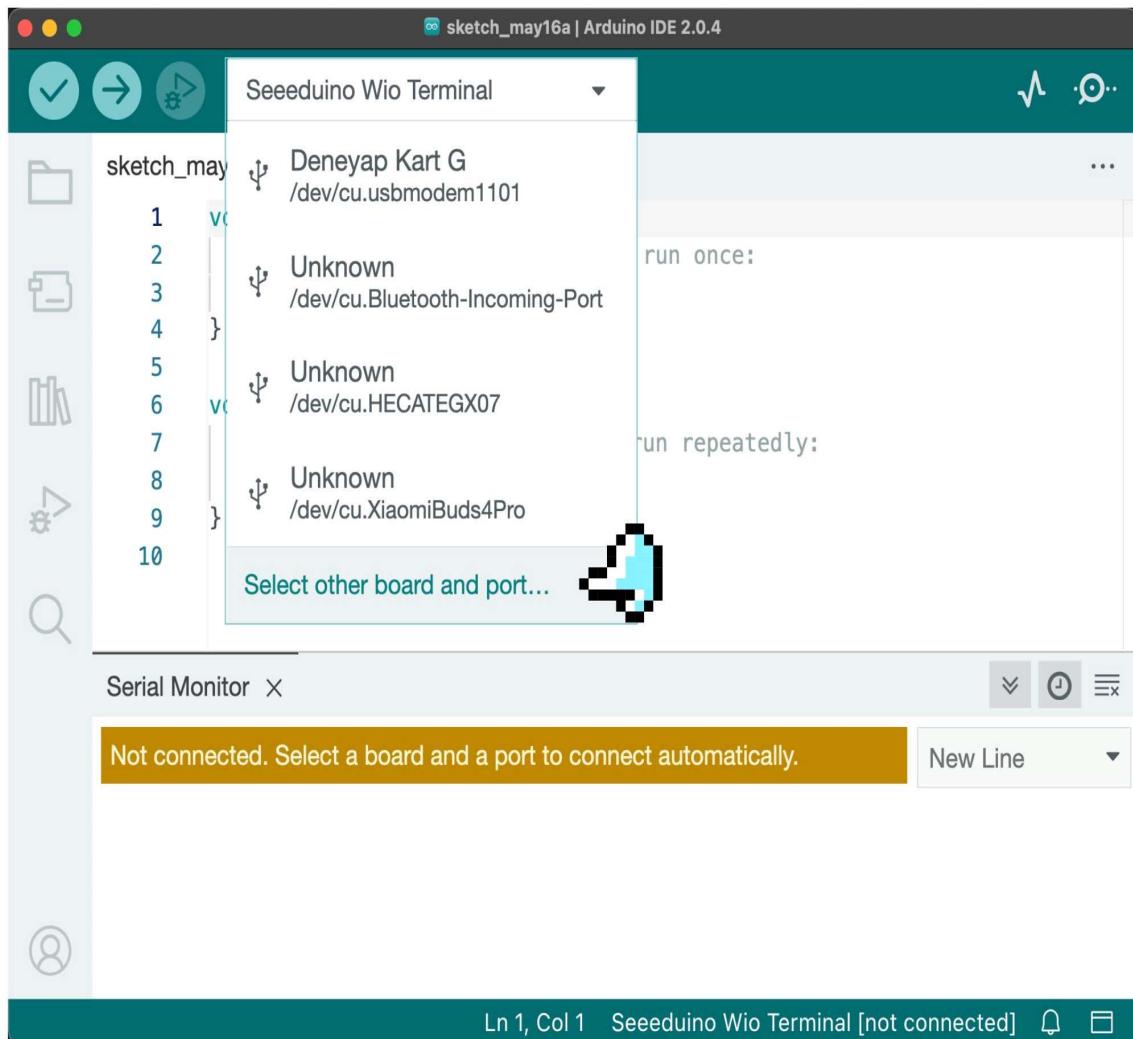
Controllare che la connessione della porta sia corretta; in caso contrario, selezionarla manualmente.

- La porta seriale sui sistemi Windows è visualizzata come “COM+numero”, come mostrato nella figura sottostante.

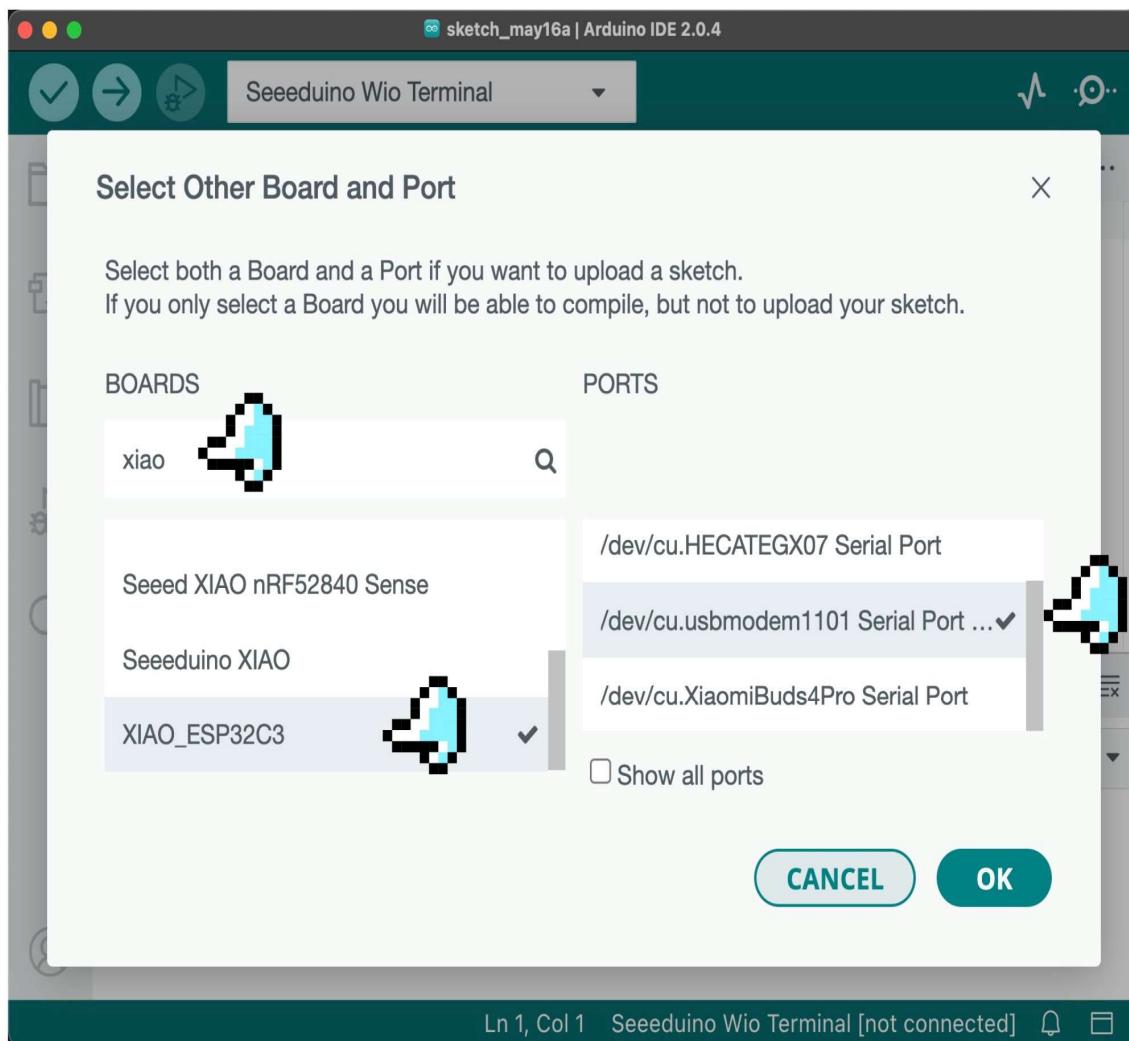
- Il nome della porta seriale sui sistemi Mac o Linux è generalmente /dev/tty.usbmodem+numero o /dev/cu.usbmodem+numero, come mostrato nella figura sottostante.

Ora, possiamo iniziare a programmare XIAO tramite il software.

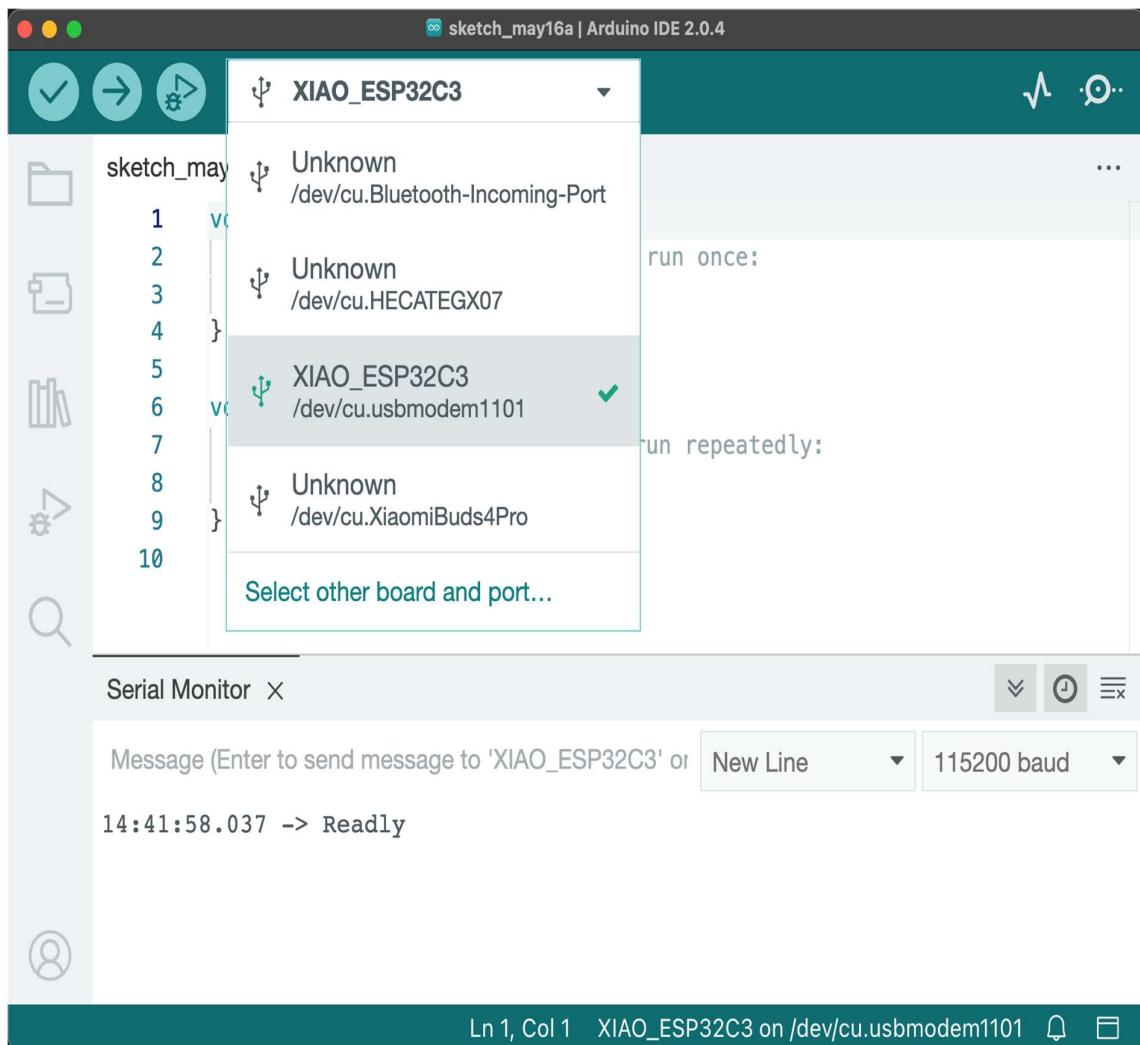
**⚠️ XIAO ESP32C3 potrebbe non essere riconosciuta adeguatamente in Arduino IDE 2, ed è necessario specificare manualmente la scheda di sviluppo e la porta.**



Quando ESP32C3 è collegato a un PC con Arduino IDE 2, potrebbe non essere in grado di abbinare automaticamente la scheda di sviluppo corretta. Come mostrato nella figura sottostante, quanto mostrato non è la scheda di sviluppo XIAO ESP32; è necessario specificarlo manualmente. Selezionare “Other Board & Port...” dal menu a discesa “Port”. Immettere “xiao” nella barra di ricerca della scheda di sviluppo, selezionare la scheda di sviluppo XIAO\_ESP32C3 dall’elenco filtrato sottostante e confermare dopo aver selezionato la porta sulla destra.



Ora si può vedere che la scheda di sviluppo e la porta sono nello stato corretto.



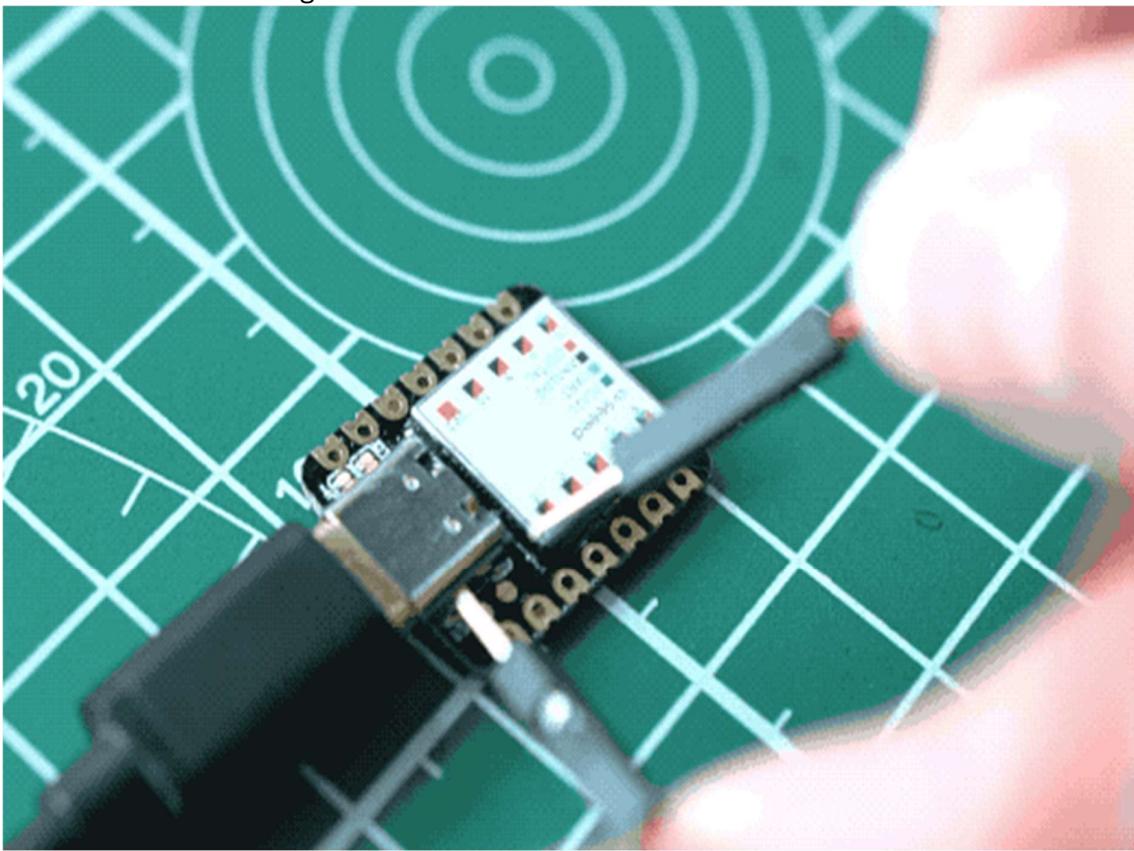
### ⚠️ Reset di Seeed Studio XIAO

A volte, quando il caricamento del programma fallisce, la porta Seeed Studio XIAO potrebbe scomparire e dobbiamo eseguire un'operazione di reset. Il metodo di reimpostazione sarà diverso per i diversi modelli di XIAO.

#### **Reset di Seeed Studio XIAO SAMD21**

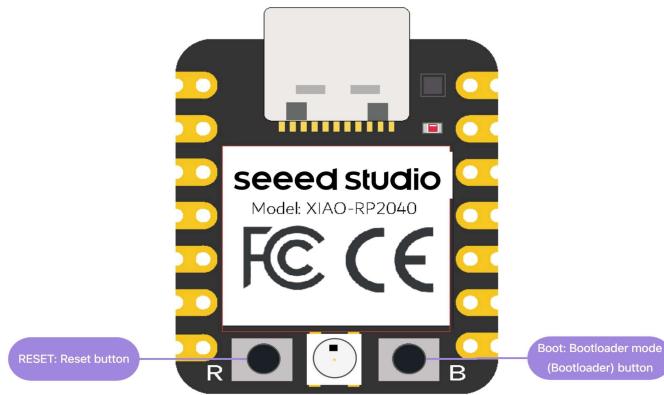
- Collegare XIAO SAMD21 al computer.
- Aprire “Blink” nel programma di esempio Arduino IDE e cliccare su upload.
- Durante il caricamento, cortocircuitare una volta il pin RST nella figura con delle pinzette o un filo.
- Il reset è completato quando il LED arancione lampeggia e si accende.

Come mostrato nella figura sottostante.



#### **Reset di Seeed Studio XIAO PR2040**

- Collegare Seeed Studio XIAO RP2040 al computer.
- Premere una volta il pulsante di reset contrassegnato con “R”, la posizione è mostrata nella figura sottostante.



Se questo non funziona, tenere premuto il pulsante di avvio contrassegnato con “B”, collegare la scheda al computer tenendo premuto il pulsante BOOT, poi rilascialo per entrare in modalità bootloader.

#### **Reset di Seeed Studio XIAO nRF52840 e della versione Sense**

- Collegare Seeed Studio XIAO nRF52840, o la versione Sense, al computer.
- Premere una volta il pulsante di reset contrassegnato con “RST”, la posizione è mostrata nella figura sottostante.

Se non funziona, ci si può cliccare rapidamente due volte per entrare in modalità bootloader.

### **Reset di Seeed Studio XIAO ESP32C3**

- Collegare Seeed Studio XIAO ESP32C3 al computer.
- Premere una volta il pulsante di reset contrassegnato con “R”, la posizione è mostrata nella figura sottostante.



Se questo non funziona, tenere premuto il pulsante di avvio contrassegnato con “B”, collegare la scheda al computer tenendo premuto il pulsante BOOT, poi rilascialo per entrare in modalità bootloader.

### **Reset di Seeed Studio XIAO ESP32S3**

- Collegare Seeed Studio XIAO ESP32S3 al computer.
- Premere una volta il pulsante di reset contrassegnato con “R”, la posizione è mostrata nella figura sottostante.



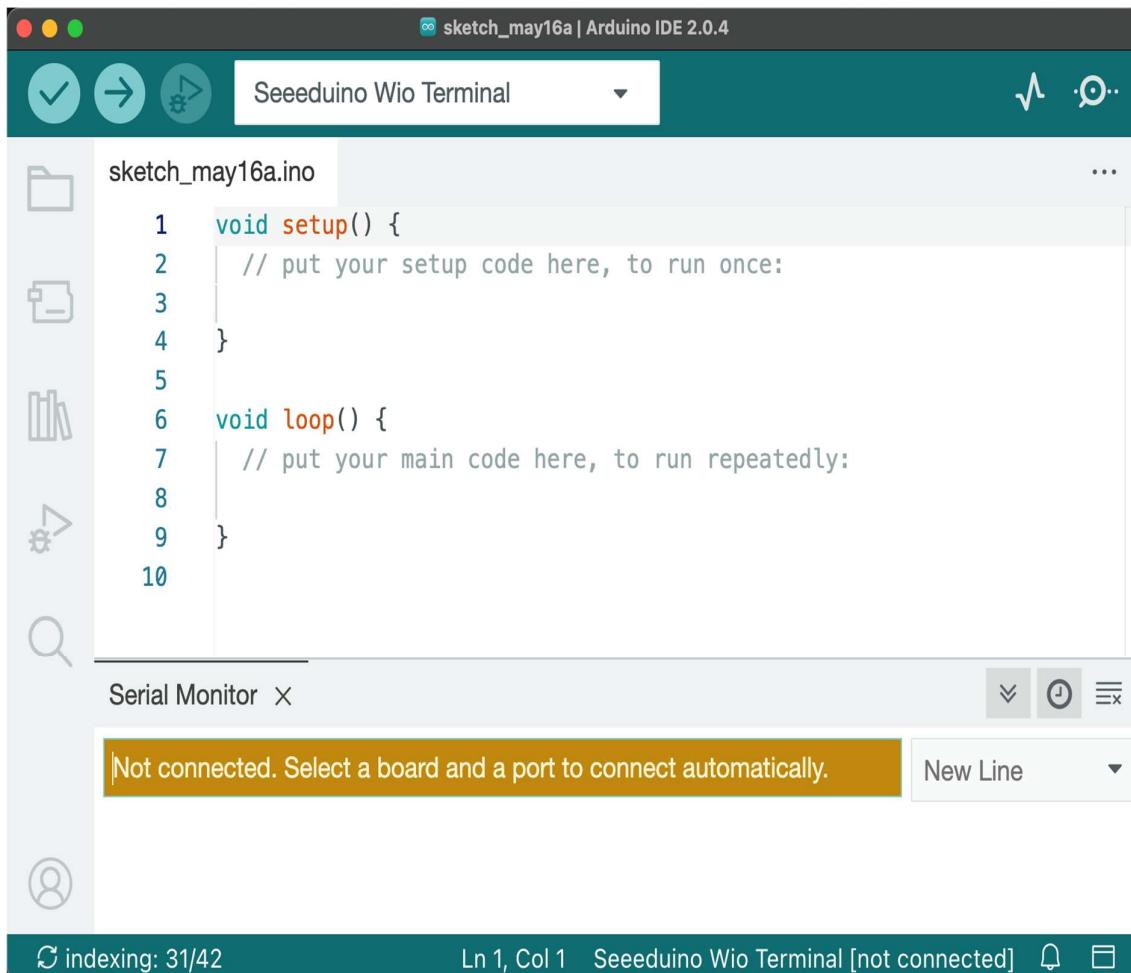
Se questo non funziona, tenere premuto il pulsante di avvio contrassegnato con “B”, collegare la scheda al computer tenendo premuto il pulsante BOOT, poi rilascialo per entrare in modalità bootloader.

### **Struttura dei Programmi Arduino**

Ora che abbiamo la scheda di sviluppo, come possiamo scrivere programmi al suo interno per controllarne le funzioni? Ecco quando l'editor di testo Arduino IDE torna utile. Abbiamo già presentato le funzioni di interfaccia di Arduino IDE nell'introduzione, è uno strumento importante per scrivere e caricare programmi. I programmi Arduino sono costituiti da due funzioni di base:

`setup()` Questa funzione viene chiamata all'inizio del programma. La si usa per inizializzare variabili, modalità pin, iniziare a usare librerie, ecc. `setup()` viene eseguita solo una volta, quando la scheda Arduino viene accesa o resettata.

`loop()` Dopo che il programma in `setup()` è stato eseguito, quello in `loop()` inizia l'esecuzione. Il programma in `loop()` viene eseguito ripetutamente.

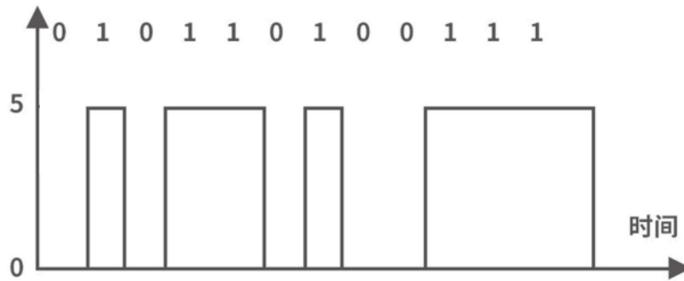


### Cosa da sapere:

- Il contenuto dopo “`/* */` e “`//`” sono commenti che aiutano a comprendere e gestire il codice, i commenti non influiranno sul normale funzionamento del programma;
- Quando si scrivono programmi, dobbiamo usare “`{}`” per racchiudere un set di codici;
- Dopo ogni riga di codice, si usa “`;`” come simbolo di fine per dire all'editor Arduino che questa riga di istruzioni di codice è terminata.

### Segnali Digitali e Impostazioni di I/O

In parole poche, i segnali digitali sono segnali rappresentati in forma binaria di 0 e 1. In Arduino, i segnali digitali sono rappresentati da livelli alti e bassi, il livello alto indica il segnale digitale 1 e il livello basso indica il segnale digitale 0. Seeed Studio XIAO ha 11 pin digitali, possiamo impostare questi pin per eseguire la funzione di input o output di segnali digitali.



In Arduino, si possono usare le funzioni per impostare lo stato e la funzione dei pin. Ecco i passaggi di base per impostare i pin tramite funzioni:

1. Per prima cosa, si determina il numero del pin da controllare.
2. Nel codice Arduino, si usa la funzione `pinMode()` per impostare la funzione del pin, come input o output. Ad esempio, per impostare il pin in modalità output, si usa il seguente codice:

```
int ledPin = 13; // Il pin da controllare
void setup() {
    pinMode(ledPin, OUTPUT); // Imposta il pin in modalità output
}
```

3. Dopo aver impostato il pin in modalità output, si usa la funzione `digitalWrite()` per impostare lo stato del pin, ad esempio impostandolo su livello alto o basso. Ad esempio, per impostare il pin su livello alto, si usa il seguente codice:

```
digitalWrite(ledPin, HIGH); // Imposta il pin su livello alto
```

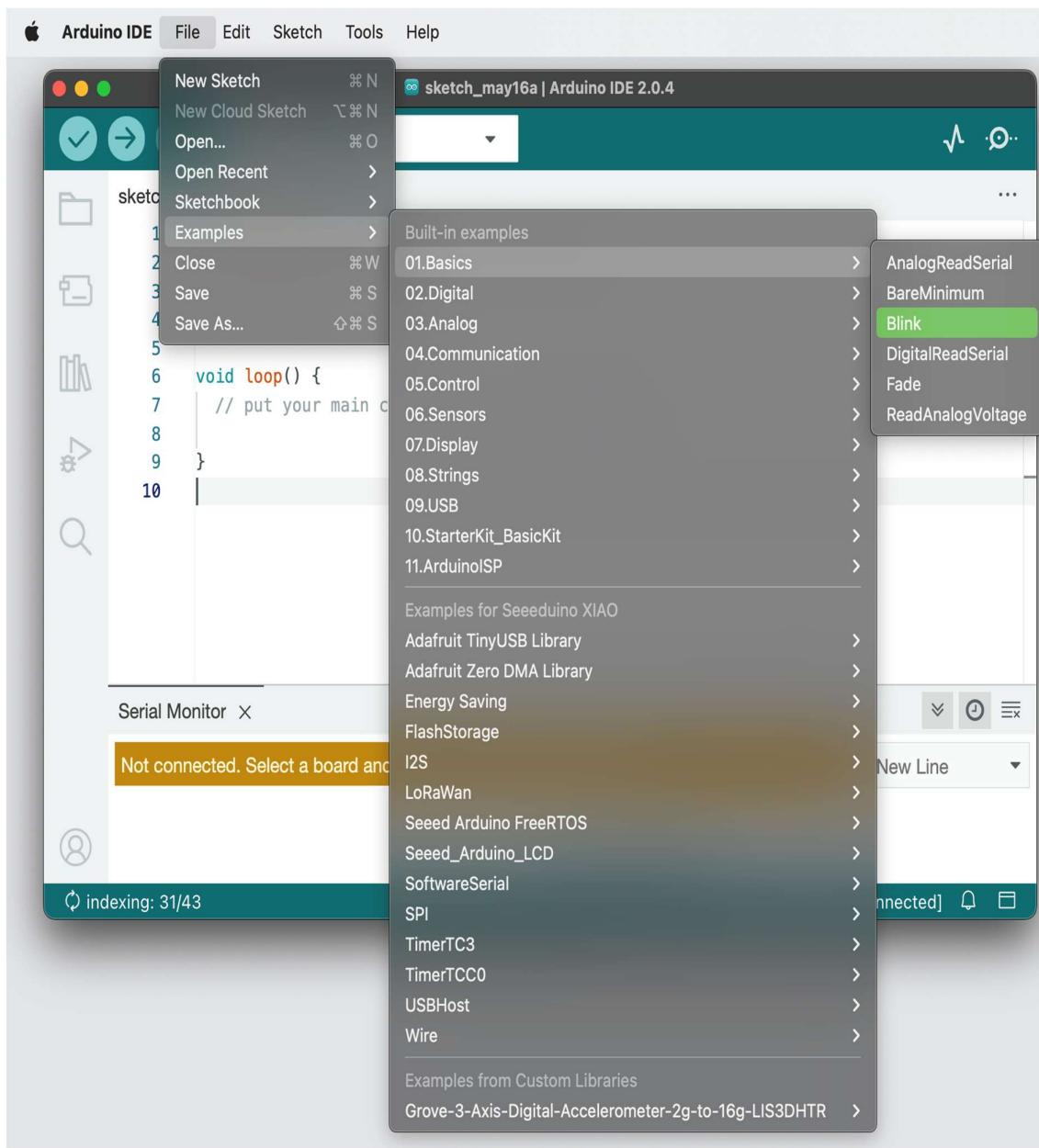
4. Se si imposta il pin su modalità input, si usa la funzione `digitalRead()` per leggere lo stato del pin, ad esempio per rilevare se è il livello è alto o basso. Ad esempio, per leggere lo stato del pin e salvarlo in una variabile, si usa il seguente codice:

```
int buttonPin = 2; // Il pin da cui leggere lo stato
int buttonState = 0; // La variabile in cui salvare lo stato
void setup() {
    pinMode(buttonPin, INPUT); // Imposta il pin in modalità input
}
void loop() {
    buttonState = digitalRead(buttonPin); // Legge lo stato del pin
}
```

Utilizzando funzioni come `pinMode()`, `digitalWrite()` e `digitalRead()`, si possono facilmente impostare e controllare lo stato e la funzione dei pin in Arduino.

### 1.1.3 Attività 1: Eseguire Blink per far lampeggiare il LED di XIAO

Così come “Hello World” è la prima sezione in tutti i linguaggi di programmazione, “Blink” è simile a “Hello World” nella programmazione Arduino. È la chiave del nostro viaggio nell'apprendimento di Arduino. Arduino fornisce molti codici di esempio per aiutarci a iniziare rapidamente e Blink è uno di questi. Possiamo selezionare “File → Examples → 01.Basics → Blink” nella finestra Arduino per aprire il programma di esempio Blink.



Dopo aver aperto il programma di esempio, si può vedere il seguente codice, che implementa l'effetto del LED lampeggiante. Si può vedere che il codice ha “prompt” di colore arancione e verde, il che dimostra che l’input è corretto. Fare attenzione alla differenza tra maiuscole e minuscole.

The screenshot shows the Arduino IDE 2.0.4 interface with the title bar "Blink | Arduino IDE 2.0.4". The left sidebar contains icons for file operations, while the main area displays the "Blink.ino" code. The code is a standard Blink sketch with comments explaining the setup and loop functions. It includes copyright information and a link to the original tutorial. The status bar at the bottom indicates "Ln 1, Col 1 Seeeduino XIAO [not connected]".

```
/*
Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink

*/
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

```
/*
Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.
```

```

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);                      // wait for a second
    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);                      // wait for a second
}

```

## Analisi del Codice

`pinMode(LED_BUILTIN, OUTPUT);`

La prima cosa che fa il codice è inizializzare `LED_BUILTIN` come pin di output nella funzione `setup()`. La maggior parte delle schede della serie Arduino imposta di default il LED di bordo sul pin digitale 13. La costante `LED_BUILTIN` collega il LED di bordo al pin 13.

`digitalWrite(LED_BUILTIN, HIGH);`

Nella funzione `loop()`, impostiamo il pin `LED_BUILTIN` sullo stato “on”, inviando una tensione di 5 V o 3,3 V a questo pin, che può essere rappresentata da `HIGH`. Tuttavia, notare che tutti i pin I/O su XIAO sono a 3,3 V. Non immettere una tensione superiore a 3,3 V, altrimenti si potrebbe danneggiare la CPU.

`digitalWrite(LED_BUILTIN, LOW);`

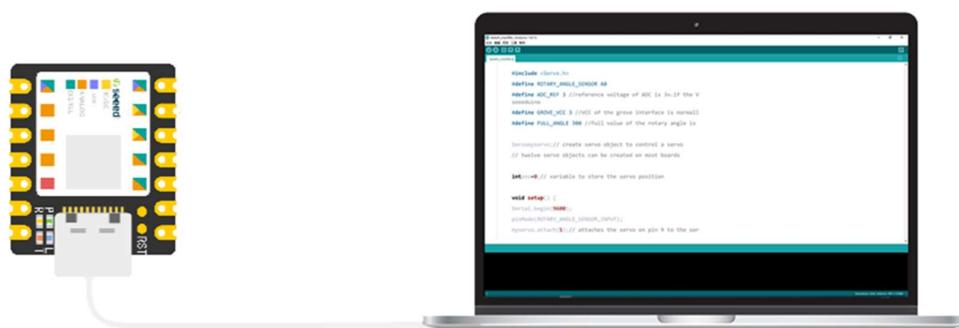
Ciò che si accende deve spegnersi. Questa istruzione imposta il pin `LED_BUILTIN` sullo stato “off”, inviando una tensione di 0 V a questo pin, che può essere rappresentato da `LOW`.

`delay(1000);`

Questa è un’istruzione di ritardo. Significa che il LED può mantenere lo stato “on” o “off” per 1 secondo, perché il parametro nella funzione è in millisecondi, quindi 1000 millisecondi sono 1 secondo. Dopo aver controllato le istruzioni “on” e “off” del LED, è necessario aggiungere un ritardo e il tempo di attesa deve essere lo stesso per garantire che il LED lampeggi in modo uniforme.

## Caricamento del Programma

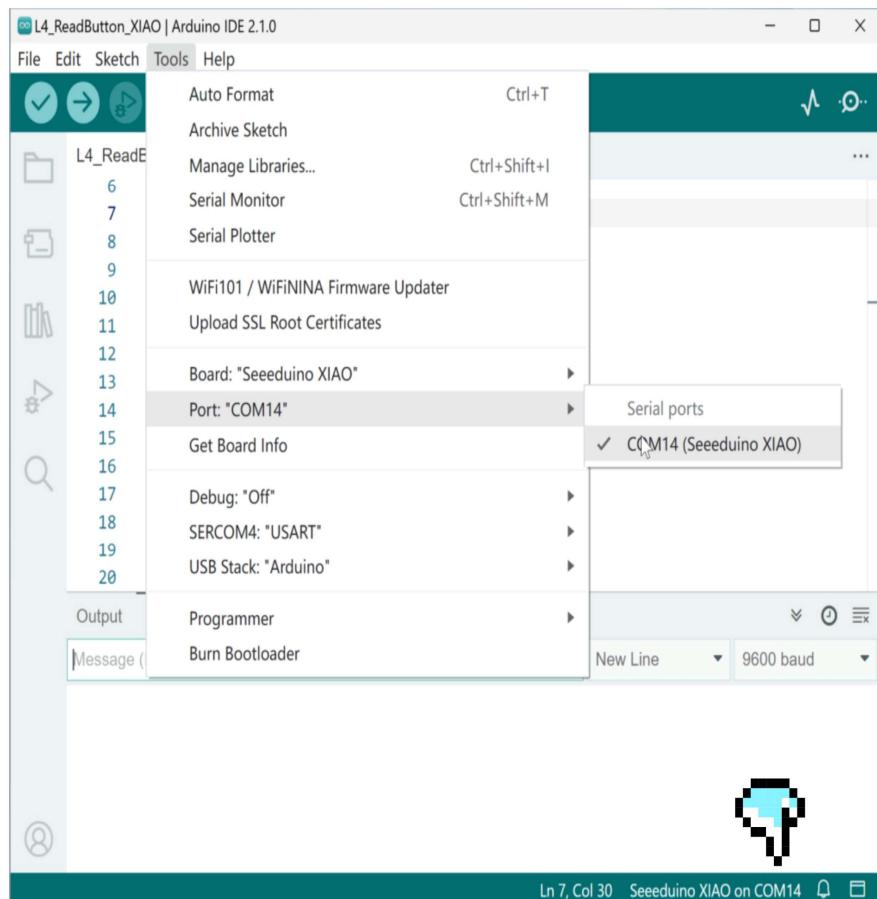
Successivamente, impareremo come caricare il programma. Si usa il cavo dati nel kit per collegare XIAO al computer, come mostrato nella figura.



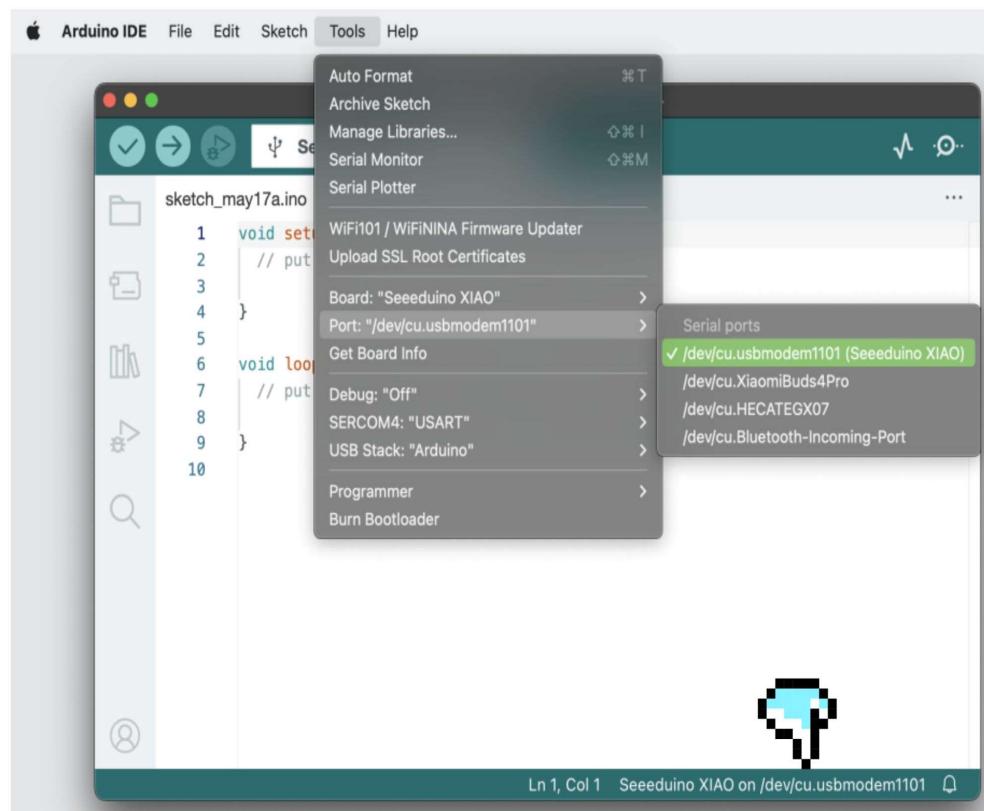
Si sceglie la porta seriale della scheda di sviluppo dalla barra “Tools”. Per gli utenti Windows, è generalmente COM3 o un numero maggiore. Lo si seleziona come mostrato nella figura sottostante.

Se vengono visualizzate più porte per la selezione, scollegare il cavo dati, riaprire la barra “Tools” e la porta che scompare è quella di XIAO. Ricollegare la scheda e selezionare questa

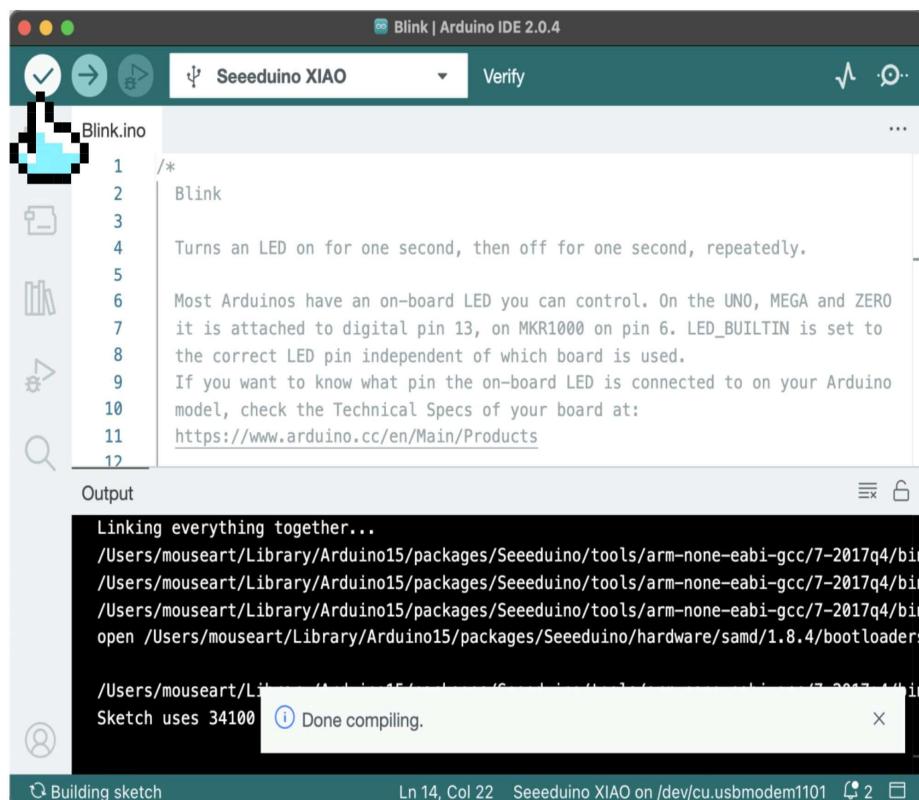
porta seriale. Dopo aver selezionato la scheda e la porta seriale, si può vedere il modello del controller e la porta seriale corrispondente che sono stati impostati nell'angolo inferiore destro dell'interfaccia IDE.



Nei sistemi Mac o Linux, il nome della porta seriale è generalmente `/dev/tty.usbmodem+numero` o `/dev/cu.usbmodem+numero`, come mostrato nella figura sottostante.

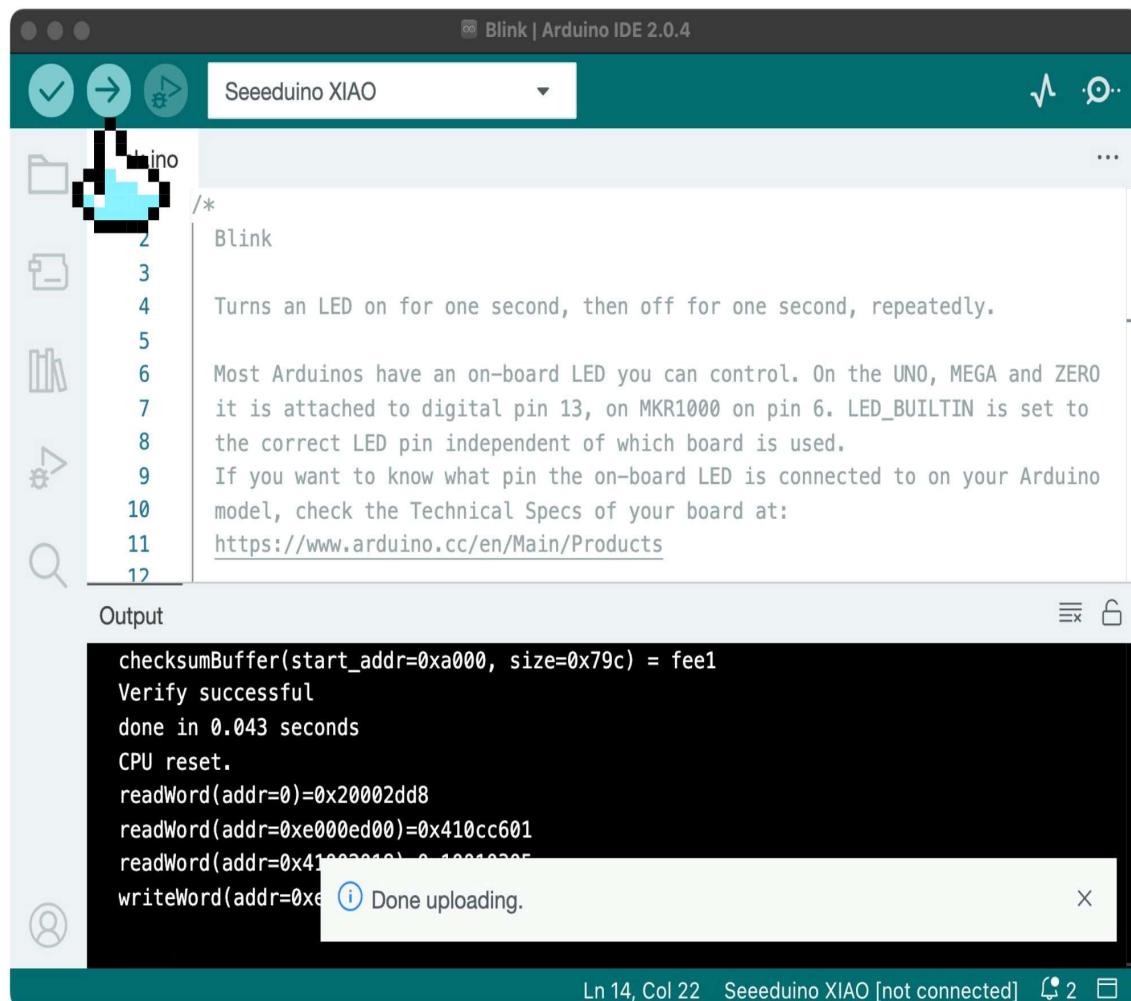


Successivamente, possiamo caricare il programma. Prima di caricare, possiamo cliccare su(verify button) per verificare se il programma è corretto. Se viene visualizzato “Compilation Completed”, il programma è corretto.



Cliccare su (pulsante di upload), la finestra di debug visualizzerà “Compiling Project→Upload”. Quando viene visualizzato “Upload Completed”, si può vedere l’effetto del

programma in esecuzione su XIAO, come mostrato nella finestra di conferma del caricamento riuscito, visualizzata su un computer Mac.

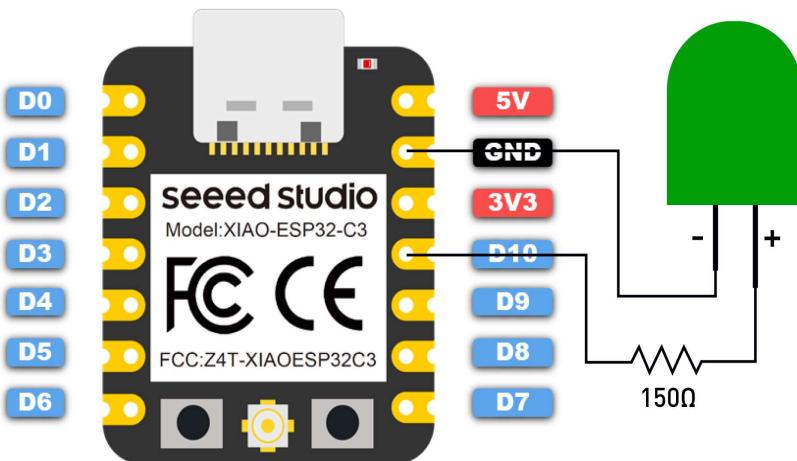


### **⚠ Nota**

Quando si inizia a scrivere codice, spesso si dimenticano le regole delle maiuscole e delle minuscole, la punteggiatura e si commettono errori. Pertanto, provare a scrivere il codice manualmente invece di copiare e incollare. Dopo che il programma di esempio è stato caricato correttamente, provare a creare un nuovo Sketch e inizia a inserire manualmente il codice.

### **1.1.4 Attività 2: Completare l'esempio Blink collegando un LED esterno a Seeed XIAO ESP32C3 senza LED**

Se lo XIAO a portata di mano è Seeed XIAO ESP32C3, poiché non ha un LED a bordo disponibile per gli utenti, per eseguire il programma Blink, si deve prima collegare un LED al pin `D10` della scheda, come mostrato di seguito:



**⚠️ Nota:** Si deve collegare una resistenza (circa  $150\Omega$ ) in serie con il LED per limitare la corrente che scorre attraverso il LED per evitare che bruci il LED.

Poi si copia il seguente programma nell'IDE Arduino:

```
// Define the LED pin according to the pin diagram
int led = D10;

void setup() {
    // Initialize the digital pin 'led' as output
    pinMode(led, OUTPUT);
}

void loop() {
    digitalWrite(led, HIGH);      // Turn the LED on
    delay(1000);                // Wait for a second
    digitalWrite(led, LOW);       // Turn the LED off
    delay(1000);                // Wait for a second
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1\\_Blinks\\_XIAO\\_ESP32C3/L1\\_Blinks\\_XIAO\\_ESP32C3.ino](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1_Blinks_XIAO_ESP32C3/L1_Blinks_XIAO_ESP32C3.ino)

### Analisi del Codice

```
int led = D10;
```

Seeed XIAO ESP32C3 non ha un LED integrato, quindi non abbiamo un pin LED corrispondente preimpostato nel core Arduino. Abbiamo appena collegato il LED al pin `D10`, quindi dobbiamo dichiararlo nel programma.

```
pinMode(led, OUTPUT);
```

Abbiamo definito `led` come `D10` e questo passaggio consiste nell'inizializzare `led(D10)` come pin di output.

### 1.1.5 Esercizio Esteso

Riscrivere il programma Blink: Nel programma di esempio, il LED è acceso e spento per 1 secondo ogni volta, quindi sembra lampeggiare in modo uniforme. Provare a regolare il tempo di attesa per dare al LED durate di lampeggiamento diversi.

Suggerimento:

```
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
```

```
    delay(500);  
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1\\_ll\\_Blinks\\_1\\_en/L1\\_ll\\_Blinks\\_1\\_en.ino](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1_ll_Blinks_1_en/L1_ll_Blinks_1_en.ino)

Per XIAO ESP32C3, dobbiamo anche modificare la parte del programma relativa alla definizione dei pin:

```
int led = D10;  
void setup() {  
    pinMode(led, OUTPUT);  
}  
void loop() {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(500);  
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1\\_ll\\_blinks\\_2\\_en/L1\\_ll\\_blinks\\_2\\_en.ino](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L1_ll_blinks_2_en/L1_ll_blinks_2_en.ino)

## 1.2 Utilizzo del pulsante sulla scheda di espansione XIAO per controllare una luce LED

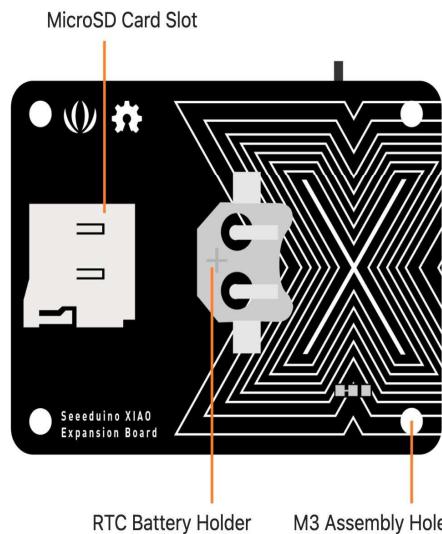
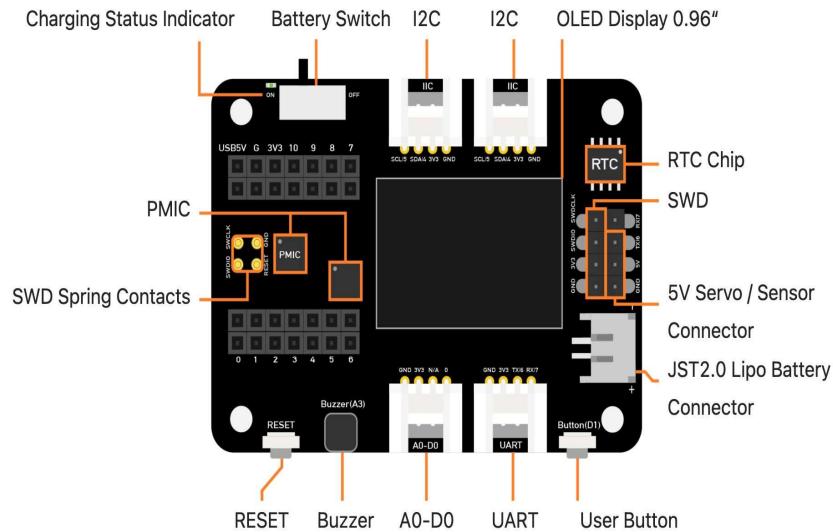
Nella sezione precedente, abbiamo imparato come controllare un LED per farlo lampeggiare utilizzando solo Seeed Studio XIAO e il LED integrato. Tuttavia, non c'era alcuna interazione con l'ambiente esterno, come il controllo del LED tramite luce o suono. In questa sezione, introdurremo un semplice sensore, il pulsante, per formare un sistema di controllo automatico di sensore-regolatore-attuatore. Prima di iniziare l'attività, dobbiamo apprendere alcune nozioni di base, come cosa sono le variabili e le strutture comuni del programma, in modo da poter comprendere ed eseguire meglio il programma.

### 1.2.1 Conoscenza di Base

Nell'ultima sezione, abbiamo utilizzato solo il LED integrato di Seeed Studio XIAO senza collegare altri moduli. Potrebbe richiedere un certo sforzo per i principianti utilizzare fili Dupont per collegare sensori esterni a una scheda delle dimensioni di un pollice e coinvolgere anche una breadboard. Esiste un metodo più semplice?

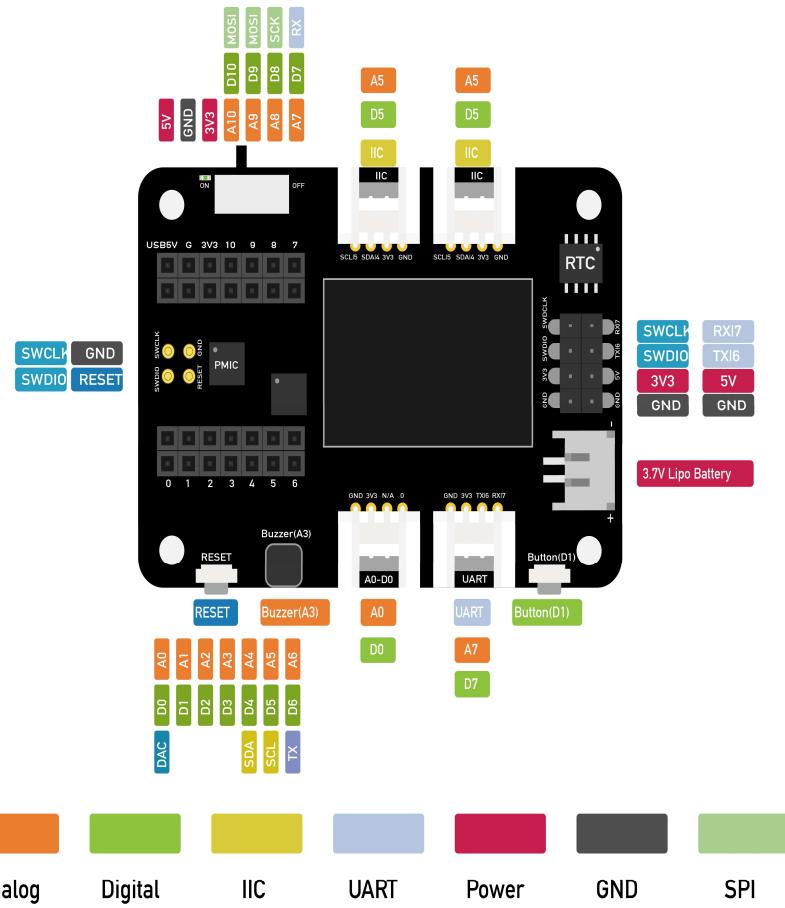
#### 1.2.1.1 Seeed Studio XIAO Expansion Board

La [Seeed Studio XIAO Expansion Board](#), grande solo la metà di Raspberry Pi 4, è potente e può creare prototipi e progetti in modo rapido e semplice. La scheda ha una varietà di periferiche come OLED, RTC, memoria espandibile, buzzer passivo, RESET/Pulsante Utente, connettore servo/sensore 5V, varie interfacce dati... Si possono esplorare le infinite possibilità di Seeed Studio XIAO. La scheda supporta anche [CircuitPython](#). Tutti i modelli della serie Seeed Studio XIAO hanno specifiche uniformi e supportano Seeed Studio XIAO [Grove Shield](#) e la [Seeed Studio XIAO Expansion Board](#). La serie include [XIAO SAMD21](#), [XIAO RP2040](#), [XIAO nRF52840](#), [XIAO nRF52840 Sense](#), [XIAO ESP32C3](#) e [XIAO ESP32S3](#). Le interfacce delle funzioni anteriore e posteriore della scheda di espansione XIAO sono illustrate nella figura seguente:



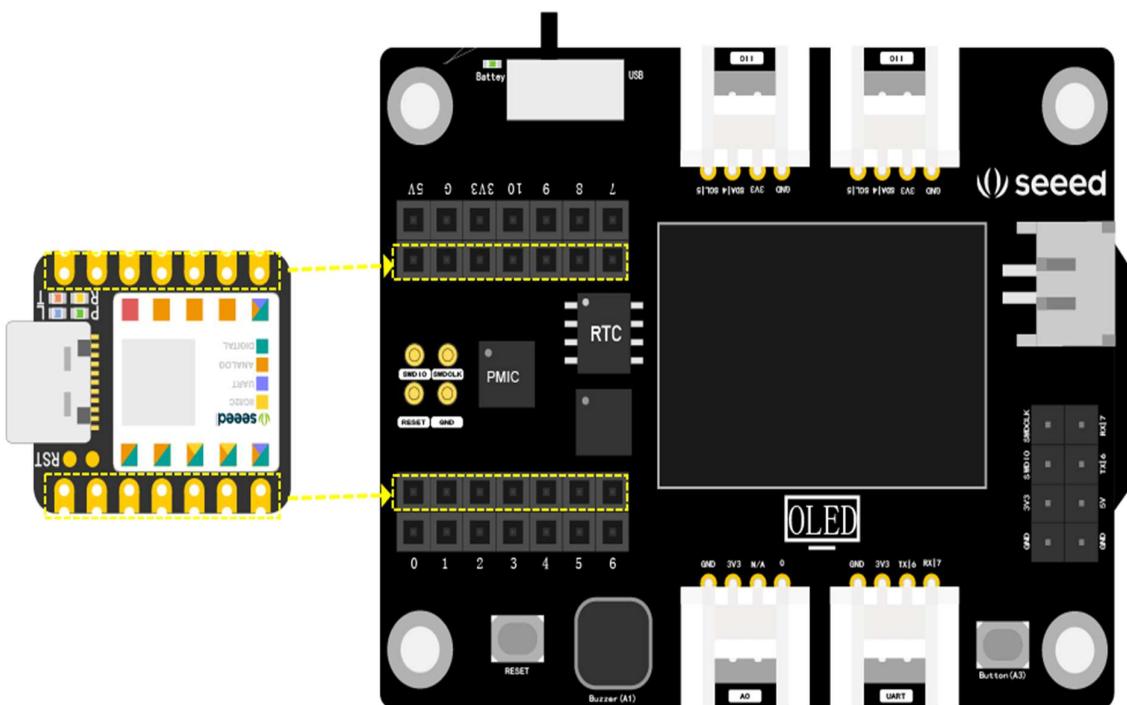
Per facilitare e velocizzare la creazione di progetti con Seeeduino XIAO, l'abbiamo dotata di una potente scheda di espansione. Questa scheda ha una vasta gamma di periferiche integrate e può connettersi rapidamente a più moduli elettronici per implementare varie funzioni. La scheda di espansione mette in evidenza tutti i pin di XIAO, come mostrato nello schema di seguito:

## Pinout



GPIO	0	1	2	3	4	5	6	7	8	9	10	SWCLK	SWDIO	RESET
Interface	A0, D0	D1	SPI	A3	IIC		UART		SPI			SWD		RESET
Components	Grove*1	User Button	mini SD	Buzzer	Grove*2		Grove*1, Headers			mini SD		Headers		Button

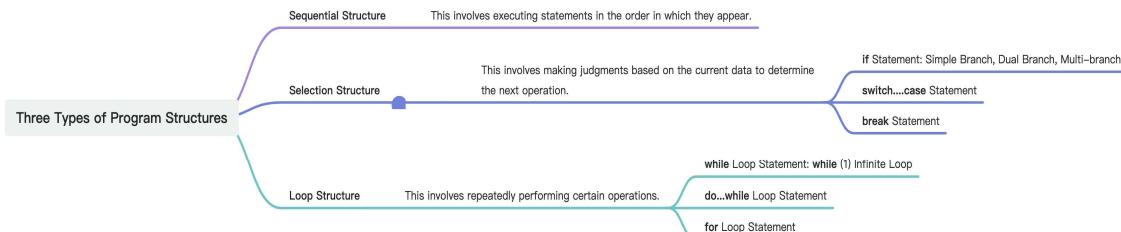
Nella maggior parte dei casi, la scheda di espansione XIAO è adatta a tutti i prodotti della serie Seeed Studio XIAO. Quando dobbiamo usare la scheda di espansione XIAO, dobbiamo collegare la scheda di sviluppo XIAO alla posizione corrispondente sulla scheda di espansione, come mostrato nella figura sottostante. Collegare i pin header sulla scheda principale XIAO alla posizione cerchiata in giallo sulla scheda di espansione. Assicurarsi di allinearla prima di premere verso il basso per evitare di danneggiare i pin. Dopodiché, possiamo iniziare a lavorare sui progetti in combinazione con la scheda di espansione.



**⚠️ Nota:** Collegare prima Seeed Studio XIAO ai due header femmina sulla scheda di espansione, poi collegare l'alimentatore Type-C, altrimenti si danneggerà la Seeed Studio XIAO e la scheda di espansione.

### 1.2.1.2 Tre Strutture di Base dei Programmi

Le tre strutture di base dei programmi sono la struttura sequenziale, quella di selezione e quella del loop.

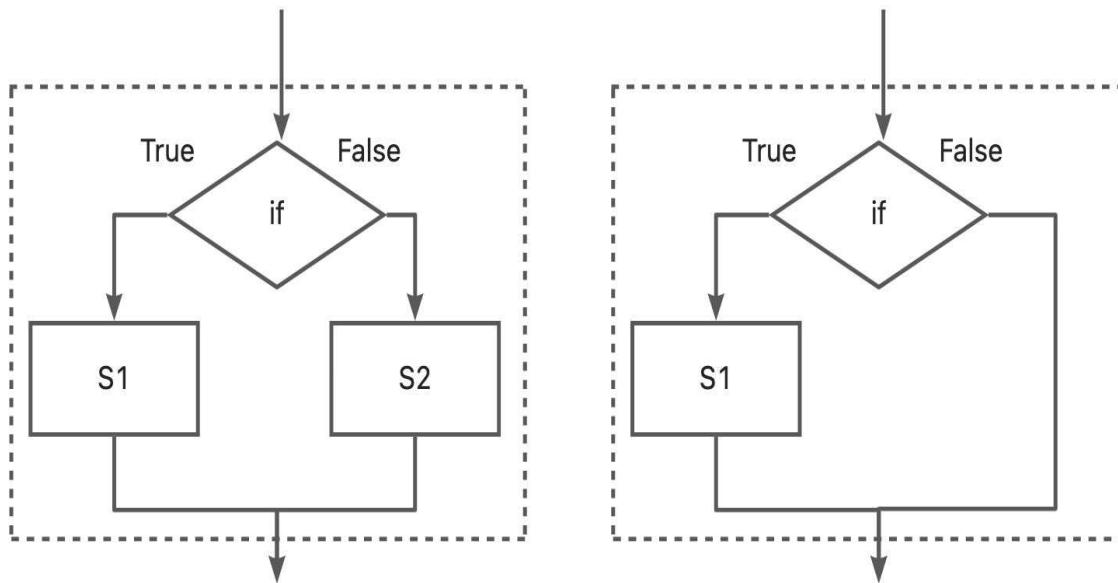


#### Struttura Sequenziale

Come suggerisce il nome, il programma in una struttura sequenziale viene eseguito nell'ordine delle istruzioni. È la struttura di programma più elementare e semplice. Come mostrato nella figura sottostante, il programma eseguirà prima l'operazione nella casella S1, poi l'operazione nella casella S2 e così via.

#### Struttura di Selezione

In un programma, a volte dobbiamo esprimere giudizi in base alla situazione per decidere il passaggio successivo. Ad esempio, il programma potrebbe dover valutare il valore della luce nell'ambiente corrente. Se il valore della luce è alto, indicando un ambiente luminoso, non è necessario accendere la luce. Se il valore della luce è basso, indicando un ambiente poco luminoso, allora è necessario accendere la luce. In questi casi, utilizziamo una struttura di selezione. Come mostrato nelle figure seguenti, la struttura di selezione giudicherà se la condizione è soddisfatta. Se "True", esegue S1; se "False", esegue S2; o se "True", esegue S1, se "False", esce dalla struttura di selezione.



### L'istruzione if

L'istruzione if è la struttura di selezione più comune, che esegue l'istruzione seguente quando l'espressione data è vera. L'istruzione if ha tre forme strutturali come mostrato nell'esempio seguente. Struttura di diramazione semplice: Esegui quando la condizione è soddisfatta.

---

```
if (expression) {
    statement;
}
```

---

Struttura a doppio ramo: Esegui statement1 quando la condizione è soddisfatta, altrimenti esegui statement2.

---

```
if (expression) {
    statement1;
}
else {
    statement2;
}
```

---

Struttura multi-ramo: Usa istruzioni if nidificate per valutare diverse situazioni.

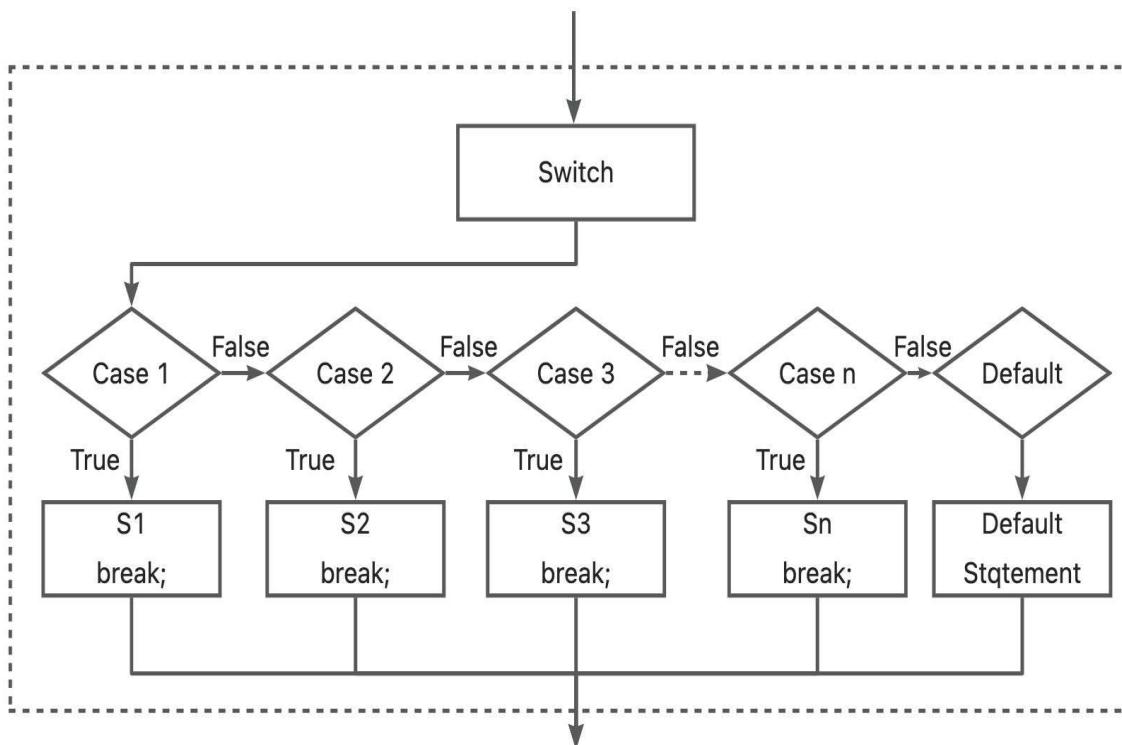
---

```
if (expression1) {
    statement1;
}
else if (expression2) {
    statement2;
}
else if (expression3) {
    statement3;
}
```

---

### Istruzione switch.....case

Quando si ha a che fare con più rami di selezione, usare una struttura "if.....else" per scrivere un programma può essere piuttosto lungo. In questo caso, è molto più comodo usare un'istruzione `switch`. La struttura `switch` confronta l'espressione tra parentesi con le costanti dopo `case`. Se corrispondono, esegue l'istruzione corrispondente ed esce dalla struttura tramite un'istruzione `break`. Se non corrisponde, esegue l'istruzione dopo `default`. È importante notare che l'espressione tra parentesi dopo `switch` deve essere di tipo intero o carattere.




---

```

switch (expression) {
  case constant_expression1:
    statement1;
    break;
  case constant_expression2:
    statement2;
    break;
  .....
  default:
    statementn;
    break;
}
  
```

---

### L'istruzione break

L'istruzione `break` può essere utilizzata solo in una struttura di selezione multi-ramo `switch` e in strutture loop. Viene utilizzata per terminare la struttura del programma corrente, consentendo al programma di passare alle istruzioni successive per l'esecuzione.

### Struttura Loop

Una struttura loop viene utilizzata quando una parte del programma deve essere eseguita ripetutamente, in base a condizioni di giudizio fornite per determinare se continuare a eseguire una determinata operazione o uscire dal loop. Esistono tre tipi comuni di istruzioni loop:

#### Il Loop while

Il loop `while` è un tipo di ciclo “when” [quando] che esegue le istruzioni nel corpo del ciclo quando viene soddisfatta una determinata condizione.

---

```

while (expression) {
  statement;
}
  
```

---

### **Il Loop do.....while**

Questo è un tipo di loop “until” [finché]. L’istruzione nel corpo del loop viene eseguita una volta prima che l’espressione venga valutata. Se l’espressione è vera, il loop continua.

---

```
do {
    statement;
} while (expression);
```

---

### **Il Loop for**

Questo include tre espressioni: Expression1 per l’inizializzazione, Expression2 per il giudizio ed Expression3 per l’incremento.

---

```
for (Expression1; Expression2; Expression3) {
    statement;
}
```

---

Oltre alle istruzioni del loop di cui sopra, ci sono istruzioni di controllo, `break` e `continue`, nella struttura del loop utilizzata per terminare prematuramente il loop o uscire da esso. In questa lezione, dobbiamo solo comprendere queste strutture di programma. Nei corsi successivi, le padroneggeremo gradualmente attraverso esempi di progetto.

## **1.2.2 Attività 1: Controllare il LED sullo XIAO tramite il pulsante sulla “XIAO expansion board”**

### **Analisi**

L’effetto che vogliamo ottenere è che quando si preme il pulsante, il LED si accende; quando si rilascia il pulsante, il LED si spegne. Il programma è scritto in tre passaggi:

- Definire i pin e creare le variabili.
- Inizializzare e impostare lo stato del pin.
- Leggere lo stato del pulsante, implementare il giudizio per la condizione. Se si preme il pulsante, la luce è accesa, altrimenti, la luce è spenta.

### **Variabile**

In un programma, un valore che può cambiare è chiamato variabile. Ad esempio, definendo una variabile intera `i` come `int i;`. Possiamo assegnare un valore alla variabile nello stesso momento in cui la definiamo, come `int i = 0;`. Inoltre, a seconda del tipo di dati, vengono utilizzate istruzioni diverse per definire le variabili, come la definizione di un numero in virgola mobile, `float x = 1.9;`, e così via. Per maggiori dettagli, fare riferimento alla documentazione sui tipi di dati e sulle costanti di Arduino <https://www.arduino.cc/reference/en/#variables>.

### **Scrittura del Programma:**

**Passaggio 1:** Definire i pin e creare le variabili. Il pulsante integrato sulla scheda di espansione XIAO è `D1`, quindi lo definiamo come pin 1 e impostiamo una variabile per lo stato del pulsante. Notare che `LED_BUILTIN` imposterà il LED sul pin corretto, quindi non è necessario definirlo manualmente:

```
const int buttonPin = 1; // L'interruttore del pulsante integrato sulla scheda di espansione XIAO è D1, che definiamo come pin 1
// Se si utilizza XIAO RP2040, modificare 1 in D1
int buttonState = 0; // buttonState è una variabile per memorizzare lo stato del pulsante
```

**Passaggio 2:** Impostare lo stato del pin. Impostare il pin LED sullo stato di output e il pin del pulsante sullo stato di input con pull-up. Utilizzare `INPUT_PULLUP` per abilitare i resistori di pull-

up interni. Quando il pulsante non è premuto, restituisce 1 o HIGH (livello alto). Quando il pulsante è premuto, restituisce 0 o LOW (livello basso).

```
void setup() {
    pinMode(LED_BUILTIN, OUTPUT); // Imposta il pin LED sullo stato di output
    pinMode(buttonPin, INPUT_PULLUP); // Imposta il pin pulsante sullo stato di input
}
```

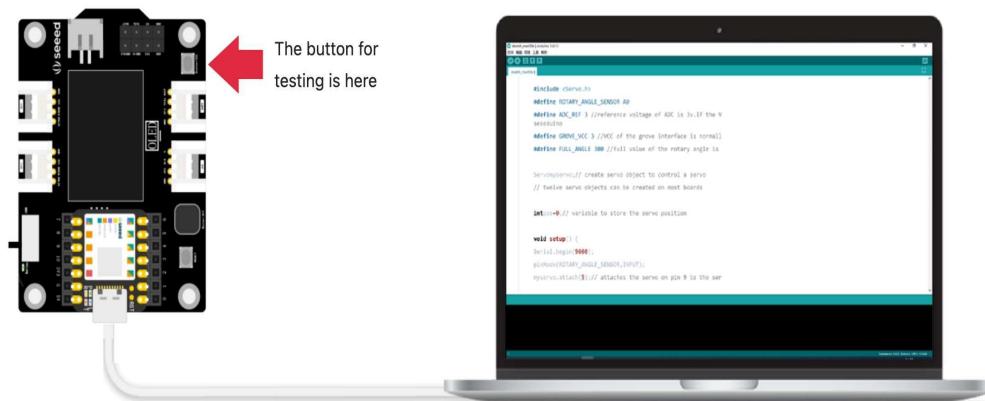
**Passaggio 3:** Leggere continuamente lo stato del pulsante. Se si preme il pulsante, la luce è accesa, altrimenti, la luce è spenta. Poiché il LED integrato dello XIAO è a logica negativa, quando il pulsante è premuto e restituisce 0, il LED è acceso; quando restituisce 1, il LED è spento.

```
void loop() {
    // Leggi lo stato del pulsante e memorizzalo nella variabile buttonState
    buttonState = digitalRead(buttonPin);
    // Controlla se il pulsante è premuto, se il pulsante è premuto
    if (buttonState == HIGH) {
        // Accende il LED:
        digitalWrite(LED_BUILTIN, HIGH);
    }
    else {
        // Spegni il LED:
        digitalWrite(LED_BUILTIN, LOW);
    }
}
```

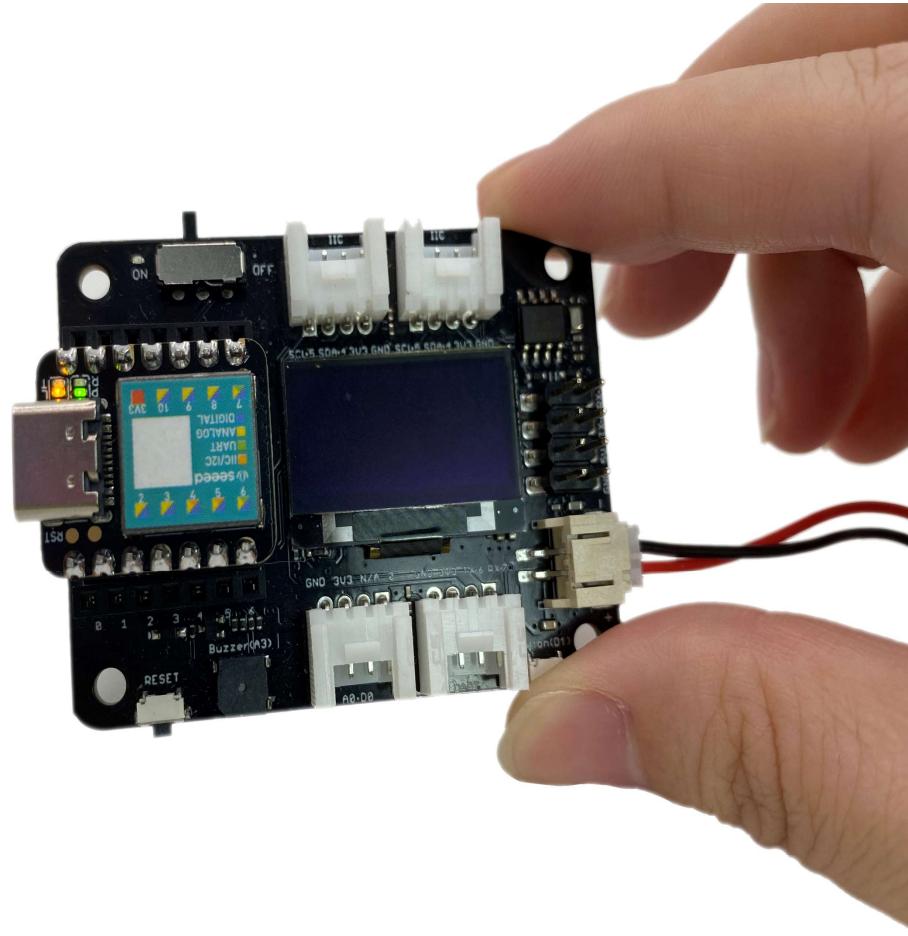
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L2\\_Button\\_XIAO\\_en/L2\\_Button\\_XIAO\\_en.ino](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/code/L2_Button_XIAO_en/L2_Button_XIAO_en.ino)

## Caricamento del Programma

Carichiamo sull'hardware il programma che abbiamo scritto. Per prima cosa, utilizziamo il cavo dati nel kit per collegare XIAO al computer.



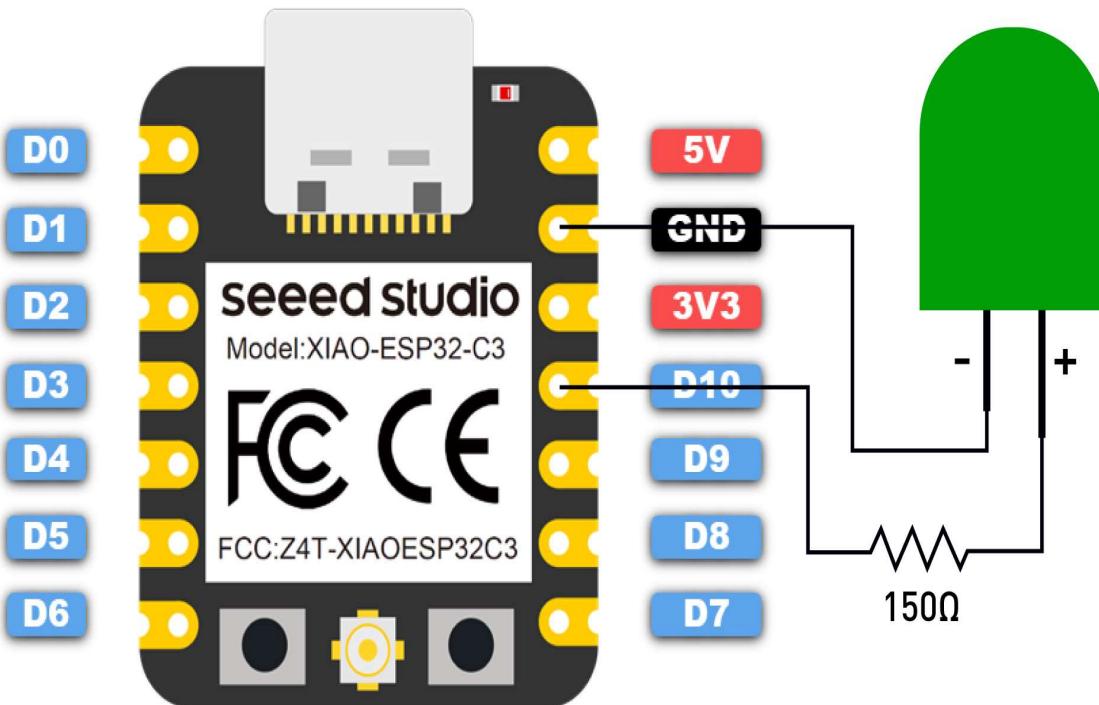
Notare la posizione dei pulsanti sulle estensioni XIAO utilizzate per il test nella figura. Quindi clicchiamo sul pulsante di verifica per verificare il programma. Se è corretto, cliccare sul pulsante di upload per caricare il programma sull'hardware. Quando l'area di debug visualizza "Done uploading.", possiamo premere il pulsante per vedere se il LED si accende.



**⚠️ Nota:** Ci sono due pulsanti identici sulla scheda di espansione. Uno è il pulsante RESET vicino all’interfaccia Type-C e l’altro è il pulsante definito dall’utente vicino all’interfaccia della batteria al litio. Provare con quello vicino all’interfaccia della batteria al litio.

### **1.2.3 Attività 2: Utilizzare il pulsante sulla scheda di espansione XIAO per controllare il LED esterno su XIAO ESP32C3**

Per Seeed XIAO ESP32C3, non è presente un LED integrato che gli utenti possono utilizzare. Per eseguire il programma Blink, si deve prima collegare un LED al pin `D10` della scheda come mostrato:



**⚠️ Nota:** Assicurarsi di aggiungere una resistenza (circa  $150\Omega$ ) in serie con il LED per limitare la corrente che scorre attraverso il LED ed evitare che la sovraccorrente bruci il LED.

Quindi copiare il seguente programma nell'IDE Arduino:

```
/*
 * Button controlling external LED of XIAO
Apologies for the confusion. It seems that there was an issue with quoting text from the document.
Let's continue:
#####
Task 2: Use the button on the XIAO expansion board to control the external LED on the XIAO
ESP32C3
For the Seeed XIAO ESP32C3, it doesn't have an on-board LED for users to use. To execute the Blink
program, you first need to connect an LED to the board's `D10` pin as shown.

> ⚠️ Note: Make sure to add a resistor (about  $150\Omega$ ) in series with the LED to limit the current
flowing through the LED and prevent overcurrent from burning out the LED.

Then, copy the following program into the Arduino IDE:
```
/*
 * Button controlling external LED of XIAO ESP32C3
 */
const int buttonPin = 1;      // The pin number of the button
int buttonState = 0;        // Variable for reading the button status
int led = D10;   // Pin number of the LED

void setup() {
    // Initialize the LED pin as an output:
    pinMode(led, OUTPUT);
    // Initialize the button pin as an input:
    pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
    // Read the state of the button:
    buttonState = digitalRead(buttonPin);
    // Check if the button is pressed. If it is, the button state is HIGH
    if (buttonState == HIGH) {
        // Turn the LED on:
        digitalWrite(led, HIGH);
    }
}
else {

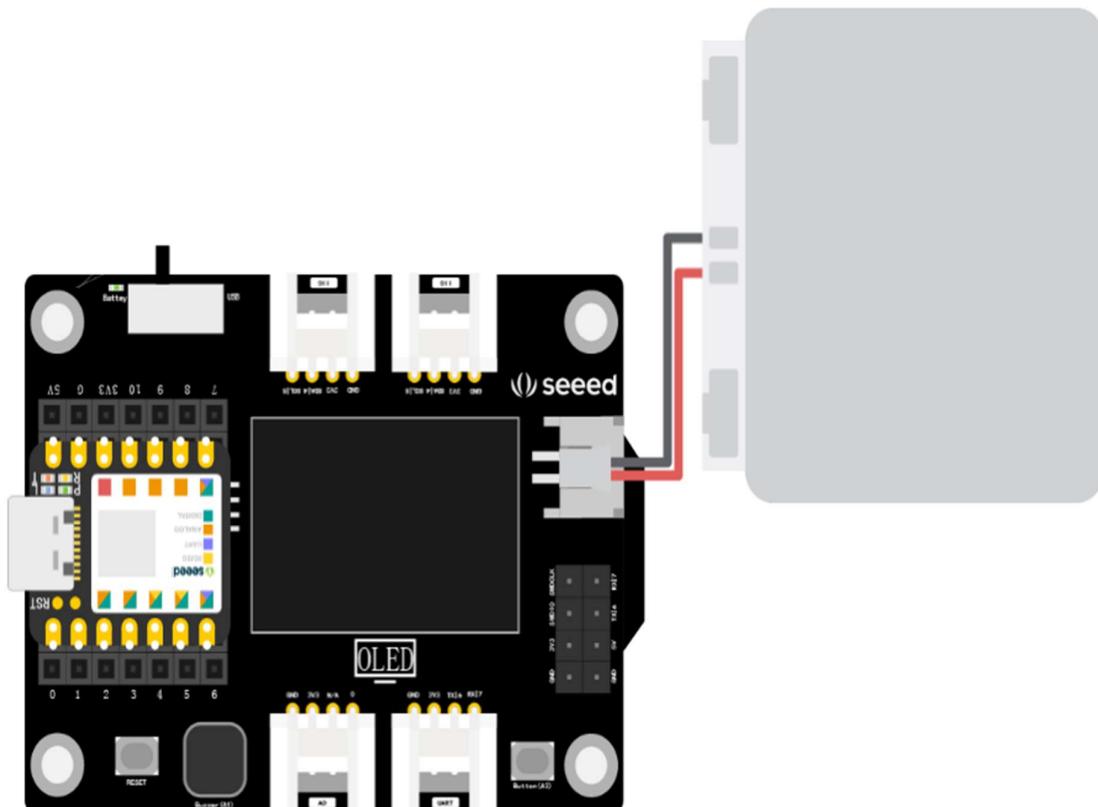
```

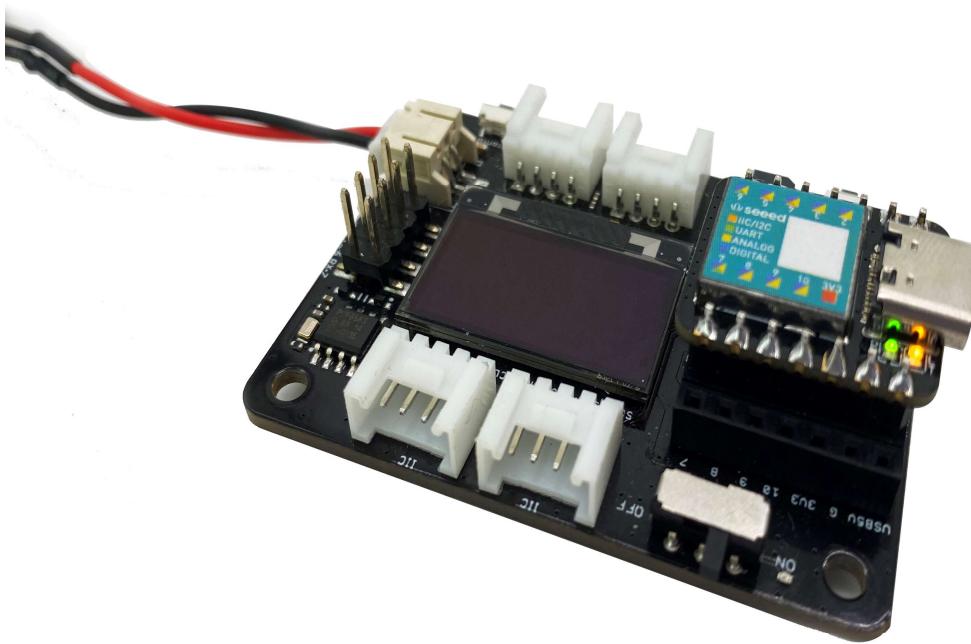
```
// Turn the LED off:  
digitalWrite(led, LOW);  
}  
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L2\\_Button\\_XIAO\\_ESP32C3\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L2_Button_XIAO_ESP32C3_en)

## Alimentazione di XIAO con una batteria esterna

Quando si dimostra l'effetto, oltre a utilizzare un cavo dati per alimentare il computer, è possibile utilizzare anche una batteria al litio esterna. Ciò rende più comodo lo spostamento e l'esecuzione di progetti, come mostrato nell'immagine.





## 1.2.4 Esercizio Esteso

### Flow Chart

Prima di scrivere il programma, è possibile disegnare un “Flow Chart” [diagramma di flusso] del programma per organizzare i propri pensieri. I simboli comuni del diagramma di flusso sono i seguenti:



Start/End box: Indicates the start or end of a program.



Process box: Indicates the execution or handling of certain tasks.



Decision box: Represents the judgment of a certain condition.



Input/Output box: Represents the input of data or the output of results.



Connection line: Represents the direction of the process flow.

Il programma del LED controllato da pulsanti che abbiamo implementato in questa sezione è rappresentato dal seguente diagramma di flusso. Potete provare a disegnarlo da soli.

## 1.3 Trasformare XIAO e la sua Expansion Board in un Trasmettitore di Codice Morse

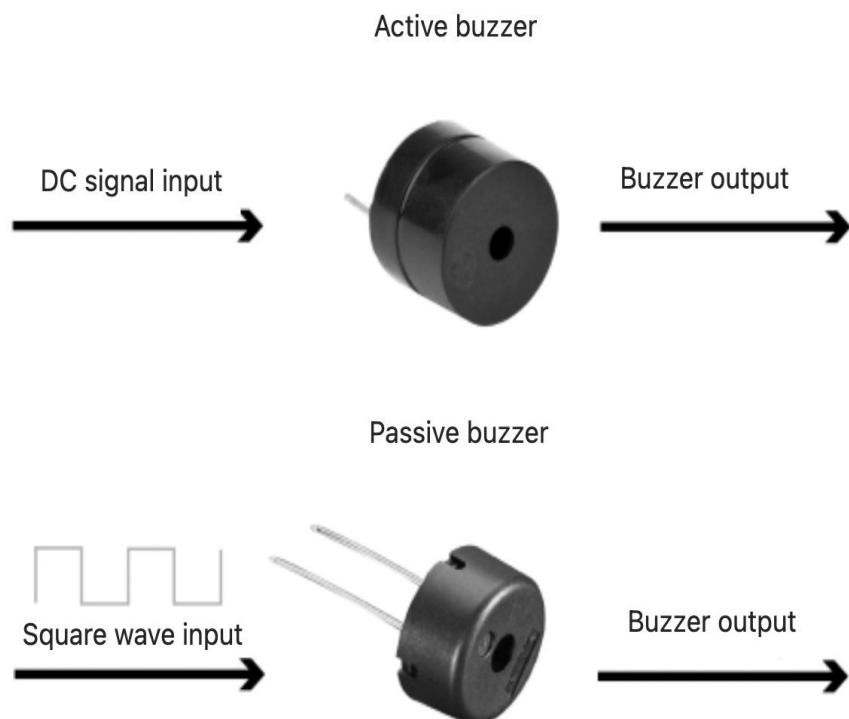
Tutti sanno che “SOS” è un segnale di emergenza riconosciuto a livello internazionale, una forma di codice Morse. Oggi trasformeremo XIAO di Seeed Studio in un trasmettitore di codice Morse. Cercheremo di far sì che il buzzer di bordo della scheda di espansione invii automaticamente segnali. Inoltre, impareremo come controllare manualmente il buzzer con un pulsante.

### 1.3.1 Conoscenza di Base

#### 1.3.1.1 Il Buzzer

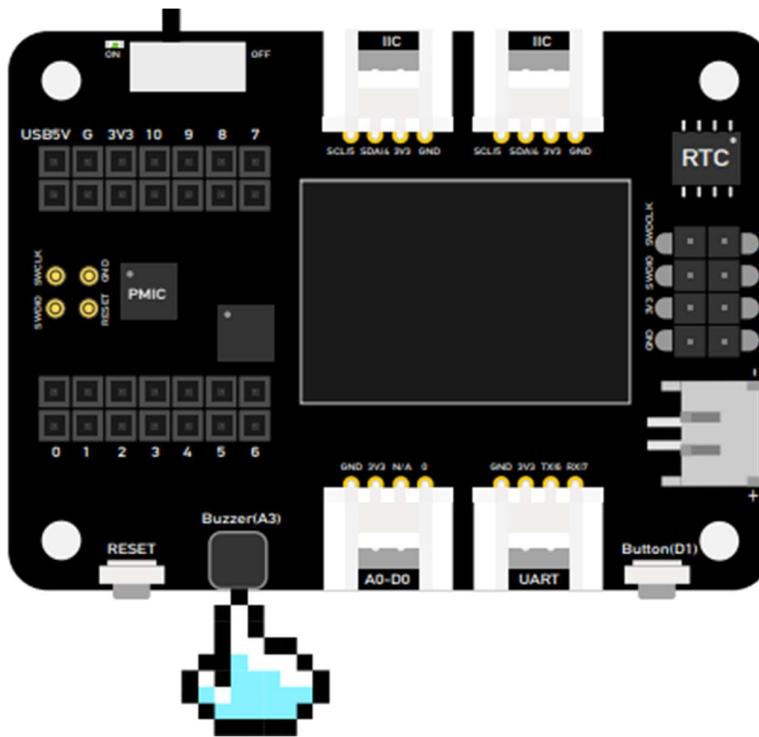
Un buzzer è un dispositivo sonoro elettronico integrato che genera suoni in base all’ingresso di un segnale elettrico. I buzzer sono spesso installati su prodotti elettronici per la generazione di suoni. Esistono due tipi di buzzer: attivi (buzzer sorgente) e passivi (buzzer senza sorgente).

- **Buzzer Attivi:** Questi buzzer hanno un semplice circuito di oscillazione al loro interno. Quando è collegato a un alimentatore CC, il buzzer può convertire una CC costante in un segnale di impulsi a una certa frequenza, facendo vibrare il foglio di alluminio interno e producendo un suono. I buzzer attivi possono solitamente emettere solo alcuni suoni a tono fisso (frequenza) e sono ampiamente utilizzati nei dispositivi sonori di computer, stampanti, fotocopiatrici, allarmi, giocattoli elettronici, elettronica per auto, telefoni, timer e altri prodotti elettronici.
- **Buzzer Passivi:** Questi buzzer funzionano in modo simile agli altoparlanti. Non hanno un oscillatore interno e devono essere collegati a un segnale di corrente variabile per funzionare. Di solito usano segnali a onda quadra di frequenza diversa per il pilotaggio. Il suono generato dai buzzer passivi cambierà in base alla variazione del segnale di ingresso e possono emettere una varietà di suoni come gli altoparlanti, non solo emettendo un singolo tono fisso (frequenza).



Il modulo buzzer standalone è mostrato nella figura sottostante.

Nella scheda di espansione Seeed Studio XIAO, c'è un buzzer passivo integrato collegato al pin A3. Possiamo emettere segnali a impulsi PWM su questo pin per controllare il buzzer.



### 1.3.1.2 Le Funzioni `tone()` e `noTone()`

#### **La Funzione `tone()`**

La funzione `tone()` può generare un segnale PWM a frequenza fissa per azionare un cicalino passivo e produrre un suono, e può definire la frequenza e la durata del suono.

#### **Sintassi:**

```
tone(pin, frequency); tone(pin, frequency, duration);
```

#### **Parametri:**

`pin`: Il pin a cui è collegato il buzzer (nella scheda di espansione Seeed Studio XIAO, è A3).

`frequency`: La frequenza del suono (unità: Hz), il tipo consentito è `unsigned integer`. `duration`: La durata del suono (unità: millisecondi, questo parametro è facoltativo), il tipo consentito è `unsigned long`.

#### **La Funzione `noTone()`**

Questa funzione è utilizzata per interrompere il suono del buzzer controllato dalla funzione `tone()`. Se non viene generato alcun suono, la funzione non è valida.

#### **Sintassi:**

```
noTone(pin);
```

#### **Parametri:**

`pin`: Il pin per interrompere il suono.

### 1.3.1.3 Operatori Comuni

Negli studi precedenti, abbiamo utilizzato alcuni operatori. Successivamente, impareremo i tipi comuni di operatori e i loro metodi di utilizzo.

#### **Operatori Aritmetici:**

Operatore Spiegazione

|           |                              |
|-----------|------------------------------|
| Operatore | Spiegazione                  |
| =         | Operatore di assegnazione    |
| +         | Operatore di addizione       |
| -         | Operatore di sottrazione     |
| *         | Operatore di moltiplicazione |
| /         | Operatore di divisione       |
| %         | Operatore modulo             |

### ***Operatori di Confronto:***

|           |                     |
|-----------|---------------------|
| Operatore | Spiegazione         |
| !=        | Diverso da          |
| <         | Minore di           |
| <=        | Minore o uguale a   |
| ==        | Uguale a            |
| >         | Maggiore di         |
| >=        | Maggiore o uguale a |

### ***Operatori Booleani:***

|           |              |
|-----------|--------------|
| Operatore | Spiegazione  |
| &&        | “and” logico |
| !         | “not” logico |

### ***Operatori Composti:***

|           |                      |
|-----------|----------------------|
| Operatore | Spiegazione          |
| ++        | Autoincremento       |
| +=        | Addizione composta   |
| -         | Autodecremento       |
| -=        | Sottrazione composta |

Per spiegazioni dettagliate, vedere: <https://www.arduino.cc/reference/en/>

### **1.3.1.4 Codice Morse**

Il codice Morse è un metodo di espressione delle informazioni nelle telecomunicazioni, che prende il nome dall'inventore del telegrafo, Samuel Morse.



Fonte: [https://en.wikipedia.org/wiki/Samuel\\_Morse](https://en.wikipedia.org/wiki/Samuel_Morse) Il codice Morse internazionale codifica le 26 lettere inglesi dalla A alla Z, alcune lettere non inglesi, numeri arabi e un piccolo numero di segni di punteggiatura e prosegini. Non c'è distinzione tra lettere maiuscole e minuscole. Ogni simbolo del codice Morse è costituito da una serie di punti (·) e trattini (—). La durata di un punto è l'unità di misura del tempo di base nella trasmissione del codice Morse. La durata di un trattino è tre volte la durata di un punto. Dopo ogni punto o trattino in un carattere, c'è un momento in cui il segnale è assente, chiamato spazio, pari alla durata di un punto. Ad

esempio, il segnale di soccorso di emergenza standard SOS è espresso in codice Morse come mostrato nella figura seguente.

● ● ● — — — ● ● ●

Acusticamente, suona così.

Il browser non supporta l'elemento audio.

Come si può vedere, il codice Morse essenzialmente codifica la durata per trasmettere segnali. Con tali regole di codifica, le persone possono presentare la durata in molti modi, come emettere suoni a intermittenza, accendere riflettori a intermittenza, ecc., per raggiungere in ultima analisi lo scopo di inviare informazioni.



Fonte:[https://en.wikipedia.org/wiki/Morse\\_code](https://en.wikipedia.org/wiki/Morse_code) Il codice Morse era utilizzato principalmente quando la comunicazione non era ancora sviluppata. Le persone utilizzavano il codice Morse per la trasmissione di informazioni a lunga distanza tramite radio. Il vecchio dispositivo qui sotto è un primo dispositivo telegrafico, in cui l'operatore controlla i segnali lunghi e corti del telegrafo premendo la maniglia rotonda sulla destra.



Fonte: [https://en.wikipedia.org/wiki/Morse\\_code](https://en.wikipedia.org/wiki/Morse_code) Se si è interessato al codice Morse, si può visitare il seguente sito Web. Il sito Web può tradurre le lettere e i numeri che si inseriscono in codice Morse e fornire un download del file audio. <https://morsedecoder.com/>

## 1.3.2 Attività 1: Trasmissione Automatica di “SOS”

### Analisi

La trasmissione automatica significa che quando la scheda di controllo viene avviata, il cicalino di bordo emette automaticamente il codice Morse di “SOS”. Il programma è scritto in tre passaggi:

- Definizione del pin del “buzzer” [cicalino]
- Inizializzazione, impostazione dello stato del pin del cicalino
- Eseguire il ciclo del cicalino per riprodurre il codice Morse di “SOS”

Diamo prima un’occhiata a come riflettere il codice Morse di “SOS” tramite il programma. Se si importa il file audio del codice Morse nel software di modifica audio, si può vedere la forma d’onda del suono e la durata di ogni sillaba, che è generalmente divisa in suoni lunghi e brevi. Per facilitare la comprensione e la programmazione, utilizziamo un metodo binario per contrassegnare l’interruttore del buzzer, 1 indica che il buzzer è acceso, 0 indica che il buzzer è spento e il numero grigio rappresenta la durata dello stato corrente. Dopo che un codice Morse termina, poiché deve essere ripetuto, è necessario lasciare del tempo tra i due codici Morse, qui è impostato su 0.8 secondi.



Per emettere un suono breve, che corrisponde a un punto in codice Morse, dal buzzer, si può utilizzare il seguente codice in Arduino:

```
tone(pinBuzzer, 200);
delay(100);
noTone(pinBuzzer);
delay(100);
```

In questo frammento di codice:

- `tone(pinBuzzer, 200)` genera un suono a una frequenza di 200 Hz sul buzzer collegato al pin `pinBuzzer`.
- `delay(100)` attende 100 millisecondi. Questa è la durata del suono.
- `noTone(pinBuzzer)` interrompe il suono del buzzer.
- L’ultimo `delay(100)` assicura che ci sia una pausa prima che venga generato il suono successivo, rappresentando lo spazio tra i segnali.

Il codice fornito è un programma Arduino completo per emettere il segnale SOS in codice Morse tramite un buzzer. Ecco la spiegazione in inglese:

### Scrittura del Programma

#### Passaggio 1: Definire i pin e creare le variabili

```
int pinBuzzer = 3; // Define the buzzer on pin 3, if you're using XIAO RP2040/XIAO ESP32, change 3 to A3
```

### Passaggio 2: Impostare lo stato del pin

```
void setup() {
    pinMode(pinBuzzer, OUTPUT); // Set the buzzer pin to output state
}
```

### Passaggio 3: Eseguire un ciclo per riprodurre il codice Morse “SOS”

```
void loop() {
    // Emetti tre brevi segnali:
    for(int i = 0; i < 3; i++){
        tone(pinBuzzer, 200);
        delay(100);

        noTone(pinBuzzer);
        delay(100);

    }

    delay(200);

    // Emette tre segnali lunghi:
    for(int i = 0; i < 3; i++){
        tone(pinBuzzer, 200);
        delay(300);
        noTone(pinBuzzer);
        delay(100);

    }

    delay(200);

    // Emetti nuovamente tre brevi segnali:
    for(int i = 0; i < 3; i++){
        tone(pinBuzzer, 200);
        delay(100);

        noTone(pinBuzzer);
        delay(100);

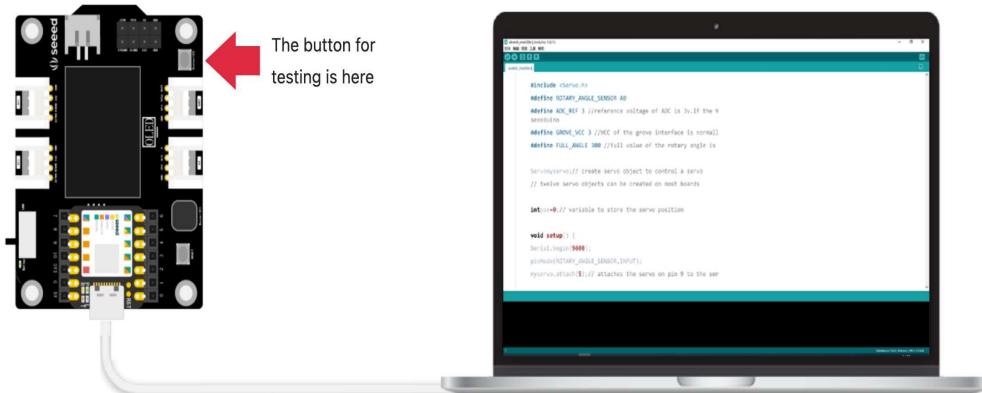
    }

    delay(800); // Attendi prima di ripetere
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3\\_SOS\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3_SOS_XIAO_en)

### Caricamento del Programma

Per caricare il programma sull'hardware, collegare il XIAO al computer utilizzando il cavo dati incluso nel kit. Dopodiché, cliccare sul pulsante di verifica per controllare il programma. Se supera la verifica, cliccare sul pulsante di upload per caricare il programma sull'hardware. Quando l'area di debug mostra “Done uploading.”, si può ascoltare il suono del codice Morse. È questo il ritmo che ci si aspettava?



Notare la posizione dei pulsanti sulle estensioni XIAO utilizzate per il test nella figura.

### 1.3.3 Attività 2: Controllare il buzzer con un pulsante

Il controllo del buzzer con un pulsante per emettere codice Morse può essere fatto manualmente. La logica del codice è semplice: utilizzare un’istruzione if per determinare se il pulsante è premuto. In tal caso, il buzzer emette un suono.

#### **Analisi**

Il programma è scritto anche in tre passaggi:

- Definire i pin del buzzer e del pulsante.
- Inizializzare impostando lo stato dei pin del buzzer e del pulsante.
- Determinare se il pulsante è premuto; se premuto, emettere un suono.

#### **\*Scrivere il programma:\*\***

##### **Passaggio 1: Definire i pin del buzzer e del pulsante**

```
const int buttonPin = 1;      // Il pulsante è collegato al pin 1; adattare alla scheda (ad esempio, D1 per XIAO RP2040/XIAO ESP32)
int buzzerPin = 3;           // Il buzzer è collegato al pin 3; adattalo alla tua scheda (ad esempio, A3 per XIAO RP2040/XIAO ESP32)
```

##### **Passaggio 2: Impostare gli stati del pulsante e del pin del buzzer**

```
void setup() {
    // Imposta il pin del buzzer come output
    pinMode(buzzerPin, OUTPUT);
    // Configura il pin del pulsante come input con resistenza pull-up interna
    pinMode(buttonPin, INPUT_PULLUP);
}
```

**Passaggio 3:** Controllare lo stato del pulsante, se il pulsante è premuto, il buzzer suona. Qui, la funzione `tone()` viene utilizzata per controllare il buzzer passivo affinché emetta un suono.

```
void loop() {
    // Leggi lo stato del pulsante
    int buttonState = digitalRead(buttonPin);

    // Se il pulsante è premuto
    if (buttonState == LOW) {
        // Fin quando il pulsante rimane premuto, continua a suonare il buzzer
        while (buttonState == LOW) {
            tone(buzzerPin, 200); // Fai suonare il buzzer a 200 Hz
            delay(10);           // Piccolo ritardo per evitare il sovraccarico della CPU; può essere regolato o può essere utilizzato un metodo di debouncing più sofisticato
            buttonState = digitalRead(buttonPin); // Controlla continuamente lo stato del pulsante all'interno del ciclo
        }

        // Quando il pulsante viene rilasciato, assicurati che il cicalino si fermi
        noTone(buzzerPin);
    }
}
```

**⚠️ Nota:** Ci sono due pulsanti identici sulla scheda di espansione, uno è il pulsante RESET, che è più vicino all’interfaccia Type-C, e l’altro è il pulsante definito dall’utente, che è più vicino all’interfaccia della batteria al litio. Durante il test, premere quello più vicino all’interfaccia della batteria al litio.

Il programma completo è il seguente:

```
/*
 * Button-Controlled Buzzer
 */
const int buttonPin = 1;      // Button is connected to pin 1; adjust for your board (e.g., D1 for XIAO RP2040/XIAO ESP32)
int buzzerPin = 3;           // Buzzer is connected to pin 3; adjust for your board (e.g., A3 for XIAO RP2040/XIAO ESP32)

void setup() {
    // Imposta il pin del buzzer come output
    pinMode(buzzerPin, OUTPUT);
    // Configura il pin del pulsante come input con resistenza pull-up interna
    pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
    // Leggi lo stato del pulsante
    int buttonState = digitalRead(buttonPin);
```

```

// Se il pulsante è premuto
if (buttonState == LOW) {
    // Fin quando il pulsante rimane premuto, continua a suonare il buzzer
    while (buttonState == LOW) {
        tone(buzzerPin, 200); // Fai suonare il buzzer a 200 Hz
        delay(10); // Piccolo ritardo per evitare il sovraccarico della CPU; può essere
regolato o può essere utilizzato un metodo di debouncing più sofisticato
        buttonState = digitalRead(buttonPin); // Controlla continuamente lo stato del pulsante
all'interno del ciclo
    }

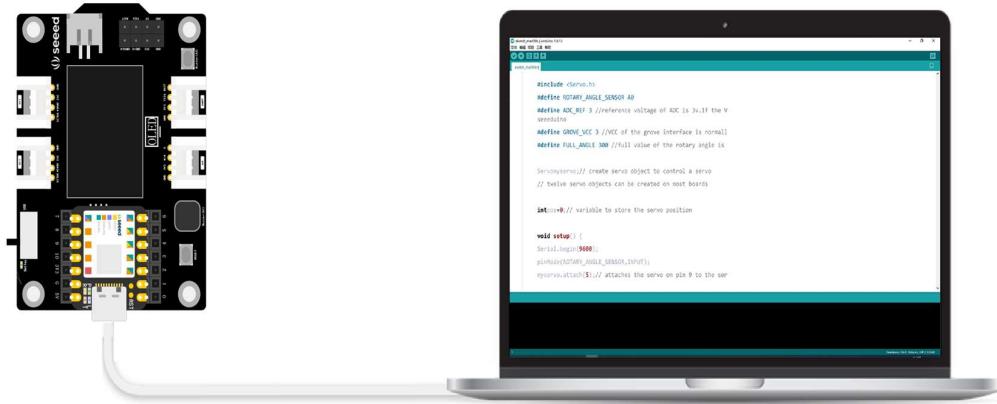
    // Quando il pulsante viene rilasciato, assicurati che il cicalino si fermi
    noTone(buzzerPin);
}
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3\\_ButtonSOS\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3_ButtonSOS_XIAO_en)

#### **Caricamento del programma**

Caricheremo il programma scritto sull'hardware. Per prima cosa, si collega XIAO al computer usando il cavo dati del kit.



Quindi, si clicca su (verify button) per convalidare il programma. Se non ci sono errori, si clicca su (upload button) per caricare il programma sull'hardware. Quando l'area di debug mostra “Done uploading.”, possiamo premere il pulsante sulla scheda di espansione XIAO e verificare se il cicalino suonerà.

#### **1.3.4 Esercizio Esteso**

Il cicalino passivo può emettere toni diversi per formare una semplice melodia. Cercare su un motore di ricerca come far suonare le note ad Arduino. Si può aprire il codice dell'esercizio esteso per sperimentare l'effetto di suonare “Happy Birthday” con il cicalino.

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3\\_HappyBirthday\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L3_HappyBirthday_en)

## **1.4 Monitorare i Cambiamenti di Valore della Manopola con il Serial Monitor**

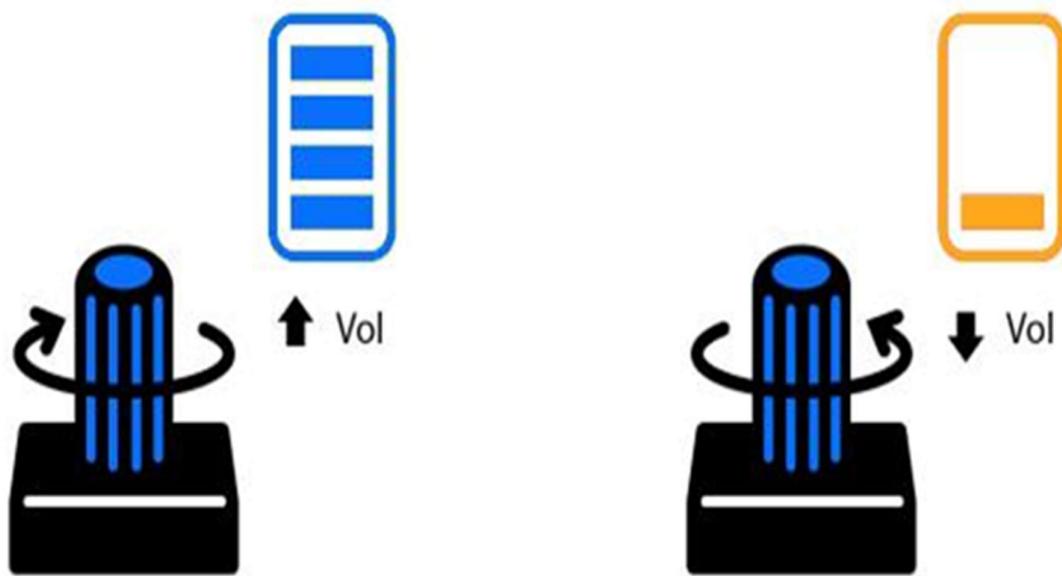
Quando scriviamo alcune righe di codice per controllare la scheda per accendere il LED o per usare un pulsante per controllare il cicalino, possiamo vedere intuitivamente lo stato del funzionamento di questi hardware esterni. Se si ottengono i risultati attesi, si è molto fortunati. E se non è così? Il programma si compila senza errori, dov'è l'errore? Sarebbe bello se potessero parlare. In questa sezione, impareremo come comunicare col computer tramite

il monitor seriale e controllare lo stato di esecuzione e le informazioni del programma e dell'hardware.

## 1.4.1 Conoscenza di Base

### 1.4.1.1 Potenziometro Rotativo

Il potenziometro rotativo, sebbene non sembri comune, ha una vasta gamma di utilizzi negli elettrodomestici e nelle apparecchiature industriali. Ad esempio, la manopola del volume sul sistema audio.



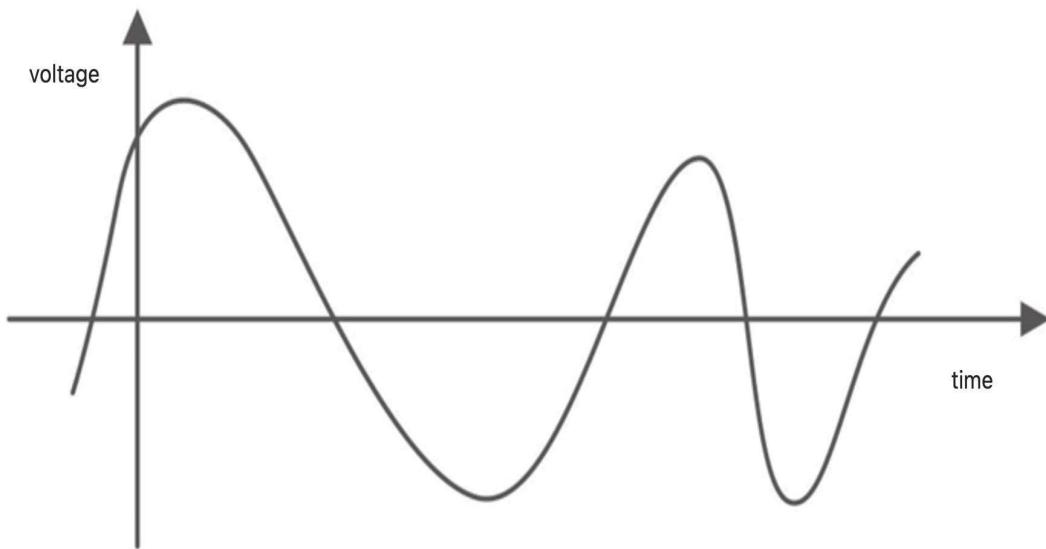
Il potenziometro rotativo può produrre un valore di uscita analogico tra 0 e VCC (la tensione del circuito collegato) sui suoi pin collegati. Ruotando la manopola, è possibile modificare il valore della tensione di uscita. L'intervallo dell'angolo della manopola è di 300° e il valore di uscita è 0-1023. Possiamo utilizzare il potenziometro rotativo per controllare il LED per mostrare cambiamenti di luminosità o controllare il servo per ruotare a diverse angolazioni, ecc.

### 1.4.1.2 I/O Analogico

Nella serie di schede di sviluppo Arduino, i pin con "A" davanti al numero del pin sono pin di input analogico. Possiamo leggere il valore analogico su questi pin per ottenere l'effetto desiderato.

#### Segnale Analogico

Nella vita, i segnali analogici sono ovunque, come il cambiamento di suono, luce, temperatura, ecc., la frequenza, l'ampiezza, ecc. del segnale possono cambiare continuamente nel tempo.



Quindi come leggiamo il valore analogico del pin tramite la scheda di sviluppo? Il pin di input analogico ha un ADC (convertitore analogico-digitale), che può convertire il segnale analogico di input esterno in un segnale digitale che la scheda di sviluppo può riconoscere, ottenendo così la funzione di lettura in valori analogici, ovvero può convertire un segnale di tensione 0-5 V in un valore intero 0-1023.

- `analogRead();`

Legge il valore dal pin analogico specificato. **Sintassi** `analogRead(pin);` **Parametri** `pin:` Il nome del pin di ingresso analogico da leggere.

- `analogWrite();`

In corrispondenza all'ingresso analogico c'è l'uscita analogica. Utilizziamo la funzione `analogWrite()` per ottenere questa funzione. Va notato che quando si utilizza questa funzione, è solo un modo speciale attraverso cui si emettono tensioni diverse per ottenere l'effetto di valori analogici approssimativi. Questo metodo è chiamato modulazione di larghezza di impulso (pulse width modulation - PWM), quindi stiamo scrivendo onde quadre PWM sul pin specificato, non sul vero valore analogico.

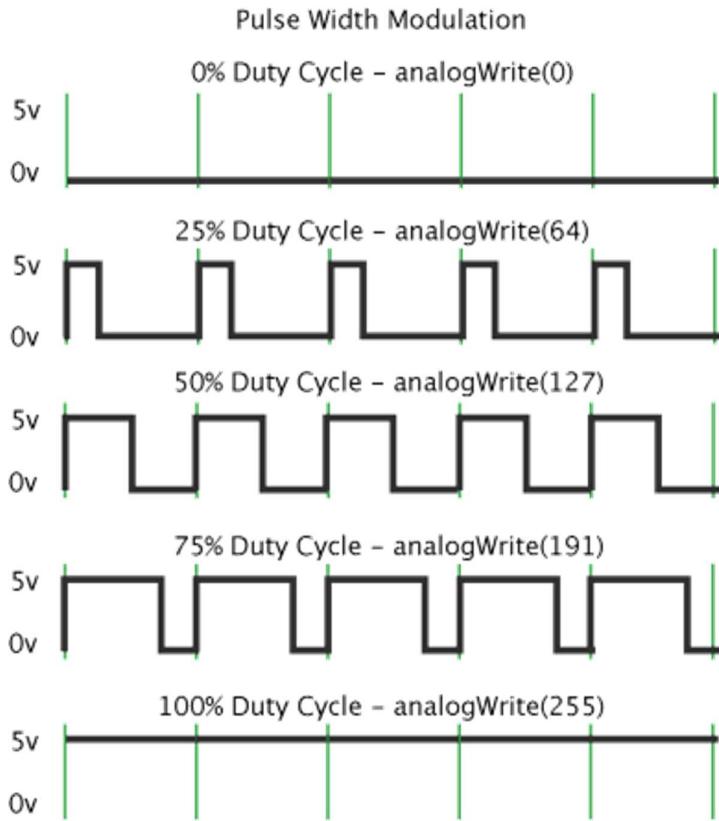
**Sintassi** `analogWrite(pin, value);`

**Parametri** `pin:` Il pin per l'emissione PWM, tipo di dato consentito: int. `value:` Duty cycle, tra 0-255, tipo di dato consentito: int.

## PWM Pulse Width Modulation

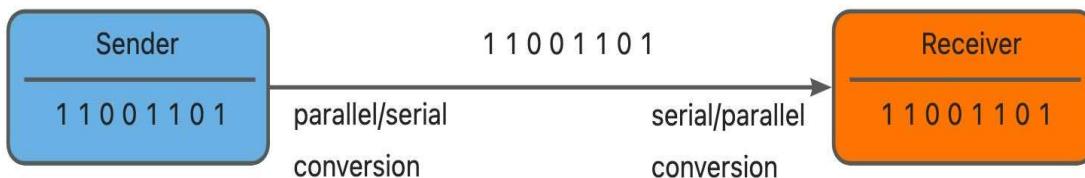
La “Pulse width modulation (PWM)” [modulazione di larghezza di impulso] è un modo per ottenere risultati analogici tramite un’uscita digitale. In parole povere, si può controllare la corrente di carica regolando il periodo di PWM e il “duty cycle” di PWM. Come mostrato nella figura, la tensione viene commutata avanti e indietro tra 0 V (livello basso) e 5 V (livello alto). Uno switchback è un periodo. In questo periodo, se il tempo di tensione alta è del 25% e il tempo di tensione bassa è del 75%, il ciclo di lavoro è del 25% e la tensione di uscita è di 5 V. Quando scriviamo alcune righe di codice per controllare l'accensione dei LED sulla scheda di sviluppo o utilizziamo pulsanti per controllare il ciclino, possiamo osservare direttamente lo stato di funzionamento di questi hardware esterni. Se raggiunge i risultati attesi, siamo fortunati. E se non è così? Il programma si compila senza errori, quindi dov'è il problema? Sarebbe bello se potessero parlare. In questa sezione, impareremo

come comunicare con il computer, monitorare lo stato di esecuzione e le informazioni del programma e dell'hardware tramite il monitor seriale.



#### 1.4.1.3 Comunicazione Seriale

Quando vogliamo comunicare con altri dispositivi tramite XIAO, il metodo più comune è la comunicazione seriale. Tutte le schede di sviluppo della serie Arduino hanno questa funzionalità. Come sappiamo, i computer comprendono i dati binari (come 1010). Pertanto, tra i dispositivi elettronici, la comunicazione seriale svolge la sua funzione inviando e ricevendo tali dati. Il componente chiave per implementare questa funzione è l'USART (Universal Synchronous/Asynchronous Receiver Transmitter). Nell'IDE Arduino, possiamo osservare i dati inviati e ricevuti tramite il Serial Monitor e abbiamo bisogno di funzioni di comunicazione seriale correlate per implementare questa funzionalità.



- **Serial.begin();**

Questa funzione viene utilizzata per aprire la porta seriale e impostare la velocità di trasmissione dei dati.

**Sintassi** `Serial.begin(speed);`

**Parametri** `Serial`: L'oggetto porta seriale. `speed`: Velocità in baud, comunemente impostata su valori come 9600, 115200, ecc.

- **Serial.println();**

**Sintassi** `Serial.println(val);`

**Parametri** `Serial`: L'oggetto porta seriale. `val`: Il valore da stampare, che può essere di qualsiasi tipo di dati.

Ad esempio, per stampare “hello world!!!” sul Serial Monitor, dobbiamo inizializzare la porta seriale nella funzione `setup()` e inviare in output “hello world!!!” tramite la porta seriale nella funzione `loop()`:

---

```
void setup() {
    Serial.begin(9600); // Inizializza la porta seriale e imposta la velocità di trasmissione dati a 9600
}
void loop() {
    Serial.println("hello world!!!"); // Output "hello world!!!" through the serial port
}
```

---

Tornando alla domanda all'inizio di questa sezione: quando abbiamo scritto il codice e verificato che fosse corretto, ma l'effetto dell'esecuzione del codice supera le aspettative o l'hardware non risponde affatto, dov'è il problema? A questo punto, possiamo usare il Serial Monitor per osservare i dati inviati o ricevuti dall'hardware per esprimere un giudizio. Ad esempio, possiamo controllare lo stato acceso-spento di un LED con un pulsante e possiamo usare il monitor seriale per controllare il valore restituito quando il pulsante viene premuto per determinare se il pulsante funziona correttamente. Successivamente, impareremo come usare il Serial Monitor per far “parlare” l'hardware.

## 1.4.2 Attività 1: Utilizzare il Serial Monitor per Verificare se il pulsante è Premuto

### Analisi

Ricordate quando si controllava lo stato on-off di un LED con un pulsante? Parte del codice può essere riutilizzato. Dobbiamo solo leggere l'impostazione on-off del pulsante e il codice dello stato on-off del pulsante, quindi aggiungere l'inizializzazione della porta seriale e i dati inviati alla porta seriale. La scrittura del programma segue ancora tre passaggi:

- Definire i pin e le variabili del pulsante.
- Inizializzare la porta seriale, impostare la velocità in baud della porta seriale e impostare lo stato del pin on-off del pulsante.
- Leggere lo stato del pulsante e inviarlo alla porta seriale.

### \*Scrivere il programma\*\*

#### Passaggio 1: Definire il pin e la variabile del pulsante.

---

```
const int buttonPin = 1; // Definisce l'interruttore del pulsante come pin 1. Se si sta utilizzando XIAO RP2040/XIAO ESP32, cambiare 1 in D1
int buttonState = 0; // Definisce buttonState come variabile per memorizzare lo stato del pulsante
```

---

#### Passaggio 2: Inizializzare la porta seriale, impostare la velocità in baud della porta seriale e impostare lo stato del pin dell'interruttore del pulsante.

---

```
void setup() {
    pinMode(buttonPin, INPUT_PULLUP); // Imposta il pin del pulsante come input
    Serial.begin(9600); // Inizializza la porta seriale
}
```

---

#### Passaggio 3: Leggere lo stato del pulsante e inviarlo alla porta seriale

---

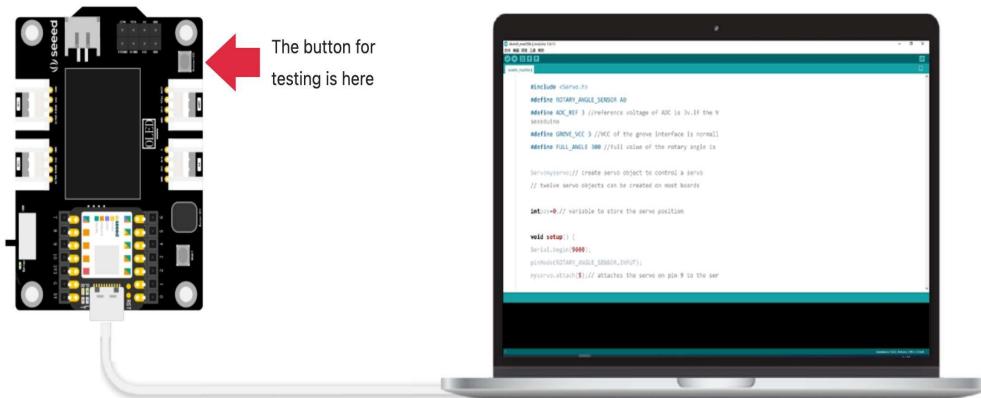
```
void loop() {
    buttonState = digitalRead(buttonPin); // Leggi lo stato del pulsante e memorizzalo nella variabile buttonState
    Serial.println(buttonState); // Invia i dati sullo stato del pulsante alla porta seriale
    delay(500);
}
```

---

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L4\\_ReadButton\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L4_ReadButton_XIAO_en)

### Caricamento del programma:

Caricheremo il programma scritto sull'hardware. Per prima cosa, collegare lo XIAO al computer tramite il cavo dati fornito nel kit.



Notare la posizione dei pulsanti sulle estensioni XIAO utilizzate per il test nella figura.

Cliccare su (Verify Button) nell’IDE Arduino per verificare il programma. Se la verifica è corretta, cliccare su (Upload Button) per caricare il programma su XIAO. Quando l’area di debug mostra “Done uploading.”, aprire il monitor seriale e osservare i cambiamenti di valore stampati dal monitor seriale quando il pulsante viene premuto e rilasciato. Cosa si vede?

```

L4_ReadButton_XIAO_en | Arduino IDE 2.0.4

L4_ReadButton_XIAO_en.ino

3 */
4
5 const int buttonPin = 1; // Define the button switch as pin 1. If you are using XIAO RP2040/XIAO ESP32, please change
6 int buttonState = 0; // Define buttonState as a variable to store the button status
7
8 void setup() {
9     pinMode(buttonPin, INPUT_PULLUP); // Set the button pin as input
10    Serial.begin(9600); // Initialize the serial port
11 }
12
13 void loop() {
14     buttonState = digitalRead(buttonPin); // Read the button status and store it in the buttonState variable
15     Serial.println(buttonState); // Send the button status data to the serial port
16     delay(500);
17 }
18

```

Output Serial Monitor X

Message (Enter to send message to 'Seeeduino XIAO' on '/dev/cu.usbmodem101') New Line 115200 baud

```

09:03:54.905 -> 1
09:03:55.398 -> 1
09:03:55.898 -> 1
09:03:56.390 -> 0
09:03:56.917 -> 0
09:03:57.409 -> 0
09:03:57.904 -> 1

```

Ln 17, Col 2 Seeeduino XIAO on /dev/cu.usbmodem101

Quando premiamo il pulsante sulla scheda di espansione XIAO, il monitor seriale mostra 0 e quando rilasciamo il pulsante, il monitor seriale mostra 1.

### 1.4.3 Attività 2: Utilizzo del Serial Monitor per Visualizzare i Cambiamenti di Valore della Manopola

#### **Analisi:**

Nell'attività 1, l'interruttore del pulsante è un ingresso digitale che invia segnali digitali 0 e 1, mentre il potenziometro della manopola restituisce un segnale analogico. Dobbiamo leggere il valore dell'angolo di rotazione del potenziometro a manopola sul pin A0 e inviarlo alla porta seriale. Il programma è composto anche da tre passaggi:

- Definire il pin del potenziometro a manopola e le variabili.
- Inizializzare la porta seriale e impostare lo stato del pin del potenziometro a manopola.
- Leggere e calcolare il valore dell'angolo di rotazione del potenziometro a manopola e inviarlo alla porta seriale.

#### **\*Scrivere il programma\*\***

**Passaggio 1:** Definire il pin del potenziometro a manopola e le variabili. Qui dobbiamo definire il valore di tensione dell'ADC (convertitore analogico-digitale) e la tensione di riferimento dell'interfaccia del modulo Grove, perché calcoleremo le variazioni di tensione nel circuito in cui è collegata la manopola tramite questi valori di tensione.

---

```
#define ROTARY_ANGLE_SENSOR A0 // Definisce l'interfaccia del potenziometro rotativo A0
#define ADC_REF 3 // La tensione di riferimento ADC è 3 V
#define GROVE_VCC 3 // La tensione di riferimento dell'interfaccia Grove è 3 V
#define FULL_ANGLE 300 // L'angolo di rotazione massimo del potenziometro a manopola è di 300°
```

---

**Passaggio 2:** Inizializzare la porta seriale, impostare la velocità in baud della porta seriale e impostare lo stato del pin del potenziometro a manopola.

---

```
void setup()
{
    Serial.begin(9600); //Inizializza la porta seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); //Imposta il pin del potenziometro rotativo sullo stato di input
}
```

---

**Passaggio 3:** Leggere e calcolare il valore dell'angolo di rotazione del potenziometro rotativo e inviarlo alla porta seriale. Qui, dobbiamo prima impostare il tipo di dati della variabile di tensione, impostare la variabile del valore analogico del pin del potenziometro rotativo e quindi calcolare la tensione in tempo reale. Dopo aver calcolato la tensione in tempo reale, calcolare il valore dell'angolo di rotazione del potenziometro rotativo.

---

```
void loop()
{
    float voltage; //La tensione variabile è di tipo a virgola mobile
    int sensorValue = analogRead(ROTARY_ANGLE_SENSOR); //Leggi il valore analogico sul pin del potenziometro rotativo
    voltage = (float)sensorValue*ADC_REF/1023; //Calcola la tensione in tempo reale
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC; //Calcola l'angolo di rotazione della manopola
    Serial.println("The angle between the mark and the starting position:"); //Stampa i caratteri sulla porta seriale
    Serial.println(degrees); //Stampa il valore dell'angolo di rotazione del potenziometro rotativo sulla porta seriale
    delay(100);
}
```

---

#### **#define Definizione della Macro**

**#define** è un comando di pre-elaborazione utilizzato per le definizioni macro. In Arduino, possiamo utilizzare **#define** per nominare le costanti. Durante la compilazione del programma, tutte le occorrenze del "nome macro" verranno sostituite con la stringa nella definizione macro, come **#define ledPin 5**. Durante la compilazione, 5 sostituirà tutti gli utilizzi di **ledPin**. Sintassi: **#define nome costante valore costante**. Il simbolo "#" è obbligatorio e non è necessario utilizzare il simbolo ";" alla fine dell'istruzione.

Il codice completo è il seguente:

```
/*
 * Use the serial monitor to view the knob potentiometer
 */
#define ROTARY_ANGLE_SENSOR A0 //Define the rotary potentiometer interface A0
#define ADC_REF 3 //ADC reference voltage 3V
#define GROVE_VCC 3 //Reference voltage 3V
#define FULL_ANGLE 300 //The maximum rotation angle of the rotary potentiometer is 300°

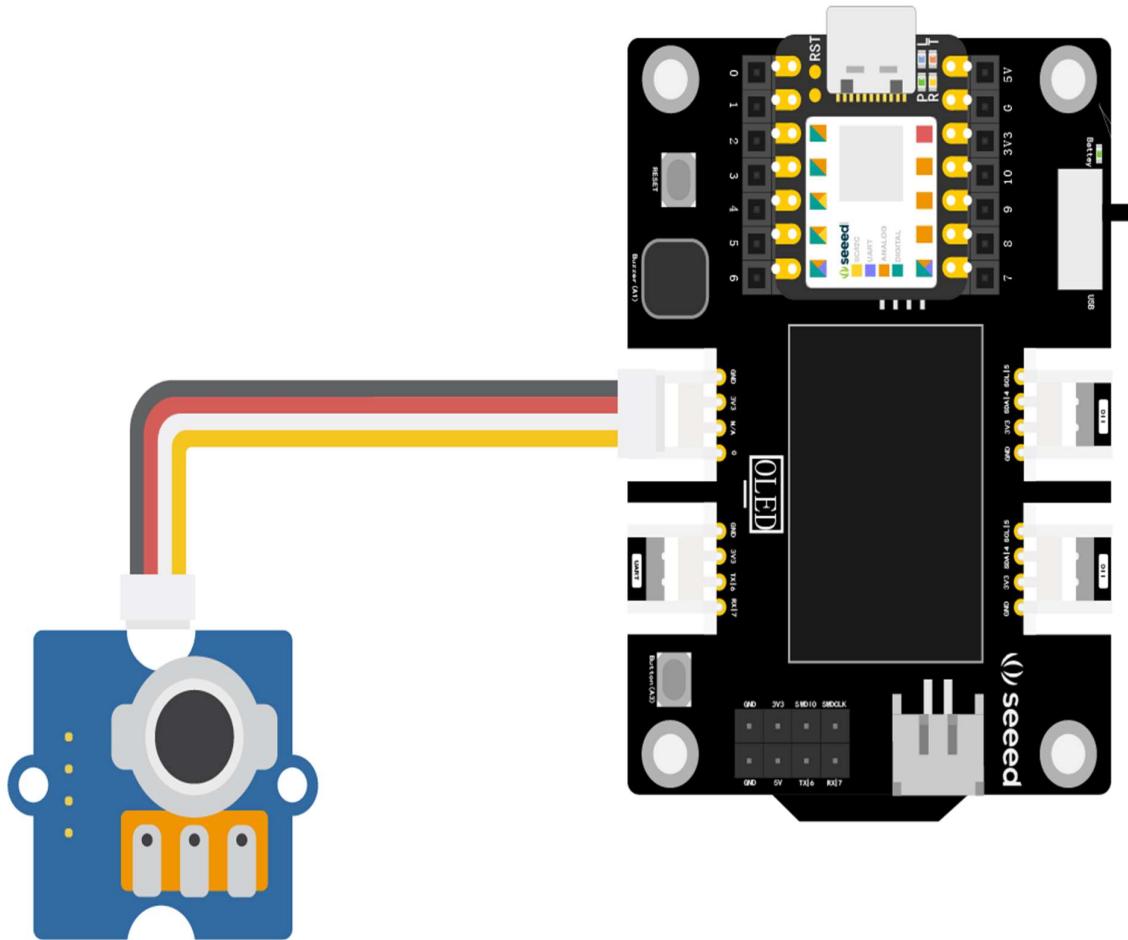
void setup()
{
    Serial.begin(9600); //Inizializza la porta seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); //Set the rotary potentiometer pin as an input
}

void loop()
{
    float voltage; //Variable voltage is of floating-point type
    int sensorValue = analogRead(ROTARY_ANGLE_SENSOR); //Read the analog value at the rotary
potentiometer pin
    voltage = (float)sensorValue*ADC_REF/1023; //Calculate real-time voltage
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC; //Calculate the rotation angle of the knob
    Serial.println("The angle between the mark and the starting position:"); //Print characters at
the serial port
    Serial.println(degrees); //Print the rotation angle value of the rotary potentiometer at the
serial port
    delay(100);
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L4\\_ReadRotary\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L4_ReadRotary_XIAO_en)

#### **Caricamento del programma:**

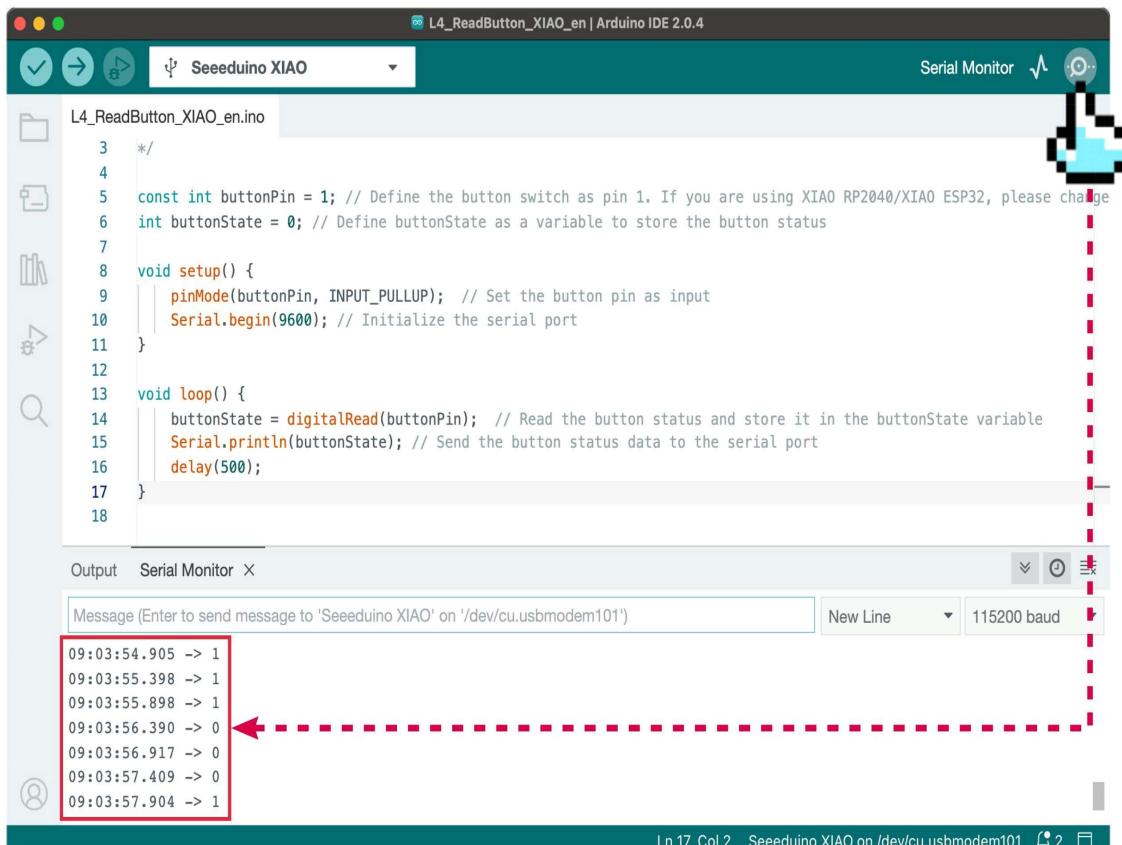
Dopo aver scritto il programma, poiché vengono utilizzati sensori esterni, collegare il modulo manopola all'interfaccia A0 utilizzando il cavo Grove a quattro colori come mostrato nell'immagine sottostante:



Dopo la connessione, collegare la scheda di controllo principale XIAO al computer utilizzando un cavo dati.

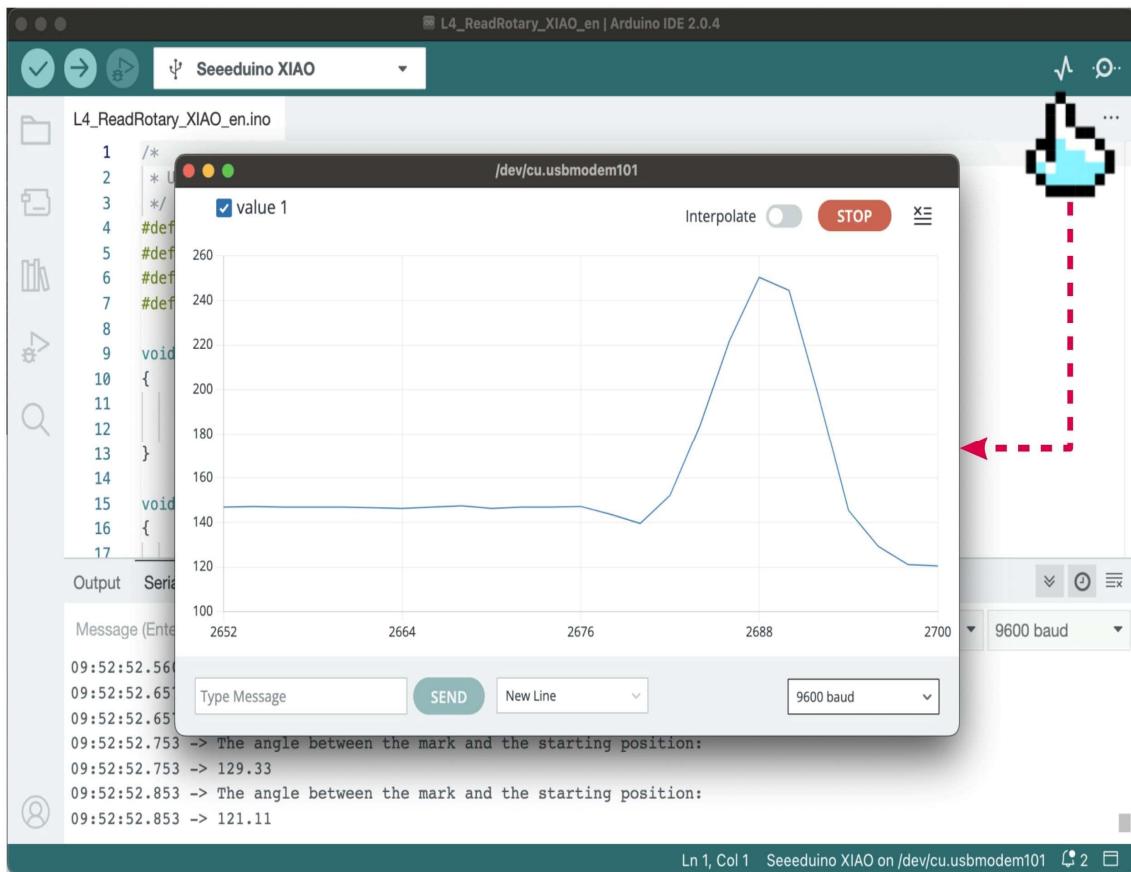
Nell'IDE Arduino, cliccare sul pulsante di verifica per verificare il programma. Se la verifica è corretta, cliccare sul pulsante di caricamento per caricare il programma nell'hardware.

Quando l'area di debug mostra "Done uploading.", si può procedere. Aprire il monitor seriale e ruotare il potenziometro della manopola per osservare le modifiche dei dati visualizzate nel monitor seriale. Queste modifiche rappresentano il valore dell'angolo della manopola.



#### 1.4.4 Esercizio Esteso

Osservando il valore angolare del potenziometro della manopola nel monitor seriale, scopriamo che il valore salta e cambia costantemente. Osservare solo attraverso i numeri non è molto intuitivo. A questo punto, possiamo usare il plotter seriale. Con esso, possiamo tracciare i dati che vengono stampati sulla porta seriale dell'Arduino in tempo reale. Sulla base del secondo compito, chiudere il monitor seriale e aprire "**Tools > Serial Plotter**" come mostrato nell'immagine sottostante:



Il plotter seriale disegna i dati ottenuti dalla porta seriale in un grafico a curve dell'asse XY, dove l'asse X rappresenta la variazione nel tempo e l'asse Y rappresenta i dati ottenuti dalla porta seriale. Attraverso il grafico, si può vedere in modo più intuitivo la variazione nei dati. Da provare.

## 1.5 Controllo del LED e del Servo con una Manopola

Nell'ultima sezione, abbiamo imparato come usare il monitor seriale e osservato, con esso, le differenze tra input digitale e input analogico. In questa sezione, esploreremo ulteriormente l'uso di valori analogici combinandoli con un potenziometro rotativo!

### 1.5.1 Conoscenza di Base

#### 1.5.1.1 Servo e Libreria Servo

##### Servo

Un “servo”, noto anche come “servomotore”, è un motore a corrente continua con ingranaggi e un sistema di feedback. Possiamo controllare il servo per ruotare in una posizione angolare specifica inviando segnali al circuito. Ciò lo rende adatto per dispositivi elettronici o robot che richiedono un controllo di posizione preciso.

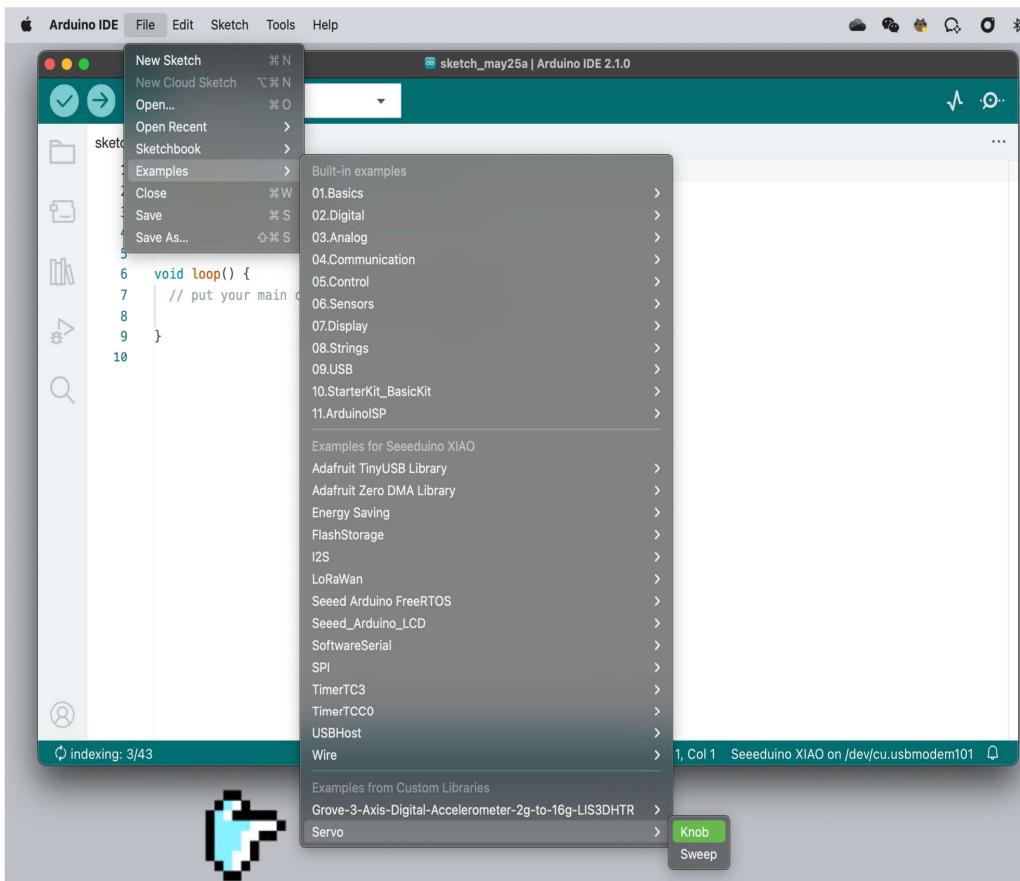
##### **Libreria Servo servo.h**

Quando vogliamo controllare un servo usando XIAO o altre schede di sviluppo Arduino, possiamo usare il file di libreria servo.h. È una delle librerie standard Arduino, che è comoda da usare ed evita anche il problema della quantità limitata di pin PWM. Ecco le funzioni rilevanti della libreria servo:

- Dichiarare il file di libreria `#include <Servo.h>`

- Creare l'oggetto `myservo` per controllare il servo `servo myservo`;
- Usare la funzione `attach()` per chiamare il pin del segnale `myservo.attach()`;
- Usare la funzione `write()` per scrivere l'angolo sul servo, impostando l'angolo di rotazione dell'albero `myservo.write()`;

Non è necessario installare manualmente la libreria servo. Si può aprire il programma di esempio “File → Examples → Servo” e controllare i due programmi di esempio “Knob” e “Sweep” per familiarizzare con l'uso della libreria servo.



Se non si riesce a trovare Servo in “Examples”, si può visitare <https://github.com/arduino-libraries/Servo> e aggiungere l'esempio Servo installando la libreria.

### 1.5.1.2 Funzione `map()`

La funzione `map()` viene utilizzata per mappare un numero da un intervallo a un altro. Vale a dire, `fromLow` viene mappato in `toLow` e `fromHigh` viene mappato in `toHigh`. È la forma più semplice di mappatura lineare.

**Sintassi** `map(value, fromLow, fromHigh, toLow, toHigh)`

**Parametri** `value`: Il numero da mappare. `fromLow`: Il limite inferiore dell'intervallo corrente del valore. `fromHigh`: Il limite superiore dell'intervallo corrente del valore. `toLow`: Il limite inferiore dell'intervallo di destinazione del valore. `toHigh`: Il limite superiore dell'intervallo di destinazione del valore.

**Esempio: Mappare `val` dall'intervallo 0-1023 a 0-255.**

```
void setup() {}
void loop() {
    int val = analogRead(0); // legge il valore dal pin analogico A0
    val = map(val, 0, 1023, 0, 255); // mappa val all'intervallo 0-255
    analogWrite(9, val); // emetti il valore analogico sul pin 9
}
```

## 1.5.2 Attività 1: Utilizzo di un potenziometro a manopola per controllare la luminosità del LED a bordo della scheda XIAO

### **Analisi:**

Quando si utilizza un potenziometro a manopola per controllare il LED, dobbiamo usare la funzione map(), perché il valore analogico emesso direttamente dal potenziometro a manopola è 0-1023, questo valore non è il valore dell'angolo di rotazione della manopola, dobbiamo prima calcolare il valore dell'angolo di rotazione del potenziometro a manopola, poi mappare questo valore all'intervallo di luminosità del LED 0-255 con la funzione map(). I passaggi per scrivere il programma sono i seguenti:

- Definire il potenziometro a manopola, il pin LED.
- Inizializzare la porta seriale, impostare lo stato del potenziometro a manopola e del pin LED.
- Leggere e calcolare il valore dell'angolo di rotazione del potenziometro a manopola e inviarlo alla porta seriale.
- Mappare il valore dell'angolo del potenziometro a manopola sul valore di luminosità del LED e memorizzarlo nella variabile luminosità, e il LED emette questo valore variabile.

### **Scrittura del programma:**

**Passaggio 1:** Definire il potenziometro a manopola, il pin LED, qui dobbiamo definire la tensione di riferimento ADC e VCC, per calcolare il valore dell'angolo del potenziometro a manopola.

```
#define ROTARY_ANGLE_SENSOR A0 //Definisci l'interfaccia del potenziometro rotativo A0
#define LEDPIN 13 //Definisci l'interfaccia LED 13
#define ADC_REF 3 //Tensione di riferimento 3 V
#define GROVE_VCC 3 //Tensione di riferimento GROVE 3 V
#define FULL_ANGLE 300 //L'angolo di rotazione massimo del potenziometro è 300°
```

**Passaggio 2:** Inizializzare la porta seriale, impostare lo stato del potenziometro a manopola e del pin LED.

```
void setup()
{
    Serial.begin(9600); //Inizializza la comunicazione seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); //Imposta il pin del potenziometro rotativo su input
    pinMode(LEDPIN, OUTPUT); //Imposta il pin LED su output
}
```

**Passaggio 3:** Leggere e calcolare il valore dell'angolo di rotazione del potenziometro a manopola e inviarlo alla porta seriale.

```
void loop()
{
    float voltage; //Variabile voltage di tipo float
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR); //Leggere il valore analogico sul pin del potenziometro rotativo
    voltage = (float)sensor_value*ADC_REF/1023; //Calcolare la tensione in tempo reale
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC; //Calcolare l'angolo di rotazione della manopola
    Serial.println("The angle between the mark and the starting position:"); //Stampare il carattere sul monitor seriale
    Serial.println(degrees); //Stampare il valore dell'angolo di rotazione del potenziometro rotativo sul monitor seriale
    delay(100);
}
```

**Passaggio 4:** Mappare il valore dell'angolo del potenziometro della manopola sul valore di luminosità del LED e memorizzarlo nella variabile luminosità, e il LED emette questo valore variabile.

```
//Dopo il Passaggio 3
    int brightness; //Definisci la variabile della luminosità
    brightness = map(degrees, 0, FULL_ANGLE, 0, 255); //Mappa il valore dell'angolo di rotazione del potenziometro rotativo sul valore di luminosità del LED e memorizzarlo nella variabile luminosità
    analogWrite(LEDPIN,brightness); //Invia il valore della variabile al LED
    delay(500);
}
```

Il codice completo finale è mostrato di seguito:

```

#define ROTARY_ANGLE_SENSOR A0 //Definisci l'interfaccia del potenziometro rotativo A0
#define LEDPIN 13 //Definisci l'interfaccia LED 13
#define ADC_REF 3 //Tensione di riferimento 3 V
#define GROVE_VCC 3 //Tensione di riferimento GROVE 3 V
#define FULL_ANGLE 300 //L'angolo di rotazione massimo del potenziometro è 300°

void setup()
{
    Serial.begin(9600); //Inizializza la comunicazione seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); //Imposta il pin del potenziometro rotativo su input
    pinMode(LEDPIN,OUTPUT); //Imposta il pin LED su output
}

void loop()
{
    float voltage; //Variabile voltage di tipo float
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR); //Leggere il valore analogico sul pin del
    potenziometro rotativo
    voltage = (float)sensor_value*ADC_REF/1023; //Calcolare la tensione in tempo reale
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC; //Calcolare l'angolo di rotazione della manopola
    Serial.println("The angle between the mark and the starting position:"); //Stampare il carattere
    sul monitor seriale
    Serial.println(degrees); //Stampare il valore dell'angolo di rotazione del potenziometro
    rotativo sul monitor seriale
    delay(100);

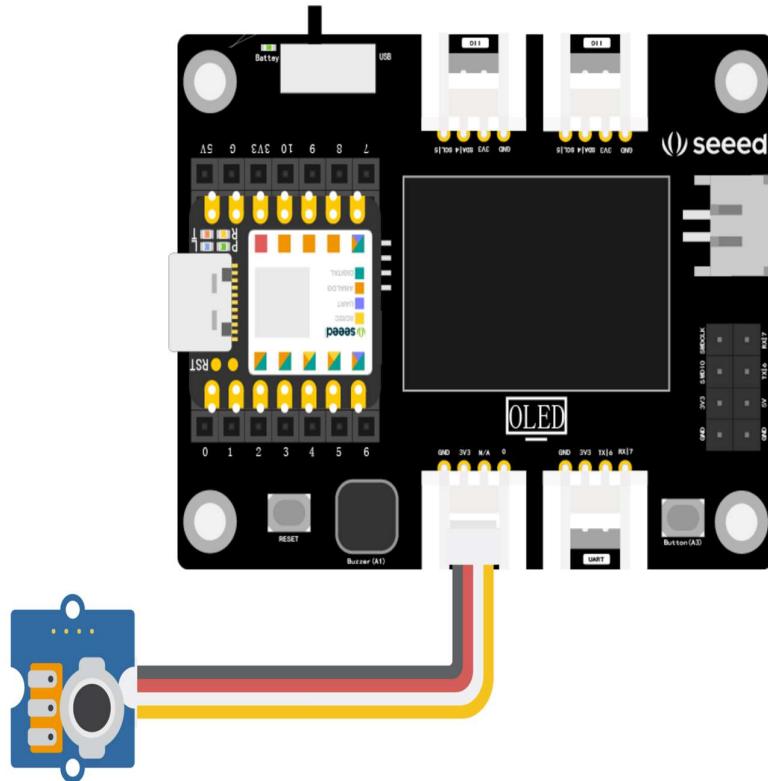
    int brightness; //Definisci la variabile della luminosità
    brightness = map(degrees, 0, FULL_ANGLE, 0, 255); //Mappa il valore dell'angolo di rotazione del
    potenziometro rotativo sul valore di luminosità del LED e memorizzalo nella variabile luminosità
    analogWrite(LEDPIN,brightness); //Invia il valore della variabile al LED
    delay(500);
}

```

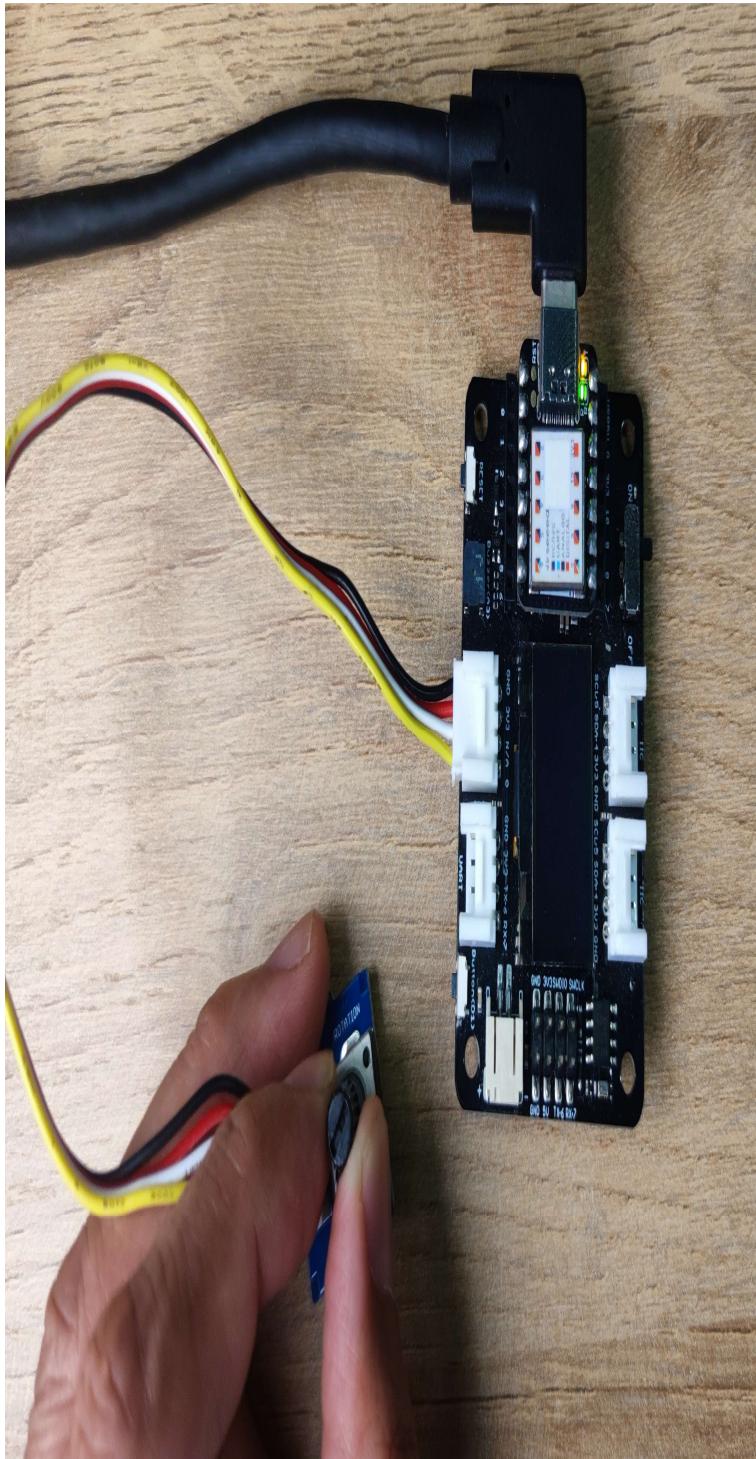
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5\\_RotaryLed\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5_RotaryLed_XIAO_en)

### **Caricamento del Programma:**

Dopo aver scritto il programma, collegare il potenziometro rotativo all'interfaccia A0 utilizzando un filo Grove a quattro colori, come mostrato nella figura seguente:

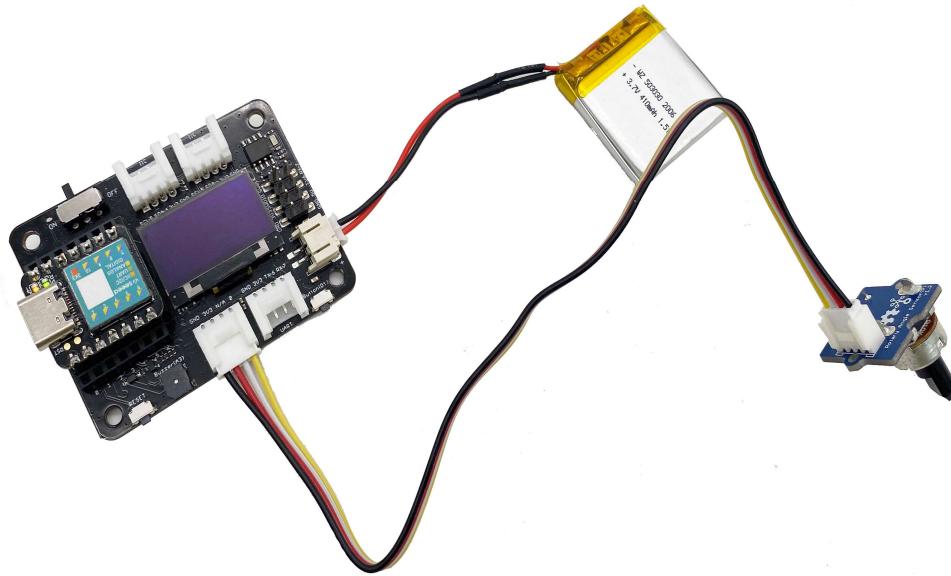


Collegare la scheda di controllo principale XIAO al computer con un cavo dati. Dopo la connessione, cliccare su (il pulsante di verifica) nell'IDE Arduino per controllare il programma. Se non ci sono errori, cliccare (upload button) per caricare il programma sull'hardware. Quando l'area di debug mostra “Done uploading.”, si può aprire il monitor seriale per osservare i valori dell'angolo di rotazione e della luminosità del LED mentre si ruota il potenziometro.



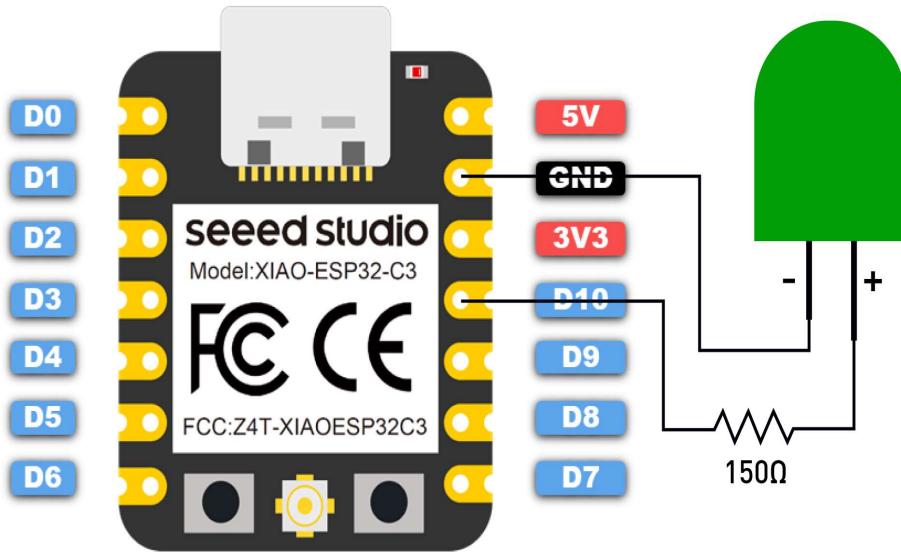
**⚠️ Nota:** In questo esempio viene utilizzato il LED integrato della scheda XIAO.

Se si deve operare offline, si può collegare una batteria al litio alla scheda di espansione, come mostrato nella figura seguente.



### Controllo di un LED esterno con una manopola su XIAO ESP32C3

Seeed XIAO ESP32C3 non ha un LED integrato per gli utenti. Per eseguire questo programma, si deve prima collegare un LED al pin `D10` della scheda, come mostrato di seguito:



**⚠️ Nota:** Si deve collegare una resistenza (circa 150Ω) in serie al LED per limitare la corrente che passa attraverso di esso ed evitare che venga danneggiato da sovraccorrente.

Poi, si copia il seguente programma nell'IDE Arduino:

```
#define ROTARY_ANGLE_SENSOR A0 // Definisce l'interfaccia del potenziometro rotativo A0
#define LEDPIN D10 // Definisce l'interfaccia del LED 10
#define ADC_REF 3 // Tensione di riferimento 3 V
#define GROVE_VCC 3 // Tensione di riferimento GROVE 3 V
#define FULL_ANGLE 300 // L'angolo di rotazione massimo del potenziometro è 300°
```

```

void setup()
{
    Serial.begin(9600); // Inizializza la comunicazione seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); // Imposta il pin del potenziometro rotativo in modalità
    input
    pinMode(LEDPIN, OUTPUT); // Imposta il pin della luce LED in modalità output
}

void loop()
{
    float voltage; // Definisce la variabile di tensione come float
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR); // Leggi il valore analogico sul pin del
    potenziometro rotativo
    voltage = (float)sensor_value*ADC_REF/1023; // Calcola la tensione in tempo reale
    float degrees = (voltage*FULL_ANGLE)/GROVE_VCC; // Calcola l'angolo di rotazione della manopola
    Serial.println("The angle between the mark and the starting position:"); // Stampa la stringa
    sulla porta seriale
    Serial.print(degrees); // Stampa il valore dell'angolo di rotazione del potenziometro rotativo
    sulla porta seriale
    delay(100);

    int brightness; // Definisci la variabile di luminosità
    brightness = map(degrees, 0, FULL_ANGLE, 0, 255); // Mappa il valore dell'angolo del
    potenziometro rotativo sul valore di luminosità della luce LED e memorizzalo nella variabile di
    luminosità
    analogWrite(LEDPIN, brightness); // Valore di luminosità in uscita sul LED
    delay(500);
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5\\_RotaryLed\\_XIAO\\_ESP32C3\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5_RotaryLed_XIAO_ESP32C3_en)

## 1.5.3 Attività 2: Controllo di un Servomotore con un Potenziometro Rotativo

### **Analisi**

Quando si controlla un servomotore con un potenziometro rotativo, possiamo usare la libreria `servo.h` e modificare leggermente la nostra prima attività. Il programma può essere suddiviso nei seguenti passaggi:

- Dichiarazione della libreria servo, definizione della variabile dell'angolo di rotazione del servo, definizione del pin e della tensione del potenziometro rotativo.
- Inizializzazione della porta seriale, impostazione dello stato del potenziometro rotativo e dei pin servo.
- Lettura e calcolo del valore dell'angolo di rotazione del potenziometro, invio alla porta seriale e comando del servo per ruotare in base alla variazione del valore dell'angolo.

### **Scrittura del Programma**

**Passaggio 1:** Dichiarazione della libreria servo, definizione della variabile dell'angolo di rotazione del servo, definizione del pin e della tensione del potenziometro rotativo.

```
#include <Servo.h> // Dichiara l'uso della libreria servo
#define ROTARY_ANGLE_SENSOR A0 // Definisci il pin del potenziometro rotativo come A0
#define ADC_REF 3 // La tensione di riferimento ADC è 3 V
#define GROVE_VCC 3 // La tensione di riferimento del modulo GROVE è 3 V
#define FULL_ANGLE 300 // L'angolo di rotazione massimo del potenziometro rotativo è 300°
Servo myservo; // Crea un oggetto myservo per controllare il servo
int pos = 0; // Variabile per memorizzare l'angolo di rotazione del servo
```

**Passaggio 2:** Inizializzare la porta seriale, impostare lo stato del potenziometro rotativo e dei pin del servo.

```
void setup() {
    Serial.begin(9600); // Inizializza la porta seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); // Imposta il pin del potenziometro come input
    myservo.attach(5); // Il segnale myservo viene trasmesso tramite il pin 5, se si utilizza XIAO
    RP2040/XIAO ESP32, modificare 5 in D5
}
```

**Passaggio 3:** Leggere e calcolare il valore dell'angolo di rotazione del potenziometro rotativo, inviarlo alla porta seriale e far ruotare il servo in base alla modifica del valore dell'angolo.

```
void loop() {
    float voltage;// Imposta voltage in virgola mobile
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);// Leggi il valore analogico sul pin del
potenziometro rotativo
    voltage = (float)sensor_value * ADC_REF / 1023;// La tensione in tempo reale è il valore
analogico letto moltiplicato per la tensione di riferimento diviso per 1023
    float degrees = (voltage * FULL_ANGLE) / GROVE_VCC;// L'angolo di rotazione della manopola è la
tensione in tempo reale moltiplicata per l'angolo di rotazione massimo del potenziometro rotativo
diviso per il valore di tensione dell'interfaccia del modulo GROVE
    Serial.println("The angle between the mark and the starting position");// Stampa i caratteri
sulla porta seriale
    Serial.println(degrees);// Stampa il valore dell'angolo di rotazione del potenziometro rotativo
sulla porta seriale
    delay(50);
    myservo.write(degrees); // Scrivere il valore dell'angolo di rotazione del potenziometro
rotativo nel servo
}
```

Il codice finale è il seguente:

```
#include <Servo.h> // Declare the use of the servo library
#define ROTARY_ANGLE_SENSOR A0 // Define the rotary potentiometer pin as A0
#define ADC_REF 3 // ADC reference voltage is 3V
#define GROVE_VCC 3 // GROVE module reference voltage is 3V
#define FULL_ANGLE 300 // The maximum rotation angle of the rotary potentiometer is 300°
Servo myservo; // Create a myservo object to control the servo
int pos = 0; // Variable to store the rotation angle of the servo

void setup() {
    Serial.begin(9600); // Inizializza la porta seriale
    pinMode(ROTARY_ANGLE_SENSOR, INPUT); // Imposta il pin del potenziometro come input
    myservo.attach(5); // The myservo signal is transmitted through pin 5, if you are using XIAO
RP2040/XIAO ESP32, please modify 5 to D5
}

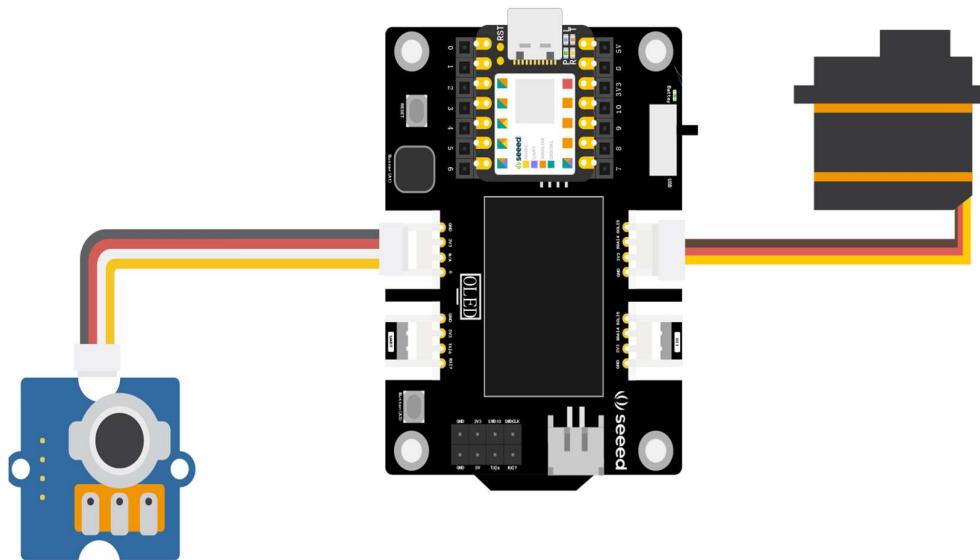
void loop() {
    float voltage;// Imposta voltage in virgola mobile
    int sensor_value = analogRead(ROTARY_ANGLE_SENSOR);// Leggi il valore analogico sul pin del
potenziometro rotativo
    voltage = (float)sensor_value * ADC_REF / 1023;// La tensione in tempo reale è il valore
analogico letto moltiplicato per la tensione di riferimento diviso per 1023
    float degrees = (voltage * FULL_ANGLE) / GROVE_VCC;// L'angolo di rotazione della manopola è la
tensione in tempo reale moltiplicata per l'angolo di rotazione massimo del potenziometro rotativo
diviso per il valore di tensione dell'interfaccia del modulo GROVE
    Serial.println("The angle between the mark and the starting position");// Stampa i caratteri
sulla porta seriale
    Serial.println(degrees);// Stampa il valore dell'angolo di rotazione del potenziometro rotativo
sulla porta seriale
    delay(50);
    myservo.write(degrees); // Scrivere il valore dell'angolo di rotazione del potenziometro
rotativo nel servo
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5\\_RotaryServo\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5_RotaryServo_XIAO_en)

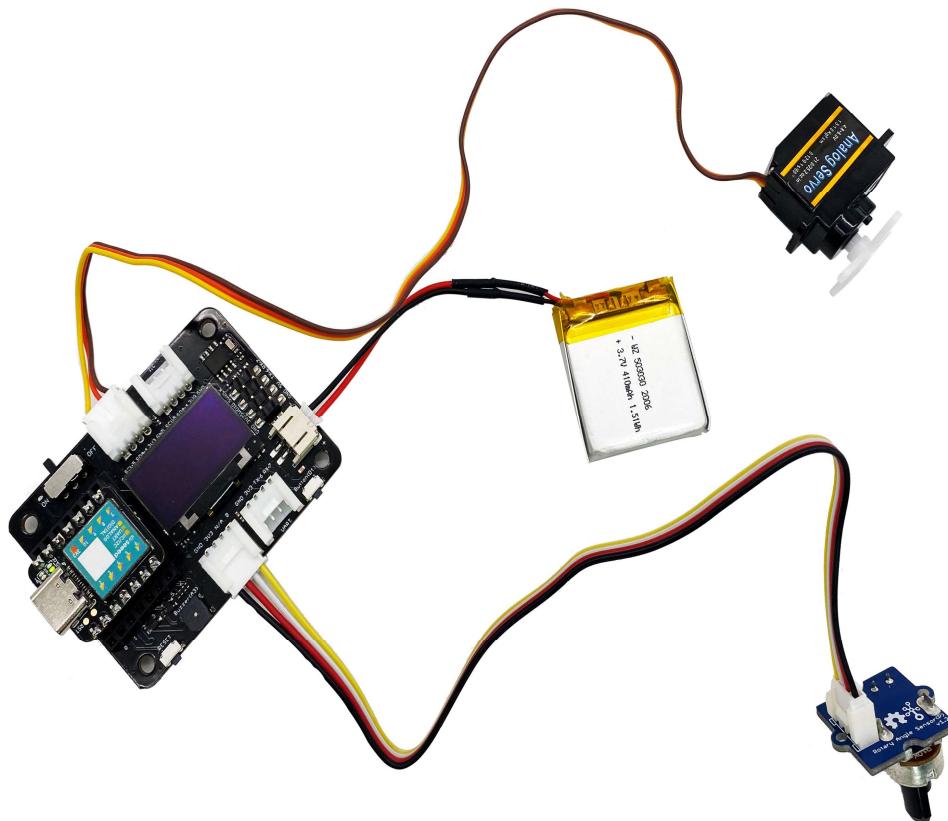
---

### Caricamento del Programma

Dopo aver scritto il programma, collegare prima il potenziometro a manopola e il servo alla scheda di espansione XIAO come mostrato nella figura sottostante. Quindi, collegare la scheda di controllo principale XIAO al computer con un cavo dati.



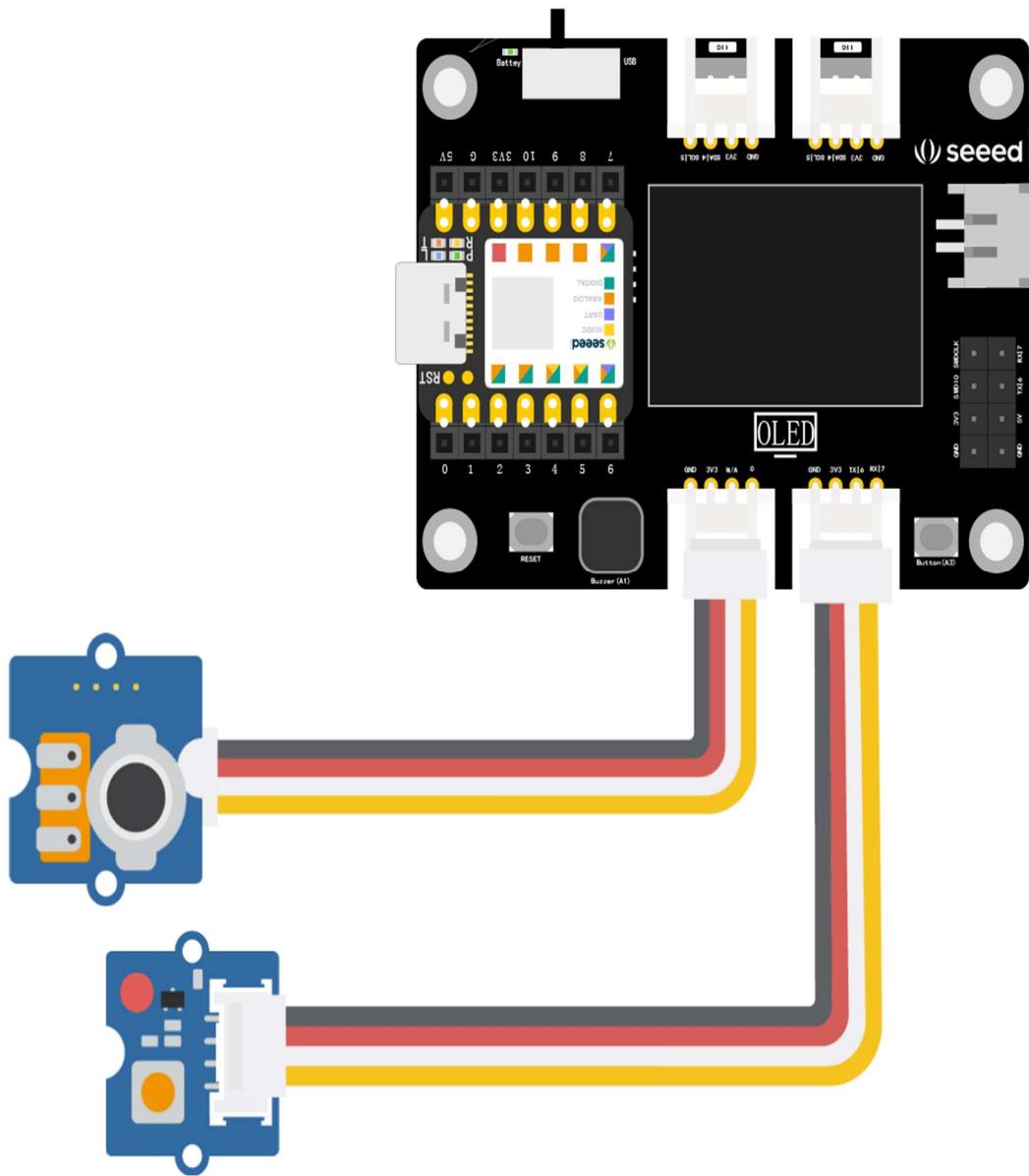
Dopo la connessione, cliccare su (the verify button) nell'IDE Arduino per verificare il programma. Se la verifica è priva di errori, cliccare su (upload button) per caricare il programma sull'hardware. Quando l'area di debug mostra "Done uploading.", si può aprire il monitor seriale, ruotare il potenziometro a manopola e osservare le modifiche nel valore dell'angolo e il movimento del servo. Cosa abbiamo scoperto?



⚠️ Nota: L'intervallo di rotazione del servo è 0°-180°, quindi si vedrà nel monitor seriale che quando il valore dell'angolo è maggiore di 180°, il servo smette di ruotare.

### 1.5.4 Esercizio Esteso

Abbiamo utilizzato il LED sulla scheda XIAO. Per utilizzare un LED esterno e controllarlo con un potenziometro a manopola per creare un effetto di luce “respiratoria”, cosa si deve fare? La scheda di espansione XIAO fornisce due interfacce Grove digitali-analogiche e c’è un’interfaccia A7/D7. Possiamo collegare il LED esterno a questa interfaccia, come mostrato nella figura:



Dopo la connessione, possiamo modificare leggermente il programma dal Task 1, cambiando `#define LEDPIN 13` in `#define LEDPIN 7`. Caricare il programma modificato e vedere se si riesce ad ottenere l’effetto desiderato.

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5\\_RotaryLed\\_ledmodule\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L5_RotaryLed_ledmodule_en)

## 1.6 Visualizzazione di “Hello World” su OLED

Nella nostra vita quotidiana, vediamo display ovunque: televisori, computer, telefoni, display per auto, cartelloni pubblicitari LCD nei centri commerciali... Senza una varietà di schermi, la

nostra vita perderebbe molto del suo divertimento. Naturalmente, questi schermi, oltre al tempo libero e all'intrattenimento, sono anche strumenti indispensabili per la vita quotidiana. I display comuni includono display LCD, display OLED, ecc. Hanno tutti i loro punti di forza e di debolezza come dispositivi di visualizzazione e possono essere applicati in diversi campi e scenari. La scheda di espansione XIAO integra un display OLED. In questa lezione, impareremo come utilizzare OLED per visualizzare testo, motivi e immagini.

## 1.6.1 Conoscenza di Base

### 1.6.1.1 Il Display OLED

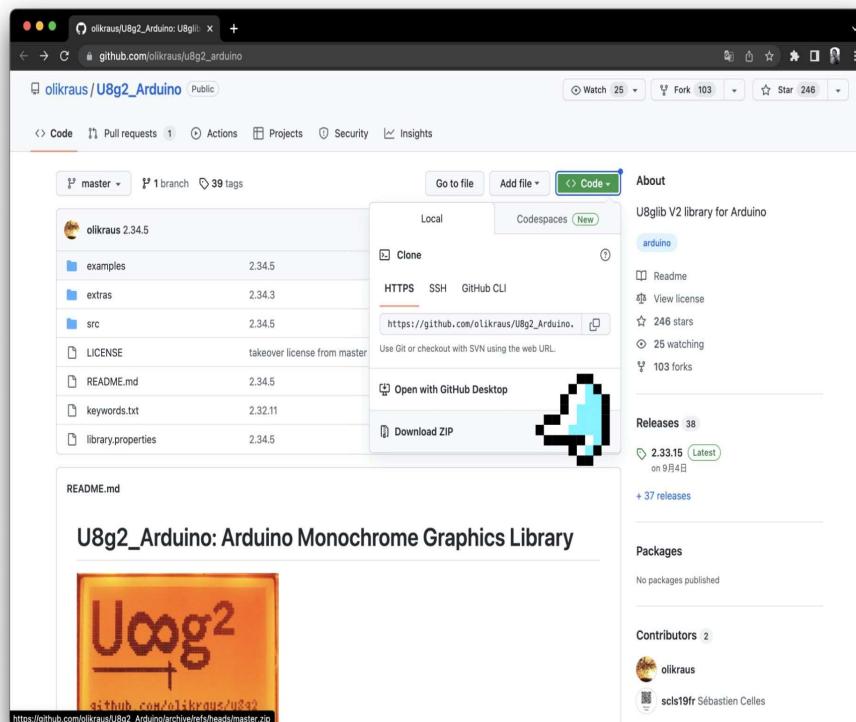
OLED, noto anche come “Organic Light Emitting Diode” [diodo organico a emissione di luce], ha vantaggi quali auto-luminosità, basso consumo energetico, velocità di risposta rapida, alta risoluzione, peso leggero, ecc. Il suo campo di applicazione è molto ampio. La scheda di espansione XIAO integra un display OLED da 0.96 pollici 128x64 pixel, utilizzabile direttamente senza cablaggio. Durante la produzione del progetto, possiamo visualizzare ora, temperatura e umidità e altri valori di ritorno del sensore tramite il display OLED e possiamo anche visualizzare direttamente lettere, numeri, grafici e persino motivi, ottenendo effetti visivi interattivi.

### 1.6.1.2 Come Scaricare e Installare la Libreria U8g2\_Arduino

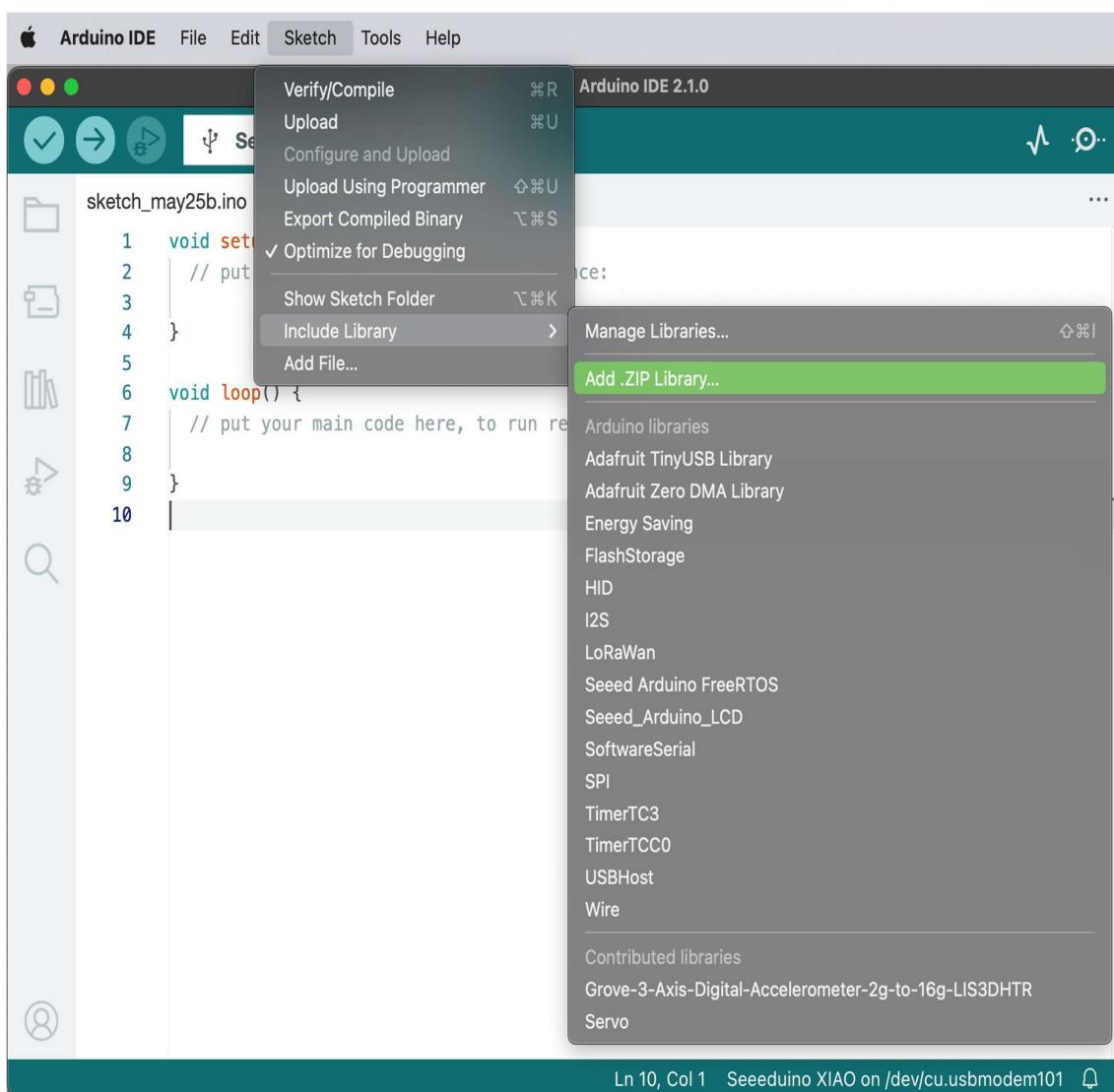
Una libreria è una raccolta di codici di programma che incapsula alcune funzioni comunemente utilizzate in un file che gli utenti possono richiamare. Quando utilizziamo display OLED, sensori di temperatura e umidità, ecc., dobbiamo utilizzare le librerie corrispondenti. Dove possono essere scaricate queste librerie e come installarle?

Spiegheremo utilizzando il file della libreria U8g2\_Arduino del display OLED come esempio.

Inserire il link del sito Web [https://github.com/olikraus/u8g2\\_arduino](https://github.com/olikraus/u8g2_arduino) per accedere alla pagina GitHub, cliccare su “Code→Download ZIP” per effettuare il download del pacchetto di risorse in locale, come mostrato nella figura seguente.



Una volta completato il download, aprire Arduino IDE, cliccare su “Sketch→Include Library→Add .ZIP Library” e seleziona il file ZIP appena scaricato.



Se la libreria è installata correttamente, si possono vedere le informazioni di richiesta per l'installazione corretta della libreria nella finestra di output.

### 1.6.1.3 Libreria U8g2 per OLED



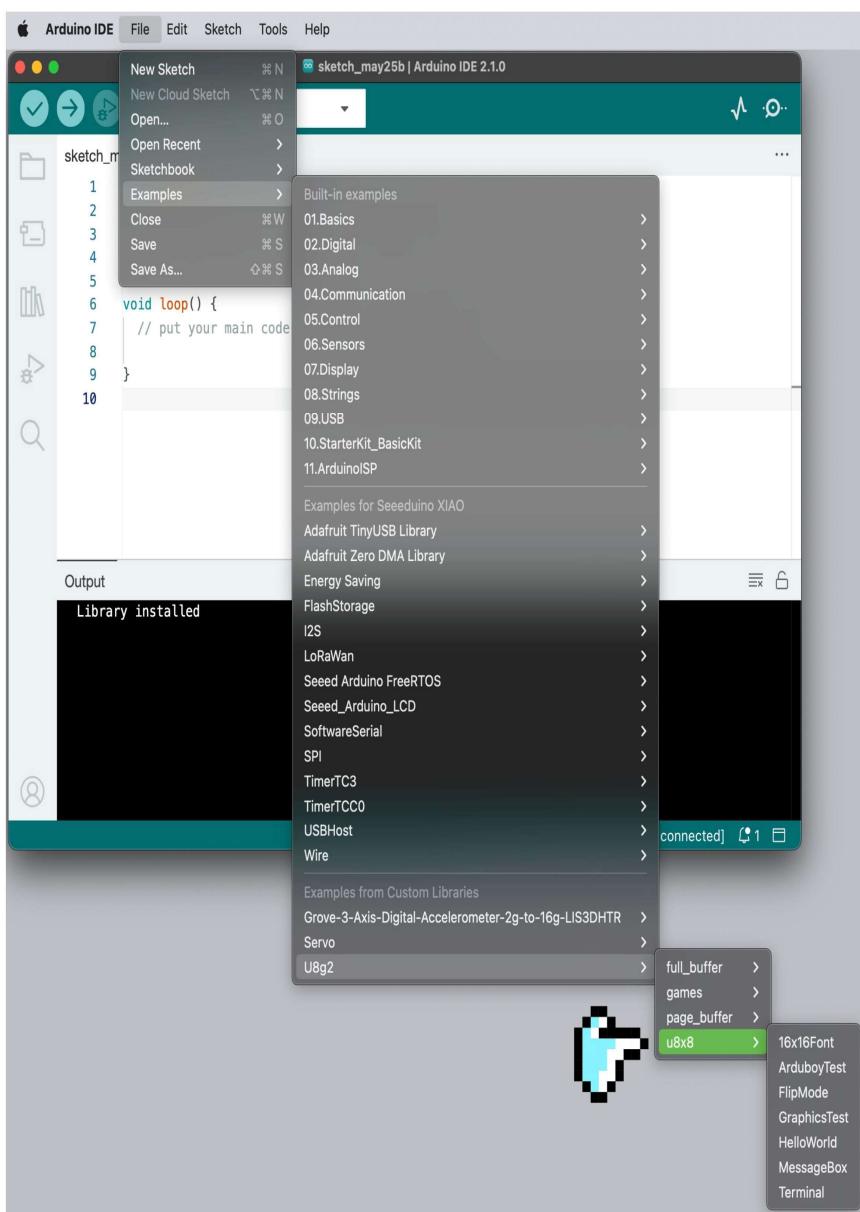
U8g2 è una libreria grafica monocromatica per dispositivi embedded, che supporta vari tipi di display OLED, semplificando la scrittura di programmi per ottenere gli effetti desiderati. La libreria U8g2 include anche la libreria U8x8 e le due librerie hanno funzioni diverse:

### ***U8g2***

Include tutte le procedure grafiche (disegno di linee/riquadri/cerchi); Supporta vari font, (quasi) nessuna restrizione sull'altezza del font; Per la visualizzazione è necessaria una certa quantità di memoria nel microcontrollore.

### ***U8x8***

Supporta solo l'output di testo (carattere); Consente a ogni carattere di utilizzare solo un font di dimensioni fisse (8x8 pixel); Scrive direttamente sul display, non è necessario alcun buffer nel microcontrollore. In parole poche, quando vogliamo che il display OLED visualizzi vari font, grafici, pattern e presenti contenuti visivi in modo più flessibile, possiamo utilizzare la libreria U8g2; quando vogliamo visualizzare i caratteri in modo più diretto, senza requisiti di font, solo per visualizzare valori di sensori, tempo, ecc., possiamo utilizzare la libreria U8x8, che è più efficiente. Possiamo trovare molti programmi di esempio in “File→Examples→U8g2” e familiarizzare con l'uso della libreria tramite tali programmi.



Successivamente, visualizzeremo i caratteri e disegneremo cerchi utilizzando rispettivamente le due librerie.

## 1.6.2 Attività 1: Visualizzare Hello World! sull’OLED della scheda di espansione XIAO

**⚠️** Nota Prima di iniziare a scrivere un programma per l’OLED della scheda di espansione XIAO, assicurarsi che l’IDE Arduino abbia caricato il file di libreria `U8g2_Arduino`. Il metodo di caricamento è consultabile nella descrizione nella sezione “Come Scaricare e Installare la Libreria Arduino” di questa lezione.

### Analisi

Per visualizzare solo “Hello World!” sull’OLED, si possono scrivere direttamente i caratteri con la libreria U8x8. I passaggi sono i seguenti:

- Dichiare il file di libreria, impostare il costruttore e questo definisce il tipo di display, il controller, la dimensione del buffer RAM e il protocollo di comunicazione.
- Inizializzare il display.
- Impostare il font del display, impostare la posizione di inizio della stampa e visualizzare “Hello World!”.

### \*Scrivere il programma\*\*

**Passaggio 1:** Dichiare il file di libreria, impostare il costruttore e il costruttore definisce il tipo di display, il controller, la dimensione del buffer RAM e il protocollo di comunicazione.

```
#include <Arduino.h>
#include <U8x8lib.h> // Usa il file di libreria U8x8
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
//Imposta il costruttore, definisci il tipo di display, il controller, la dimensione del buffer RAM
e il protocollo di comunicazione, generalmente determinati in base al modello di display utilizzato
```

**Passaggio 2:** Inizializzare il display. Dopo aver dichiarato il file di libreria nel passaggio precedente, si può usare le funzioni nella libreria per impostare il display OLED.

```
void setup(void) {
    u8x8.begin(); //Inizializza la libreria u8x8
    u8x8.setFlipMode(1); //Capovolge il display di 180 gradi, generalmente i numeri 0 e 1
}
```

**Passaggio 3:** Impostare il font del display (ci sono vari font tra cui scegliere nella libreria u8x8, possiamo fare riferimento a <https://github.com/olikraus/u8g2/wiki/fntlist8x8> per scegliere), impostare la posizione di partenza della stampa e visualizzare “Hello World!”.

```
void loop(void) {
    u8x8.setFont(u8x8_font_chroma48medium8_r); //Definisci il font u8x8
    u8x8.setCursor(0, 0); //Imposta la posizione del cursore di disegno
    u8x8.print("Hello World!"); //Disegna il contenuto su OLED: Hello World!
}
```

Il programma completo è il seguente:

```
#include <Arduino.h>
#include <U8x8lib.h> //Use U8x8 library file
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
//Set the constructor, define the display type, controller, RAM buffer size, and communication
protocol, generally determine according to the used display model

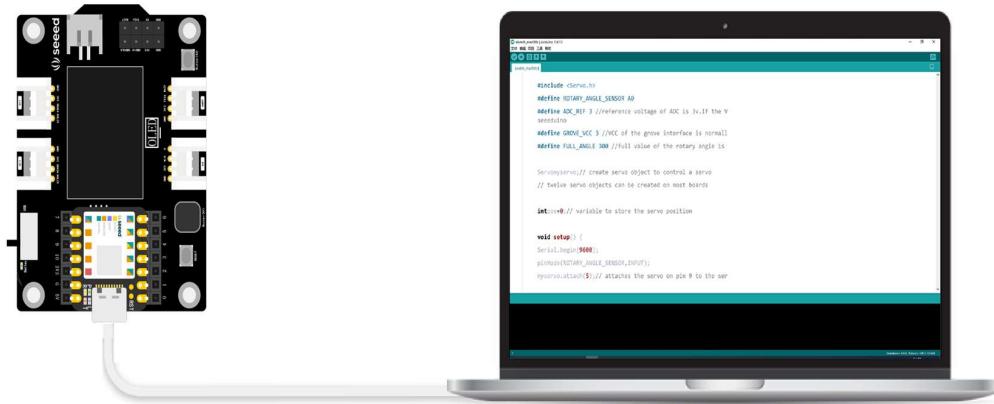
void setup(void) {
    u8x8.begin(); //Inizializza la libreria u8x8
    u8x8.setFlipMode(1); //Flip the display 180 degrees, generally numbers 0 and 1
}

void loop(void) {
    u8x8.setFont(u8x8_font_chroma48medium8_r); //Definisci il font u8x8
    u8x8.setCursor(0, 0); //Imposta la posizione del cursore di disegno
    u8x8.print("Hello World!"); //Disegna il contenuto su OLED: Hello World!
}
```

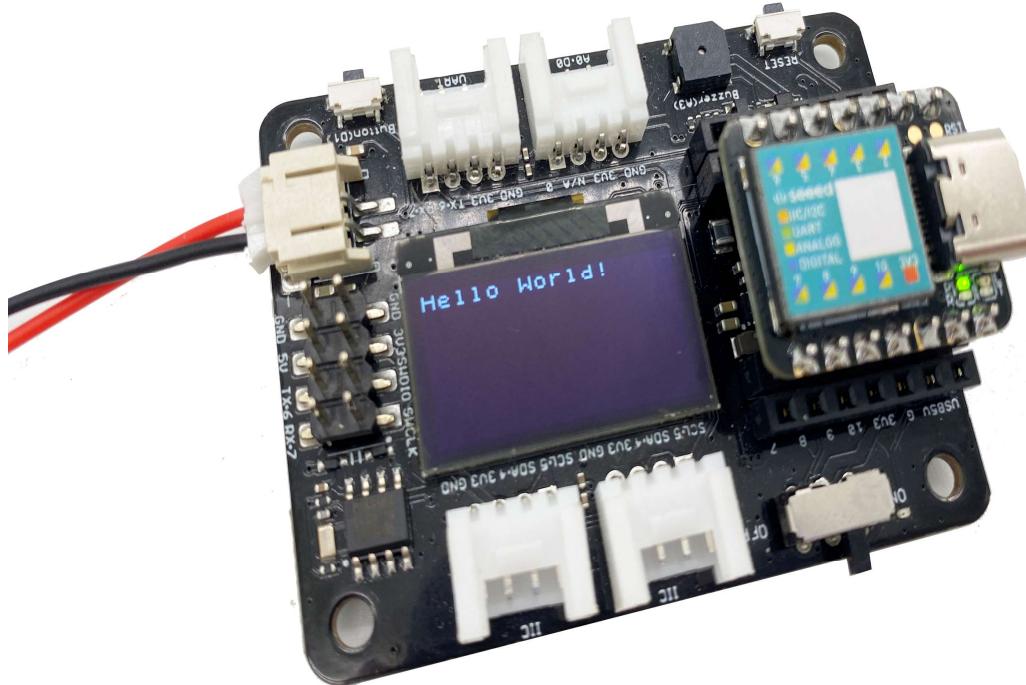
Prelevare questo programma da Github[https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L6\\_HelloWorld\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L6_HelloWorld_XIAO_en)

### Caricamento Programma:

Dopo aver scritto il programma, collegiamo la scheda di controllo principale XIAO all’interfaccia del computer tramite un cavo dati, come mostrato nell’immagine sottostante:



Cliccare su “Upload” per trasferire il programma alla scheda di controllo principale. Una volta completato il caricamento, controllare se il display OLED mostra “Hello World!”.



### 1.6.3 Attività 2: Disegnare un cerchio sul display OLED

#### **Analisi**

Per disegnare un cerchio sul display OLED, dobbiamo usare la libreria `ug92`. La programmazione prevede quattro passaggi:

- Dichiare il file della libreria `u8g2`, determinare se usare il protocollo SPI o I2C e impostare il costruttore per connettersi al display OLED.
- La funzione `draw()` usa la funzione `u8g2.drawCircle` per disegnare un cerchio sull'OLED.
- Inizializzare la libreria `u8g2`.
- Nella funzione `loop()`, chiamare le funzioni correlate per disegnare immagini sull'OLED.

### **Scrittura del Programma**

**Passaggio 1:** Dichiare il file della libreria `u8g2`, determinare se usare il protocollo SPI o I2C e impostare il costruttore per connettersi al display OLED.

```
#include<Arduino.h>
#include<U8g2lib.h> // Utilizza la libreria U8g2

// Determina se utilizzare il protocollo SPI o I2C
#ifndef U8X8_HAVE_HW_SPI
#include<SPI.h>
#endif
#ifndef U8X8_HAVE_HW_I2C
#include<Wire.h>
#endif

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);
// Imposta il costruttore, definisci il tipo di display, il controller, la dimensione del buffer RAM
e il protocollo di comunicazione
```

**Passaggio 2:** La funzione `draw()` usa la funzione `u8g2.drawCircle` per disegnare un cerchio sull'OLED. I parametri della funzione `u8g2.drawCircle(x0,y0,rad,opt)` sono i seguenti:

- `x0,y0`: La posizione del centro del cerchio.
- `rad`: Definisce la dimensione del cerchio, con un diametro di  $2 \times \text{rad} + 1$ .
- `opt`: Sceglie una parte o tutto il cerchio.

```
void draw(void) {
    u8g2.drawCircle(20, 25, 10, U8G2_DRAW_ALL); // Disegna un cerchio completo con un diametro di 21
alle coordinate (20, 25)
}
```

**Passaggio 3:** Inizializza la libreria `u8g2`.

```
void setup(void) {
    u8g2.begin(); // Inizializza la libreria
}
```

**Passaggio 4:** Nella funzione `loop()`, chiamare le funzioni correlate per disegnare le immagini sull'OLED. Utilizzare le funzioni `firstPage` e `nextPage` per scorrere il contenuto dell'immagine. Devono essere utilizzate insieme, come mostrato nel programma seguente:

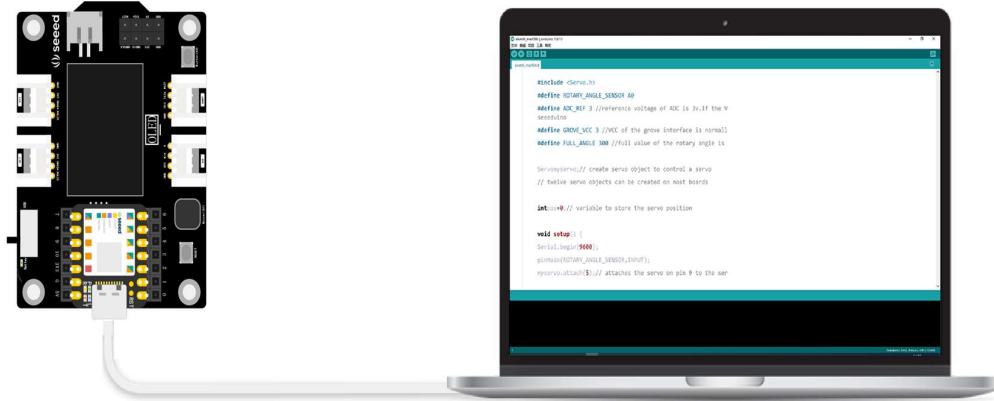
```
void loop(void) {
    // Scorsi la visualizzazione dell'immagine
    u8g2.firstPage();
    do {
        draw(); // Usa la funzione draw
    } while( u8g2.nextPage() );
}

delay(1000);
```

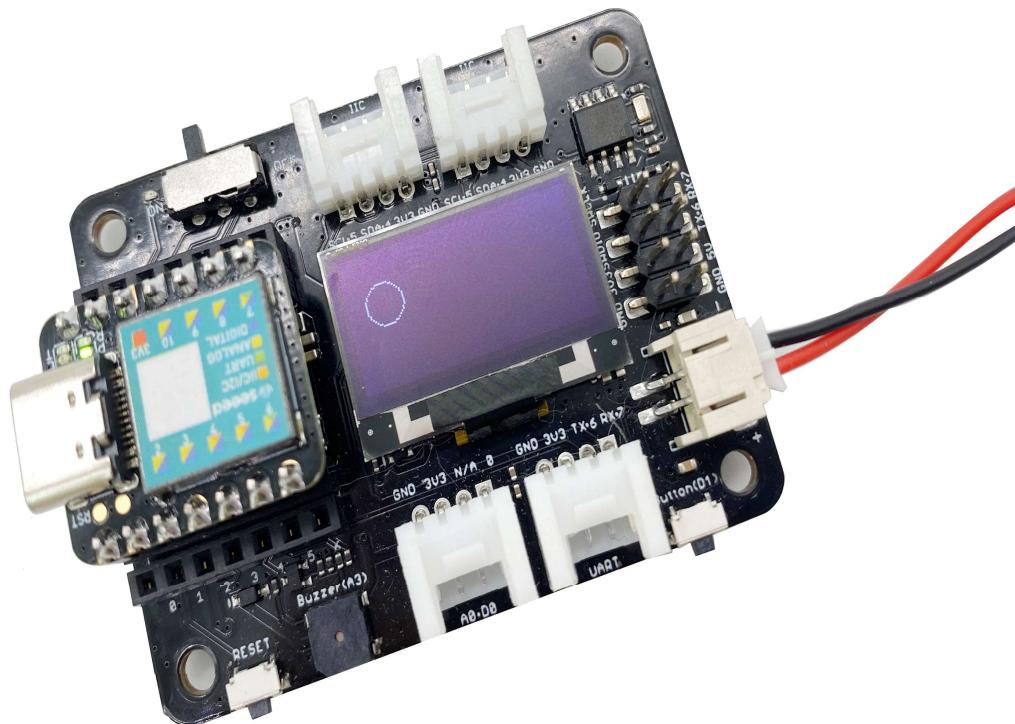
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L6\\_DrawCircle\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L6_DrawCircle_XIAO_en)

### **Caricamento del Programma**

Dopo aver scritto il programma, collegare la scheda di controllo principale XIAO al computer tramite un cavo dati, come mostrato di seguito:



Una volta connesso, cliccare sul pulsante “Upload” per caricare il programma sull’hardware. Quando l’area di debug visualizza “Upload Successful”, controllare se lo schermo OLED ha mostrato un motivo circolare.



#### 1.6.4 Esercizio Esteso

Provare a disegnare alcuni motivi più complessi.

## 2.1 Introduzione alla Progettazione di Prototipi di Prodotti

Nella prima unità, siamo entrati nel regno dell'hardware elettronico e della programmazione, imparando come controllare l'hardware elettronico tramite codice per ottenere gli effetti desiderati, come controllare un LED in vari modi, emettere un suono di cicalino, visualizzare testo su uno schermo OLED e altro ancora. Padroneggiare questa conoscenza ci aiuterà a trasformare le idee nella nostra mente in realtà. In questa sezione, impareremo il processo da un'idea a un prototipo e poi a un prodotto. Solo quando si possiederà questa conoscenza si potrà entrare nel mondo della progettazione di prototipi di prodotti. Se siete riusciti a seguire questo corso fino a questo punto, non c'è dubbio che siete dei "maker" nel profondo. L'idea di "voller fare qualcosa da solo" continua a turbinare nella mente. Questa sezione fornirà alcuni consigli su come diventare un maker e una guida su come creare progetti di prototipi di prodotti elettronici.

### 2.1.1 Coltivare la Mentalità del Maker

Diventare un maker eccellente non significa solo imparare moduli hardware e conoscenze di programmazione, ma anche coltivare consapevolmente alcune abitudini.



Dale Dougherty, fondatore della rivista "Make:", ha dato alcuni consigli su come coltivare la mentalità del maker nel suo discorso di benvenuto al Maker Camp (cfr. <https://makercamp.com/get-started>).

#### Mantieni un atteggiamento giocoso

Il gioco ci apre a idee creative e nuove esperienze. Mentre giochiamo, coinvolgiamo il nostro corpo e la nostra mente e spesso interagiamo con gli altri. Mentre giochiamo, imparare ci sembra naturale e possiamo correre dei rischi per fare cose che non sapevamo di poter fare.

#### Sii Curioso

Fai domande: chi, cosa, perché e come. Come vengono realizzate le cose intorno a te? Chi le realizza e dove vengono realizzate?

## Sii fisico

Usa i tuoi sensi per sperimentare il mondo fisico che ti circonda. Quali sono le differenze tra il mondo naturale e quello costruito?

## Trova uno strumento preferito

Esistono strumenti per tutti i tipi di applicazioni. Dato un ambito che ti interessa, come biciclette o musica, quali sono alcuni degli strumenti, sia fisici che digitali, che potresti voler imparare a usare? Scegli un nuovo strumento e condividerlo con noi.

## Fai qualcosa che non hai mai fatto prima

A volte decidiamo di non essere bravi in qualcosa e non proviamo mai a farla. Parte dello spirito del “DIY” [fai da te] è provare qualcosa che non hai mai provato prima, anche se non sei particolarmente bravo. Consideralo un esperimento. Vedi se ti piace. Prova a cucinare, fare giardinaggio o suonare uno strumento musicale. Oppure prova a riparare qualcosa che è rotto. Condividi questa nuova abilità.

## Crea qualcosa

Potresti progettare qualcosa che risolva un problema, potrebbe essere un problema per te o per gli altri. Potresti costruire qualcosa di interattivo, come un giocattolo, un’auto o un aeroplano giocattolo. I lanciatori di aeroplani di carta sono popolari, così come quelli dei razzi.

## Quali sono i vantaggi del coinvolgimento dei giovani nella creazione?

Ecco le cinque competenze chiave che abbiamo identificato come risultati per i giovani che partecipano a Start Making!

1. Identificarsi come creatore o maker. I giovani sviluppano atteggiamenti positivi verso la creazione di progetti pratici.
2. Sviluppano sicurezza nell’espressione creativa. I giovani si sentono in grado di dare vita alle proprie idee progettando, sperimentando, ripetendo e persistendo nonostante i fallimenti.
3. Acquisire competenza negli strumenti tecnici. I giovani acquisiscono familiarità con una varietà di strumenti e tecnologie che possono utilizzare per realizzare progetti.
4. Diventare consapevoli di STEAM. I giovani acquisiscono consapevolezza di idee e concetti che collegano Scienza, Tecnologia, Ingegneria (engineering), Arte e Matematica e dimostrano curiosità per saperne di più.
5. Imparare capacità di collaborazione e networking. I giovani si impegnano attivamente nella collaborazione e nell’aiutare gli altri.

**Iniziare a Creare! Una Guida per Coinvolgere i Giovani nelle Attività di Maker** Di Danielle Martin e Alisha Panjwani a cura di Natalie Rusk

## Quali sono gli attributi di un maker? Cos’è una mentalità da maker?

- I maker sono curiosi. Sono esploratori. Portano avanti progetti che personalmente trovano interessanti.
- I maker sono giocosi. Spesso lavorano a progetti che mostrano un senso di stravaganza.
- I maker sono disposti ad assumersi dei rischi. Non hanno paura di provare cose mai fatte prima.
- I maker si assumono delle responsabilità. Amano intraprendere progetti che possono aiutare gli altri.

- I maker sono persistenti. Non si arrendono facilmente.
- I maker sono pieni di risorse. Cercano materiali e ispirazione in luoghi improbabili.
- I maker condividono—le loro conoscenze, i loro strumenti e il loro supporto.
- I maker sono ottimisti. Credono di poter fare la differenza nel mondo.

**Making Makers: Kids, Tools and the Future of Innovation** di AnnMarie Thomas

## 2.1.2 Illuminazioni sulla Progettazione di Prototipi di Prodotto

Presentazione dell'Autrice:



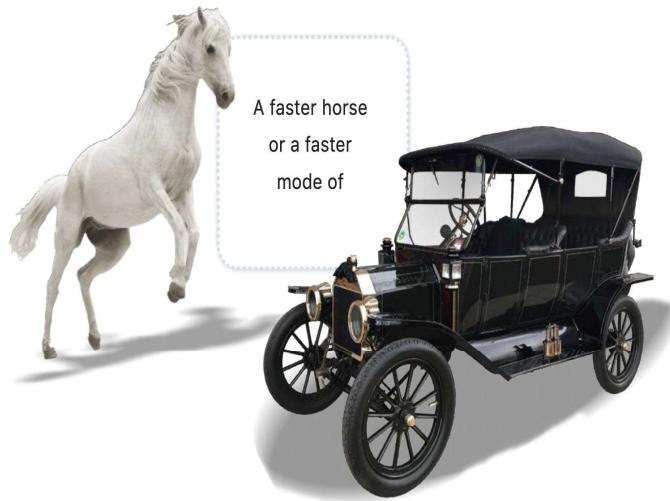
Wen Yanming, una donna post-90, si è laureata presso la Chinese University of Hong Kong e la South China University of Technology, con un master in giurisprudenza. È una product manager hardware, un'inventrice, un'imprenditrice, con oltre un decennio di esperienza nella pratica tecnologica e nella produzione.

### 2.1.2.1 Processo di Base della Progettazione di Prototipi di Prodotto

Dall'idea al prototipo di prodotto e poi al prodotto, questo è un processo che ogni prodotto deve attraversare. Un prototipo di prodotto ci consente di verificare rapidamente idee, funzionalità e fattibilità del prodotto in modo conveniente, fornendo la base per test, ottimizzazione e aggiornamenti iterativi. Dietro ogni prodotto di successo che vediamo, potrebbero esserci state innumerevoli iterazioni di prototipi di prodotto. Pertanto, creare un buon prototipo di prodotto è un processo essenziale e una solida base per un prodotto di successo. I prototipi necessari per diversi tipi di prodotti e diverse fasi del prodotto non sono gli stessi. Quando parliamo di un prototipo di prodotto, ci si può riferire a un prototipo concettuale, un prototipo funzionale, un prototipo di produzione in piccoli lotti, un modello di fabbrica, ecc. Va notato che per i prodotti hardware elettronici, la discussione qui riguarda principalmente i prototipi di prodotto per i concetti di prodotto e l'implementazione funzionale. In generale, la progettazione di un prototipo di prodotto funzionale include principalmente i seguenti processi:

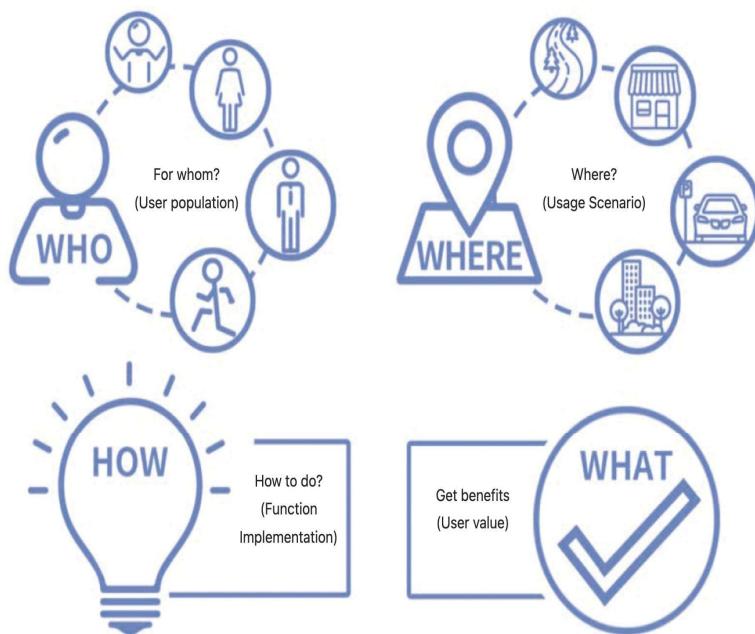
#### 1. Identificare e Chiarire il Problema da Risolvere

Einstein una volta disse: "Porre un problema è spesso più importante che risolverlo". Ogni prodotto deve esistere per risolvere un certo problema o per fornire un beneficio alle persone. Pertanto, identificare e chiarire il problema da risolvere è un prerequisito per chiarire le esigenze di progettazione del prodotto e procedere con la sua progettazione. È importante notare che solo perché abbiamo identificato un problema non significa che comprendiamo veramente e definiamo accuratamente questo problema. Ad esempio, più di 100 anni fa, quando Henry Ford, il fondatore della Ford Motor Company, andava in giro a chiedere ai clienti di che tipo di trasporto avessero bisogno, quasi tutti rispondevano: "Voglio un cavallo più veloce". Ma le persone hanno davvero bisogno solo di un cavallo più veloce? Se il signor Ford avesse definito il problema in base a questo, forse non avremmo avuto auto più veloci e comode così rapidamente.



## 2. Analisi della Domanda e Definizione del Prodotto

Una volta definito chiaramente il problema, possiamo estrarne le esigenze insoddisfatte. Come nell'esempio precedente, il problema a quel tempo era in realtà come arrivare a destinazione più velocemente, quindi l'esigenza corrispondente era "un mezzo di trasporto più veloce", non "un cavallo più veloce". Pertanto, dobbiamo essere bravi a scavare più a fondo nei problemi che scopriamo per trovare le esigenze reali. L'analisi della domanda richiede generalmente un'analisi della popolazione di utenti e degli scenari di utilizzo, da cui ricavare le funzioni necessarie per risolvere il problema, ovvero chiarire: per chi, in quali scenari, per ottenere quali funzioni, per ottenere quali benefici.

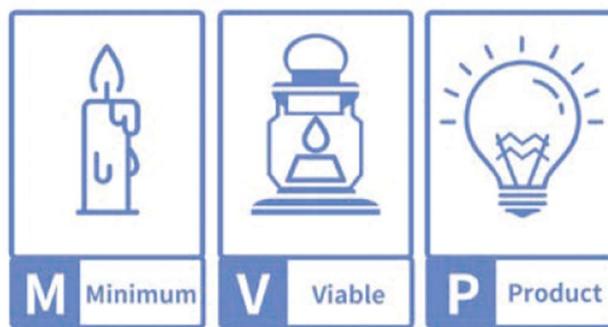


Esistono molti tipi di esigenze: vere esigenze degli utenti, esigenze superficiali, esigenze urgenti, esigenze ordinarie, esigenze ad alta frequenza, esigenze a bassa frequenza e così via. Tutte queste devono essere analizzate alla luce della situazione reale, che può quindi formare la corretta definizione del prodotto in base a queste esigenze. Ogni prodotto alla fine deve essere commercializzato per realizzare il suo massimo valore. Pertanto, quando si

progetta un prodotto per il mercato, dobbiamo anche condurre una serie di analisi di mercato, tra cui dimensioni del mercato, aspettative di vendita, analisi degli utili, periodo di ammortamento, analisi del rapporto input-output e così via.

### **3. Selezione e Assemblaggio dell'Hardware**

Per la progettazione di prodotti elettronici, una volta definite le esigenze, dobbiamo trovare l'hardware in grado di implementare queste esigenze funzionali. Quando si sceglie l'hardware, gli elementi che generalmente devono essere considerati includono: fattibilità, livello di soddisfazione delle esigenze, costo, volume, peso, prestazioni, durata, aspetto, ecc. Una delle capacità più importanti di un eccellente progettista di prodotti è quella di tenere conto di vari fattori in base alla definizione e alle esigenze del prodotto, bilanciare questi fattori e fare dei compromessi. Spesso, non esiste una risposta corretta. In generale, quando costruiamo un prototipo, la prima cosa che dovremmo considerare è la creazione di un “Minimum Viable Product (MVP)” [prodotto minimo praticabile]. La sua funzione è quella di utilizzare il minor numero di risorse per verificare rapidamente il prodotto e migliorare e iterare rapidamente.



### **4. Sviluppo Software e Implementazione Funzionale**

Molti ingegneri di sviluppo software esperti disegneranno un diagramma di flusso di implementazione funzionale prima dello sviluppo software. Disegneranno un diagramma di flusso di implementazione funzionale in base alle funzioni da implementare. Questo può aiutare a chiarire il pensiero progettuale del software, controllare la logica della funzione, facilitare l'identificazione di perdite e carenze e farvi riferimento in qualsiasi momento durante la programmazione, assicurandosi di avere una chiara comprensione. Pertanto, indipendentemente dalla complessità dello sviluppo della funzione software, si raccomanda a tutti di sviluppare una buona abitudine di disegnare prima un diagramma di flusso di implementazione funzionale. Può essere un semplice schizzo disegnato a mano o si può utilizzare un software professionale, come Visio o Axure, per disegnarlo.

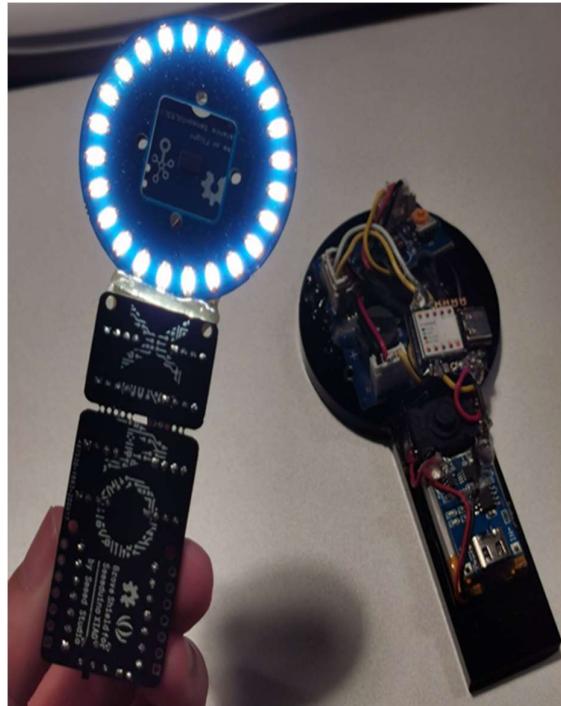
Quando si sviluppa un software, cercare di essere efficienti e concisi. Sfruttare appieno i vantaggi della comunità open source e imparare a utilizzare le risorse hardware e software esistenti in modo più efficace. Ad esempio, molti componenti hardware o applicazioni hanno già molte librerie e routine open source pronte all'uso. Durante lo sviluppo, è possibile fare riferimento a queste, rispettare i corrispondenti accordi open source per utilizzare le risorse correlate ed evitare di perdere tempo a reinventare la ruota.

### **5. Test e Ottimizzazione del Prototipo**

Dopo aver realizzato il prototipo, dobbiamo testarlo per verificarne l'implementazione funzionale e se soddisfa le esigenze di progettazione originali. Questo processo dovrebbe coinvolgere quanti più utenti target possibile per raccogliere il loro feedback. In questo modo, possiamo scoprire meglio i difetti nel prototipo del prodotto, apportare misure correttive e

miglioramenti, aggiornare e iterare il design e infine creare uno schema di progettazione che soddisfi le esigenze dell'utente, gettando solide basi per la progettazione formale del prodotto.

### 2.1.2.2 Pratica di Prototipo di Prodotto - Prototipo “Allarme a Distanza di un Metro”



Successivamente, prendiamo come esempio il processo di produzione del prototipo di un “Allarme a Distanza di un Metro” per sperimentare il processo di progettazione del prototipo di prodotto.

#### 1. Identificazione e Definizione del Problema da Risolvere

All'inizio del 2020, la pandemia di COVID-19 è scoppiata a livello globale, la situazione era molto grave. Per impedire al virus di diffondersi attraverso goccioline e contatti aerei ravvicinati, i governi e i dipartimenti sanitari di tutto il mondo hanno esortato tutti a ridurre gli assembramenti e a mantenere il distanziamento sociale di almeno un metro ogni volta che era possibile. Tuttavia, non è facile per tutti ricordarlo costantemente e mantenere una distanza sociale accurata di oltre un metro. I bambini, spesso dimenticano di mantenere la distanza perché stanno giocando felici o non hanno idea di quanto si dovrebbero distanziare. Quando si esce, ci sono anche degli sconosciuti che, a causa della mancanza di consapevolezza della prevenzione delle epidemie, si avvicinano inconsciamente a noi e dobbiamo trovare un modo educato per ricordarglielo. Pertanto, abbiamo tratto una domanda dalla vita: **Come possiamo ricordare costantemente alle persone di mantenere una distanza sociale di oltre un metro?**

#### 2. Analisi delle Esigenze e Definizione del Prodotto

Col problema definito, analizziamo le esigenze principali che questo problema innesca: un dispositivo di promemoria per la prevenzione delle epidemie per uso pubblico che invia un promemoria quando altri entrano entro un metro, incoraggiando così tutti a mantenere consapevolmente una distanza sociale di un metro. Pensando ulteriormente alle esigenze principali, che tipo di promemoria dovrebbe essere? Possiamo pensare ai prodotti elettronici che utilizziamo di solito, che tipo di promemoria hanno? Non sono altro che suoni, luci, vibrazioni, messaggi di testo sullo schermo, ecc. Considerando che il promemoria deve essere tempestivo, diretto e ovvio, non è facile vedere chiaramente lo schermo a una

distanza di circa un metro, il volume sarà maggiore e il costo più elevato dopo aver aggiunto uno schermo, quindi non prendiamo in considerazione l'aggiunta di uno schermo. Le opzioni rimanenti sono suono, luce e vibrazione. Possiamo continuare a bilanciare e scegliere il metodo di promemoria necessario in base al costo, al volume e all'aspetto. Non esiste una risposta univoca qui. Quindi, in base al processo di analisi delle esigenze, definiamo provvisoriamente il prodotto come: un dispositivo che emette luce e vibra per ricordare quando rileva qualcuno che entra entro una distanza di un metro.

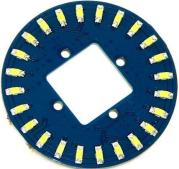
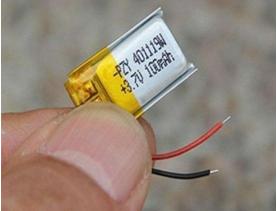
### 3. Selezione e Assemblaggio dell'Hardware

Con il prodotto definito, possiamo scomporre i requisiti funzionali principali:

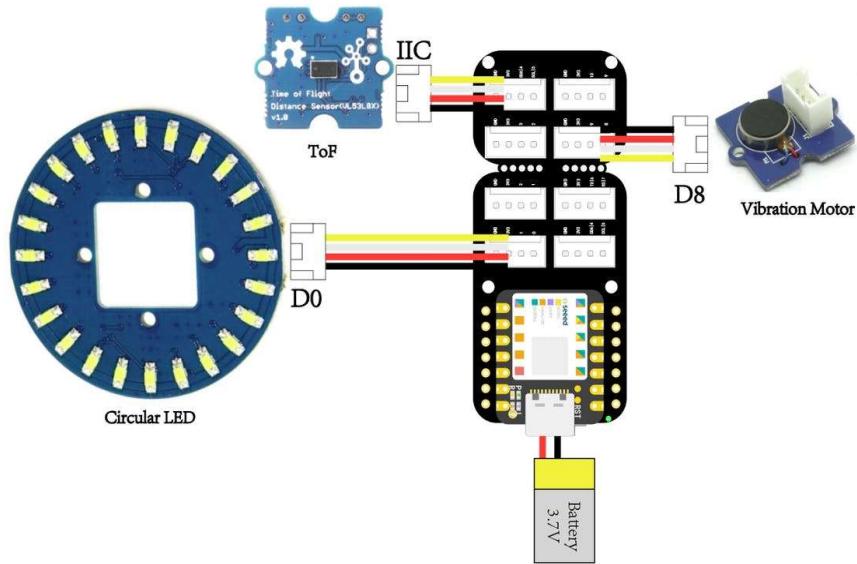
- (1) Rilevare quando una persona si avvicina entro una distanza di un metro
- (2) Allertare se stessi e gli altri
- (3) Dimensioni ridotte, facile da trasportare

Quindi, che tipo di hardware dovrebbe essere utilizzato per implementarli rispettivamente? Nel processo di implementazione del prototipo del prodotto, di solito sceglieremo hardware open source a basso costo, informazioni complete e molte routine per implementare le funzioni hardware. Dopo aver considerato in modo esaustivo il costo, la realizzazione delle funzioni, la difficoltà di assemblaggio, il volume, le risorse di sviluppo software e altri elementi, ho scelto il seguente hardware:

| Requisiti Funzionali              | Prodotto Hardware                                                                                                                | Presentazione della Funzione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scheda Madre</b>               |  Seeeduino XIAO (SAMD21)                       | Questa è una mini-scheda-madre di controllo sviluppata da Seeed Technology basata su SAMD21. Il volume è molto piccolo, solo 20x17.5 mm, le dimensioni di un pollice, l'interfaccia è ricca, le prestazioni sono elevate, molto adatta allo sviluppo di vari dispositivi in poche quantità.                                                                                                                                                                                                                                                                                                                             |
| <b>Scheda di Espansione</b>       |  Seeed Studio Grove Base for XIAO             | Grove Shield per Seeed Studio XIAO è una scheda di espansione Grove plug-and-play per la serie Seeed Studio XIAO. Con il chip di gestione della batteria integrato e il pad di collegamento della batteria, si può facilmente alimentare il Seeed Studio XIAO con una batteria al litio e ricaricarlo. 8 connettori Grove integrati includono due Grove I2C e una UART. Agisce come un ponte per Seeed Studio XIAO e il sistema Grove di Seeed. Il pad di collegamento Flash SPI consente di aggiungere Flash a Seeed Studio XIAO per espandere il suo spazio di memoria, offrendo a Seeed Studio XIAO più possibilità. |
| <b>Rilevamento della Distanza</b> |  Grove - Time of Flight Distance Sensor (ToF) | Esistono molti sensori per rilevare la distanza, la maggior parte dei quali misura tramite ultrasuoni, infrarossi, laser, ecc. Tra questi, il sensore di distanza Time of Flight Grove è una nuova generazione di modulo laser ToF basato su VL53L0X, in grado di fornire una misura accurata della distanza.                                                                                                                                                                                                                                                                                                           |

| Requisiti Funzionali        | Prodotto Hardware                                                                                                                                                                                    | Presentazione della Funzione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Allarme Luminoso</b>     | <br>Grove - Circular LED                                                                                            | <p>fino a 2 metri. Le dimensioni ridotte e l'elevata precisione di questo modulo lo hanno reso la prima scelta.</p> <p>Un Grove - Circular LED con un cerchio di LED può accendere una luce bianca. È esteticamente gradevole e fornisce un promemoria luminoso più grande e più evidente rispetto a un singolo LED.</p>                                                                                                                                                                                                      |
| <b>Allarme a Vibrazione</b> | <br>Grove - Vibration Motor                                                                                         | <p>Un modulo Grove con motore integrato per la vibrazione . Può essere utilizzato plug-and-play ed è comodo per generare promemoria di vibrazione continui o intermittenti controllando il segnale digitale.</p>                                                                                                                                                                                                                                                                                                              |
| <b>Alimentazione</b>        | <br>型号 : 401119 厚度 : 4mm<br>电压 : 3.7v 宽度 : 11mm<br>容量 : 100毫安 长度 : 20mm<br><br>Batteria al litio da 3.7V (401119) | <p>Una batteria al litio da 3.7 V di piccole dimensioni comunemente utilizzata per l'alimentazione delle cuffie Bluetooth. Il modello è il 401119, che rappresenta lo spessore, la larghezza e la lunghezza della batteria rispettivamente come 4 mm, 11 mm e 19 mm. Dopo aver saldato questa piccola batteria al litio al relativo pad sulla scheda di espansione Grove, può essere posizionata direttamente nello spazio tra Seeeduino XIAO e la scheda di espansione Grove, rendendo il prodotto più ordinato e bello.</p> |
| <b>Cablaggio</b>            | <br>Cavo di collegamento universale Grove (5cm)                                                                   | <p>Il connettore universale Grove è un connettore standard per il sistema Grove. Può essere utilizzato comodamente plug-and-play, senza saldature e tenendo conto della sequenza delle linee. La piattina Grove collega vari sensori e attuatori alla scheda di espansione, rendendo la costruzione del progetto semplice come i blocchi di costruzione e risparmiando molto tempo. La piattina corta da 5 cm è molto adatta per prototipi di prodotti compatti in termini di spazio.</p>                                     |

La connessione del modulo è la seguente, come illustrato nell'immagine:



I moduli hardware scelti hanno un ottimo design strutturale, utilizzabile direttamente per costruire il fattore di forma dell'allarme di distanza, risparmiando tempo nella realizzazione di un guscio. Pertanto, il metodo di produzione è abbastanza semplice: tutto ciò che serve è collegare ogni pezzo di hardware all'interfaccia appropriata, disporre le rispettive posizioni e quindi unirli insieme con adesivo hot melt [colla a caldo]. Questo completa rapidamente la connessione hardware e la creazione del fattore di forma di un allarme a distanza di un metro. Il prodotto hardware completato è il seguente:

#### 4. Sviluppo Software e Implementazione delle Funzioni

Prima di scrivere ufficialmente il programma, ho pianificato le funzioni e la logica che il software deve implementare e ho disegnato il seguente diagramma di flusso di implementazione funzionale utilizzando Visio:

Poiché Seeeduino XIAO supporta Arduino IDE, ho scelto di programmare nell'IDE Arduino. La maggior parte dell'hardware fornito da Seed Technology è open source e offre un eccellente supporto di documentazione per i propri prodotti. Pertanto, durante il processo di programmazione, ho trovato il corrispondente Wiki hardware open source sul [sito web ufficiale di Seeedstudio](#), ho scaricato i file di libreria pertinenti (nota: i file di libreria sono una raccolta di funzionalità specifiche fornite dagli sviluppatori che possono essere utilizzate semplicemente chiamandole, senza dover riscrivere il codice) e ho fatto riferimento alle routine di esempio dei moduli utilizzati. Ho completato il programma rapidamente.

Dopo che il programma è stato scritto e compilato correttamente, ho collegato Seeeduino XIAO al computer tramite una connessione Type-C e ho scaricato il codice scritto su Seeeduino XIAO tramite l'IDE Arduino. Una volta caricato correttamente il codice, il prototipo è stato completato.

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** L7\_tof\_XIAO\_en | Arduino IDE 2.1.0
- Board Selector:** Seeeduino XIAO
- Sketch Content (L7\_tof\_XIAO\_en.ino):**

```
1 #include <Grove_LED_Bar.h>
2 #include "Seeed_vl53l0x.h"
3
4 const int Buzzer = 8;//Vibration motor connected to D8
5 Grove_LED_Bar bar(0, 1, 0, LED_CIRCULAR_24); //Grove-LED ring connected to D0
6 Seeed_vl53l0x VL53L0X; //Grove-tof distance sensor connected to IIC (D4/D5)
7
8 #if defined(ARDUINO_SAMD_VARIANT_COMPLIANCE) && defined(SerialUSB)
9 #define SERIAL SerialUSB
10#else
11#define SERIAL Serial
12#endif
13
14
15 void setup() {
16    bar.begin();
17}
```

- Output Window:**

```
/Users/mouseart/Library/Arduino15/packages/Seeeduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-objcopy -O ihex -R .text -R .rodata -R .data
open /Users/mouseart/Library/Arduino15/packages/Seeeduino/hardware/samd/1.8.4/bootloaders/XIAOM0/bootloader-XIAO_M0-v3.7.0-...
```

Using library Grove LED Bar at version 1.0.0 in folder: /Users/mouseart/Documents/Arduino/libraries/Grove\_LED\_Bar
Using library Grove Ranging sensor - VL53L0X at version 1.1.1 in folder: /Users/mouseart/Documents/Arduino/libraries/Grove\_VL53L0X
Using library Wire at version 1.0 in folder: /Users/mouseart/Library/Arduino15/packages/Seeeduino/hardware/samd/1.8.4/libraries/Wire
/Users/mouseart/Library/Arduino15/packages/Seeeduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-size -A /private/var/folders/...

Sketch uses 48380 bytes (18%) of program storage space. Maximum is 262144 bytes.

## **5. Test e Ottimizzazione del Prototipo**



Dopo aver completato il prototipo, era il momento del test. Innanzitutto, dovevo testare se il prototipo implementava la funzionalità di base, ovvero se suonava e accendeva un allarme quando veniva rilevata una persona entro un raggio di un metro. Quindi, dovevo usarlo in uno scenario reale per vedere se l'esperienza utente era sufficientemente buona. Se riusciva a soddisfare in modo soddisfacente i requisiti e la definizione del prodotto, il prototipo del prodotto poteva essere considerato un successo e si poteva dare inizio alla fase successiva nello sviluppo del prodotto. Naturalmente, se venivano riscontrati problemi durante il test, erano necessari aggiustamenti e miglioramenti, seguiti da nuovi test. Questo processo viene

ripetuto finché il prototipo del prodotto non soddisfa i requisiti e viene determinato lo schema finale.

Completare il prototipo è solo il primo passo per realizzare un prodotto di successo. La nascita di ogni prodotto richiede molto impegno, continui tentativi ed errori e aggiustamenti per ottenere i migliori risultati. Il successo finale di un prodotto, oltre a soddisfare le esigenze dell'utente, deve anche superare molti test di mercato. Ciò richiede agli studenti, quando iniziano a imparare a realizzare prodotti, di mantenere sempre lo spirito di un artigiano, mantenendo al contempo un acuto senso del mercato e imparando conoscenze che vanno oltre il prodotto stesso. C'è ancora molta strada da fare e spero che tutti possano attenersi alle proprie intenzioni originali, continuare a esplorare e, in definitiva, realizzare prodotti di successo.

Il codice sorgente del programma è il seguente:

```
#include <Grove_LED_Bar.h>
#include "Seeed_vl53l0x.h"

const int Buzzer = 8; //Vibration motor connected to D8
Grove_LED_Bar bar(0, 1, 0, LED_CIRCULAR_24); //Grove-LED ring connected to D0
Seeed_vl53l0x VL53L0X; //Grove-tof distance sensor connected to IIC (D4/D5)

#if defined(ARDUINO_SAMD_VARIANT_COMPLIANCE) && defined(SerialUSB)
#define SERIAL SerialUSB
#else
#define SERIAL Serial
#endif

void setup() {
    bar.begin();

    pinMode(Buzzer, OUTPUT);
    digitalWrite(Buzzer, LOW); // turn the Buzzer on (HIGH is the voltage level)
    // Turn off all LEDs
    bar.setBits(0x0);

    VL53L0X_Error Status = VL53L0X_ERROR_NONE;
    SERIAL.begin(115200);
    Status = VL53L0X.VL53L0X_common_init();
    if (VL53L0X_ERROR_NONE != Status) {
        SERIAL.println("Starting VL53L0X measurement failed!");
        VL53L0X.print_pal_error(Status);
        while (1);
    }
}

VL53L0X.VL53L0X_long_distance_ranging_init();

if (VL53L0X_ERROR_NONE != Status) {
    SERIAL.println("Starting VL53L0X measurement failed!");
    VL53L0X.print_pal_error(Status);
    while (1);
}

void loop() {

    VL53L0X_RangingMeasurementData_t RangingMeasurementData;
    VL53L0X_Error Status = VL53L0X_ERROR_NONE;

    memset(&RangingMeasurementData, 0, sizeof(VL53L0X_RangingMeasurementData_t));
    Status = VL53L0X.PerformSingleRangingMeasurement(&RangingMeasurementData);
    if (VL53L0X_ERROR_NONE == Status) {
        if (RangingMeasurementData.RangeMilliMeter >= 2000) {
            SERIAL.println("Out of range!!!");
            digitalWrite(Buzzer, LOW); // turn the Buzzer off (LOW is the voltage level)

            // Turn off all LEDs
            bar.setBits(0x0);
        }
    }
    else if (RangingMeasurementData.RangeMilliMeter <= 1000) {
}
```

```

        digitalWrite(Buzzer, HIGH); // turn the Buzzer on (HIGH is the voltage level)
        // Turn on all LEDs
        bar.setBits(0b111111111111111111111111);

        SERIAL.print("Distance:");
        SERIAL.print(RangingMeasurementData.RangeMilliMeter);
        SERIAL.println(" mm");
    }

    else {
        digitalWrite(Buzzer, LOW); // turn the Buzzer off (LOW is the voltage level)

        // Turn off all LEDs
        bar.setBits(0x0);

        SERIAL.print("Distance:");
        SERIAL.print(RangingMeasurementData.RangeMilliMeter);
        SERIAL.println(" mm");
    }

}

else {
    SERIAL.print("Measurement failed!! Status code =");
    SERIAL.println(Status);
    digitalWrite(Buzzer, LOW); // turn the Buzzer off (LOW is the voltage level)

    // Turn off all LEDs
    bar.setBits(0x0);
}

delay(250);
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L7\\_tof\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L7_tof_XIAO_en)

## 2.2 Igrometro e Termometro Intelligenti

Termometri e igrometri sono onnipresenti nella vita quotidiana, fornendo misure in tempo reale della temperatura e dell'umidità nel nostro ambiente. Li usiamo spesso per misurare la temperatura corporea quando ci sentiamo febbricitanti o malati. L'invenzione di questi dispositivi ha portato un'enorme comodità nelle nostre vite. Nonostante le loro dimensioni, questi dispositivi contengono molta scienza. In questa sezione, creeremo un igrometro e un termometro intelligenti utilizzando un sensore di temperatura e umidità. Cos'è un sensore di temperatura e umidità e cosa può fare?

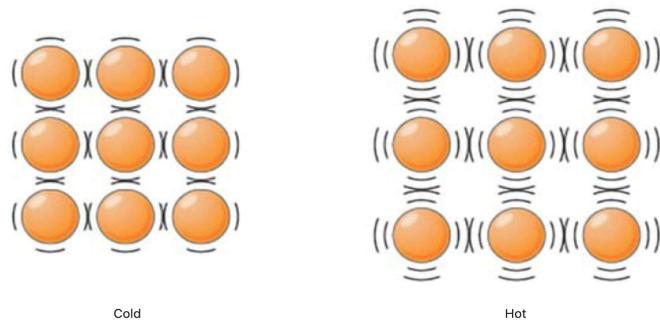
### 2.2.1 Conoscenza di Base

#### 2.2.1.1 Temperatura

La temperatura è strettamente legata alla nostra vita quotidiana; ci informa su quali vestiti dobbiamo indossare prima di uscire e assicura che il cibo o le bevande che consumiamo non siano troppo caldi o troppo freddi. Quando si esce di casa, si può percepire il freddo o il caldo, ma per quantificare esattamente quanto è freddo o caldo, usiamo la "temperatura".

La temperatura è una grandezza fisica che indica il grado di freddo o caldo di un oggetto. La temperatura alta e bassa di un oggetto è un fenomeno macroscopico che riflette l'intensità del moto termico delle molecole che compongono l'oggetto a livello microscopico. Quindi, la temperatura è una manifestazione dell'intensità del moto termico di un gran numero di molecole che costituiscono un oggetto. Più veloce è il moto molecolare, più alta è la temperatura e più caldo è l'oggetto; più lento è il moto molecolare, più bassa è la temperatura e più freddo è l'oggetto.

Per misurare la temperatura in modo accurato, dobbiamo stabilire uno standard di unità di temperatura e progettare gli strumenti di misura della temperatura corrispondenti.



### Scala della Temperatura

L'unità standard per la temperatura è nota come scala di temperatura. Nel corso dello sviluppo della scienza, sono state ideate diverse scale di temperatura, ma la loro metodologia di base è la stessa: stabilendo i valori di temperatura di determinati fenomeni o cose, tutte le altre temperature possono essere calibrate. Le scale di temperatura comuni includono Fahrenheit, Celsius e Kelvin. Solo cinque paesi, tra cui gli Stati Uniti e alcuni altri paesi di lingua inglese, usano ancora la scala Fahrenheit. La stragrande maggioranza del mondo, tra cui la Cina, usa la scala Celsius. Nei campi di ricerca, gli scienziati preferiscono usare la scala Kelvin.

- Nella scala Fahrenheit, a pressione atmosferica standard, la temperatura alla quale l'acqua inizia a congelare è fissata a 32 gradi Fahrenheit e la temperatura alla quale l'acqua bolle è di 212 gradi Fahrenheit. La scala è divisa in 180 parti uguali tra questi due punti, con ogni parte che rappresenta un grado Fahrenheit, indicato come 1°F. Nella scala Fahrenheit, la temperatura corporea umana normale è di circa 98°F.
- Nella scala Celsius, a pressione atmosferica standard, la temperatura alla quale l'acqua inizia a congelare è fissata a 0 gradi Celsius e la temperatura alla quale l'acqua bolle è fissata a 100 gradi Celsius. La scala è divisa in 100 parti uguali tra questi due punti, con ogni parte che rappresenta un grado Celsius, indicato come 1°C. Nella scala Celsius, la normale temperatura corporea umana è di circa 36.5°C.
- La scala Kelvin è stabilita sulla base dello zero assoluto. Gli scienziati hanno scoperto che esiste una temperatura minima nell'universo, -273,15°C, che non può essere raggiunta ma solo avvicinata asintoticamente. Questa temperatura minima è stata designata come zero assoluto e fissata a 0 Kelvin, indicata come 0K. La temperatura alla quale l'acqua inizia a congelare a pressione atmosferica standard è fissata a 273,15K e la temperatura alla quale l'acqua bolle è fissata a 373,15K. Nella scala Kelvin, la temperatura corporea normale di un essere umano è di circa 309.7 K.

#### 2.2.1.2 Termometro

Un termometro è uno strumento per misurare la temperatura. Poiché la temperatura non è una grandezza fisica che può essere vista direttamente, la misura della temperatura richiede l'assistenza di fenomeni fisici direttamente correlati alla temperatura. Ad esempio, nell'antica Cina, c'è una registrazione di 'fiamma blu pura e lucente', che viene misurata osservando il colore della fiamma.

Un altro esempio è il termometro a infrarossi, come mostrato nell'immagine a destra, che misura la temperatura attraverso le differenze di radiazione di oggetti a diverse temperature. Anche gli esseri umani, come altri organismi, irradiano energia infrarossa attorno a sé. Questa energia ha in genere una lunghezza d'onda di 9-13 $\mu\text{m}$  e rientra nella banda del vicino infrarosso di 0.76-100 $\mu\text{m}$ . Poiché la luce all'interno di questo intervallo di lunghezza d'onda non viene assorbita dall'aria, la temperatura superficiale del corpo umano può essere misurata con precisione semplicemente misurando l'energia infrarossa irradiata dal corpo

umano. Il sensore di temperatura a infrarossi del corpo umano è progettato e prodotto in base a questo principio.

Inoltre, il fenomeno dell'espansione e della contrazione termica è spesso utilizzato nella misura della temperatura. I termometri e i termometri corporei comunemente visti si basano sul principio di misura della temperatura tramite l'espansione e la contrazione di un liquido quando riscaldato o raffreddato. L'immagine sottostante mostra un termometro ad alcol comunemente utilizzato che misura la temperatura tramite la proprietà di espansione e contrazione dell'alcol con la temperatura. La temperatura diurna invernale visualizzata nell'immagine è -17 °C.

### 2.2.1.3 Umidità

L'umidità è una grandezza fisica che indica il grado quanto è asciutta l'atmosfera. A una certa temperatura, meno vapore acqueo contiene un certo volume d'aria, più secca è l'aria; più vapore acqueo contiene, più umida è l'aria. La secchezza dell'aria è chiamata "umidità". Le previsioni del tempo in genere riportano i valori di umidità in termini di umidità relativa, che è una percentuale ottenuta confrontando la quantità effettiva di vapore acqueo nell'aria con la quantità massima di vapore acqueo che l'aria può contenere alla stessa temperatura.

### 2.2.1.4 Sensore di Temperatura e Umidità — Sensore di Temperatura e Umidità Grove V2.0 (DHT20)

Come suggerisce il nome, un sensore di temperatura e umidità è un sensore in grado di rilevare la temperatura e l'umidità nell'ambiente. Esistono molti tipi di sensori di temperatura e umidità e quello che abbiamo scelto è il [Sensore di temperatura e umidità Grove V2.0 \(DHT20\)](#). Si tratta di un prodotto a basso consumo, ad alta precisione e ad alta stabilità con un'interfaccia I2C digitale completamente calibrata e un intervallo di misura della temperatura di -40~80°C. I sensori di temperatura e umidità hanno un'ampia gamma di applicazioni nei settori dell'agricoltura, della protezione ambientale e della vita domestica.

#### **⚠️ Grove - Sensore di temperatura e umidità (DHT11)**

Se si utilizza la versione Grove DHT11 del sensore di temperatura e umidità (con una custodia blu del sensore), fare riferimento al [documento Wiki](#) di questa versione.

DHT11 è un sensore di temperatura e umidità che emette segnali digitali calibrati.

La differenza principale tra questo e DHT20 è il loro metodo di comunicazione:

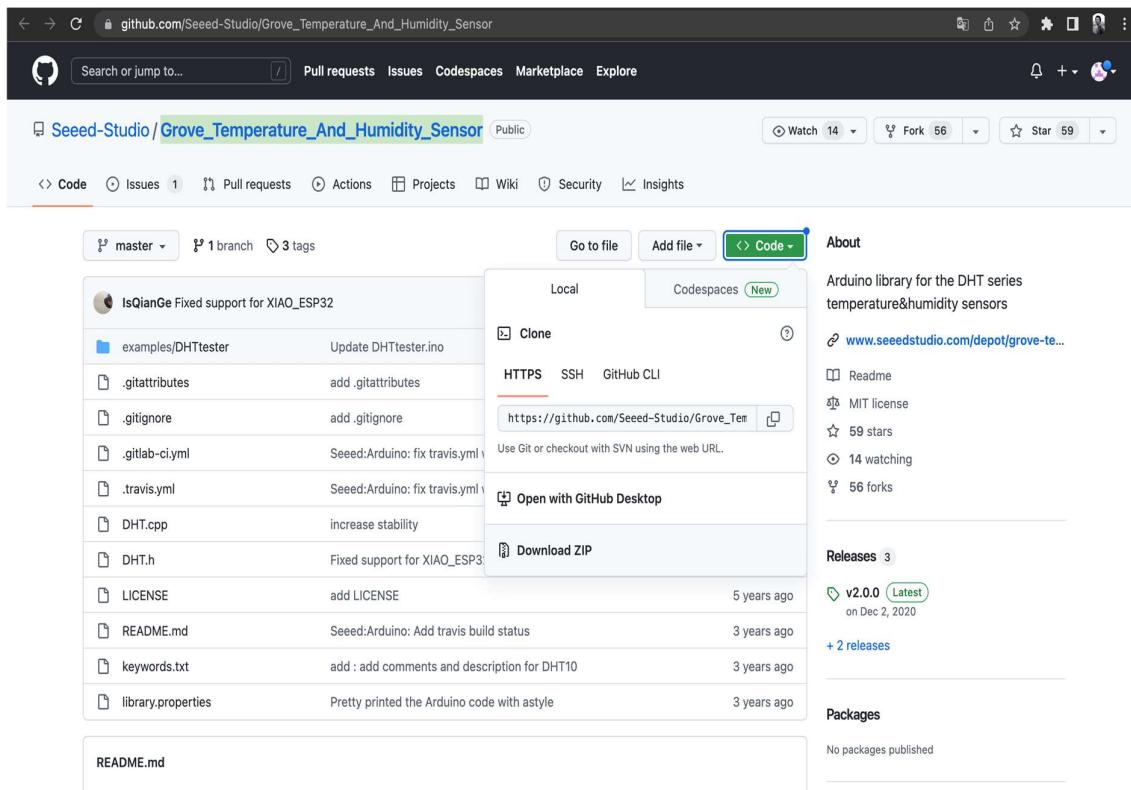
DHT11 utilizza un segnale digitale a bus singolo, mentre DHT20 utilizza un segnale I2C.

### 2.2.2 Attività 1: Lettura dei Valori di Temperatura e Umidità nel Monitor Seriale (Basato sul modello DHT20)

#### ***Aggiungere il File della Libreria Grove\_Temperature\_And\_Humidity\_Sensor***

Prima di iniziare a programmare il sensore di temperatura e umidità Grove con l'IDE Arduino, è necessario aggiungere i file di libreria necessari per il sensore. Digitare l'indirizzo del file di libreria nella barra degli indirizzi del browser:  [https://github.com/Seeed-Studio/Grove\\_Temperature\\_And\\_Humidity\\_Sensor](https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor), accedere alla pagina GitHub e cliccare su

Code-Download **.zip** per scaricare il pacchetto di risorse **Grove\_Temperature\_And\_Humidity\_Sensor-master.zip** sul computer locale, come mostrato nell'immagine sottostante.



Aggiungere il pacchetto di risorse **Grove\_Temperature\_And\_Humidity\_Sensor-master.zip** scaricato nel passaggio precedente nella barra dei menuù **Sketch->Include Library->Add .ZIP Library**, finché non si vede un messaggio che indica il caricamento riuscito della libreria.

### **Apertura dell'Esempio "DHTtester"**

Una volta aggiunto correttamente il file della libreria, è possibile utilizzare la libreria DHT. L'esempio "DHTtester" può essere aperto tramite il seguente path: **File->Examples->Grove Temperature And Humidity Sensor->DHTtester**.

**⚠️ Nota** Se l'esempio DHTtester non si trova nel menù dopo l'installazione dei file della libreria, può essere visualizzato chiudendo e riaprendo l'IDE Arduino.

Dopo aver aperto il programma di esempio, possiamo vedere un programma come quello mostrato di seguito. Questo programma legge la temperatura e l'umidità relativa nell'ambiente e visualizza i dati in tempo reale nel monitor seriale. Parte del codice del programma di esempio deve essere modificato.

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

// Uncomment whatever type you're using!
#ifndef DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
//#define DHTTYPE DHT10 // DHT 10
//#define DHTTYPE DHT20 // DHT 20

/*Notice: The DHT10 and DHT20 is different from other DHT* sensor ,it uses i2c interface rather than
one wire*/
/*So it doesn't require a pin.*/
#define DHTPIN 2 // what pin we're connected to (DHT10 and DHT20 don't need define it)
DHT dht(DHTPIN, DHTTYPE); // DHT11 DHT21 DHT22
//DHT dht(DHTTYPE); // DHT10 DHT20 don't need to define Pin
```

```

// Connect pin 1 (on the left) of the sensor to +5V
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

#if defined(ARDUINO_ARCH_AVR)
    #define debug Serial

#elif defined(ARDUINO_ARCH_SAMD) || defined(ARDUINO_ARCH_SAM)
    #define debug SerialUSB
#else
    #define debug Serial
#endif

void setup() {

    debug.begin(115200);
    debug.println("DHTxx test!");
    Wire.begin();

    /*if using WIO link,must pull up the power pin.*/
    // pinMode(PIN_GROVE_POWER, OUTPUT);
    // digitalWrite(PIN_GROVE_POWER, 1);

    dht.begin();
}

void loop() {
    float temp_hum_val[2] = {0};
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

    if (!dht.readTempAndHumidity(temp_hum_val)) {
        debug.print("Humidity: ");
        debug.print(temp_hum_val[0]);
        debug.print(" %\t");
        debug.print("Temperature: ");
        debug.print(temp_hum_val[1]);
        debug.println(" *C");
    } else {
        debug.println("Failed to get temprature and humidity value.");
    }

    delay(1500);
}

```

Prestare attenzione ai commenti del documento. Il programma sopra fornisce diversi tipi di modelli di sensori di temperatura e umidità (DHT22 è impostato di default), ma abbiamo bisogno di DHT20. Quindi, rimuovere il commento dalla parte per DHT20 ed eliminare le definizioni per altri modelli di sensori non necessari. DHT10 e DHT20 non richiedono definizioni di pin, quindi il codice rivisto dopo la modifica è il seguente:

```

#include "DHT.h"
#define DHTTYPE DHT20 // DHT 20
DHT dht(DHTTYPE);
#if defined(ARDUINO_ARCH_AVR)
#define debug Serial

#elif defined(ARDUINO_ARCH_SAMD) || defined(ARDUINO_ARCH_SAM)
#define debug SerialUSB
#else
#define debug Serial
#endif

void setup() {
    debug.begin(115200);
    debug.println("DHTxx test!");
    Wire.begin();
    dht.begin();
}

void loop() {
    float temp_hum_val[2] = {0};
    if (!dht.readTempAndHumidity(temp_hum_val)) {
        debug.print("Humidity: ");
        debug.print(temp_hum_val[0]);
        debug.print(" %\t");
        debug.print("Temperature: ");

```

```

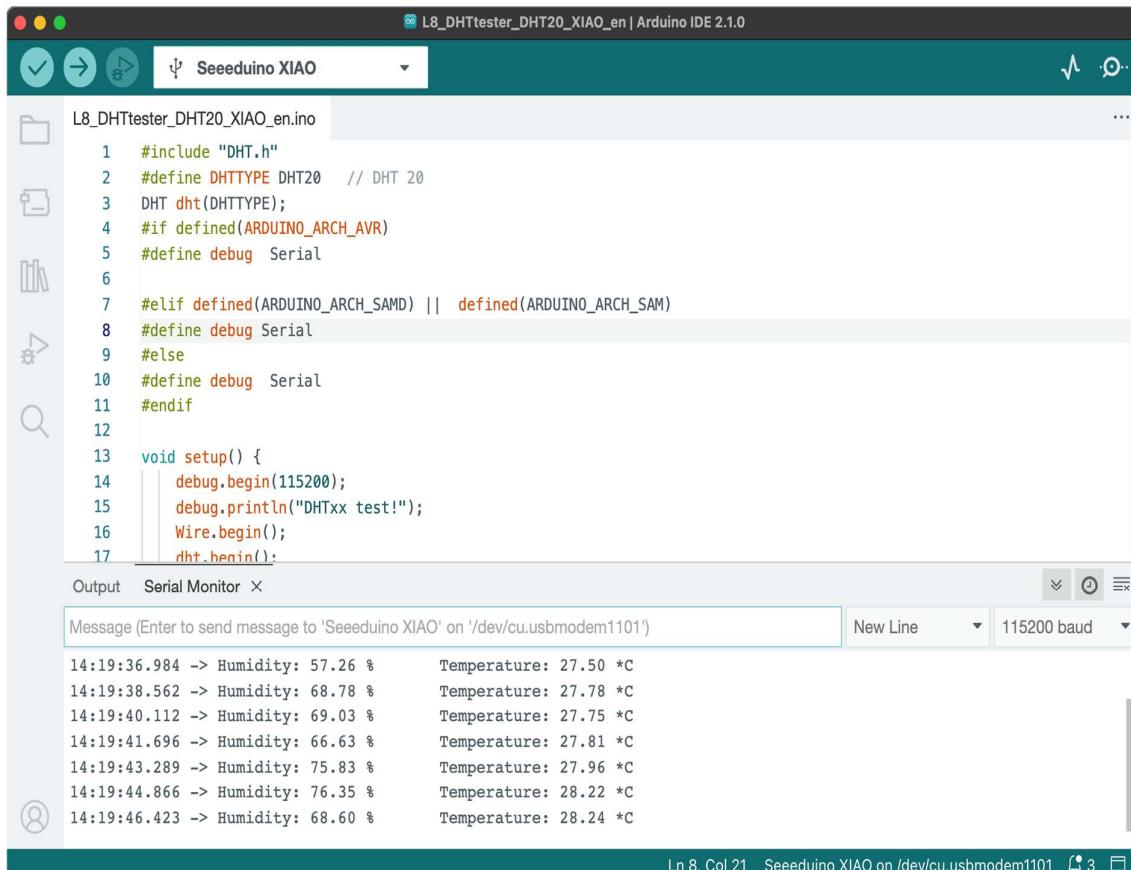
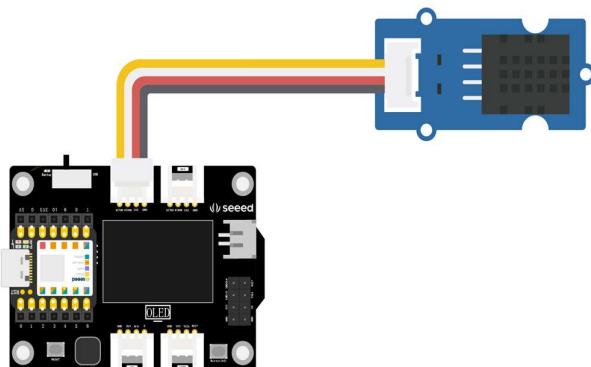
        debug.print(temp_hum_val[1]);
        debug.println(" *C");
    } else {
        debug.println("Failed to get temprature and humidity value.");
    }

    delay(1500);
}

```

Prelevare questo programma da Github[https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8\\_DHTtester\\_DHT20\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8_DHTtester_DHT20_XIAO_en)

Dopo aver modificato il codice, collegare prima il sensore di temperatura e umidità all’interfaccia I2C della scheda di espansione XIAO, come mostrato di seguito. Poi collegare la scheda di sviluppo XIAO al computer, caricare il programma di esempio modificato su XIAO nell’IDE Arduino e aprire il monitor seriale nell’IDE Arduino. Ora si sarà in grado di vedere i valori di temperatura e umidità. Provare a posizionare il sensore in ambienti diversi per osservare se i valori di temperatura e umidità cambiano.



Sembra che il sensore di temperatura e umidità funzioni correttamente.

## **Lettura dei Valori di Temperatura e Umidità nel Monitor Seriale (in Base al Sensore DHT11)**

Se si utilizza il sensore di temperatura e umidità Grove DHT11 con un involucro blu, alcune parti del codice del programma devono essere modificate come segue: `#define DHTPIN 0` deve essere modificato in base al numero di pin effettivo a cui è collegato il sensore. `#define DHTTYPE DHT11` deve essere impostato perché ci sono diversi modelli di sensori di temperatura e umidità e si deve scegliere quello corretto, ovvero DHT11. Il codice di esempio dopo la modifica è mostrato di seguito:

```
#include "DHT.h"
#define DHTTYPE DHT11 // DHT 11
#define DHTPIN 0
DHT dht(DHTPIN, DHTTYPE);

#if defined(ARDUINO_ARCH_AVR)
    #define debug Serial

#elif defined(ARDUINO_ARCH_SAMD) || defined(ARDUINO_ARCH_SAM)
    #define debug SerialUSB
#else
    #define debug Serial
#endif

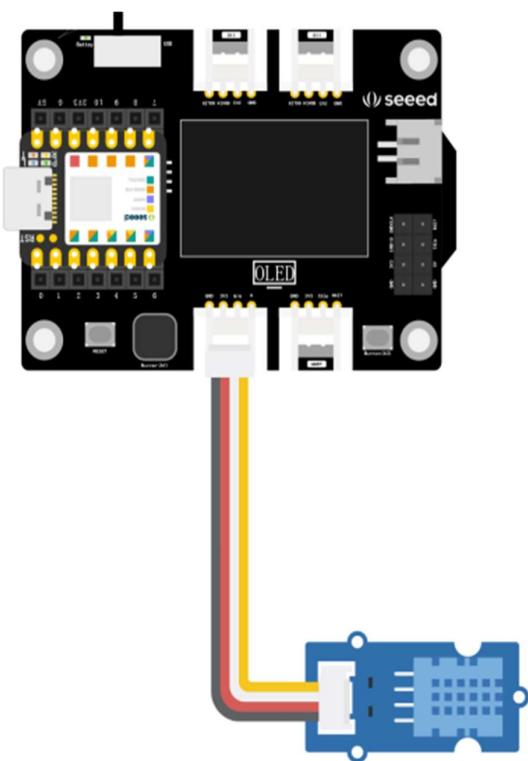
void setup() {
    debug.begin(115200);
    debug.println("DHTxx test!");
    Wire.begin();
    dht.begin();
}

void loop() {
    float temp_hum_val[2] = {0};
    if (!dht.readTempAndHumidity(temp_hum_val)) {
        debug.print("Humidity: ");
        debug.print(temp_hum_val[0]);
        debug.print(" %\t");
        debug.print("Temperature: ");
        debug.print(temp_hum_val[1]);
        debug.println(" *C");
    } else {
        debug.println("Failed to get temprature and humidity value.");
    }

    delay(1500);
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8\\_DHTrster\\_DHT11\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8_DHTrster_DHT11_XIAO_en)

Dopo aver modificato il codice, collegare prima il sensore di temperatura e umidità alla porta A0 della scheda di espansione XIAO, come mostrato nella figura sottostante. Poi, collegare la scheda di sviluppo XIAO al computer, caricare il programma di esempio modificato su XIAO nell'IDE Arduino e aprire il monitor seriale nell'IDE Arduino per vedere i valori di temperatura e umidità. Si può posizionare il sensore di temperatura e umidità in diversi ambienti per vedere se i valori di temperatura e umidità cambieranno.



### 2.2.3 Creazione del Progetto: Misuratore Intelligente di Temperatura e Umidità

#### Descrizione del Progetto

Stiamo per realizzare un mini rilevatore portatile di temperatura e umidità che rileva tali valori tramite un sensore di temperatura e umidità e li visualizza sul display OLED della scheda di espansione XIAO. Tuttavia, non è abbastanza ricco da avere solo la funzione di visualizzazione. Possiamo aggiungere una funzione di allarme acustico. Quando la temperatura e l'umidità rilevate superano un certo intervallo, verrà emesso un allarme come promemoria. L'intervallo di valori può essere regolato in base a diversi scenari applicativi. Ad esempio, in uno scenario di vita domestica, impostare un intervallo di temperatura e umidità confortevole in base alle sensazioni umane; oppure utilizzarlo in luoghi di piantagione di piante, impostare l'intervallo di valori di temperatura e umidità in base alla crescita adatta delle piante, superare l'allarme e ricordare alle persone di regalarsi.

#### Scrittura del Programma

Facendo riferimento al programma di esempio sopra, uno degli effetti che vogliamo ottenere è visualizzare i valori di temperatura e umidità sul display OLED della scheda di espansione XIAO. Il codice per la lettura dei valori di rilevamento del sensore di temperatura e umidità può essere riutilizzato semplicemente cambiando il mezzo di visualizzazione. In combinazione con la Sezione 1.6, abbiamo imparato come visualizzare i caratteri sull'OLED, quindi dobbiamo solo aggiungere un'istruzione di giudizio della condizione if...else per giudicare i valori di temperatura e umidità. L'idea di scrittura del programma è la seguente:

- Dichiarare la libreria DHT.h, la libreria U8x8, ecc. e collegare il pin del cicalino come promemoria per far suonare il dispositivo.
- Inizializzare il file della libreria, definire lo stato del pin del cicalino.
- Definire le variabili di temperatura e umidità per memorizzare le letture e visualizzarle sullo schermo OLED, aggiungere un giudizio logico e implementare l'allarme del cicalino.

Per facilitare la comprensione e l'implementazione, dividiamo l'implementazione del programma in due attività:

1. Rilevare la temperatura e l'umidità e visualizzarle sullo schermo OLED della scheda di espansione XIAO.
2. Aggiungere la funzione di allarme.

### **Attività 1: Utilizzare il sensore Grove DHT20 per rilevare temperatura e umidità e visualizzarle sullo schermo OLED della scheda di espansione XIAO**

**Passaggio 1:** Header, dichiarare i file di libreria da chiamare.

```
#include "DHT.h"      //Utilizza la libreria DHT
#include <Arduino.h>
#include <U8x8lib.h>    //Utilizza la libreria u8x8
#define DHTTYPE DHT20
DHT dht(DHTTYPE);    //DHT20 non ha bisogno di definire pin

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8 /* reset=*/ U8X8_PIN_NONE); //Imposta il costruttore per la connessione allo schermo OLED
```

**Passaggio 2:** Inizializzare la libreria DHT e la libreria u8x8.

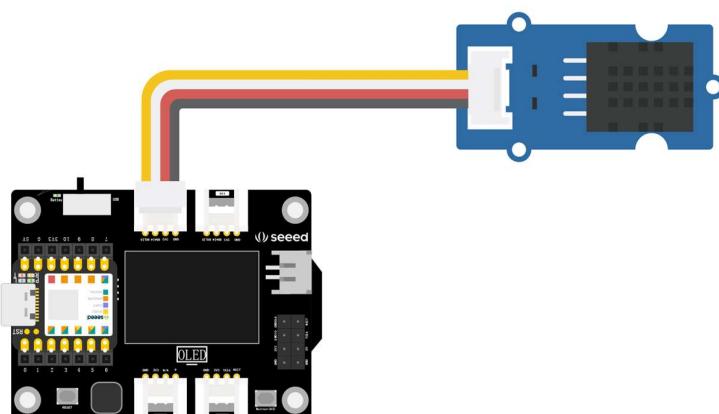
```
void setup() {
    Wire.begin();    //Initialize wire library, and join I2C network
    dht.begin();    //DHT starts working
    u8x8.begin();    //u8x8 starts working
    u8x8.setPowerSave(0); //Turn off power saving mode, 1 is on, and nothing can be seen on the screen after power saving mode is on
    u8x8.setFlipMode(1);
}
```

**Passaggio 3:** Definire le variabili di temperatura e umidità per memorizzare le letture, leggere i valori di temperatura e umidità e visualizzarli sullo schermo OLED. Presta attenzione alle posizioni delle coordinate della visualizzazione di temperatura e umidità.

```
void loop() {
    float temp, humi;    //Set the variables temp and humi to floating point type, representing temperature and humidity respectively
    temp = dht.readTemperature(); //Read temperature value and store it in temp
    humi = dht.readHumidity(); //Read humidity value and store it in humi
    u8x8.setFont(u8x8_font_chroma48medium8_r); //Set display font
    u8x8.setCursor(0, 33); //Set the position of the drawing cursor (0,33)
    u8x8.print("Temp:");
    u8x8.print(temp); //Display real-time temperature value
    u8x8.print("C"); //Display the unit "C" of temperature
    u8x8.setCursor(0, 50);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
}
```

Prelevare questo programma da Github[https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8\\_dht20\\_tem\\_humi\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8_dht20_tem_humi_XIAO_en)

**Passaggio 4:** Collegare l'hardware, caricare il programma, collegare il sensore di temperatura e umidità all'interfaccia I2C della scheda di espansione XIAO, come mostrato nella figura:



Utilizzare il cavo dati per collegare XIAO al computer, cliccare sul pulsante “upload” nell’IDE Arduino e caricare il programma sull’hardware. Quando l’area di debug mostra “upload successful”, è possibile osservare se i valori di temperatura e umidità vengono visualizzati sullo schermo OLED e tenere la parte nera del sensore con il palmo della mano per osservare se i valori cambiano.

### Attività 2: Aggiungere una funzione di allarme

**Passaggio 1:** Aggiungere il codice della funzione di allarme. La funzione di allarme richiede l’integrazione di un cicalino nel circuito, che può essere facilitato utilizzando il cicalino di bordo della scheda di espansione XIAO. Il programma deve impostare lo stato del pin del cicalino, aggiungere una parte per il giudizio di condizione: quando la temperatura supera un certo valore o l’umidità scende al di sotto di un certo valore, il cicalino emetterà un allarme. Qui, è necessario scrivere un’espressione logica utilizzando l’operatore logico “`&&`” “and” [e].

### Operatori Booleani

- `&&`: AND logico, rappresenta “e”, `if (espressione1 && espressione2)`, solo quando tutte le espressioni tra parentesi sono `vere` eseguirà le istruzioni in `if {}`.
- `||`: OR logico, rappresenta “o”, `if (espressione1 || espressione2)`, se una delle espressioni è soddisfatta, l’intera espressione è `vera` e le istruzioni in `if {}` vengono eseguite.
- `!`: NOT logico, rappresenta “non”, `if (!espressione1)`, solo quando il valore di espressione1 tra parentesi è `falso` eseguirà le istruzioni in `if {}`.

**Esempio di utilizzo:** Quando la temperatura supera i 30 o l’umidità scende sotto i 40, il soddisfacimento di una delle due condizioni farà suonare un allarme.

---

```
if (temp > 30 || humi < 40) {
    tone(buzzerPin, 200, 200);
}
```

---

La parte aggiunta del programma imposta principalmente il cicalino e prende decisioni in base alla temperatura e all’umidità, controllando il cicalino per emettere un suono.

---

```
// Part of the program, will not run
int buzzerPin = A3; // Connects the buzzer to pin A3

void setup() {
    pinMode(buzzerPin, OUTPUT); // Sets the buzzer pin as output
}

void loop() {
    float temp, humi;
    temp = dht.readTemperature();
    humi = dht.readHumidity();
    if (temp > 30 || humi < 40) { // Quando la temperatura supera 30 o l'umidità scende sotto 40,
        il soddisfacimento di una delle due condizioni farà suonare un allarme.
        tone(buzzerPin, 200, 200);
    }
}
```

---

Aggiungere il codice precedente alla posizione corrispondente del programma Task 1 per realizzare tutte le funzioni. Il programma completo è mostrato di seguito:

---

```
#include "DHT.h" // Use DHT library
#include <Arduino.h>
#include <U8x8lib.h> // Use u8x8 library
#define DHTTYPE DHT20
DHT dht(DHTTYPE); // DHT20 does not require pin definition
int buzzerPin = A3;
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8 /* reset=*/ U8X8_PIN_NONE; // Set constructor to connect
OLED display

void setup() {
    pinMode(buzzerPin, OUTPUT); // Set buzzer pin to output mode
    Wire.begin(); // Initialize Wire library and join to I2C network
    dht.begin(); // DHT begins operation
    u8x8.begin(); // u8x8 begins operation
    u8x8.setPowerSave(0); // Disable power save mode, 1 is enable. After enabling power save mode,
nothing will be seen on the screen
    u8x8.setFlipMode(1);
```

```

}

void loop() {
    float temp, humi; // Set variables temp and humi to floating point type, representing
temperature and humidity respectively
    temp = dht.readTemperature(); // Read temperature value and store it in temp
    humi = dht.readHumidity(); // Read humidity value and store it in humi
    if (temp > 30 || humi < 40) { // When the temperature is above 30 or the humidity is below 40,
if either condition is met, the buzzer will sound an alarm
        tone(buzzerPin, 200, 200);
    }

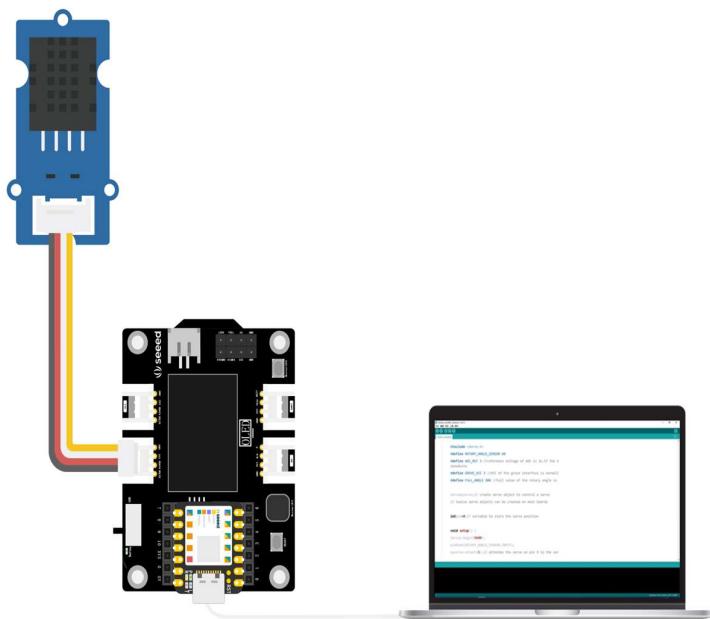
    u8x8.setFont(u8x8_font_chroma48medium8_r); // Set display font
    u8x8.setCursor(0, 33); // Set the position of the drawing cursor (0,33)
    u8x8.print("Temp:"); // Display "Temp:" at the position (0,33)
    u8x8.print(temp); // Then display the real-time temperature value
    u8x8.print("C"); // Then display the unit of temperature "C"
    u8x8.setCursor(0, 50);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
}

```

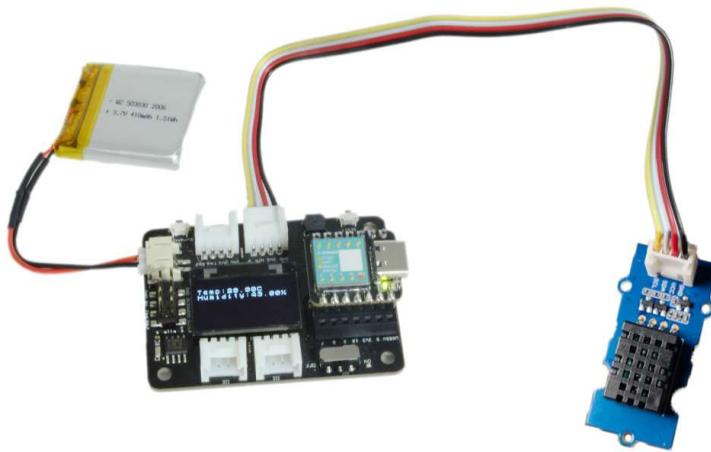
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8\\_dht20\\_alarm\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8_dht20_alarm_XIAO_en)

### Passaggio 2: Caricare il programma.

Dopo aver scritto il programma, collegare la scheda di controllo principale XIAO al computer tramite un cavo dati, come mostrato nell'immagine sottostante:



Dopo la connessione, cliccare sul pulsante “Verify” per controllare il programma. Se la verifica ha esito positivo, fai clic sul pulsante “Upload” per caricare il programma sull’hardware. Quando l’area di debug mostra “Upload Successful”, è completato. Per verificare se la funzione di allarme funziona senza problemi, afferrare saldamente il sensore di temperatura e umidità con la mano, osserva il cambiamento di valore sul display OLED e ascoltare l’allarme acustico quando la temperatura supera i 30°C.



**Attività 2-2: Utilizzare il sensore Grove DHT11 per visualizzare la temperatura e l'umidità sull'OLED della scheda di estensione XIAO e aggiungere una funzione di allarme.**

Per il sensore Grove DHT11 con involucro blu, il programma è mostrato di seguito:

```
#include "DHT.h" //Use DHT library
#include <Arduino.h>
#include <U8x8lib.h> //Use u8x8 library
#define DHTPIN 0
#define DHTTYPE DHT11 //Specify using DHT11
DHT dht(DHTPIN, DHTTYPE);
int buzzerPin = A3;
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE); //Set constructor to connect OLED
display

void setup() {
    pinMode(buzzerPin, OUTPUT); //Set buzzer pin to output mode
    Wire.begin(); //Initialize wire library and join to I2C network
    dht.begin(); //DHT begins operation
    u8x8.begin(); //u8x8 begins operation
    u8x8.setPowerSave(0); //Disable power save mode, 1 is enable. After enabling power save mode,
    nothing will be seen on the screen
    u8x8.setFlipMode(1);
}

void loop() {
    float temp, humi; //Set variables temp and humi to floating point type, representing temperature
    and humidity respectively
    temp = dht.readTemperature(); //Read temperature value and store it in temp
    humi = dht.readHumidity(); //Read humidity value and store it in humi
    if (temp > 30 || humi < 40) { //When the temperature is above 30 or the humidity is below 40, if
    either condition is met, the buzzer will sound an alarm
        tone(buzzerPin, 200, 200);
    }

    u8x8.setFont(u8x8_font_chroma48medium8_r); //Set display font
    u8x8.setCursor(0, 33); //Set the position of the drawing cursor (0,33)
    u8x8.print("Temp:");
    u8x8.print(temp); //Then display the real-time temperature value
    u8x8.print("C"); //Then display the unit of temperature "C"
    u8x8.setCursor(0, 50);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
}
```

Prelevare questo programma da Github[https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8\\_dht11\\_alarm\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L8_dht11_alarm_XIAO_en)

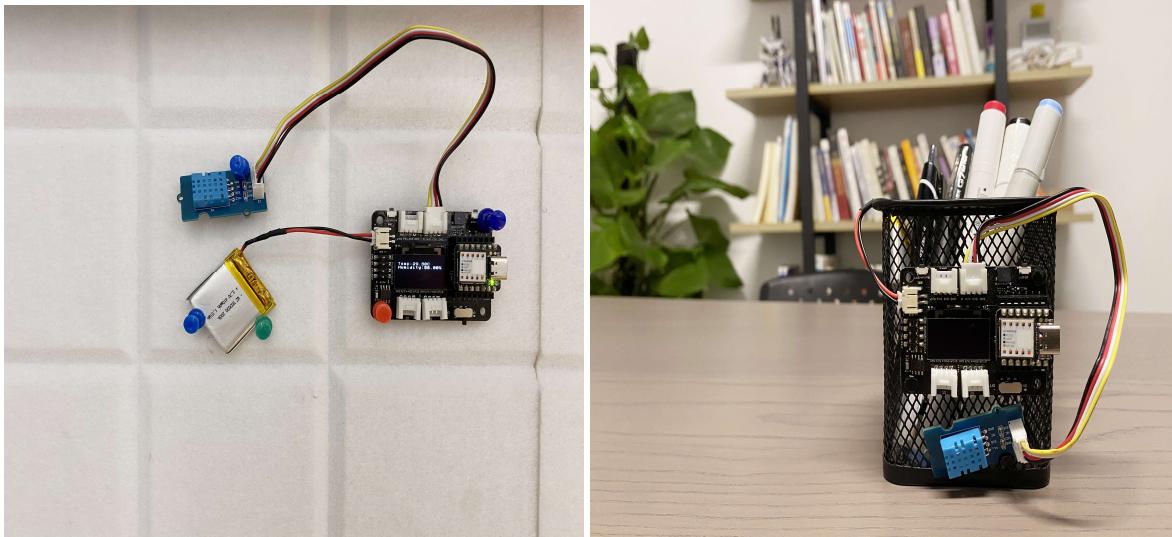
## 2.2.4 Design dell'Aspetto

A partire da questa sezione, aggiungeremo la parte del design dell'aspetto, iniziando a esplorare la produzione completa del prototipo del prodotto. Inizialmente, possiamo provare a disegnare schizzi di design e apportare una semplice modifica con i materiali a portata di mano. Tornando al misuratore di temperatura e umidità intelligente in questa sezione, si

prega di progettare l'aspetto del prototipo in base alle caratteristiche e alle funzioni del prodotto.

| Nome del Prodotto            | Misuratore di Temperatura e Umidità Intelligente                                                                                                                           |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Caratteristiche del Prodotto | Piccolo, portatile, alta sensibilità.                                                                                                                                      |
| Funzioni del Prodotto        | Visualizzazione in tempo reale dei valori di temperatura e umidità ed emissione di un allarme quando i valori di temperatura e umidità superano l'intervallo confortevole. |
| Aspetto del Prodotto         | (Ad esempio, trasformato in un ciondolo da appendere allo zaino che si porta in giro, da attaccare alla scatola porta fazzoletti in camera da letto, ecc.)                 |

## Caso di riferimento



## 2.3 Scatola Regalo a Sorpresa Basata sul Sensore di Luce

State pensando di fare un regalo di compleanno speciale ad un amico? Invece di comprarne uno, lo si può creare con i moduli che abbiamo a portata di mano. In questa sezione, creeremo una scatola regalo a sorpresa per un caro amico. Che tipo di sorpresa apparirà quando la scatola regalo verrà aperta? Di che tipo di moduli abbiamo bisogno per completare una scatola regalo a sorpresa del genere? Iniziamo la lezione di oggi con queste domande.

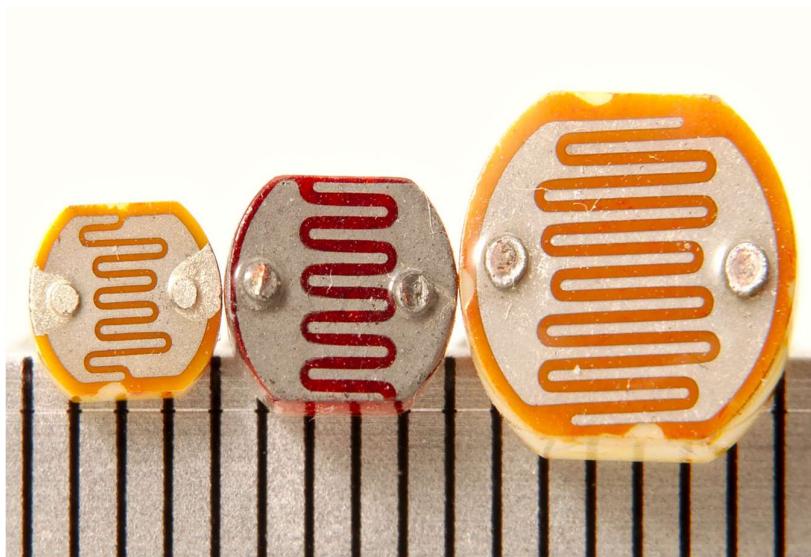
### 2.3.1 Conoscenza di Base

#### 2.3.1.1 Sensore di Luce

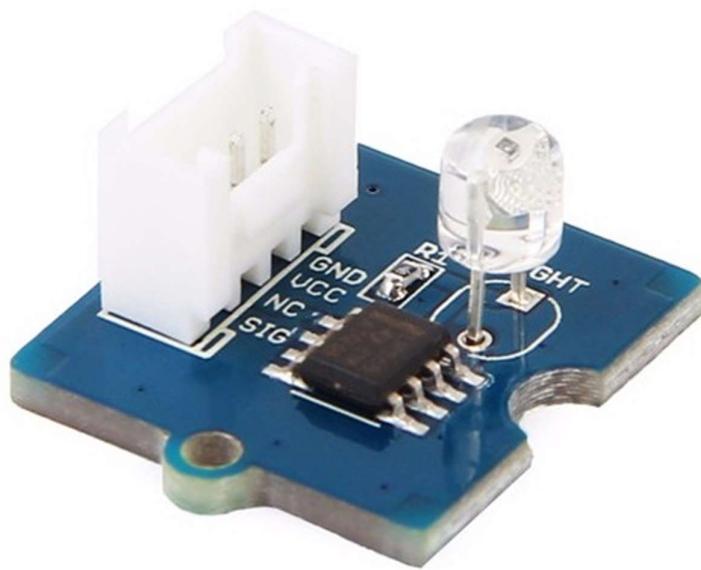
I sensori di luce possono rilevare l'intensità della luce nell'ambiente circostante e convertire l'energia luminosa rilevata in energia elettrica. I sensori di luce sono divisi in tipi come fotoresistivi, fotodiodo e transistor fotoelettrico. Qui, introdurremo semplicemente due sensori di luce comunemente usati, fotoresistivi e fotodiodo.

**Tipo Fotoresistivo** Innanzitutto, il tipo fotoresistivo, il suo modulo integrerà un fotoresistore, come mostrato di seguito. Il fotoresistore è estremamente sensibile alla luce, qualsiasi luce visibile ai nostri occhi può causare la sua reazione. La luce ad alta intensità causerà la diminuzione del valore di resistenza, mentre la luce a bassa intensità causerà l'aumento del

valore di resistenza. Regolando il valore di resistenza nel circuito tramite l'intensità della luce, si possono controllare altri dispositivi, come il controllo dell'accensione e dello spegnimento della luce LED.



**Tipo Fotodiodo** I fotodiodi, noti anche come sensori fotoelettrici o fotodetector, quando un raggio di luce colpisce il diodo, gli elettroni nel tubo si disperdono rapidamente per formare lacune elettroniche, causando così il flusso di corrente. Più forte è la luce, più forte è la corrente. Poiché la corrente generata dal fotodiodo è proporzionale all'intensità della luce, è molto utile per il rilevamento della luce che richiede un rapido cambiamento nella risposta alla luce. Il sensore di luce che utilizzeremo in questa lezione è di questo tipo.



Parlando degli usi dei sensori di luce, possiamo costruire un interruttore controllato dalla luce tramite un sensore di luce, ad esempio controllare l'accensione e lo spegnimento della luce tramite un sensore di luce, spegnere la luce durante il giorno e accendere la luce di notte. Lo scopo principale del dispositivo di controllo della luce è risparmiare energia, migliorare l'efficienza tramite automazione intelligente, il più comune nella vita è probabilmente la luce controllata dalla luce, la lampada da tavolo controllata dalla luce, il lampioncino controllato dalla luce, l'illuminazione del tunnel autostradale, ecc., portando

comodità nella nostra vita e contribuendo anche alla protezione ambientale e al risparmio energetico.

### 2.3.1.2 Striscia di LED RGB



Il progetto in questa lezione è abbinato a una striscia LED RGB. La striscia integra più perle luminose regolabili in base al colore. Rispetto a un singolo LED, può ottenere più effetti di luce e fantastici impatti visivi, rendendolo ideale per creare sorprese. Le strisce LED RGB sono disponibili in vari stili e modelli. Quella che utilizzeremo è la [striscia LED RGB Grove - WS2813](#), modello a 30 perle. Possiamo controllare la striscia LED RGB per ottenere un ricco effetto di illuminazione tramite programmazione e creare progetti di illuminazione più interessanti.

### 2.3.2 Attività 1: Accendere la Striscia LED RGB Per Iniziare a Usare le Strisce LED RGB, Iniziare installando e comprendendone la libreria.

#### **Aggiungere la Libreria Adafruit\_NeoPixel**

Prima di iniziare a programmare la striscia LED RGB con l'IDE Arduino, si devono aggiungere i file di libreria necessari. Si inserisci l'indirizzo del file di libreria [https://github.com/adafruit/Adafruit\\_NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel) nella barra degli indirizzi del browser, si accede alla pagina GitHub, si clicca su `Code > Download ZIP` per scaricare il pacchetto di risorse `Adafruit_NeoPixel-master.zip` sul computer locale.

The screenshot shows the GitHub repository page for Adafruit\_NeoPixel. The repository has 218 stars, 1.2k forks, and 2.8k issues. It contains 6 branches and 52 tags. The code tab is selected, showing a list of files including README.md, esp.c, examples, ignore, Adafruit\_NeoPixel.cpp, Adafruit\_NeoPixel.h, CONTRIBUTING.md, COPYING, and several .c and .h files. On the right side, there's an 'About' section describing the library as an Arduino library for controlling single-wire LED pixels (NeoPixel, WS2812, etc.), an 'arduino-library' badge, and sections for Releases (with the latest being v1.1.1), Packages (none published), and Contributors (67). At the bottom, there are links for Arduino Library CI, passing, Docs, and Doxygen.

Quindi, si aggiunge il pacchetto di risorse `Adafruit_NeoPixel-master.zip` scaricato nel passaggio precedente tramite la barra dei menu `Sketch->Include Library->Add .ZIP Library` finché non si vede la libreria caricata correttamente.

### **Aprire l'Esempio Semplice**

Si può aprire un semplice esempio tramite il seguente path: File → Examples → Adafruit NeoPixel → simple. Una volta aperto il programma di esempio, possiamo vedere il seguente programma:

```
// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// Released under the GPLv3 license to match the rest of the
// Adafruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifndef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN 6 // On Trinket or Gemma, suggest changing this to 1

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 16 // Popular NeoPixel ring size

// When setting up the NeoPixel library, we tell it how many pixels,
// and which pin to use to send signals. Note that for older NeoPixel
// strips you might need to change the third parameter -- see the
// strandtest example for more information on possible values.
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

#define DELAYVAL 500 // Time (in milliseconds) to pause between pixels

void setup() {
    // These lines are specifically to support the Adafruit Trinket 5V 16 MHz.
    // Any other board, you can remove this part (but no harm leaving it):
    #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
    clock_prescale_set(clock_div_1);
    #endif
    // END of Trinket-specific code.

    pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
}

void loop() {
    pixels.clear(); // Set all pixel colors to 'off'

    // The first NeoPixel in a strand is #0, second is 1, all the way up
    // to the count of pixels minus one.
    for(int i=0; i<NUMPIXELS; i++) { // For each pixel...

        // pixels.Color() takes RGB values, from 0,0,0 up to 255,255,255
        // Here we're using a moderately bright green color:
        pixels.setPixelColor(i, pixels.Color(0, 150, 0));

        pixels.show(); // Send the updated pixel colors to the hardware.

        delay(DELAYVAL); // Pause before next pass through loop
    }
}
```

Questo programma consente alla striscia di accendere 30 perline (luce verde) in sequenza. Questo è un semplice esempio di striscia luminosa e dobbiamo modificare alcuni parametri:

`#define PIN 0`, si deve modificare il pin collegato alla striscia luminosa in base alla situazione effettiva. È collegato all'interfaccia `A0` della scheda di espansione XIAO, quindi è `PIN 0`. `#define NUMPIXELS 30`, definisce il numero di LED nella striscia luminosa. Poiché la striscia luminosa ha modelli diversi e il numero di perline integrate è diverso, utilizziamo una striscia luminosa con 30 perline, quindi è `NUMPIXELS 30`.

Dopo aver modificato i parametri, puoi rimuovere i commenti in inglese per una visione più chiara del codice. Occupa una grande quantità di spazio.

---

```
#include <Adafruit_NeoPixel.h> // Header file, declaring the library
#ifndef __AVR__
#include <avr/power.h>
```

```
#endif

#define PIN 0 // The light strip is connected to pin 0. If you are using XIAO RP2040, please change
0 to A0
#define NUMPIXELS 30 // The number of LED lights on the light strip
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800); // Create a new light strip object,
define data mode
#define DELAYVAL 500 // The interval time for each light to light up

void setup() {
    #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
    clock_prescale_set(clock_div_1);
    #endif
    pixels.begin(); // The light strip is ready to output data
}

void loop() {
    pixels.clear(); // All beads on the light strip are turned off
    for(int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0, 150, 0)); // Light up the beads in sequence, the
color is green
        pixels.show(); // Display the light strip
        delay(DELAYVAL);
    }
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9\\_NeoPixel30\\_simple\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9_NeoPixel30_simple_XIAO_en)

Nel codice sopra, `pixels.color(0,150,0)` è una funzione per impostare il colore della striscia luminosa a LED. I numeri tra parentesi rappresentano rispettivamente i tre colori primari (rosso, verde, blu). Se è `(0,150,0)`, significa che la luminosità del rosso è 0, la luminosità del verde è 150 e la luminosità del blu è 0. L'intera striscia luminosa mostrerà un effetto verde. Più grande è il numero, più luminosa sarà, con un massimo di 255. Poi, collegare la striscia luminosa all'interfaccia `A0/D0` della scheda di espansione XIAO, come mostrato nella figura seguente:



Collegare la scheda madre XIAO al computer con un cavo dati e caricare il programma sulla scheda madre. Dopo che il caricamento è riuscito, osservare l'effetto della striscia luminosa.

La striscia luminosa può cambiare colore, tremolare e presentare vari effetti di luce come il “respiro”. Possiamo fare riferimento al programma di esempio nella libreria: **File → Example → Adafruit NeoPixel → buttoncycler**. Questo programma di esempio commuta diversi effetti di luce sulla striscia luminosa tramite pulsanti. Possiamo trovare il codice per vari effetti di luce al suo interno, come tremolio, luci arcobaleno, inseguimento, ecc.

### 2.3.3 Progetto: Scatola Regalo a Sorpresa

#### Descrizione del Progetto

Il programma per la scatola regalo a sorpresa che si vuole realizzare: Utilizzare un sensore di luce per controllare l'accensione e lo spegnimento della striscia luminosa a LED RGB, proprio come una lampada a luce controllata, ma l'effetto è opposto. Quando il valore rilevato dal sensore di luce è inferiore a un valore fisso, ovvero in un ambiente poco

illuminato, la striscia luminosa a LED RGB è spenta. Quando il valore rilevato dal sensore di luce è superiore a un valore fisso, ovvero in un ambiente luminoso, la striscia luminosa a LED RGB accende la luce arcobaleno.

## Scrittura del Programma

L'idea di scrittura del programma è la seguente:

- Dichiarare i file da chiamare, creare un nuovo oggetto striscia luminosa, definire il pin del sensore e il numero di LED sulla striscia luminosa.
- Inizializzare la striscia luminosa e impostare la modalità pin del sensore di luce.
- Leggere il valore della luce. Se il valore della luce è superiore a 100, la striscia luminosa presenterà un effetto di luce arcobaleno e “pulsazione”. In caso contrario, la striscia luminosa si spegnerà.

Il programma si completa in due task:

### **Task 1: Fare in Modo che la Striscia Luminosa Presenti un Effetto di Luce Arcobaleno e “Respiro”**

**Passaggio 1:** Dichiarare i file da chiamare, dichiarare l'oggetto striscia luminosa e definire il pin e il numero di LED sulla striscia luminosa.

---

```
#include <Adafruit_NeoPixel.h> // Header file, declaring the library
#ifndef __AVR__
#include <avr/power.h>
#endif

#define PIXEL_PIN 0 // The light strip is connected to pin A0. If you are using XIAO RP2040, please
change 0 to A0
#define PIXEL_COUNT 30 // The number of LED lights on the light strip
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
// Declare a new light strip object and define the data mode
```

---

### **Passaggio 2: Inizializzare la striscia luminosa.**

---

```
void setup() {
    strip.begin(); // Initialize the light strip, the light strip is ready to output data
}
```

---

### **Passaggio 3:** La striscia luminosa presenta un effetto di luce arcobaleno e “respirazione”.

Questa parte utilizza la funzione `for()` per presentare l'effetto respirazione. Per esempio, `for(i = 0; i<5; i++){};` significa che il valore iniziale di `i` è 0, quando `i` è minore di 5, viene eseguita l'istruzione nel corpo del ciclo {}, ogni volta che il ciclo viene eseguito, `i` viene incrementato di 1. Questo ciclo verrà eseguito 5 volte.

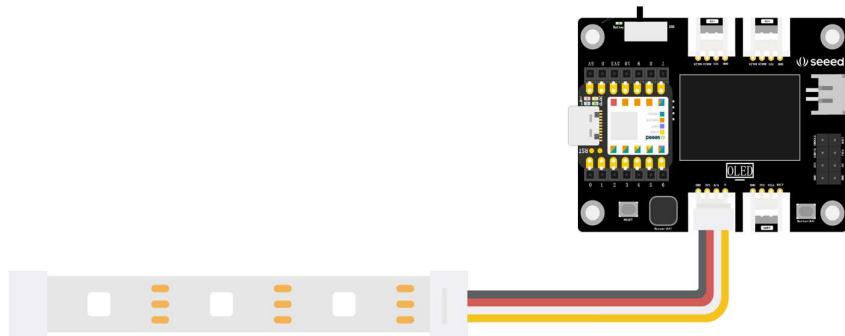
---

```
void loop() {
    strip.clear(); // Turn off all the lights on the light strip
    rainbow(10); // The light strip displays a rainbow light effect. The number in the parenthesis
represents the speed of the rainbow light circulation. The smaller the number, the faster the
circulation speed
}
// The following is the code for the rainbow light effect, presenting the breathing light effect.
This code can be found in the example program buttoncycler
void rainbow(int wait) {
    for(long firstPixelHue = 0; firstPixelHue < 3*65536; firstPixelHue += 256) {
        for(int i=0; i<strip.numPixels(); i++) {
            int pixelHue = firstPixelHue + (i * 65536L / strip.numPixels());
            strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
        }
    }
    strip.show(); // The light strip presents a light effect
    delay(wait); // Delay
}
```

---

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9\\_Rainbow\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9_Rainbow_XIAO_en)

**Passaggio 4:** Collegare l'hardware e caricare il programma. Per prima cosa, collega la striscia luminosa LED RGB all'interfaccia `A0/D0` della scheda di espansione XIAO, come mostrato nella figura:



Utilizzare un cavo dati per collegare XIAO al computer, cliccare sul pulsante “Upload” e caricare il programma sull’hardware. Quando l’area di debug mostra “Upload successful”, si può osservare l’effetto luminoso della striscia luminosa.



**Attività 2: Aaggiunta della Funzione di Interruttore di Controllo della Luce**

### **Passaggio 1: Aggiungere il codice.**

La funzione aggiunta è principalmente quella di leggere il valore di luce rilevato dal sensore di luce e utilizzare l'istruzione `if...else...` per giudicare il valore di luce. Quando è maggiore di 100 (questo valore può essere regolato in base all'ambiente reale), la striscia luminosa a LED RGB mostrerà un effetto di luce arcobaleno. Parte aggiunta del programma:

```

// Questa è una parte aggiunta del programma, non può essere eseguita direttamente
#define LIGHT_PIN 7 // Definisce il sensore di luce collegato ad A7. Se si utilizza XIAO RP2040,
modificare 7 in A3. Se si utilizza XIAO BLE, modificare 7 in 5
#define PIXEL_PIN 0 // Definisce la striscia luminosa. Se si utilizza XIAO RP2040, modificare 0 in A0
int readValue = 0; // Definire la variabile readValue per memorizzare il valore della luce
void setup() {
    pinMode(LIGHT_PIN, INPUT); // Impostare il pin del sensore di luce come stato di input
}
void loop() {
    readValue = analogRead(A7); // Leggere il valore analogico della luce del pin A7 e memorizzarlo
nella variabile readValue. Se si utilizza XIAO RP2040, modificare A7 in A3. Se si utilizza XIAO BLE,
modificare A7 in A5
    if(readValue > 500){ // Giudizio condizionale, se il valore della luce è maggiore di 500, la
striscia luminosa presenta un effetto di luce arcobaleno, altrimenti la striscia luminosa viene
spenta
        rainbow(10);
    }else {
        strip.clear();
        strip.show();
    }
}

```

```
    }
}
```

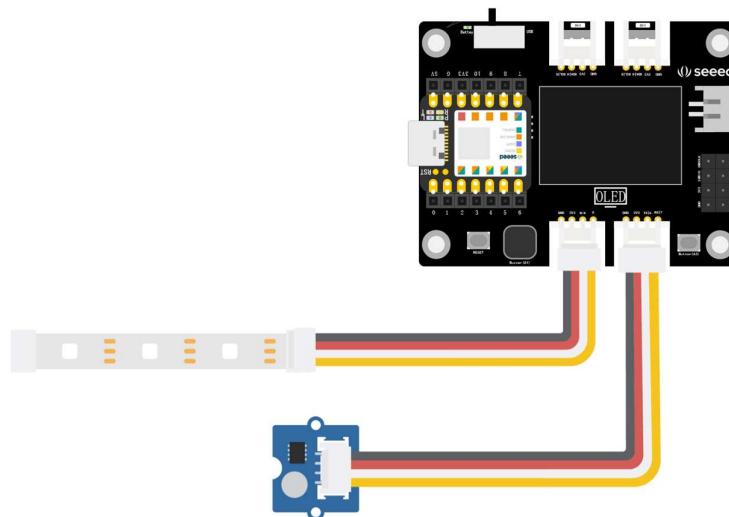
Aggiungiamo l'istruzione immessa alla posizione corrispondente del programma Task 1.

Guardare il programma completo:

```
#include <Adafruit_NeoPixel.h> // Header file, declare library
#ifndef __AVR__
#include <avr/power.h>
#endif
#define LIGHT_PIN 7 // Define the light sensor connected to A7. Se si utilizza XIAO RP2040,
modificare 7 in A3. If you are using XIAO BLE, please change 7 to 5
#define PIXEL_PIN 0 // The light strip is connected to the A0 pin. If you are using XIAO RP2040,
please change 0 to A0
#define PIXEL_COUNT 30 // The number of LEDs on the light strip
int readValue = 0; // Define variable readValue to store light values
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
// Declare the light strip object and define the data mode
void setup() {
    strip.begin(); // Initialize the light strip and prepare the light strip to output data
    pinMode(LIGHT_PIN, INPUT); // Set the pin of the light sensor to input state
}
void loop() {
    strip.clear(); // Turn off all the beads on the light strip
    rainbow(10); // The light strip shows a rainbow light effect. The number in the parentheses
    represents the speed of the rainbow light rotation. The smaller the number, the faster the rotation
    speed
    readValue = analogRead(A7); // Read the analog value of the light on the A7 pin and store it in the
    readValue variable. Se si utilizza XIAO RP2040, modificare A7 in A3. Se si utilizza XIAO BLE,
    modificare A7 in A5
    if(readValue > 500) { // Conditional judgment, if the light value is greater than 500, then the
    light strip presents a rainbow light effect, otherwise, the light strip is turned off
        rainbow(10);
    } else {
        strip.clear();
        strip.show();
    }
}
// The following is the code for the rainbow light effect, presenting the breathing light effect,
this code can be found in the sample program buttoncycler
void rainbow(int wait) {
    for(long firstPixelHue = 0; firstPixelHue < 3*65536; firstPixelHue += 256) {
        for(int i=0; i<strip.numPixels(); i++) {
            int pixelHue = firstPixelHue + (i * 65536L / strip.numPixels());
            strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
        }
    }
    strip.show(); // The light strip presents a light effect
    delay(wait); // Delay
}
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9\\_StripLight\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L9_StripLight_XIAO_en)

**Passaggio 2:** Collegare l'hardware e caricare il programma. Innanzitutto, collegare la striscia luminosa LED RGB all'interfaccia A0 della scheda di espansione XIAO e collega il sensore di luce all'interfaccia A7, come mostrato nella figura seguente:



**⚠️ Nota** Se si sta utilizzando XIAO BLE, collegare il sensore di luce all'interfaccia I2C della scheda di espansione XIAO. Se si sta utilizzando XIAO RP2040, a causa del numero limitato di pin esposti, si deve collegare il pin SIG del sensore di luce e il pin A3 di XIAO RP2040 con cavi Dupont manualmente.

Quindi, si collega XIAO al computer con un cavo dati, di clicca sul pulsante “Upload” nell’IDE Arduino per caricare il programma sull’hardware. Quando l’area di debug mostra “Upload successful”, puoi coprire il sensore di luce con la mano, quindi rilasciare il sensore di luce e osservare i cambiamenti nella striscia luminosa. Notare che poiché ci vuole un certo lasso di tempo prima che la striscia luminosa visualizzi gli effetti luminosi, la striscia luminosa non si spegnerà immediatamente quando si copre il sensore di luce.



### 2.3.4 Design Esteriore

Combinando il design del programma della scatola regalo a sorpresa, quando il sensore di luce si trova in un ambiente poco illuminato, la striscia luminosa a LED RGB è spenta e quando il sensore di luce si trova in un ambiente luminoso, la striscia luminosa a LED RGB si accende con luci arcobaleno. Possiamo immaginare che la parte elettronica sia posizionata in una scatola chiusa, che può corrispondere alla funzione implementata dal programma e

può anche soddisfare il posizionamento del regalo. Naturalmente, si possono anche avere altri design.

|                              |                                                                                  |
|------------------------------|----------------------------------------------------------------------------------|
| Nome Prodotto                | Surprise Gift Box                                                                |
| Caratteristiche del Prodotto | Fantastici effetti di luce, fotocontrollo, sorpresa, compleanno                  |
| Funzioni del Prodotto        | Controlla l'illuminazione della striscia luminosa LED RGB con un sensore di luce |
| Aspetto del Prodotto         |                                                                                  |

### Caso di riferimento



## 2.4 Danza Ritmica con un Accelerometro Triassiale

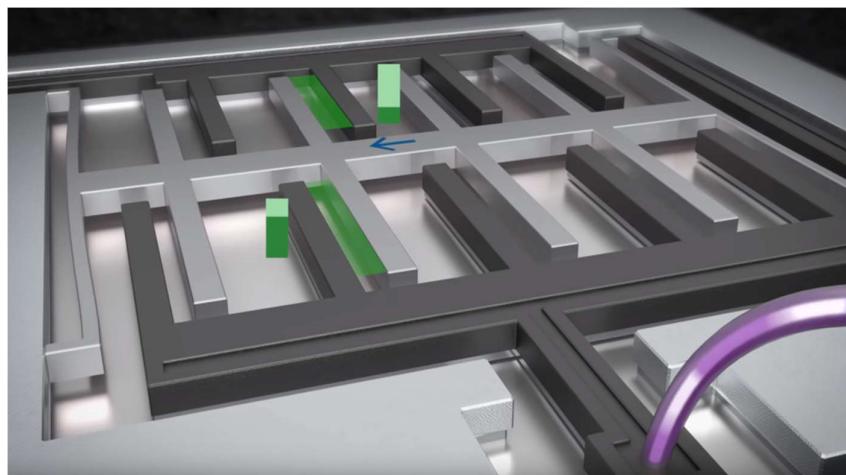
Quando utilizziamo smartphone o tablet, notiamo che lo schermo si capovolge automaticamente a seconda che il dispositivo sia in posizione verticale o orizzontale. Nei giochi di corsa o di volo, telefoni e tablet possono essere utilizzati come volanti, con la rotazione ottenuta inclinando il dispositivo. I droni sempre più popolari, per la maggior parte, ora possono volare in modo sempre più stabile rilevando e controllando l'assetto del velivolo. Tutte queste imprese sono merito dell'accelerometro triassiale. In questa sezione, impareremo a utilizzare la programmazione per recuperare dati da un accelerometro triassiale e utilizzare questi dati per la visualizzazione e il controllo.

### 2.4.1 Conoscenza di Base

#### 2.4.1.1 Accelerometro Triassiale

Dato che le persone prestano sempre più attenzione alla propria salute, un numero crescente di persone sta iniziando a indossare braccialetti, contapassi o utilizzare smartphone per registrare i propri passi, il che è diventato un'abitudine di vita per molti. Quindi, come funziona esattamente un contapassi? La risposta sta in un piccolo chip chiamato accelerometro triassiale che di solito si trova nei telefoni o nei braccialetti moderni ed è il componente chiave nel conteggio dei passi. Un accelerometro è un sensore in grado di misurare l'accelerazione. Di solito è costituito da una massa, uno smorzatore, un elemento elastico, un elemento sensibile e un circuito di regolazione. Durante l'accelerazione, il sensore misura la forza inerziale applicata alla massa e utilizza la seconda legge di Newton per determinare l'accelerazione. A seconda dei diversi elementi sensibili del sensore, gli

accelerometri comuni includono capacitivo, induttivo, di deformazione, piezoresistivo, piezoelettrico e altri.

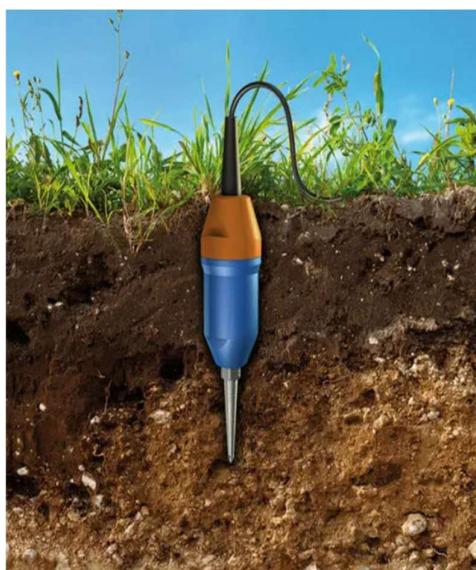


L'accelerometro capacitivo, basato sul principio della capacità, è un tipo comune di accelerometro ed è indispensabile in determinati campi, come airbag di sicurezza, dispositivi mobili come telefoni, ecc. Gli accelerometri capacativi impiegano la tecnologia Micro-Electro-Mechanical Systems (MEMS), che li rende molto economici quando prodotti in serie, garantendo così costi bassi.

### ***Applicazioni degli Accelerometri***

Gli accelerometri possono aiutare i robot a comprendere il loro ambiente. Stanno salendo una collina? O stanno scendendo, o sono caduti? Per le auto o i droni, gli accelerometri possono aiutarli a mantenere l'equilibrio. Oltre alle aree quotidiane come smartphone e braccialetti sanitari, gli accelerometri hanno trovato ampia applicazione anche in altri campi.

- **Accelerometri nella progettazione di sonde sismiche:** I rilevatori sismici sono sensori speciali utilizzati per l'esplorazione geologica e le misure ingegneristiche. Sono sensori che convertono le vibrazioni del terreno in segnali elettrici, trasformando il movimento del terreno causato dalle onde sismiche in segnali elettrici, che vengono poi convertiti in dati binari tramite un convertitore analogico/digitale, organizzati, archiviati ed elaborati.



- **Monitoraggio della tensione di linee ad alta tensione:** Attualmente, il monitoraggio domestico della tensione delle linee adotta principalmente due schemi tecnici principali: acquisizione di immagini video e misura dell'accelerazione del movimento.

Il primo richiede elevata affidabilità e stabilità delle apparecchiature video in condizioni di alta temperatura, alta umidità, freddo intenso, nebbia fitta, tempeste di polvere e altre condizioni meteorologiche, e anche gli effetti delle immagini video riprese saranno influenzati. Pertanto, può servire solo come mezzo di monitoraggio ausiliario e non può analizzare quantitativamente i parametri del movimento della linea. L'utilizzo di un accelerometro per monitorare il “line dancing” consente un'analisi quantitativa della vibrazione su e giù e dell'oscillazione sinistra-destra delle linee di trasmissione in un certo punto, ma può solo misurare l'ampiezza e la frequenza del movimento lineare della linea e non misurare accuratamente il movimento circolare complesso.



- **Sicurezza Automobilistica:** Gli accelerometri sono utilizzati principalmente negli airbag di sicurezza per autoveicoli, nei sistemi di frenata antibloccaggio, nei sistemi di controllo della trazione e in altre funzioni di sicurezza. Nelle applicazioni di sicurezza, la risposta rapida dell'accelerometro è fondamentale. Deve essere determinato rapidamente quando un airbag di sicurezza deve essere attivato, quindi l'accelerometro deve rispondere immediatamente. Un design del sensore che può raggiungere rapidamente uno stato stabile anziché continuare a vibrare può ridurre il tempo di risposta del dispositivo.



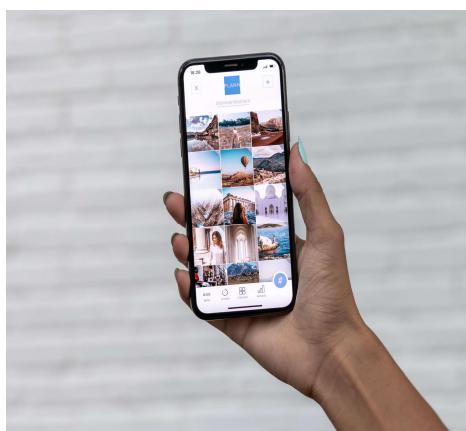
- **Droni:** Gli accelerometri sono anche componenti chiave del controllo, del posizionamento e della stabilità dei droni.



- **Controllo del Gioco:** Gli accelerometri possono rilevare cambiamenti negli angoli di inclinazione verso l'alto, il basso, a sinistra e a destra, quindi diventa semplice controllare le direzioni degli oggetti nei giochi inclinando i dispositivi portatili in avanti e indietro. Molti nuovi controller per console di gioco e controller per dispositivi VR incorporano accelerometri.



- **Ribaltamento Automatico dell'Immagine:** Gli accelerometri rilevano i movimenti di rotazione e le direzioni dei dispositivi portatili, rendendo le immagini visualizzate sempre dritte.



- **Compensazione per i Punti Ciechi del Sistema di Navigazione GPS:** I sistemi GPS determinano la posizione di un oggetto ricevendo segnali da tre satelliti distribuiti a 120 gradi. In circostanze e terreni speciali, come tunnel, edifici densi, aree della giungla, i segnali GPS possono indebolirsi o addirittura perdere completamente, creando punti ciechi. Aggiungendo un accelerometro e utilizzando la navigazione inerziale menzionata in precedenza, possiamo misurare le zone morte del sistema. Integrando l'accelerometro una volta, lo trasformiamo nella variazione di velocità per unità di tempo, misurando così il movimento di un oggetto nella zona morta.



- **Funzione Contapassi:** Gli accelerometri possono rilevare segnali CA e vibrazioni degli oggetti. Quando le persone camminano, producono vibrazioni regolari e l'accelerometro può rilevare l'attraversamento dello zero delle vibrazioni, calcolando così il numero di passi fatti o di corsa e quindi calcolando lo spostamento effettuato dalla persona. Utilizzando alcune formule, possiamo anche calcolare le calorie bruciate.



- **Stabilizzazione dell'Immagine e Stabilizzatori dello Scatto:** La funzione di stabilizzazione dell'immagine utilizza un accelerometro per rilevare l'ampiezza di vibrazione/oscillazione dei dispositivi portatili e, quando l'ampiezza di vibrazione/oscillazione è troppo grande, blocca l'otturatore della fotocamera per garantire che le immagini scattate siano sempre nitide. Lo stabilizzatore dello scatto utilizza un accelerometro per mantenere la stabilità dell'intero dispositivo.



- **Protezione del Disco Rigido:** Rilevando lo stato di caduta libera con un accelerometro, è possibile implementare la protezione necessaria per i dischi rigidi. È risaputo che quando un disco rigido legge dati, lo spazio tra la testina di lettura/scrittura e il piatto è minuscolo e anche piccole vibrazioni esterne possono avere gravi conseguenze per il disco rigido, portando alla perdita di dati. Utilizzando un accelerometro, è possibile rilevare lo stato di caduta libera. Quando viene rilevato lo stato di caduta libera, la testina di lettura/scrittura viene ripristinata per ridurre l'entità del danno al disco rigido.



#### **Accelerometro a tre assi Grove**

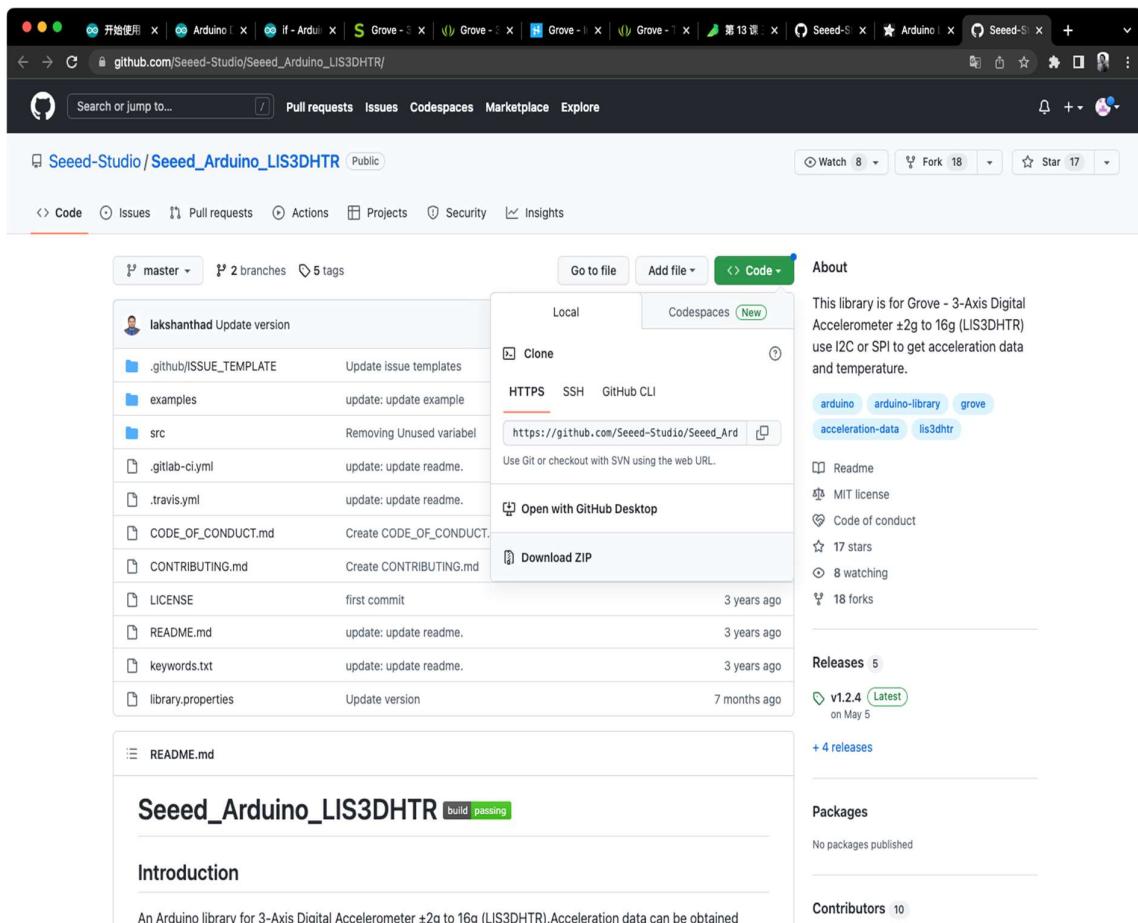
Nel nostro kit, abbiamo un modulo accelerometro a tre assi: [Modulo accelerometro a tre assi Grove](#). Questo piccolo e incredibile accelerometro a tre assi supporta le interfacce I2C, SPI e ADC GPIO, il che significa che si può scegliere qualsiasi modo per connettersi alla scheda di sviluppo. Inoltre, l'accelerometro può anche monitorare la temperatura circostante per correggere gli errori causati da essa.

## 2.4.2 Attività 1: Lettura dei Valori dagli Assi XYZ dell’Accelerometro a Tre Assi

La chiave per usare un accelerometro a tre assi per la creazione di progetti è imparare a leggere i valori degli assi X, Y, Z dell’accelerometro.

### **Aggiungere la libreria `Seeed_Arduino_LIS3DHTR`**

Prima di iniziare a programmare l’accelerometro a tre assi Grove con l’IDE Arduino, è necessario aggiungere la libreria richiesta per il sensore. Inserire l’indirizzo della libreria [https://github.com/Seeed-Studio/Seeed\\_Arduino\\_LIS3DHTR/](https://github.com/Seeed-Studio/Seeed_Arduino_LIS3DHTR/) nella barra degli indirizzi del browser, accedere alla pagina GitHub, cliccare su `Code → Download ZIP` per scaricare il pacchetto di risorse `Seeed_Arduino_LIS3DHTR-master.zip` in locale, come mostrato nella figura seguente.



Aggiungere il pacchetto di risorse scaricato in precedenza `Seeed_Arduino_LIS3DHTR-master.zip` tramite `Sketch → Include Library → Add .ZIP Library` nella barra dei menu, finché non si vede un messaggio di caricamento della libreria riuscito.

### **Apri il File di Esempio**

Allo stesso modo, si può fare riferimento al file della libreria e aprire l’esempio `LIS3DHTR_IIC` tramite il seguente path: **File→Examples→Grove-3-Axis-Digital-Accelerometer-2g-to-16g-LIS3DHTR→LIS3DHTR\_IIC**.

```
// This example use I2C.
#include "LIS3DHTR.h"
#include <Wire.h>
LIS3DHTR<TwoWire> LIS; //IIC
#define WIRE Wire

void setup()
```

```

{
  Serial.begin(115200);
  while (!Serial)
  {
  };
  LIS.begin(WIRE); //IIC init default :0x18
  //LIS.begin(WIRE, 0x19); //IIC init
  LIS.openTemp(); //If ADC3 is used, the temperature detection needs to be turned off.
  // LIS.closeTemp(); //default
  delay(100);

  // LIS.setFullScaleRange(LIS3DHTR_RANGE_2G);
  // LIS.setFullScaleRange(LIS3DHTR_RANGE_4G);
  // LIS.setFullScaleRange(LIS3DHTR_RANGE_8G);
  // LIS.setFullScaleRange(LIS3DHTR_RANGE_16G);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1HZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_10HZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_25HZ);
  LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_100HZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_200HZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_1_6KHZ);
  // LIS.setOutputDataRate(LIS3DHTR_DATARATE_5KHZ);
  LIS.setHighSolution(true); //High solution enable
}
void loop()
{
  if (!LIS)
  {
    Serial.println("LIS3DHTR didn't connect.");
    while (1)
    ;
    return;
  }

  //3 axis
  // Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print(" ");
  // Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print(" ");
  // Serial.print("z:"); Serial.println(LIS.getAccelerationZ());
  //ADC
  // Serial.print("adc1:"); Serial.println(LIS.readbitADC1());
  // Serial.print("adc2:"); Serial.println(LIS.readbitADC2());
  // Serial.print("adc3:"); Serial.println(LIS.readbitADC3());

  //temperature
  Serial.print("temp:");
  Serial.println(LIS.getTemperature());
  delay(500);
}

```

Il programma di esempio può leggere i valori degli assi X, Y, Z dell'accelerometro a tre assi e l'output tramite il monitor seriale. Il programma di esempio ci fornisce diverse scelte di impostazione utilizzando il metodo di commento “//”, ma è necessario selezionare manualmente le parti richieste, come segue: `LIS.begin(WIRE)` : inizializza i valori predefiniti, è possibile scegliere tra 0x18 e 0x19, è necessario scegliere `LIS.begin(WIRE,0x19)` .  
`LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ)` : La frequenza di output dell'accelerometro ha più scelte, scegliere 50 Hz. L'accelerometro a tre assi può anche monitorare la temperatura ambiente, temporaneamente non abbiamo bisogno di eliminare il codice correlato, il programma completo è il seguente:

```

// This example shows the 3-axis acceleration.
#include "LIS3DHTR.h" // Declare library
#include <Wire.h>
LIS3DHTR<TwoWire> LIS;
#define WIRE Wire // Initialize the module above using hardware I2C

void setup()
{
  Serial.begin(9600);
  while (!Serial) { }; // If you can't open the serial monitor, the code will stop here
  LIS.begin(WIRE, 0x19); // Initialize I2C with default value
  delay(100);

  LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ); // Set the accelerometer's output rate to 50Hz.
}

void loop()
{
  if (!LIS) {
    Serial.println("LIS3DHTR didn't connect.");
  }
}

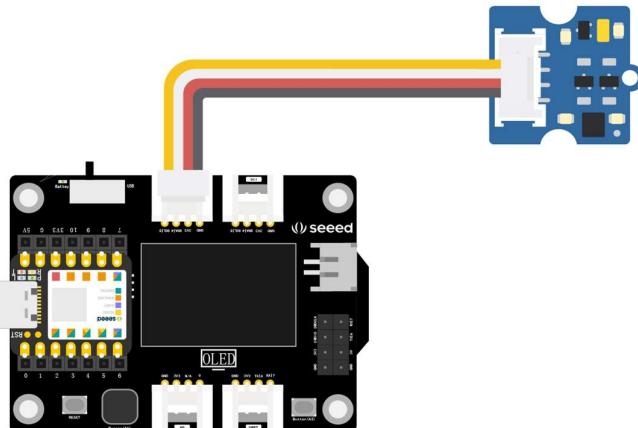
```

```
    while (1);
    return;
}

// Read the values of the X, Y, Z axes from the sensor, and display them on the serial monitor
Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print("  ");
Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print("  ");
Serial.print("z:"); Serial.println(LIS.getAccelerationZ());
delay(500);
}
```

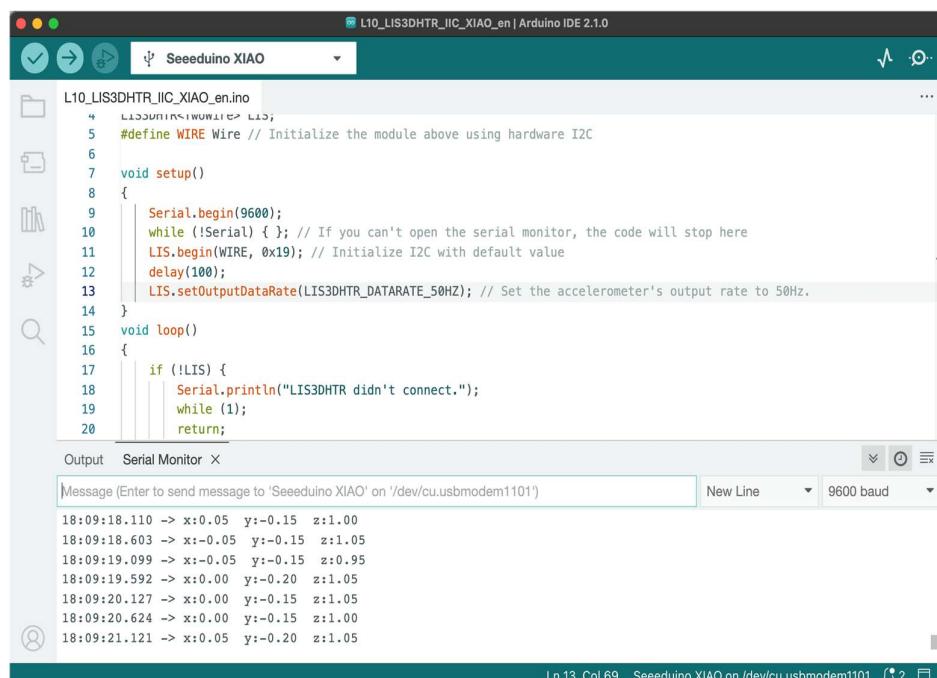
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L10\\_LIS3DHTR\\_IIC\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L10_LIS3DHTR_IIC_XIAO_en)

Poi, collegare l'accelerometro a tre assi all'interfaccia I2C. Ci sono due interfacce I2C sulla scheda di espansione XIAO, come mostrato nell'immagine qui sotto:



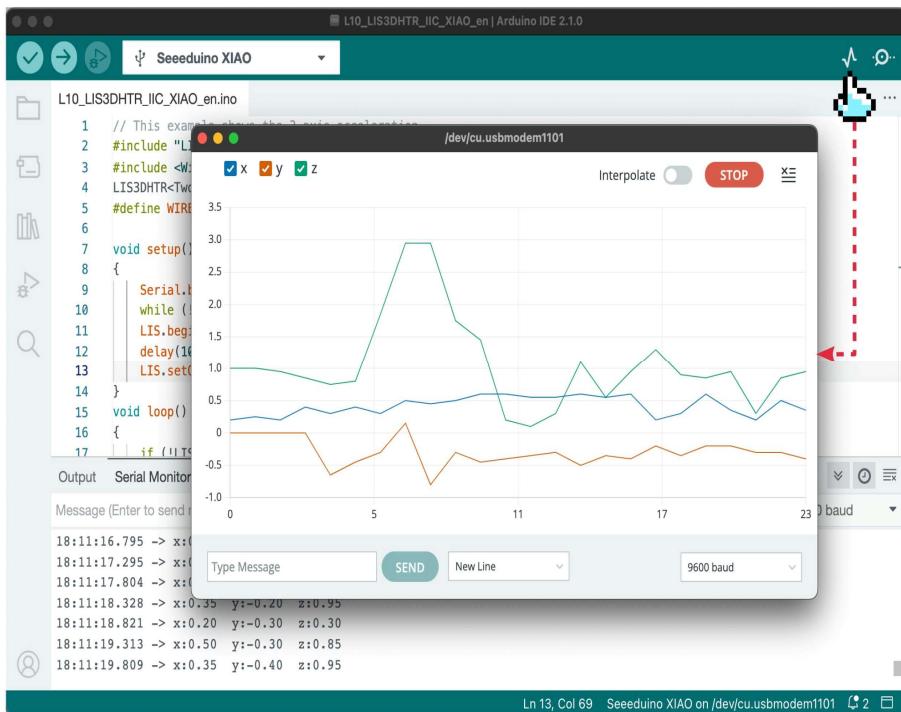
## **Monitorare le Modifiche dei Dati Tramite Serial Monitor**

Collegare XIAO al computer con un cavo dati, caricare il programma, attendere che il programma venga caricato correttamente, quindi aprire il monitor seriale. Spostare l'accelerometro a tre assi nella direzione degli assi X, Y, Z e osservare le modifiche nelle letture.



#### **Monitorare le Modifiche dei Dati con il Plotter Seriale**

Il modo numerico di presentare le modifiche nei valori a 3 assi dell'accelerometro non è molto intuitivo. È possibile aprire il plotter seriale, come mostrato nell'immagine sottostante.



## 2.4.3 Progetto di Produzione: Danza Ritmica

### Descrizione del Progetto

Possiamo aggiungere una striscia LED RGB nel progetto per ottenere fantastici effetti di luce. L'accelerometro a tre assi viene utilizzato per rilevare il movimento e diversi effetti di luce vengono attivati in base a diversi valori sugli assi X, Y, Z dell'accelerometro.

### Scrittura del Programma

Per controllare la striscia LED RGB per modificare gli effetti di luce tramite l'accelerometro a tre assi, seguire questi passaggi:

- Dichiarare i file di libreria che devono essere richiamati, definire il pin della striscia e la quantità di LED.
- Inizializzare l'accelerometro a tre assi e la striscia.
- Impostare l'effetto di luce della striscia su rosso, verde e blu lampeggiante, impostare il giudizio condizionale e controllare la modifica tramite diversi intervalli di valore sugli assi X, Y, Z dell'accelerometro a tre assi.

### Attività: Controllare la Striscia LED RGB per Modificare gli Effetti di Luce

#### Tramite Accelerometro a Tre Assi

**Passaggio 1:** Dichiarare i file di libreria che devono essere richiamati, definire il pin della striscia e il numero di LED.

```

#include "LIS3DHTR.h" // Declare the library file of the three-axis accelerometer
#include <Adafruit_NeoPixel.h> // Declare the strip's library file
#ifndef __AVR__
#include <avr/power.h>
#endif
// Below are to initialize the module using software I2C or hardware I2C
#ifndef SOFTWAREWIRE
#include <SoftwareWire.h>
SoftwareWire myWire(3, 2);
LIS3DHTR<SoftwareWire> LIS;
#define WIRE myWire
#else
#include <Wire.h>
LIS3DHTR<TwoWire> LIS;
#define WIRE Wire
#endif

```

```
#define PIXEL_PIN 0 // Define the pin of the strip, if you use XIAO RP2040/XIAO ESP32, please modify
0 to A0
#define PIXEL_COUNT 30 // Define the number of LEDs in the strip as 30
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Declare the strip object,
set the data type
```

**Passaggio 2:** Inizializzare l'accelerometro a tre assi e la striscia. Qui, si deve inizializzare l'accelerometro e impostare la frequenza a 50 Hz.

```
void setup() {
    Serial.begin(9600); // Initialize the serial monitor
    while (!Serial) {} // If the serial monitor isn't opened, the code will stop here, so please
open the serial monitor
    LIS.begin(WIRE, 0x19); // Initialize I2C
    delay(100);

    LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ); // Set the accelerometer's output rate to 50Hz
    strip.begin(); // Start the strip
    strip.show(); // Display the strip
}
```

**Passaggio 3:** Impostare gli effetti di luce in modo che lampeggino rispettivamente in rosso, verde e blu. Le condizioni vengono utilizzate per modificare il colore della striscia luminosa in base alle diverse letture sugli assi X, Y e Z dell'accelerometro a 3 assi. Queste letture sono visualizzabili tramite il monitor seriale. Osservando la variazione dei valori quando l'accelerometro viene spostato lungo gli assi X, Y e Z, possiamo determinare le impostazioni appropriate per la striscia luminosa. Poiché le letture possono talvolta essere negative, prendiamo il valore assoluto delle letture. La funzione `abs()` è utilizzabile per ottenere il valore assoluto, ad esempio, `abs(LIS.getAccelerationX())` fornirebbe il valore assoluto della lettura sull'asse X.

```
void loop() {
    if (!LIS) { // Check if the 3-axis accelerometer is connected properly
        Serial.println("LIS3DHTR didn't connect.");
        while (1);
        return;
    }

    if ((abs(LIS.getAccelerationX()) > 0.2)) {
        theaterChase(strip.Color(127, 0, 0), 50); // The light strip turns red
    }

    if ((abs(LIS.getAccelerationY()) > 0.2)) {
        theaterChase(strip.Color(0, 127, 0), 50); // The light strip turns green
    }

    if ((abs(LIS.getAccelerationZ()) > 1.0)) {
        theaterChase(strip.Color(0, 0, 127), 50); // The light strip turns blue
    }

    else {
        strip.clear();
        strip.show();
    }

    // Read the values of the X, Y, and Z axes from the sensor and display them on the serial
monitor
    Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print(" ");
    Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print(" ");
    Serial.print("z:"); Serial.println(LIS.getAccelerationZ());

    delay(500);
}

// Set theaterChase for flashing light effects
void theaterChase(uint32_t color, int wait) {
    for(int a=0; a<10; a++) {
        for(int b=0; b<3; b++) {
            strip.clear();
            for(int c=b; c<strip.numPixels(); c += 3) {
                strip.setPixelColor(c, color);
            }
            strip.show();
            delay(wait);
        }
    }
}
```

```

    }
}
```

### Il programma completo è il seguente:

```

#include "LIS3DHTR.h">// Declare the library file for the 3-axis accelerometer
#include <Adafruit_NeoPixel.h>// Declare the library file for the light strip
#ifndef _AVR_
#include <avr/power.h>
#endif
// The following is to initialize the module using software I2C or hardware I2C
#ifndef SOFTWAREWIRE
#include <SoftwareWire.h>
SoftwareWire myWire(3, 2);
LIS3DHTR<SoftwareWire> LIS;
#define WIRE myWire
#else
#include <Wire.h>
LIS3DHTR<TwoWire> LIS;
#define WIRE Wire
#endif

#define PIXEL_PIN 0 // Define the pin of the light strip, if you are using XIAO RP2040/XIAO ESP32,
please change 0 to A0
#define PIXEL_COUNT 30 // Define the number of LEDs on the light strip as 30
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800); // Declare the light strip
object and set the data type

void setup() {
    Serial.begin(9600); // Initialize the serial monitor
    while (!Serial) {}// If you do not open the serial monitor, the code will stop here, so please
open the serial monitor
    LIS.begin(WIRE, 0x19); // IIC initialization
    delay(100);

    LIS.setOutputDataRate(LIS3DHTR_DATARATE_50HZ); // Set the output rate of the accelerometer to
50Hz
    strip.begin(); // The light strip starts working
    strip.show(); // The light strip displays
}

void loop() {
    if (!LIS) { // Check if the 3-axis accelerometer is connected correctly
        Serial.println("LIS3DHTR didn't connect.");
        while (1);
        return;
    }

    if ((abs(LIS.getAccelerationX()) > 0.2)) {
        theaterChase(strip.Color(127, 0, 0), 50); // The light strip turns red
    }

    if ((abs(LIS.getAccelerationY()) > 0.2)) {
        theaterChase(strip.Color(0, 127, 0), 50); // The light strip turns green
    }

    if ((abs(LIS.getAccelerationZ()) > 1.0)) {
        theaterChase(strip.Color(0, 0, 127), 50); // The light strip turns blue
    }

    else
    {
        strip.clear();
        strip.show();
    }

    // Read the values of the X, Y, and Z axes from the sensor and display them on the serial
monitor
    Serial.print("x:"); Serial.print(LIS.getAccelerationX()); Serial.print(" ");
    Serial.print("y:"); Serial.print(LIS.getAccelerationY()); Serial.print(" ");
    Serial.print("z:"); Serial.println(LIS.getAccelerationZ());

    delay(500);
}

// Set theaterChase for flashing light effects
void theaterChase(uint32_t color, int wait) {
    for(int a=0; a<10; a++) {
        for(int b=0; b<3; b++) {
            strip.clear();
            for(int c=b; c<strip.numPixels(); c += 3) {
                strip.setPixelColor(c, color);
            }
        }
    }
}

```

```

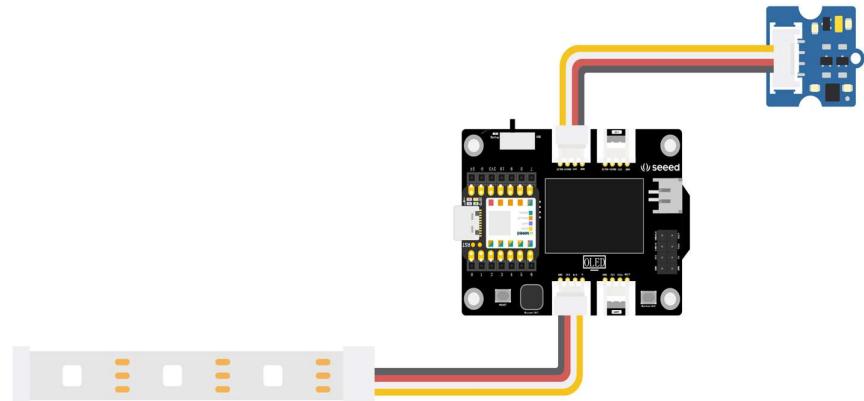
        strip.show();
        delay(wait);
    }

}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L10\\_MovementRGBLED\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L10_MovementRGBLED_XIAO_en)

**Passaggio 4:** Collegare l'hardware e caricare il programma. Per prima cosa, collegare la striscia luminosa a LED RGB all'interfaccia A0/D0 della scheda di espansione XIAO e l'accelerometro a tre assi all'interfaccia I2C, come mostrato nella figura:



Usare un cavo dati per collegare XIAO al computer, cliccare sul pulsante “Upload” in Arduino IDE e caricare il programma sull'hardware. Una volta che l'area di debug mostra “Upload Successful”, si può aprire il monitor seriale e provare a far oscillare l'accelerometro a tre assi a sinistra, a destra, in alto e in basso per sentire i cambiamenti dell'effetto luminoso della striscia luminosa.

#### 2.4.4 Design Esteriore

Si immagini quanto sarebbe bello se ci fossero luci che lampeggiano con i passi di danza mentre si muovono appassionatamente le braccia. È da qui che nasce l'ispirazione per Rhythm Dance [Danza Ritmica]. Può essere abbinato a vestiti o accessori per creare uno stile indossabile.

|                              |                                                                                                                                                                 |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nome Prodotto                | Rhythm Dance                                                                                                                                                    |
| Caratteristiche del Prodotto | Indossabile, Effetti di luce cool, Rilevamento della postura                                                                                                    |
| Funzioni del Prodotto        | La striscia luminosa a LED RGB visualizza diversi effetti di luce in base ai valori rilevati dall'accelerometro a tre assi                                      |
| Aspetto del Prodotto         | (Ad esempio: Lo strato impermeabile all'esterno della striscia luminosa a LED RGB può essere rimosso e può essere cucito insieme a vestiti o una cintura, ecc.) |

### Riferimento per la custodia



## 3.1 Porta con Telecomando Intelligente

Nella vita di tutti i giorni, la privacy e la sicurezza sono questioni di grande preoccupazione per tutti. Negli ultimi anni, le porte delle aree residenziali sono diventate sempre più intelligenti, accessibili solo tramite chiavi elettroniche o password, impedendo agli estranei di entrare. Nelle aree pubbliche o negli ingressi dei parcheggi, avere un telecomando intelligente per azionare le porte potrebbe facilitare il lavoro del personale di sicurezza. Una semplice porta con telecomando intelligente può essere implementata utilizzando un trasmettitore a infrarossi e un ricevitore a infrarossi, che invia e riceve segnali a infrarossi per aprire e chiudere la porta. In questa sezione, costruiremo una porta con telecomando intelligente.

### 3.1.1 Conoscenza di Base

#### 3.1.1.1 Ricevitori e Trasmettitori a Infrarossi

Un ricevitore a infrarossi viene utilizzato per ricevere segnali a infrarossi e anche per il rilevamento del telecomando. Il ricevitore a infrarossi ha un rilevatore a infrarossi per captare la luce a infrarossi emessa dal trasmettitore a infrarossi. Il [Grove - IR Infrared Receiver Module](#) ha una portata di 10 metri; i segnali non possono essere ricevuti oltre questa portata effettiva. In genere, il ricevitore a infrarossi e il trasmettitore a infrarossi funzionano insieme.

#### Grove - IR Infrared Receiver Module

Un trasmettitore a infrarossi è un tipo di dispositivo di controllo remoto con una funzione di controllo remoto. Emette luce attraverso un tubo di emissione a infrarossi entro un certo intervallo, ottenendo così funzioni di segnale di controllo. I telecomandi che utilizziamo per controllare TV, condizionatori d'aria e portiere delle auto nella vita quotidiana sono trasmettitori a infrarossi. I comuni trasmettitori a infrarossi includono quelli modulari come il [Grove - Infrared Emitter Module](#), così come i normali telecomandi, ognuno con scenari e metodi di utilizzo corrispondenti. Per la nostra porta con telecomando intelligente, utilizzeremo un telecomando a infrarossi.

#### Grove - Modulo Trasmettitore a Infrarossi

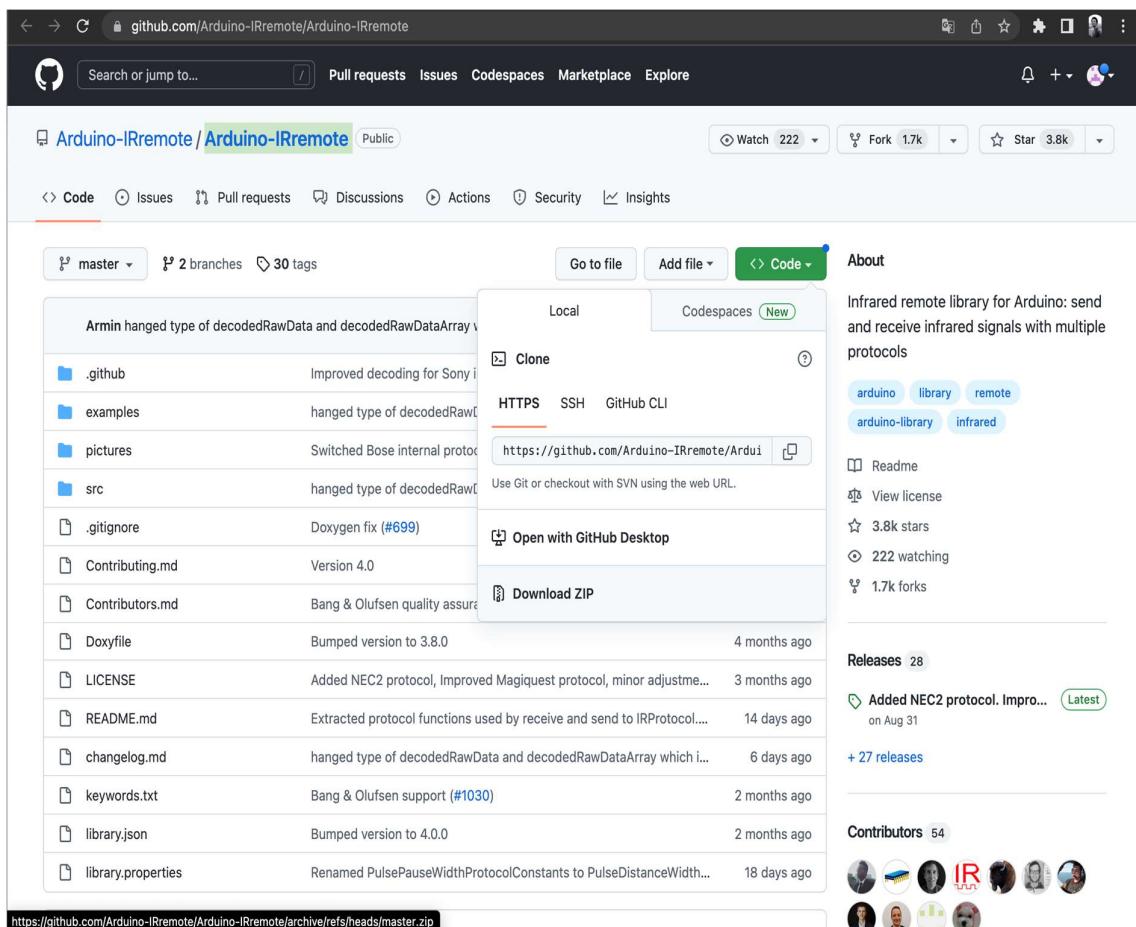
## Grove - Modulo Trasmettitore a Infrarossi

Per spiegarlo in parole poche, il principio della trasmissione e ricezione a infrarossi è che il trasmettitore a infrarossi immette il segnale, lo amplifica e lo invia attraverso il tubo di trasmissione a infrarossi. Il ricevitore a infrarossi riceve quindi questo segnale a infrarossi, lo amplifica e lo riconverte in un segnale elettrico, realizzando così il controllo a infrarossi.

### 3.1.2 Attività 1: Lettura dei Codici Chiave del Telecomando

#### **Aggiungere il File di Libreria Arduino-IRremote**

Prima di iniziare a programmare il ricevitore a infrarossi Grove - IR con l'IDE Arduino, dobbiamo aggiungere i file di libreria necessari. Inserire l'indirizzo del file della libreria <https://github.com/Arduino-IRremote/Arduino-IRremote> nella barra degli indirizzi del browser, si accede alla pagina GitHub e si clicca su **Code → Download ZIP** per scaricare il pacchetto di risorse **Arduino-IRremote-master.zip** sul computer locale, come mostrato nell'immagine qui sotto:



Aggiungere il pacchetto di risorse **Arduino-IRremote-master.zip** appena scaricato tramite **Sketch → Include Library → Add .ZIP Library** nella barra dei menù dell'IDE Arduino finché non si vede un messaggio che indica il caricamento riuscito della libreria.

#### **Aprire il File di Esempio**

Per controllare altri dispositivi tramite il telecomando a infrarossi, ad esempio premendo il tasto sinistro sul mini telecomando a infrarossi per ruotare il servo a sinistra o premendo il tasto destro per ruotare il servo a destra, si deve prima sapere che tipo di codice emetterà ogni tasto sul telecomando. In questo modo, lo si può impostare tramite il programma. Ma come si leggono i codici dei diversi tasti sul telecomando? Si può usare la libreria **IRremote** e

aprire l'esempio `IRrecvDemo` tramite il seguente path: \*\*File Examples IRremote ReceiveDemo\*\*. Questo programma di esempio può leggere i codici dei tasti del telecomando, ma alcuni parametri devono essere modificati: \*\*int RECV\_PIN = 7\*\*, modifica il numero in base al pin di connessione hardware. Abbiamo collegato il ricevitore a infrarossi al pin 7. Successivamente, selezioniamo il codice utile. Dobbiamo solo definire il file header e la parte che legge i codici dei tasti del telecomando. Dopo la riduzione, il programma è il seguente:

```
#include <Arduino.h>
#include <IRremote.h>

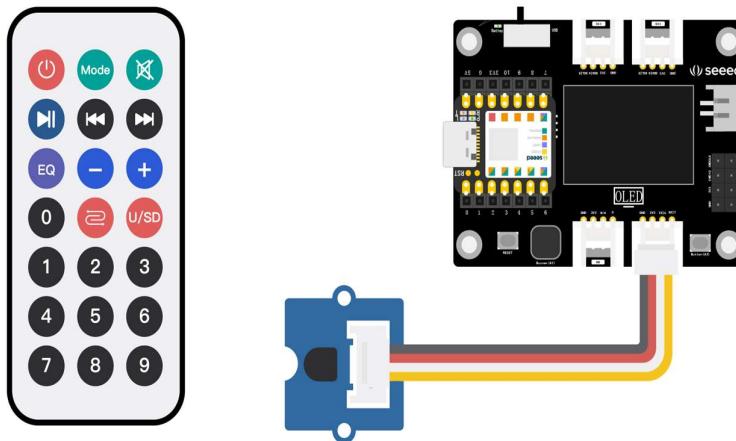
const byte IR_RECEIVE_PIN=7; // The infrared receiver is connected to pin 7. If you are using XIAO RP2040/XIAO ESP32, please change 7 to A0

void setup() {
    Serial.begin(115200);
    Serial.println(F("Enabling IRin"));
    IrReceiver.begin(IR_RECEIVE_PIN,ENABLE_LED_FEEDBACK); // Start infrared decoding
    Serial.print(F("Ready to receive IR signals at pin "));
    Serial.println(IR_RECEIVE_PIN);
    delay(1000);
}

void loop() {
    if (IrReceiver.decode()) // Decode successfully, receive a set of infrared signals
    {
        Serial.println(IrReceiver.decodedIRData.command, HEX); // Output infrared decoding result (hexadecimal)
        Serial.println(IrReceiver.decodedIRData.command); // Output infrared decoding result (octal)
        IrReceiver.resume(); // Receive the next set of values
    }
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L11\\_IRrecvDemo\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L11_IRrecvDemo_en)

Il modulo ricevitore a infrarossi è collegato alla porta 7, come mostrato nella figura seguente:



**⚠️ Nota:** Se si utilizza XIAO RP2040/XIAO ESP32, modificare 7 in A0.

Dopo aver caricato il codice, aprire il monitor seriale, puntare il telecomando verso il componente nero del ricevitore a infrarossi a distanza ravvicinata, premere un tasto qualsiasi e osservare i caratteri emessi dal monitor seriale. Il codice esadecimale appare nella prima riga e quello ottale appare nella seconda riga. Le due righe formano un gruppo, che rappresenta un tasto. Notare che se si preme il tasto troppo a lungo, apparirà "FFFFFF" e questa riga di codice e il codice numerico sottostante non sono validi.

```

L11_IRecvDemo_en.ino
1 #include <Arduino.h>
2 #include <IRremote.h>
3
4 const byte IR_RECEIVE_PIN=7; // The infrared receiver is connected to pin 7. If you are using XIAO RP2040/XIAO ESP32,
5
6 void setup() {
7     Serial.begin(115200);
8     Serial.println(F("Enabling IRin"));
9     IrReceiver.begin(IR_RECEIVE_PIN,ENABLE_LED_FEEDBACK); // Start infrared decoding
10    Serial.print(F("Ready to receive IR signals at pin "));
11    Serial.println(IR_RECEIVE_PIN);
12    delay(1000);
13 }
14
15 void loop() {
16     if (IrReceiver.decode()) // Decode successfully, receive a set of infrared signals
17 }

```

Output    Serial Monitor ×

Message (Enter to send message to 'Seeeduino XIAO' on '/dev/cu.usbmodem1101')  
New Line    9600 baud

09:33:31.023 -> 25  
09:33:31.121 -> 19  
09:33:31.121 -> 25  
09:33:34.092 -> D  
09:33:34.092 -> 13  
09:33:34.126 -> D  
09:33:34.126 -> 13

Ln 5, Col 1 Seeeduino XIAO on /dev/cu.usbmodem1101 4 3

⚠ Nota: Telecomandi diversi possono fornire valori diversi.

### 3.1.3 Creazione del Progetto: Smart Remote Door

#### Descrizione del Progetto

Come possiamo ricreare una porta telecomandata intelligente? Con un telecomando e un ricevitore a infrarossi, il passo successivo è controllare l'apertura e la chiusura della porta. Ricordate come funzionano le porte telecomandate nella nostra vita? Quando si preme il telecomando, la porta si apre lentamente. Quando si apre a una certa angolazione, si chiude lentamente. Possiamo usare un servo per controllare la rotazione della porta. Quando si chiude la porta, il servo ruota da 90° a 0°. Quando si apre la porta, il servo ruota da 0° a 90°. Trasmettendo i segnali per aprire e chiudere la porta con un telecomando, possiamo implementare la funzione di una porta telecomandata intelligente.

#### Scrittura del Programma

Per controllare la rotazione del servo con un telecomando a infrarossi, è necessario seguire questi passaggi:

- Dichiarare la libreria `IRremote` e la libreria `Servo` da chiamare e definire le variabili.
- Inizializzare i file della libreria, inizializzare il servo.
- Leggere il risultato della decodifica a infrarossi e controllare la rotazione del servo in base alle istruzioni a sinistra e a destra.

### Attività 2: Controllare la Rotazione del Servo con un Telecomando a Infrarossi

#### Passaggio 1: Dichiarare la libreria `IRremote` e la libreria `Servo` da chiamare e definire le variabili.

```
#include <IRremote.h>
#include <Servo.h>
```

```
Servo myservo; // Create a servo object myservo to control the servo
```

```

int RECV_PIN = 7; // The infrared receiver is connected to pin 7. If you are using XIAO RP2040/XIAO
ESP32, please change 7 to A0
IRrecv irrecv(RECV_PIN); // Define an IRrecv object to receive infrared signals
decode_results results; // Decoding results are placed in results

int pos = 90; // Define pos as 90°

```

## Passaggio 2: Inizializzare i file della libreria, inizializzare il servo.

```

void setup()
{
    Serial.begin(9600);
    Serial.println("Enabling IRIn");
    irrecv.enableIRIn();
    myservo.attach(5); // Connect the servo on pin 5 to myservo. If you are using XIAO RP2040/XIAO
ESP32, please change 5 to D5
}

```

## Passaggio 3: Leggere il risultato della decodifica a infrarossi e controllare la rotazione del servo secondo le istruzioni a sinistra e a destra. Se ci sono domande sul programma, si può fare riferimento alla sezione commenti.

**⚠️ Nota:** Nell'esempio, il valore del segnale a infrarossi del tasto destro **16761405** e il valore del segnale a infrarossi del tasto sinistro **16712445** devono essere sostituiti dai valori ottenuti dall'esempio "Read Remote Control Key Code" utilizzando il telecomando in mano. Altrimenti, non ci sarà alcuna risposta dopo aver premuto il tasto.

```

void loop() {
    if (irrecv.decode(&results)) { // If decoding is successful, a set of infrared signals is
received
        if (results.value == 16761405) { // If the received signal is 16761405 (right key)
            for (pos; pos <= 89; pos += 1) { // Then the servo is incremented from 0° to 90° in
sequence
                myservo.write(pos); // Write the rotation angle value to the servo pin
                delay(40);
                // The following is to interrupt the above instruction and exit the loop
                if (irrecv.decode(&results)) {
                    irrecv.resume();
                    if (results.value == 16712445)
                        break;
                }
            }
        }

        if (results.value == 16712445) { // If the received signal is 16712445 (left key)
            for (pos; pos >= 1; pos -= 1) { // Then the servo is decremented from 90° to 0° in
sequence
                myservo.write(pos); // Write the rotation angle value to the servo pin
                delay(40);
                // The following is to interrupt the above instruction and exit the loop
                if (irrecv.decode(&results)) {
                    irrecv.resume();
                    if (results.value == 16761405)
                        break;
                }
            }
        }

        // Display hexadecimal and octal codes in the serial port
        Serial.println(pos);
        Serial.println(results.value, HEX);
        Serial.println(results.value);
        irrecv.resume();
    }

    delay(100);
}

```

## Dettagli del programma completo:

```

#include <IRremote.h>
#include <Servo.h>

Servo myservo; // Create a servo object myservo to control the servo

```

```

int RECV_PIN = 7; // The infrared receiver is connected to pin 7. If you are using XIAO RP2040/XIAO
ESP32, please change 7 to A0
IRrecv irrecv(RECV_PIN); // Define an IRrecv object to receive infrared signals
decode_results results; // Decoding results are placed in results

int pos = 90; // Define pos as 90°

void setup()
{
    Serial.begin(9600);
    Serial.println("Enabling IRin");
    irrecv.enableIRIn();
    myservo.attach(5); // Connect the servo on pin 5 to myservo. If you are using XIAO RP2040/XIAO
ESP32, please change 5 to D5
}

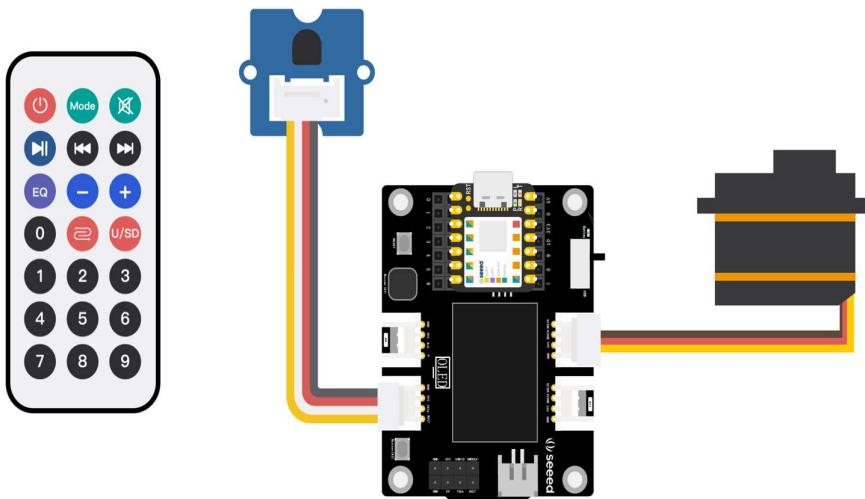
// Note: Left 16712445 Right 16761405, please replace with the key values read from your own remote
control
void loop() {
    if (irrecv.decode(&results)) { // If decoding is successful, a set of infrared signals is
received
        if (results.value == 16761405) { // If the received signal is 16761405 (right key)
            for (pos; pos <= 89; pos += 1) { // Then the servo is incremented from 0° to 90° in
sequence
                myservo.write(pos); // Write the rotation angle value to the servo pin
                delay(40);
                // The following is to interrupt the above instruction and exit the loop
                if (irrecv.decode(&results)) {
                    irrecv.resume();
                    if (results.value == 16712445)
                        break;
                }
            }
        }
        if (results.value == 16712445) { // If the received signal is 16712445 (left key)
            for (pos; pos >= 1; pos -= 1) { // Then the servo is decremented from 90° to 0° in
sequence
                myservo.write(pos); // Write the rotation angle value to the servo pin
                delay(40);
                // The following is to interrupt the above instruction and exit the loop
                if (irrecv.decode(&results)) {
                    irrecv.resume();
                    if (results.value == 16761405)
                        break;
                }
            }
        }
    }
    // Display hexadecimal and octal codes in the serial port
    Serial.println(pos);
    Serial.println(results.value, HEX);
    Serial.println(results.value);
    irrecv.resume();
}

delay(100);
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L11\\_IR\\_Servo\\_ino\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L11_IR_Servo_ino_XIAO_en)

**Passaggio 4:** Collegare l'hardware e caricare il programma. Innanzitutto, si collega il modulo di ricezione a infrarossi alla settima interfaccia della scheda di espansione XIAO e si collega il servo all'interfaccia I2C, come mostrato nella figura seguente:



**⚠️ Nota:** Se si sta utilizzando XIAO RP2040, collegare il modulo di ricezione a infrarossi all'interfaccia `A0`.

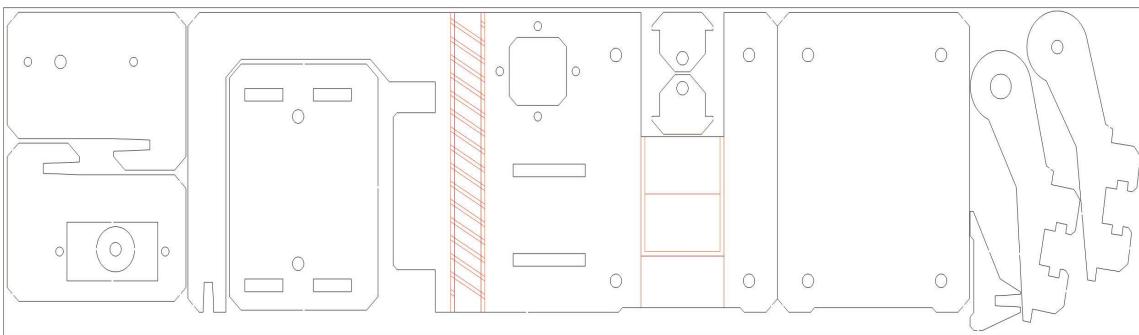
Collegare XIAO al computer con un cavo dati, fai clic sul pulsante “Upload”, caricare il programma sull’hardware e quando l’area di debug mostra “Upload Successful”, aprire il monitor seriale, puntare il telecomando verso il ricevitore a infrarossi, premere il tasto “Sinistra” e il tasto “Destra”, osservare la rotazione del servo e controllare le informazioni di codifica in uscita dal monitor seriale.

### 3.1.4 Design Esteriore

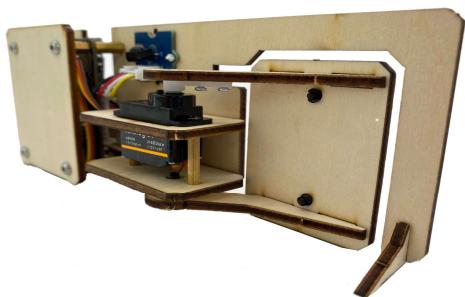
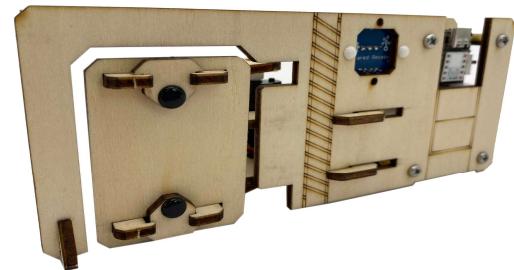
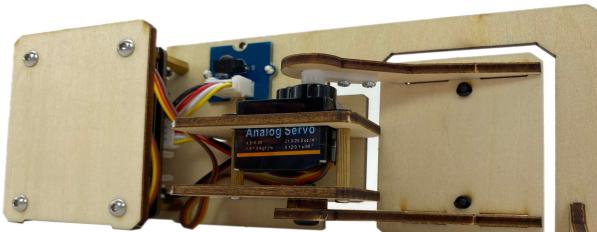
In questa unità, dobbiamo implementare un progetto più completo, combinando le funzioni implementate dal programma, i moduli e l’aspetto della struttura per formare un prototipo. Tornando al progetto della porta con telecomando intelligente, dobbiamo controllare la rotazione del servo tramite il telecomando, simulare l’apertura e la chiusura della porta. Quando creiamo l’aspetto, dobbiamo concentrarci sui seguenti problemi:

- Come combinare il servo e il pannello della porta per far sì che la rotazione del servo azioni la rotazione del pannello della porta.
- Il ricevitore a infrarossi deve essere esposto in una posizione ben visibile, senza alcuna copertura.
- Sebbene il controllo principale, la scheda di espansione e i cavi di collegamento sono coperti per mantenere l’aspetto pulito.
- Come far sì che il lavoro resti fermo.

La figura seguente fornisce un caso di apparenza, che è tagliato al laser dal tiglio, e fornisce file di taglio per riferimento. Se si può usare un software di disegno, lo si può elaborare e progettare in autonomia. Se non si ha una macchina per il taglio laser, si può anche usare carta ondulata, cartoncino, tessuto non tessuto e altri materiali fatti a mano per realizzarlo, il che metterà alla prova l’abilità pratica.



Download dei file per l'uso con un laser cutter [🔗 https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO\\_ADR.dxf](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO_ADR.dxf).



## 3.2 Orologio Intelligente

L'orologio è un oggetto comune nella vita. Anche se vari dispositivi elettronici ora hanno funzioni di cronometraggio e i telefoni cellulari possono sostituire gli orologi come strumenti di cronometraggio, gli orologi sono ancora un oggetto popolare. Non sono solo strumenti di cronometraggio, ma hanno anche funzioni di abbinamento alla moda. Sebbene gli orologi

siano delicati e piccoli, implicano una lavorazione artigianale complessa. Ora con XIAO e la sua scheda di espansione, possiamo realizzarli facilmente.

### 3.2.1 Conoscenza di Base

#### 3.2.1.1 RTC Clock

RTC sta per Real\_Time Clock, che è un circuito integrato utilizzato per visualizzare l'ora, chiamato anche chip di clock RTC. Gli RTC sono ampiamente utilizzati e possiamo trovare RTC in quasi tutti i dispositivi elettronici. Nella scheda di espansione XIAO, c'è un chip di clock RTC, come mostrato nella figura seguente. Possiamo visualizzare la data e l'ora sul display OLED sulla scheda di espansione e può essere alimentato da una batteria a bottone o da una batteria al litio. Anche se ci scolleghiamo, può continuare a tracciare il tempo.

Quando ricolleghiamo l'alimentazione, scopriremo che il tempo è ancora in movimento. Con l'orologio RTC, possiamo realizzare dispositivi di promemoria temporizzati, come l'irrigazione temporizzata, l'alimentazione temporizzata degli animali domestici e così via.

Posizione del chip dell'orologio RTC sulla scheda di espansione XIAO

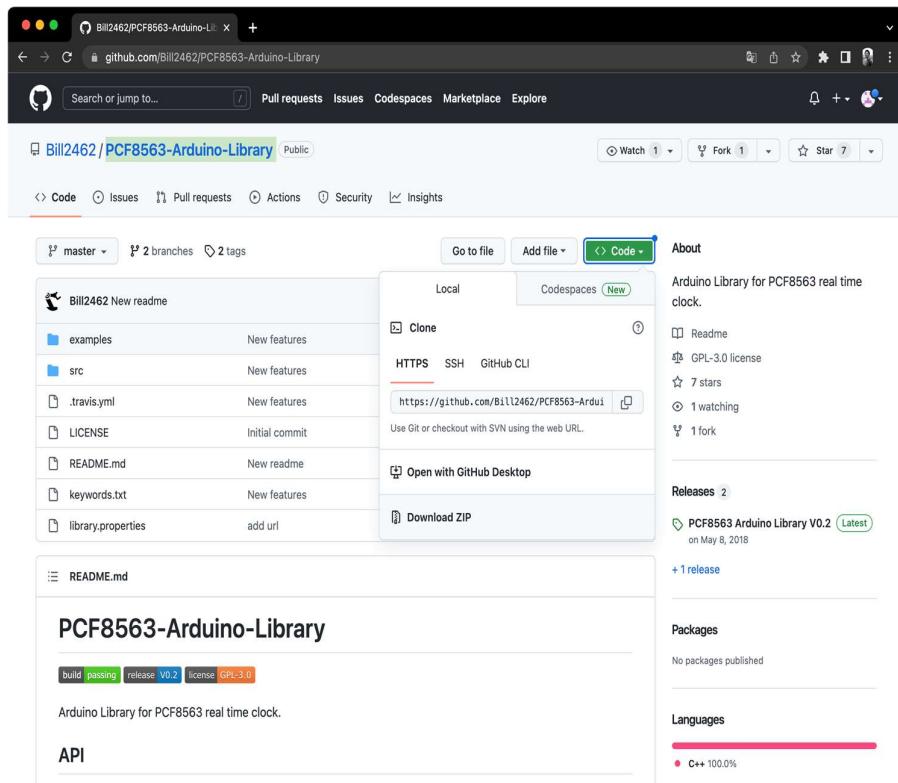
Grove ha anche un modulo RTC: [Grove - DS1307 RTC \(Real Time Clock\) per Arduino](#), come mostrato nella figura sottostante.

Modulo RTC Grove

### 3.2.2 Attività 1: Visualizzazione dell'Orologio RTC nel Monitor Seriale

#### **Aggiunta della Libreria PCF8563-Arduino-Library**

Prima di iniziare a programmare l'RTC sulla scheda di espansione XIAO con Arduino IDE, è necessario aggiungere i file di libreria necessari. Inserire l'indirizzo del file della libreria  <https://github.com/Bill2462/PCF8563-Arduino-Library> nella barra degli indirizzi del browser, accedere alla pagina GitHub, fai clic su `code-Download ZIP` per scaricare il pacchetto di risorse `PCF8563-Arduino-Library-master.zip` in locale, come mostrato nella figura sottostante.



Aggiungere il pacchetto di risorse scaricato in precedenza `PCF8563-Arduino-Library-master.zip` in `Sketch->Include Library->Add .ZIP Library` nella barra dei menu finché non vedi il messaggio di caricamento della libreria.

### **Apertura del File di Esempio**

La creazione di un orologio RTC non ci può essere senza il potente file di libreria. Aprire l'esempio `simple` tramite il seguente path: `File->Examples->PCF8563->simple`. Questo programma di esempio può visualizzare l'orologio RCT tramite il monitor seriale. Dopo aver aperto il programma di esempio, dobbiamo solo modificare la data corrente e l'ora di inizio:

```
#include <PCF8563.h> //Declare library file
PCF8563 pcf;//Define variable pcf

void setup() {
    Serial.begin(9600);
    pcf.init(); //Initialize the clock
    pcf.stopClock(); //Stop the clock

    //Set the current date and time. After setting, it will start timing from this moment

    pcf.setYear(23); //Year
    pcf.setMonth(05); //Month
    pcf.setDay(29); //Day
    pcf.setHour(16); //Hour
    pcf.setMinut(10); //Minute
    pcf.setSecond(0); //Second

    pcf.startClock(); //Clock starts timing
}

void loop() {
    Time nowTime = pcf.getTime(); //Get time

    //Print the current date and time on the serial monitor
    Serial.print(nowTime.day);
    Serial.print("/");
    Serial.print(nowTime.month);
    Serial.print("/");
    Serial.print("20"); // Manually input the set year
    Serial.print(nowTime.year);
    Serial.print("/");
    Serial.print(nowTime.hour);
    Serial.print(":");
    Serial.print(nowTime.minute);
    Serial.print(":");
    Serial.print(nowTime.second);
}
```

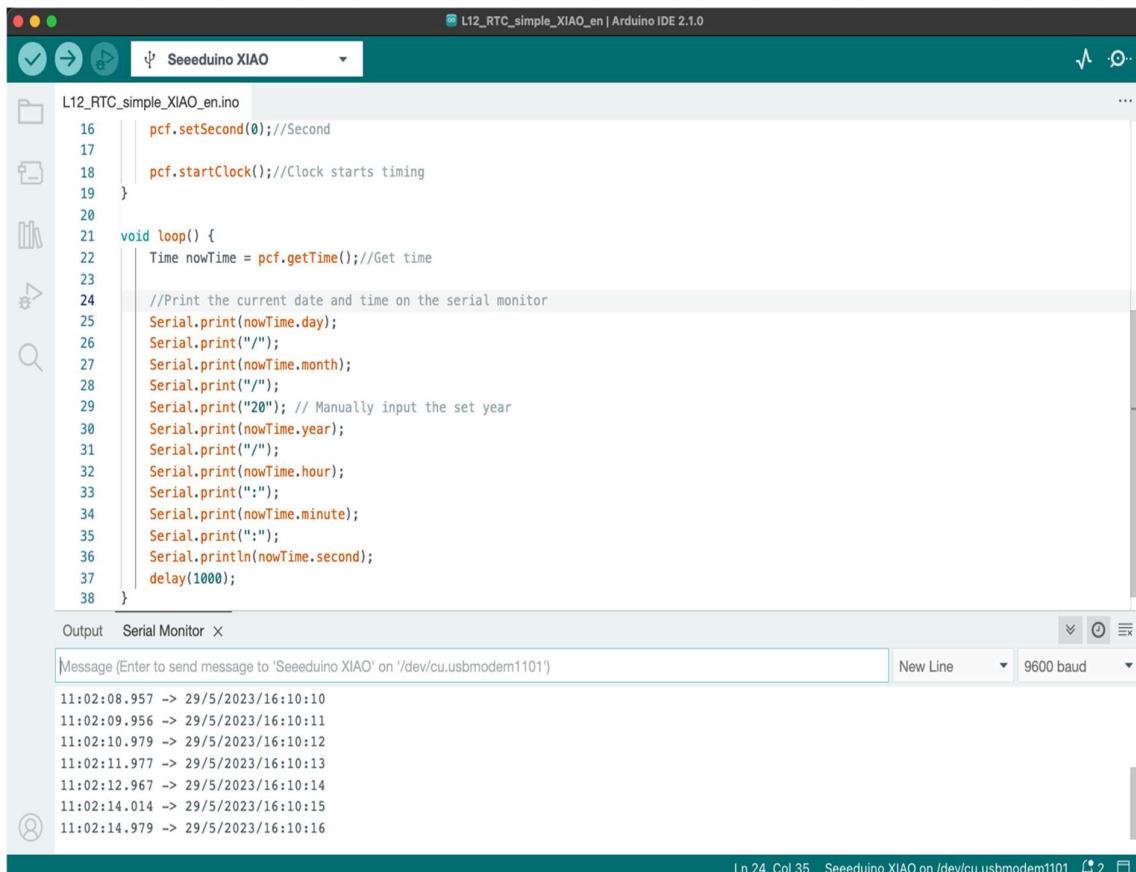
```

    Serial.print(nowTime.minute);
    Serial.print(":");
    Serial.println(nowTime.second);
    delay(1000);
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12\\_RTC\\_simple\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12_RTC_simple_XIAO_en)

Senza collegare altri moduli elettronici, si può cliccare sul pulsante di caricamento del programma. Dopo aver caricato il codice, aprire il monitor seriale e si sarà in grado di vedere l'ora.



### 3.2.3 Project Making: Orologio Smart

#### Descrizione del Progetto

In questa sezione realizzeremo un orologio smart in grado di visualizzare data, ora, temperatura e umidità in tempo reale. Per visualizzare data e ora, abbiamo solo bisogno di XIAO e della scheda di espansione. Per visualizzare temperatura e umidità, dobbiamo aggiungere un sensore di temperatura e umidità.

#### Programmazione

Il programma è composto dai seguenti passaggi:

- Dichiarare le librerie necessarie e definire le variabili.
- Inizializzare le librerie e impostare l'ora corrente.
- Leggere le variabili di temperatura e umidità, ottenere l'ora corrente e visualizzare temperatura, umidità e data/ora sullo schermo OLED.

**⚠ Nota**

- Prima di iniziare a programmare per l’OLED della scheda di espansione XIAO, assicurarsi che la libreria `u8g2_Arduino` sia stata caricata nell’IDE Arduino. Il metodo di caricamento può essere consultato nelle istruzioni nella sezione “Come scaricare e installare le librerie Arduino” della Sezione 1.1.
- Prima di iniziare a programmare per il sensore di temperatura e umidità Grove, assicurati che l’IDE Arduino abbia caricato la libreria `Grove_Temperature_And_Humidity_Sensor`. Il metodo di caricamento può essere consultato nelle istruzioni nella sezione “Aggiunta della libreria `Grove_Temperature_And_Humidity_Sensor`” della Sezione 2.2.

**Attività 2: Visualizzare l’ora corrente e i valori di temperatura/umidità sul display OLED della scheda di espansione XIAO (basata sul sensore DHT20)**

Passaggio 1: dichiarare le librerie necessarie e definire le variabili.

```
#include <Arduino.h>
#include <U8x8lib.h> //use u8x8 library
#include <PCF8563.h> //RTC library
PCF8563 pcf; //define variable pcf
#include <Wire.h>
#include "DHT.h" //DHT library
#define DHTTYPE DHT20 //The type of the temperature and humidity sensor is DHT20
DHT dht(DHTTYPE);
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE); //OLED's constructor, set data type, connect OLED display
```

Passaggio 2: Inizializzare le librerie e impostare l’ora corrente.

```
void setup() {
    Serial.begin(9600);
    u8x8.begin(); //u8x8 starts working
    u8x8.setFlipMode(1);
    Wire.begin();
    pcf.init(); //Initialize the clock
    pcf.stopClock(); //Stop the clock
    //Set the current time and date:
    pcf.setYear(23);
    pcf.setMonth(05);
    pcf.setDay(29);
    pcf.setHour(18);
    pcf.setMinut(53);
    pcf.setSecond(0);
    pcf.startClock(); //The clock starts timing
}
```

Passaggio 3: Leggere le variabili di temperatura e umidità, ottenere l’ora corrente e visualizzare temperatura, umidità e data/ora sullo schermo OLED.

```
void loop() {
    float temp, humi; //Define temperature and humidity variables
    temp = dht.readTemperature(); //Read the temperature value
    humi = dht.readHumidity(); //Read the humidity value
    Time nowTime = pcf.getTime(); //Get the time
    u8x8.setFont(u8x8_font_chroma48medium8_r); //u8x8 font

    //Display the current date, time, temperature, and humidity at different coordinates on the OLED screen.
    u8x8.setCursor(0, 0);
    u8x8.print(nowTime.day);
    u8x8.print("/");
    u8x8.print(nowTime.month);
    u8x8.print("/");
    u8x8.print("20");
    u8x8.print(nowTime.year);
    u8x8.setCursor(0, 1);
    u8x8.print(nowTime.hour);
    u8x8.print(":");
    u8x8.print(nowTime.minute);
    u8x8.print(":");
    u8x8.println(nowTime.second);
    delay(1000);
    u8x8.setCursor(0, 2);
    u8x8.print("Temp:");
    u8x8.print(temp);
    u8x8.print("C");
    u8x8.setCursor(0, 3);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
```

```

    u8x8.refreshDisplay();
    delay(200);
}

```

Per il programma completo, fare riferimento a:

```

#include <Arduino.h>
#include <U8x8lib.h> //use u8x8 library
#include <PCF8563.h> //RTC library
PCF8563 pcf; //define variable pcf
#include <Wire.h>
#include "DHT.h" //DHT library
#define DHTTYPE DHT20 //The type of the temperature and humidity sensor is DHT20
DHT dht(DHTTYPE);
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8 /* reset=*/ U8X8_PIN_NONE); //OLED's constructor, set data
type, connect OLED display

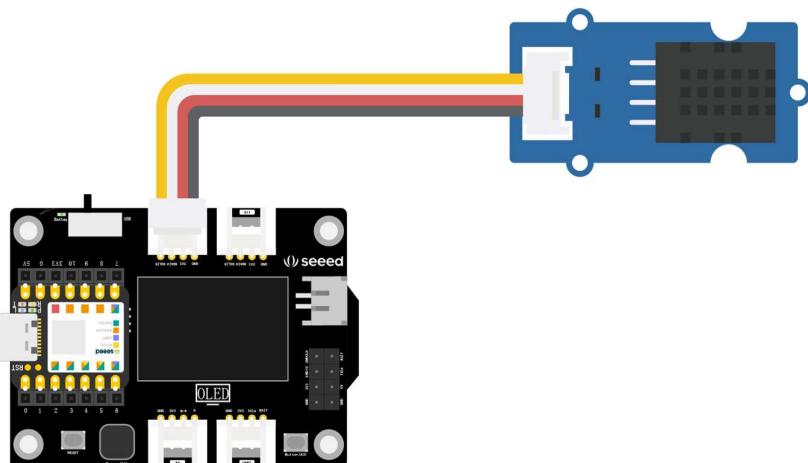
void setup() {
    Serial.begin(9600);
    u8x8.begin(); //u8x8 starts working
    u8x8.setFlipMode(1);
    Wire.begin();
    pcf.init(); //Initialize the clock
    pcf.stopClock(); //Stop the clock
    //Set the current time and date:
    pcf.setYear(23);
    pcf.setMonth(05);
    pcf.setDay(29);
    pcf.setHour(18);
    pcf.setMinut(53);
    pcf.setSecond(0);
    pcf.startClock(); //The clock starts timing
}
void loop() {
    float temp, humi; //Define temperature and humidity variables
    temp = dht.readTemperature(); //Read the temperature value
    humi = dht.readHumidity(); //Read the humidity value
    Time nowTime = pcf.getTime(); //Get the time
    u8x8.setFont(u8x8_font_chroma48medium8_r); //u8x8 font

    //Display the current date, time, temperature, and humidity at different coordinates on the OLED
    screen.
    u8x8.setCursor(0, 0);
    u8x8.print(nowTime.day);
    u8x8.print("/");
    u8x8.print(nowTime.month);
    u8x8.print("/");
    u8x8.print("20");
    u8x8.print(nowTime.year);
    u8x8.setCursor(0, 1);
    u8x8.print(nowTime.hour);
    u8x8.print(":");
    u8x8.print(nowTime.minute);
    u8x8.print(":");
    u8x8.println(nowTime.second);
    delay(1000);
    u8x8.setCursor(0, 2);
    u8x8.print("Temp:");
    u8x8.print(temp);
    u8x8.print("°");
    u8x8.setCursor(0, 3);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
}

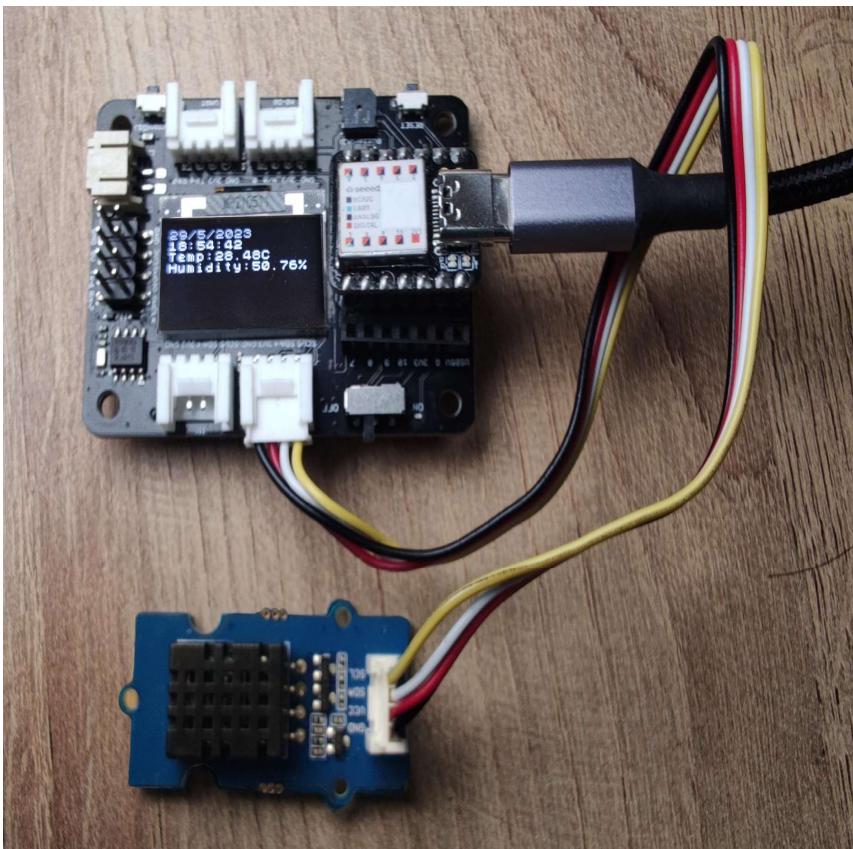
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12\\_SmartWatch\\_DHT20\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12_SmartWatch_DHT20_XIAO_en)

**Passaggio 4:** Collegare l'hardware e caricare il programma. Per prima cosa, collegare il sensore di temperatura e umidità DHT20 all'interfaccia I2C della scheda di espansione XIAO e collega XIAO al computer con un cavo dati, come mostrato nell'immagine:



Cliccare sul pulsante “Upload” nell’IDE Arduino per caricare il programma sull’hardware. Quando l’area di debug mostra “Upload successful”, osservare se il display OLED visualizza correttamente l’ora corrente e inizia a cronometrare, nonché la temperatura e l’umidità in tempo reale.



### **Attività 3: Visualizzare l’ora corrente e i valori di temperatura e umidità sul display OLED della scheda di espansione XIAO (basata sul sensore DHT11)**

Se si utilizza il sensore di temperatura e umidità Grove DHT11 con un involucro blu, parte del codice del programma deve essere modificata come segue: `#define DHTPIN 0`, deve essere modificato in base al numero di pin effettivo a cui è collegato il sensore di temperatura e umidità. `#define DHTTYPE DHT11`, poiché esistono diversi modelli di sensori di temperatura e umidità, è necessario selezionare il modello corretto — DHT11. Il codice di esempio modificato è il seguente:

```
#include <Arduino.h>
#include <U8x8lib.h>
#include <PCF8563.h>
```

```

PCF8563 pcf;
#include <Wire.h>
#include "DHT.h"
#define DHTPIN 0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/ U8X8_PIN_NONE);
//U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock=*/ SCL, /* data=*/ SDA, /* reset=*/ U8X8_PIN_NONE); // OLEDs without Reset of the Display

void setup() {
    Serial.begin(115200);
    u8x8.begin();
    u8x8.setFlipMode(1);
    Wire.begin();
    pcf.init(); //initialize the clock
    pcf.stopClock(); //stop the clock
    pcf.setYear(23); //set year
    pcf.setMonth(05); //set month
    pcf.setDay(29); //set date
    pcf.setHour(18); //set hour
    pcf.setMinut(53); //set minute
    pcf.setSecond(0); //set second
    pcf.startClock(); //start the clock
}

void loop() {
    float temp, humi;
    temp = dht.readTemperature();
    humi = dht.readHumidity();
    Time nowTime = pcf.getTime(); //get current time
    u8x8.setFont(u8x8_font_chroma48medium8_r); // choose a suitable font

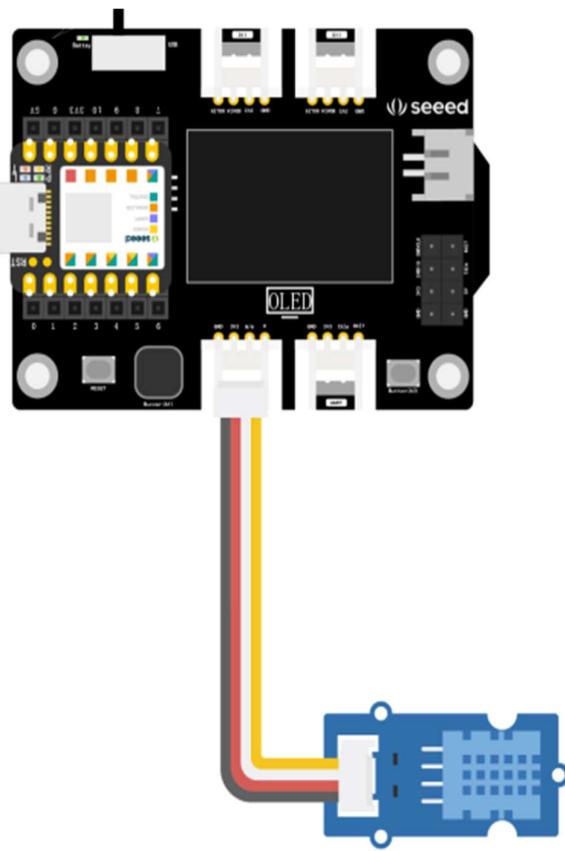
    u8x8.setCursor(0, 0);
    u8x8.print(nowTime.day);
    u8x8.print("/");
    u8x8.print(nowTime.month);
    u8x8.print("/");
    u8x8.print("20");
    u8x8.print(nowTime.year);
    u8x8.setCursor(0, 1);
    u8x8.print(nowTime.hour);
    u8x8.print(":");
    u8x8.print(nowTime.minute);
    u8x8.print(":");
    u8x8.println(nowTime.second);
    delay(1000);
    u8x8.setCursor(0, 2);
    u8x8.print("Temp:");
    u8x8.print(temp);
    u8x8.print("C");
    u8x8.setCursor(0, 3);
    u8x8.print("Humidity:");
    u8x8.print(humi);
    u8x8.print("%");
    u8x8.refreshDisplay();
    delay(200);
}

```

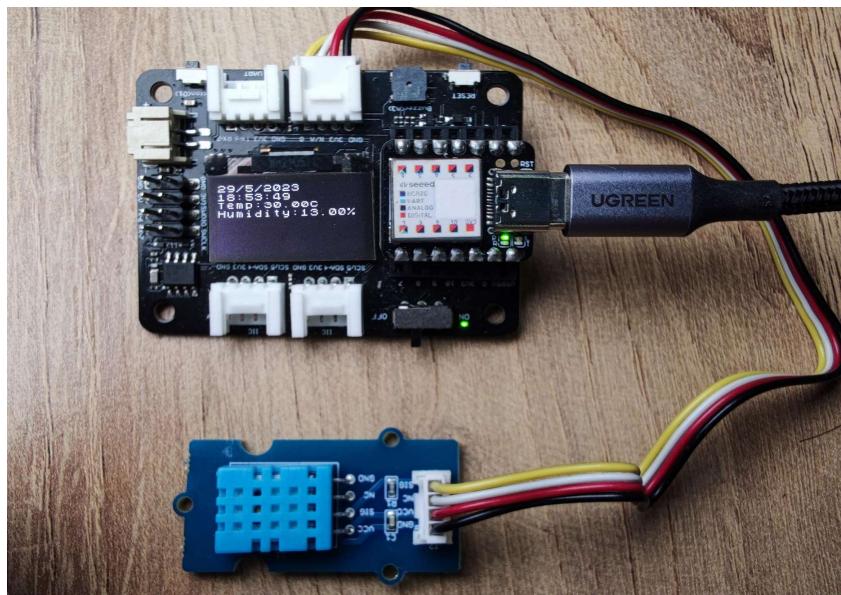
Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12\\_SmartWatch\\_DHT11\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L12_SmartWatch_DHT11_XIAO_en)

---

Dopo aver modificato il codice, collegare prima il sensore di temperatura e umidità DHT11 all’interfaccia **A0** della scheda di espansione XIAO, come mostrato nell’immagine sottostante.



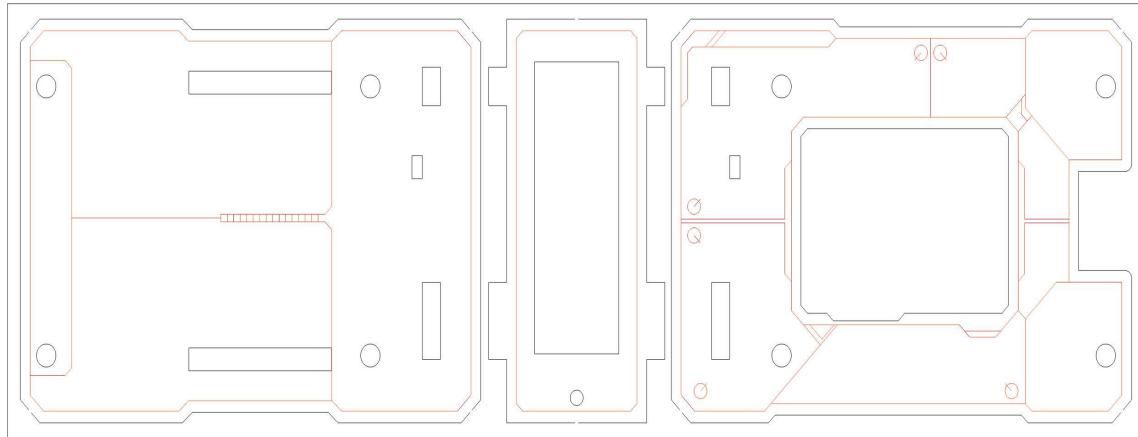
Collegare, poi, la scheda di sviluppo XIAO al computer, caricare il programma di esempio modificato su XIAO tramite Arduino IDE e si dovrebbe essere in grado di vedere le letture di ora, temperatura e umidità sull'OLED della scheda di espansione XIAO. Si può posizionare il sensore di temperatura e umidità in diversi ambienti per osservare i cambiamenti nelle letture di temperatura e umidità.



### 3.2.4 Design Esteriore

Date le sue dimensioni compatte, XIAO è particolarmente adatto per la creazione di dispositivi indossabili. La scheda di espansione incorpora un chip RTC, un cicalino e uno schermo OLED, il che significa che si possono creare una varietà di applicazioni anche senza

aggiungere altri moduli. In questa sezione, abbiamo realizzato uno smartwatch utilizzando il display OLED integrato, il chip RTC e un sensore esterno di temperatura e umidità. Quando creiamo l'aspetto, dobbiamo solo considerare la vestibilità, l'organizzazione dei moduli e dei cavi di collegamento e l'esposizione dello schermo del display OLED. Come mostrato di seguito, forniamo uno stile di orologio indossabile e i file di taglio col laser per esso. Con una semplice installazione, il dispositivo indossabile è pronto.



Download dei file per la macchina per il taglio laser [🔗 https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO\\_X\\_watch.dxf](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO_X_watch.dxf).



## 3.3 Pianoforte ad Aria

Normalmente, quando suoniamo uno strumento musicale, dobbiamo pizzicare le corde o premere i tasti per produrre note musicali. Tuttavia, con i moduli elettronici, suonare la musica può diventare più emozionante. Ad esempio, si può simulare il pianoforte con interruttori a pulsante e persino integrare effetti di luce per musica interattiva. Ma se si usano interruttori a pulsante come tasti del pianoforte, devi integrare molti moduli nel circuito. Esiste un'idea più semplice e unica? Una combinazione di sensore di distanza a ultrasuoni e cicalino passivo può fare proprio questo: rilevare diverse distanze con ultrasuoni per attivare note diverse, proprio come suonare un pianoforte ad aria.

### 3.3.1 Conoscenza di Base

#### 3.3.1.1 Sensore di Distanza a Ultrasuoni Grove

Il [sensore di distanza a ultrasuoni Grove](#) è un modulo di misura della distanza senza contatto. Grazie alla sua forte direttività, le onde ultrasoniche che emette possono percorrere lunghe distanze in un mezzo. I calcoli sono semplici ed è facile da controllare, quindi è spesso utilizzato per le misure della distanza. Quando il sensore di distanza a ultrasuoni funziona, il trasmettitore emette onde ultrasoniche in una determinata direzione. Quando le onde colpiscono un ostacolo, si riflettono. Il ricevitore a ultrasuoni interrompe il cronometraggio non appena riceve le onde riflesse. La distanza effettiva dal punto di emissione all'ostacolo viene calcolata in base alla differenza di tempo tra emissione e ricezione, in modo molto simile all'ecolocalizzazione dei pipistrelli. La gamma di applicazioni delle onde ultrasoniche sta diventando più ampia, comunemente osservata nei sistemi radar inversi, nei sistemi di guida intelligenti, nei sistemi per evitare gli ostacoli dei robot, negli esami ecografici medici e altro ancora.

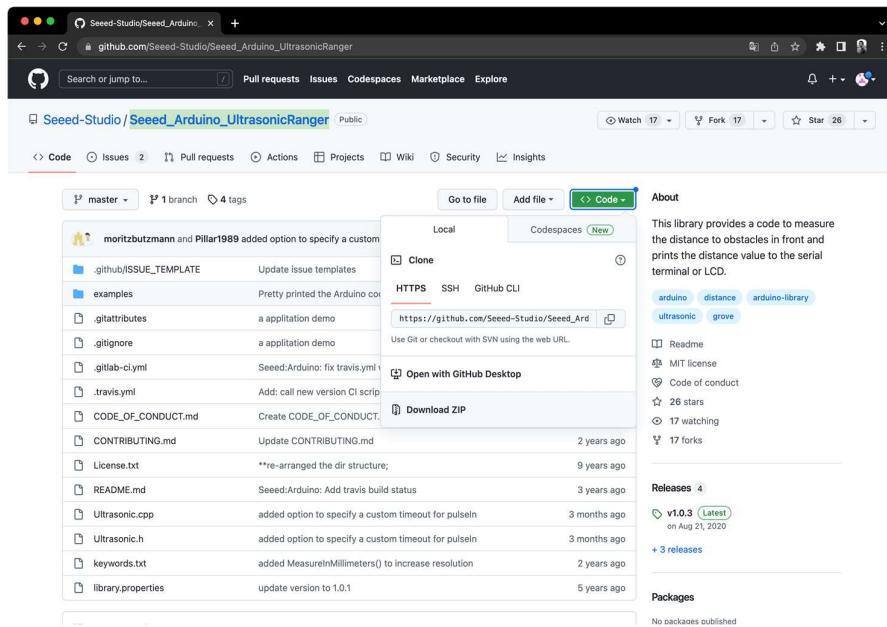
 Nota: Il modulo Grove Ultrasonic Distance Sensor non è incluso nel Seeed Studio XIAO Starter Kit!



#### 3.3.2 Attività 1: Lettura del Valore del Grove Ultrasonic Distance Sensor

##### [Aggiungere la Libreria Seeed\\_Arduino\\_UltrasonicRanger](#)

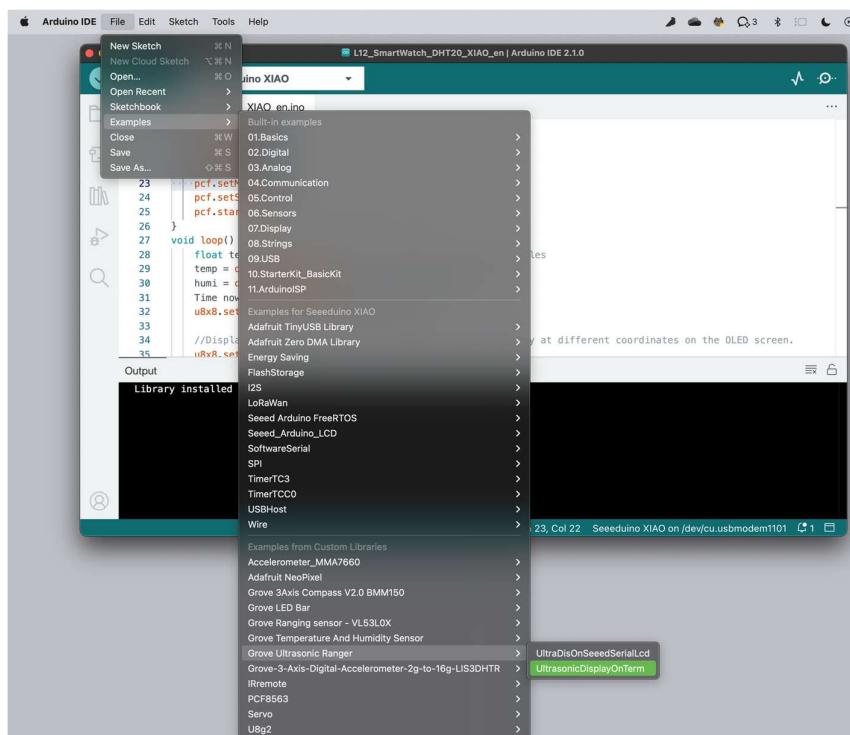
Prima di iniziare a programmare il Grove Ultrasonic Distance Sensor con Arduino IDE, è necessario aggiungere la libreria essenziale per il sensore. Digitare l'indirizzo della libreria [https://github.com/Seeed-Studio/Seeed\\_Arduino\\_UltrasonicRanger](https://github.com/Seeed-Studio/Seeed_Arduino_UltrasonicRanger) nella barra degli indirizzi del browser, accedere alla pagina GitHub, cliccare su `Code > Download ZIP` per scaricare il pacchetto di risorse `seeed_Arduino_UltrasonicRanger-master.zip` sul disco locale, come mostrato di seguito.



Aggiungere il pacchetto di risorse scaricato [Seeed\\_Arduino\\_UltrasonicRanger-master.zip](#) a Sketch->Include Library->Add .ZIP Library dalla barra dei menù finché non si vede un prompt di caricamento riuscito della libreria.

### **Apertura del File di Esempio**

Dopo aver installato correttamente la libreria, un nuovo elemento **Grove Ultrasonic Ranger** verrà aggiunto all'elenco **File->Examples** di Arduino. Da qui, aprire il programma di esempio **ultrasonicDisplayOnTerm**. Questo programma può visualizzare il valore del sensore di distanza ultrasonico sul monitor seriale. Modificare **ultrasonic ultrasonic(7)**; nel programma di esempio in **ultrasonic ultrasonic(0)**; (il sensore di distanza ultrasonico sarà collegato alla porta **A0** della scheda di espansione XIAO).



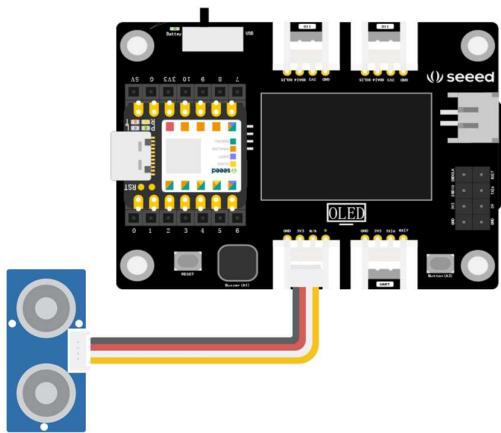
Aprire il file di esempio modificato tramite il seguente path, [🔗](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L13_UltrasonicDisplayOnTerm_XIAO_en)  
[https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L13\\_UltrasonicDisplayOnTerm\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L13_UltrasonicDisplayOnTerm_XIAO_en).

```
#include "Ultrasonic.h" //declare the library file
Ultrasonic ultrasonic(0); //define variables, connect pins. If you're using XIAO RP2040/XIAO ESP32,
please change 0 to D0
void setup() {
    Serial.begin(9600);
}
void loop() {
    long RangeInInches; //define a long integer variable named RangeInInches
    long RangeInCentimeters; //define a long integer variable named RangeInCentimeters

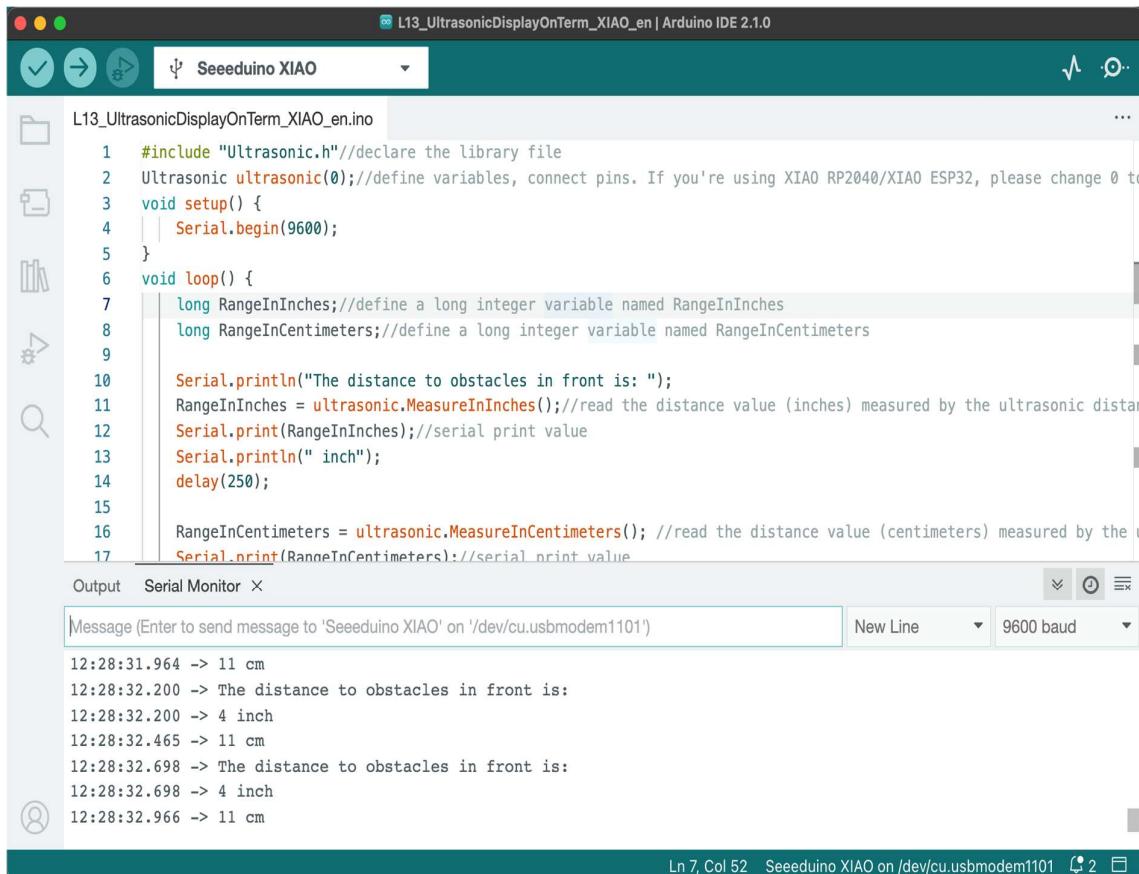
    Serial.println("The distance to obstacles in front is: ");
    RangeInInches = ultrasonic.MeasureInInches(); //read the distance value (inches) measured by the
    ultrasonic distance sensor and store it in the variable RangeInInches
    Serial.print(RangeInInches); //serial print value
    Serial.println(" inch");
    delay(250);

    RangeInCentimeters = ultrasonic.MeasureInCentimeters(); //read the distance value (centimeters)
    measured by the ultrasonic distance sensor and store it in the variable RangeInCentimeters
    Serial.print(RangeInCentimeters); //serial print value
    Serial.println(" cm");
    delay(250);
}
```

Il sensore di distanza a ultrasuoni è collegato all'interfaccia `A0`, come mostrato nella figura seguente:



Dopo aver caricato il codice, aprire il monitor seriale. Posizionare la mano o una scheda in qualsiasi posizione di fronte al sensore di distanza a ultrasuoni e osservare il cambiamento nei valori emessi dal monitor seriale.



```

L13_UltrasonicDisplayOnTerm_XIAO_en.ino
1 #include "Ultrasonic.h">//declare the library file
2 Ultrasonic ultrasonic(0);//define variables, connect pins. If you're using XIAO RP2040/XIAO ESP32, please change 0 to
3 void setup() {
4   Serial.begin(9600);
5 }
6 void loop() {
7   long RangeInInches;//define a long integer variable named RangeInInches
8   long RangeInCentimeters;//define a long integer variable named RangeInCentimeters
9
10  Serial.println("The distance to obstacles in front is:");
11  RangeInInches = ultrasonic.MeasureInInches();//read the distance value (inches) measured by the ultrasonic distan
12  Serial.print(RangeInInches);//serial print value
13  Serial.println(" inch");
14  delay(250);
15
16  RangeInCentimeters = ultrasonic.MeasureInCentimeters(); //read the distance value (centimeters) measured by the u
17  Serial.print(RangeInCentimeters);//serial print value

```

Output Serial Monitor X

Message (Enter to send message to 'Seeeduino XIAO' on '/dev/cu.usbmodem1101') New Line 9600 baud

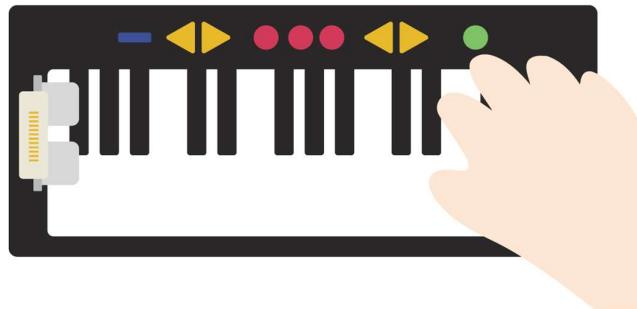
12:28:31.964 -> 11 cm  
12:28:32.200 -> The distance to obstacles in front is:  
12:28:32.200 -> 4 inch  
12:28:32.465 -> 11 cm  
12:28:32.698 -> The distance to obstacles in front is:  
12:28:32.698 -> 4 inch  
12:28:32.966 -> 11 cm

Ln 7, Col 52 Seeeduino XIAO on /dev/cu.usbmodem1101 ↻ 2 □

### 3.3.3 Progetto di Produzione: Arpa ad Aria Ultrasonica

#### Descrizione del Progetto

Il principio di funzionamento dell'arpa ad aria è quello di misurare la distanza dal modulo al palmo della mano tramite il sensore di distanza a ultrasuoni. A seconda della distanza, il cicalino emette note musicali diverse. Abbiamo già imparato come misurare la distanza e leggere i valori tramite il sensore di distanza a ultrasuoni con il programma di esempio. Ora, dobbiamo solo definire note musicali diverse per le distanze corrispondenti. Come mostrato nella figura seguente: In base alla larghezza del palmo, una nota musicale corrisponde a un'unità di 2 cm e l'esecuzione inizia da 4 cm. "Do, Re, Mi, Fa, Sol, La, Si, Do" corrispondono rispettivamente a 4cm, 6cm, 8cm, 10cm, 12cm, 14cm, 16cm, 18cm... e così via.



#### Scrittura del Programma

L'implementazione del programma di "air harp" richiede i seguenti passaggi:

- Dichiarazione del file di libreria, definizione di note e pin del buzzer diversi.
- Inizializzazione, impostazione dello stato del pin del buzzer.

- Lettura della distanza (cm) misurata dal sensore di distanza ultrasonico ed emissione di un giudizio di condizione per impostare diverse distanze per emettere note diverse.

## Utilizzo della Funzione tone() per Riprodurre una Melodia

Quando vogliamo controllare il buzzer per riprodurre note o canzoni tramite il programma, dobbiamo impostare noi stessi il valore di frequenza di ogni nota. Se una canzone ha molte note, è troppo problematico regolarne una per una e mette alla prova la nostra conoscenza della teoria musicale e l'intonazione. Esiste un metodo più semplice? Certo!

Quando definiamo le note, possiamo fare riferimento alla funzione `tone()` scritta sul sito web di Arduino 

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody>, questa funzione definisce la frequenza corrispondente di diverse note tramite `pitches.h`, che ci è comodo per usare la funzione `tone()` per impostare le note emesse dal buzzer. Il codice di `pitches.h` è mostrato di seguito:

```
/*
 * pitches.h
 */

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
```

---

```
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

---

## Attività 2: Arpa ad Aria Ultrasonica

**Passaggio 1:** Dichiarare il file di libreria, definire note diverse e pin del buzzer. Le note principali che utilizziamo sono “Do Re Mi Fa Sol La Si Do”, corrispondenti a “C5 D5 E5 F5 G5 A5 B5 C6”. Si possono definire solo le note di cui si ha bisogno per evitare che il programma sembri troppo lungo.

```
#include "Ultrasonic.h" //declare the library file
Ultrasonic ultrasonic(0); //define the ultrasonic object and connect the ultrasonic wave to the A0
interface. If you're using XIAO RP2040, please change 0 to D0
int buzzerPin = 3; //The buzzer is connected to the A3 interface, if you're using XIAO RP2040, please
change 3 to A3

#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
```

---

**Passaggio 2:** Inizializzare la velocità in baud e impostare lo stato del pin del cicalino.

```
void setup()
{
    Serial.begin(9600);
    pinMode(buzzerPin, OUTPUT);
}
```

---

**Passaggio 3:** Leggere la distanza (cm) misurata dal sensore di distanza ultrasonico ed emettere un giudizio di condizione per impostare diverse distanze per emettere note diverse. Dato che l'impostazione dell'arpa ad aria è che diverse distanze attivano note diverse, e questa distanza è un valore intero lungo, dobbiamo usare la funzione `long()` per definire il valore restituito dall'onda ultrasonica. Ad esempio, `(long)RangeInCentimeters== 4`, ovvero il valore di distanza restituito dall'onda ultrasonica è 4.. Corrispondente al buzzer che emette

note diverse, utilizzare la funzione `tone()`, ad esempio, `tone(3,NOTE_C5,100)`, ovvero il buzzer sul pin 3, emette la nota `NOTE_C5` (`DO`), che dura 100 millisecondi.

```
void loop()
{
    // Read the distance value detected by the ultrasonic distance sensor, in centimeters, and print
    // it on the serial monitor
    long RangeInCentimeters;
    RangeInCentimeters = ultrasonic.MeasureInCentimeters();
    Serial.print(RangeInCentimeters);
    Serial.println(" cm");
    delay(250);
    // Using an if statement for conditional judgment, when the distance is 4, 6, 8, 10, 12, 14, 16,
    18, it corresponds to C5, D5, E5, F5, G5, A5, B5, C6
    if (((long)RangeInCentimeters== 4)) { //Do
        tone(3,NOTE_C5,100);
    }

    if (((long) RangeInCentimeters== 6)) { //Re
        tone(3,NOTE_D5,100);
    }

    if (((long) RangeInCentimeters== 8)) { //Mi
        tone(3,NOTE_E5,100);
    }

    if (((long) RangeInCentimeters== 10)) { //Fa
        tone(3,NOTE_F5,100);
    }

    if (((long) RangeInCentimeters== 12)) { //Sol
        tone(3,NOTE_G5,100);
    }

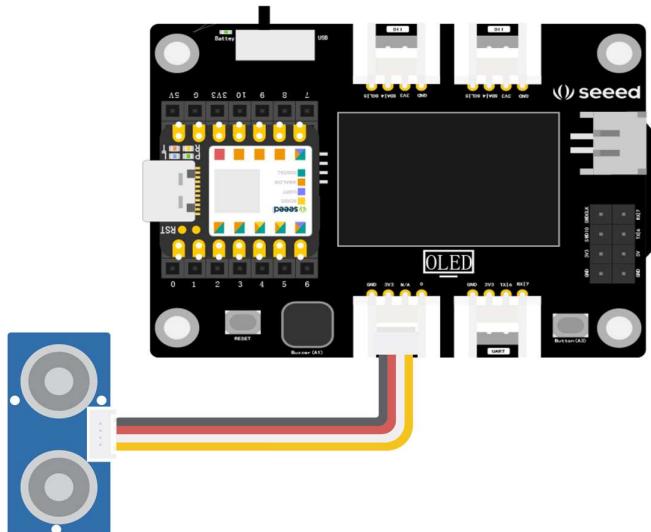
    if (((long) RangeInCentimeters== 14)) { //La
        tone(3,NOTE_A5,100);
    }

    if (((long) RangeInCentimeters== 16)) { //Xi
        tone(3,NOTE_B5,100);
    }

    if (((long) RangeInCentimeters== 18)) { //Do
        tone(3,NOTE_C6,100);
    }
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L13\\_UltrasonicPiano\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L13_UltrasonicPiano_XIAO_en)

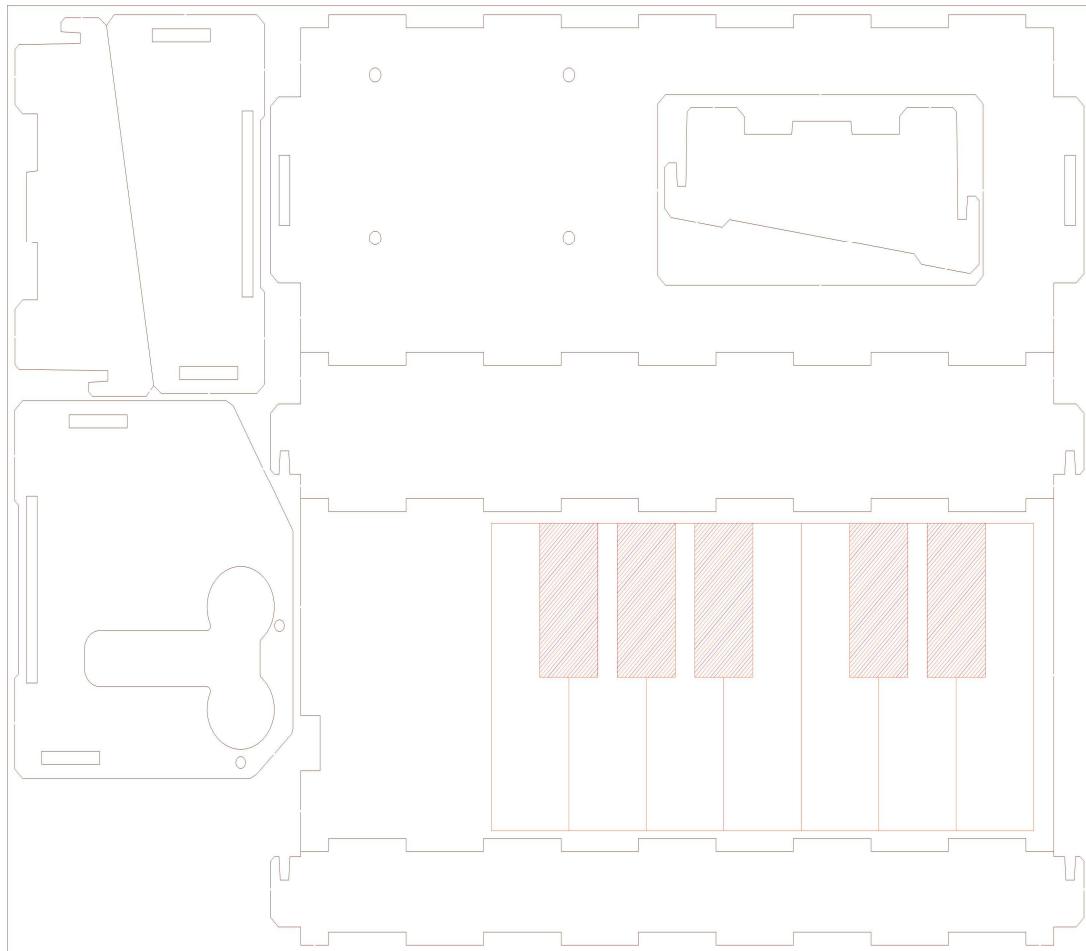
**Passaggio 4:** Collegare l'hardware e caricare il programma. Collegare il sensore di distanza ultrasonico all'interfaccia `A0` della scheda di espansione XIAO come mostrato di seguito:



Utilizzare il cavo dati per collegare XIAO al computer, cliccare sul pulsante “Upload”, caricare il programma sull’hardware, quando l’area di debug mostra “Upload Successful”, aprire il monitor seriale e iniziare a suonare con il palmo della mano.

### 3.3.4 Design Esteriore

L’ispirazione per l’arpa ad aria deriva dal pianoforte, con una nota ogni 2 cm progettata anche in base allo stile dei tasti del pianoforte. Nel processo di creazione dell’aspetto, possiamo tagliare una superficie di arpa da una tavola di tiglio e fissare il sensore di portata ultrasonica all’estremità sinistra dell’arpa. Forniamo anche file per il taglio laser come riferimento, che possono essere facilmente assemblati, come mostrato nell’immagine:



Download dei file adatti alla macchina per il taglio laser [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO\\_Air\\_Piano.dxf](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/blob/main/dxf/XIAO_Air_Piano.dxf).





## 3.4 Implementazione di connessioni Wi-Fi e applicazioni con XIAO ESP32C3

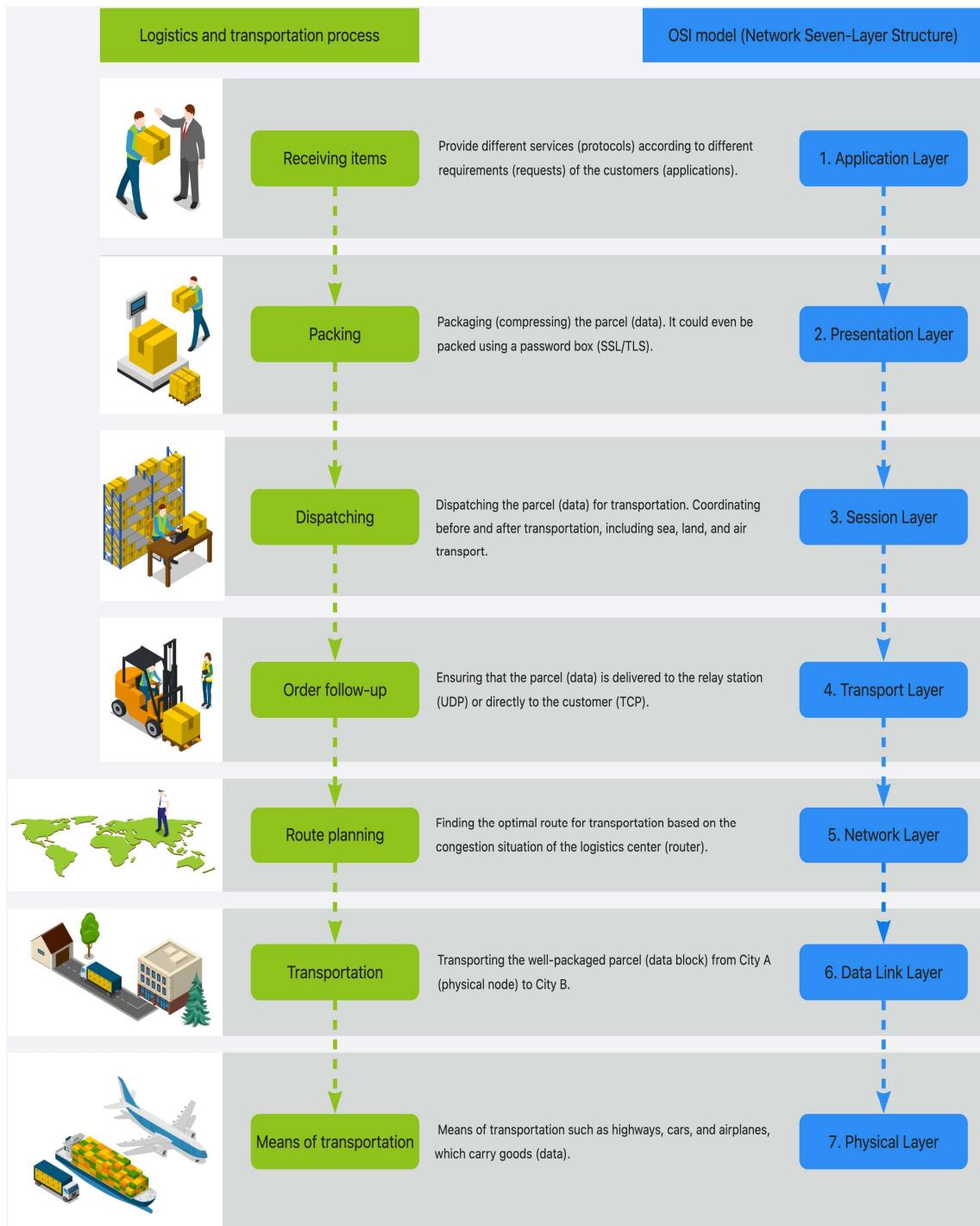
Piuttosto che dire che i computer hanno cambiato il mondo, sarebbe più corretto dire che lo hanno fatto le reti di computer. L'emergere delle reti ha reso i computer davvero diversi dagli strumenti precedenti. La condivisione e lo scambio di informazioni hanno reso i computer un prodotto epocale. In questa sezione, impareremo come implementare le richieste di rete utilizzando XIAO ESP32C3, che ha funzionalità Wi-Fi e Bluetooth (BLE). Ciò include la connessione di XIAO a una rete Wi-Fi, il ping di siti Web specifici e l'emissione di richieste GET/POST utilizzando il protocollo HTTP.

Seeed Studio XIAO ESP32C3

### 3.4.1 Conoscenza di Base

#### 3.4.1.1 Modello di Riferimento OSI (Modello di Rete a Sette Livelli)

OSI (Open System Interconnect) è comunemente noto come modello di riferimento OSI o struttura di rete a sette livelli, che è il modello di interconnessione di rete studiato dall'organizzazione ISO nel 1985. Questo standard architettonico definisce il framework a sette livelli (livello fisico, livello di collegamento dati, livello di rete, livello di trasporto, livello di sessione, livello di presentazione e livello di applicazione) per l'interconnessione di rete. Per facilitare la comprensione, il diagramma seguente utilizza un processo di trasporto logistico per corrispondere a ciascun livello del modello OSI.



Le seguenti conoscenze utilizzeranno i concetti di questi livelli.

### 3.4.1.2 ICMP (Internet Control Message Protocol) e Comando ping

ICMP (Internet Control Message Protocol) è un sottoprotocollo della suite di protocolli TCP/IP. Viene utilizzato per trasmettere messaggi di controllo tra host IP e router. I messaggi di controllo si riferiscono a messaggi sulla rete stessa, ad esempio se la rete è disponibile, se l'host è raggiungibile, se il percorso è disponibile, ecc. Sebbene questi messaggi di controllo non trasmettano dati utente, svolgono un ruolo importante nella trasmissione dei dati utente.

 Per saperne di più: visitare [la voce di Wikipedia su ICMP](#).

Spesso utilizziamo il protocollo ICMP nella rete, come il comando `ping` (disponibile sia in Linux che in Windows) che utilizziamo spesso per verificare se la rete è disponibile. Questo

processo `ping` è in realtà il processo di lavoro del protocollo ICMP. `ping` può testare la velocità di connessione tra due dispositivi e segnalare con precisione il tempo impiegato da un pacchetto per raggiungere la destinazione e tornare al dispositivo del mittente. Sebbene `ping` non fornisca dati su routing o hop, è comunque una metrica utile per misurare la latenza tra due dispositivi. Di seguito impareremo come implementare le richieste `ping` su XIAO ESP32C3. Prima di iniziare questo tentativo, dobbiamo imparare come connettere XIAO ESP32C3 al Wi-Fi.

### 3.4.2 Attività 1: Utilizzo della rete Wi-Fi su XIAO ESP32C3

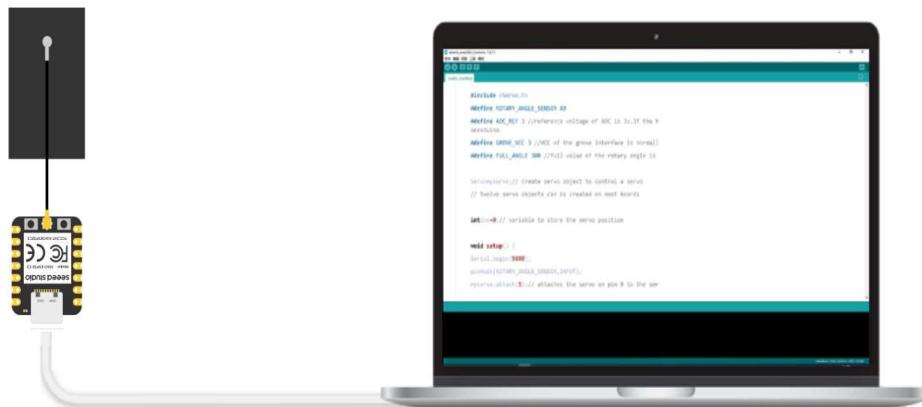
XIAO ESP32C3 supporta connessioni Wi-Fi con IEEE 802.11b/g/n. Di seguito verranno fornite le nozioni di base sull'utilizzo del Wi-Fi su questa scheda.

**⚠️ Nota:** Fare attenzione quando si prova a utilizzare la scheda di sviluppo XIAO ESP32C3 come hotspot (punto di accesso). Potrebbero verificarsi problemi di surriscaldamento e causare ustioni.

**Configurazione Hardware: Collegare un'Antenna a XIAO ESP32C3 e Collegarla al Computer**

**Passaggio 1.** Collegare l'antenna Wi-Fi/Bluetooth in dotazione al connettore IPEX sulla scheda di sviluppo.

**Passaggio 2.** Collegare XIAO ESP32C3 al computer tramite un cavo dati USB Type-C.



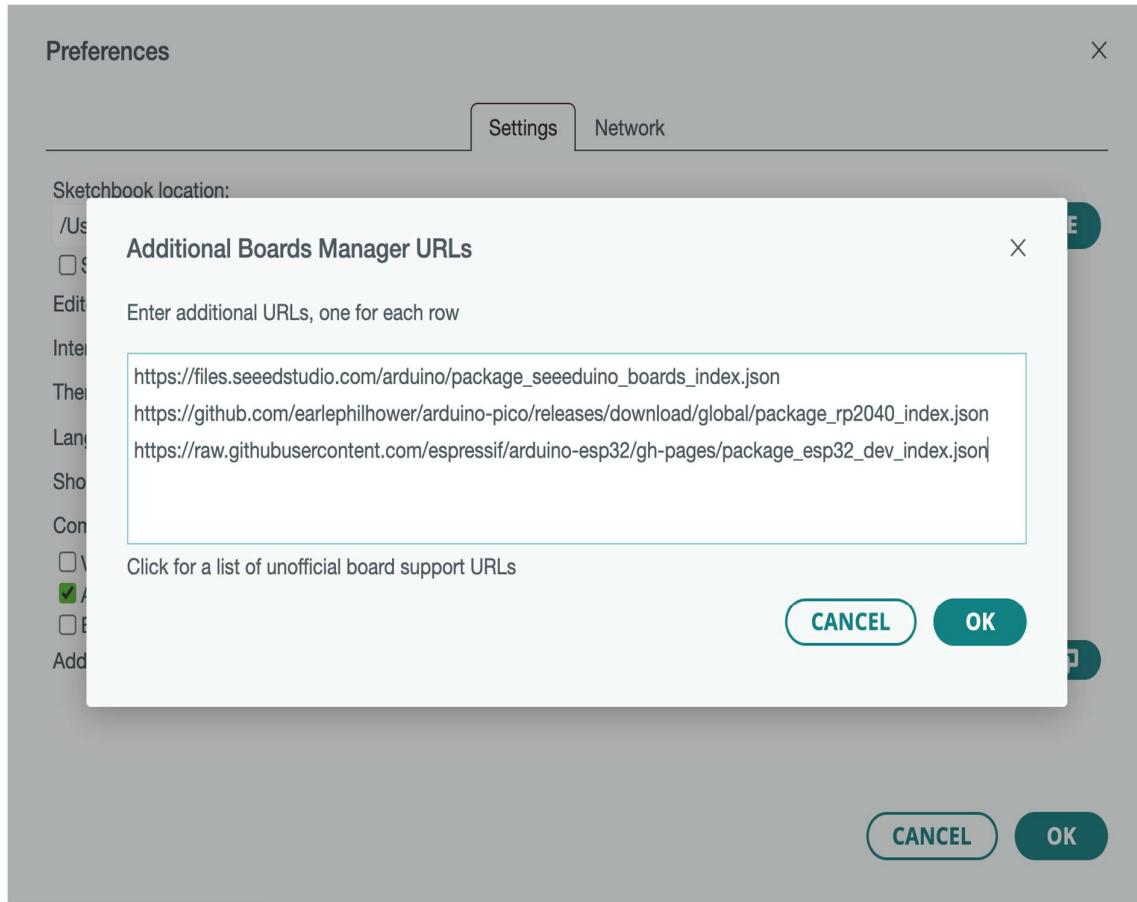
**Impostazione Software: Aggiungere il Pacchetto della Scheda ESP32 all'IDE Arduino**

**Passaggio 1.** Aprire le preferenze di Arduino IDE per aggiungere l'URL di Board Manager.

- Per gli utenti Windows, aprire prima l'Arduino IDE, cliccare su “File→Preferences” nella barra dei menù in alto e copiare il seguente URL in “URL Additional Board Manager URLs”.
- Per gli utenti Mac, aprire prima l'Arduino IDE, cliccare su “Arduino IDE→Preferences” nella barra dei menù in alto e copiare il seguente URL in “Additional Board Manager URLs”.

Per Seeed Studio XIAO ESP32C3, copiare il link qui sotto:

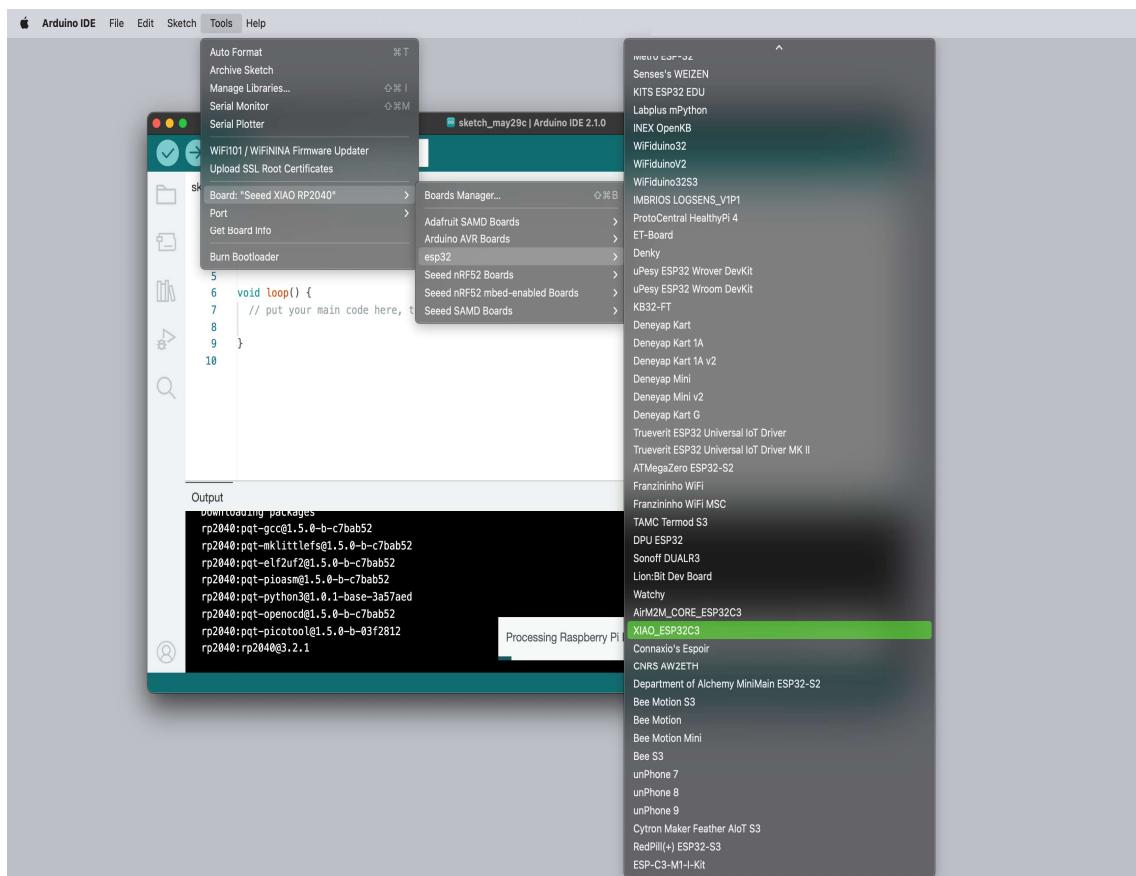
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json) nella barra degli URL di Board Manager e confermare, come mostrato nella figura qui sotto.



**Passaggio 2.** Nel menù Arduino IDE, cliccare su “Tools→Board→Board Manager”, digitare “esp32” nella barra di ricerca, trovare l’ultima versione di ESP32 Arduino nelle voci risultanti e cliccare su “Install”. Quando l’installazione inizia, si vedrà un pop-up di output. Una volta completata l’installazione, apparirà l’opzione “Installed”.

**Passaggio 3.** Selezionare la Scheda.

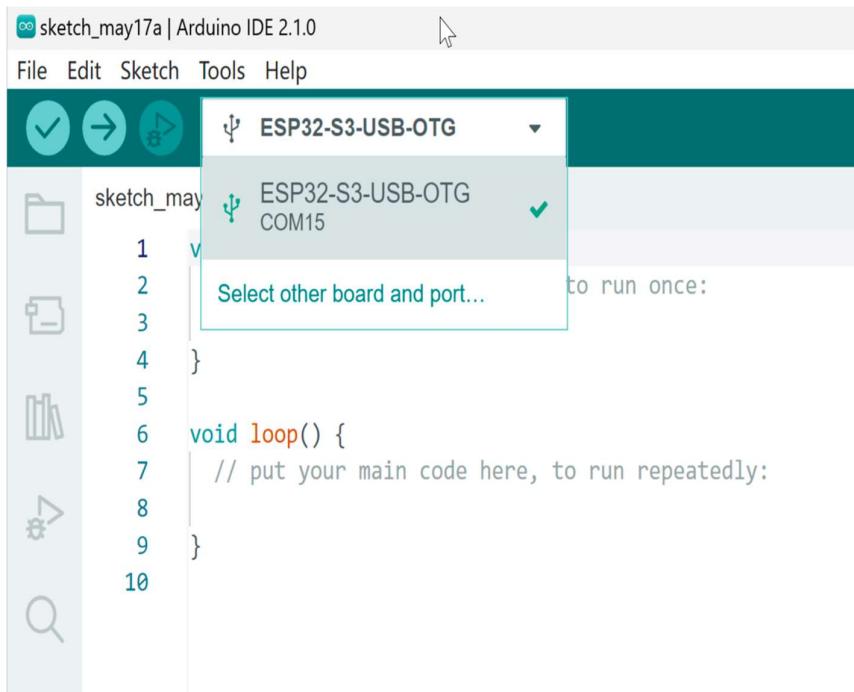
Si va su “Tools > Board > ESP32 Arduino” e selezionare “XIAO\_ESP32C3”. L’elenco sarà un po’ lungo e si dovrà scorrere verso il basso per trovarlo, come mostrato nella figura sottostante.



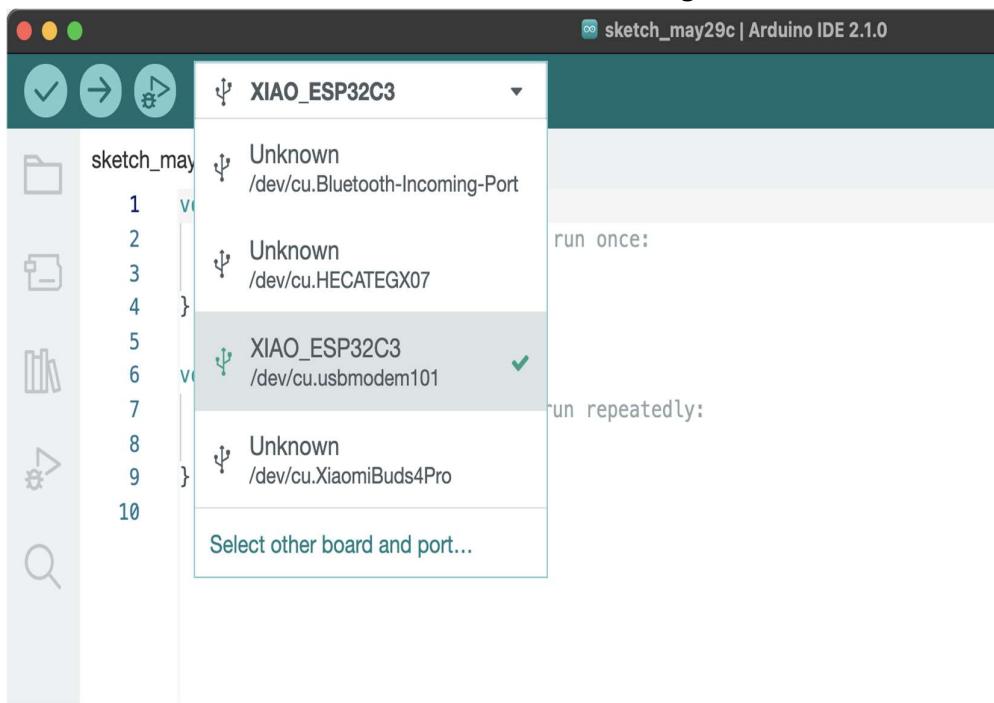
#### Passaggio 4. Aggiungere la Porta.

Controllare se la connessione della porta è corretta nell'Arudino IDE. In caso contrario, è necessario selezionarla manualmente.

- Per i sistemi Windows, la porta seriale è visualizzata come “COM+numero”, come mostrato nella figura sottostante.



- Sui sistemi Mac o Linux, il nome della porta è in genere `/dev/tty.usbmodem+numero` o `/dev/cu.usbmodem+numero`, come mostrato nella figura sottostante.



### **Scansione delle Reti Wi-Fi nelle Vicinanze (Modalità STA)**

In questo esempio, utilizzeremo XIAO ESP32C3 per cercare le reti Wi-Fi disponibili nell'area. La scheda di sviluppo in questo esempio sarà configurata in modalità STA.

**Passaggio 1.** Copiare e incollare il codice sottostante nell'IDE Arduino.

```
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop()
{
    Serial.println("scan start");

    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0) {
        Serial.println("no networks found");
    } else {
        Serial.print(n);
        Serial.println(" networks found");
        for (int i = 0; i < n; ++i) {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));
            Serial.print(")");
            Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " ":"*");
            delay(10);
        }
    }
}
```

```

    }

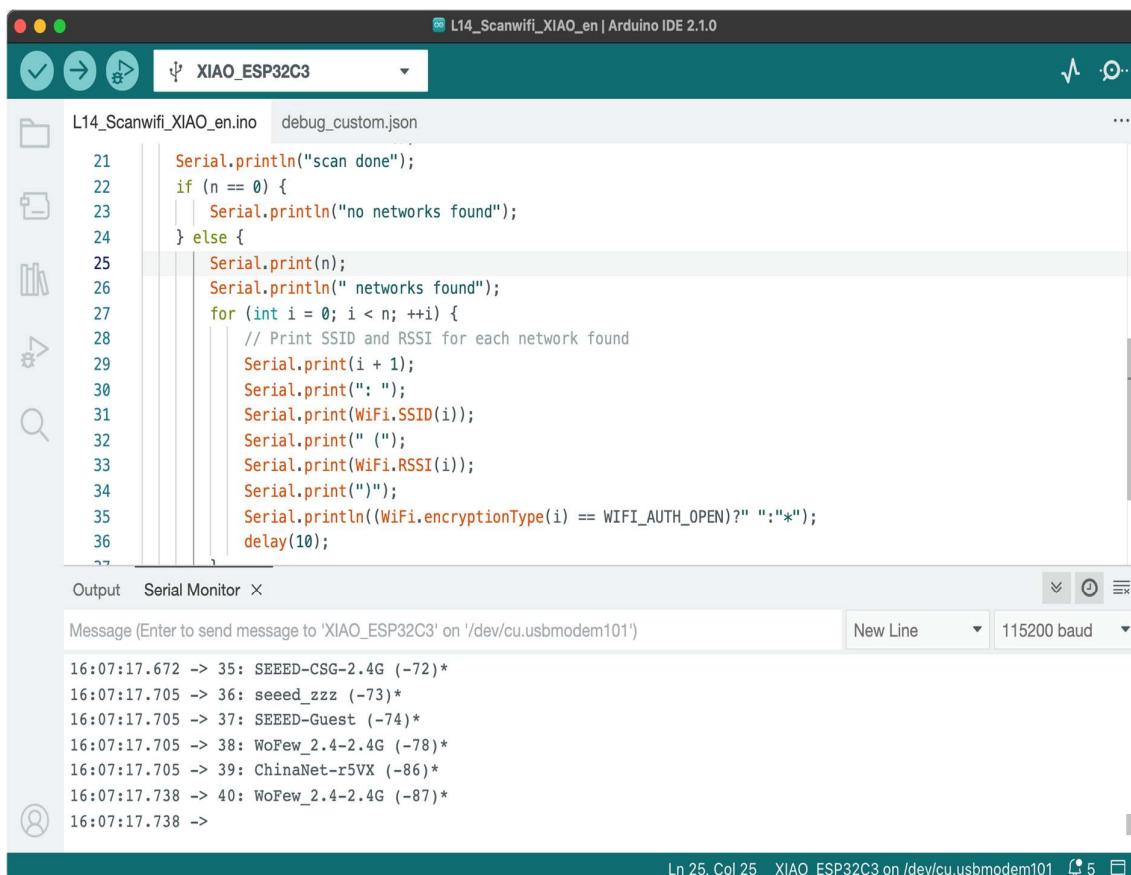
    Serial.println("");

    // Wait a bit before scanning again
    delay(5000);
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14\\_Scanwifi\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14_Scanwifi_XIAO_en)

**Passaggio 2.** Caricare il codice e aprire il monitor seriale per avviare la scansione delle reti Wi-Fi, come mostrato nella figura sottostante.



### Connessione a una Rete Wi-Fi

In questo esempio, utilizzeremo XIAO ESP32C3 per connetterci alla propria rete Wi-Fi.

**Passaggio 1.** Copiare e incollare il codice sottostante nell'IDE Arduino.

```

#include <WiFi.h>

const char* ssid      = "your-ssid";
const char* password = "your-password";

void setup()
{
    Serial.begin(115200);
    delay(10);

    // We start by connecting to a WiFi network

    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
}

```

```

        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

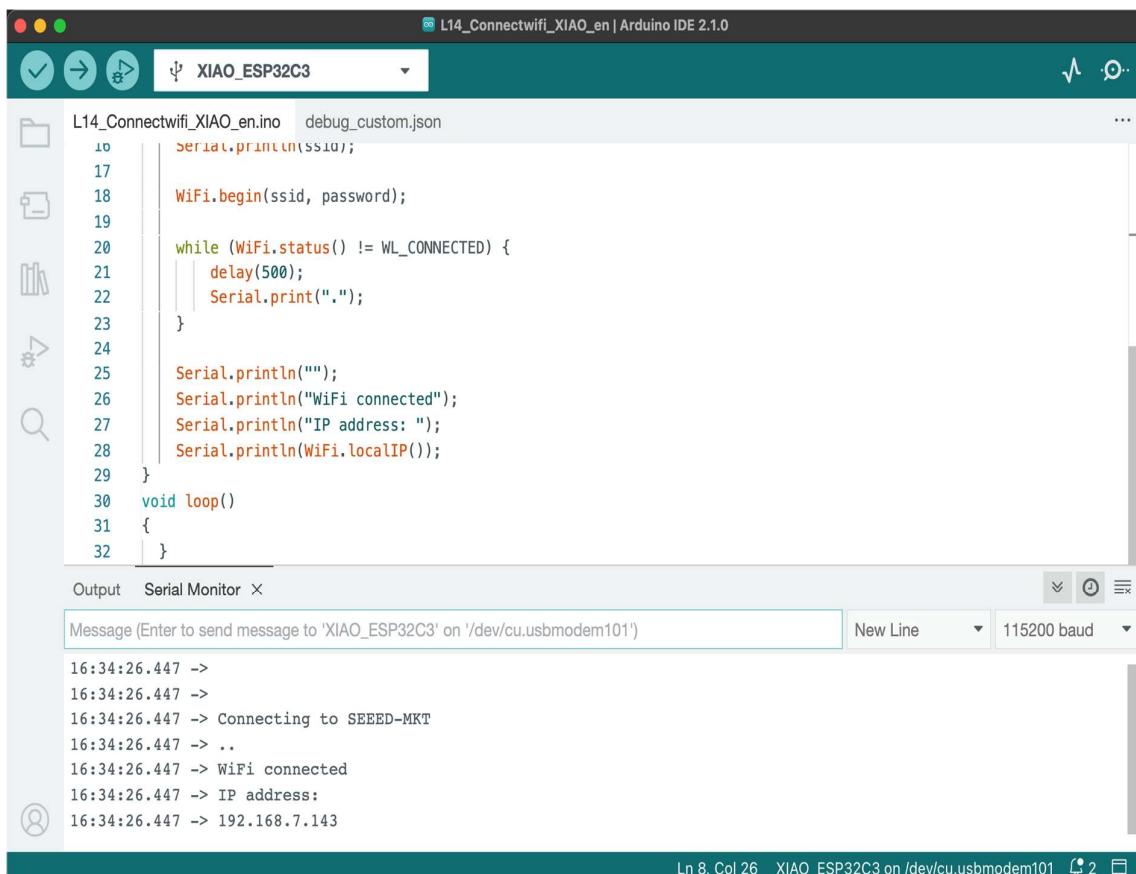
void loop()
{
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14\\_Connectwifi\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14_Connectwifi_XIAO_en)

Quindi, sostituire `your-ssid` nel codice con il nome della propria rete Wi-Fi e sostituire `your-password` nel codice con la password della rete Wi-Fi.

**Passaggio 2.** Caricare il codice e aprire il monitor seriale per verificare se la scheda di sviluppo è connessa alla rete Wi-Fi, come mostrato nella figura sottostante.



💡 Per imparare di più: Leggere la [documentazione Wiki](#) sull'uso di XIAO ESP32C3.

### 3.4.3 Attività 2: Ping di un Sito Web Specifico

Con le conoscenze di cui sopra, ora possiamo imparare come utilizzare XIAO ESP32C3 per effettuare il ping di un sito Web specificato.

**Passaggio 1.** Scaricare e installare la libreria ESP32Ping. Inserire l'URL [ESP32Ping](#)

<https://github.com/marian-craciunescu/ESP32Ping> per andare alla pagina GitHub, cliccare su Code→Download ZIP per scaricare il pacchetto di risorse sul computer locale, come mostrato nella figura seguente.

This branch is 52 commits ahead, 4 commits behind [mariancraciunescu](#)/Update library.properties

[Clone](#)

Local Codespaces [New](#)

HTTPS SSH GitHub CLI

<https://github.com/mariancraciunescu/ESP32Ping/archive/.../master.zip>

Use Git or checkout with SVN using the web URL.

[Open with GitHub Desktop](#)

[Download ZIP](#)

[About](#)

Ping library for ESP32 Arduino core

[Readme](#)

[LGPL-2.1 license](#)

72 stars

7 watching

122 forks

[Releases](#) 1

1.6 [Latest](#) on Feb 10, 2021

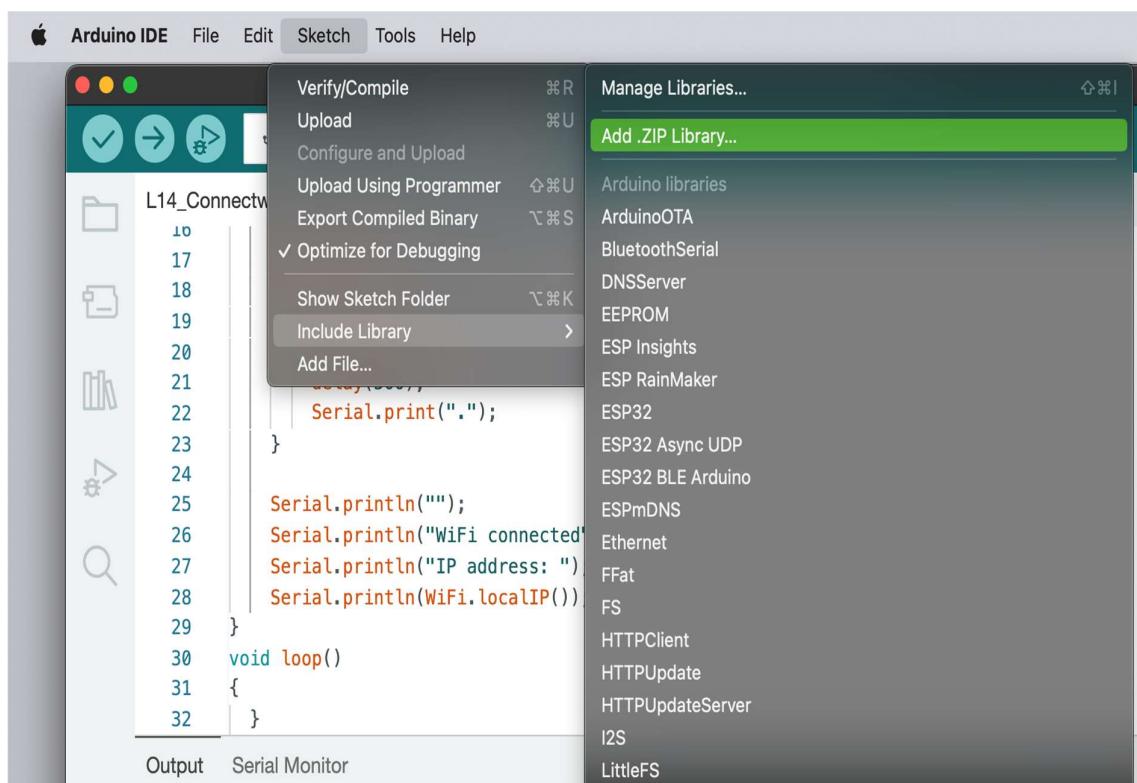
[Packages](#)

No packages published

[Languages](#)

C++ 95.3% C 4.7%

Dopo il download, aprire l'IDE Arduino, cliccare su Sketch→Include Library→Add .ZIP Library e scegliere il file ZIP appena scaricato.



**Passaggio 2.** Copiare e incollare il codice sottostante nell'IDE Arduino. Questo codice imposta il sito Web di prova su [www.seeedstudio.com](https://www.seeedstudio.com). Ricordarsi di sostituire `your-ssid` nel codice con il nome della rete Wi-Fi e `your-password` nel codice con la password del Wi-Fi.

---

```
////////// IDE: // Arduino 2.0.3 // Platform: // esp32 2.0.4 - https://github.com/espressif/arduino-esp32 // Board: // XIAO ESP32C3 // Libraries: // ESP32Ping 1.6 - https://github.com/marian-craciunescu/ESP32Ping

////////// Includes

#include <WiFi.h>
#include <ESP32Ping.h>

static constexpr unsigned long INTERVAL = 3000; // [msec.]

static const char WIFI_SSID[] = "your-ssid";
static const char WIFI_PASSPHRASE[] = "your-password";

static const char SERVER[] = "www.google.com";

void setup()
{
    Serial.begin(115200);
    delay(1000);
    Serial.println();
    Serial.println();

    Serial.println("WIFI: Start.");
    WiFi.mode(WIFI_STA);
    if (WIFI_SSID[0] != '\0')
    {
        WiFi.begin(WIFI_SSID, WIFI_PASSPHRASE);
    }

    else
    {
        WiFi.begin();
    }
}

void loop()
{
    static int count = 0;

    const bool wifiStatus = WiFi.status() == WL_CONNECTED;
    const int wifiRssi = WiFi.RSSI();

    const bool pingResult = !wifiStatus ? false : Ping.ping(SERVER, 1);
    const float pingTime = !pingResult ? 0.f : Ping.averageTime();

    Serial.print(count);
    Serial.print('\t');
    Serial.print(wifiStatus ? 1 : 0);
    Serial.print('\t');
    Serial.print(wifiRssi);
    Serial.print('\t');
    Serial.print(pingResult ? 1 : 0);
    Serial.print('\t');
    Serial.println(pingTime);
    count++;

    delay(INTERVAL);
}
```

---

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14\\_Ping\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14_Ping_XIAO_en)

**Passaggio 3.** Caricare il codice e aprire il monitor seriale per controllare i risultati del ping, come mostrato nella figura sottostante.

```

L14_Ping_XIAO_en.ino debug_custom.json ...
15 #include <ESP32Ping.h>
16
17 static constexpr unsigned long INTERVAL = 3000; // [msec.]
18
19 static const char WIFI_SSID[] = "SEEED-MKT";
20 static const char WIFI_PASSPHRASE[] = "edgemaker2023";
21
22 static const char SERVER[] = "www.seeedstudio.com";
23
24 void setup()
25 {
26     Serial.begin(115200);
27     delay(1000);
28     Serial.println();
29     Serial.println();
30
31     Serial.println("WIFI: Start.");

```

Output Serial Monitor ×

Message (Enter to send message to 'XIAO\_ESP32C3' on '/dev/cu.usbmodem101')

| 17:02:33.178 | -> | 3 | 1 | -51    |
|--------------|----|---|---|--------|
| 17:02:36.446 | -> | 4 | 1 | -51    |
| 17:02:39.903 | -> | 5 | 1 | -53    |
| 17:02:43.270 | -> | 6 | 1 | -51    |
| 17:02:46.582 | -> | 7 | 1 | -51    |
| 17:02:50.605 | -> | 8 | 1 | -49    |
| 17:02:54.640 | -> | 9 | 1 | -51    |
|              |    |   | 1 | 285.97 |
|              |    |   | 1 | 265.26 |
|              |    |   | 1 | 403.01 |
|              |    |   | 1 | 389.11 |
|              |    |   | 1 | 321.51 |
|              |    |   | 0 | 0.00   |
|              |    |   | 0 | 0.00   |

Ln 20, Col 55 XIAO\_ESP32C3 on /dev/cu.usbmodem101 4 2

### 3.4.4 Creazione del Progetto: Utilizzo di XIAO ESP32C3 per Effettuare Richieste HTTP GET e HTTP POST

#### 3.4.4.1 Introduzione al Protocollo HTTP

HTTP sta per HyperText Transfer Protocol. È un protocollo a livello applicativo per sistemi informativi distribuiti, collaborativi e ipermediati. HTTP è il protocollo di trasmissione di rete più ampiamente utilizzato su Internet e tutti i file WWW devono essere conformi a questo standard. HTTP è progettato per la comunicazione tra browser Web e server Web, ma può essere utilizzato anche per altri scopi. HTTP è un protocollo che utilizza TCP/IP per trasmettere dati (come file HTML, file di immagini, risultati di query, ecc.). Nonostante il suo ampio utilizzo, HTTP presenta notevoli falle di sicurezza, principalmente la trasmissione di dati in testo normale e la mancanza di controlli di integrità dei messaggi. Questi sono esattamente i due aspetti di sicurezza più critici nelle applicazioni emergenti come i pagamenti online, il trading online, Internet of Things, ecc. Browser come Google Chrome, Internet Explorer e Firefox emettono avvisi sulle connessioni non sicure quando si visitano siti Web con contenuti misti composti da contenuti crittografati e non tramite HTTP.

#### 3.4.4.2 Introduzione al Protocollo HTTPS

HTTPS sta per HyperText Transfer Protocol Secure. È un protocollo per la comunicazione sicura su una rete di computer. HTTPS comunica tramite HTTP ma utilizza SSL/TLS per crittografare i pacchetti. Lo scopo principale di HTTPS è autenticare l'identità del server del sito Web e proteggere la privacy e l'integrità dei dati scambiati.



### 3.4.4.3 Metodi di richiesta HTTP

Secondo lo standard HTTP, le richieste HTTP possono utilizzare più metodi di richiesta.

HTTP1.0 ha definito tre metodi di richiesta: GET, POST e HEAD. HTTP1.1 ha aggiunto sei nuovi metodi di richiesta: OPTIONS, PUT, PATCH, DELETE, TRACE e CONNECT.

| N. | Metodo  | Descrizione                                                                                                                                                                                                                                                             |
|----|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | GET     | Richiede informazioni sulla pagina specificata e restituisce il corpo dell'entità.                                                                                                                                                                                      |
| 2  | HEAD    | Simile a una richiesta GET, ma la risposta restituita non contiene contenuti specifici, utilizzata per ottenere headers.                                                                                                                                                |
| 3  | POST    | Invia dati per l'elaborazione a una risorsa specificata (ad esempio, invia un modulo o carica un file). I dati sono inclusi nel corpo della richiesta. Le richieste POST possono comportare la creazione di una nuova risorsa e/o la modifica di una risorsa esistente. |
| 4  | PUT     | I dati inviati dal client al server sostituiscono il contenuto di un documento specificato.                                                                                                                                                                             |
| 5  | DELETE  | Richiede al server di eliminare una pagina specificata.                                                                                                                                                                                                                 |
| 6  | CONNECT | Riservato in HTTP/1.1 per i server proxy che possono passare la connessione a una modalità pipe.                                                                                                                                                                        |
| 7  | OPTIONS | Consente al client di visualizzare le capacità del server.                                                                                                                                                                                                              |
| 8  | TRACE   | Esegue l'eco della richiesta ricevuta dal server, utilizzata principalmente per test o diagnosi.                                                                                                                                                                        |
| 9  | PATCH   | È un supplemento al metodo PUT, utilizzato per aggiornare parzialmente una risorsa nota.                                                                                                                                                                                |

Abbiamo già imparato come connettersi a una rete Wi-Fi utilizzando XIAO ESP32C3. Ora, proviamo alcune operazioni più complesse basate sulla rete. Le sezioni seguenti introdurranno come utilizzare XIAO ESP32C3 per inviare richieste HTTP GET e HTTP POST.

### 3.4.4.4 Attività 3: Utilizzo di XIAO ESP32C3 per Inviare una Richiesta HTTP GET

Per inviare una richiesta HTTP GET, è necessario un server backend corrispondente che supporti la richiesta. Per un test conveniente, possiamo impostare un server backend sul nostro PC, consentendo a XIAO ESP32C3 di inviare una richiesta HTTP GET al PC tramite la connessione Wi-Fi locale.

Esistono molti modi per impostare un servizio backend. In questo caso, utilizzeremo il famoso framework web Python — FastAPI per configurare il server backend. Per saperne di più su questo strumento, esaminare la sua [documentazione ufficiale](#).

## Configurazione di un Server Backend con FastAPI

Ecco il codice del server Python.

```
from typing import Union
from pydantic import BaseModel
from fastapi import FastAPI
import datetime

app = FastAPI()
items = {}

class Sensor_Item(BaseModel):
    name: str
    value: float

@app.on_event("startup")
async def startup_event():
    items["sensor"] = {"name": "Sensor", "Value": 0}

@app.get("/items/{item_id}")
async def read_items(item_id: str):

    return items[item_id], datetime.datetime.now()

@app.post("/sensor/")
async def update_sensor(si: Sensor_Item):
    items["sensor"]["Value"] = si.value
    return si

@app.get("/")
def read_root():
    return {"Hello": "World"}
```

Questo frammento di codice, implementato utilizzando il framework Python FastAPI, può restituire le informazioni più recenti del sensore memorizzate sul server backend quando utilizziamo una richiesta `get` SU `http://domain/items/sensor`. Quando utilizziamo `post` per inviare dati a `http://domain/sensor/`, può modificare e registrare l'ultimo valore del sensore. I passaggi operativi sono i seguenti:

**Passaggio 1.** Creare un file python denominato `main.py` localmente, copiare e incollare il codice sopra in `main.py`. Quindi, sul PC, aprire il terminale ed eseguire i seguenti comandi per installare FastAPI.

---

```
pip install fastapi
pip install "uvicorn[standard]"
```

---

**Passaggio 2.** Eseguire il seguente comando per avviare il servizio backend e il monitoraggio locale.

---

```
uvicorn main:app --reload --host 0.0.0.0
```

---

⚠️ Nota: Quando si esegue il comando sopra, assicurarsi che il terminale sia attualmente nella directory in cui risiede `main:app`. Se durante l'esecuzione viene visualizzato un prompt:

---

```
ERROR: [Errno 48] Address already in use
```

---

Significa che l'indirizzo corrente è già occupato e c'è un conflitto. Si può specificare una porta specifica come mostrato nel comando seguente.

---

```
uvicorn main:app --reload --host 0.0.0.0 --port 1234
```

---

Se l'errore [Errno 48] continua a comparire, si può modificare il numero della porta dopo `port`.

Le informazioni del prompt dopo l'esecuzione corretta del comando sono le seguenti

---

```
INFO: Will watch for changes in these directories: ['']
INFO: Uvicorn running on http://0.0.0.0:1234 (Press CTRL+C to quit)
INFO: Started reloader process [53850] using WatchFiles
INFO: Started server process [53852]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

---

Il server backend per i test ora funziona normalmente.

### Utilizzo di XIAO ESP32C3 per inviare una richiesta HTTP GET

Successivamente, eseguiremo un test di richiesta su XIAO ESP32C3. Il metodo GET dovrebbe essere utilizzato solo per la lettura dei dati e non in operazioni che generano “effetti collaterali”. Le richieste GET emesse direttamente dai browser possono essere attivate solo da un URL. Per trasportare alcuni parametri al di fuori dell’URL in GET, ci si può affidare solo alla querystring (stringa di query) allegata all’URL.

**Passaggio 1:** Copiare e incollare il seguente codice nell’IDE Arduino. Questo codice imposta il `serverName` testato su `http://192.168.1.2/items/sensor`. `192.168.1.2` deve essere sostituito con l’indirizzo IP del PC che funge da server backend. Per ottenere l’indirizzo IP del PC, gli utenti Windows possono immettere il comando `ipconfig` nella finestra della riga di comando, mentre gli utenti Mac possono immettere il comando `ifconfig` nella finestra del terminale. Si deve modificare `your-ssid` nel codice con il nome della propria rete Wi-Fi e `your-password` con la password corrispondente.

```
#include "WiFi.h"
#include <HTTPClient.h>

const char* ssid = "your-ssid";
const char* password = "your-password";
String serverName = "http://192.168.1.2/items/sensor";
unsigned long lastTime = 0;
unsigned long timerDelay = 5000;

void setup()
{
    Serial.begin(115200);

    WiFi.begin(ssid, password);
    Serial.println("Connecting");
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
    Serial.println(WiFi.localIP());

    Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before publishing the first reading.");
    Serial.println("Setup done");
}

void loop()
{
    if ((millis() - lastTime) > timerDelay) {
        //Check WiFi connection status
        if(WiFi.status()== WL_CONNECTED) {
            HTTPClient http;

            String serverPath = serverName ;
            http.begin(serverPath.c_str());

            int httpResponseCode = http.GET();

            if (httpResponseCode>0) {
                Serial.print("HTTP Response code: ");
                Serial.println(httpResponseCode);
                String payload = http.getString();
                Serial.println(payload);
            }
            else {
                Serial.print("Error code: ");
                Serial.println(httpResponseCode);
            }

            http.end();
        }
        else {
    }
```

```

        Serial.println("WiFi Disconnected");
    }

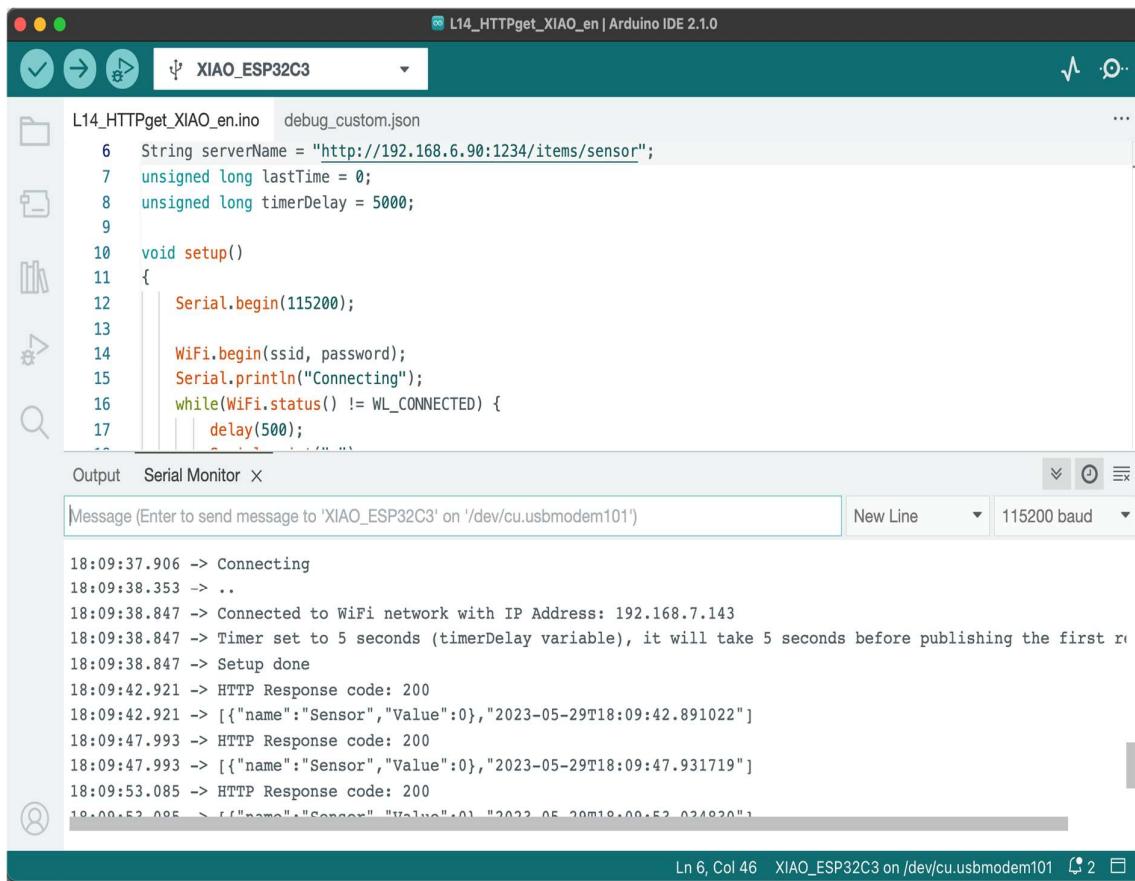
    lastTime = millis();
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14\\_HTTPget\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14_HTTPget_XIAO_en)

**⚠️ Nota:** Dobbiamo modificare `serverName` nel codice Arduino con l'indirizzo IP dell'host che esegue il servizio backend. Lo XIAO ESP32C3 deve trovarsi sulla stessa rete locale. Se l'IP della rete locale del server backend (in questo esempio, il proprio PC) è 192.168.1.2, l'interfaccia di richiesta GET è `http://192.168.1.2/items/sensor` e le altre interfacce sono simili. Specificando una porta durante l'esecuzione del servizio backend e del monitoraggio locale, l'interfaccia di richiesta GET sarebbe `http://192.168.1.2:1234/items/sensor`.

**Passaggio 2:** Caricare il Codice su XIAO ESP32C3 nell'IDE Arduino. Dopo che il caricamento è riuscito, aprire il monitor seriale per controllare il risultato restituito dal nostro server backend dopo l'emissione del GET, come mostrato nella figura sottostante.



Il corpo di una richiesta POST emessa da un browser ha principalmente due formati, uno è `application/x-www-form-urlencoded` usato per trasmettere dati semplici, più o meno nel formato di `key1=value1&key2=value2`. L'altro è per trasmettere file e userà il formato `multipart/form-data`. Quest'ultimo è adottato perché il metodo di codifica di `application/x-www-form-urlencoded` è molto inefficiente per dati binari come i file. Successivamente, invieremo dati sperimentali al server backend creato sulla nostra macchina in modo simile all'invio di un modulo e verificheremo se il server backend ha ricevuto i dati.

**Passaggio 1:** Prima di iniziare questo esempio, assicurarsi che il server backend creato con FastAPI nel passaggio precedente funzioni normalmente. In caso contrario, fare riferimento alle istruzioni sopra riportate per avviare il programma del server.

**Passaggio 2:** Copiare e incollare il seguente codice nell'IDE Arduino. Questo codice imposta il `serverName` a `http://192.168.1.2/sensor/`. 192.168.1.2 deve essere sostituito con l'indirizzo IP del PC che funge da server backend. Ricordarsi di cambiare `your-ssid` nel codice con il nome della propria rete Wi-Fi e `your-password` con la password corrispondente.

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "your-ssid";
const char* password = "your-password";

const char* serverName = "https://192.168.1.2/sensor/";

unsigned long lastTime = 0;
unsigned long timerDelay = 5000;

void setup() {
    Serial.begin(115200);

    WiFi.begin(ssid, password);
    Serial.println("Connecting");
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.print("Connected to WiFi network with IP Address: ");
    Serial.println(WiFi.localIP());

    Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before publishing the first reading.");
}

void loop() {
    //Send an HTTP POST request every 10 minutes
    if ((millis() - lastTime) > timerDelay) {
        //Check WiFi connection status
        if(WiFi.status()== WL_CONNECTED) {
            WiFiClient client;
            HTTPClient http;

            http.begin(client, serverName);

            http.addHeader("Content-Type", "application/json");
            int httpResponseCode = http.POST("{\"name\":\"sensor\", \"value\":\"123\"}");

            Serial.print("HTTP Response code: ");
            Serial.println(httpResponseCode);

            // Free resources
            http.end();
        }
    }

    else {
        Serial.println("WiFi Disconnected");
    }

    lastTime = millis();
}
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14\\_HTTPpost\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L14_HTTPpost_XIAO_en)

Passaggio 2: Caricare il codice su XIAO ESP32C3 usando l'IDE Arduino. Dopo un caricamento riuscito, aprire Serial Monitor per esaminare il risultato restituito dal nostro server backend in risposta alla richiesta GET. L'immagine qui sotto illustra il processo.

The screenshot shows the Arduino IDE interface. The top bar displays "L14\_HTTPpost\_XIAO\_en | Arduino IDE 2.1.0". The left sidebar shows files "L14\_HTTPpost\_XIAO\_en.ino" and "debug\_custom.json". The main code area contains the following C++ code:

```

1 #include <WiFi.h>
2 #include <HTTPClient.h>
3
4 const char* ssid = "SEED-MKT";
5 const char* password = "edgemaker2023";
6
7 const char* serverName = "http://192.168.6.90:1234/sensor/";
8
9 unsigned long lastTime = 0;
10 unsigned long timerDelay = 5000;
11
12 void setup() {
13   Serial.begin(115200);
14
15   WiFi.begin(ssid, password);
16   Serial.println("Connecting");
17   while(WiFi.status() != WL_CONNECTED) {
18     delay(500);
19   }
20 }
```

The "Output" tab is selected, showing the "Serial Monitor" output:

```

Message (Enter to send message to 'XIAO_ESP32C3' on '/dev/cu.usbmodem101')
18:25:09.474 -> Connecting
18:25:09.474 -> ..
18:25:09.474 -> Connected to WiFi network with IP Address: 192.168.7.143
18:25:09.474 -> Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before publishing the first rea
18:25:10.859 -> HTTP Response code: 200
18:25:15.884 -> HTTP Response code: 200
18:25:20.925 -> HTTP Response code: 200
```

The status bar at the bottom indicates "Ln 7, Col 59 XIAO\_ESP32C3 on /dev/cu.usbmodem101 ⌂ 2 ⌂".

Il messaggio **\*\*HTTP Response code: 200\*\*** indica una richiesta riuscita. Sul proprio PC locale, aprire un browser e andare su **\*\*http://192.168.1.2/items/sensor\*\*** (sostituire l'indirizzo IP in base all'indirizzo IP effettivo del PC e, se è stata impostata una porta, aggiungere due punti seguiti dal numero di porta designato dopo l'indirizzo IP). Ora si dovrebbero vedere i dati più recenti inviati da XIAO ESP32C3. Poiché XIAO invia dati ogni 5 secondi, si possono sempre visualizzare i dati più recenti ricevuti dal server backend aggiornando la pagina corrente (il timestamp dei dati cambierà).



Abbiamo ora inviato correttamente i dati da XIAO ESP32C3 al server backend locale.

## 3.5 Telemetria e Comandi Tramite Protocollo MQTT con XIAO ESP32C3

Nella sezione precedente, abbiamo imparato come inviare richieste HTTP GET o POST da XIAO ESP32C3 a una macchina locale su una rete locale tramite Wi-Fi. In questa sezione, esamineremo: protocolli di comunicazione, Message Queuing Telemetry Transport (MQTT),

telemetria (dati raccolti dai sensori e inviati al cloud) e comandi (messaggi inviati dal cloud a un dispositivo che gli ordinano di fare qualcosa).

### 3.5.1 Conoscenza di Base

#### 3.5.1.1 IoT (Internet of Things)

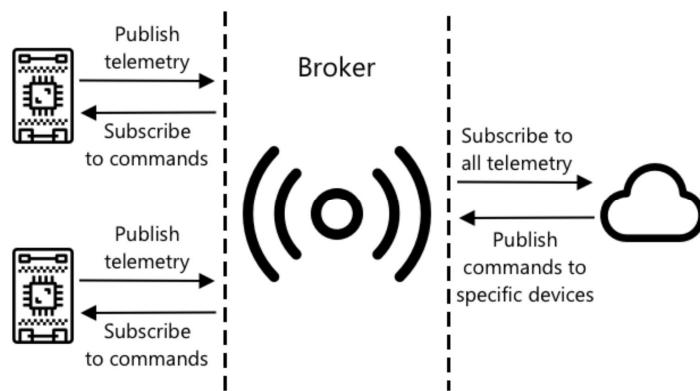
La “I” in IoT sta per Internet—connettività cloud e servizi che abilitano molte delle funzioni dei dispositivi IoT, dalla raccolta di misure dei sensori collegati ai dispositivi all’invio di messaggi agli attuatori di controllo. I dispositivi IoT in genere si collegano a un singolo servizio cloud IoT utilizzando protocolli di comunicazione standard e questo servizio è strettamente integrato con il resto dell’applicazione IoT, dai servizi di intelligenza artificiale che prendono decisioni intelligenti sui dati alle applicazioni Web per il controllo o la creazione di report.

 I dati raccolti dai sensori e inviati al cloud sono chiamati telemetria.

I dispositivi IoT possono anche ricevere informazioni dal cloud. Queste informazioni in genere sono costituite da comandi, istruzioni per eseguire azioni interne (come il riavvio o l’aggiornamento del firmware) o per azionare (ad esempio, accendere una luce).

#### 3.5.1.2 Protocolli di Comunicazione

Esistono molti protocolli di comunicazione popolari che i dispositivi IoT utilizzano per comunicare con Internet. I più popolari si basano sulla pubblicazione/sottoscrizione di messaggi tramite un agente: i dispositivi IoT si collegano all’agente, pubblicano dati di telemetria e si iscrivono ai comandi. Anche i servizi cloud si collegano all’agente, si iscrivono a tutte le informazioni di telemetria e pubblicano comandi su dispositivi specifici o gruppi di dispositivi, come mostrato nella figura seguente.



MQTT è il protocollo di comunicazione più diffuso per i dispositivi IoT e verrà trattato in questa sezione. Altri protocolli includono AMQP e HTTP/HTTPS, che abbiamo introdotto nella sezione precedente.

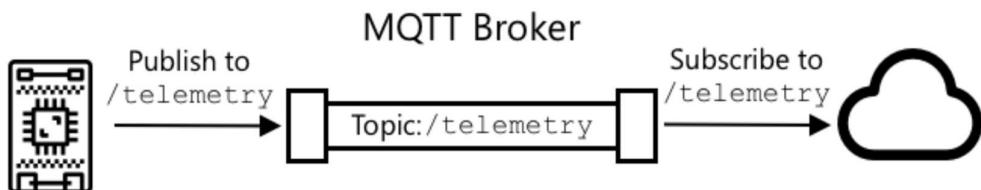
#### 3.5.1.3 Message Queuing Telemetry Transport (MQTT)

MQTT è l’abbreviazione di Message Queuing Telemetry Transport. È un protocollo di messaggistica basato sul paradigma publish/subscribe secondo lo standard ISO: ISO/IEC PRF 20922. Può essere visto come un “ponte per la distribuzione dei dati”. Opera in cima allo stack di protocolli TCP/IP ed è un protocollo di messaggistica di tipo publish/subscribe progettato per dispositivi remoti con scarse prestazioni hardware e scarse condizioni di rete. È un protocollo standard di trasporto di messaggistica leggero e open che può inviare messaggi tra dispositivi. Originariamente progettato nel 1999 per il monitoraggio degli oleodotti, è stato pubblicato come standard open da IBM 15 anni dopo.

Il vantaggio principale di MQTT è che fornisce un servizio di messaggistica affidabile e in tempo reale per la connessione di dispositivi remoti con codice minimo e larghezza di banda

limitata. Come protocollo di comunicazione istantanea a basso overhead e basso consumo di larghezza di banda, è ampiamente utilizzato in IoT, piccoli dispositivi, applicazioni mobili e così via.

MQTT ha un broker e più client. Tutti i client si collegano al broker, che quindi instrada i messaggi ai client pertinenti. I messaggi vengono instradati utilizzando argomenti denominati, non inviati direttamente a un singolo client. I client possono pubblicare su un argomento e qualsiasi client abbonato a quell'argomento riceverà il messaggio.



- Facciamo delle ricerche. Se si ha un gran numero di dispositivi IoT, come si può essere sicuri che il broker MQTT possa gestire tutti i messaggi?

### Alcuni Broker MQTT Open Source

Sebbene possiamo configurare il nostro broker MQTT se le circostanze lo consentono, si potrebbe non essere ancora pronti per approfondire la configurazione di server e applicazioni. Se si sta imparando, si può iniziare con alcuni broker MQTT open source.

#### Eclipse Mosquitto <https://www.mosquitto.org/>

Questo è un broker MQTT open source. Invece di occuparsi delle complessità di configurazione di un broker MQTT come parte di questa attività, questo broker di prova è disponibile pubblicamente su [test.mosquitto.org](https://test.mosquitto.org) e non richiede la configurazione di un account. È un ottimo strumento per testare client e server MQTT.

The screenshot shows the official website for Eclipse Mosquitto. At the top, there's a navigation bar with links for Home, Blog, Download, and Documentation. Below the header, there's a brief introduction: "Eclipse Mosquitto™ An open source MQTT broker". The page then details the project's purpose: "Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers." It highlights the MQTT publish/subscribe model and the availability of a C library for clients. On the right side, there are logos for the Eclipse Foundation and Cedalo, along with text about their involvement in the project. At the bottom, there's a "Download and Security" section.

#### Download and Security

Mosquitto is highly portable and available for a wide range of platforms.

- Go to the dedicated [download page](#) to find the source or binaries for your platform.
- Read the [Change Log](#) to find out about recent releases.
- Use the [security page](#) to find out how to report vulnerabilities or responses to past security issues.

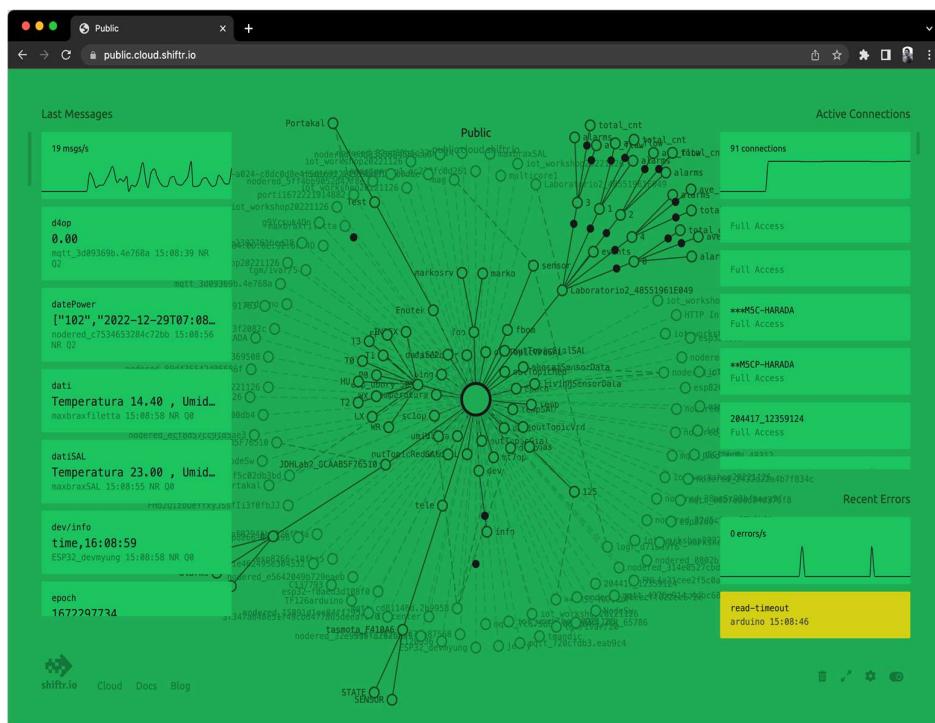
#### Test

You can have your own instance of Mosquitto running in minutes, but to make testing even easier, the Mosquitto Project runs a test server at [test.mosquitto.org](https://test.mosquitto.org) where you can test your clients in a variety of ways: plain MQTT, MQTT over TLS, MQTT over TLS (with [client certificate](#)), MQTT over WebSockets and MQTT over WebSockets with TLS.

Alternatively, Cedalo offer a [two week free hosted trial instance](#).

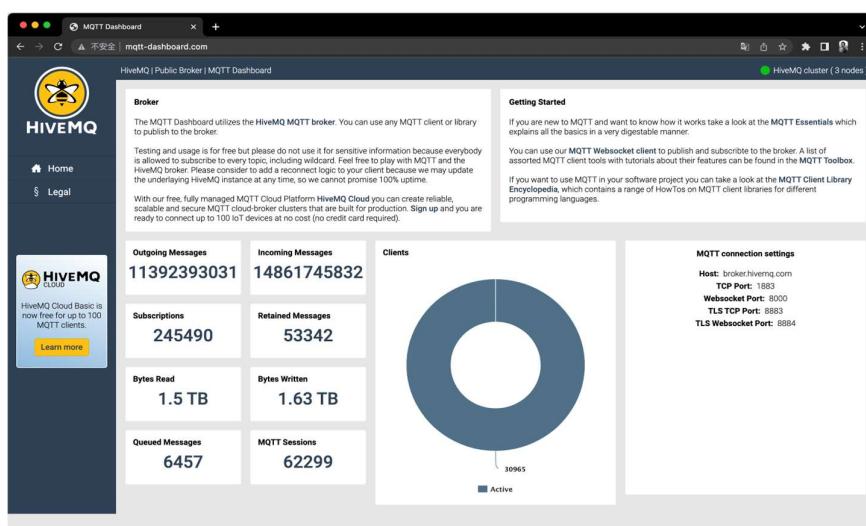
## shiftr.io

Una piattaforma IoT per progetti interconnessi, collega rapidamente hardware e software con il suo servizio cloud e le applicazioni desktop. La piattaforma fornisce anche una chiara visione di tutte le connessioni, argomenti e messaggi nella rete tramite grafici in tempo reale. Il broker shiftr.io supporta MQTT e HTTP per la pubblicazione, la sottoscrizione e il recupero dei messaggi e la piattaforma supporta account gratuiti, sufficienti per imparare e utilizzare. Forniscono anche un server pubblico su [public.cloud.shiftr.io](http://public.cloud.shiftr.io) con nome utente `public` sulle porte 1883 (MQTT) e 8883 (MQTTS). La vista animata dei servizi connessi e dei dati scambiati sul server pubblico è molto interessante, come mostrato nell'immagine sottostante.



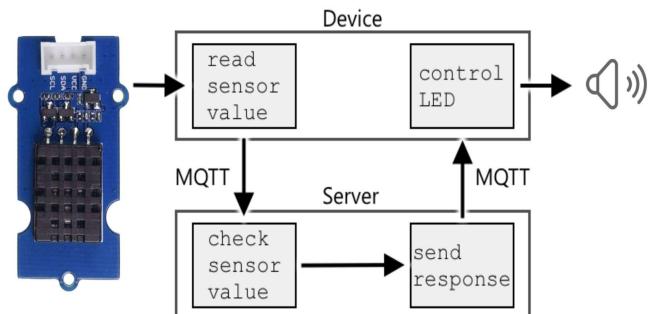
## HiveMQ

HiveMQ è una piattaforma MQTT basata su cloud, che offre servizi di comunicazione IoT scalabili, sicuri e affidabili. HiveMQ può aiutare aziende e sviluppatori a creare e gestire rapidamente applicazioni IoT, supportando milioni di dispositivi e messaggi.



### 3.5.2 Attività 1: Collegare XIAO ESP32C3 al Broker MQTT

Il primo passaggio per aggiungere il controllo Internet al misuratore di temperatura e umidità intelligente è collegare XIAO ESP32C3 a un broker MQTT. In questa parte della sezione, collegheremo il misuratore di temperatura e umidità intelligente della Sezione 2.2 a Internet, consentendogli di fornire telemetria ed essere controllato da remoto. Più avanti in questa sezione, il dispositivo invierà un messaggio di telemetria tramite MQTT a un broker MQTT pubblico, che verrà ricevuto da un codice server che scriveremo. Questo codice controllerà i valori di temperatura e umidità e invierà un comando al dispositivo, dicendogli di accendere o spegnere un cicalino.



Un utilizzo reale di questa configurazione potrebbe essere in un ampio spazio interno con molti sensori di temperatura e umidità, come una fattoria. Prima di decidere di accendere l'aria condizionata, i dati possono essere raccolti da più sensori di temperatura e umidità. Se solo una lettura del sensore supera la soglia, ma le letture degli altri sensori sono normali, questo può impedire l'accensione dell'intero sistema di aria condizionata.

- Riiscrite a pensare ad altre situazioni in cui è richiesta una valutazione dei dati da più sensori prima di impartire un comando?

 Ricordare, questo broker di test è pubblico e non sicuro e chiunque può ascoltare ciò che si pubblica, quindi non dovrebbe essere utilizzato per dati che devono essere mantenuti riservati.

Seguire i passaggi correlati di seguito per connettere il dispositivo al broker MQTT che introdotto in precedenza: [public.cloud.shiftr.io](https://public.cloud.shiftr.io).

#### Aggiungere la libreria `arduino-mqtt library`

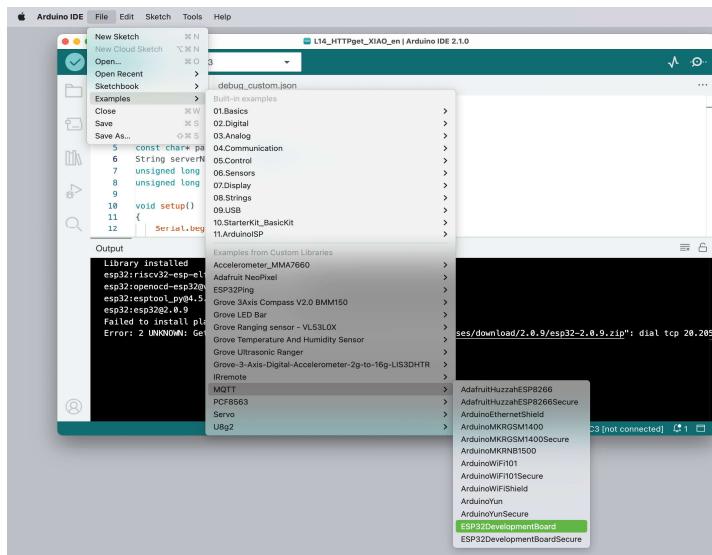
Prima di iniziare a programmare XIAO ESP32C3 con l'IDE Arduino, si devono aggiungere le librerie necessarie. Digita l'URL della libreria  <https://github.com/256dpi/arduino-mqtt> nella barra degli indirizzi del browser per andare alla pagina GitHub. Cliccare su `Code - Download ZIP` per scaricare il pacchetto di risorse `arduino-mqtt-master.zip` sul computer locale, come mostrato nell'immagine qui sotto.

Dalla barra dei menù, selezionare **Sketch**→**Include Library**→**Add .ZIP Library** per aggiungere il pacchetto di risorse **arduino-mqtt-master.zip** scaricato nel passaggio precedente. Continuare finché non si vede un messaggio che indica il caricamento riuscito della libreria.

### Eseguire l'esempio ESP32 MQTT

Dopo che la libreria è stata caricata correttamente, aprire l'esempio "ESP32DevelopmentBoard" nell'IDE Arduino tramite il seguente path:

**File**→**Examples**→**MQTT**→**ESP32DevelopmentBoard**, come mostrato nell'immagine sottostante.



Dopo che il programma di esempio è stato aperto, si può vedere il programma come mostrato di seguito. Modificare, quindi, **ssid** nel codice con il nome della tua rete Wi-Fi e modificare **pass** nel codice con la password Wi-Fi corrispondente per la rete Wi-Fi.

```
// This example uses an ESP32 Development Board
// to connect to shiftr.io.
//
// You can check on your device after a successful
// connection here: https://www.shiftr.io/try.
//
// by Joël Gähwiler
// https://github.com/256dpi/arduino-mqtt
```

```

#include <WiFi.h>
#include <MQTT.h>

const char ssid[] = "ssid";
const char pass[] = "pass";

WiFiClient net;
MQTTClient client;

unsigned long lastMillis = 0;

void connect() {
    Serial.print("checking wifi...");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
    }
}

Serial.print("\nconnecting...");
while (!client.connect("arduino", "public", "public")) {
    Serial.print(".");
    delay(1000);
}

Serial.println("\nconnected!");

client.subscribe("/hello");
// client.unsubscribe("/hello");
}

void messageReceived(String &topic, String &payload) {
    Serial.println("incoming: " + topic + " - " + payload);

    // Note: Do not use the client in the callback to publish, subscribe or
    // unsubscribe as it may cause deadlocks when other things arrive while
    // sending and receiving acknowledgments. Instead, change a global variable,
    // or push to a queue and handle it in the loop after calling `client.loop()`.

}

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, pass);

    // Note: Local domain names (e.g. "Computer.local" on OSX) are not supported
    // by Arduino. You need to set the IP address directly.
    client.begin("public.cloud.shiftr.io", net);
    client.onMessage(messageReceived);

    connect();
}

void loop() {
    client.loop();
    delay(10); // <- fixes some issues with WiFi stability

    if (!client.connected()) {
        connect();
    }

    // publish a message roughly every second.
    if (millis() - lastMillis > 1000) {
        lastMillis = millis();
        client.publish("/hello", "world");
    }
}

```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15\\_ESP32DevelopmentBoard\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15_ESP32DevelopmentBoard_XIAO_en)

Eseguire l'esempio e controllare il monitor seriale per: `connected!`. Se si vede un client connesso e un flusso di messaggi nel grafico live, XIAO invia continuamente dati a questo broker MQTT pubblico!

The screenshot shows the Arduino IDE interface with the following details:

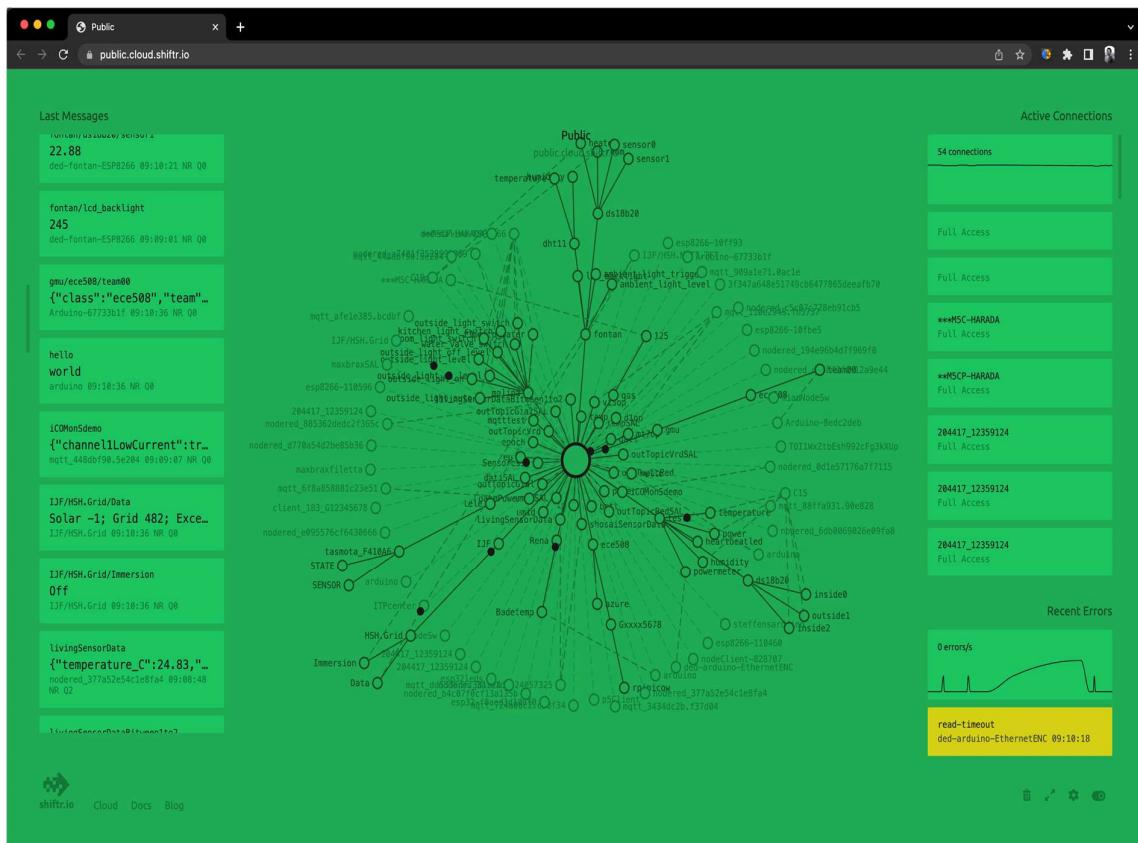
- Title Bar:** L15\_ESP32DevelopmentBoard\_XIAO\_en | Arduino IDE 2.1.0
- File:** L15\_ESP32DevelopmentBoard\_XIAO\_en.ino
- Code Area:** The code is for an ESP32 connected to a Xiao module. It includes a WiFiClient and MQTTClient, and a connect() function that prints "connecting..." to the Serial Monitor.

```
16 WiFiClient net;
17 MQTTClient client;
18
19 unsigned long lastMillis = 0;
20
21 void connect() {
22     Serial.print("checking wifi...");
23     while (WiFi.status() != WL_CONNECTED) {
24         Serial.print(".");
25         delay(1000);
26     }
27
28     Serial.print("\nconnecting...");
```

- Output Area:** Shows the Serial Monitor output. The device is connecting to WiFi and receiving multiple "hello - world" messages from an incoming connection.

```
Message (Enter to send message to 'XIAO_ESP32C3' on '/dev/cu.usbmodem101')
08:59:36.239 -> connecting...
08:59:36.239 -> connected!
08:59:36.239 -> incoming: /hello - world
08:59:36.603 -> incoming: /hello - world
08:59:37.627 -> incoming: /hello - world
08:59:38.551 -> incoming: /hello - world
```

Si possono vedere i messaggi inviati accedendo a [public.cloud.shiftr.io](http://public.cloud.shiftr.io) nel browser. Tuttavia, poiché si tratta di un broker pubblico, il dispositivo si perderà rapidamente nella folla.





Da tenere presente che questo broker di prova è pubblico e non sicuro. Chiunque può ascoltare ciò che viene pubblicato, quindi non dovrebbe essere utilizzato per nulla che richieda riservatezza.

### 3.5.3 Approfondimento su MQTT

Gli argomenti possono avere una gerarchia e i client possono utilizzare caratteri jolly per iscriversi a diversi livelli di gerarchie diverse. Ad esempio: si può inviare la telemetria della temperatura all'argomento `/telemetry/temperature`, i dati sull'umidità all'argomento a `/telemetry/humidity` e quindi iscriversi all'argomento `/telemetry/*` nell'applicazione cloud per ricevere sia la telemetria della temperatura che quella dell'umidità. Quando vengono inviati messaggi, è possibile specificare una qualità del servizio (QoS) che determina la garanzia di recapito del messaggio.

- Al massimo una volta: Il messaggio viene inviato una sola volta e il client e il broker non eseguono ulteriori passaggi per confermare il recapito (Fire and Forget).
- Almeno una volta: Il messaggio viene ritentato dal mittente finché non riceve una conferma (Acknowledged delivery).
- Esattamente una volta: Il mittente e il destinatario eseguono un handshake a due livelli per garantire che venga ricevuta solo una copia del messaggio (Assured delivery).



In quali scenari si potrebbe aver bisogno di recapitare messaggi su base “Fire and Forget”?

Sebbene MQTT (Message Queuing Telemetry Transport) abbia “Message Queuing” nel suo nome (le prime due lettere di MQTT), in realtà non supporta le code di messaggi. Ciò significa che se un client si disconnette e poi si riconnette, non riceverà i messaggi inviati mentre era disconnesso, ad eccezione di quei messaggi che aveva già iniziato a elaborare utilizzando il processo QoS. È possibile impostare un flag di conservazione su un messaggio. Se questo flag è impostato, il broker MQTT memorizzerà l'ultimo messaggio inviato su un argomento con questo flag e lo invierà a tutti i client che successivamente si iscriveranno a quell'argomento. In questo modo, i client riceveranno sempre il messaggio più recente. MQTT supporta anche una funzionalità keep-alive per verificare se la connessione è ancora online durante lunghi intervalli tra i messaggi. Le connessioni MQTT possono essere pubbliche o crittografate e protette tramite nomi utente, password o certificati.



MQTT comunica tramite TCP/IP, lo stesso protocollo di rete sottostante di HTTP, ma su una porta diversa. Si può anche comunicare con applicazioni web in esecuzione in un browser tramite MQTT su websocket o in situazioni in cui firewall o altre regole di rete bloccano le connessioni MQTT standard.

### 3.5.4 Telemetria

La parola “telemetria” deriva da radici greche che significano “misura remota”. La telemetria si riferisce all’atto di raccogliere dati dai sensori e inviarli al cloud.



Uno dei primi dispositivi di telemetria fu inventato in Francia nel 1874, inviando dati meteorologici e sulla profondità della neve in tempo reale dal Monte Bianco a Parigi. Poiché all’epoca non esisteva la tecnologia wireless, utilizzava un cavo fisico.

Torniamo all’esempio del termostato intelligente della Sezione 1.1.

Architettura del Sistema del Termostato Intelligente

Il termostato ha sensori di temperatura per raccogliere dati di telemetria. Probabilmente ha un sensore di temperatura integrato e può connettersi a più sensori di temperatura esterni tramite protocolli wireless come Low Energy Bluetooth (BLE).

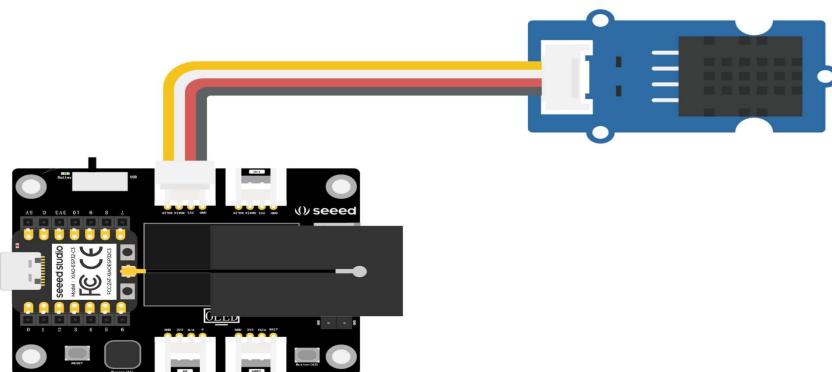
Un esempio dei dati di telemetria che invia potrebbe essere:

| Nome                    | Valore | Descrizione                                                                                                                        |
|-------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------|
| AC_Temperature          | 18°C   | Temperatura misurata dal sensore di temperatura integrato nel termostato                                                           |
| Living_Room_Temperature | 19°C   | Temperatura misurata da un sensore di temperatura remoto denominato <code>livingroom</code> , che indica la stanza in cui si trova |
| Bedroom_Temperature     | 21°C   | Temperatura misurata da un sensore di temperatura remoto denominato <code>bedroom</code> , che indica la stanza in cui si trova    |

Quindi, il servizio cloud può utilizzare questi dati di telemetria per decidere quali comandi inviare per controllare il raffreddamento o il riscaldamento.

### 3.5.5 Attività 2: Invio di Informazioni di Telemetria da XIAO al Broker MQTT

La fase successiva dell'aggiunta del controllo Internet all'igrotermografo intelligente è l'invio dei dati di telemetria di temperatura e umidità all'argomento di telemetria del broker MQTT. Sostituire lo XIAO del dispositivo igrotermografo intelligente dalla Sezione 2.2 con XIAO ESP32C3, come mostrato nell'immagine sottostante.



Caricare il seguente programma nell'IDE Arduino per testare l'invio di dati di telemetria dal dispositivo al broker MQTT. Notare che in questo esempio stiamo provando un broker MQTT diverso da quello dell'attività 1: `broker.hivemq.com` e abbiamo impostato `xiao_ESP32C3_Telemetry/` come nome di sottoscrizione.

```
/////////
// IDE:
// Arduino 2.0.0
// Platform:
// esp32 2.0.5 - https://github.com/espressif/arduino-esp32
// Board:
// XIAO_ESP32C3
// Libraries:
// MQTT 2.5.0 - https://github.com/knolleary/pubsubclient
// ArduinoJson 6.19.4 - https://github.com/bblanchon/ArduinoJson

/////////
// Includes

#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include "DHT.h"
#define DHATYPE DHT20
DHT dht(DHATYPE);

const char* ssid = "ssid";
```

```
const char* password = "pass";
const char* mqtt_server = "broker.hivemq.com";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

float temperature = 0;
float humidity = 0;

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    Wire.begin();
    dht.begin();
}

void setup_wifi() {
    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("XIAO_ESP32")) {
            Serial.println("connected");
            // Subscribe
            client.subscribe("XIAO_ESP32/LEDOUTPUT");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }

    client.loop();

    long now = millis();
    float temp_hum_val[2] = {0};
    if (now - lastMsg > 5000) {
        lastMsg = now;

        dht.readTempAndHumidity(temp_hum_val);
        temperature = temp_hum_val[1];

        char tempString[8];
        dtostrf(temperature, 1, 2, tempString);
        Serial.print("Temperature: ");
        Serial.println(tempString);
        client.publish("XIAO_ESP32C3_Telemetry/Temperaturedataread", tempString);
    }
}
```

```

humidity = temp_hum_val[0];

char humString[8];
dtostrf(humidity, 1, 2, humString);
Serial.print("Humidity: ");
Serial.println(humString);
client.publish("XIAO_ESP32_Telemetry/Humiditydataread", humString);
}

}

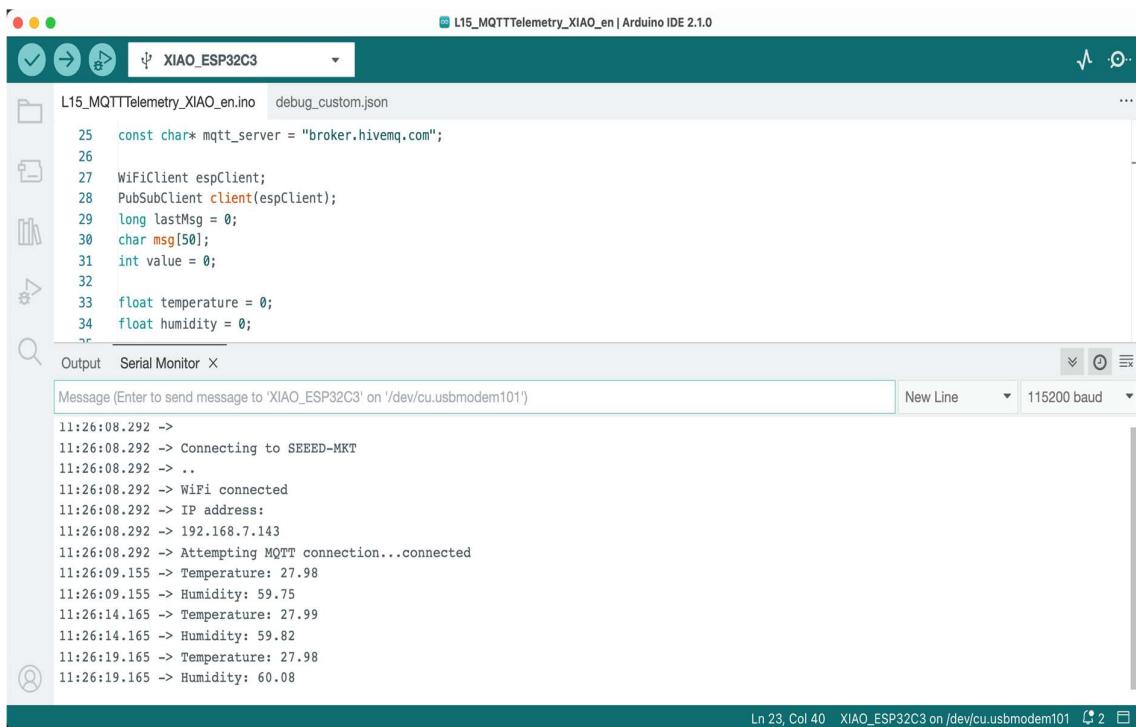
```

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15\\_MQTTTelemetry\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15_MQTTTelemetry_XIAO_en)

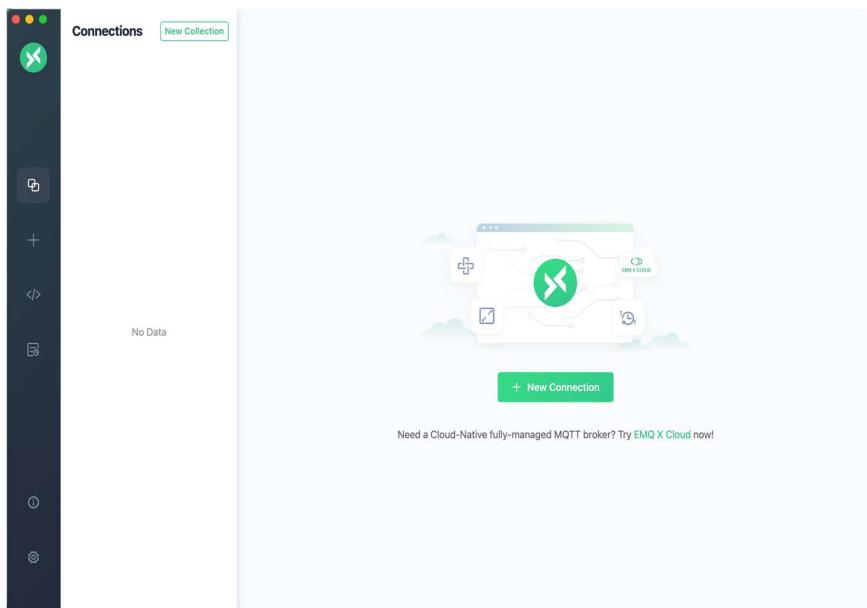
Poiché questo esempio si basa sulla libreria `PubSubClient.h`, se si prova a compilarla direttamente, si incontrerà l'errore “**PubSubClient.h: No such file or directory**”. Per risolvere questo problema, seguire i passaggi sottostanti per installare la libreria.

1. Aprire l'IDE Arduino.
2. Andare su “Sketch” -> “Include Library” -> “Manage Libraries”.
3. Nel Library Manager, digitare “PubSubClient” nella barra di ricerca.
4. Cercare la libreria “PubSubClient” di Nick O’Leary e cliccarci sopra.
5. Cliccare sul pulsante “Install” per installare la libreria.

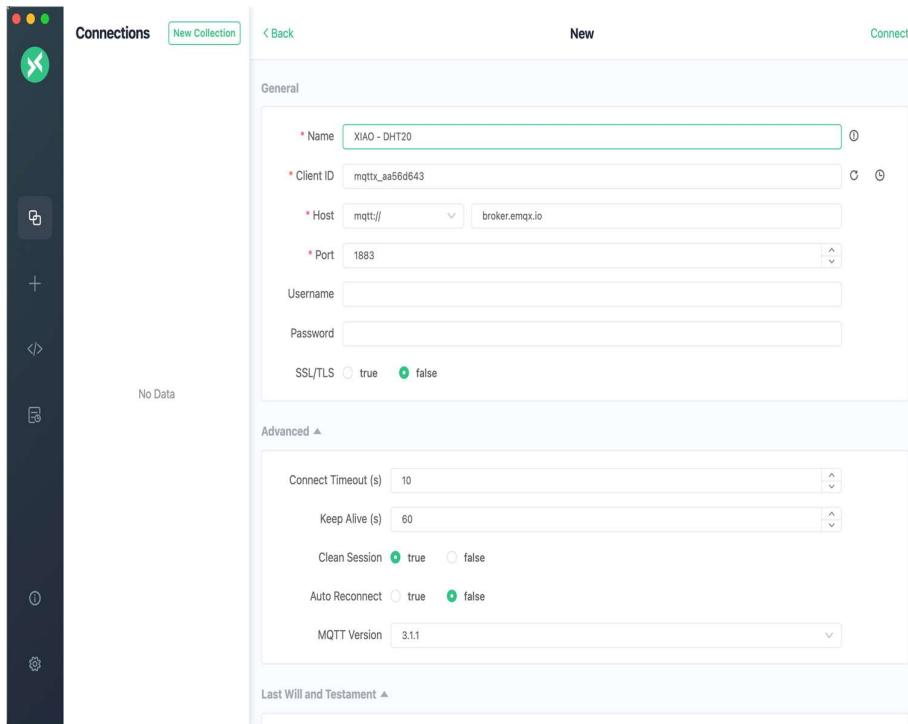
Quindi modificare `ssid` nel codice con il nome della rete Wi-Fi e `pass` con la tua password Wi-Fi corrispondente. Dopo aver caricato correttamente il programma, aprire il monitor seriale. Se tutto va bene, si vedrà il dispositivo iniziare a inviare dati di temperatura e umidità, come mostrato nell’immagine qui sotto.



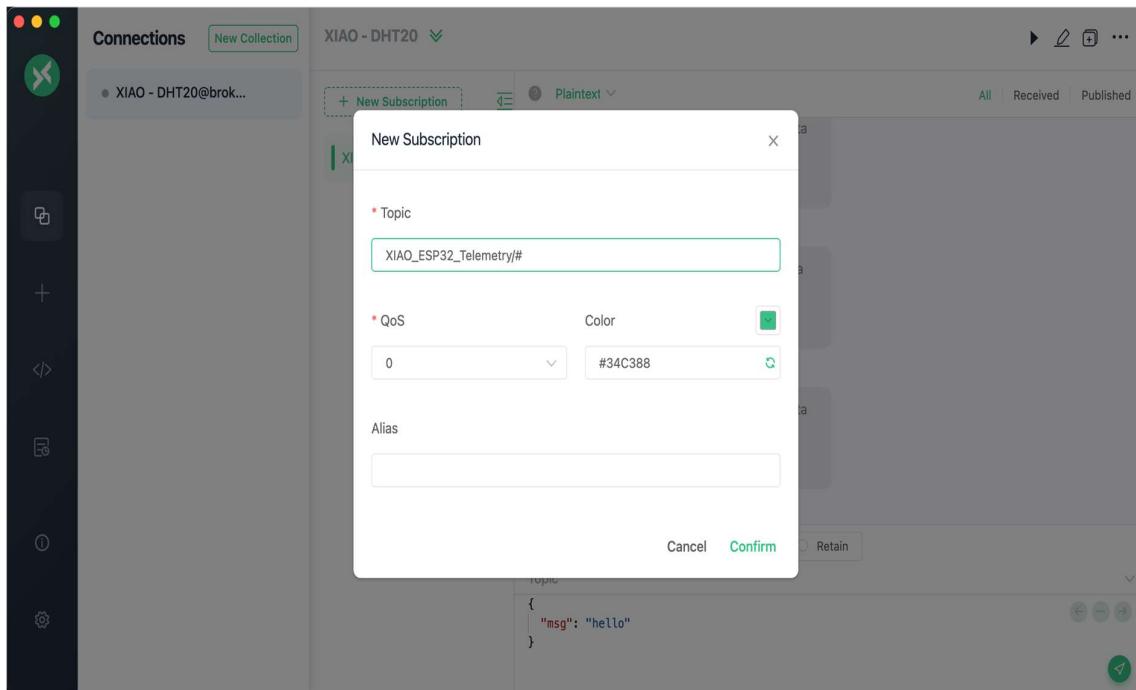
Come si possono vedere i dati del sensore da un’altra piattaforma? Ci sono molti modi, come [MQTT X](#). Dopo aver scaricato e installato il software adatto al proprio sistema PC, l’interfaccia è come mostrato nell’immagine qui sotto.



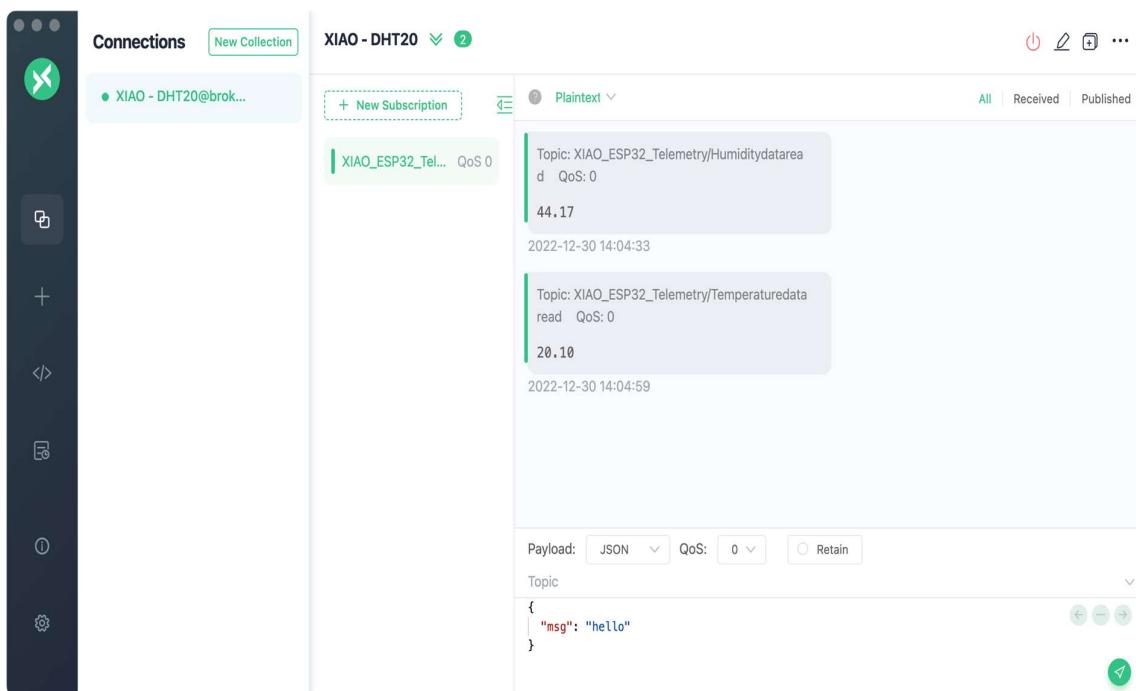
Cliccando sul pulsante `+ New connection` si verrà indirizzati alla finestra di creazione della connessione, come mostrato nell'immagine qui sotto. Si inserisce `XIAO-DHT20` nella casella `Name` come nome della connessione. L'host è `broker.hivemq.com` che abbiamo impostato nel programma, non sono necessarie altre impostazioni, cliccare su `Connect` nell'angolo in alto a destra.



Creare un nuovo abbonamento, mostrando tutte le informazioni in `XIAO_ESP32C3_Telemetry/`, come mostrato nell'immagine sottostante.



Ora possiamo vedere i dati di telemetria inviati da XIAO ESP32C3, come mostrato nell'immagine sottostante.



### Con quale frequenza deve essere inviata la telemetria?

Una domanda che richiede un'attenta considerazione con la telemetria è: con quale frequenza si devono misurare e inviare i dati? La risposta è: dipende dalle esigenze del dispositivo monitorato e dall'attività da svolgere. Se si eseguono misure frequenti, si può effettivamente rispondere più rapidamente ai cambiamenti, ma ciò causerebbe al dispositivo un consumo di più energia, più larghezza di banda, la generazione di più dati e richiederebbe più risorse cloud per essere gestito. Si deve trovare un equilibrio tra misure abbastanza frequenti ma non troppo.

Per un termostato, la misura ogni pochi minuti potrebbe essere sufficiente perché è improbabile che la temperatura cambi frequentemente. Se si misura solo una volta al giorno, si potrebbe riscaldare la casa per le temperature notturne in una giornata di sole, e se si misura ogni secondo, co sarebbero migliaia di misure di temperatura ripetute e non necessarie che consumerebbero la velocità e la larghezza di banda di Internet degli utenti (il che è un problema per le persone con piani di larghezza di banda limitati) e consumerebbero anche più energia, il che è un problema per dispositivi come i sensori remoti che si basano sull'alimentazione a batteria, e aumenterebbero ulteriormente il costo delle risorse di cloud computing per elaborarli e archiviarli.

Se si stanno monitorando i dati attorno a una macchina in una fabbrica che potrebbero causare danni catastrofici e milioni di mancati guadagni in caso di guasto, allora potrebbe essere necessario misurare più volte al secondo. Sprecare larghezza di banda è meglio che perdere dati di telemetria che potrebbero segnalare la necessità di fermarsi e riparare prima che una macchina si guasti.



In questa situazione, si potrebbe considerare l'uso un dispositivo edge per gestire i dati di telemetria per ridurre la dipendenza da Internet.

## Perdita di connessione

Le connessioni Internet possono essere inaffidabili ed è comune perdere il segnale. In questo caso, cosa dovrebbe fare un dispositivo IoT? Dovrebbe perdere dati o dovrebbe memorizzarli finché la connessione non viene ripristinata? Di nuovo, la risposta è: dipende dal dispositivo monitorato.

Per un termostato, è probabile che i dati vengano persi una volta effettuata una nuova misura della temperatura. Se la temperatura attuale è di 19 °C, al sistema di riscaldamento non importa che la temperatura di 20 minuti fa fosse di 20,5 °C; è la temperatura attuale a determinare se il riscaldamento deve essere acceso o spento.

Per alcune macchine, si potrebbe voler conservare questi dati, soprattutto se vengono utilizzati per cercare tendenze. Alcuni modelli di apprendimento automatico possono identificare anomalie nei flussi di dati esaminando un periodo di tempo definito (ad esempio, l'ultima ora). Questo viene spesso utilizzato per la manutenzione predittiva, alla ricerca di segnali che qualcosa potrebbe essere sul punto di guastarsi in modo da poterlo riparare o sostituire prima che si verifichi un disastro. Si potrebbe voler inviare ogni punto di telemetria da una macchina in modo che possa essere utilizzato per il rilevamento delle anomalie, quindi una volta che un dispositivo IoT può riconnettersi, invierà tutti i dati di telemetria generati durante l'interruzione di Internet.

I progettisti di dispositivi IoT dovrebbero anche considerare se un dispositivo IoT può funzionare durante un'interruzione di Internet o se perde il segnale a causa della posizione. Se un termostato intelligente non è in grado di inviare dati di telemetria al cloud a causa di un'interruzione di Internet, dovrebbe essere in grado di prendere alcune decisioni limitate per controllare il riscaldamento.

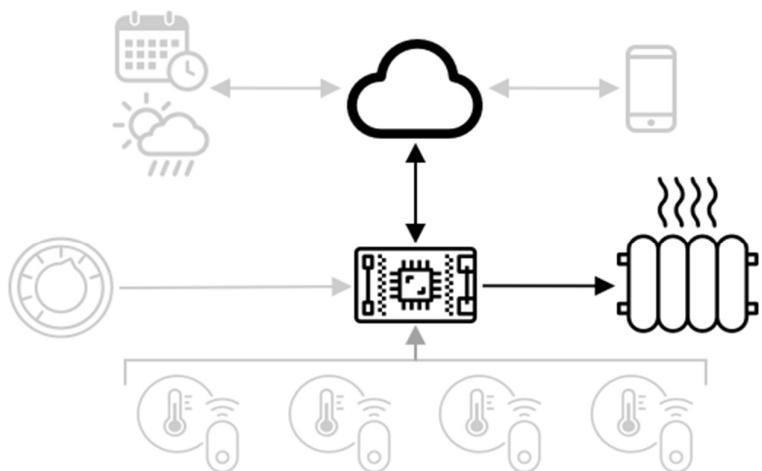


Questa Ferrari è diventata un mattone quando qualcuno ha provato ad aggiornarla in un parcheggio sotterraneo... ma lì non c'era segnale cellulare.

Per MQTT che gestisce le interruzioni di connessione, se necessario, il dispositivo e il codice del server dovranno essere responsabili di garantire la consegna dei messaggi, ad esempio, richiedendo che tutti i messaggi inviati ricevano risposta con un messaggio aggiuntivo nell'argomento di risposta e, in caso contrario, di metterli manualmente in coda per un successivo reinvio.

### 3.5.6 Comandi

I comandi sono messaggi inviati dal cloud a un dispositivo con cui gli ordinano di fare qualcosa. Nella maggior parte dei casi, ciò comporta la fornitura di un output tramite un attuatore, ma potrebbe trattarsi di un'istruzione al dispositivo stesso, come il riavvio o la raccolta di dati di telemetria aggiuntivi e l'invio di tali dati come risposta al comando.



Un termostato potrebbe ricevere un comando dal cloud per accendere il riscaldamento. In base ai dati di telemetria di tutti i sensori, se il servizio cloud ha deciso che il riscaldamento deve essere acceso, invia il comando appropriato.

### 3.5.7 Attività 3: Inviare Comandi a XIAO Tramite Broker MQTT

Dopo aver padroneggiato la telemetria, il passaggio successivo consiste nell'inviare comandi ai dispositivi IoT tramite un broker MQTT. In questa attività, proveremo a usare un computer con broker MQTT, spesso chiamato computer host, per inviare caratteri specifici e lasciare che il Wi-Fi connesso XIAO ESP32C3 controlli un cicalino collegato a una scheda di espansione per emettere un suono.

Nell'IDE Arduino, caricare il seguente programma per testare l'invio di caratteri specifici (il primo carattere è '0') dal broker MQTT per attivare il cicalino. In questo esempio utilizziamo il broker MQTT: `broker.hivemq.com`.

```
///////////////////////////// // IDE: // Arduino 2.0.0 // Platform: // esp32 2.0.5 - https://github.com/espressif/arduino-esp32 // Board: // XIAO ESP32C3 // Libraries: // MQTT 2.5.0 - https://github.com/knolleary/pubsubclient // ArduinoJson 6.19.4 - https://github.com/bblanchon/ArduinoJson // https://github.com/Seeed-Studio/Seeed_Arduino_MultiGas //////////////////// // Includes

#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>

const char* ssid = "ssid";
const char* password = "pass";

const char* mqtt_server = "broker.hivemq.com";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

int speakerPin = A3;

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print(":");
  Serial.print(payload);
  Serial.print("] ");
  Serial.print((char*)payload);
  Serial.println();
}
```

```

    Serial.print(topic);
    Serial.print("] ");
    for (int i=0;i<length;i++) {
        Serial.print((char)payload[i]);
    }

    if((char)payload[0]=='0'){
        Serial.print(" RUN");
        digitalWrite(speakerPin, HIGH);
        delay(2000);
        digitalWrite(speakerPin, LOW);
        delay(100);
    }

    Serial.println();
}

void setup() {
    Serial.begin(115200);
    pinMode(speakerPin, OUTPUT);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.subscribe("XIAO_ESP32/Recieve");
    client.setCallback(callback);
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("XIAO_ESP32")) {
            Serial.println("Connected");
            // Subscribe
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
        client.subscribe("XIAO_ESP32/Recieve");
    }
    client.loop();
}

```

---

Prelevare questo programma da Github [https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15\\_MQTTCommand\\_XIAO\\_en](https://github.com/mouseart/XIAO-Mastering-Arduino-and-TinyML/tree/main/code/L15_MQTTCommand_XIAO_en)

Modificare `ssid` nel codice con il nome della rete Wi-Fi e modifica `pass` nel codice con la password Wi-Fi corrispondente. La logica dell'esecuzione del programma è spiegata come segue:

---

```
client.setServer(mqtt_server, 1883);
client.subscribe("XIAO_ESP32/Recieve");
client.setCallback(callback);
```

---

Durante la fase di `setup`, la connessione tra XIAO e il server MQTT viene inizializzata e vengono impostate gli argomenti della sottoscrizione e le funzioni di callback. Qui ci iscriviamo all'argomento `xiao_ESP32/Recieve`. Come esempio. Quando inviamo un messaggio a questo argomento dal computer host, verrà eseguita la funzione di `callback` corrispondente:

---

```
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
```

```

Serial.print(topic);
Serial.print("] ");
for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
}

if((char)payload[0]=='0'){
    Serial.print(" RUN");
    digitalWrite(speakerPin, HIGH);
    delay(2000);
    digitalWrite(speakerPin, LOW);
    delay(100);
}

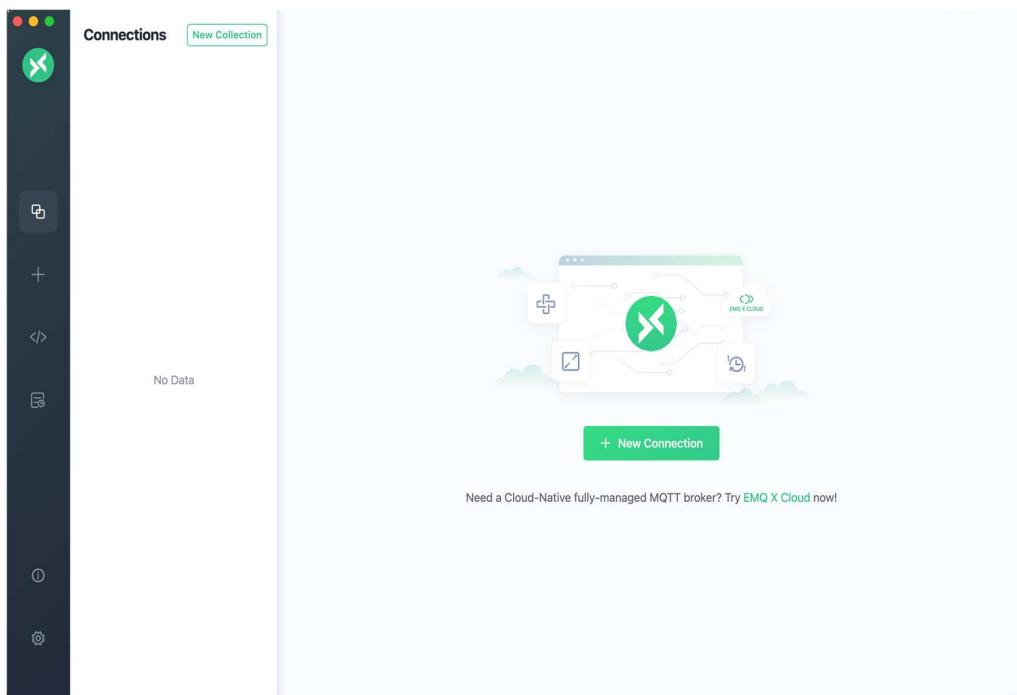
Serial.println();
}

```

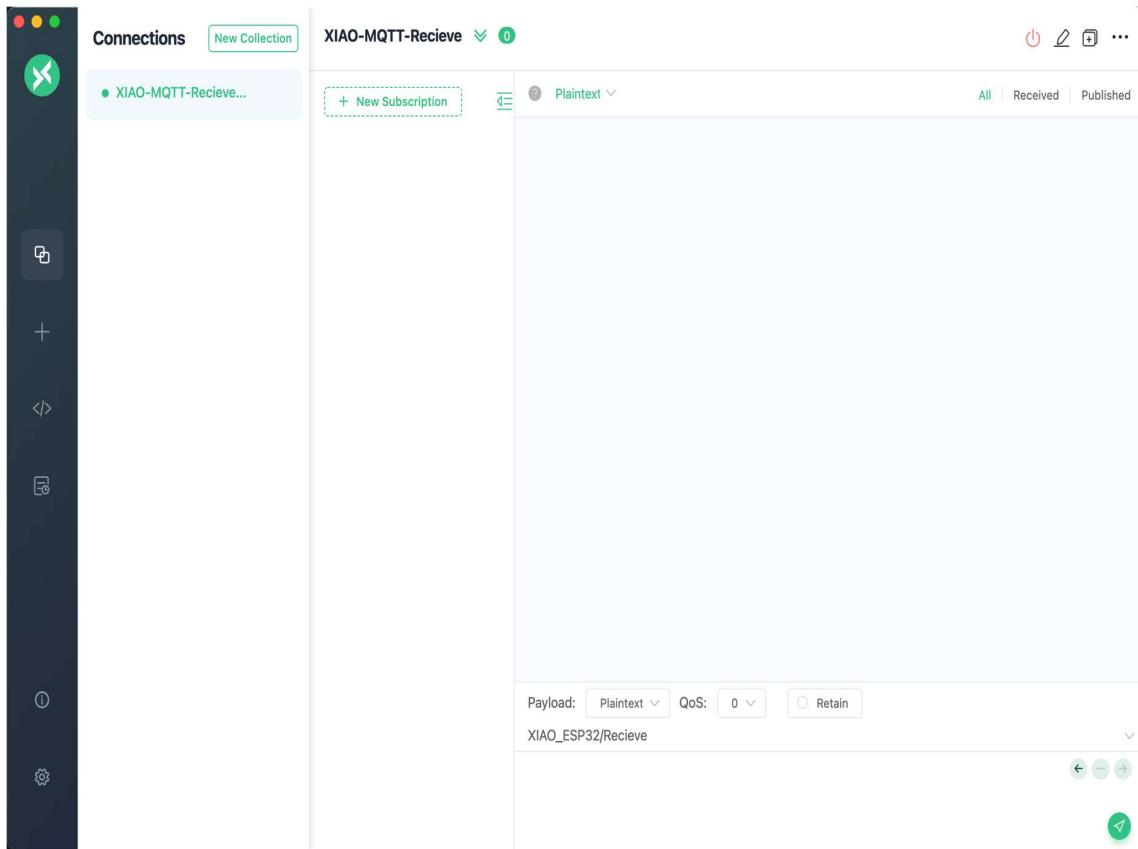
Qui prima stamperà il messaggio ricevuto, poi estrarrà il carattere in posizione 0. Quando il carattere in posizione 0, che è il primo carattere, è 0, soddisfa la condizione affinché l'istruzione `if` esegua un'azione. Qui colleghiamo insieme XIAO ESP32C3 e la scheda di espansione. Quando la condizione è soddisfatta, il cicalino sulla scheda di espansione cambierà brevemente il suo livello ed emetterà un segnale acustico per 2 secondi, mentre invia il messaggio di richiesta `RUN` alla porta seriale.

Nel processo di sviluppo e test da parte dei lettori, si può anche provare a integrare le funzioni di ricezione e invio di MQTT e inviare messaggi ad argomenti specifici nella funzione di callback, in modo che il mittente possa assicurarsi che XIAO abbia ricevuto il messaggio.

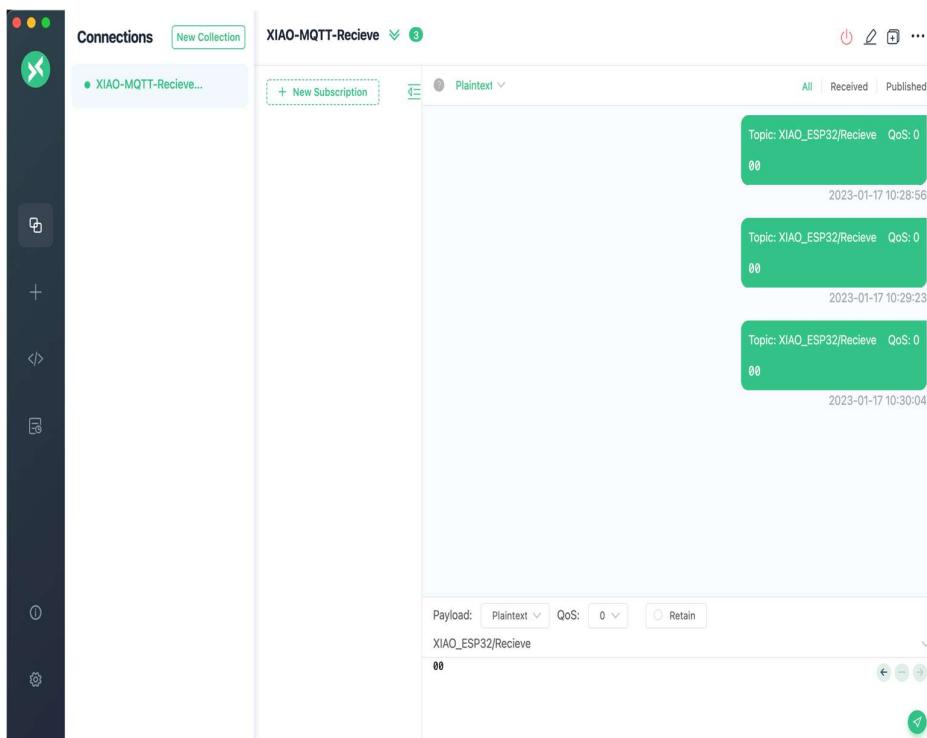
Sul computer host, utilizziamo **MQTT X** per il test. Aprire MQTT X, l'interfaccia è come mostrato nella figura seguente.



Cliccare sul pulsante `+ New Connection` per accedere alla finestra di creazione della connessione, come mostrato nella figura seguente. Compilare la casella Name con `xiao-MQTT-Receive` come nome della connessione. Host è `broker.hivemq.com` che abbiamo impostato nel programma e non è necessario impostare altro. Cliccare `Connect` nell'angolo in alto a destra. L'interfaccia dopo la connessione riuscita è come mostrato nella figura seguente.



Ora possiamo pubblicare messaggi nell'argomento specificato, che è l'argomento **XIAO\_ESP32/Recieve** a cui ci siamo iscritti su XIAO. Quindi inseriamo **00** nella casella di input di **XIAO\_ESP32/Recieve** nell'angolo in basso a destra dell'interfaccia, e clicchiamo sul pulsante **Send** nell'angolo in basso a destra.



A questo punto, nel monitor seriale sul lato PC, si vede il messaggio ricevuto da XIAO, come mostrato nella figura seguente, e il prompt `RUN`, il cicalino suonerà per 2 secondi, indicando che il messaggio è stato ricevuto.

The screenshot shows the Arduino IDE interface. The top bar displays "L15\_MQTTCommand\_XIAO\_en | Arduino IDE 2.1.0". The left sidebar shows the file structure with "L15\_MQTTCommand\_XIAO\_en.ino" selected. The code editor contains the following C++ code:

```

68     for (int i=0;i<length;i++) {
69         Serial.print((char)payload[i]);
70     }
71     if((char)payload[0]=='0'){
72         Serial.print(" RUN");
73         digitalWrite(speakerPin, HIGH);
74         delay(2000);
75         digitalWrite(speakerPin, LOW);
76         delay(100);
77     }
78     Serial.println();
79 }

```

The bottom section shows the "Serial Monitor" tab active. The message input field contains "Message (Enter to send message to 'XIAO\_ESP32C3' on '/dev/cu.usbmodem1101')". The output window displays the following log:

```

16:28:16.563 ->
16:28:16.563 -> Connecting to kong
16:28:17.001 -> .
16:28:17.001 -> WiFi connected
16:28:17.001 -> IP address:
16:28:17.001 -> 192.168.63.11
16:28:17.001 -> Attempting MQTT connection...connected
16:31:06.008 -> Message arrived [XIAO_ESP32/Recieve] 00 RUN

```

The status bar at the bottom right indicates "Ln 24, Col 33 XIAO\_ESP32C3 on /dev/cu.usbmodem1101" and "2 messages".

Ora, abbiamo pilotato con successo il cicalino sulla scheda di espansione collegata al Wi-Fi connesso XIAO ESP32C3 tramite l'istruzione inviata dal lato PC. L'azione del cicalino può essere sostituita con il controllo di qualsiasi periferica per ottenere la funzione desiderata.

### Connessione persa

Se un servizio cloud deve inviare un comando a un dispositivo IoT offline, cosa dovrebbe fare? Di nuovo, la risposta dipende dalla situazione. Se l'ultimo comando sovrascrive quello precedente, il comando precedente può essere ignorato. Se il servizio cloud invia un comando per accendere il riscaldamento e poi invia un altro comando per spegnerlo, il comando di accensione può essere ignorato e non deve essere reinviato. Se i comandi devono essere elaborati in ordine, ad esempio prima spostare il braccio del robot verso l'alto e poi chiudere la pinza, devono essere inviati in ordine una volta ripristinata la connessione.

- In che modo il codice del dispositivo o del server può garantire che i comandi vengano sempre inviati ed elaborati in ordine tramite MQTT, se necessario?

### Utilizzo della funzione Bluetooth di XIAO

XIAO nRF2840, XIAO nRF2840 Sense, XIAO ESP32C3 supportano tutte la funzione Bluetooth, si può fare riferimento ai documenti Wiki correlati per scoprire come utilizzare la funzione Bluetooth.

- [Utilizzo del Bluetooth su Seeed Studio XIAO ESP32C3](#)
- [Utilizzo del Bluetooth \(libreria schede Seeed nRF52\)](#)
- [Utilizzo del Bluetooth \(libreria schede nrf52 mbed-enabled\)](#)

## 4.1 Informazioni su TinyML ed Edge Impulse Studio

Questa sezione spiegherà l'apprendimento automatico embedded, le differenze tra TinyML e altre intelligenze artificiali e alcune applicazioni essenziali. Questa sezione aiuterà a capire cos'è TinyML e perché ne abbiamo bisogno. Edge Impulse Studio è uno degli strumenti che consente agli sviluppatori di creare soluzioni per dispositivi intelligenti di nuova generazione tramite l'apprendimento automatico embedded. Questa sezione introdurrà a questo strumento e aiuterà a comprendere i passaggi di base per creare un modello di apprendimento automatico embedded.

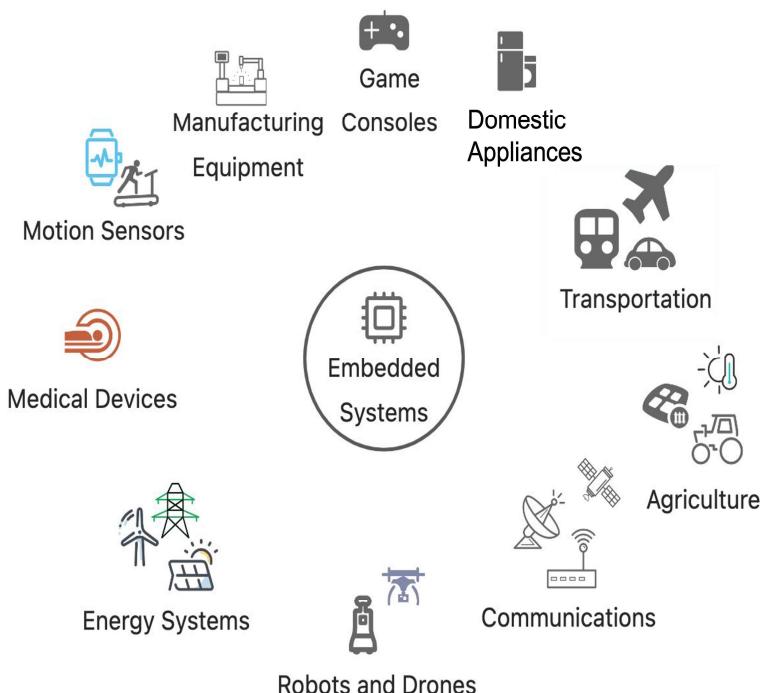
### 4.1.1 Termini Comuni

Oltre a TinyML, sentiamo spesso termini concettualmente simili come edge computing, edge AI, "machine learning" [apprendimento automatico] embedded, ecc. Quindi, prima di apprendere TinyML, si devono comprendere questi termini e il loro significato.

#### 4.1.1.1 Sistemi Embedded

Un sistema embedded è un computer utilizzato solo per risolvere alcuni problemi particolari ed è difficile da modificare. Il termine "embedded" [incorporato] significa che è integrato nel sistema. È una parte permanente del sistema più esteso. Di solito non sembra un computer; nella maggior parte dei casi, non ha una tastiera, un monitor o un mouse. Ma come qualsiasi computer, ha un processore, un software, input e output.

I sistemi embedded sono computer che controllano vari dispositivi fisici ed elettronici e ora sono quasi ovunque. Dalle cuffie Bluetooth, alle apparecchiature audiovisive domestiche, alle console di gioco, ai condizionatori d'aria, ai robot spazzaneve, alle cuoci-riso e alle lavatrici, alle unità di controllo dei veicoli elettrici, alle apparecchiature di comunicazione, alle apparecchiature di fabbrica, alle apparecchiature mediche, agli uffici e persino alle apparecchiature militari, quasi tutti i dispositivi azionati elettricamente hanno la presenza di sistemi embedded. Il software embedded è il software in esecuzione su di essi e le figure seguenti mostrano alcune scene in cui si possono vedere sistemi embedded.



I sistemi embedded possono essere grandi o piccoli, piccoli come il microcontrollore dell'orologio digitale e grandi come il computer embedded nell'auto autonoma. A differenza

dei computer generici come laptop o smartphone, i sistemi embedded solitamente eseguono un'attività specializzata specifica.

 Fate qualche ricerca: Guardatevi intorno, quali dispositivi potrebbero avere sistemi embedded al loro interno?

Le dimensioni e la complessità dei sistemi embedded variano, ma tutti contengono tre componenti di base:

- **Hardware:** Questi sistemi utilizzano microprocessori e microcontrollori come hardware. I microprocessori sono simili ai microcontrollori perché sono entrambi correlati alle CPU (unità di elaborazione centrale) e le CPU sono combinate con altre parti essenziali del computer (come chip di archiviazione e processori di segnale digitale (DSP)). I microcontrollori integrano tutte queste parti in un singolo chip.
- **Firmware e Software:** La complessità del software di sistema differisce dai microcontrollori di livello industriale e dai sistemi IoT embedded, d'altra parte, di solito eseguono software relativamente basilari, utilizzando pochissima memoria.
- **Real-Time Operating System (RTOS):** Questi sistemi, in particolare quelli più piccoli, spesso non includono questi sistemi operativi. Supervisionando il software durante l'esecuzione del programma e stabilendo gli standard, RTOS determina il funzionamento del sistema.

I sistemi embedded sono spesso vincolati anche dal loro ambiente di distribuzione. Ad esempio, molti sistemi embedded devono funzionare a batteria. Quindi, la loro progettazione deve considerare parametri di efficienza energetica, forse la memoria è limitata o la velocità di clock è estremamente lenta.

La sfida per gli ingegneri che programmano per sistemi embedded è spesso l'implementazione di requisiti funzionali all'interno di questi vincoli limitati di risorse hardware e ambientali. Si devono considerare i vincoli di risorse hardware quando si impara a costruire il proprio modello di progetto TinyML per XIAO in seguito.

#### 4.1.1.2 Edge Computing e Internet of Things (IoT)

Il concetto di “Edge” è relativo al “Centro”. Nei primi tempi, computer come l'ENIAC (Electronic Numerical Integrator and Computer), costruito nel 1945, erano enormi, pesavano quasi 30 tonnellate e occupavano 170 metri quadrati.

Personale che programmava l'ENIAC

A questo stadio, le attività di calcolo erano centralizzate sulla macchina centrale. Questi grandi computer si sono evoluti in “minicomputer”, in genere costituiti da un host centrale e più terminali collegati all'host. Più utenti potevano impartire istruzioni di calcolo tramite i terminali, ma la maggior parte del calcolo avveniva ancora sull'host centrale. Con il passare del tempo, i terminali sono diventati più complessi e hanno assunto sempre più funzioni del computer centrale.

La copertina del manuale del processore per il minicomputer pdp11/70, mostra l'host e i terminali connessi.

Solo con l'avvento dei personal computer il calcolo si è veramente espanso fino al “limite” (edge). Il rapido sviluppo dei personal computer ha portato anche al declino di quelle enormi macchine e la scala del calcolo umano si è rapidamente spostata verso l’“edge”.

Il computer Wang degli anni '70, un tempo leader mondiale in termini di quota di mercato.  
(Personal computer Wang 2200 PCS II. Si trova nel Museo della tecnologia di Belgrado.)

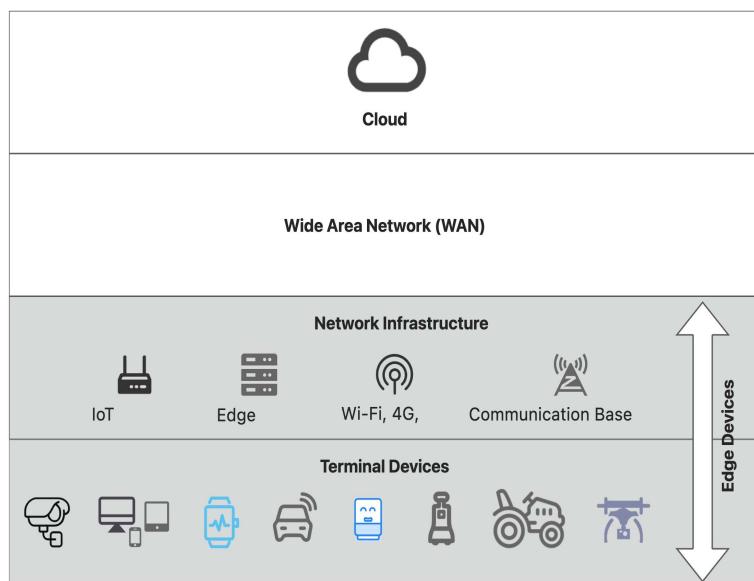
L'emergere e lo sviluppo di Internet hanno portato a una grande concentrazione di server per fornire una varietà di servizi di archiviazione dati e di calcolo, motori di ricerca, streaming video, giochi online, social network, ecc. L'era del cloud computing altamente centralizzato era arrivata e molti provider di servizi Internet possedevano enormi sale di data center.

Il data center di Google situato alla periferia di Pryor, Oklahoma, USA

In teoria, tutti i nostri servizi informatici possono essere completati nel cloud. Ma questi servizi basati sul cloud sono inutili in molte aree senza connessioni Internet o quando Internet non funziona.

I computer che utilizziamo per lavoro e intrattenimento sono solo alcuni dispositivi connessi a questi servizi cloud. Nel 2021, c'erano ben 12,2 miliardi di dispositivi connessi in tutto il mondo e chiamiamo questa rete di dispositivi **IoT (Internet of Things)**. Include tutto ciò a cui si può e non si può pensare, come telefoni cellulari, altoparlanti intelligenti, telecamere di sicurezza connesse, automobili, container, localizzatori per animali domestici, sensori industriali, ecc.

Questi dispositivi sono sistemi embedded contenenti microprocessori che eseguono software scritto da ingegneri di software embedded. Possiamo anche chiamarli dispositivi edge perché si trovano ai margini della rete. Il calcolo eseguito sui dispositivi edge è noto come **Edge Computing**. La seguente illustrazione esprime la relazione tra il cloud e i dispositivi edge.



I dispositivi all'edge della rete possono comunicare con il cloud, l'infrastruttura edge e tra loro. Le applicazioni edge in genere si estendono su più posizioni su questa mappa. Ad esempio, i dati possono essere inviati da un dispositivo IoT dotato di sensori a un server edge locale per l'elaborazione.

#### 4.1.1.3 Intelligenza artificiale (IA)

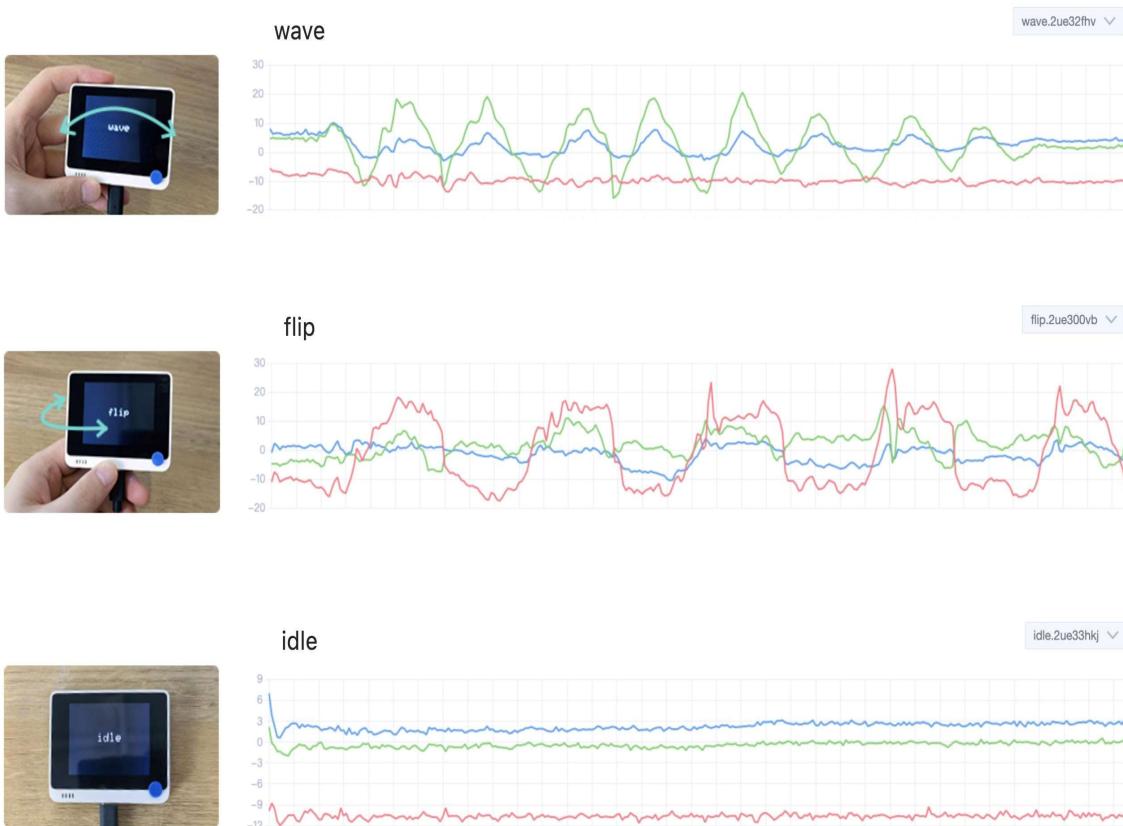
L'intelligenza artificiale (IA) è un concetto vasto e difficile da definire. In un senso vago, si tratta di far sì che le cose possiedano un'intelligenza e capacità di pensiero simili a quelle umane. Ma anche la definizione stessa di intelligenza è controversa e questo è un campo

all'avanguardia con molte incognite, che i lettori sono invitati a esplorare da soli. Per facilitare la comprensione dei concetti seguenti, questo libro fornisce una definizione relativamente ristretta di IA: un sistema artificiale in grado di prendere decisioni sagge basate su input specifici.

#### 4.1.1.4 Machine Learning (ML)

Il “Machine Learning (ML)” [apprendimento automatico] mira principalmente a progettare e analizzare algoritmi che consentano ai computer di “imparare” automaticamente. Gli algoritmi ML sono una classe di algoritmi che analizzano e apprendono automaticamente modelli dai dati e utilizzano tali modelli per prevedere dati sconosciuti.

Prendiamo un tipico esempio di apprendimento automatico: il riconoscimento continuo del movimento. Nella Sezione 2.4, abbiamo conosciuto l’accelerometro triassiale. L’immagine seguente mostra un **Terminale Wio** con un accelerometro triassiale embedded e uno schermo. Possiamo utilizzarlo per registrare i dati dell’accelerometro per diversi movimenti: ondeggiamento (wave), capovolgimento (flip) e inattività (idle).



Guardando i dati generati da questi diversi movimenti, si può trovare un metodo per far sì che la macchina riconosca questi schemi di movimento. Il metodo tradizionale prevede l’analisi e il controllo manuale dei dati, l’identificazione di regole logiche specifiche per diversi movimenti tramite analisi matematica e quindi la scrittura di un programma di riconoscimento per eseguire l’azione desiderata in base a queste regole. Sembra complicato, vero?

Fortunatamente, ora abbiamo metodi di apprendimento automatico. Il “training” [addestramento] e il test di questi dati produrranno un algoritmo e il dispositivo deve solo eseguire questo algoritmo per completare automaticamente il nostro processo di “inferenza” desiderato e fornire risultati. Dallo stato attuale dello sviluppo dell’apprendimento

automatico, questo metodo è eccezionalmente competente nella gestione di scenari di dati complessi. Impareremo di più su questo processo nelle sezioni successive.

#### **4.1.1.5 Edge AI**

Come suggerisce il termine, Edge AI combina dispositivi edge e intelligenza artificiale. Lo sviluppo di Edge AI deriva dalla ricerca di un consumo energetico del sistema inferiore e di una maggiore efficienza. Ad esempio, gli smartwatch o le fitness band più diffusi hanno spesso accelerometri integrati che generano centinaia di letture al secondo, un grande volume di dati, e la lettura continua dei dati è necessaria per riconoscere gli stati di movimento. Se il riconoscimento dei movimenti viene eseguito nel cloud, lo smartwatch dovrebbe consumare molta energia per inviare dati al cloud e solitamente ci sarebbe un ritardo nella ricezione del risultato. Ciò rende l'intero processo di calcolo antieconomico, con elevato consumo energetico e latenza. Questa latenza può anche impedirci di utilizzare efficacemente i dati per il feedback in tempo reale.

Edge AI risolve questo problema riconoscendo i movimenti sullo smartwatch o sulla "band" stessa. Ciò consente di ottenere risultati rapidi senza dover fare affidamento sul cloud. Se i dati necessari devono essere caricati sul cloud, non serve inviare una grande quantità di dati del sensore; vengono invece inviati solo i risultati essenziali del riconoscimento del movimento, riducendo significativamente il volume di comunicazione e consumando meno energia elettrica.

#### **4.1.1.6 Apprendimento Automatico Embedded**

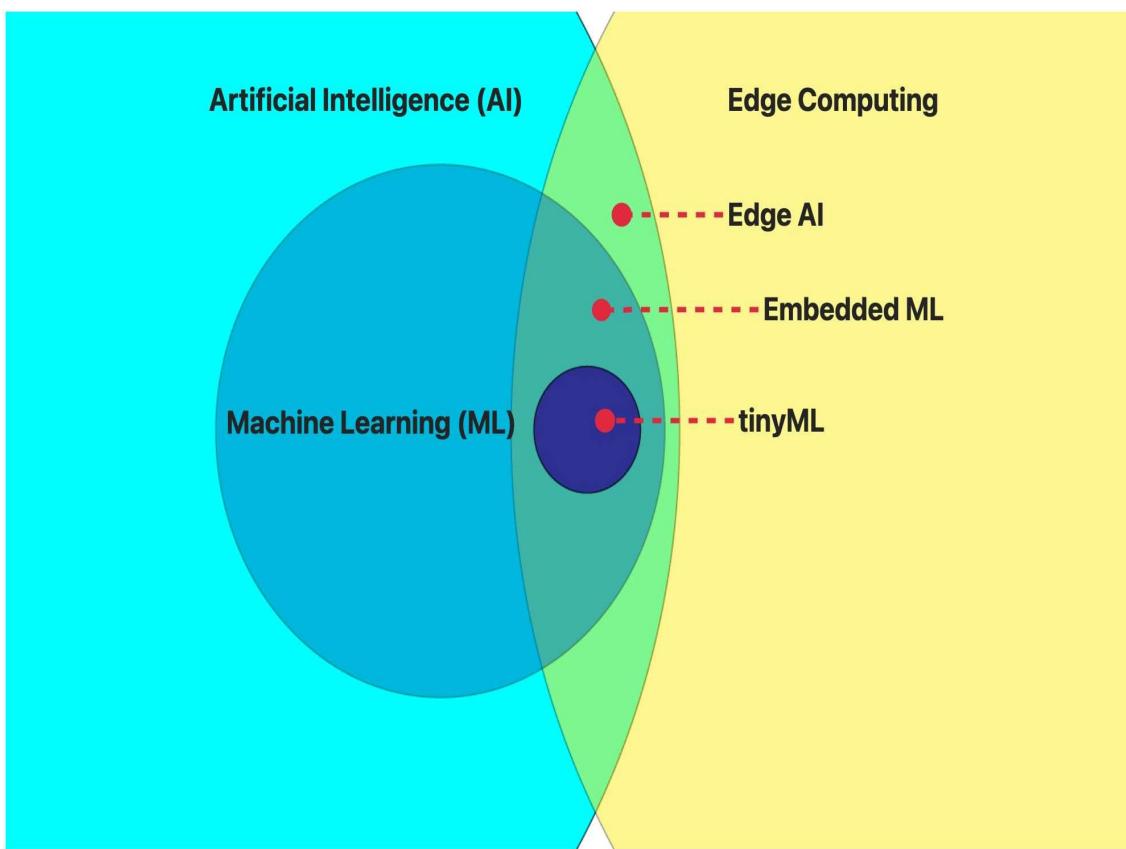
Il "Machine Learning" [apprendimento automatico] embedded è l'arte e la scienza di eseguire modelli di apprendimento automatico su sistemi embedded. Quando parliamo di apprendimento automatico embedded, in genere ci riferiamo all'inferenza dell'apprendimento automatico, ovvero il processo di acquisizione di input e di elaborazione di una previsione (ad esempio, indovinando lo stato del movimento in base ai dati dell'accelerometro). La parte di addestramento viene in genere ancora eseguita su computer tradizionali.

Inoltre, i sistemi embedded hanno solitamente una memoria limitata. Ciò rappresenta una sfida per l'esecuzione di molti modelli di apprendimento automatico, che spesso hanno elevate richieste di memoria di sola lettura (per l'archiviazione del modello) e RAM (per la gestione dei risultati intermedi generati durante l'inferenza). Spesso necessitano anche di una maggiore capacità di elaborazione. Molti modelli di apprendimento automatico sono piuttosto intensivi in termini di elaborazione, il che può anche creare problemi.

#### **4.1.1.7 Tiny Machine Learning (TinyML)**

TinyML implica l'implementazione del processo di inferenza dell'apprendimento automatico sull'hardware embedded più limitato, come Micro Processor Unit (MCU), Digital Signal Processor (DSP) e Field Programmable Gate Array (FPGA).

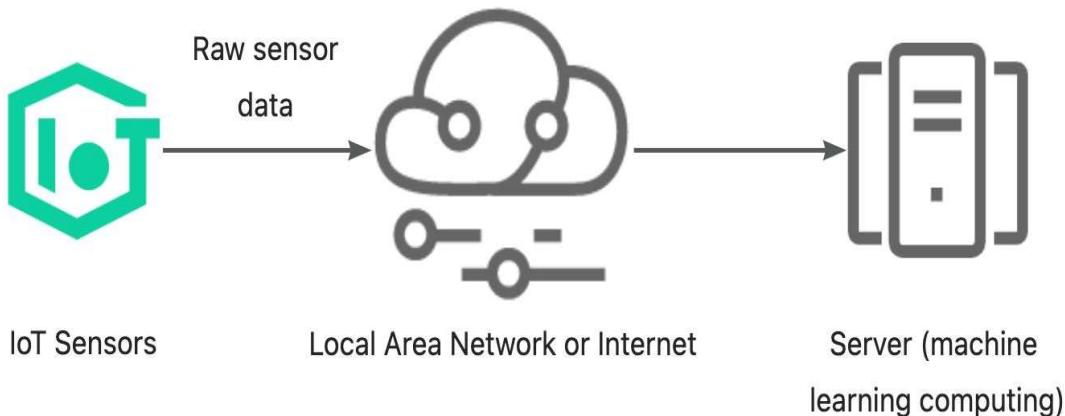
L'immagine sottostante aiuta a comprendere meglio la relazione tra questi termini.



#### 4.1.2 Vantaggi e Funzionamento dell'IA Edge

Per molti anni, l'Internet of Things (IoT) è stato definito “[machine-to-machine](#)” (M2M). Implica la connessione di sensori e vari dispositivi di elaborazione per il controllo dell'automazione dei processi ed è stato ampiamente adottato nei macchinari e nei processi industriali.

L'apprendimento automatico offre la possibilità di compiere ulteriori progressi nell'automazione introducendo modelli in grado di fare previsioni o decisioni senza intervento umano. A causa della complessità di molti algoritmi di apprendimento automatico, l'integrazione tradizionale di IoT e ML comporta l'invio di dati grezzi dei sensori a un server centrale, che esegue i calcoli di inferenza necessari per generare previsioni.



Questa configurazione potrebbe essere accettabile per bassi volumi di dati grezzi e modelli complessi. Tuttavia, sono emersi diversi potenziali problemi:

- La trasmissione di dati estesi dei sensori (come le immagini) può consumare molta larghezza di banda di rete.
- La trasmissione dei dati richiede anche energia.
- I sensori devono connettersi continuamente al server per fornire calcoli di apprendimento automatico quasi in tempo reale.

Date queste sfide e il rapido sviluppo dell'apprendimento automatico, l'Edge AI ha iniziato a emergere. Jeff Bier, fondatore di Edge AI e Vision Alliance, ha delineato cinque fattori che spingono l'intelligenza artificiale al limite nel suo articolo [Cosa Spinge l'IA e la Visione verso l'Edge —BLERP](#), che sta per **Bandwidth, Latency, Economics, Reliability e Privacy**.

- **Bandwidth:** [Larghezza di banda] Se si ha una serra, un'officina o un centro commerciale con centinaia di telecamere, è impossibile inviare queste informazioni al cloud per l'elaborazione: i dati soffocherebbero qualsiasi tipo di connessione Internet disponibile. Si devono semplicemente elaborarli localmente.
- **Latency:** [Latenza] La latenza in questo caso si riferisce al tempo che intercorre tra la ricezione dell'input del sensore da parte del sistema e la risposta. Consideriamo i veicoli autonomi: se un pedone appare all'improvviso sulle strisce pedonali, il computer dell'auto potrebbe avere solo poche centinaia di millisecondi per decidere. Non c'è abbastanza tempo per inviare l'immagine al cloud e attendere una risposta.
- **Economics:** [Economia] Il cloud computing e la comunicazione stanno migliorando e diventando più economici, ma costano ancora soldi, forse molti soldi, soprattutto per quanto riguarda i dati video. L'edge computing riduce la quantità di dati che devono essere inviati al cloud e il carico di lavoro di elaborazione una volta arrivati, riducendo significativamente i costi.
- **Reliability:** [Affidabilità] Si pensi a un sistema di sicurezza domestico con riconoscimento facciale: anche se si verifica un'interruzione di Internet, si vuole comunque che consenta ai familiari di entrare. L'elaborazione locale lo rende possibile e conferisce al sistema una tolleranza agli errori più solida.
- **Privacy:** Il rapido sviluppo di sensori audio e video edge ha causato gravi problemi di privacy e l'invio di queste informazioni al cloud aumenta notevolmente queste preoccupazioni. Più informazioni possono essere elaborate localmente, minore è il potenziale di abuso: ciò che accade sull'edge rimane sull'edge.

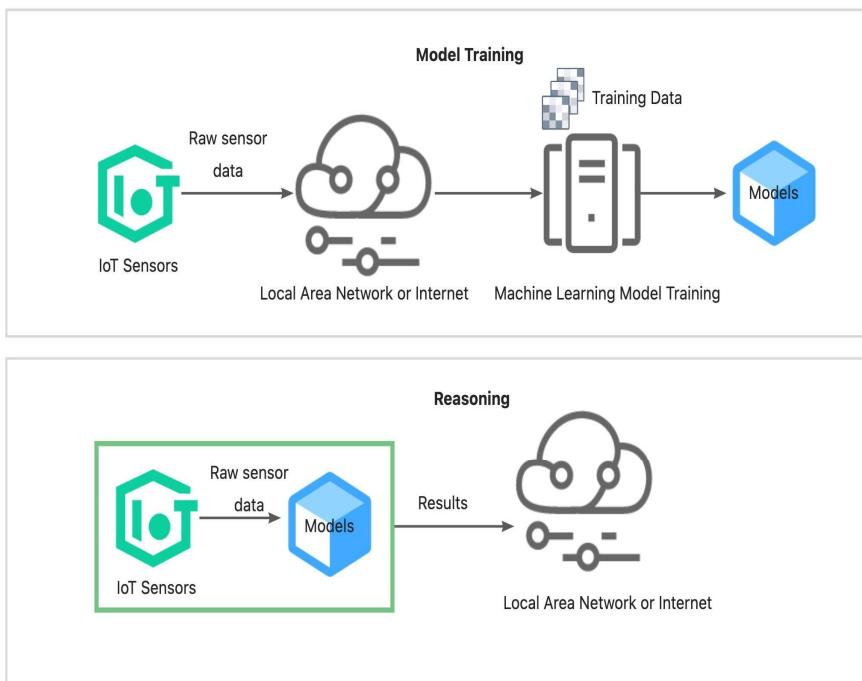
Nella maggior parte dei casi, l'addestramento di un modello di apprendimento automatico comporta un processo in tre fasi: **Modello → Addestramento → Inferenza**, in cui ottenere il modello richiede un calcolo più intensivo rispetto all'esecuzione dell'inferenza.

- **Modello:** Una formula matematica che cerca di generalizzare le informazioni da un dato set di dati.
- **Addestramento:** (o training) Il processo di aggiornamento automatico dei parametri dai dati all'interno di un modello. Questo modello "impara" a trarre conclusioni e generalizzare sui dati.
- **Inferenza:** Fornire dati nuovi e inediti a un modello addestrato per fare previsioni, decisioni o classificazioni.

Pertanto, in circostanze normali, faremmo affidamento su potenti cluster di server per addestrare nuovi modelli, creando set di dati da dati grezzi raccolti in loco (immagini, dati dei sensori, ecc.) e utilizzando questo set di dati per addestrare i nostri modelli di apprendimento automatico.

 Nota: In alcuni casi, possiamo addestrare sul dispositivo. Tuttavia, questo è generalmente irrealizzabile a causa delle limitazioni di memoria ed elaborazione di tali dispositivi edge.

Una volta che abbiamo un modello addestrato, che è solo un modello matematico (sotto forma di libreria software), possiamo distribuirlo ai nostri sensori intelligenti o ad altri dispositivi edge. Possiamo utilizzare questo modello per scrivere firmware o software per raccogliere nuove letture dei sensori grezzi, eseguire inferenze e intraprendere azioni in base a questi risultati di inferenza, come mostrato nella figura seguente. Queste azioni potrebbero essere auto a guida autonoma, bracci robotici in movimento o l'invio di notifiche agli utenti sui guasti del motore. Poiché l'inferenza viene eseguita localmente sul dispositivo edge, il dispositivo non ha bisogno di mantenere una connessione di rete (la connessione facoltativa è mostrata come una linea tratteggiata nel grafico).



### 4.1.3 Applicazioni di Edge AI

L'esecuzione di modelli di apprendimento automatico su dispositivi edge senza rimanere connessi a un computer più potente apre possibilità per vari strumenti automatizzati e sistemi IoT più intelligenti. Ecco alcuni esempi di Edge AI che abilitano l'innovazione in più settori.

Benjamin Cabé ha utilizzato la tecnologia TinyML per creare un naso artificiale che distingue tra vari odori distinti.

#### **Protezione Ambientale**

- Monitoraggio della rete intelligente per rilevare precocemente guasti nelle linee elettriche
- Monitoraggio della fauna selvatica e ricerca comportamentale
- Rilevamento e allerta precoce degli incendi boschivi

#### **Agricoltura**

- Diserbo di precisione, fertilizzazione, applicazione di pesticidi o irrigazione
- Riconoscimento automatico delle esigenze di irrigazione
- Riconoscimento automatico dello stato delle colture e delle condizioni di infestazione da malattie/insetti

#### **Edifici Intelligenti**

- Monitoraggio delle intrusioni e riconoscimento di stati anomali

- Sistemi di condizionamento dell'aria che si adattano in base al numero di persone in una stanza

### **Salute e Sport**

- Dispositivi indossabili che monitorano le condizioni del sonno e dell'esercizio fisico
- Dispositivi medici portatili
- Riconoscimento dei gesti

### **Interazione Uomo-Macchina**

- Rilevamento delle parole di attivazione vocale
- Riconoscimento dei gesti e del movimento del dispositivo per il controllo ausiliario

### **Industria**

- Rilevamento automatico del casco di sicurezza
- Monitoraggio delle condizioni di macchine, apparecchiature e impianti
- Rilevamento dei difetti della linea di produzione
- Rilevamento dello stato di posizione e movimento

La potenza di calcolo solitamente richiesta per eseguire l'inferenza di apprendimento automatico nell'edge è spesso più significativa del semplice polling dei sensori e della trasmissione di dati grezzi. Tuttavia, localmente, ottenere tali calcoli richiede meno potenza rispetto all'invio di dati grezzi a un server remoto.

La tabella seguente fornisce alcuni suggerimenti sul tipo di hardware necessario per eseguire l'inferenza di apprendimento automatico in periferia, a seconda dell'applicazione richiesta.

[Fonte: <https://docs.edgeimpulse.com/docs/concepts/what-is-edge-machine-learning>]

|                 |                | XIAO per questo tipo di attività |                       |                                |                                             |                          |
|-----------------|----------------|----------------------------------|-----------------------|--------------------------------|---------------------------------------------|--------------------------|
|                 |                | MCU di fascia bassa              | MCU di fascia alta    | NPU (Neural Network Processor) | MPU (Microprocessor)                        | GPU (Graphics Processor) |
| <b>Attività</b> |                | Classificazione Sensor Fusion    | Classificazione Audio | Classificazione delle Immagini | Immagini o Suoni Complessi e Video Semplici | Classificazione Video    |
| <b>Memoria</b>  |                | 18KB                             | 50KB                  | 256KB                          | 1MB+                                        | 1GB+                     |
| <b>Sensori</b>  |                | ✓                                | ✓                     | ✓                              | ✓                                           | ✓                        |
| <b>Audio</b>    |                |                                  | ✓                     | ✓                              | ✓                                           | ✓                        |
| <b>Immagini</b> |                |                                  |                       | ✓                              | ✓                                           | ✓                        |
| <b>Video</b>    |                |                                  |                       |                                | ✓                                           | ✓                        |
| <b>XIAO</b>     | nRF52840 Sense | nRF52840 & ESP32S3 Sense         | ESP32S3 Sense         | ESP32S3 Sense                  | —                                           | —                        |

Anche l'hardware embedded si sta evolvendo rapidamente e si prevede che il contenuto di questa tabella cambierà presto.

#### 4.1.4 Introduzione a Edge Impulse Studio

Edge Impulse è stata fondata da Zach Shelby e Jan Jongboom nel 2019. È la piattaforma di sviluppo di apprendimento automatico per dispositivi all'avanguardia. Questa piattaforma consente agli sviluppatori di creare e ottimizzare soluzioni con dati del mondo reale, rendendo il processo di creazione, distribuzione e ridimensionamento di applicazioni ML embedded più accessibile e veloce che mai.

Logo di Edge Impulse.

Si può visitare [il sito web ufficiale di Edge Impulse](#) per maggiori informazioni su questo strumento e consultare la documentazione ufficiale per una spiegazione di base.

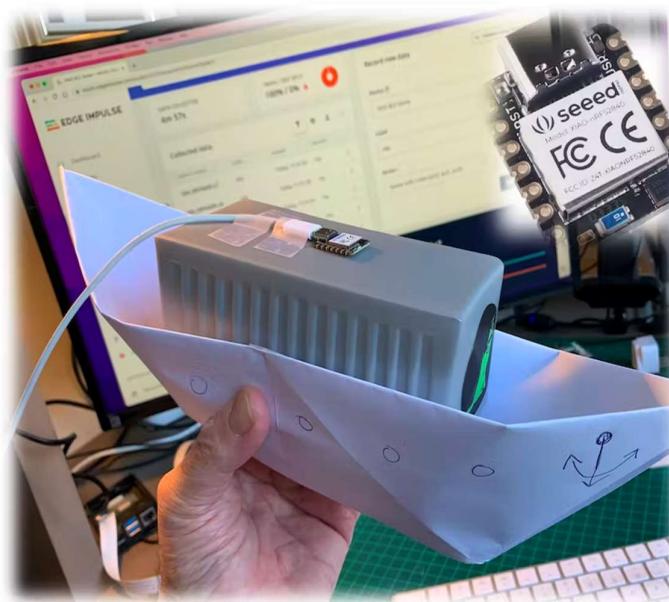
Nelle sezioni seguenti, impareremo a ottenere un riconoscimento continuo del movimento con l'accelerometro a 6 assi a bordo del [XIAO nRF52840 Sense](#) mostrato di seguito e la funzionalità di attivazione tramite parola chiave vocale utilizzando il microfono PDM di bordo.

Seeed Studio XIAO nRF52840 Sense

Le applicazioni di Computer Vision come Image Classification e Object Detection saranno implementate anche utilizzando la fotocamera del XIAO ESP32S3 Sense mostrata di seguito.

Seeed Studio XIAO ESP32S3 Sense

## 4.2 Rilevamento di Anomalie e Classificazione del Movimento



## 4.2.1 Cose utilizzate in questo progetto

### Componenti hardware

- Seeed Studio XIAO nRF52840 Sense × 1

### App software e servizi online



- Arduino IDE



- Edge Impulse Studio

## 4.2.2 Introduzione

Come imparato nella sezione precedente, i microcontrollori (MCU) sono componenti elettronici molto economici, solitamente con solo pochi kilobyte di RAM, progettati per utilizzare piccole quantità di energia. Si trovano in quasi tutti i dispositivi di consumo, medici, automobilistici e industriali. Quest'anno saranno venduti oltre 40 miliardi di microcontrollori e probabilmente ce ne sono centinaia di miliardi in servizio al giorno d'oggi. Tuttavia, questi dispositivi ricevono poca attenzione perché spesso vengono utilizzati solo per sostituire la funzionalità di vecchi sistemi elettromeccanici in auto, lavatrici o telecomandi. Più di recente, con l'era dell'Internet of Things (IoT), una parte significativa di questi MCU sta generando "quintilioni" di dati che, nella maggior parte, non vengono utilizzati a causa dell'elevato costo e della complessità (larghezza di banda e latenza) della trasmissione dei dati.

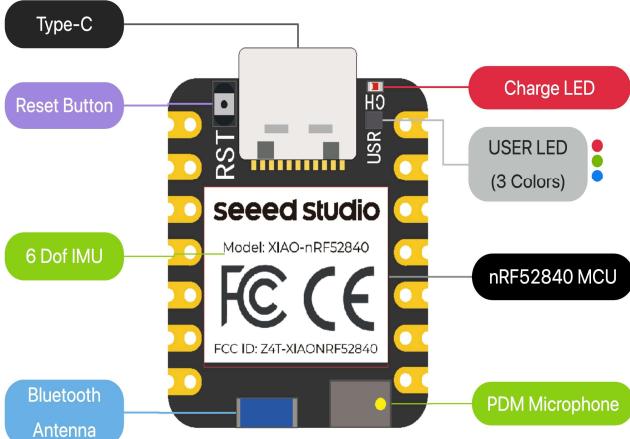
D'altra parte, negli ultimi decenni, abbiamo assistito a un grande sviluppo nei modelli di apprendimento automatico addestrati con grandi quantità di dati in mainframe molto potenti e affamati di energia. E ciò che sta accadendo oggi è che, grazie a questi sviluppi, è ora possibile prendere segnali rumorosi come immagini, audio o accelerometri ed estrarne il significato utilizzando algoritmi di apprendimento automatico come le reti neurali.

E ciò che è più importante è che possiamo eseguire questi algoritmi su microcontrollori e sensori stessi utilizzando pochissima energia, interpretando molti più dati dei sensori che attualmente ignoriamo. Questo è TinyML, una nuova tecnologia che consente l'intelligenza artificiale proprio accanto al mondo fisico.

TinyML può avere molte applicazioni entusiasmanti a beneficio della società in generale.

Questa sezione esplorerà TinyML, in esecuzione su un dispositivo robusto e minuscolo, il [Seed XIAO nRF52840 Sense](#) (chiamato anche XIAO BLE Sense).

### 4.2.3 XIAO nRF52840 Sense



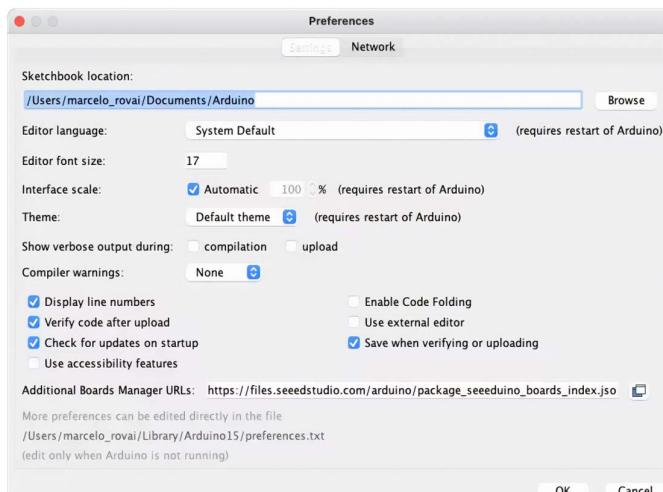
#### Caratteristiche Principali

- Bluetooth 5.0 con antenna integrata
- CPU: Nordic nRF52840, processore ARM® Cortex®-M4 a 32 bit con FPU, 64 MHz
- Consumo Ultra-basso: Il consumo energetico in standby è inferiore a 5 $\mu$ A
- Chip di ricarica della batteria: Supporta la gestione della carica e scarica della batteria al litio
- Flash da 2 MB
- 256 KB RAM
- Microfono PDM
- IMU LSM6DS3TR-C a 6 assi
- Dimensioni ultra ridotte: 20 x 17,5 mm, fattore di forma classico della serie XIAO per dispositivi indossabili
- Interfacce avanzate: 1xUART, 1xI2C, 1xSPI, 1xNFC, 1xSWD, 11xGPIO(PWM), 6xADC
- Componenti monofacciali, design con montaggio superficiale

#### 4.2.3.1 Collegamento di XIAO nRF52840 Sense con Arduino IDE

Il modo più semplice per testare e utilizzare questo dispositivo è utilizzare [Arduino IDE](#). Una volta installato l'IDE sul computer, si va su `File > Preferences` e si compila “Additional Boards Manager URLs” con l’URL seguente:

[https://files.seeedstudio.com/arduino/package\\_seeeduino\\_boards\\_index.json](https://files.seeedstudio.com/arduino/package_seeeduino_boards_index.json)

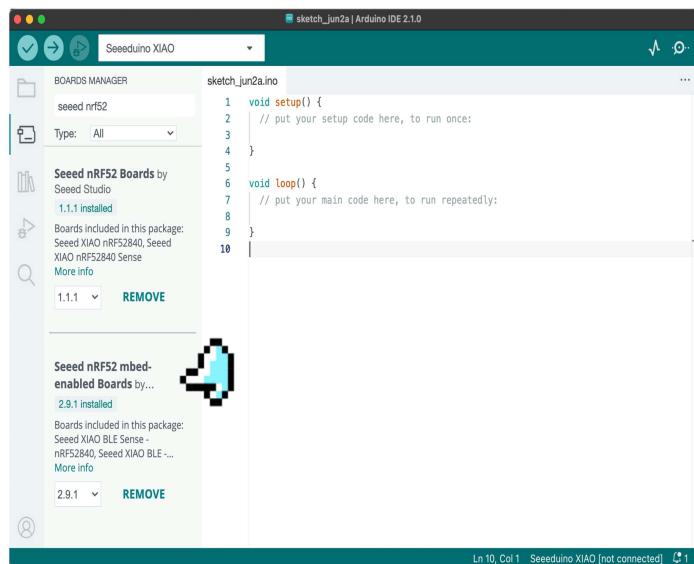


Ora, si va su **Tools>Board>Board Manager** nel menù in alto e si digita la parola chiave filtro **seeed nrf52** nella casella di ricerca.

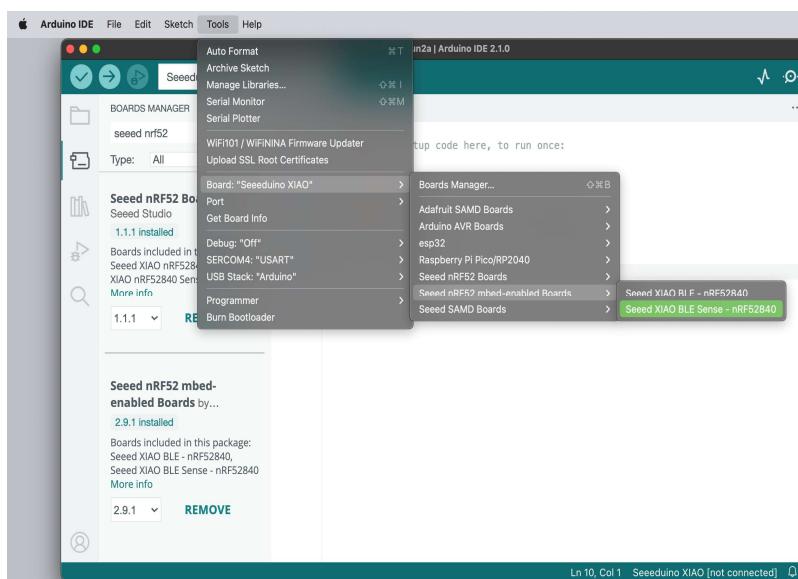
Si vedranno due pacchetti di installazione: **Seeed nRF52 Boards** e **Seeed nRF52 mbed-enabled Boards**, le differenze tra questi due pacchetti sono le seguenti:

- **Seeed nRF52 Boards:** Compatibile con Bluetooth e bassa potenza, adatto per applicazioni Bluetooth e bassa potenza.
- **Seeed nRF52 mbed-enabled Boards:** Compatibile con il supporto TinyML, adatto per realizzare progetti TinyML o Bluetooth, ma non adatto per applicazioni con elevati requisiti di bassa potenza.

Poiché svilupperemo un progetto TinyML, abbiamo scelto l'ultima versione del pacchetto **Seeed nRF52 mbed-enabled Boards**. Installarlo e attendere che venga visualizzata un prompt di installazione corretta nella finestra di output.



Ora si può accedere a questo dispositivo dall'IDE Arduino selezionando la scheda di sviluppo e la porta seriale, come mostrato nella figura sottostante.



La scheda di sviluppo è ora pronta per eseguire il codice su di essa. Cominciamo con Blink, accendendo il LED. Notare che la scheda non ha un LED normale come la maggior parte delle

schede Arduino. Invece, si troverà un LED RGB che può essere attivato con “logica inversa” (si deve applicare LOW per attivare ciascuno dei tre LED separati). Testare il LED RGB con il seguente codice:

---

```

void setup() {
    // initialize serial.
    Serial.begin(115200);
    while (!Serial);
    Serial.println("Serial Started");

    // Pins for the built-in RGB LEDs on the Arduino Nano 33 BLE Sense
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(LED_B, OUTPUT);

    // Note: The RGB LEDs are ON when the pin is LOW and off when HIGH.
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_G, HIGH);
    digitalWrite(LED_B, HIGH);
}

void loop() {
    digitalWrite(LED_R, LOW);
    Serial.println("LED RED ON");
    delay(1000);
    digitalWrite(LED_R, HIGH);
    Serial.println("LED RED OFF");
    delay(1000);

    digitalWrite(LED_G, LOW);
    Serial.println("LED GREEN ON");
    delay(1000);
    digitalWrite(LED_G, HIGH);
    Serial.println("LED GREEN OFF");
    delay(1000);

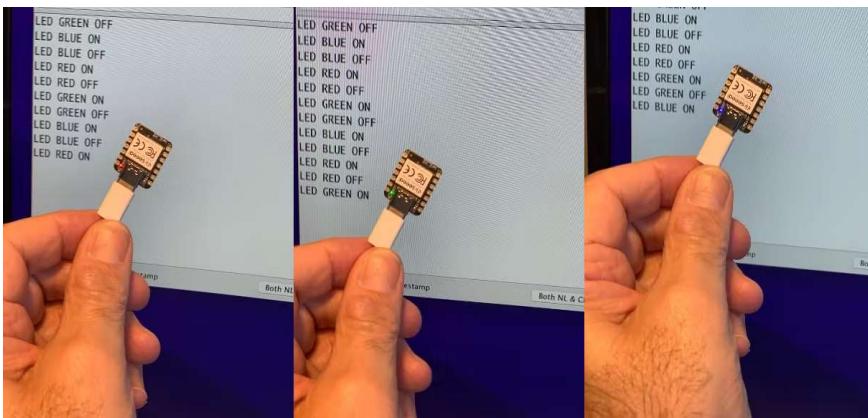
    digitalWrite(LED_B, LOW);
    Serial.println("LED BLUE ON");
    delay(1000);
    digitalWrite(LED_B, HIGH);
    Serial.println("LED BLUE OFF");
    delay(1000);
}

```

---

Prelevare questo codice online  [https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/tree/main/Seeed\\_Xiao\\_Sense\\_bilnk\\_RGB](https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/tree/main/Seeed_Xiao_Sense_bilnk_RGB)

Ecco il risultato:



#### 4.2.3.2 Test del Microfono

XIAO nRF52840 Sense ha un **microfono MEMS** con uscita digitale **PDM**. Eseguire il codice seguente per testarlo:

---

```

#include <PDM.h>

// buffer to read samples into, each sample is 16-bits
short sampleBuffer[256];

```

```

// number of samples read
volatile int samplesRead;

void setup() {
    Serial.begin(9600);
    while (!Serial);

    // configure the data receive callback
    PDM.onReceive(onPDMdata);

    // optionally set the gain, defaults to 20
    // PDM.setGain(30);

    // initialize PDM with:
    // - one channel (mono mode)
    // - a 16 kHz sample rate
    if (!PDM.begin(1, 16000)) {
        Serial.println("Failed to start PDM!");
        while (1);
    }
}

void loop() {
    // wait for samples to be read
    if (samplesRead) {

        // print samples to the serial monitor or plotter
        for (int i = 0; i < samplesRead; i++) {
            Serial.println(sampleBuffer[i]);
            // check if the sound value is higher than 500
            if (sampleBuffer[i]>=500){
                digitalWrite(LED_R,LOW);
                digitalWrite(LED_G,HIGH);
                digitalWrite(LED_B,HIGH);
            }

            // check if the sound value is higher than 250 and lower than 500
            if (sampleBuffer[i]>=250 && sampleBuffer[i] < 500{
                digitalWrite(LED_B,LOW);
                digitalWrite(LED_R,HIGH);
                digitalWrite(LED_G,HIGH);
            }

            //check if the sound value is higher than 0 and lower than 250
            if (sampleBuffer[i]>=0 && sampleBuffer[i] < 250){
                digitalWrite(LED_G,LOW);
                digitalWrite(LED_R,HIGH);
                digitalWrite(LED_B,HIGH);
            }
        }

        // clear the read count
        samplesRead = 0;
    }
}

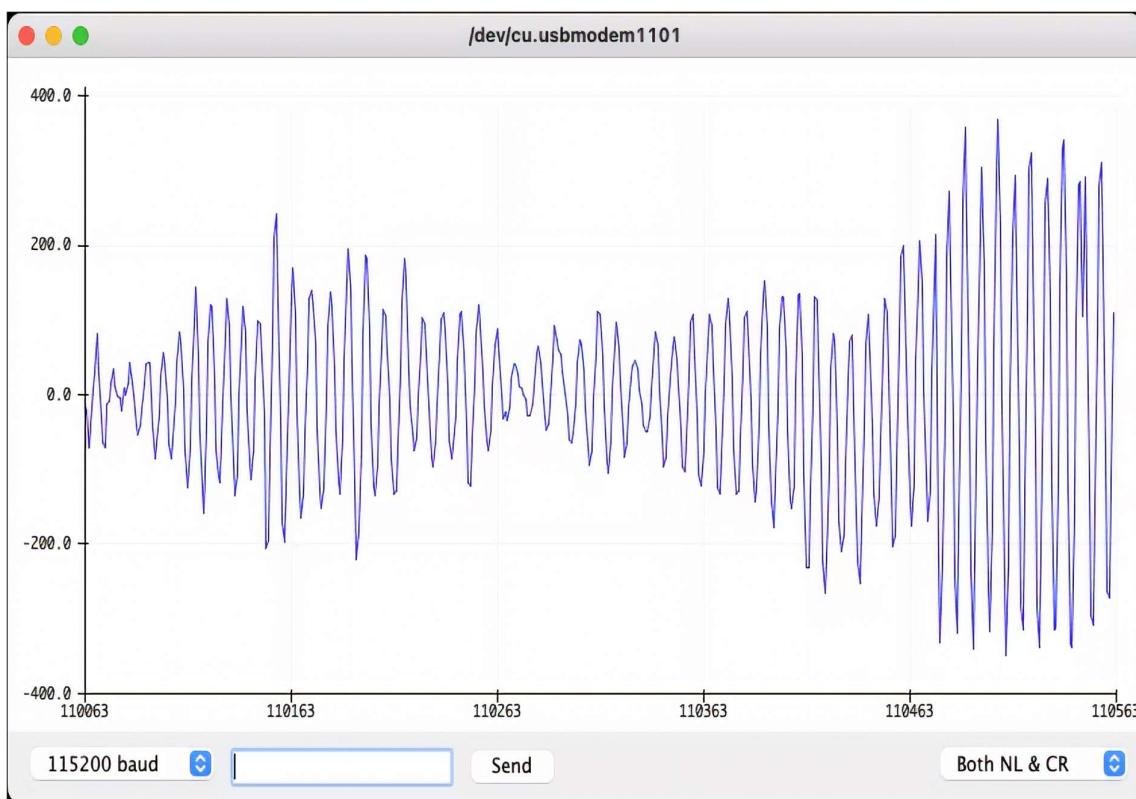
void onPDMdata() {
    // query the number of bytes available
    int bytesAvailable = PDM.available();

    // read into the sample buffer
    PDM.read(sampleBuffer, bytesAvailable);

    // 16-bit, 2 bytes per sample
    samplesRead = bytesAvailable / 2;
}

```

Il codice precedente catturerà continuamente i dati nel suo buffer, visualizzandoli nel Monitor seriale e nel Plotter:



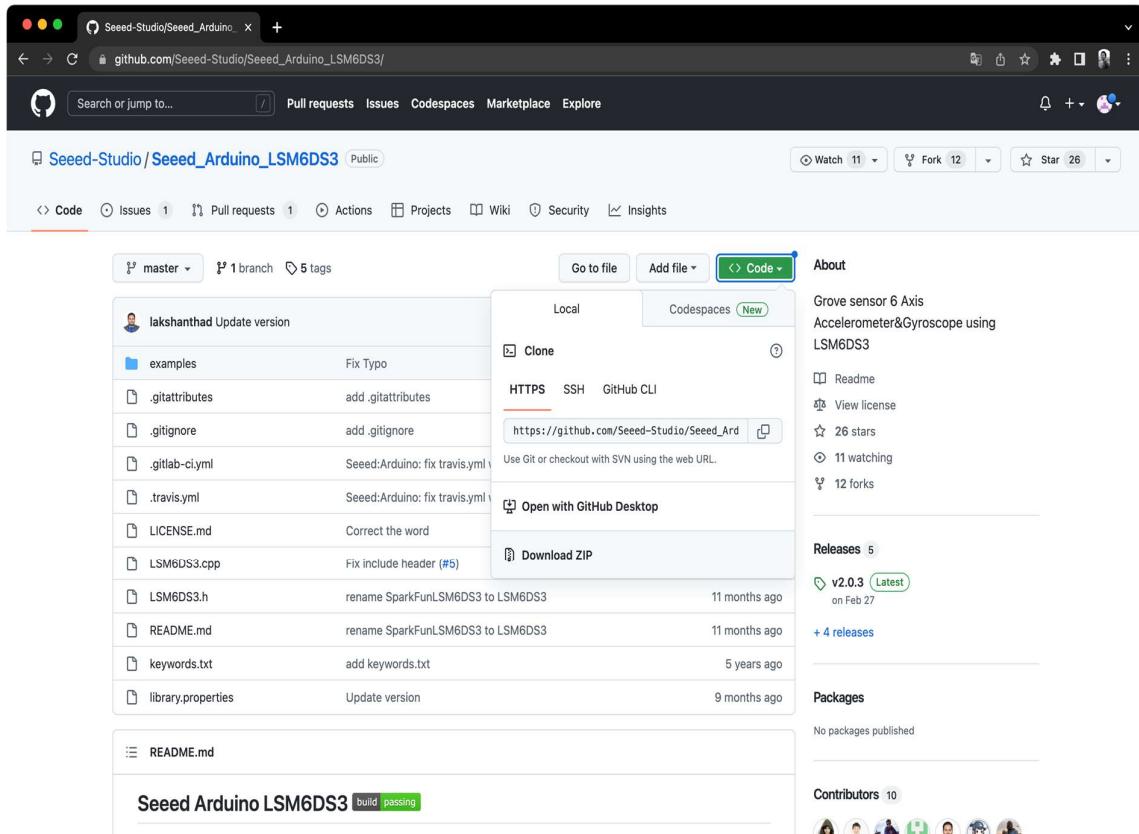
Notare, inoltre, che il LED RGB verrà impostato in base all'intensità del suono.

Il microfono non verrà utilizzato in questo progetto in particolare, ma è bene testarlo se è la prima volta che si usa XIAO nRF52840 Sense.

#### 4.2.3.3 Test dell'IMU

Il nostro piccolo dispositivo ha anche integrato un IMU a 6 assi, il [LSM6DS3TR-C](#), un accelerometro digitale 3D system-in-package e un giroscopio digitale 3D. Per il test, si deve prima installare la sua libreria '[Seeed Arduino LSM6DS3](#)'.

Prima di programmare l'accelerometro con l'IDE Arduino, si deve aggiungere la libreria necessaria per il sensore. Si immette l'indirizzo della libreria [https://github.com/Seeed-Studio/Seeed\\_Arduino\\_LSM6DS3/](https://github.com/Seeed-Studio/Seeed_Arduino_LSM6DS3/) nella barra degli indirizzi del browser, si va alla pagina GitHub, e si clicca su `Code → Download ZIP` per scaricare il pacchetto di risorse `Seeed_Arduino_LSM6DS3-master.zip` nell'area locale, come mostrato di seguito.



Si aggiunge il pacchetto di risorse `Seeed_Arduino_LSM6DS3-master.zip` scaricato nel passaggio precedente in `sketch->Include Library->Add .ZIP Library` della barra dei menù finché non si vede un messaggio che indica che la libreria è stata caricata correttamente.

### **Eseguire il codice di prova basato su tinymlx - Sensor Test dell'Università di Harvard**

Ora, eseguire il seguente codice di prova basato su tinymlx - Sensor Test dell'Università di Harvard.

```
#include "LSM6DS3.h"
#include "Wire.h"

//Create an instance of class LSM6DS3
LSM6DS3 xIMU(I2C_MODE, 0x6A); //I2C device address 0x6A

char c;
int sign = 0;

void setup() {
  Serial.begin(115200);
  while (!Serial);

  // configure the IMU
  if (xIMU.begin() != 0) {
    Serial.println("Device error");
  } else {
    Serial.println("Device OK!");
  }

  Serial.println("Welcome to the IMU test for the built-in IMU on the XIAO BLE Sense\n");
  Serial.println("Available commands:");
  Serial.println("a - display accelerometer readings in g's in x, y, and z directions");
  Serial.println("g - display gyroscope readings in deg/s in x, y, and z directions");
  Serial.println("t - display temperature readings in oC and oF");
}

void loop() {
  // Read incoming commands from serial monitor

  if (Serial.available()) {
    c = Serial.read();
    Serial.println(c);
  }
}
```

```

}

if(c == 'a')sign=1;
else if(c == 'g')sign=2;
else if(c == 't')sign=3;

float x, y, z;
if (sign ==1) { // testing accelerometer
//Accelerometer
x = xIMU.readFloatAccelX();
y = xIMU.readFloatAccelY();
z = xIMU.readFloatAccelZ();
Serial.print("\nAccelerometer:\n");
Serial.print("Ax:");
Serial.print(x);
Serial.print(' ');
Serial.print("Ay:");
Serial.print(y);
Serial.print(' ');
Serial.print("Az:");
Serial.println(z);
}

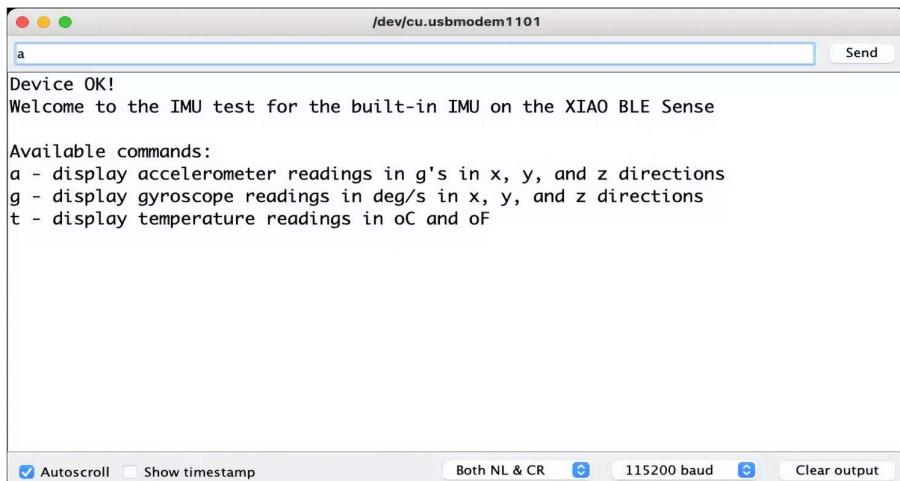
else if (sign ==2) { // testing gyroscope
//Gyroscope
Serial.print("\nGyroscope:\n");
x = xIMU.readFloatGyroX();
y = xIMU.readFloatGyroY();
z = xIMU.readFloatGyroZ();
Serial.print("wx:");
Serial.print(x);
Serial.print(' ');
Serial.print("wy:");
Serial.print(y);
Serial.print(' ');
Serial.print("wz:");
Serial.println(z);
}

else if (sign ==3) { // testing thermometer
//Thermometer
Serial.print("\nThermometer:\n");
Serial.print(" Degrees oC = ");
Serial.println(xIMU.readTempC(), 0);
Serial.print(" Degrees oF = ");
Serial.println(xIMU.readTempF(), 0);
delay(1000);
}
}

```

Prelevare questo codice online  [https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/xiao\\_test\\_IMU/xiao\\_test\\_IMU.ino](https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/xiao_test_IMU/xiao_test_IMU.ino)

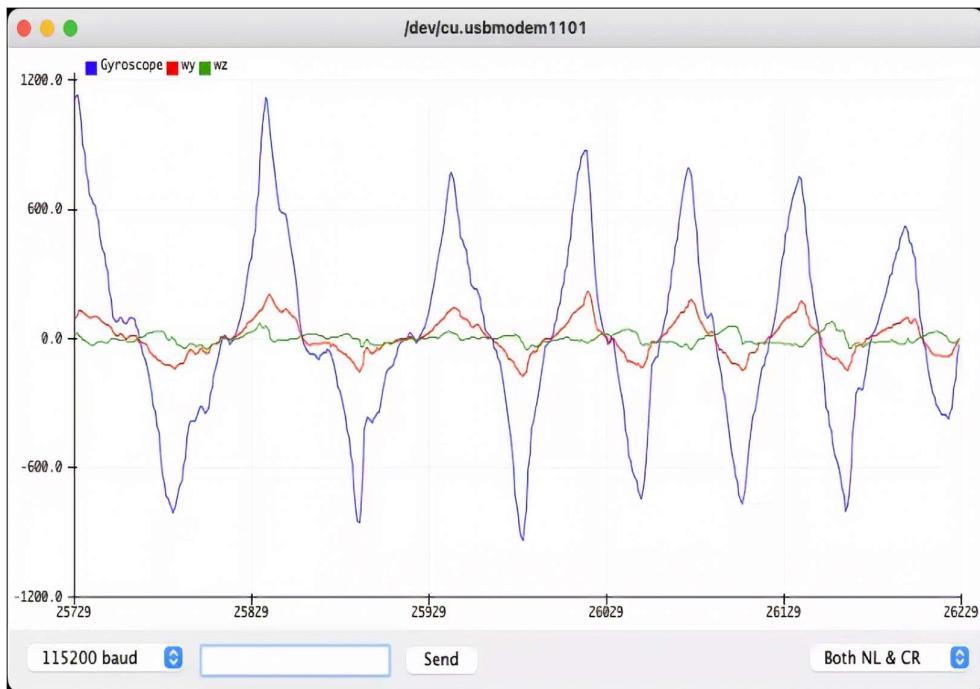
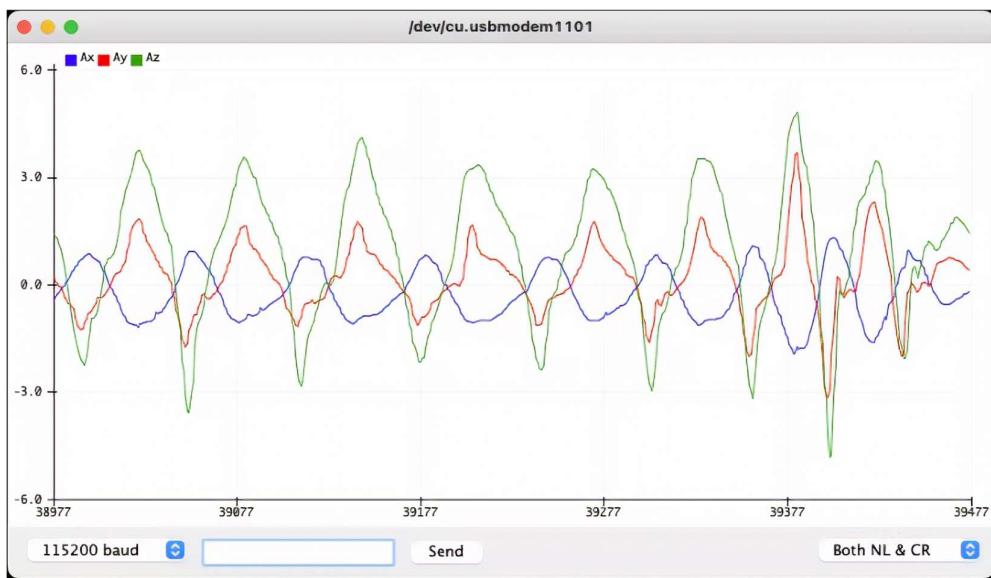
Una volta eseguito lo sketch precedente, aprire il Serial Monitor:



Scegliere una delle tre opzioni per il test:

- **a:** Accelerometro (vedere il risultato su Plotter)
- **g:** Giroscopio (vedere il risultato su Plotter)
- **t:** Temperatura (vedere il risultato su Serial Monitor)

Le seguenti immagini mostrano il risultato:



```

Thermometer:
Degrees oC = 25
Degrees oF = 77

Thermometer:
Degrees oC = 25
Degrees oF = 77

Thermometer:
Degrees oC = 25
Degrees oF = 76

Thermometer:
Degrees oC = 25
Degrees oF = 76

```

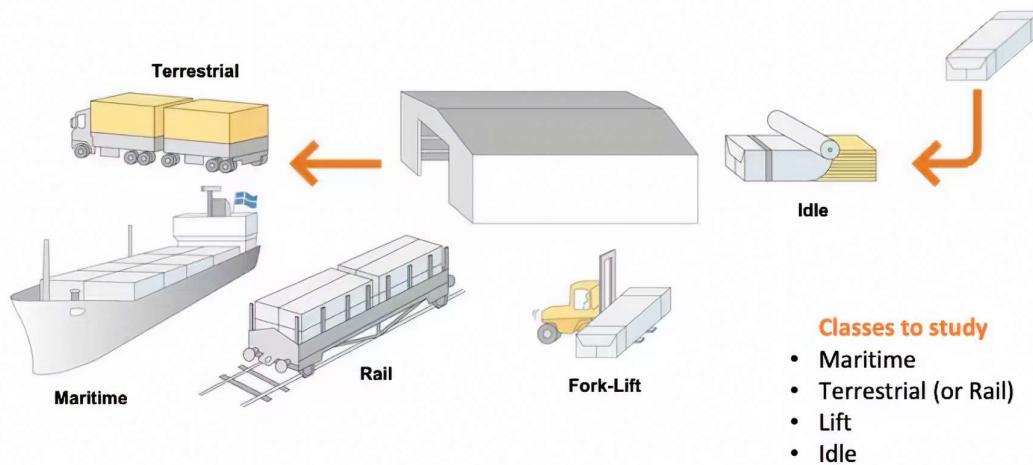
Autoscroll  Show timestamp   Both NL & CR   115200 baud   Clear output

#### 4.2.4 Il Modello di Classificazione del Movimento TinyML

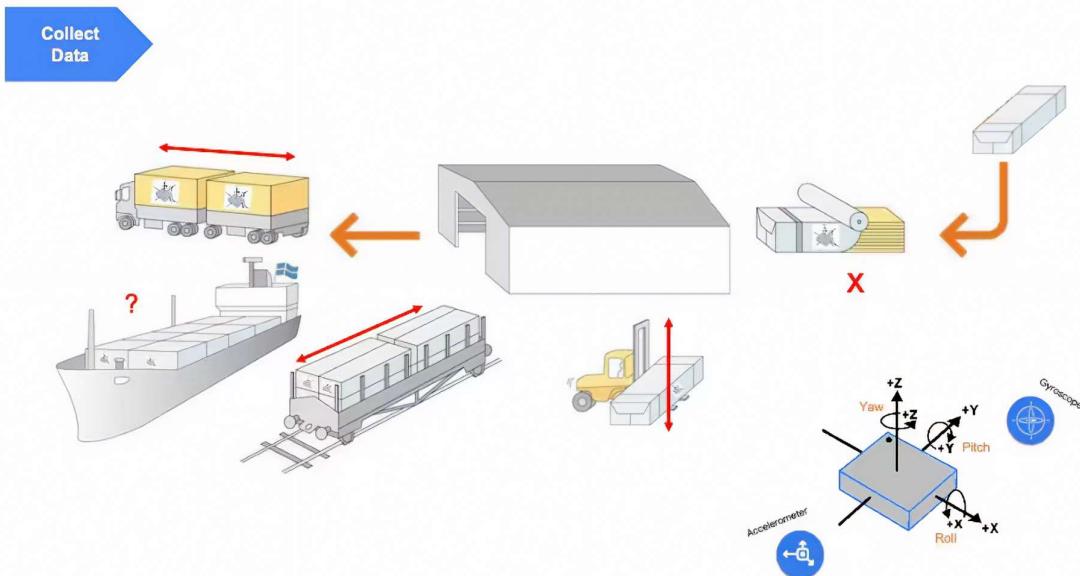
Per il nostro progetto, simuleremo sollecitazioni meccaniche nel trasporto. Il nostro problema sarà classificare quattro classi di movimento:

- **Maritime** (Pallet in navi)
- **Terrestrial** (Pallet in un camion o treno)
- **Lift** (Pallet movimentati da carrello elevatore)
- **Idle** (Pallet in magazzini)

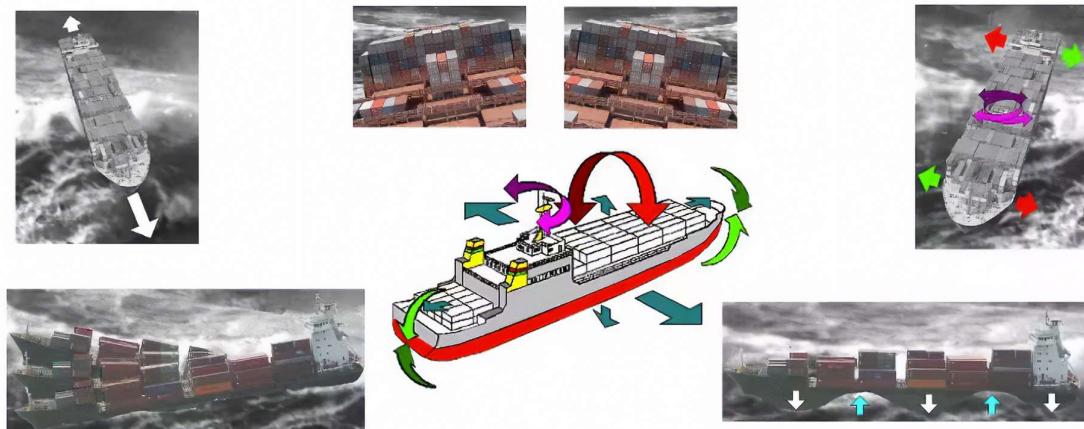
#### Case Study: Mechanical Stresses in Transport



Quindi, per iniziare, dovremmo raccogliere dati. Quindi, gli accelerometri forniranno i dati sul pallet (o sul contenitore).



## Mechanical Stresses in Maritime Transport



Dalle immagini sopra, possiamo vedere che i movimenti principalmente orizzontali dovrebbero essere associati alla “classe Terrestre” [Terrestre], i movimenti verticali alla “classe Lift” [sollevamento], nessuna attività alla “classe Idle” [inattiva] e i movimenti su tutti e tre gli assi alla **classe Marittima**.

### 4.2.4.1 Collegamento di un dispositivo a Edge Impulse Studio

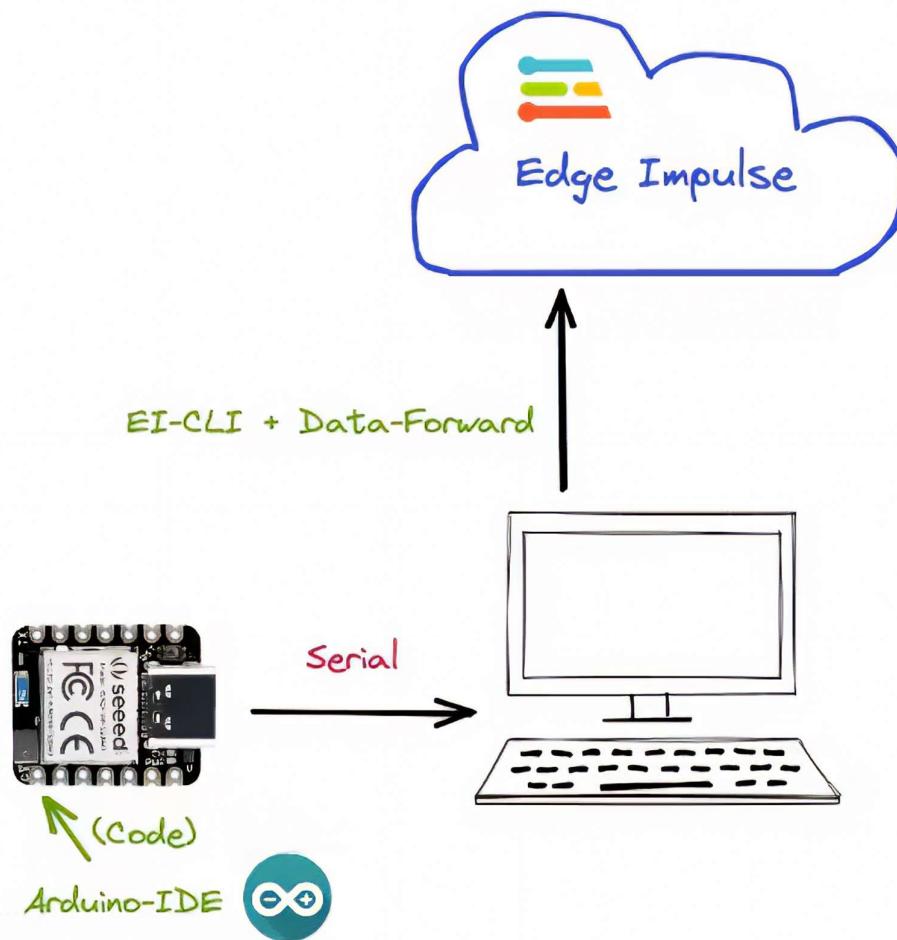
Per la raccolta dati, possiamo avere diverse opzioni. In un caso reale, possiamo avere il nostro dispositivo, ad esempio, collegato direttamente a un contenitore e i dati raccolti su un file (ad esempio .CSV) e memorizzati su una scheda SD (tramite connessione SPI) o un repository offline nel computer. I dati possono anche essere inviati in remoto a un repository nelle vicinanze, come un telefono cellulare, tramite Bluetooth come fatto in questo progetto: [Sensor DataLogger](#). Una volta che il dataset è stato raccolto e archiviato come file .CSV, può essere caricato su Studio utilizzando lo [strumento CSV Wizard](#).

In questo [video](#), si possono imparare modi alternativi per inviare dati a Edge Impulse Studio.

In questo progetto, dovremmo prima connettere il nostro dispositivo a Edge Impulse Studio per la raccolta dati, che verrà utilizzata anche per la pre-elaborazione dei dati, l'addestramento del modello, i test e la distribuzione.

Seguire le istruzioni [qui](#) per installare [Node.js](#) e Edge Impulse CLI sul computer.

Una volta che XIAO nRF52840 Sense non è una scheda di sviluppo completamente supportata da Edge Impulse, dovremmo utilizzare [CLI Data Forwarder](#) per acquisire dati dall'accelerometro e inviarli a Studio, come mostrato in questo diagramma:



Il dispositivo dovrebbe essere collegato alla seriale del computer ed eseguire un codice per catturare i dati IMU (accelerometro) e “stamparli” sulla seriale. Poi, Edge Impulse Studio li “catturerà”. Eseguire il codice seguente:

```
#include "LSM6DS3.h"
#include "Wire.h"

//Create an instance of class LSM6DS3
LSM6DS3 xIMU(I2C_MODE, 0x6A); //I2C device address 0x6A

#define CONVERT_G_TO_MS2    9.80665f
#define FREQUENCY_HZ        50
#define INTERVAL_MS          (1000 / (FREQUENCY_HZ + 1))

static unsigned long last_interval_ms = 0;

void setup() {
    Serial.begin(115200);
    while (!Serial);

    // configure the IMU
    if (xIMU.begin() != 0) {
        Serial.println("Device error");
    }
}
```

```

} else {
    Serial.println("Device OK!");
}

Serial.println("Data Forwarder - Built-in IMU (Accelerometer) on the XIAO BLE Sense\n");
}

void loop() {
    float x, y, z;

    if (millis() > last_interval_ms + INTERVAL_MS) {
        last_interval_ms = millis();

        x = xIMU.readFloatAccelX();
        y = xIMU.readFloatAccelY();
        z = xIMU.readFloatAccelZ();

        Serial.print(x * CONVERT_G_TO_MS2);
        Serial.print('\t');
        Serial.print(y * CONVERT_G_TO_MS2);
        Serial.print('\t');
        Serial.println(z * CONVERT_G_TO_MS2);
    }
}

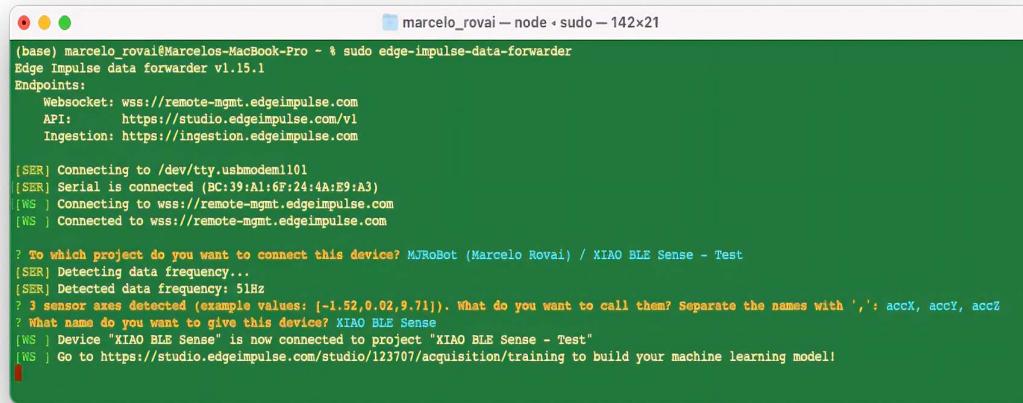
```

Prelevare questo codice online  [https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/XIAO\\_BLE\\_Sense\\_Accelerometer\\_Data\\_Forewarder/XIAO\\_BLE\\_Sense\\_Accelerometer\\_Data\\_Forewarder.ino](https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/XIAO_BLE_Sense_Accelerometer_Data_Forewarder/XIAO_BLE_Sense_Accelerometer_Data_Forewarder.ino)

Si va alla pagina Edge Impulse e si crea un progetto. Quindi, avviare CLI Data Forwarder sul terminale, immettendo (se è la prima volta) il seguente comando:

```
$ edge-impulse-data-forwarder --clean
```

Quindi, immettere le credenziali EI e scegliere i nomi di progetto, variabile e dispositivo:



```

(base) marcelo_rovai@Marcelos-MacBook-Pro ~ $ sudo edge-impulse-data-forwarder
Edge Impulse data forwarder v1.15.1
Endpoints:
  WebSocket: ws://remote-mgmt.edgeimpulse.com
  API:      https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

[SER] Connecting to /dev/tty.usbmodem1101
[SER] Serial is connected (BC:39:A1:6F:24:4A:E9:A3)
[WS ] Connecting to wss://remote-mgmt.edgeimpulse.com
[WS ] Connected to wss://remote-mgmt.edgeimpulse.com

? To which project do you want to connect this device? MJRBot (Marcelo Rovai) / XIAO BLE Sense - Test
[SER] Detecting data frequency...
[SER] Detected data frequency: 51Hz
? 3 sensor axes detected (example values: [-1.52,0.02,0.71]). What do you want to call them? Separate the names with ',' : accX, accY, accZ
? What name do you want to give this device? XIAO BLE Sense
[WS ] Device "XIAO BLE Sense" is now connected to project "XIAO BLE Sense - Test"
[WS ] Go to https://studio.edgeimpulse.com/studio/123707/acquisition/training to build your machine learning model!

```

Studio può leggere la frequenza campionata come 51 Hz anziché i 50 Hz definiti in precedenza nel codice. Va bene.

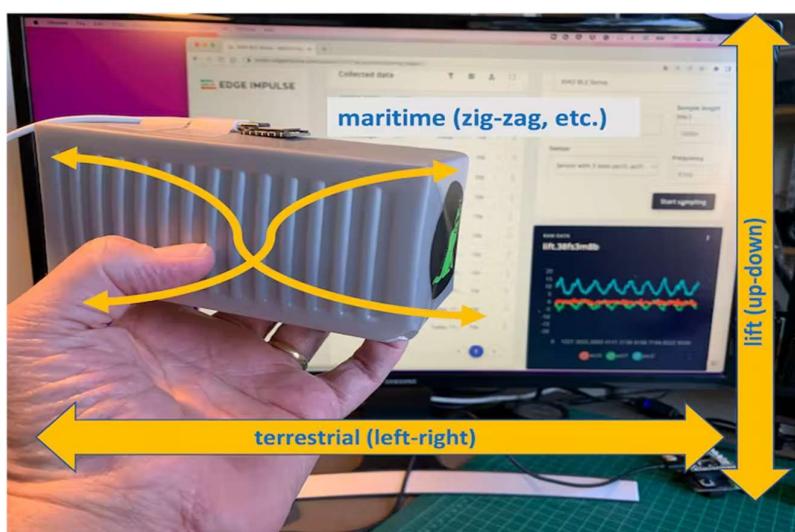
Andare alla sezione `Devices` sul progetto EI e verificare se il dispositivo è connesso (il punto dovrebbe essere verde):

The screenshot shows the Edge Impulse studio interface. On the left is a sidebar with various options like Dashboard, Devices, Data sources, etc. The main area is titled 'Your devices' and lists connected devices. One device, 'XIAO BLE Sease', is highlighted with an orange border. The device card displays its name, ID (BB:3A:90:BB:AC:3E:8B:83), type (DATA\_FORWARDER), sensors (Sensor with 3 axes (accX...)), and last seen (Dec 02 2022, 16:19:56). A cursor arrow is positioned over the right edge of the device card.

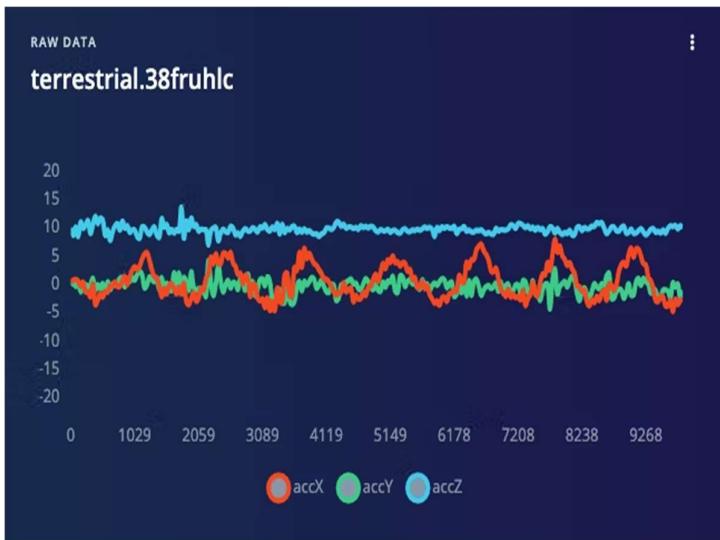
#### 4.2.4.2 Raccolta Dati

Come discusso in precedenza, dovremmo acquisire dati da tutte e quattro le classi di trasporto:

- **lift** (su-giù)
- **terrestrial** (sinistra-destra)
- **maritime** (zig-zag, ecc.)
- **idle**



Di seguito è riportato un campione (10 secondi di dati grezzi):



È possibile acquisire, ad esempio, circa 2 minuti (dodici campioni di 10 secondi) per ciascuna delle quattro classi (un totale di 8 minuti di dati). Utilizzando il menù **tre puntini** dopo ciascuno dei campioni, selezionarne 2, riservandoli per il set di Test. In alternativa, si può utilizzare lo strumento automatico **Train/Test Split tool** nella scheda **Danger zone** della scheda **Dashboard**.

Una volta acquisito il dataset, lo si può esplorare più in dettaglio utilizzando **Data Explorer**, uno strumento visivo per trovare valori anomali o dati etichettati in modo errato (aiutando a correggerli). Data Explorer tenta prima di estrarre feature significative dai dati (applicando l'elaborazione del segnale e gli “embedding” della rete neurale) e poi utilizza un algoritmo di riduzione della dimensionalità come **PCA** o **t-SNE** per mappare queste feature in uno spazio 2D. Questo fornisce una panoramica immediata del dataset completo.

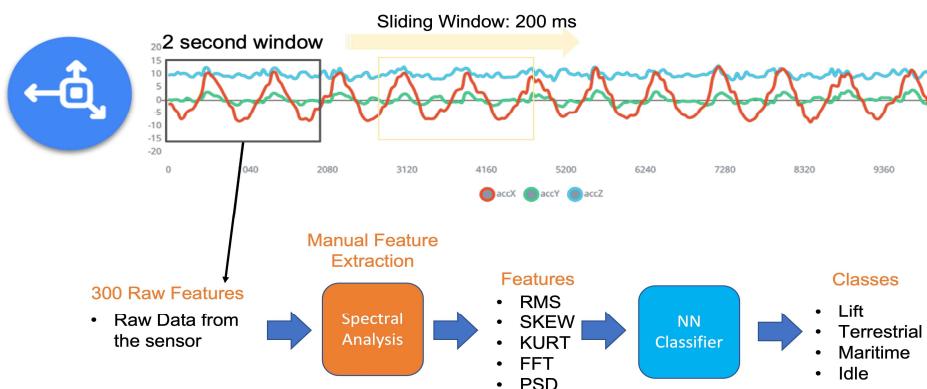
#### 4.2.4.3 Pre-elaborazione dei Dati

La pre-elaborazione dei dati consiste nell'estrapolare le feature dal set di dati acquisito con l'accelerometro, il che implica l'elaborazione e l'analisi dei dati grezzi. Gli accelerometri misurano l'accelerazione di un oggetto lungo uno o più assi (in genere tre, indicati come X, Y e

Z). Queste misure possono essere utilizzate per comprendere vari aspetti del movimento dell'oggetto, come pattern di movimento e vibrazioni.

I dati grezzi dell'accelerometro possono essere rumorosi e contenere errori o informazioni irrilevanti. Le fasi di pre-elaborazione, come il filtraggio e la normalizzazione, possono pulire e standardizzare i dati, rendendoli più adatti all'estrazione di feature. Nel nostro caso, dovremmo dividere i dati in segmenti più piccoli o **windows** [finestre]. Ciò può aiutare a concentrarsi su eventi o attività specifici all'interno del dataset, rendendo l'estrazione di feature più gestibile e significativa. La scelta della **window size** e della sovrapposizione (**window increase**) dipende dall'applicazione e dalla frequenza degli eventi di interesse. Come regola generale, dovremmo provare a catturare un paio di "cicli di dati".

Con un "sampling rate" (SR) [frequenza di campionamento] di 50 Hz e una dimensione della finestra di 2 secondi, otterremo 100 campioni per asse, o 300 in totale (3 assi x 2 secondi x 50 campioni). Faremo scorrere questa finestra ogni 200 ms, creando un set di dati più grande in cui ogni istanza ha 300 feature grezze.



Una volta che i dati sono stati pre-elaborati e segmentati, si possono estrarre feature che descrivono le caratteristiche del movimento. Alcune caratteristiche tipiche estratte dai dati dell'accelerometro includono: - Le caratteristiche del **dominio del tempo** descrivono le proprietà statistiche dei dati all'interno di ciascun segmento, come media, mediana, deviazione standard, asimmetria, curtosi e frequenza di attraversamento dello zero. - Le caratteristiche del **dominio della frequenza** si ottengono trasformando i dati nel dominio della frequenza utilizzando tecniche come la trasformata di Fourier veloce (FFT). Alcune feature tipiche del dominio della frequenza includono lo spettro di potenza, l'energia spettrale, le frequenze dominanti (ampiezza e frequenza) e l'entropia spettrale. - Le caratteristiche del dominio **tempo-frequenza** combinano le informazioni del dominio del tempo e della frequenza, come la trasformata di Fourier a breve termine (STFT) o la trasformata wavelet discreta (DWT). Possono fornire una comprensione più dettagliata di come il contenuto di frequenza del segnale cambia nel tempo.

In molti casi, il numero di feature estrapolate può essere elevato, il che può portare a un overfitting o a una maggiore complessità computazionale. Le tecniche di selezione delle feature, come le informazioni reciproche, i metodi basati sulla correlazione o l'analisi delle componenti principali (PCA), possono aiutare a identificare le feature più rilevanti per una determinata applicazione e ridurre la dimensionalità del dataset. Studio può aiutare con tali calcoli di importanza delle feature.

### Feature Spettrali di El Studio

La pre-elaborazione dei dati è un'area impegnativa per l'apprendimento automatico embedded. Tuttavia, Edge Impulse aiuta a superare questo problema con la sua fase di pre-elaborazione dell'elaborazione del segnale digitale (DSP) e, più specificamente, gli [Spectral Features Block](#).

In Studio, il dataset grezzo raccolto sarà l'input di un blocco di Spectral Analysis, che è eccezionale per analizzare il movimento ripetitivo, come i dati degli accelerometri. Questo blocco eseguirà un DSP (Digital Signal Processing), estraendo feature come la [FFT](#) o le [Wavelet](#).

Per il nostro progetto, una volta che il segnale temporale è continuo, dovremmo usare FFT con, ad esempio, una lunghezza di [32].

Le feature **Time Domain Statistical** per asse/canale sono:

- [RMS](#): 1 feature
- [Skewness](#) [Asimmetria]: 1 feature
- [Curto](#): 1 feature

Le feature **Frequency Domain Spectral features** per asse/canale sono:

- [Potenza Spettrale](#): 16 feature (lunghezza FFT/2)
- Asimmetria: 1 feature
- Curto: 1 feature

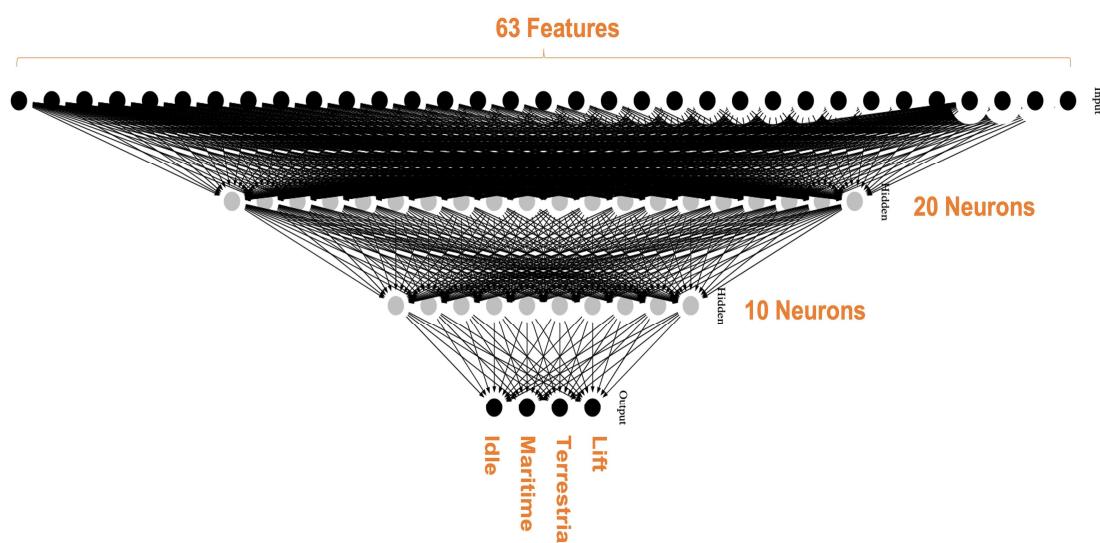
Quindi, per una lunghezza FFT di 32 punti, l'output risultante dello "Spectral Analysis Block" sarà di 21 feature per asse (un totale di 63 feature).

Si può scoprire di più su come viene calcolata ogni feature scaricando il notebook [Edge Impulse - Spectral Features Block Analysis TinyML under the hood: Spectral Analysis](#) o aprendolo direttamente su Google CoLab.

Queste 63 caratteristiche costituiranno il tensore di input di un classificatore di reti neurali.

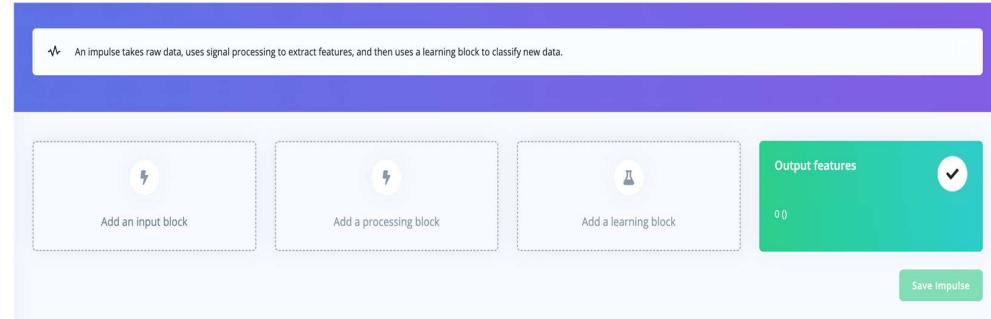
#### 4.2.4.4 Progettazione del Modello

Il nostro classificatore sarà una Dense Neural Network (DNN) che avrà 63 neuroni sul suo layer di input, due layer nascosti con 20 e 10 neuroni e un layer di output con quattro neuroni (uno per ogni classe), come mostrato qui:



#### 4.2.4.5 Impulse Design

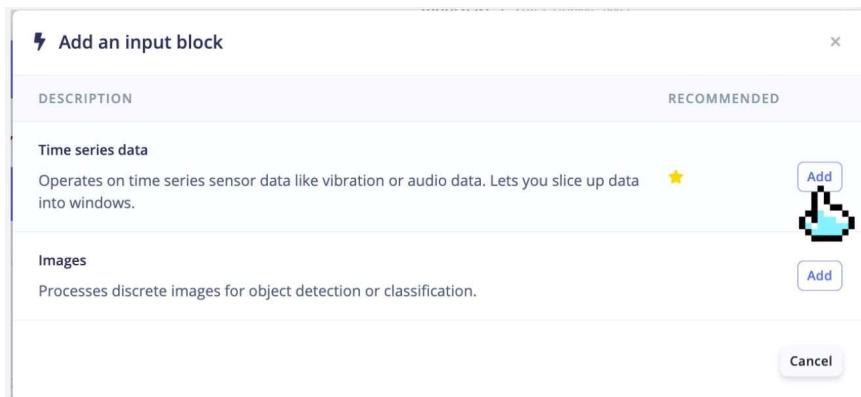
Un Impulse completo comprende tre blocchi di costruzione principali: il blocco di input, che ottiene i dati grezzi, il blocco di elaborazione, che estrae le feature, e il blocco di apprendimento, che classifica i dati. L'immagine seguente mostra l'interfaccia quando i tre blocchi di costruzione devono ancora essere aggiunti e la nostra pipeline di apprendimento automatico verrà implementata aggiungendo questi tre blocchi.



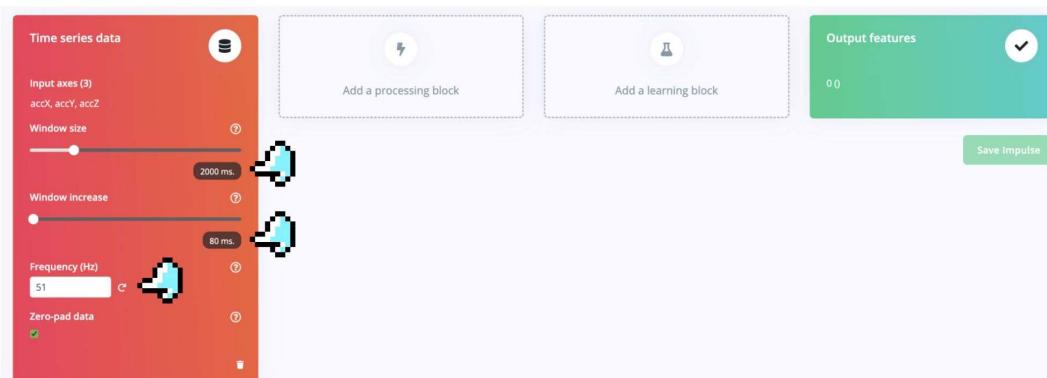
Impulse ottiene i dati grezzi tramite il blocco di input, utilizza il blocco di elaborazione per estrarre le caratteristiche e poi utilizza il blocco di apprendimento per classificare nuovi dati. Nel nostro riconoscimento di azioni continue, i blocchi aggiunti includono:

##### 1. Aggiunta del blocco di input: Dati di Serie Temporale

Cliccare sul pulsante “Add an Input Block” e selezionare **Time Series Data** nella finestra pop-up come mostrato di seguito per abbinare il tipo di dati del sensore che abbiamo raccolto.



Come mostrato nella figura sottostante, impostare **Window Size** su 2000 ms (2 secondi), **Window Increase** su 80 millisecondi e **Frequency** su 51 Hz in base ai calcoli che abbiamo effettuato nella sezione di pre-elaborazione dei dati sul blocco Time Series Data visualizzato.



## 2. Aggiunta del blocco di elaborazione: Analisi Spettrale

Cliccare sul pulsante “Add a Processing Block” e selezionare **Spectral Analysis** nella finestra pop-up come mostrato di seguito per abbinare il nostro tipo di attività di analisi del movimento.

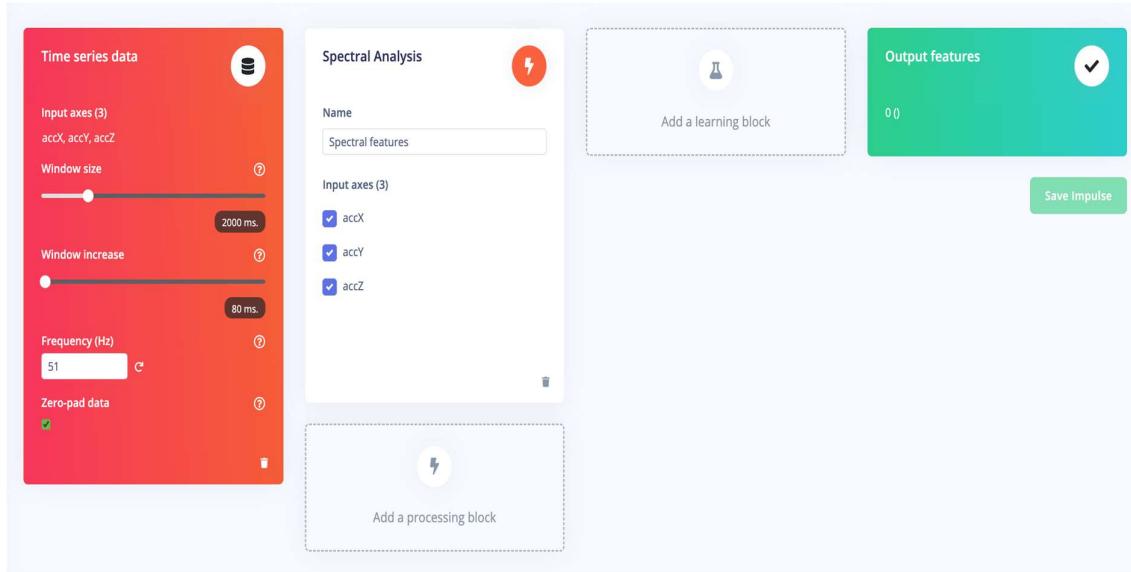
The screenshot shows a dialog box titled "Add a processing block". At the top, there is a note: "Did you know? You can bring your own DSP code." Below this is a table with three columns: DESCRIPTION, AUTHOR, and RECOMMENDED. The first row contains the following information:

| DESCRIPTION                                                                                                                                                                              | AUTHOR         | RECOMMENDED |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------------|
| <b>Spectral Analysis</b><br>Great for analyzing repetitive motion, such as data from accelerometers.<br>Extracts the frequency and power characteristics of a signal over time.          | Edge Impulse ★ |             |
| <b>IMU (Syntiant)</b><br>Syntiant only. Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time. | Syntiant ★     |             |
| <b>Flatten</b><br>Flatten an axis into a single value, useful for slow-moving averages like temperature data, in combination with other blocks.                                          | Edge Impulse   |             |
| <b>Spectrogram</b><br>Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies.                                                   | Edge Impulse   |             |
| <b>Raw Data</b><br>Use data without pre-processing. Useful if you want to use deep learning to learn features.                                                                           | Edge Impulse   |             |

At the bottom of the dialog box, there is a message: "Some processing blocks have been hidden based on the data in your project. Show all blocks anyway".

At the very bottom left is a link "Add custom block" and at the bottom right is a "Cancel" button.

L'effetto dopo l'aggiunta del blocco di elaborazione è mostrato nella figura sottostante.



### 3. Aggiunta del blocco di apprendimento: Classificazione

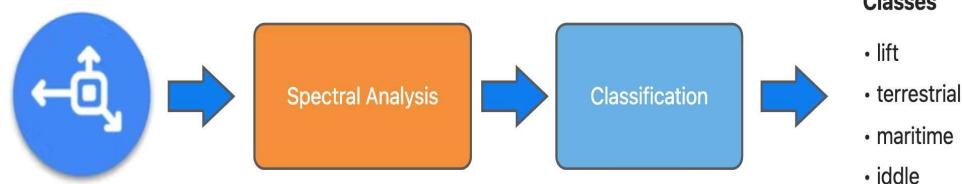
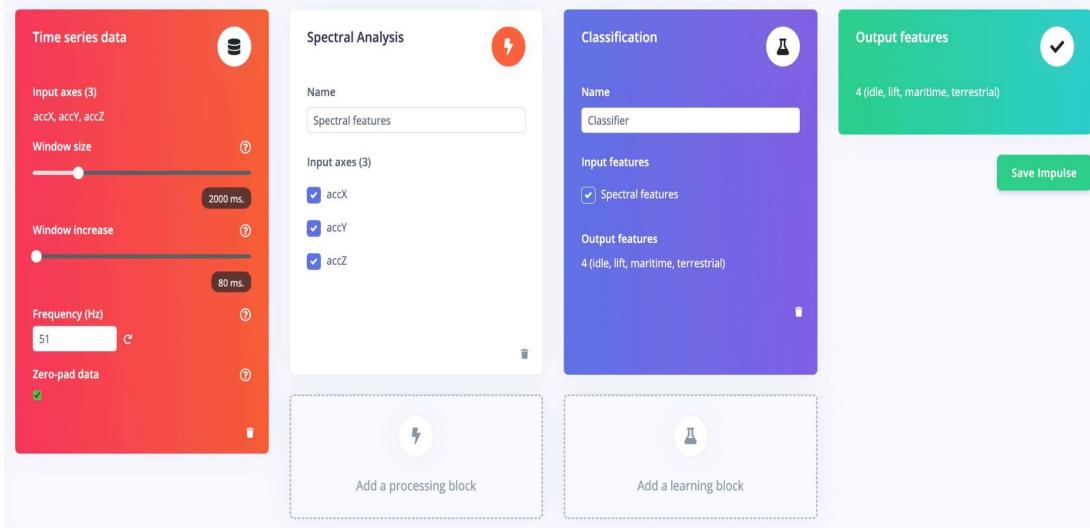
Cliccare sul pulsante “Add Learning Block” e selezionare **Classification** nella finestra pop-up come mostrato di seguito per abbinare il nostro tipo di attività di analisi del movimento.

**💡 Add a learning block**

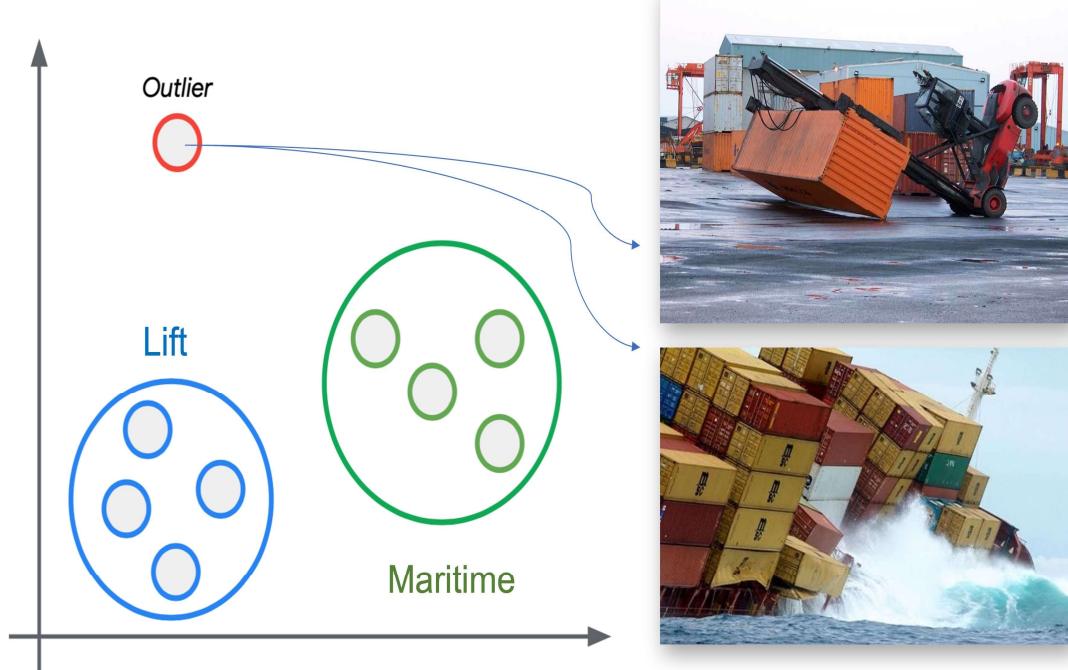
Did you know? You can bring your own model in PyTorch, Keras or scikit-learn.

| DESCRIPTION                                                                                                                                                                                                                      | AUTHOR       | RECOMMENDED |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------|
| <b>Classification</b><br>Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.                                                                                       | Edge Impulse |             |
| <b>Anomaly Detection (K-means)</b><br>Find outliers in new data. Good for recognizing unknown states, and to complement classifiers. Works best with low dimensionality features like the output of the spectral features block. | Edge Impulse |             |
| <b>Regression</b><br>Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.                                                                                                 | Edge Impulse |             |
| Some learning blocks have been hidden based on the data in your project. <a href="#">Show all blocks anyway</a>                                                                                                                  |              |             |

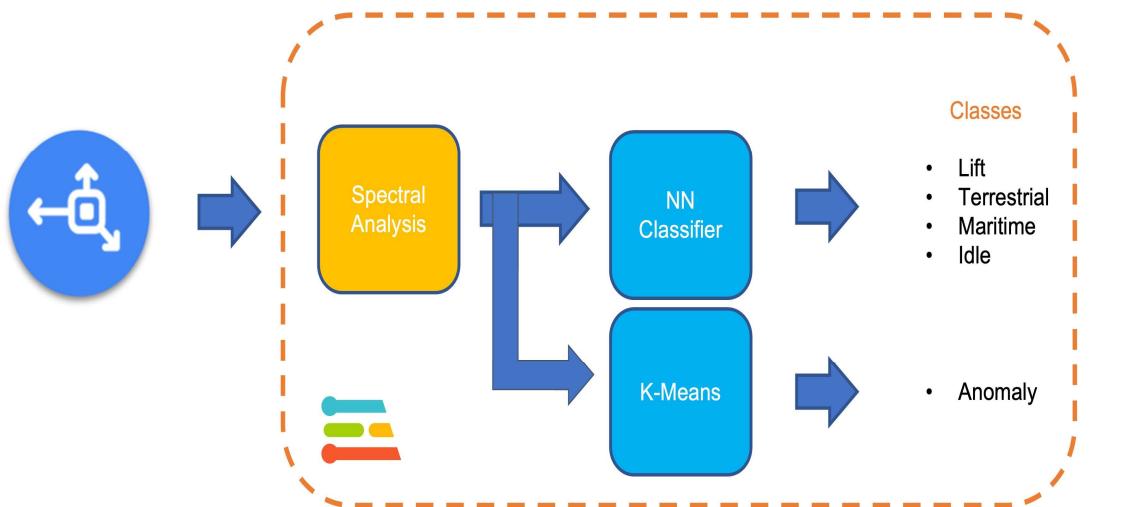
L’interfaccia di Impulse Design dopo l’aggiunta è mostrata nella figura sottostante e ora è stata creata la pipeline di apprendimento automatico.



Inoltre, possiamo anche utilizzare un secondo modello, K-means, che può essere utilizzato per il rilevamento delle anomalie. Se immaginiamo di poter trattare le nostre classi note come cluster, qualsiasi campione che non vi si adatta potrebbe essere un'anomalia (ad esempio, un container che cade in mare quando la nave è in mare).



Per questo, possiamo usare lo stesso tensore di input che entra nel classificatore NN come input per il modello K-means:



Cliccare di nuovo sul pulsante “Add Learning Block” e selezionare **Anomaly Detection (K-means)** nella finestra pop-up sottostante.

**Add a learning block**

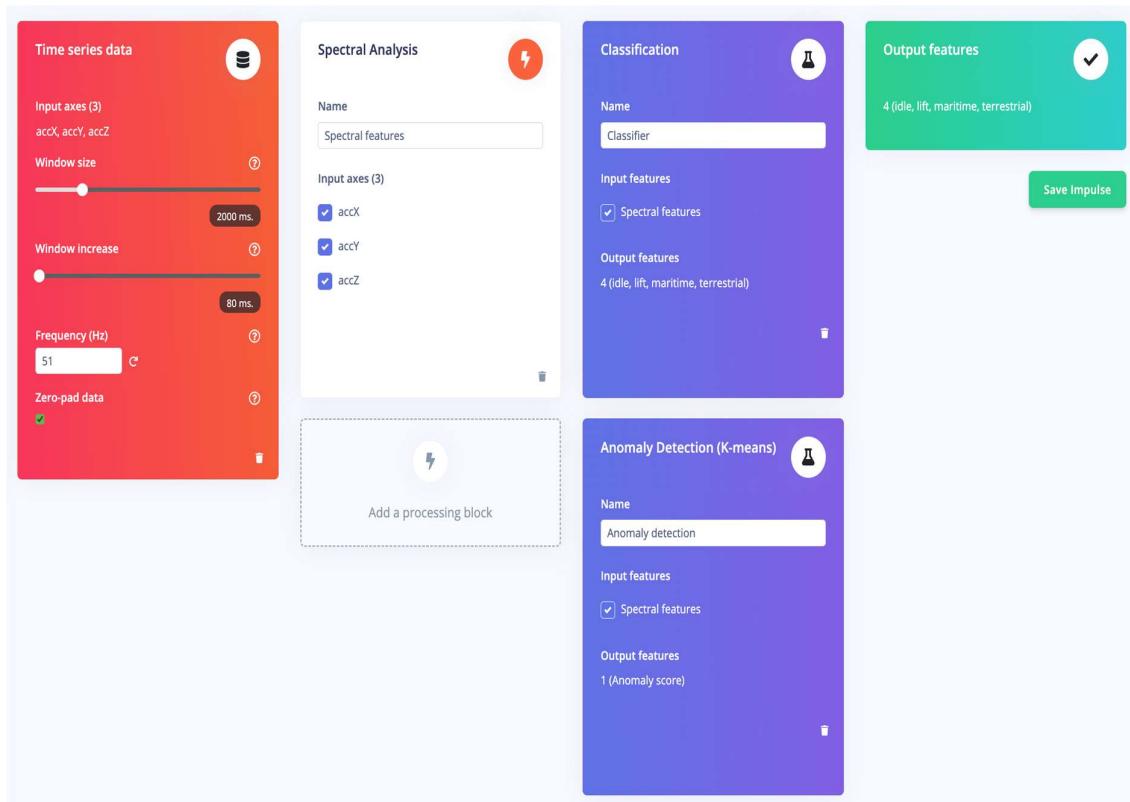
Did you know? You can bring your own model in PyTorch, Keras or scikit-learn.

| DESCRIPTION                                                                                                                                                                                                                      | AUTHOR         | RECOMMENDED |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------------|
| <b>Classification</b><br>Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.                                                                                       | Edge Impulse ★ | <b>Add</b>  |
| <b>Anomaly Detection (K-means)</b><br>Find outliers in new data. Good for recognizing unknown states, and to complement classifiers. Works best with low dimensionality features like the output of the spectral features block. | Edge Impulse ★ | <b>Add</b>  |
| <b>Regression</b><br>Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.                                                                                                 | Edge Impulse   | <b>Add</b>  |

Some learning blocks have been hidden based on the data in your project. [Show all blocks anyway](#)

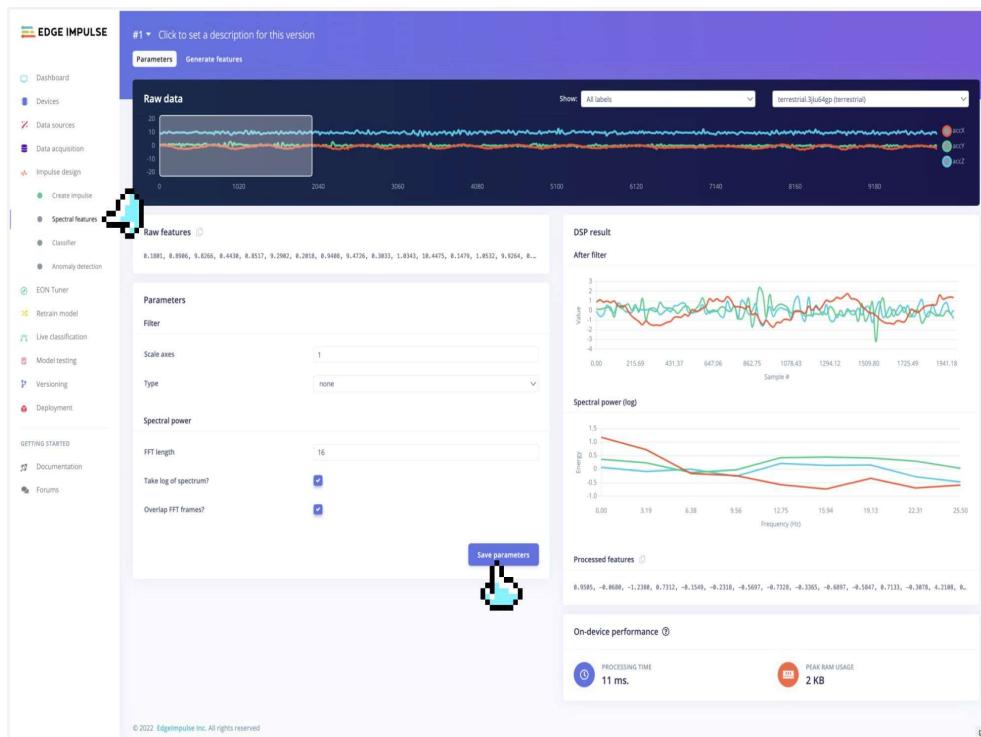
**Cancel**

Il design finale di Impulse è come mostrato nella figura sottostante, cliccare sul pulsante **Save Impulse** all'estrema destra.

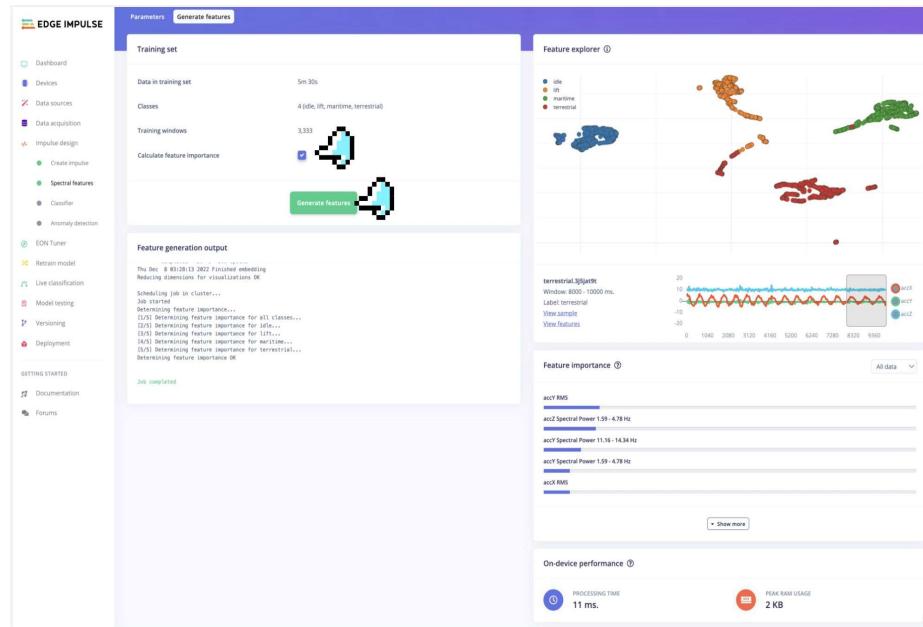


#### 4.2.4.6 Generazione di feature

A questo punto del nostro progetto, abbiamo definito il metodo di pre-elaborazione e il modello progettato. Ora è il momento di terminare il lavoro. Per prima cosa, prendiamo i dati grezzi (tipo serie temporale) e convertiamoli in dati tabulari. Andare alla scheda **Spectral Features**, selezionare **Save Parameters**,



e nel menù in alto, selezionare l'opzione **Generate Features** e il pulsante **Generate Features**:



Ogni dato della finestra di 2 secondi verrà convertito in un punto dati di 63 feature. Feature Explorer mostrerà quei dati in 2D usando **UMAP**.

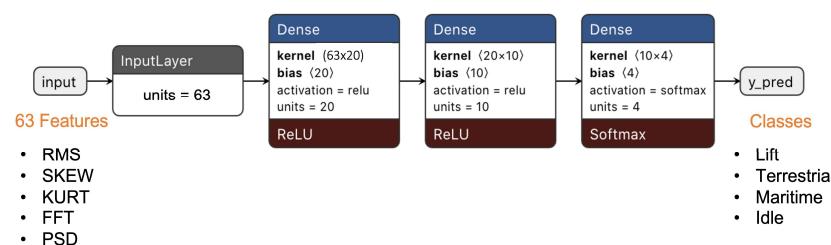
Uniform Manifold Approximation and Projection (UMAP) è una tecnica di riduzione delle dimensioni che può essere usata per la visualizzazione in modo simile a t-SNE ma anche per la riduzione generale delle dimensioni non lineari.

La visualizzazione consente di verificare che le classi presentino un'eccellente separazione, il che indica che il classificatore dovrebbe funzionare bene.

Facoltativamente, si può analizzare quanto è importante ciascuna delle feature per una classe rispetto alle altre.

#### 4.2.4.7 Training

Il nostro modello ha quattro layer, come mostrato di seguito:

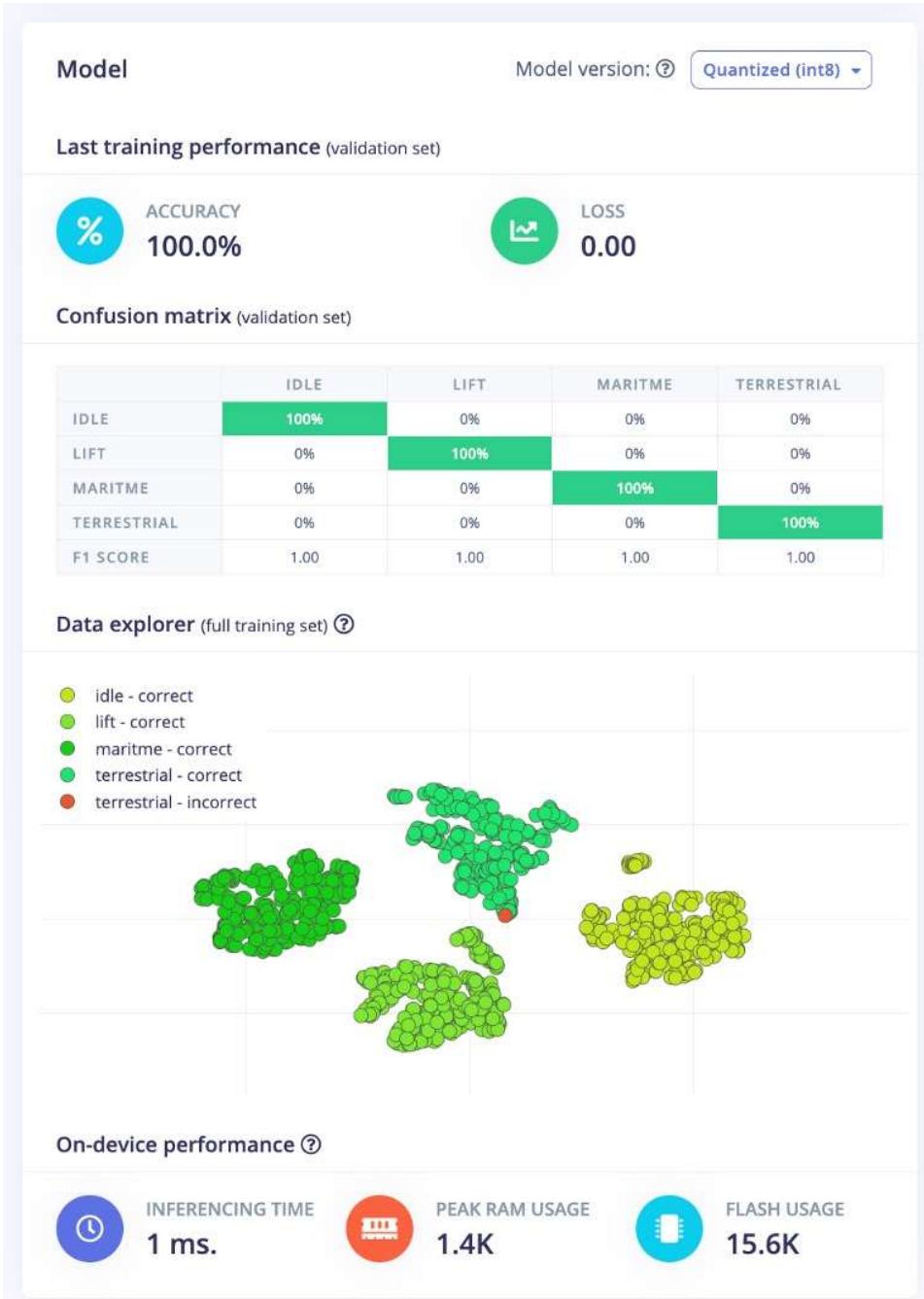


Come iperparametri, useremo un Learning Rate di 0,005 e il 20% di dati per la convalida per 30 epochhe.

The screenshot shows the 'Neural Network settings' page in the Edge Impulse web interface. On the left, a sidebar lists various project management and technical options. The main area is divided into sections: 'Training settings', 'Advanced training settings', and 'Neural network architecture'. In 'Training settings', users can set the number of training cycles (30) and learning rate (0.0005). In 'Advanced training settings', validation set size is set to 20%, and the 'Profile int8 model' option is checked. The 'Neural network architecture' section shows a stack of layers: Input layer (63 features), Dense layer (20 neurons), Dense layer (10 neurons), and Output layer (4 classes). A green 'Start training' button is at the bottom.

image.png

Dopo l'addestramento, possiamo vedere che l'accuratezza è del 100%.



*image.png*

Se è stato aggiunto un blocco K-means per il rilevamento delle anomalie durante la progettazione del modello, verrà visualizzata una sezione aggiuntiva per **Anomaly Detection** nella colonna **Impulse Design** a sinistra, come mostrato nell'immagine sottostante. Una volta all'interno della sezione Anomaly Detection, cliccare su **[Select Suggested Axes]** e il sistema effettuerà automaticamente selezioni in base alle funzionalità importanti calcolate in precedenza. Cliccare, poi, sul pulsante **[Start Training]** per iniziare l'addestramento. I risultati verranno visualizzati in Anomaly Explorer sulla destra dopo il completamento.

The screenshot shows the Edge Impulse interface. On the left, the navigation bar includes 'Dashboard', 'Devices', 'Data sources', 'Data acquisition', 'Impulse design' (with 'Create impulse', 'Spectral features', 'Classifier', and 'Anomaly detection' sub-options), 'EON Tuner', 'Retrain model', 'Live classification', 'Model testing' (selected), 'Versioning', and 'Deployment'. Under 'Model testing', there are 'Getting Started', 'Documentation', and 'Forums'. The main area has two tabs: 'Anomaly detection settings' and 'Anomaly explorer (3,333 samples)'. In the 'Anomaly detection settings' tab, a section for 'Axes' lists various spectral power ranges with checkboxes, one of which is checked and highlighted with a blue circle and a cursor icon. Below this is a 'Start training' button. The 'Anomaly explorer' tab shows a scatter plot of 'accY RMS' vs 'accZ Spectral Power 1.59 - 27.09 Hz'. The plot area is labeled 'trained'. A legend at the top right indicates 'Test data' with a grey square. The bottom of the page contains a large amount of JSON-like code representing training output data.

A questo punto, abbiamo completato il processo di addestramento di base dell'apprendimento automatico.

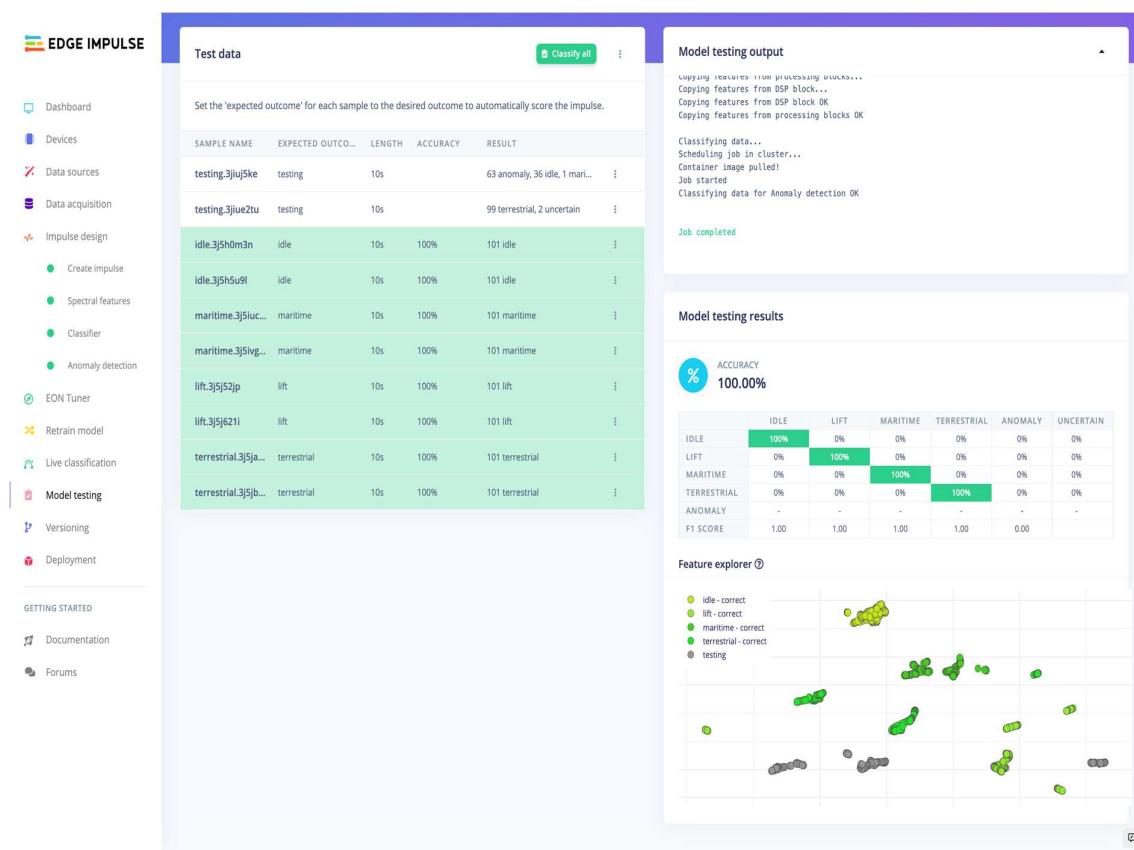
#### 4.2.4.8 Test

Utilizzando il 20% di dati accantonati durante la fase di raccolta dati, possiamo verificare le prestazioni del modello con dati sconosciuti. Come mostrato nell'immagine sottostante, cliccare sulla sezione **Model Testing** sul lato sinistro dell'interfaccia di Edge Impulse. Accanto al pulsante **[classify All]**, c'è un'icona con tre punti, cliccare su di essa per aprire la finestra **Set Confidence Thresholds**. Qui si possono impostare soglie di confidenza per i risultati dei due blocchi di apprendimento. Dovremmo definire una soglia accettabile per i risultati considerati anomalie. Se un risultato non è del 100% (il che accade spesso) ma rientra nell'intervallo di soglia, è comunque utilizzabile.

The screenshot shows the Edge Impulse interface with the 'Model testing' section selected in the sidebar. A red arrow points from the 'Model testing output' section to a 'Set confidence thresholds' dialog box. The dialog box contains two sections: 'Classifier' (Minimum confidence rating: 0.6) and 'Anomaly detection' (Minimum score before tagging as anomaly: 0.3). A green button at the bottom right of the dialog box says 'Set confidence thresholds'. To the right of the dialog box is a table titled 'blocks OK' showing classification results for different samples. At the bottom right is a 'Feature explorer' section with a histogram.

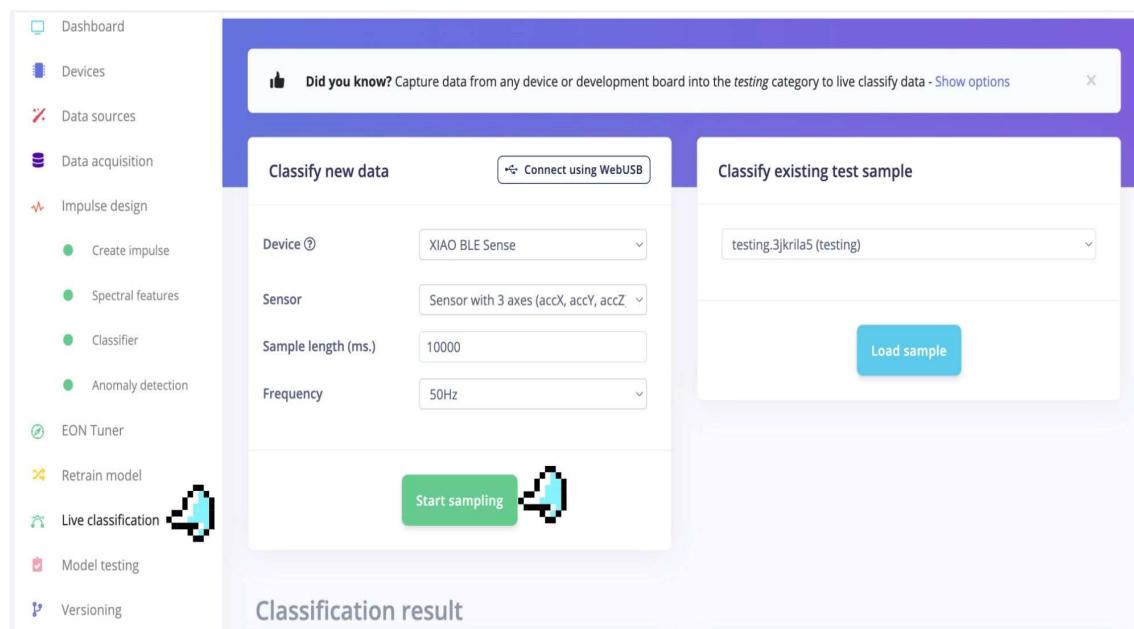
|           | MARITIME | TERRESTRI | ANOMALY | UNCERTA |
|-----------|----------|-----------|---------|---------|
| IDLE      | 100%     | 0%        | 0%      | 0%      |
| LIFT      | 0%       | 100%      | 0%      | 0%      |
| MARITIME  | 0%       | 0%        | 100%    | 0%      |
| TERRESTRI | 0%       | 0%        | 0%      | 0%      |
| ANOMALY   | -        | -         | -       | -       |
| F1 SCORE  | 1.00     | 1.00      | 1.00    | 0.00    |

Premere il pulsante `classify All` per avviare il test del modello. I risultati del test del modello verranno visualizzati al termine, come mostrato nell'immagine sottostante.

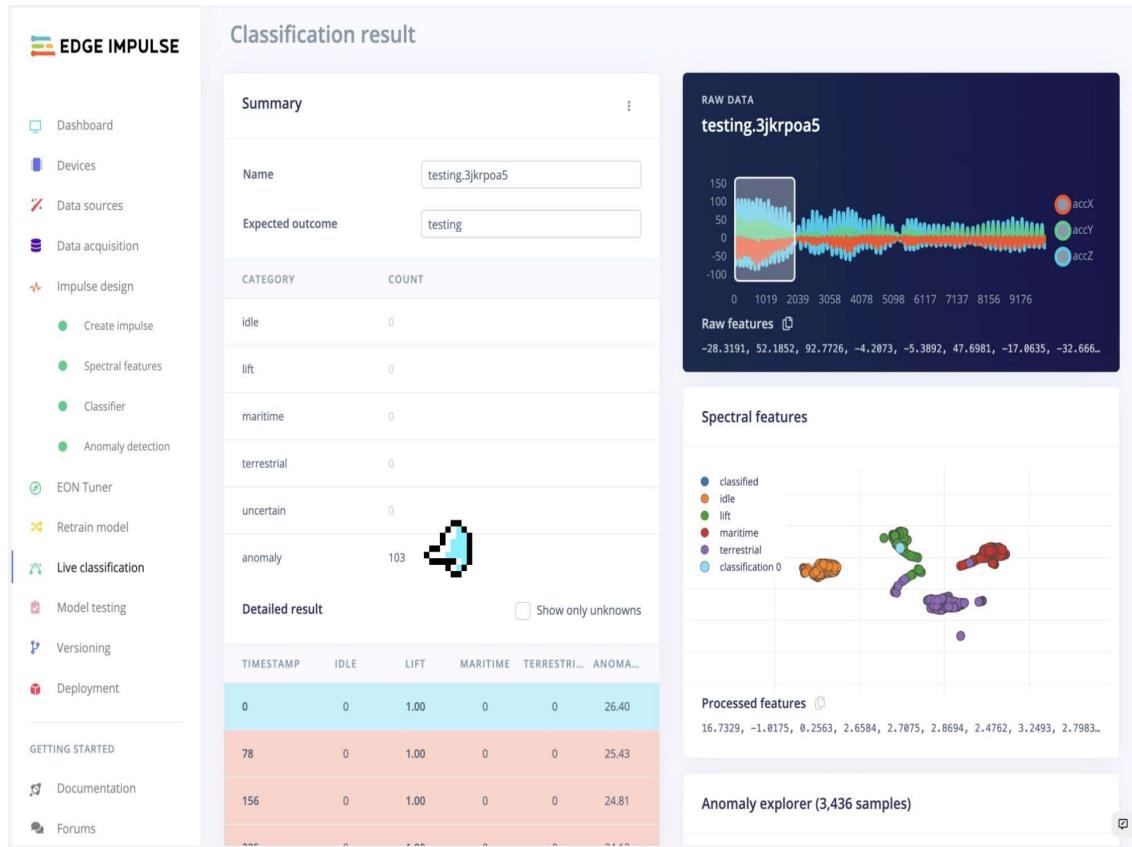


#### 4.2.4.9 Classificazione Live

Una volta ottenuto il modello, si deve sfruttare l'opportunità di testare la classificazione live quando il dispositivo è ancora connesso a Edge Impulse Studio. Come mostrato nell'immagine sottostante, cliccare sulla sezione `Live classification` sul lato sinistro dell'interfaccia di Edge Impulse, poi cliccare sul pulsante `[start sampling]`.



A questo punto, si può, ad esempio, scuotere lo XIAO, il processo è lo stesso del campionamento; attendere qualche secondo e verranno forniti i risultati della classificazione. Come mostrato nell'immagine sottostante, ho scosso vigorosamente lo XIAO e il modello ha dedotto senza esitazione che l'intero processo era **anomalo**.



Provare ora con gli stessi movimenti utilizzati durante l'acquisizione dei dati. Il risultato dovrebbe corrispondere alla classe utilizzata per l'addestramento.

**⚠️ Nota:** Qui, si acquisiranno dati reali con il dispositivo e si caricheranno su Edge Impulse Studio, dove il modello addestrato verrà utilizzato per l'inferenza (anche se il modello non è nel dispositivo).

#### 4.2.4.10 Deployment

Adesso è il momento della magia! Studio impacchetterà tutte le librerie necessarie, le funzioni di pre-elaborazione e i modelli addestrati, scaricandoli sul computer. Si deve selezionare l'opzione “Arduino Library” e, in basso, selezionare `Quantized (Int8)` e `Build`.

The screenshot shows the Edge Impulse web interface for deploying a machine learning model. On the left, a sidebar menu lists various features: Dashboard, Devices, Data acquisition, Impulse design (with sub-options Create impulse, Spectral features, Classifier, Anomaly detection), EON Tuner, Retrain model, Live classification, Model testing, Versioning, Deployment, and sections for Getting Started, Documentation, and Forums.

The main content area is titled "Configure your deployment". It includes a search bar for "Arduino library". Below it, under "SELECTED DEPLOYMENT", is a section for "Arduino library", which is described as an Arduino library with examples that runs on most Arm-based Arduino development boards. There is also a "MODEL OPTIMIZATIONS" section with a toggle switch for "Enable EON™ Compiler" (which offers "Same accuracy, up to 50% less memory").

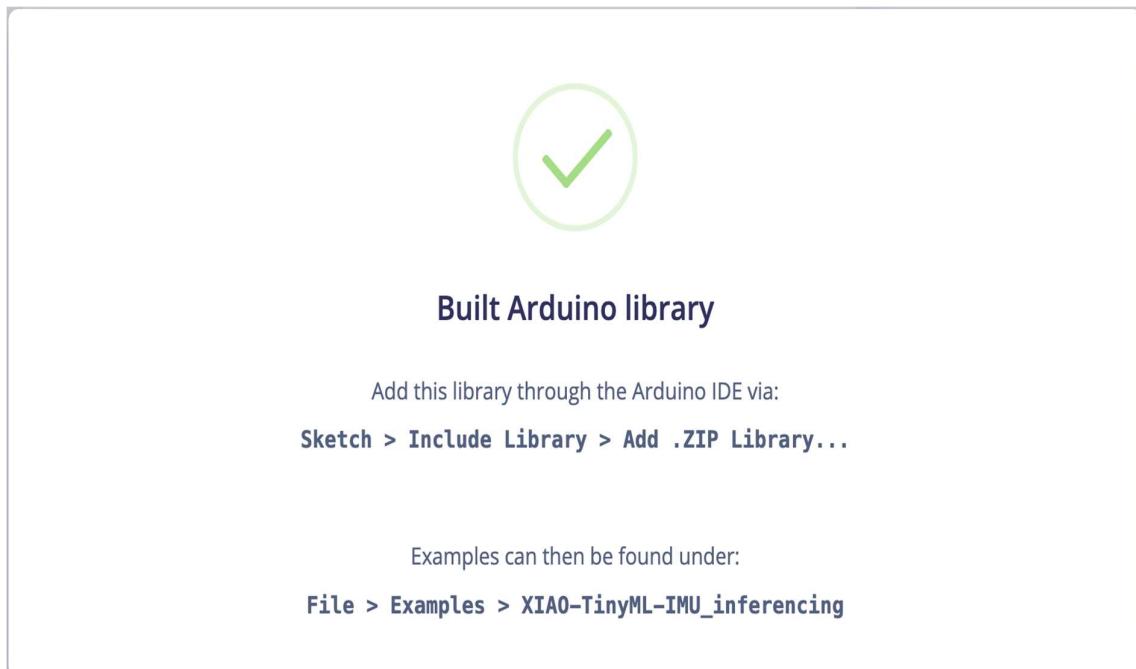
Two tables compare different optimization levels:

|          | SPECTRAL FEATURES | CLASSIFIER | TOTAL  |
|----------|-------------------|------------|--------|
| LATENCY  | 18 ms.            | 1 ms.      | 19 ms. |
| RAM      | 1.9K              | 1.4K       | 1.9K   |
| FLASH    | -                 | 15.6K      | -      |
| ACCURACY | -                 |            |        |

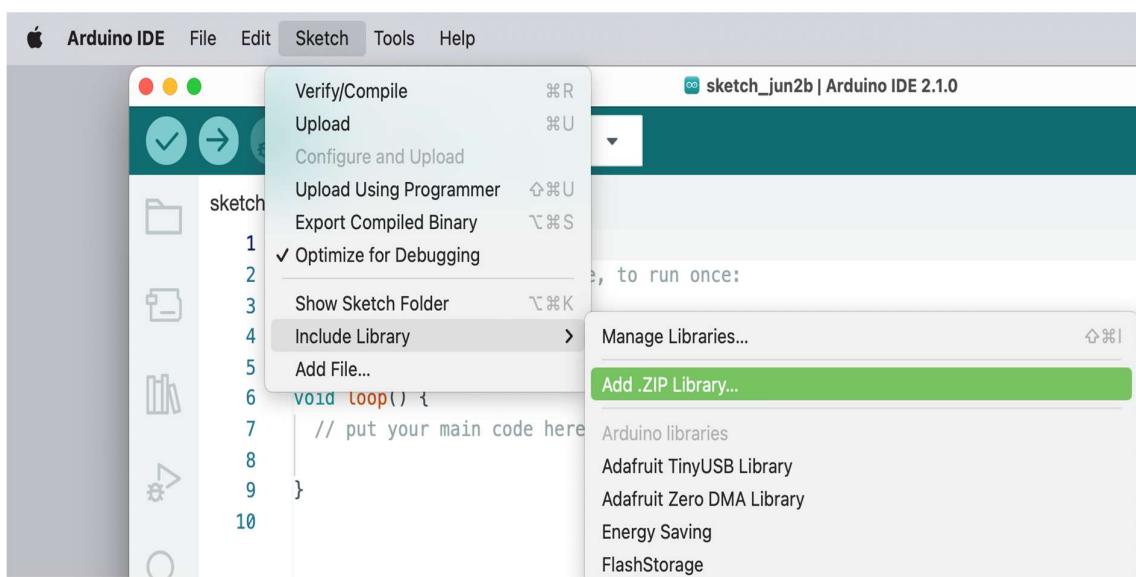
|          | SPECTRAL FEATURES | CLASSIFIER | TOTAL  |
|----------|-------------------|------------|--------|
| LATENCY  | 18 ms.            | 27 ms.     | 45 ms. |
| RAM      | 1.9K              | 1.6K       | 1.9K   |
| FLASH    | -                 | 16.5K      | -      |
| ACCURACY | -                 |            |        |

A note states: "To compare model accuracy, run model testing for all available optimizations." A "Run model testing" button is present. At the bottom, a large blue "Build" button is visible.

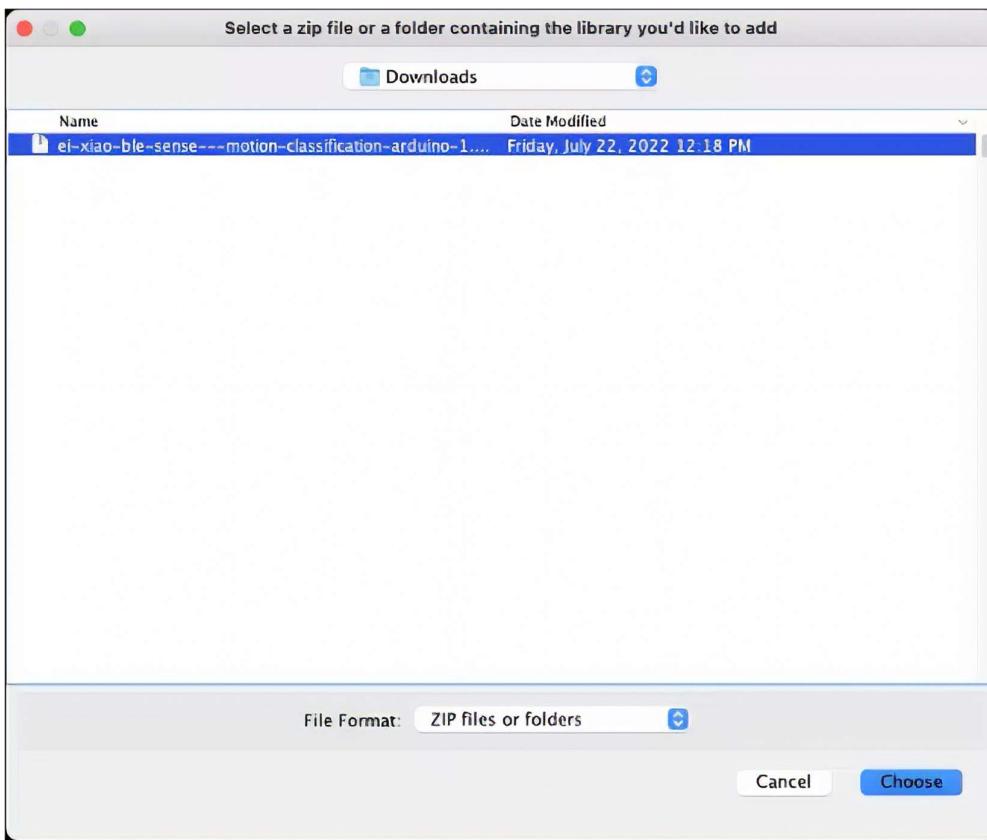
Verrà creato un file Zip e scaricato sul computer.



Sull'Arduino IDE, si va alla scheda `sketch` e si seleziona l'opzione `Add .ZIP Library`.



e si sceglie il file .zip scaricato dallo Studio:



#### 4.2.4.11 Inferenza

Ora è il momento di un vero test. Faremo inferenze completamente scollegate da Studio. Modifichiamo uno degli esempi di codice creati quando si distribuisce la libreria Arduino.

Nell'IDE Arduino, si va alla scheda `File/Examples` e si cerca il proprio progetto, e negli esempi, si seleziona `nano_ble_sense_accelerometer`:

Ovviamente, l'Arduino Nano BLE 33 è diverso dalla propria scheda, la XIAO, ma possiamo far funzionare il codice con solo poche modifiche. Ad esempio, all'inizio del codice, c'è la libreria relativa ad Arduino Sense IMU:

```
/* Includes ----- */
#include <XIAO_BLE_Sense_Motion_Classification_inferencing.h>
#include <Arduino_LSM9DS1.h>
```

Modificare la parte “include” con il codice relativo alla IMU Sense XIAO nRF52840:

```
/* Includes ----- */
#include <XIAO_BLE_Sense_Motion_Classification_inferencing.h>
#include "LSM6DS3.h"
#include "Wire.h"
```

```
//Create an instance of class LSM6DS3
LSM6DS3 xIMU(I2C_MODE, 0x6A); //I2C device address 0x6A
```

Nella funzione `setup`, avviare l'IMU utilizzando il nome che indicato in precedenza:

```
if (xIMU.begin() != 0) {
    ei_printf("Failed to initialize IMU!\r\n");
}
else {
    ei_printf("IMU initialized\r\n");
}
```

Nella funzione `loop`, i buffer: `buffer[ix]`, `buffer[ix + 1]` e `buffer[ix + 2]` riceveranno i dati a 3 assi catturati dall'accelerometro. Nel codice originale, si ha la riga:

```
IMU.readAcceleration(buffer[ix], buffer[ix + 1], buffer[ix + 2]);
```

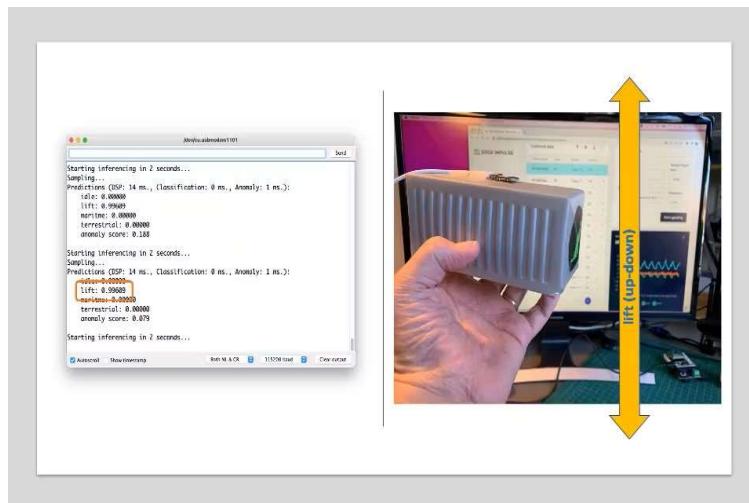
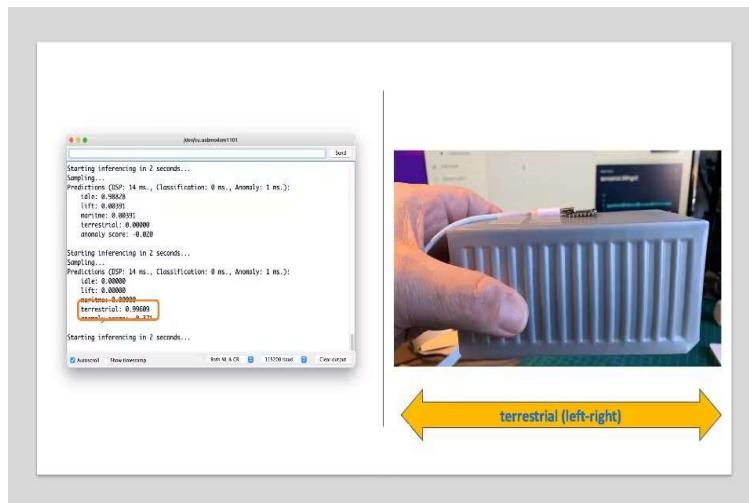
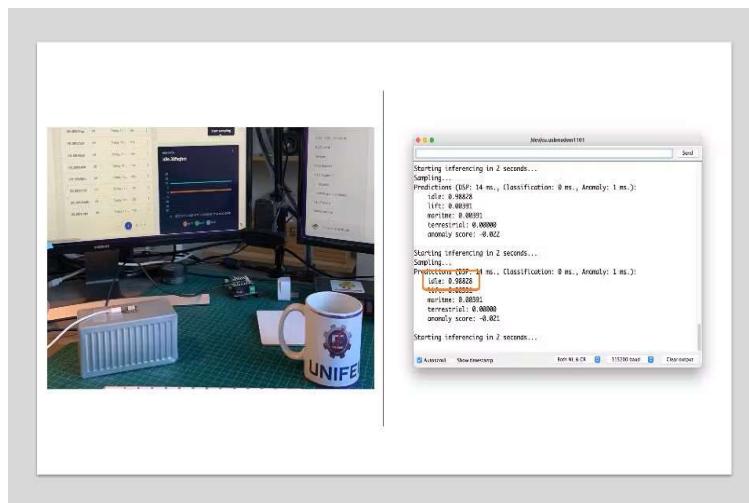
Modificarlo con questo blocco di codice:

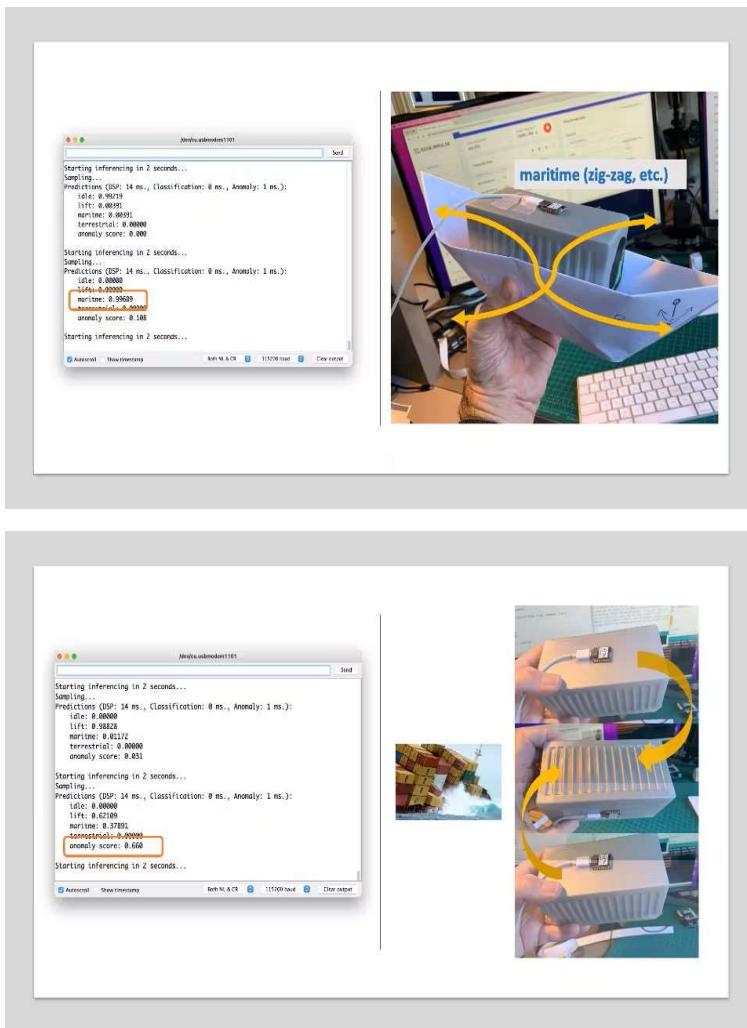
```
buffer[ix]      = xIMU.readFloatAccelX();
buffer[ix + 1] = xIMU.readFloatAccelY();
buffer[ix + 2] = xIMU.readFloatAccelZ();
```

Prelevare questo codice online  [https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/XIAO\\_BLE\\_Sense\\_accelerometer/XIAO\\_BLE\\_Sense\\_accelerometer.ino](https://github.com/Mjrovai/Seeed-XIAO-BLE-Sense/blob/main/XIAO_BLE_Sense_accelerometer/XIAO_BLE_Sense_accelerometer.ino)

E questo è tutto! Ora si può caricare il codice sul dispositivo e procedere con le inferenze.

È possibile vedere il risultato dell'inferenza di ciascuna classe nelle immagini:





## Post-elaborazione

Ora che sappiamo che il modello funziona poiché rileva i movimenti, suggeriamo di modificare il codice per vedere il risultato con XIAO completamente offline (scollegato dal PC e alimentato da una batteria, un power bank o un alimentatore indipendente da 5 V).

L'idea è che se viene rilevato un movimento specifico, potrebbe accendersi un LED particolare. Ad esempio, se viene rilevato un movimento *terrestre*, si accenderà il LED verde; se è un movimento *marittimo*, si accenderà il LED rosso, se è un *ascensore*, si accenderà il LED blu; e se non viene rilevato alcun movimento (*inattivo*), i LED saranno SPENTI. Si può anche aggiungere una condizione quando viene rilevata un'anomalia, in questo caso, ad esempio, può essere utilizzato un colore bianco (tutti i LED e si accendono contemporaneamente).

### 4.2.4.12 Conclusione

Seeed Studio XIAO nRF52840 Sense è un minuscolo dispositivo gigantesco! È potente, affidabile, non costoso, a basso consumo e ha sensori adatti per essere utilizzato nelle applicazioni di apprendimento automatico embedded più comuni. Sebbene Edge Impulse non supporti ufficialmente XIAO nRF52840 Sense, ci siamo resi conto che potrebbe essere facilmente collegato allo Studio.

Nel repository GitHub, c'è l'ultima versione dei codici: [Seeed-XIAO-BLE-Sense](#).

Le applicazioni per la classificazione del movimento e il rilevamento delle anomalie sono estese e XIAO è adatto per scenari in cui il basso consumo energetico e l'elaborazione edge

sono vantaggiosi. Il suo piccolo fattore di forma e l'efficienza nell'elaborazione lo rendono una scelta ideale per l'implementazione di applicazioni portatili e remote in cui l'elaborazione in tempo reale è fondamentale e la connettività potrebbe essere limitata.

#### 4.2.4.13 Applicazioni di Casi

Prima di concludere, considerare che la classificazione del movimento e il rilevamento degli oggetti possono essere utilizzati in molte applicazioni in vari domini. Ecco alcune delle potenziali applicazioni:

##### **Industriale e Manifatturiero**

- **Manutenzione Predittiva:** Rilevamento di anomalie nel movimento dei macchinari per prevedere guasti prima che si verifichino.
- **Controllo Qualità:** Monitoraggio del movimento di linee di montaggio o bracci robotici per la valutazione della precisione e il rilevamento di deviazioni dal modello di movimento standard.
- **Logistica di Magazzino:** Gestione e monitoraggio del movimento delle merci con sistemi automatizzati che classificano diversi tipi di movimento e rilevano anomalie nella movimentazione.

##### **Sanità**

- **Monitoraggio dei Pazienti:** Rilevamento di cadute o movimenti anomali negli anziani o in coloro con problemi di mobilità.
- **Riabilitazione:** Monitoraggio dei progressi dei pazienti in fase di recupero da infortuni mediante la classificazione dei modelli di movimento durante le sedute di fisioterapia.
- **Riconoscimento dell'Attività:** Classificazione dei tipi di attività fisica per applicazioni di fitness o monitoraggio dei pazienti.

##### **Elettronica di Consumo**

- **Controllo Gestuale:** Interpretazione di movimenti specifici per controllare dispositivi, come accendere le luci con un gesto della mano.
- **Giochi:** Miglioramento delle esperienze di gioco con input controllati dal movimento.

##### **Trasporti e Logistica**

- **Telematica dei Veicoli:** Monitoraggio del movimento del veicolo per comportamenti insoliti come frenate brusche, curve strette o incidenti.
- **Monitoraggio del Carico:** Garanzia dell'integrità delle merci durante il trasporto rilevando movimenti insoliti che potrebbero indicare manomissioni o cattiva gestione.

##### **Città Intelligenti e Infrastrutture**

- **Monitoraggio della Salute Strutturale:** Rilevamento di vibrazioni o movimenti all'interno delle strutture che potrebbero indicare potenziali guasti o esigenze di manutenzione.
- **Gestione del Traffico:** Analisi del flusso di pedoni o veicoli per migliorare la mobilità e la sicurezza urbana.

##### **Sicurezza e Sorveglianza**

- **Rilevamento Intrusi:** Rilevamento di modelli di movimento tipici di accessi non autorizzati o altre violazioni della sicurezza.
- **Monitoraggio della Fauna Selvatica:** Rilevamento di bracconieri o movimenti anomali di animali in aree protette.

### Agricoltura

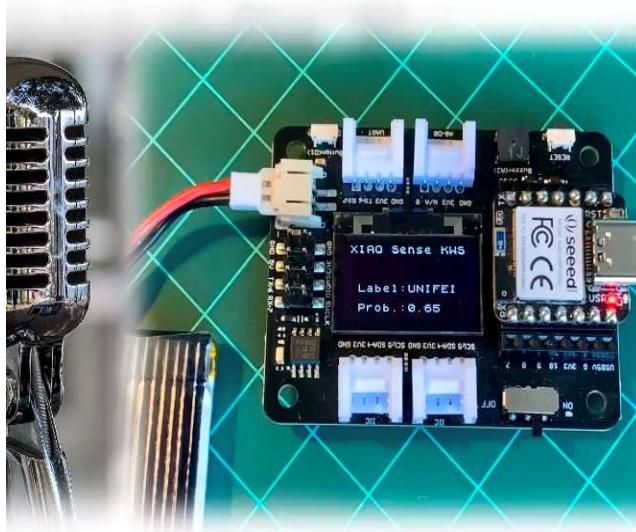
- **Monitoraggio delle Attrezzature:** Monitoraggio delle prestazioni e dell'utilizzo di macchinari agricoli.
- **Analisi del Comportamento Animale:** Monitoraggio dei movimenti del bestiame per rilevare comportamenti che indicano problemi di salute o stress.

### Monitoraggio Ambientale

- **Attività Sismica:** Rilevamento di pattern di movimento irregolari che precedono terremoti o altri eventi geologicamente rilevanti.
- **Oceanografia:** Studio di pattern di onde o movimenti marini per scopi di ricerca e sicurezza.

## 4.3 Classificazione del Suono (KWS)

In questa sezione, continuiamo la nostra esplorazione del Machine Learning su Seeed Studio XIAO nRF52840 Sense (chiamato anche XIAO BLE Sense), che ora classifica le onde sonore.



### 4.3.1 Cose utilizzate in questo progetto

#### Componenti hardware

- Seeed Studio XIAO nRF52840 Sense × 1
- Seeed Studio Seeeduino XIAO Expansion board × 1

#### App software e servizi online

- Arduino IDE 
- Edge Impulse Studio 

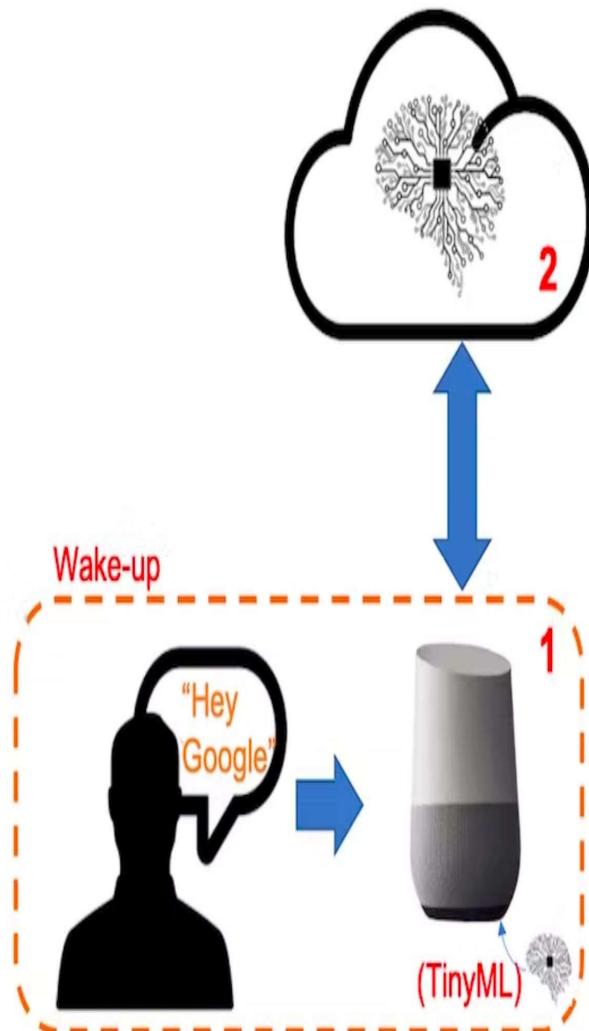
### 4.3.2 Introduzione

Nell'ultima sezione, **Rilevamento di Anomalie e Classificazione del Movimento**, abbiamo esplorato Embedded Machine Learning, o semplicemente TinyML, in esecuzione su **Seeed XIAO nRF52840 Sense**. Oltre a installare e testare il dispositivo, abbiamo esplorato la classificazione del movimento utilizzando segnali di dati effettivi dal suo accelerometro di bordo. Questo nuovo progetto utilizzerà lo stesso XIAO nRF52840 Sense per classificare il

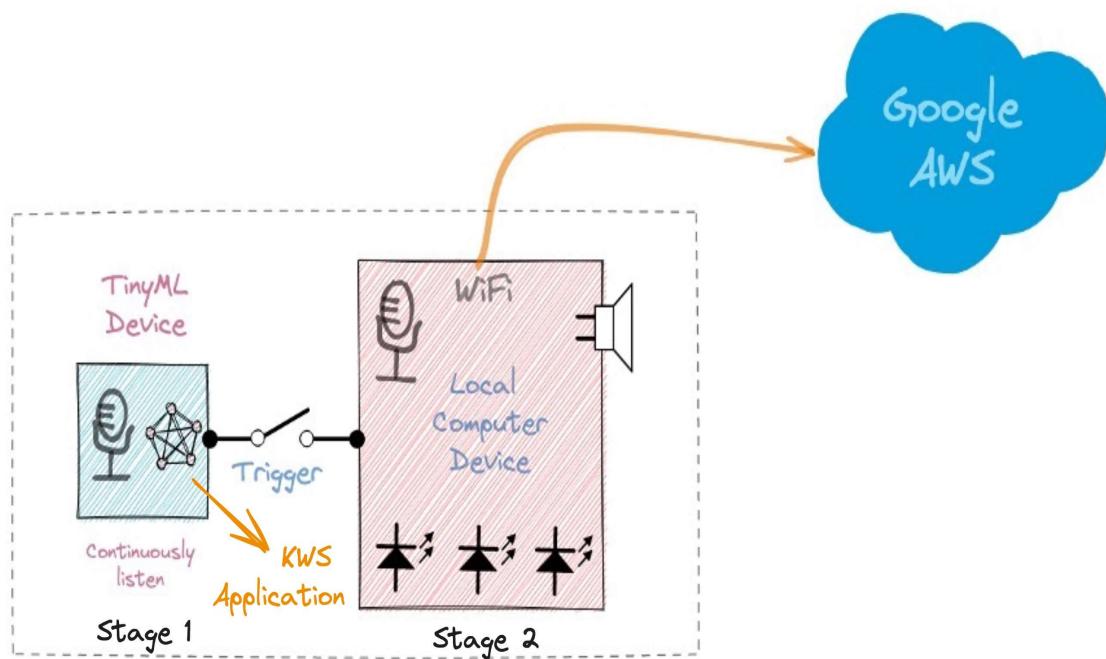
suono, funzionando esplicitamente come “Key Word Spotting” (KWS). Un KWS è una tipica applicazione TinyML e una parte essenziale di un assistente vocale.

### Ma come funziona un assistente vocale?

Per iniziare, è essenziale rendersi conto che gli assistenti vocali sul mercato, come Google Home o Amazon Echo-Dot, reagiscono agli umani solo quando vengono “svegliati” da parole chiave specifiche come “Hey Google” sul primo e “Alexa” sul secondo.



In altre parole, il riconoscimento dei comandi vocali si basa su un modello multi-fase o Cascade Detection.



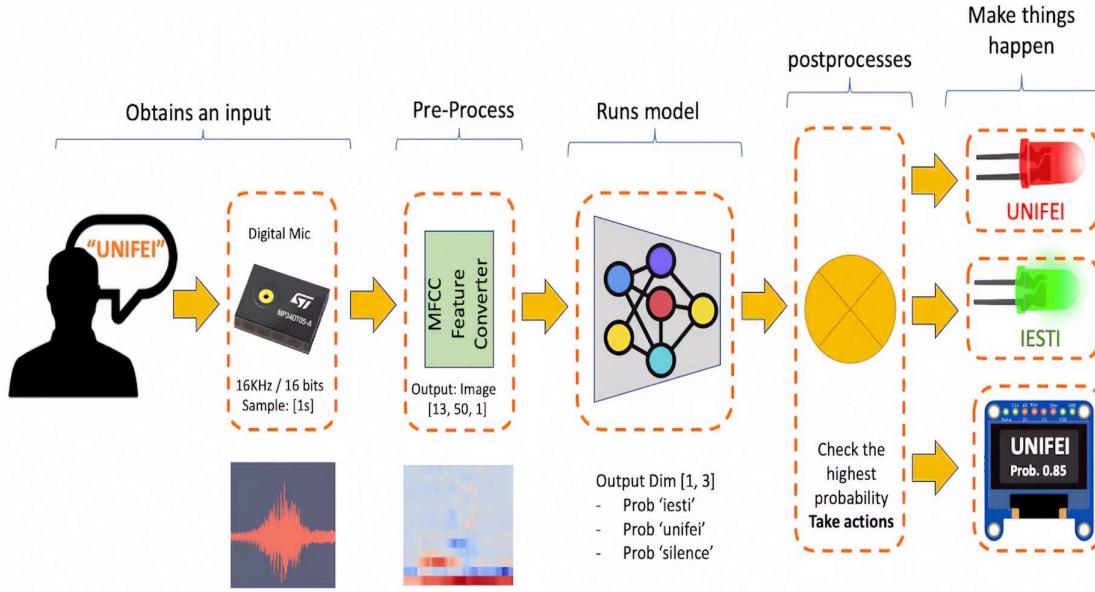
**Fase 1:** Un microprocessore più piccolo all'interno di *Echo Dot* o *Google Home* ascolta continuamente il suono, in attesa che venga individuata la parola chiave. Per tale rilevamento, viene utilizzato un modello TinyML all'edge (applicazione KWS).

**Fase 2:** Solo quando vengono attivati dall'applicazione KWS nella Fase 1, i dati vengono inviati al cloud ed elaborati su un modello più grande.

Il video qui sotto mostra un esempio di un Google Assistant programmato su un Raspberry Pi (Fase 2), con un Arduino Nano 33 BLE come dispositivo tinyML (Fase 1);  
[https://youtu.be/e\\_OPgcnsyvM](https://youtu.be/e_OPgcnsyvM)

Per esplorare il progetto Google Assistant di cui sopra, consultare il tutorial:  
[Building an Intelligent Voice Assistant From Scratch](#).

### 4.3.3 Il progetto KWS



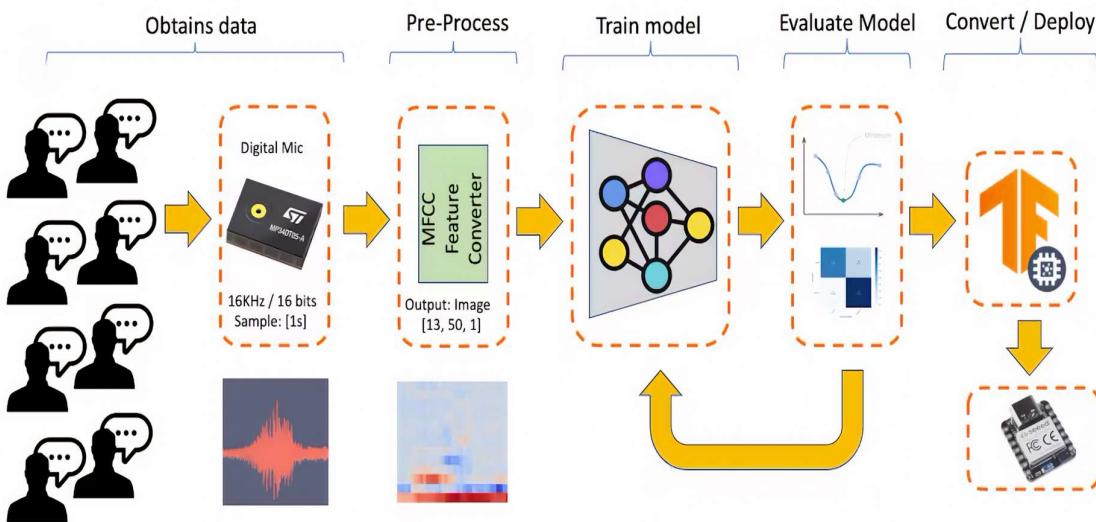
La nostra applicazione KWS riconoscerà tre classi di suono:

- Parola chiave 1: **UNIFEI**
- Parola chiave 2: **IESTI**
- “**SILENZIO**” (nessuna parola chiave pronunciata, è presente solo rumore di fondo)

Facoltativamente, per progetti reali, si consiglia di includere parole diverse dalle parole chiave 1 e 2 nella classe “Silenzio” (o Sfondo) o persino creare una classe extra con tali parole (ad esempio una classe “altri”).

#### 4.3.3.1 Il Flusso di Lavoro del Machine Learning

Il componente principale dell'applicazione KWS è il suo modello. Quindi, dobbiamo addestrare tale modello con le nostre parole chiave specifiche:

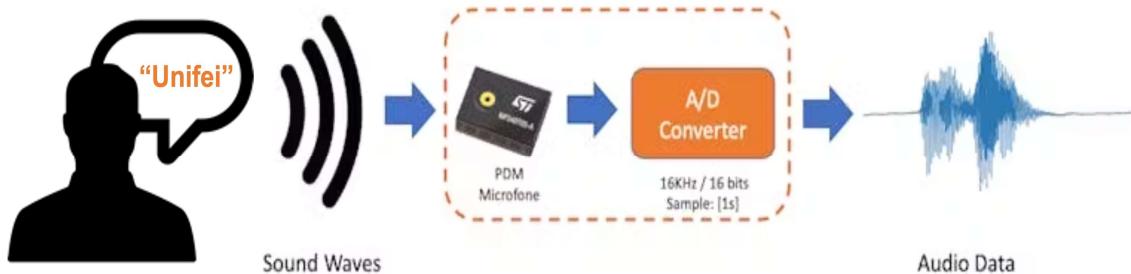


#### 4.3.3.2 Il Dataset

Il componente critico del flusso di lavoro di apprendimento automatico è il dataset. Una volta decise le parole chiave specifiche (UNIFEI e IESTI), tutti i dataset dovrebbero essere creati da zero. Lavorando con gli accelerometri, creare un dataset con dati acquisiti dallo stesso tipo di sensore era essenziale. Nel caso del suono, è diverso perché classificheremo come dati audio.

La differenza fondamentale tra suono e audio è il tipo di energia. Il suono è una perturbazione meccanica (onde sonore longitudinali) che si propagano attraverso un mezzo, causando variazioni di pressione in esso. L'audio è un segnale elettrico (analogico o digitale) che rappresenta il suono.

Le onde sonore dovrebbero essere convertite in dati audio quando pronunciamo una parola chiave. La conversione dovrebbe essere eseguita campionando il segnale generato dal microfono a 16 KHz con una profondità di 16 bit.



Quindi, qualsiasi dispositivo in grado di generare dati audio con questa specifica di base (16Khz/16bit) funzionerà bene. Come dispositivo, possiamo usare il corretto XIAO nRF52840 Sense, un computer o persino il cellulare.

#### 4.3.3.3 Acquisizione di dati audio online con Edge Impulse e uno smartphone

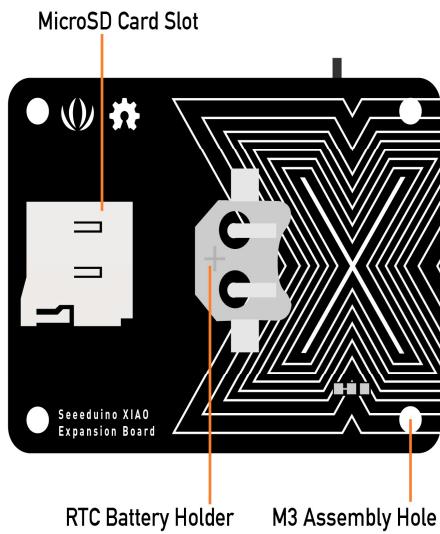
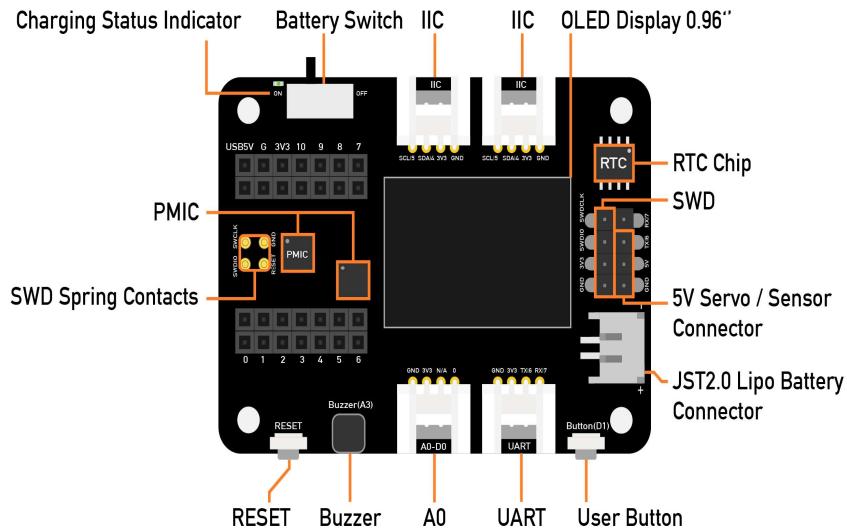
Nella sezione *Rilevamento di Anomalie e Classificazione del Movimento*, abbiamo imparato come installare e testare il nostro dispositivo utilizzando l'IDE Arduino e collegarlo a Edge Impulse Studio per l'acquisizione dei dati. Per questo, utilizziamo la *funzione EI CLI Data Forwarder*, ma secondo Jan Jongboom, CTO di Edge Impulse, l'audio è troppo veloce perché il data forwarder possa essere acquisito. Se si hanno già dati PCM, trasformarli in un file WAV e caricarli con l'uploader è la soluzione più semplice. Con gli accelerometri, la nostra frequenza di campionamento era di circa 50 Hz, con audio a 16 KHz.

Quindi, non possiamo collegare XIAO direttamente allo Studio. Ma possiamo acquisire l'audio utilizzando qualsiasi smartphone connesso allo Studio online.

Non esploreremo questa opzione qui, ma si può facilmente seguire la [documentazione](#) e il [tutorial](#) El.

#### 4.3.3.4 Acquisizione di Dati Audio con XIAO nRF52840 Sense

Il modo più semplice per acquisire l'audio e salvarlo localmente come file .wav è utilizzare una scheda di espansione per la famiglia di dispositivi XIAO, la [scheda di espansione Seeed Studio XIAO](#).

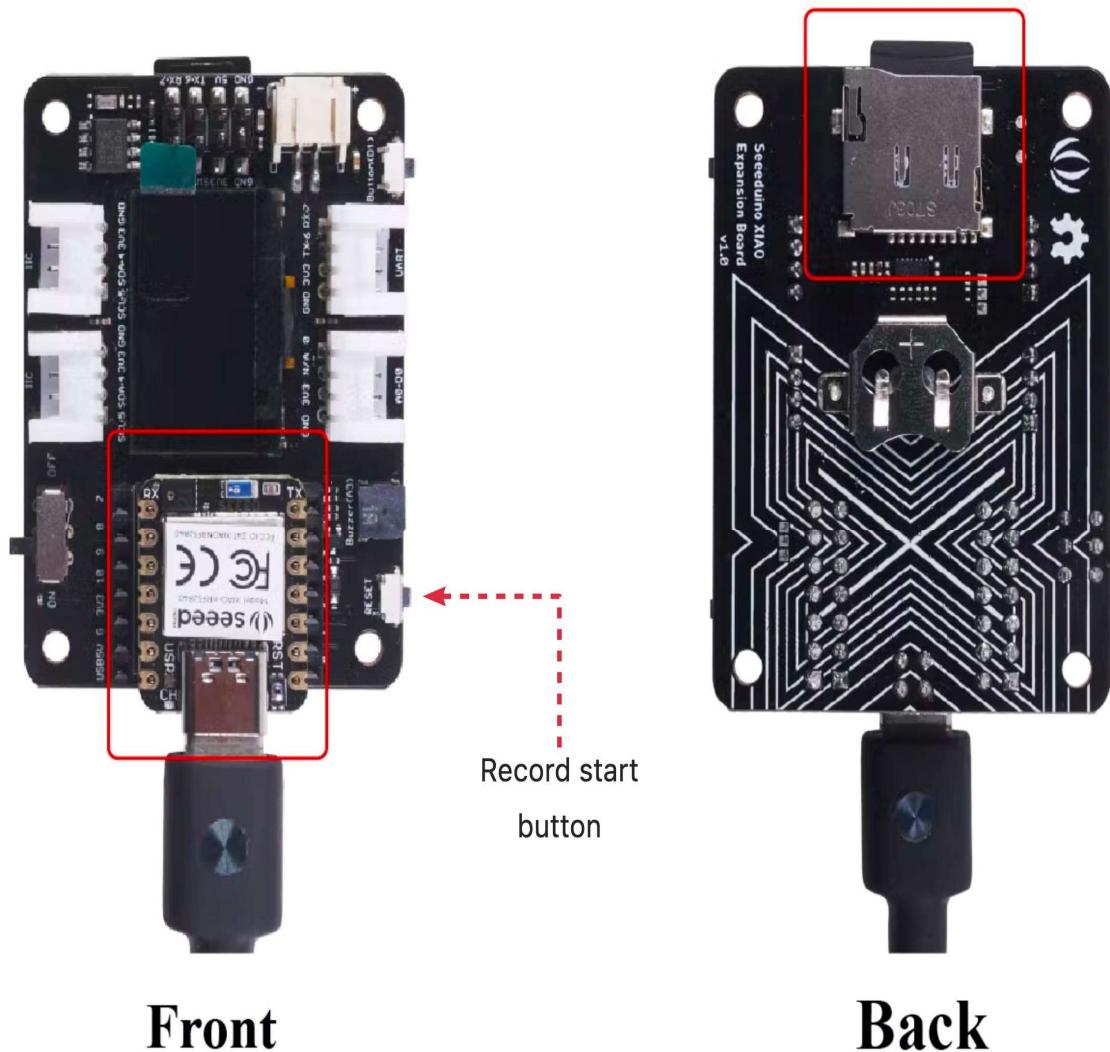


Questa scheda di espansione consente di creare prototipi e progetti in modo semplice e rapido, utilizzando le sue periferiche avanzate come display OLED, interfaccia scheda SD, RTC, buzzer passivo, pulsante RESET/utente, connettore servo 5V e interfacce dati multiple.

Questo progetto si concentrerà sulla classificazione delle parole chiave e la scheda MicroSD disponibile sul dispositivo sarà molto importante per aiutarci con l'acquisizione dei dati.

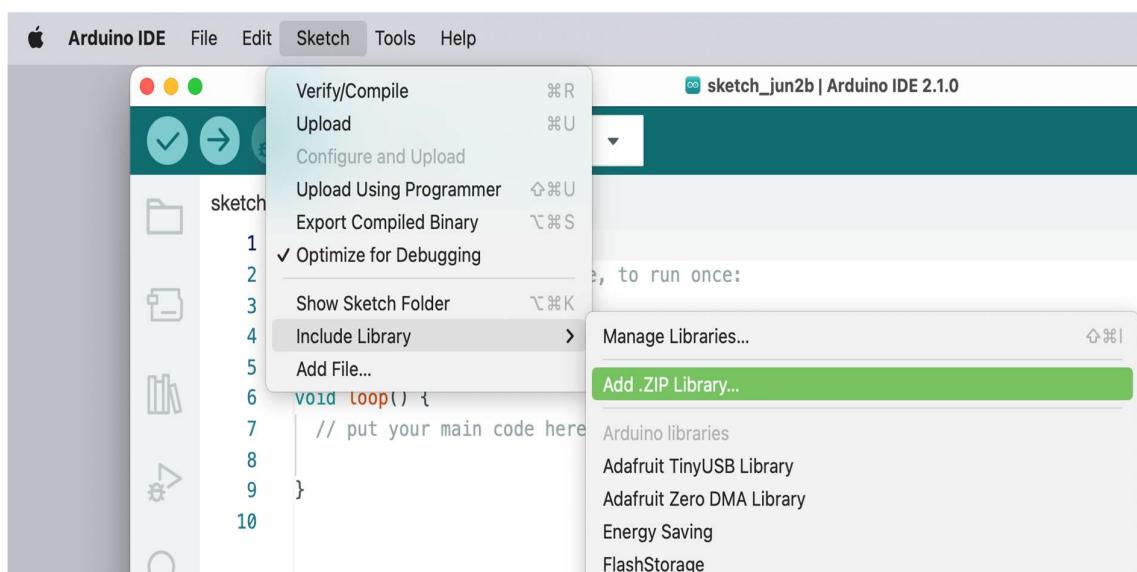
### **Salvataggio dell'audio registrato dal microfono su una scheda SD**

Collegare XIAO nRF52840 Sense sulla scheda di espansione e inserire una scheda SD nello slot per schede SD sul retro. > La scheda SD deve essere preformattata come FAT o exFAT.



Scaricare poi la [Seeed\\_Arduino\\_FS Library](#) come file zip:

E installare la libreria scaricata: `seeed_Arduino_Mic-master.zip` sull'Arduino IDE: Sketch -> Include Library -> Add .ZIP Library...



Successivamente, si va su File > Examples > Seeed Arduino Mic > mic\_Saved\_OnSDcard per aprire lo sketch: **mic\_Saved\_OnSDcard**.

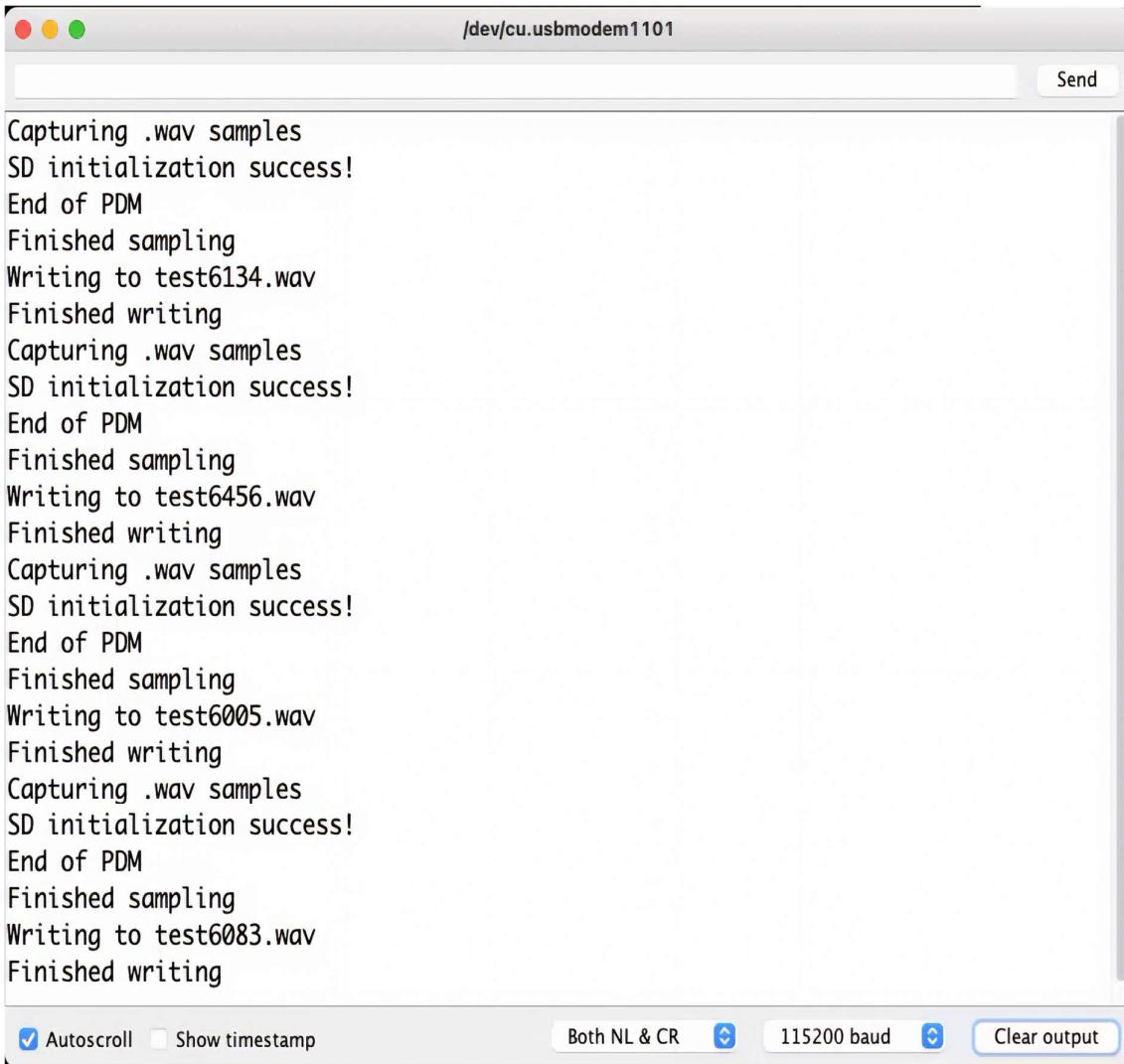
Ogni volta che si preme il pulsante di reset, un **campione audio di 5 secondi** viene registrato e salvato sulla scheda SD. Il file originale è stato modificato per aggiungere LED per aiutare durante il processo di registrazione come di seguito:

- Durante il tempo in cui il LED rosso è ACCESO è possibile registrare ==> REGISTRARE
- Durante il processo di scrittura del file, il LED rosso è SPENTO ==> ATTENDERE
- Al termine della scrittura, il LED verde è ACCESO ==> Premere il pulsante di reset una volta e attendere che il LED rosso si accenda di nuovo, quindi procedere con una nuova registrazione del campione

Mi sono accorto che a volte all'inizio e alla fine di ogni campione veniva registrato uno "spike", quindi ho tagliato i 300 ms iniziali da ogni campione di 5 s. Lo spike alla fine si è sempre verificato dopo il processo di registrazione e dovrebbe essere eliminato su Edge Impulse Studio prima del training. Inoltre, è stato aumentato il guadagno del microfono a 30 dB.

Il file completo (Xiao\_mic\_Saved\_OnSDcard.ino) si trova su Git Hub (3\_KWS): [Seeed-XIAO-BLE-Sense](#).

Durante il processo di registrazione, i nomi dei file .wav vengono mostrati su Serial Monitor:

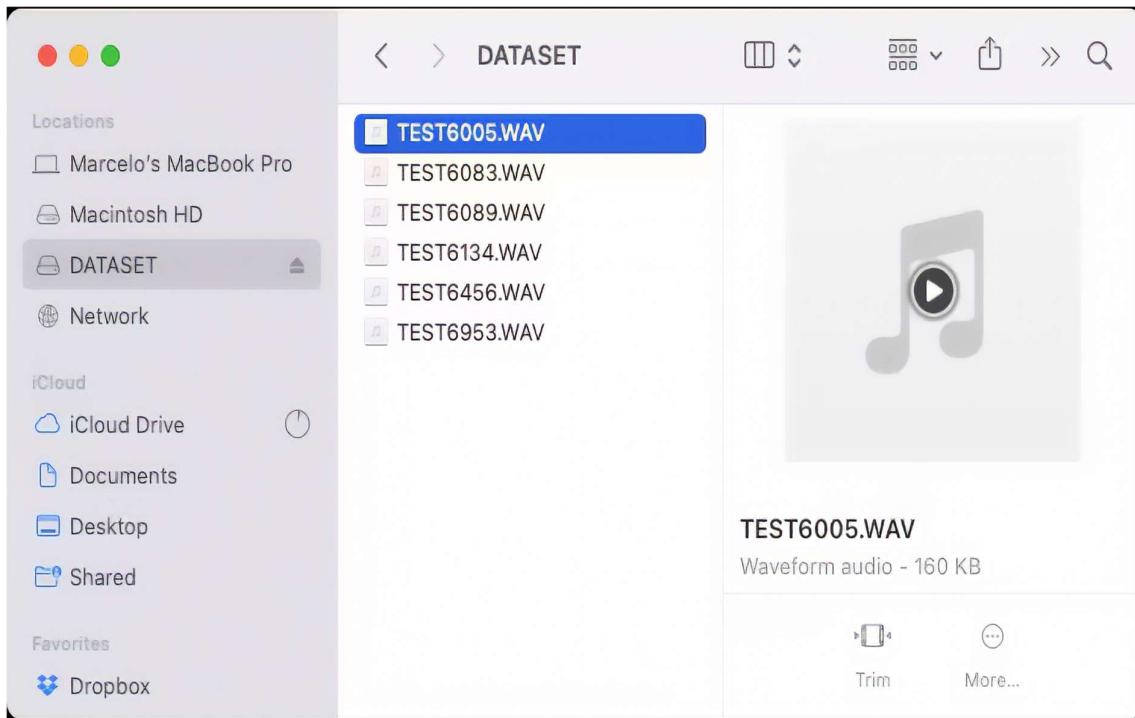


The screenshot shows a Mac OS X-style terminal window titled '/dev/cu.usbmodem1101'. The window contains the following text output from a serial monitor session:

```
Capturing .wav samples
SD initialization success!
End of PDM
Finished sampling
Writing to test6134.wav
Finished writing
Capturing .wav samples
SD initialization success!
End of PDM
Finished sampling
Writing to test6456.wav
Finished writing
Capturing .wav samples
SD initialization success!
End of PDM
Finished sampling
Writing to test6005.wav
Finished writing
Capturing .wav samples
SD initialization success!
End of PDM
Finished sampling
Writing to test6083.wav
Finished writing
```

At the bottom of the window, there are several configuration buttons: 'Autoscroll' (checked), 'Show timestamp' (unchecked), 'Both NL & CR' (selected), '115200 baud' (selected), and 'Clear output'.

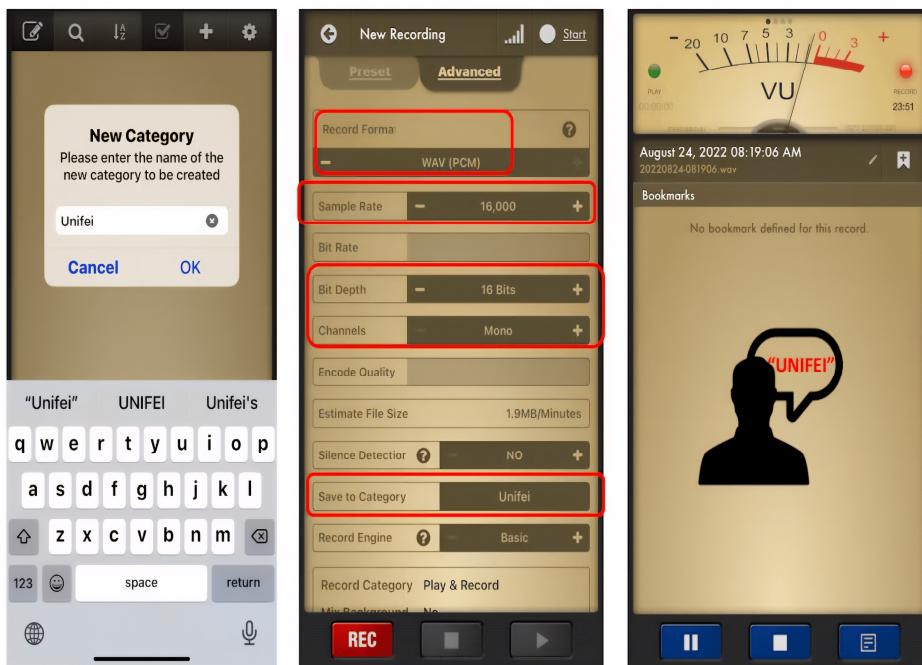
Prendere la scheda SD dalla scheda di espansione e inserirla nel computer:



I file sono pronti per essere caricati su Edge Impulse Studio

#### 4.3.3.5 Acquisizione (offline) di Dati Audio con uno smartphone o un PC

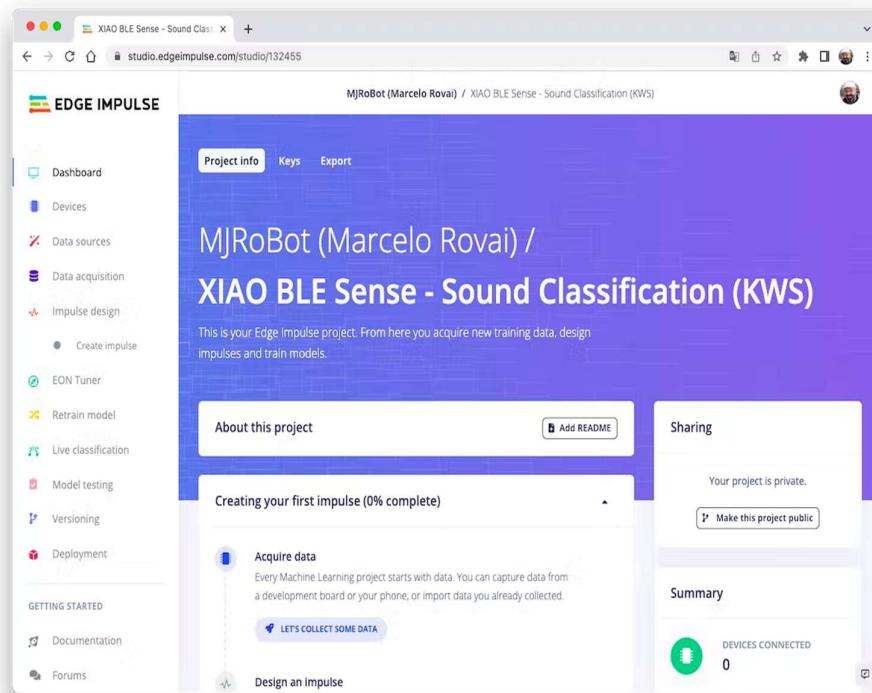
In alternativa, si può usare il PC o lo smartphone per acquisire dati audio con una frequenza di campionamento di 16 KHz e una profondità di 16 bit. Una buona app per questo è [Voice Recorder Pro](#) (IOS). Salvare la registrazione come file .wav e inviarla al computer.



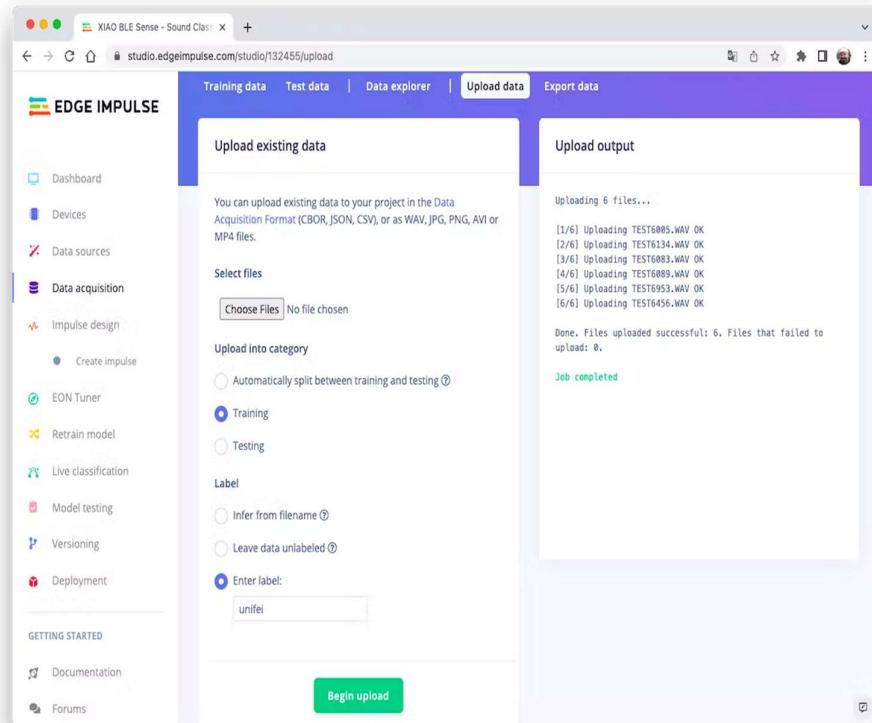
Notare che qualsiasi app per smartphone può essere usata per la registrazione audio o anche il computer, ad esempio usando [Audacity](#).

#### 4.3.3.6 Modello di training con Edge Impulse Studio

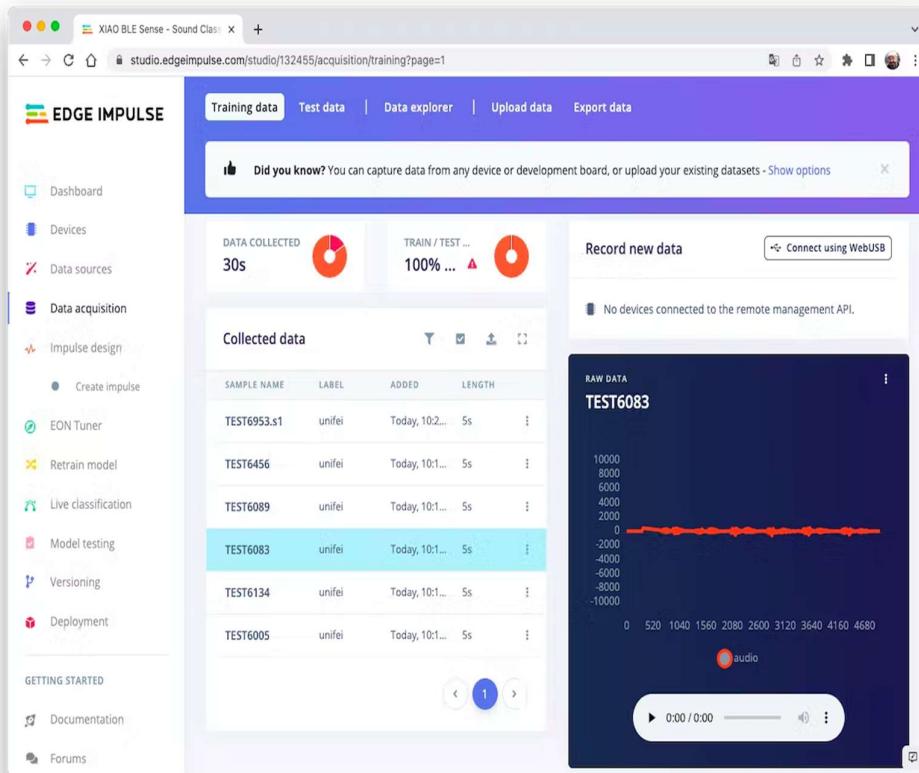
Quando viene creato il dataset grezzo, si deve avviare un nuovo progetto in Edge Impulse Studio:



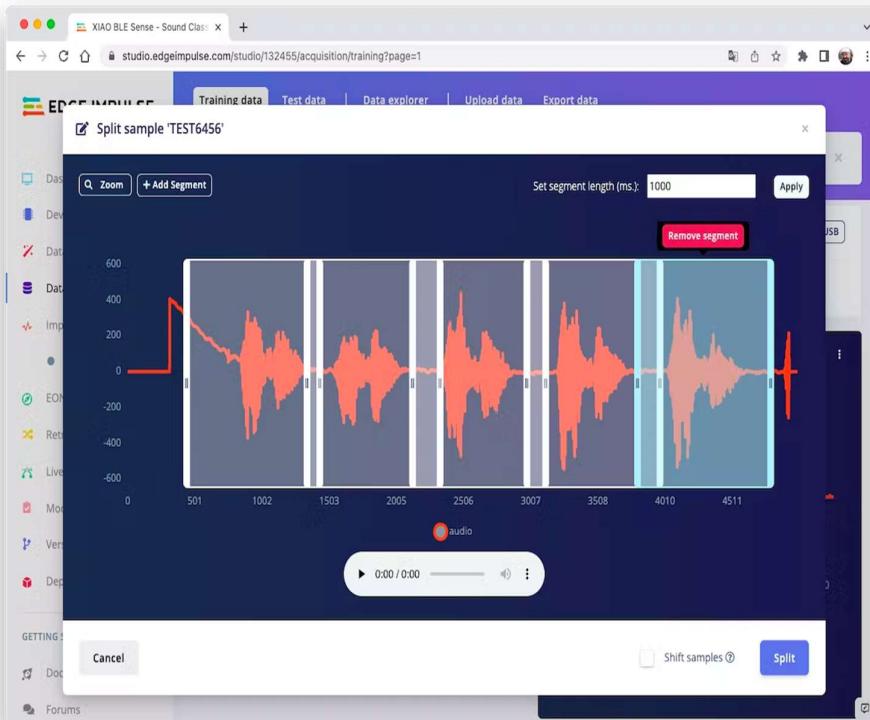
Una volta creato il progetto, si va alla sezione **Data Acquisition** e si seleziona il tool **upload Existing Data**. Si scelgono i file da caricare, ad esempio, ho iniziato a caricare i campioni registrati con XIAO nRF52840 Sense:



I campioni appariranno ora nella sezione **Data acquisition**.



Cliccare sui tre punti dopo il nome del campione e selezionare `split sample`. Una volta all'interno del tool, si dividono i dati in record da 1 secondo (cercare di evitare le parti di inizio e fine):



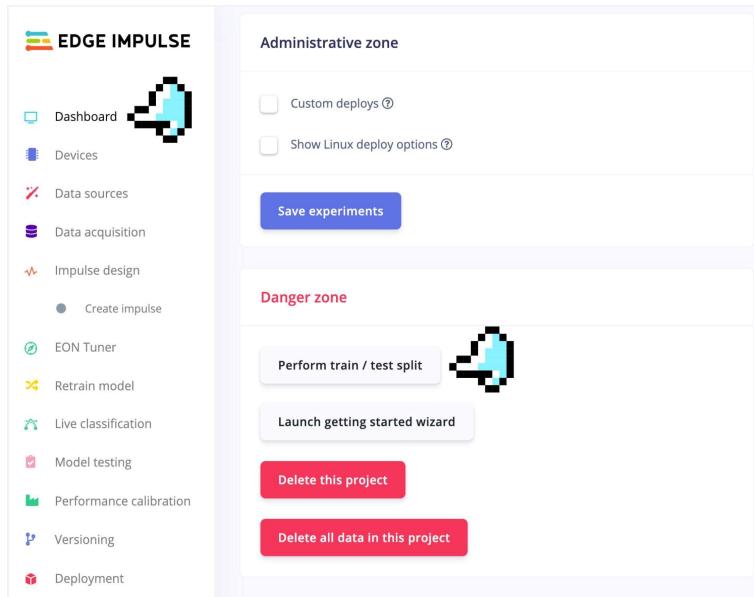
Questa procedura deve essere ripetuta per tutti i campioni. Dopodiché, caricare altri campioni di classe (IESTI e SILENCE) acquisiti con XIAO e il PC o lo smartphone.

Nota: Per file audio più lunghi (minuti), prima si dividono in segmenti da 10 secondi e poi usa di nuovo lo strumento per ottenere le divisioni finali da 1 secondo.

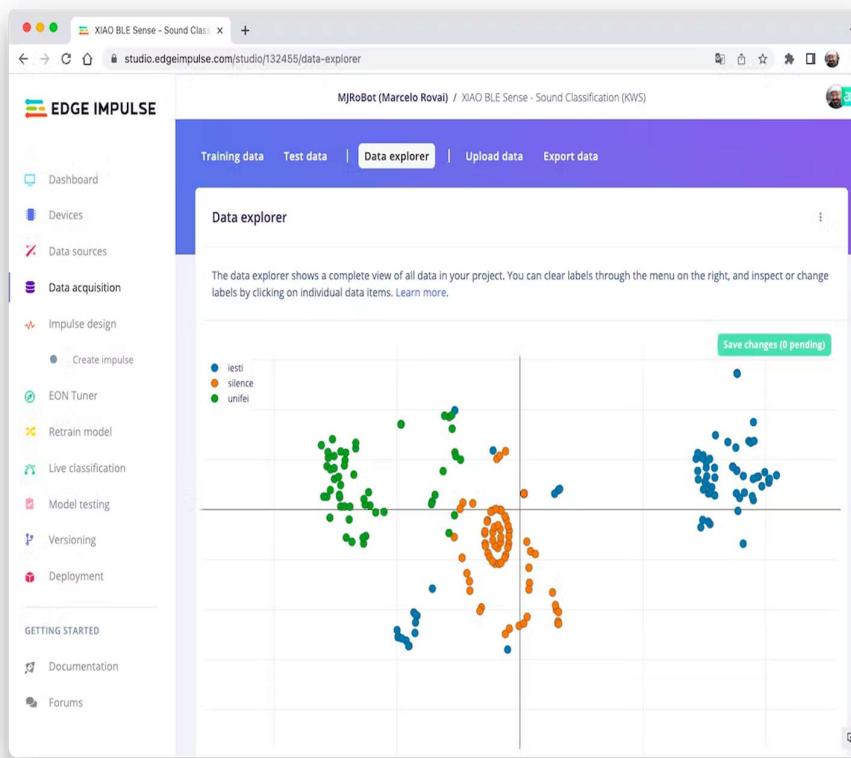
Alla fine, il set di dati ha circa 70 campioni da 1 secondo per ogni classe:

The screenshot shows the Edge Impulse Studio interface. On the left, there's a sidebar with various options like Dashboard, Devices, Data sources, Data acquisition, Impulse design, Create impulse, EON Tuner, Retrain model, Live classification, Model testing, Versioning, Deployment, Documentation, and Forums. The main area has tabs for Training data, Test data, Data explorer, Upload data, and Export data. A prominent message says "Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options". Below this, there's a "DATA COLLECT..." section showing "3m 49s" and a progress bar at "100%...". To the right, there's a "Record new data" section with a "Connect using WebUSB" button and a note that "No devices connected to the remote management API". The central part displays a table titled "Collected data" with columns for SAMPLE NAME, LABEL, ADDED, and LENGTH. It lists several entries, all labeled "iesti" and added today, each lasting 1s. To the right of the table is a "RAW DATA" section for "TEST5990.s4" showing a waveform plot with axes from -10000 to 10000 and a time axis from 0 to 936. A red circle highlights the "audio" label below the plot. At the bottom, there's a playback control bar showing "0:00 / 0:00".

Ora, si deve dividere quel dataset in Train/Test. Lo si può fare manualmente (utilizzando il menu **tre punti**, spostando i campioni singolarmente) o utilizzando **Perform Train / Test Split** SU Dashboard - **Danger Zone**.

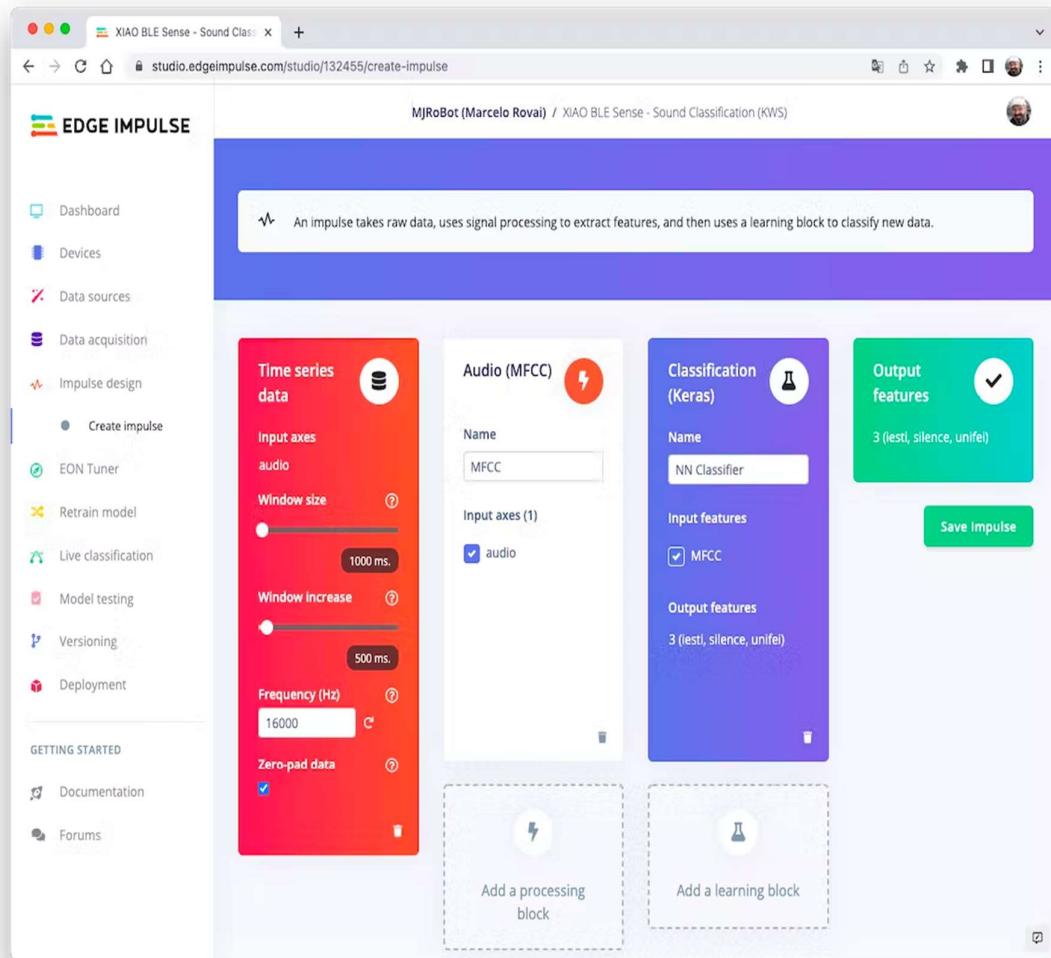


Possiamo facoltativamente controllare tutti i dataset utilizzando la scheda **Data Explorer**. I dati sembrano separati, il che significa che il modello di classificazione dovrebbe funzionare:



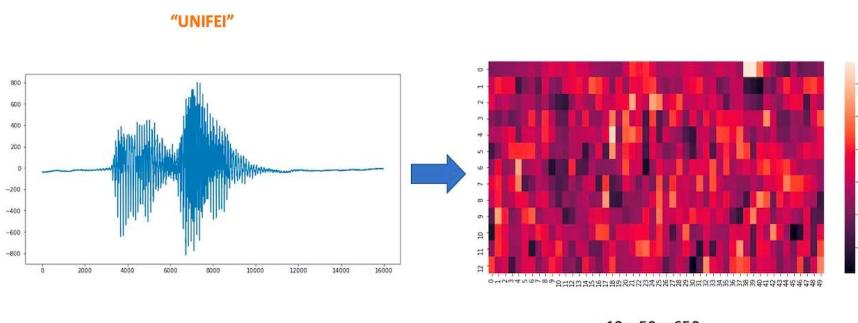
#### 4.3.3.7 Creazione di Impulse (Pre-Process / Definizione del Modello)

Un impulso prende dati grezzi, usa l'elaborazione del segnale per estrarre le caratteristiche e poi usa un blocco di apprendimento per classificare nuovi dati.



Per prima cosa, prenderemo i dati con una finestra di 1 secondo, aumentando i dati, facendo scorrere quella finestra ogni 500 ms. Notare che l'opzione zero-point pad è settata. È importante riempire con zeri i campioni più piccoli di 1 secondo in alcuni casi, è stata ridotta la finestra di 1000 ms sullo strumento di divisione per evitare rumori e picchi.

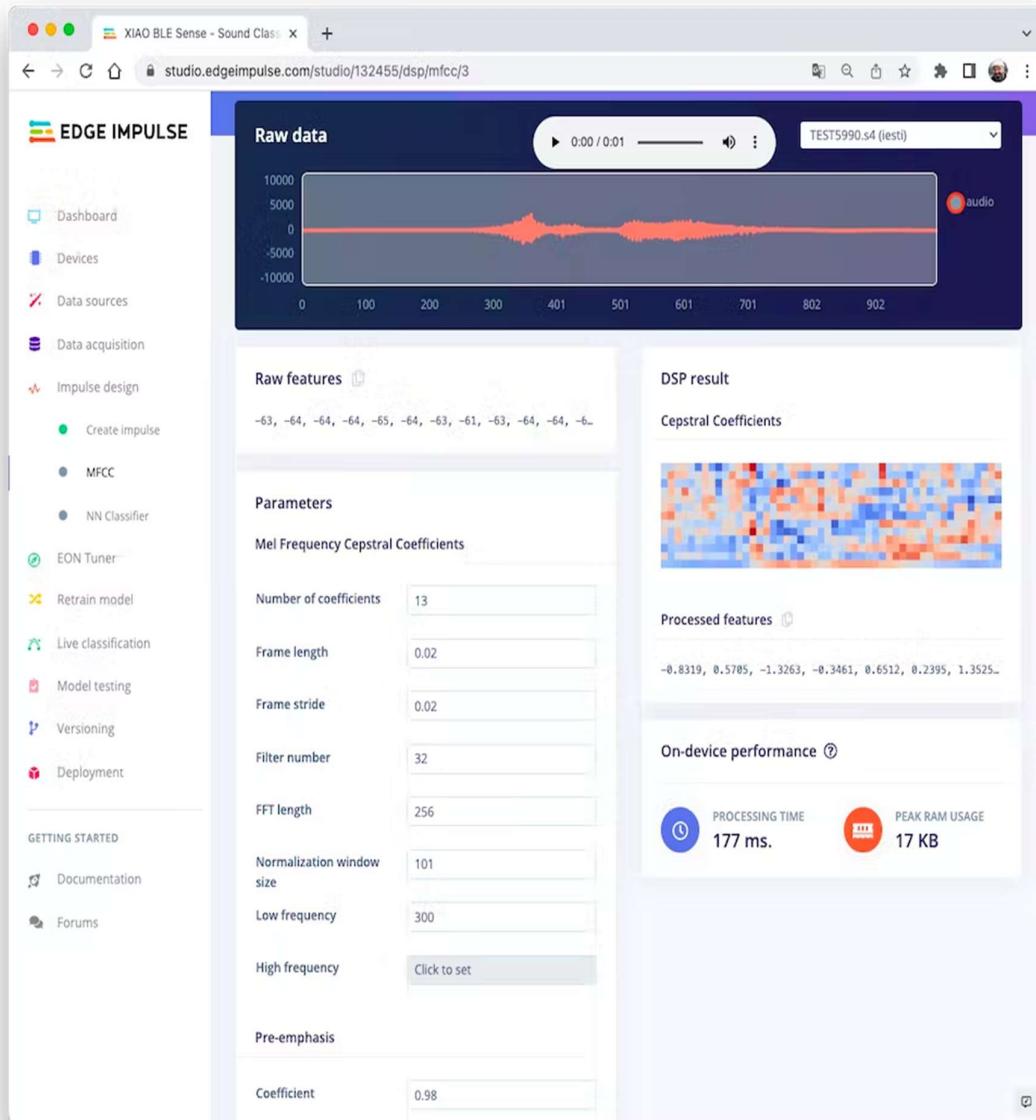
Ogni campione audio di 1 secondo dovrebbe essere pre-elaborato e convertito in un'immagine (ad esempio,  $13 \times 50 \times 1$ ). Utilizzeremo **Audio (MFCC)**, che estrae le caratteristiche dai segnali audio utilizzando **Mel Frequency Cepstral Coefficients**, che sono adatti alla voce umana, che è il nostro caso qui.



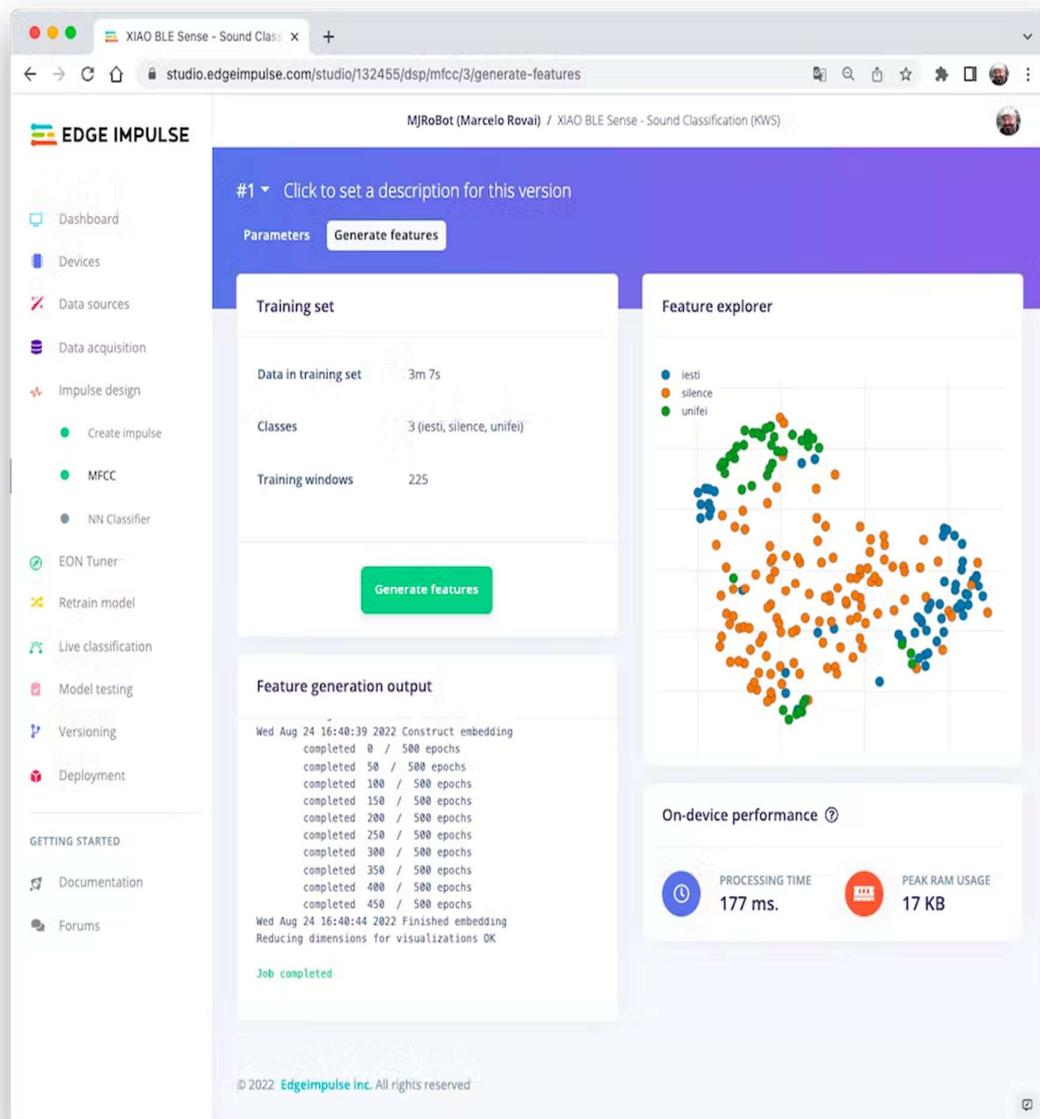
Successivamente, selezioniamo il blocco **classification** per costruire il nostro modello da zero utilizzando una Convolution Neural Network (CNN).

#### **4.3.3.8 Pre-elaborazione (MFCC)**

Il passo successivo è creare le immagini da addestrare nella fase successiva:



Manterremo i valori di default dei parametri. Non utilizziamo molta memoria per pre-elaborare i dati (solo 17 KB), ma il tempo di elaborazione è relativamente alto (177 ms per una CPU Cortex-M4 come la nostra XIAO). Salvataggio dei parametri e generazione delle feature:

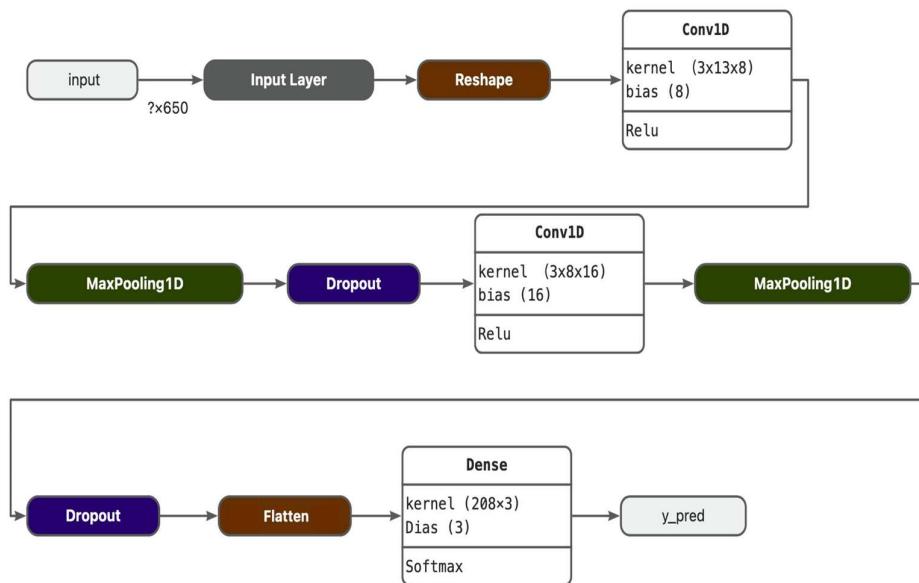


### Andiamo sotto il cofano

Per comprendere meglio come viene pre-elaborato il suono grezzo, leggere il capitolo *Feature Engineering for Audio Classification*. Si può sperimentare con la generazione delle feature MFCC scaricando questo [notebook](#) da GitHub o [aprendendo in Colab](#).

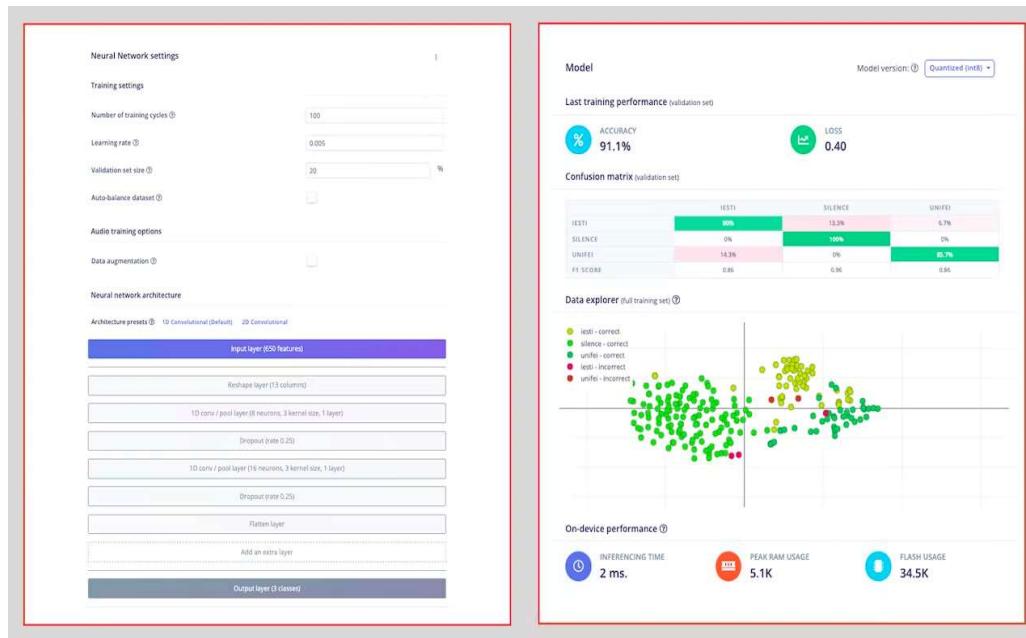
#### 4.3.3.9 Progettazione e Addestramento del Modello

Useremo un semplice modello di rete neurale convoluzionale (CNN), testato con convoluzioni 1D e 2D. L'architettura di base ha due blocchi di Convolution + MaxPooling (rispettivamente [8] e [16] filtri, e un Dropout di [0.25] per 1D e [0.5] per il 2D. Per l'ultimo layer, dopo Flattening, abbiamo [3] neuroni, uno per ogni classe:

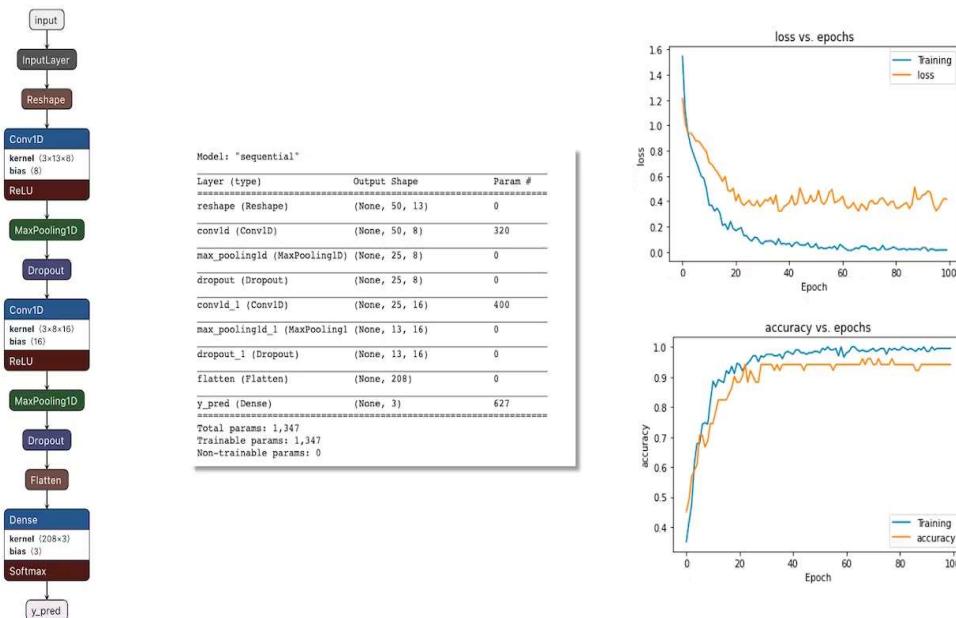


Come iperparametri, avremo un **Learning Rate** di [0.005] e un modello addestrato da [100] epocha. Includeremo anche un metodo di data augmentation basato su **SpecAugment**. Abbiamo addestrato i modelli 1D e 2D con gli stessi iperparametri. L'architettura 1D ha avuto un risultato complessivo migliore (accuratezza del 91.1%) rispetto all'88% della 2D, quindi useremo la 1D.

L'utilizzo di convoluzioni 1D è più efficiente perché richiede meno parametri rispetto alle convoluzioni 2D, rendendole più adatte ad ambienti con risorse limitate.



Per capire cosa sta succedendo “sotto il cofano”, si può scaricare il set di dati pre-elaborato ([dati di training MFCC](#)) dalla scheda **Dashboard** ed eseguire questo [Jupyter Notebook](#), giocando con il codice o [aprendendo in Colab](#). Si deve adattare il notebook ai propri dati e al proprio modello. Ad esempio, si può analizzare l'accuratezza per ogni epoca:



### 4.3.3.10 Test

Testando il modello con i dati messi da parte prima dell'addestramento (Test Data), abbiamo ottenuto un'accuratezza del 75%. Sulla base della piccola quantità di dati utilizzata, va bene, ma consiglio vivamente di aumentare il numero di campioni.

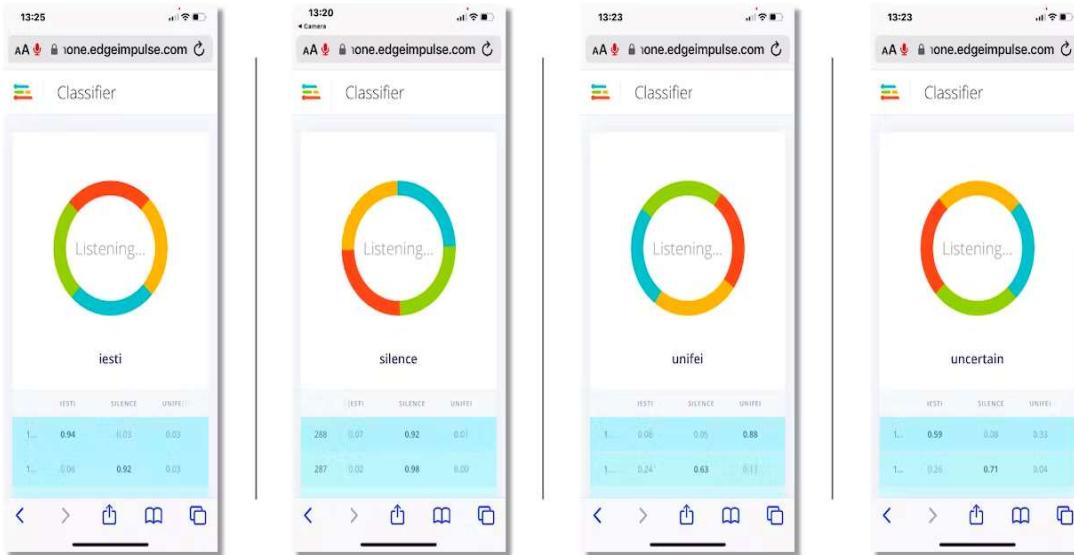
The screenshot shows the Edge Impulse web interface with the following sections:

- Dashboard**: Shows the project name "XIAO BLE Sense - Sound Classification".
- Data sources**: Shows "Data acquisition" and "Impulse design" sections.
- Test data**: A table showing test samples with columns: SAMPLE NAME, EXPECTED OUTC..., LENGTH, ACCURACY, and RESULT. Examples include TEST5990.s2, TEST6831.s3, TEST6770.s1, TEST6786.s5, TEST6786.s1, TEST6650.s1, and several entries for "iesti\_voice\_r...".
- Model testing output**: Shows the overall accuracy as 75.00%.
- Model testing results**: A table showing the distribution of results for categories: IESTI, SILENCE, UNIFEI, and UNCERTAIN. The table shows values like 60% for IESTI, 0% for SILENCE, 6.7% for UNIFEI, and 33.3% for UNCERTAIN.
- Feature explorer**: A scatter plot showing features for different categories: iesti - correct (green dots), silence - correct (light green dots), unifei - correct (dark green dots), iesti - incorrect (red dots), and unifei - incorrect (dark red dots).

Raccogliendo più dati, l'accuracy del test è aumentata di circa il 5%, passando dal 75% a circa l'81%:

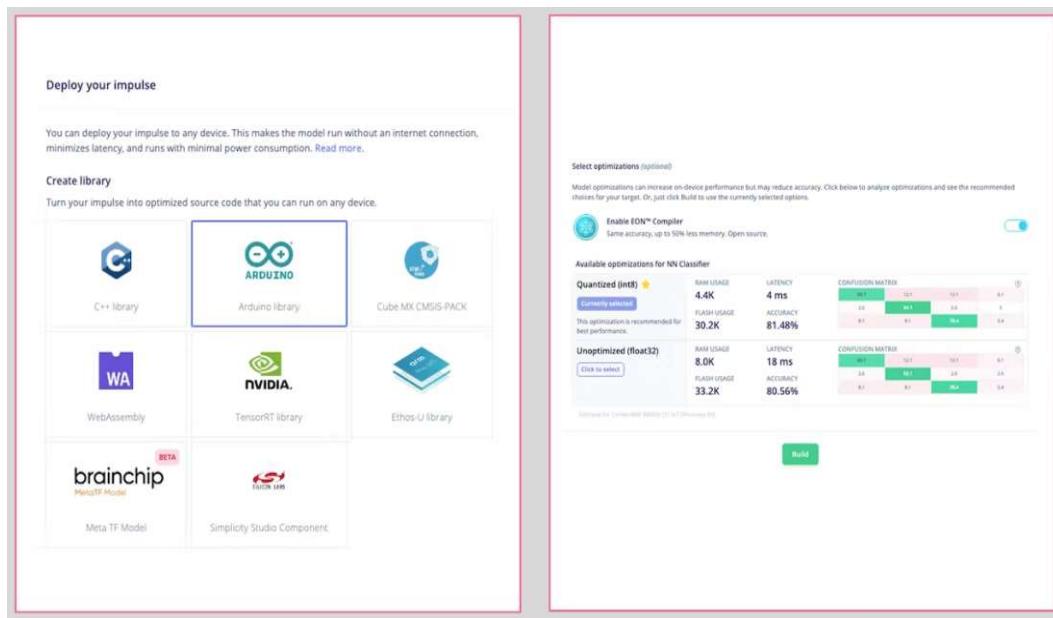


Ora possiamo procedere con il progetto, ma prima della distribuzione sul nostro dispositivo, è possibile eseguire Live Classification utilizzando uno smartphone, confermando che il modello funziona con dati live e reali:

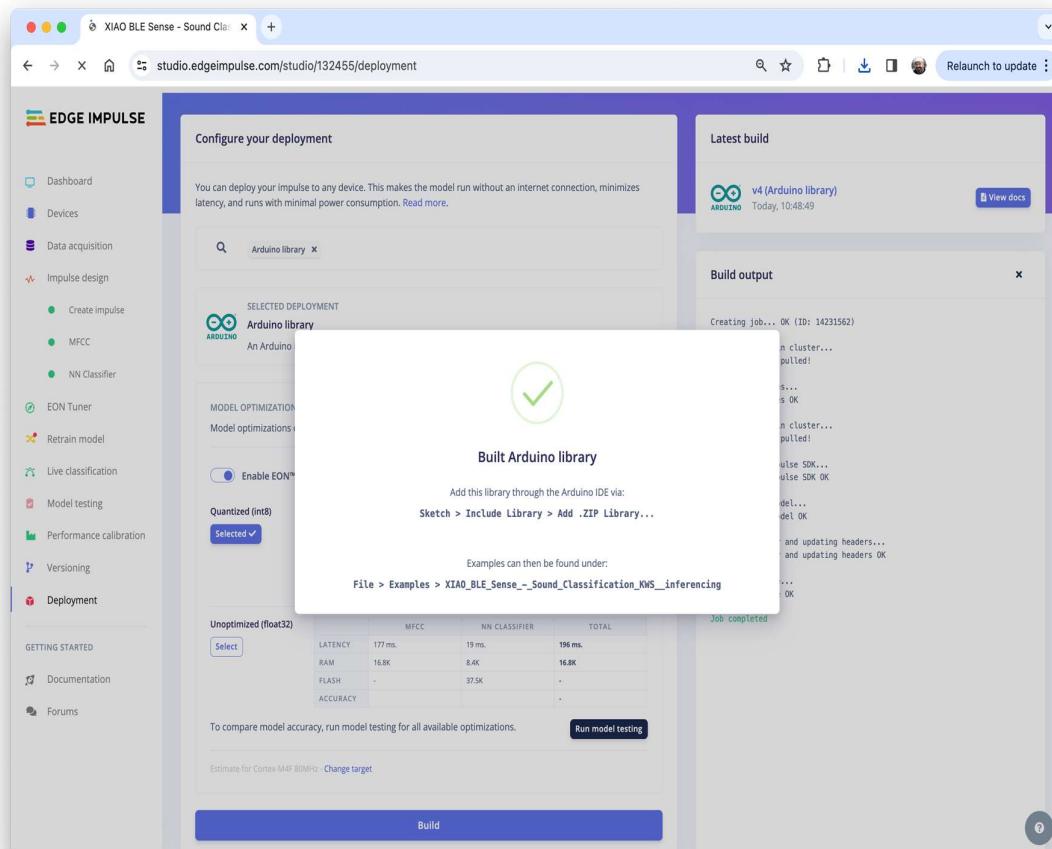


#### 4.3.3.11 Distribuzione e Inferenza

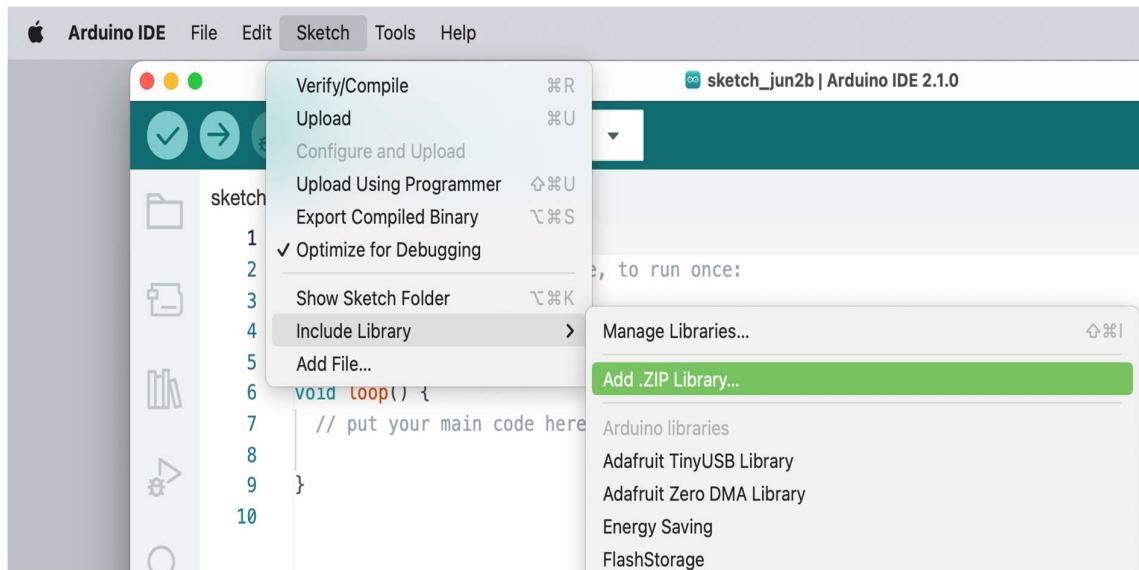
Studio impacchetterà tutte le librerie necessarie, le funzioni di pre-elaborazione e i modelli addestrati, scaricandoli sul computer. Si deve selezionare l'opzione Arduino Library e in basso, scegliere `Quantized (Int8)` e [Build].



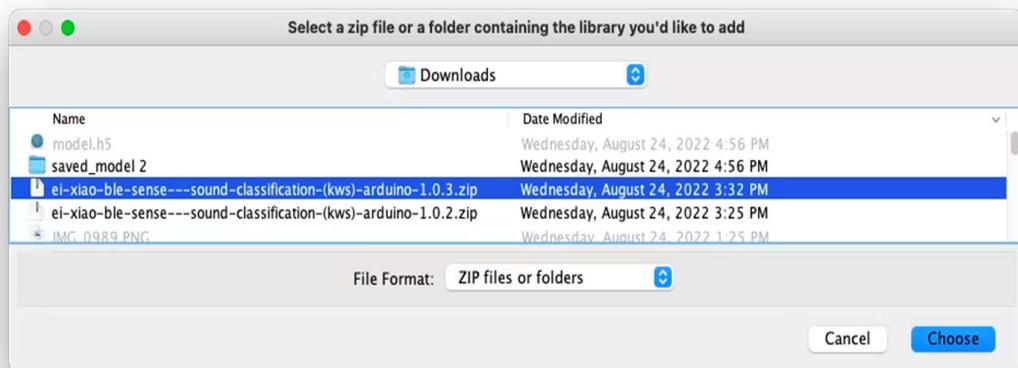
Un file Zip verrà creato e scaricato sul computer:



Sull'Arduino IDE, si va alla scheda **sketch** e si seleziona l'opzione **Add .ZIP Library**.

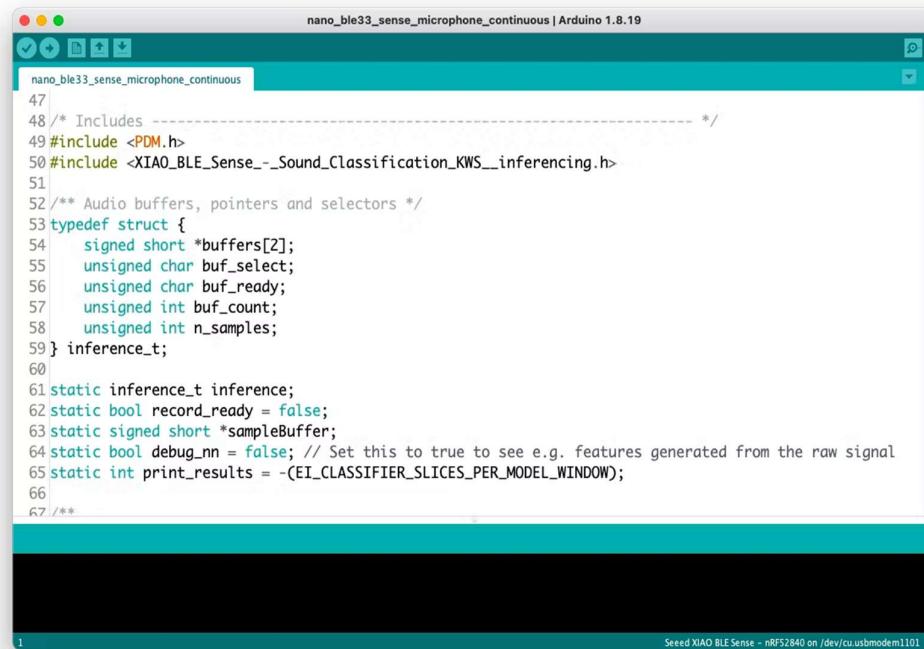


E si sceglie il file .zip scaricato dallo Studio:



Ora è il momento di un vero test. Faremo inferenze completamente scollegate da Studio. Modifichiamo uno degli esempi di codice creati quando si distribuisce la libreria Arduino.

Nell'IDE Arduino, si va alla scheda **File/Examples** e si cerca il progetto, e negli esempi, si seleziona nano\_ble33\_sense\_microphone\_continuous:



```
nano_ble33_sense_microphone_continuous | Arduino 1.8.19

nano_ble33_sense_microphone_continuous
47
48 /* Includes -----
49 #include <PDM.h>
50 #include <XIAO_BLE_Sense_Sound_Classification_KWS_inferencing.h>
51
52 /** Audio buffers, pointers and selectors */
53 typedef struct {
54     signed short *buffers[2];
55     unsigned char buf_select;
56     unsigned char buf_ready;
57     unsigned int buf_count;
58     unsigned int n_samples;
59 } inference_t;
60
61 static inference_t inference;
62 static bool record_ready = false;
63 static signed short *sampleBuffer;
64 static bool debug_nn = false; // Set this to true to see e.g. features generated from the raw signal
65 static int print_results = -(EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW);
66
67 /**
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
50
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
52
530
531
532
533
534
535
536
537
538
539
53
540
541
542
543
544
545
546
547
548
549
54
550
551
552
553
554
555
556
557
558
559
55
560
561
562
563
564
565
566
567
568
569
56
570
571
572
573
574
575
576
577
578
579
57
580
581
582
583
584
585
586
587
588
589
58
590
591
592
593
594
595
596
597
598
599
59
5
600
601
602
603
604
605
606
607
608
609
60
610
611
612
613
614
615
616
617
618
619
61
620
621
622
623
624
625
626
627
628
629
62
630
631
632
633
634
635
636
637
638
639
63
640
641
642
643
644
645
646
647
648
649
64
650
651
652
653
654
655
656
657
658
659
65
660
661
662
663
664
665
666
667
668
669
66
670
671
672
673
674
675
676
677
678
679
67
680
681
682
683
684
685
686
687
688
689
68
690
691
692
693
694
695
696
697
698
699
69
6
700
701
702
703
704
705
706
707
708
709
70
710
711
712
713
714
715
716
717
718
719
71
720
721
722
723
724
725
726
727
728
729
72
730
731
732
733
734
735
736
737
738
739
73
740
741
742
743
744
745
746
747
748
749
74
750
751
752
753
754
755
756
757
758
759
75
760
761
762
763
764
765
766
767
768
769
76
770
771
772
773
774
775
776
777
778
779
77
780
781
782
783
784
785
786
787
788
789
78
790
791
792
793
794
795
796
797
798
799
79
7
800
801
802
803
804
805
806
807
808
809
80
810
811
812
813
814
815
816
817
818
819
81
820
821
822
823
824
825
826
827
828
829
82
830
831
832
833
834
835
836
837
838
839
83
840
841
842
843
844
845
846
847
848
849
84
850
851
852
853
854
855
856
857
858
859
85
860
861
862
863
864
865
866
867
868
869
86
870
871
872
873
874
875
876
877
878
879
87
880
881
882
883
884
885
886
887
888
889
88
890
891
892
893
894
895
896
897
898
899
89
8
900
901
902
903
904
905
906
907
908
909
90
910
911
912
913
914
915
916
917
918
919
91
920
921
922
923
924
925
926
927
928
929
92
930
931
932
933
934
935
936
937
938
939
93
940
941
942
943
944
945
946
947
948
949
94
950
951
952
953
954
955
956
957
958
959
95
960
961
962
963
964
965
966
967
968
969
96
970
971
972
973
974
975
976
977
978
979
97
980
981
982
983
984
985
986
987
988
989
98
990
991
992
993
994
995
996
997
998
999
99
9
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
100
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
101
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
102
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
103
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
104
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
105
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
106
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
107
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
108
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
109
10
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
110
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
111
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
112
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
113
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
114
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
115
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
116
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
117
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
118
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
119
11
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
120
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
121
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
122
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
123
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
124
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
125
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
126
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
127
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
128
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
129
12
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
130
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
131
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
132
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
133
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
134
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
135
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
136
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
137
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
138
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
139
13
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
140
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
141
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
142
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
143
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
144
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
145
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
146
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
147
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
148
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
149
14
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
150
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
151
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
152
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
153
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
154
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
155
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
156
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
157
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
158
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
159
15
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
160
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
161
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
162
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
163
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
164
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
165
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
166
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
167
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
168
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
169
16
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
170
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
171
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
172
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
173
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
174
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
175
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
176
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
177
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
178
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
179
17
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
180
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
181
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
182
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
183
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
184
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
185
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
186
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
187
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
188
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
189
18
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
190
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
191
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
192
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
193
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
194
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
195
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
196
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
197
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
198
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
199
19
1
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
200
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
201
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
202
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
203
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
204
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
205
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
206
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
207
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
208
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
209
20
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
210
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
211
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
212
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
213
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
214
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
215
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
216
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
217
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
218
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
219
21
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
220
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
221
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
222
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
223
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
224
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
225
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
226
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
227
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
228
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
229
22
2300
2301
2302
2303
23
```

#### 4.3.3.12 Post-elaborazione

Ora che sappiamo che il modello funziona rilevando le nostre due parole chiave, modifichiamo il codice in modo da poter vedere il risultato con XIAO nRF52840 Sense completamente offline (scollegato dal PC e alimentato da una batteria).

L'idea è che ogni volta che viene rilevata la parola chiave UNIFEI, il LED rosso sarà ACCESO; se è IESTI, il LED verde sarà ACCESO e se è SILENCE (nessuna parola chiave), entrambi i LED saranno SPENTI.

Se si ha XIAO nRF52840 Sense installato sulla scheda di espansione, possiamo visualizzare l'etichetta della classe e la sua probabilità. Altrimenti, si usano solo i LED.

Andiamo per Parti: Installazione e test del display SSD Nell'IDE Arduino, Installare la libreria [u8g2](#) ed eseguire il codice seguente per il test:

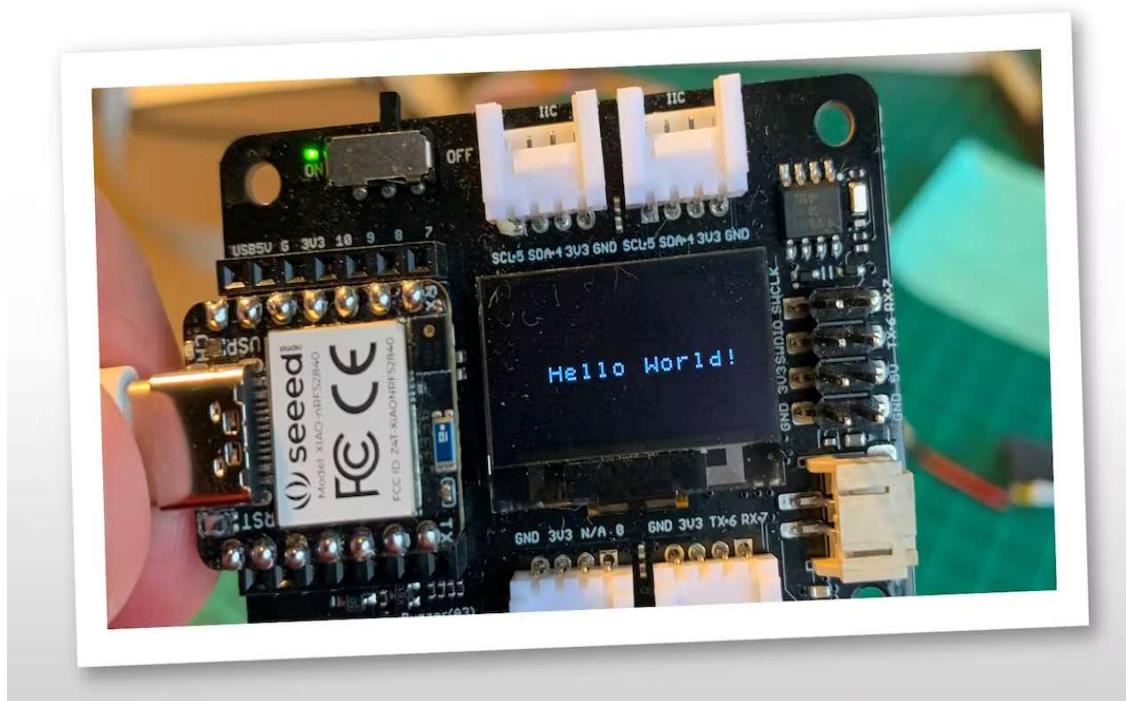
```
#include <Arduino.h>
#include <U8x8lib.h>
#include <Wire.h>

U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(PIN_WIRE_SCL, PIN_WIRE_SDA, U8X8_PIN_NONE);

void setup(void) {
    u8x8.begin();
    u8x8.setFlipMode(0); // set number from 1 to 3, the screen word should rotate 180
}

void loop(void) {
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 0);
    u8x8.print("Hello World!");
}
```

Si dovrebbe vedere la scritta “Hello World” visualizzata sull’SSD:



Ora, creiamo alcune funzioni che, a seconda dei valori di pred\_index e pred\_value, attiveranno il LED appropriato e visualizzeranno la classe e la probabilità. Il codice seguente simulerà alcuni risultati di inferenza e li presenterà su display e LED:

```
/* Includes ----- */
#include <Arduino.h>
#include <U8x8lib.h>
```

```
#include <Wire.h>

#define NUMBER_CLASSES 3

/** OLED */
U8X8_SSD1306_128X64_NONAME_HW_I2C oled(PIN_WIRE_SCL, PIN_WIRE_SDA, U8X8_PIN_NONE);

int pred_index = 0;
float pred_value = 0;
String lbl = " ";

void setup() {
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(LED_B, OUTPUT);

    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_G, HIGH);
    digitalWrite(LED_B, HIGH);

    oled.begin();
    oled.setFlipMode(2);
    oled.setFont(u8x8_font_chroma48medium8_r);
    oled.setCursor(0, 0);
    oled.print(" XIAO Sense KWS");
}

/***
* @brief      turn_off_leds function - turn-off all RGB LEDs
*/
void turn_off_leds(){
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_G, HIGH);
    digitalWrite(LED_B, HIGH);
}

/***
* @brief      Show Inference Results on OLED Display
*/
void display_oled(int pred_index, float pred_value){
    switch (pred_index) {
        case 0:
            turn_off_leds();
            digitalWrite(LED_G, LOW);
            lbl = "IESTI ";
            break;

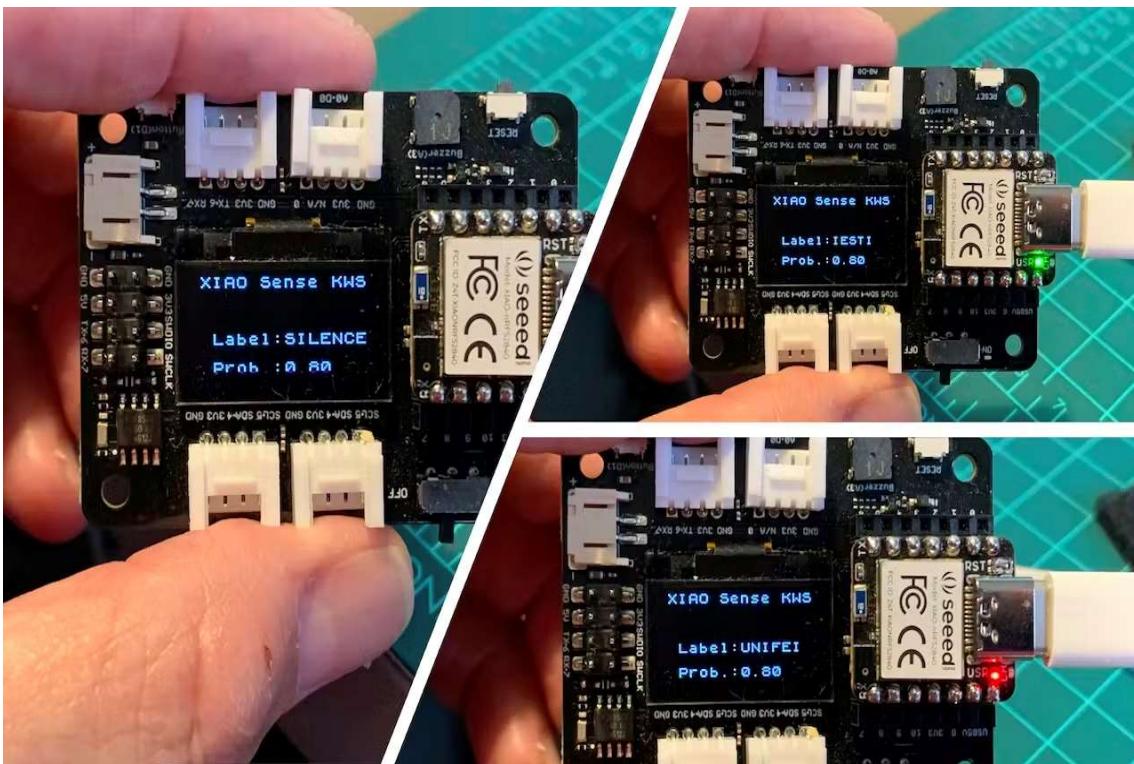
        case 1:
            turn_off_leds();
            lbl = "SILENCE";
            break;

        case 2:
            turn_off_leds();
            digitalWrite(LED_R, LOW);
            lbl = "UNIFEI ";
            break;
    }

    oled.setCursor(0, 2);
    oled.print("      ");
    oled.setCursor(2, 4);
    oled.print("Label:");
    oled.print(lbl);
    oled.setCursor(2, 6);
    oled.print("Prob.:");
    oled.print(pred_value);
}

void loop() {
    for (int i = 0; i < NUMBER_CLASSES; i++) {
        pred_index = i;
        pred_value = 0.8;
        display_oled(pred_index, pred_value);
        delay(2000);
    }
}
```

Eseguendo il codice precedente, si dovrebbe ottenere il risultato seguente:



Si deve unire il codice soprastante (Inizializzazione e funzioni) con il `nano_ble33_sense_microphone_continuous.ino` utilizzato inizialmente per testare il modello. Inoltre, si deve includere il codice sottostante su `loop()` tra le righe:

```
ei_printf(": \n");
...
#if EI_CLASSIFIER_HAS_ANOMALY == 1
```

E sostituire la funzione originale per stampare i risultati dell'inferenza sul monitor seriale:

```
int pred_index = 0; // Initialize pred_index
float pred_value = 0; // Initialize pred_value

for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
    ei_printf(" %s: %.5f\n", result.classification[ix].label, result.classification[ix].value);
    if (result.classification[ix].value > pred_value) {
        pred_index = ix;
        pred_value = result.classification[ix].value;
    }
}
display_oled(pred_index, pred_value);
```

Ecco come è il progetto finale: <https://youtu.be/1ex88hSqqyl>

Il codice completo può essere trovato su GitHub (3\_KWS): [Seeed-XIAO-BLE-Sense](#).

#### 4.3.3.13 Conclusione

Seeed XIAO nRF52840 Sense è davvero un minuscolo dispositivo gigantesco! Tuttavia, è potente, affidabile, non costoso, a basso consumo e ha sensori adatti per essere utilizzato nelle applicazioni di apprendimento automatico embedded più comuni come movimento e suono.

Anche se Edge Impulse non supporta ufficialmente XIAO nRF52840 Sense (ancora!), ci siamo resi conto che potrebbe utilizzare Studio per il training e la distribuzione.

Nel repository GitHub, si trova l'ultima versione dei codici nella cartella 3\_KWS:

[Seeed-XIAO-BLE-Sense](#)

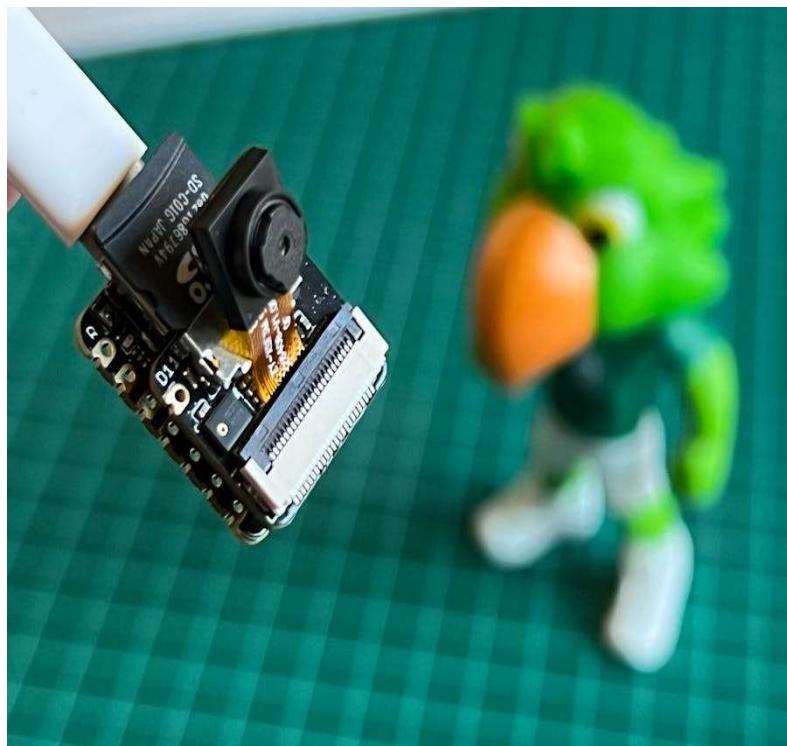
Prima di concludere, considerare che la classificazione dei suoni è più di una semplice voce. Ad esempio, si possono sviluppare progetti TinyML sul suono in diverse aree come:

- Sicurezza (rilevamento vetri rotti)

- Industria (rilevamento anomalie)
- Medicina (russamento, tosse, malattie polmonari)
- Natura (controllo alveari, suono insetti)

## 4.4 Classificazione delle Immagini

In questa sezione esploreremo le applicazioni di Computer Vision ML su XIAO ESP32S3 Sense.



### 4.4.1 Cose utilizzate in questo progetto

#### 4.4.1.1 Componenti hardware

Seeed Studio Seeed XIAO ESP32S3 Sense x 1

#### 4.4.2 App software e servizi online



- Arduino IDE



- Edge Impulse Studio

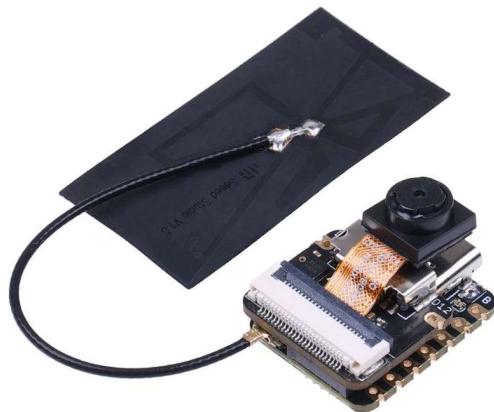
## 4.4.2 Introduzione

Sempre di più, ci troviamo di fronte a una rivoluzione dell'intelligenza artificiale (IA) in cui, come affermato da Gartner, **Edge AI** ha un potenziale di impatto molto elevato, ed è ora!



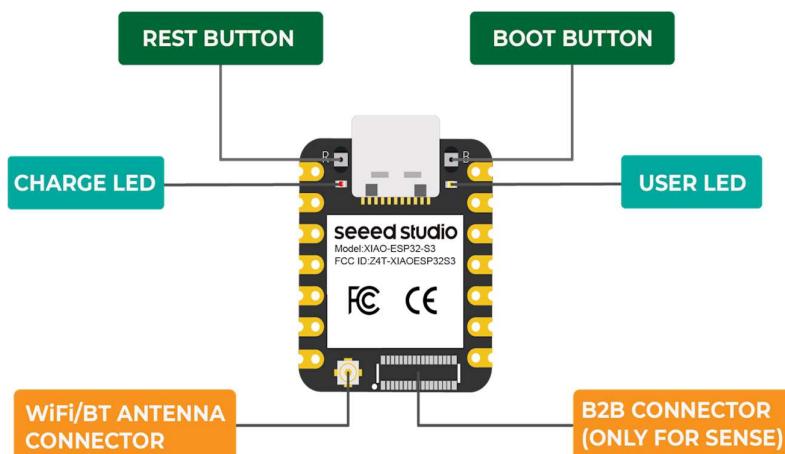
Nel “centro” delle tecnologie emergenti, il radar è l'*Edge Computer Vision*, e quando parliamo di Machine Learning (ML) applicato alla visione, la prima cosa che viene in mente è la **Classificazione delle immagini**, una specie di “Hello World” ML!

Seeed Studio ha rilasciato una nuova scheda di sviluppo conveniente, la **XIAO ESP32S3 Sense**, che integra un sensore della fotocamera, un microfono digitale e il supporto per schede SD. Combinando la potenza di elaborazione ML embedded e la capacità fotografica, questa scheda di sviluppo è un ottimo strumento per iniziare con TinyML (IA vocale e visiva).

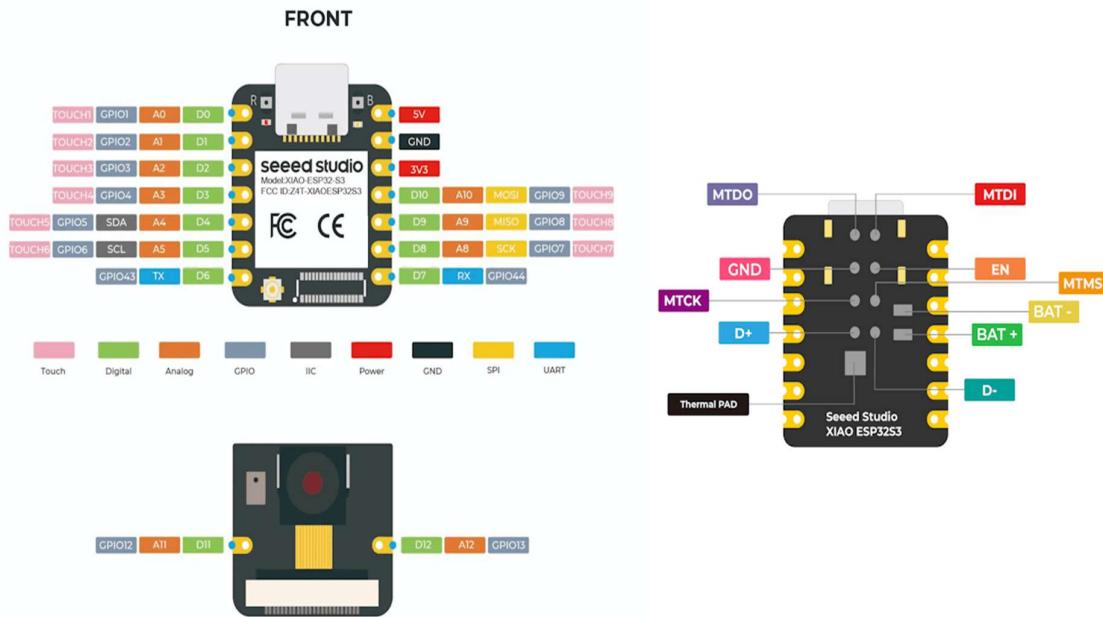


### Caratteristiche Principali di XIAO ESP32S3 Sense

- **Potente Scheda MCU:** Incorpora il chip del processore Xtensa dual-core ESP32S3 a 32 bit che funziona fino a 240 MHz, più porte di sviluppo montate, supporto per Arduino/MicroPython
- **Funzionalità Avanzate:** Sensore della fotocamera OV2640 staccabile per una risoluzione di 1600 \* 1200, compatibile con il sensore della fotocamera OV5640, integra un microfono digitale aggiuntivo
- **Progetto di Alimentazione Elaborato:** La capacità di gestione della carica della batteria al litio offre quattro modelli di consumo energetico, che consentono la modalità di “deep sleep” [sospensione profonda] con un consumo energetico basso fino a 14 $\mu$ A
- **Grande Memoria per più Possibilità:** Offre 8 MB di PSRAM e 8 MB di FLASH, supportando uno slot per schede SD di memoria FAT esterno da 32 GB
- **Prestazioni RF Eccezionali:** Supporta la comunicazione wireless duale Wi-Fi e BLE a 2,4 GHz, supporta la comunicazione remota a 100m+ se connesso all’antenna U.FL
- **Design Compatto delle Dimensioni di un Pollice:** 21 x 17,5 mm, adotta il fattore di forma classico di XIAO, adatto per progetti con spazio limitato come i dispositivi indossabili



Di seguito è riportato il pinout generale della scheda:



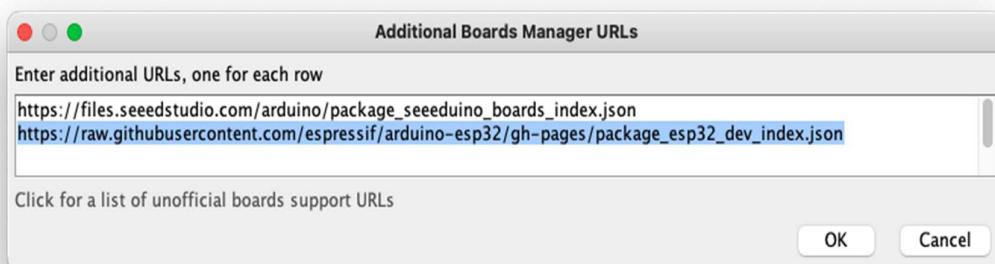
Per maggiori dettagli, fare riferimento alla pagina WiKi di Seeed Studio:  
[https://wiki.seeedstudio.com/xiao\\_esp32s3\\_getting\\_started/](https://wiki.seeedstudio.com/xiao_esp32s3_getting_started/)

#### 4.4.3 Installazione di XIAO ESP32S3 Sense su Arduino IDE

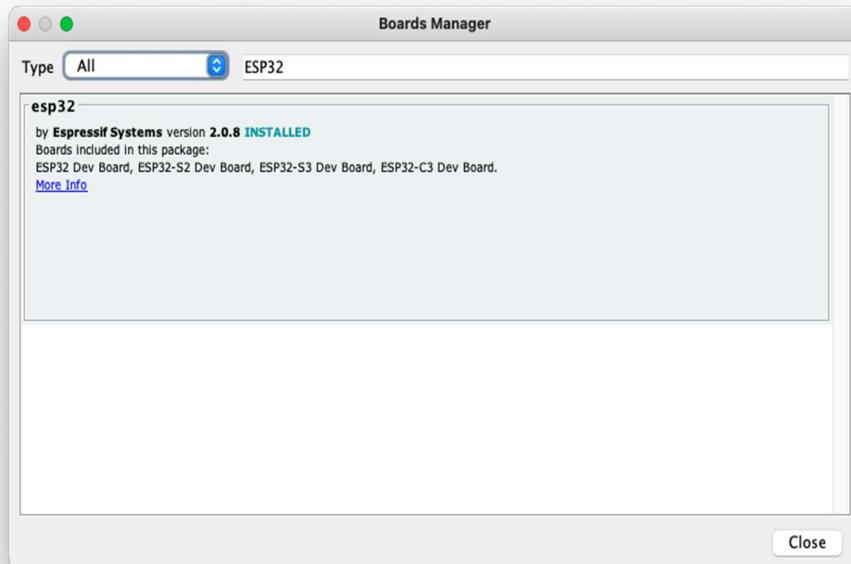
Su Arduino IDE, si va su **File > Preferences** e si inserisce l'URL:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)

nel campo ==> **Additional Boards Manager URLs**



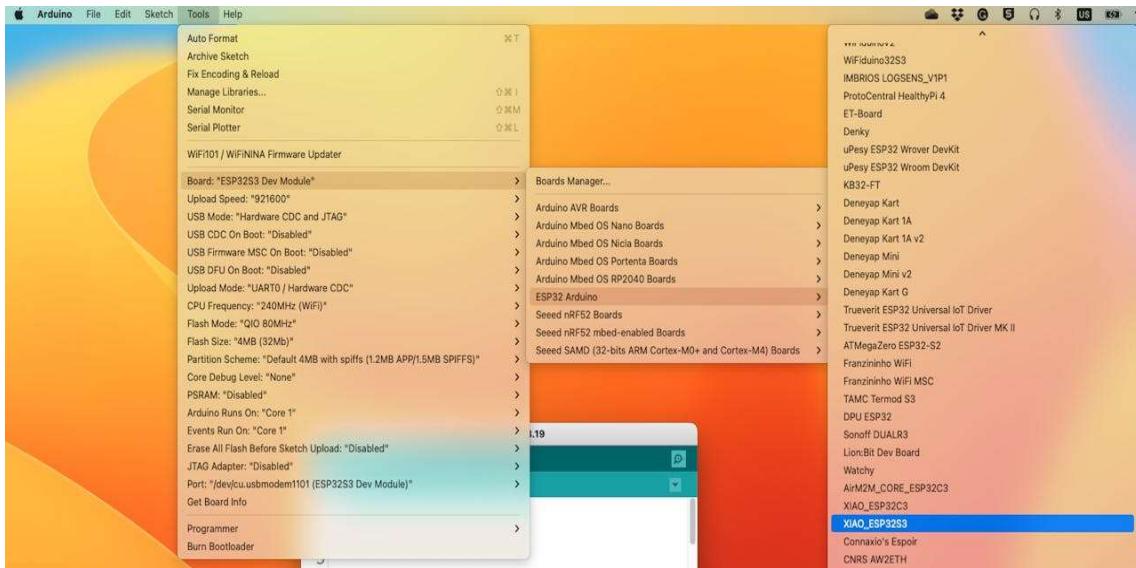
Poi si apre il gestore delle schede. Si va su **Tools > Board > Boards Manager...** e immettendo `esp32`. Selezionare e installare il pacchetto più aggiornato e stabile (evitare le versioni *alpha*):



### ⚠️ Attenzione

Le versioni Alpha (ad esempio, 3.x-alpha) non funzionano correttamente con XIAO ed Edge Impulse. Utilizzare invece l'ultima versione stabile (ad esempio, 2.0.11).

Su Tools, selezionare la Board (**XIAO ESP32S3**):



Ultimo ma non meno importante, scegliere la **Porta** a cui è collegato l'ESP32S3.

Ecco fatto! Il dispositivo dovrebbe funzionare. Facciamo qualche test.

#### 4.4.4 Test della scheda con BLINK

XIAO ESP32S3 Sense ha un LED integrato che è collegato a GPIO21. Quindi, si può eseguire lo sketch Blink così com'è (utilizzando la costante Arduino `LED_BUILTIN`) o modificando di conseguenza lo sketch Blink:

```
#define LED_BUILTIN 21

void setup() {
    pinMode(LED_BUILTIN, OUTPUT); // Set the pin as output
```

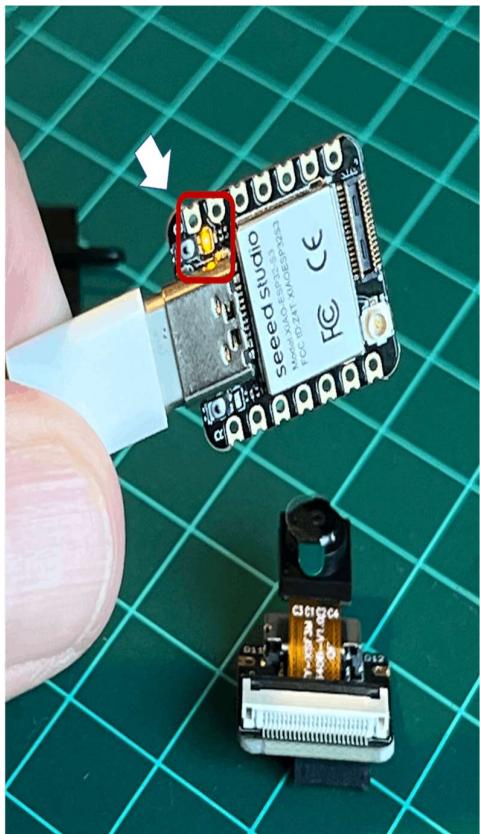
```

}

// Remember that the pin work with inverted logic
// LOW to Turn on and HIGH to turn off
void loop() {
    digitalWrite(LED_BUILTIN, LOW); //Turn on
    delay (1000); //Wait 1 sec
    digitalWrite(LED_BUILTIN, HIGH); //Turn off
    delay (1000); //Wait 1 sec
}

```

Notare che i pin funzionano con logica invertita: BASSO per accendere e ALTO per spegnere.



#### 4.4.5 Collegamento del Modulo Sense (Scheda di Espansione)

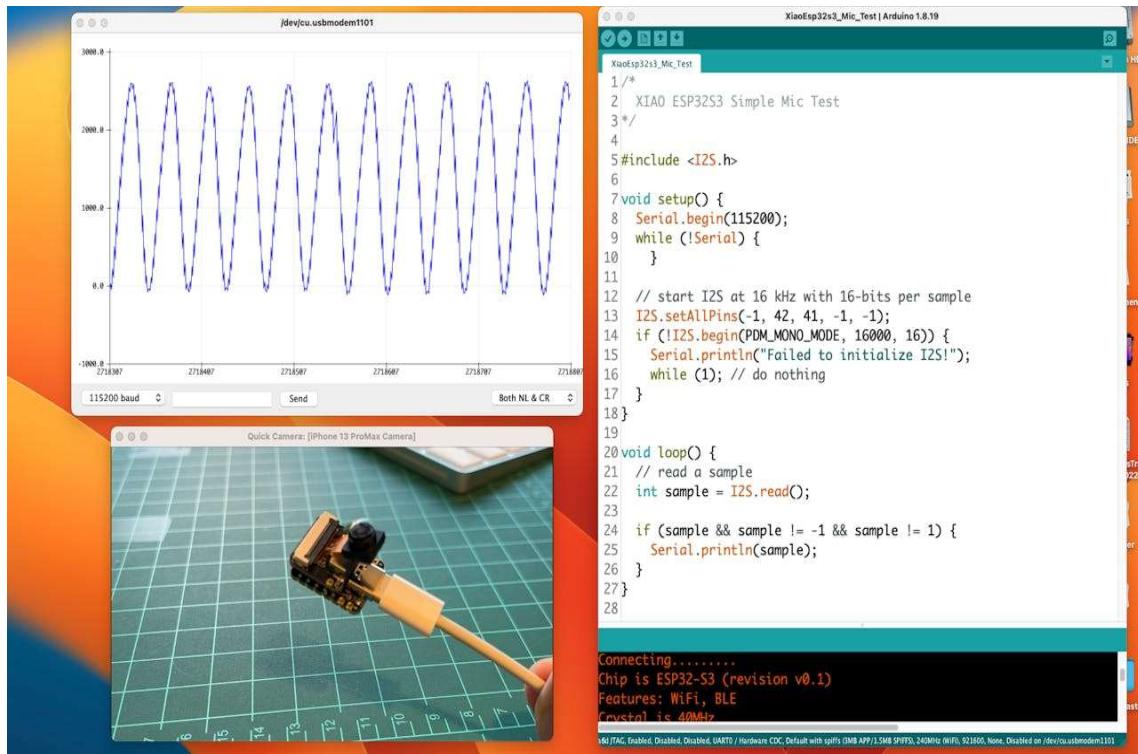
Quando viene acquistata, la scheda di espansione è separata da quella principale, ma installarla è molto semplice. Si deve allineare il connettore sulla scheda di espansione col connettore B2B sullo XIAO ESP32S3, premerlo con forza e quando si sente un “clic”, l’installazione è completa.



Come commentato nell'introduzione, la scheda di espansione, o la parte "sense" del dispositivo, ha una fotocamera OV2640 da 1600x1200, uno slot per schede SD e un microfono digitale.

#### 4.4.6 Test del Microfono

Cominciamo con il rilevamento del suono. Si va nei [progetti GitHub](#) e si scarica lo sketch: [XIAOEsp2s3\\_Mic\\_Test](#) poi lo si esegue sull'IDE Arduino:



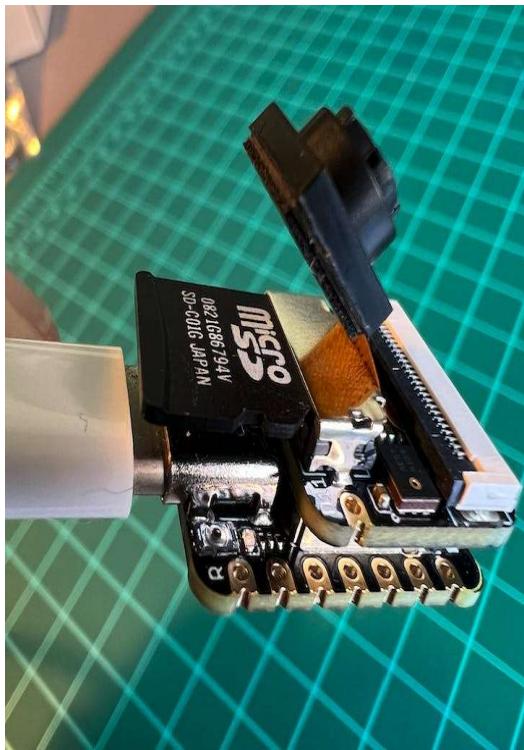
Quando si produce un suono, è possibile verificarlo sul Serial Plotter.

#### Salvare il suono registrato (file audio .wav) su una scheda microSD.

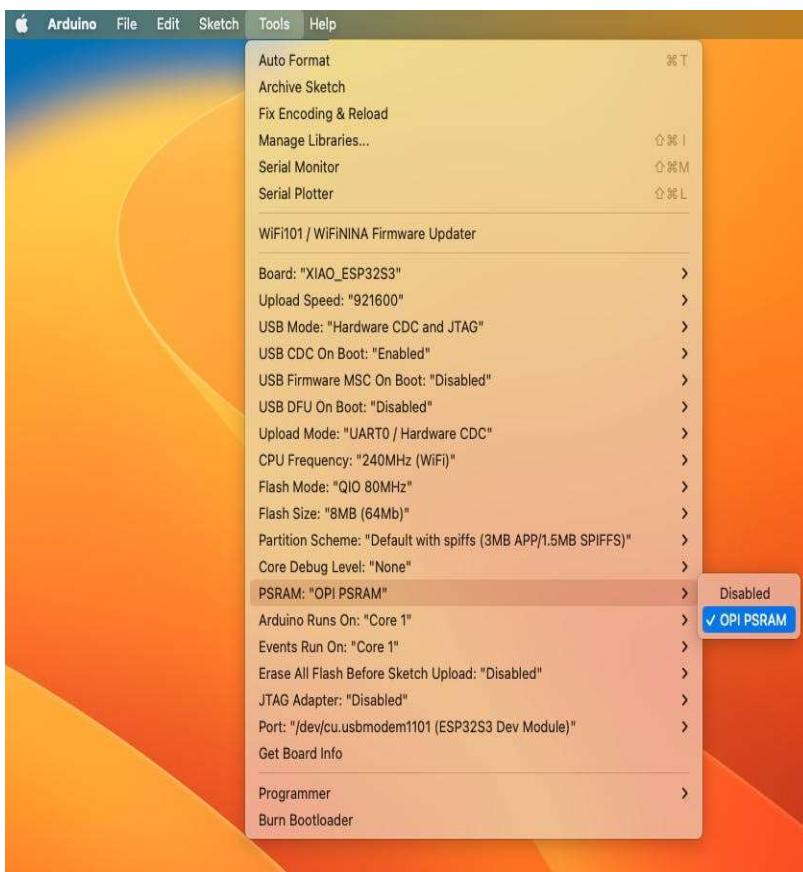
Ora, il lettore di schede SD integrato può salvare file audio .wav. Per questo, dobbiamo abilitare la XIAO PSRAM.

ESP32-S3 ha solo poche centinaia di kilobyte di RAM interna sul chip MCU. Potrebbe essere insufficiente per alcuni scopi, quindi l'ESP32-S3 può utilizzare fino a 16 MB di PSRAM esterna (Psuedostatic RAM) collegata al chip flash SPI. La memoria esterna è incorporata nella mappa di memoria e, con alcune restrizioni, è utilizzabile allo stesso modo della RAM dati interna.

Per iniziare, inserire la scheda SD su XIAO come mostrato nella foto qui sotto (la scheda SD deve essere formattata in **FAT32**).

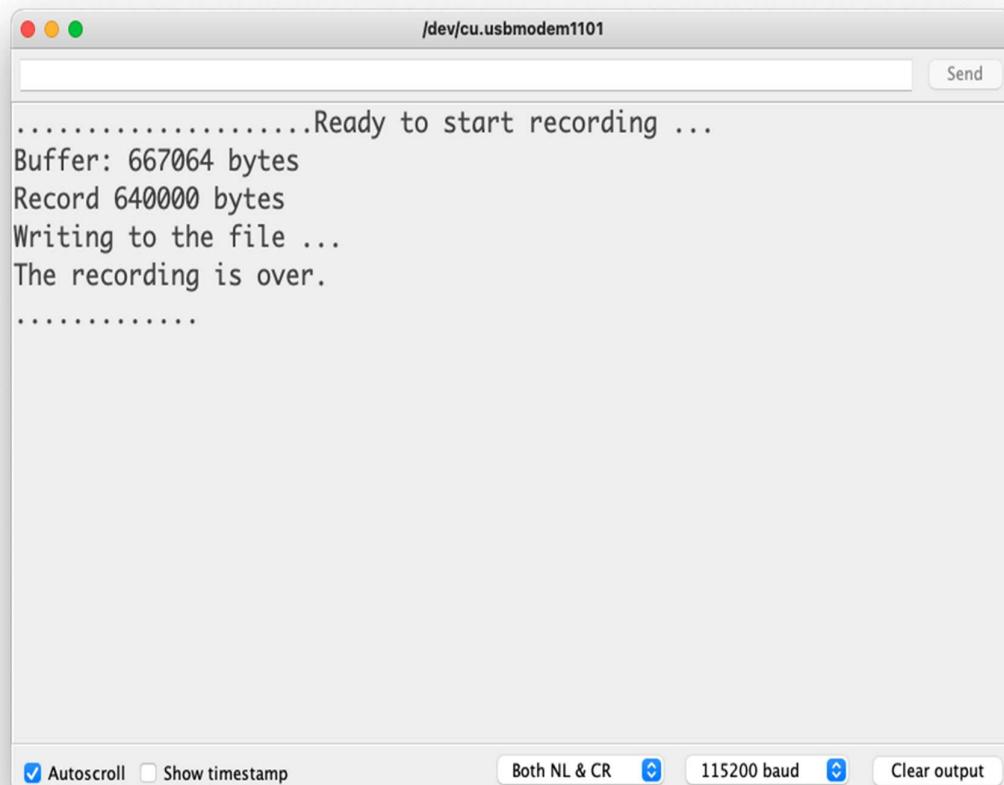


- Scaricare lo sketch [Wav\\_Record](#), che si può trovare su GitHub.
- Per eseguire il codice (Wav Record), è necessario utilizzare la funzione PSRAM del chip ESP-32, quindi la si deve attivare prima di caricare.: Tools>PSRAM: “OPI PSRAM”>OPI PSRAM



- Eseguire il codice [Wav\\_Record.ino](#)

- Questo programma viene eseguito una sola volta dopo che l'utente **accende il monitor seriale**, registrando per 20 secondi e salvando il file di registrazione su una scheda microSD come “arduino\_rec.wav”.
- Quando “.” viene emesso ogni 1 secondo nel monitor seriale, l'esecuzione del programma è terminata e si può riprodurre il file audio registrato con l'aiuto di un lettore di schede.



La qualità del suono è eccellente!

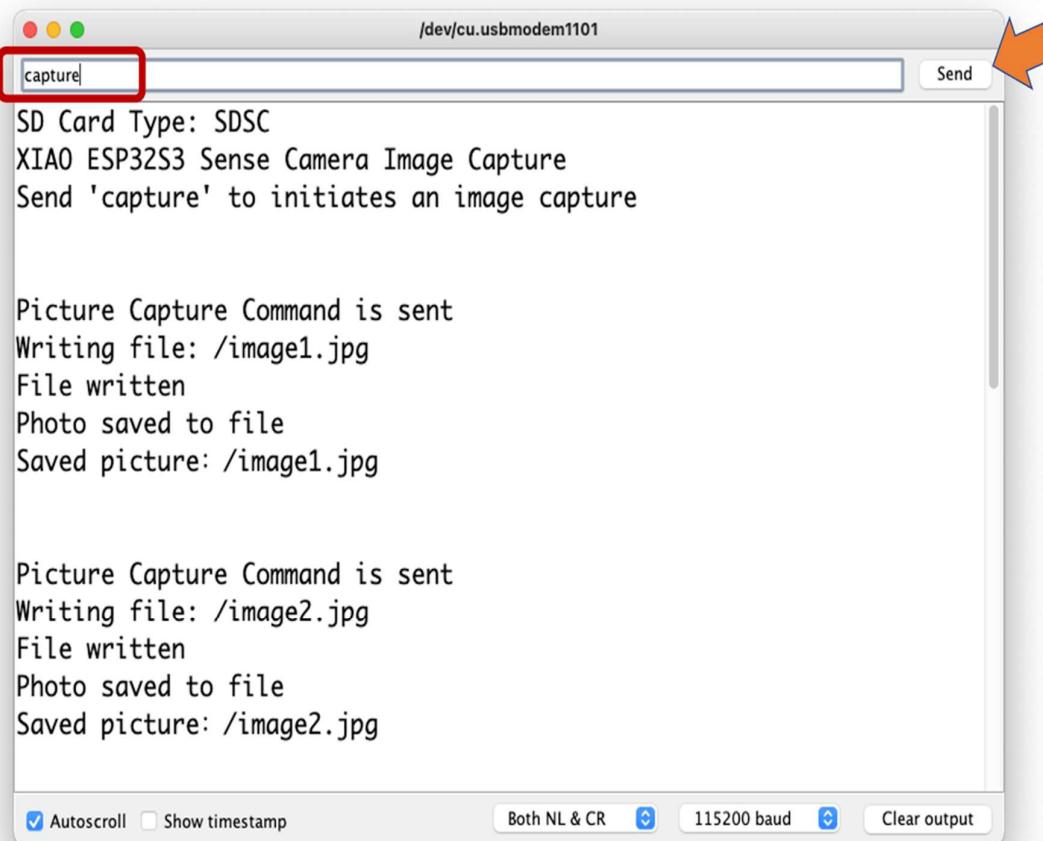
La spiegazione di come funziona il codice va oltre lo scopo di questo tutorial, ma c'è un'eccellente descrizione sulla pagina [wiki](#).

#### 4.4.7 Test della Fotocamera

Per testare la fotocamera, si deve scaricare la cartella [take\\_photos\\_command](#) da GitHub. La cartella contiene lo sketch (.ino) e due file .h con i dettagli della fotocamera.

- Eseguire il codice: `take_photos_command.ino`. Aprire il Serial Monitor e inviare il comando `capture` per catturare e salvare l'immagine sulla scheda SD:

Verificare che `[Both NL & CR]` sia selezionato sul monitor seriale.



The screenshot shows a terminal window titled '/dev/cu.usbmodem1101'. In the input field, the command 'capture' is typed and highlighted with a red box. An orange arrow points from the top right towards the 'Send' button, which is located to the right of the input field. The terminal output displays the following text:

```
SD Card Type: SDSC
XIAO ESP32S3 Sense Camera Image Capture
Send 'capture' to initiates an image capture

Picture Capture Command is sent
Writing file: /image1.jpg
File written
Photo saved to file
Saved picture: /image1.jpg

Picture Capture Command is sent
Writing file: /image2.jpg
File written
Photo saved to file
Saved picture: /image2.jpg
```

At the bottom of the terminal window, there are several configuration options: 'Autoscroll' (checked), 'Show timestamp' (unchecked), 'Both NL & CR' (selected), '115200 baud' (selected), and 'Clear output'.

Ecco un esempio di una foto scattata:

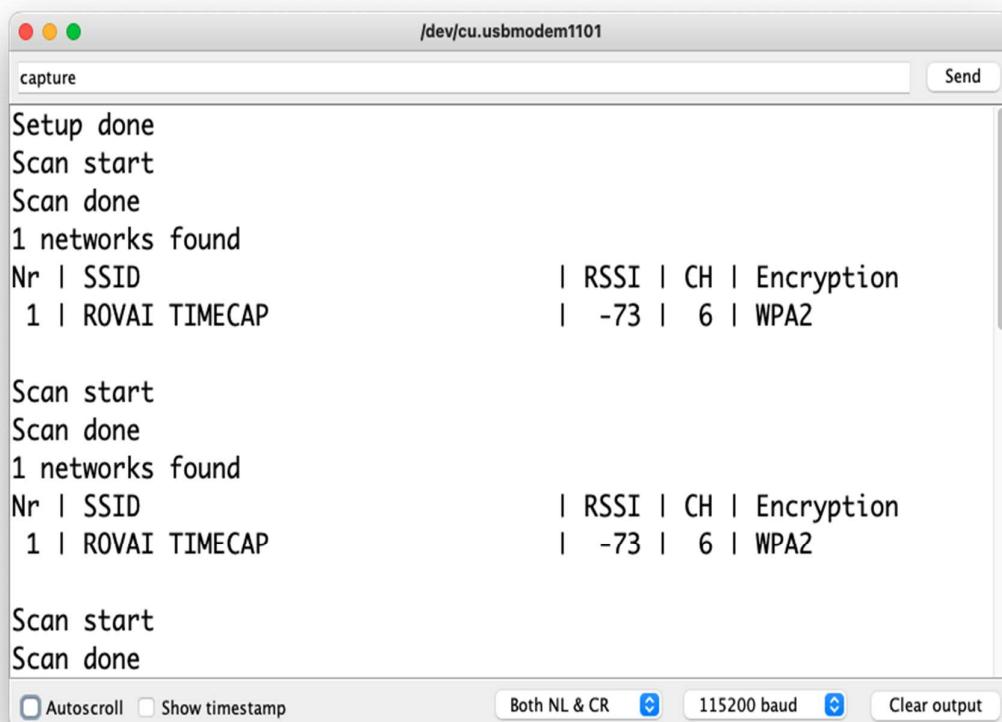


### 4.4.8 Test del WiFi

Uno dei differenziatori dello XIAO ESP32S3 è la sua capacità WiFi. Quindi, testiamo la sua radio, scansionando le reti WiFi intorno a sé. Lo si può fare eseguendo uno degli esempi di codice sulla scheda.

Si va su Arduino IDE Examples e si cerca **WiFi ==> WiFiScan**

Sul monitor seriale, si vedono le reti WiFi (SSID e RSSI) nel raggio del dispositivo. Ecco cosa abbiamo ottenuto in laboratorio:



The screenshot shows the Arduino Serial Monitor window titled '/dev/cu.usbmodem1101'. The text output is as follows:

```
capture
Setup done
Scan start
Scan done
1 networks found
Nr | SSID | RSSI | CH | Encryption
1 | ROVAI TIMECAP | -73 | 6 | WPA2

Scan start
Scan done
1 networks found
Nr | SSID | RSSI | CH | Encryption
1 | ROVAI TIMECAP | -73 | 6 | WPA2

Scan start
Scan done
```

At the bottom, there are checkboxes for 'Autoscroll' and 'Show timestamp', a dropdown for 'Both NL & CR', a dropdown for '115200 baud', and a 'Clear output' button.

### Un Semplice Server WiFi (accensione/spegnimento del LED)

Testiamo la capacità del dispositivo di comportarsi come un server WiFi. Ospiteremo una semplice pagina sul dispositivo che invia comandi per accendere e spegnere il LED integrato di XIAO.

Come prima, si va su GitHub per scaricare la cartella usando lo sketch [SimpleWiFiServer](#).

Prima di eseguire lo sketch, si devono inserire le credenziali di rete:

```
const char* ssid      = "Your credentials here";
const char* password = "Your credentials here";
```

Si può monitorare il funzionamento del server con Serial Monitor.

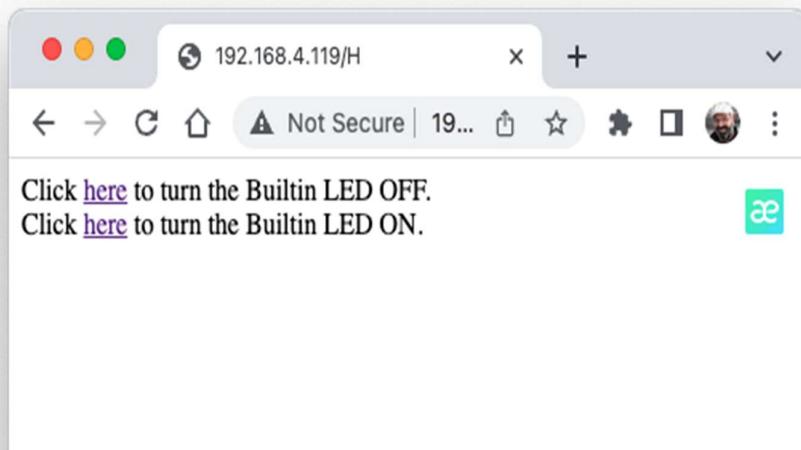
```

Connecting to ROVAI TIMECAP
...
WiFi connected.
IP address:
192.168.4.119
New Client.
GET / HTTP/1.1
Host: 192.168.4.119
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0 Safari/605.1.15
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,es;q=0.8,pt-BR;q=0.7,pt;q=0.6

```

Autoscroll  Show timestamp   Both NL & CR   115200 baud   Clear output

Prendere l'indirizzo IP e inserirlo nel browser:



Si vedrà una pagina con link che possono accendere e spegnere il LED integrato del XIAO.

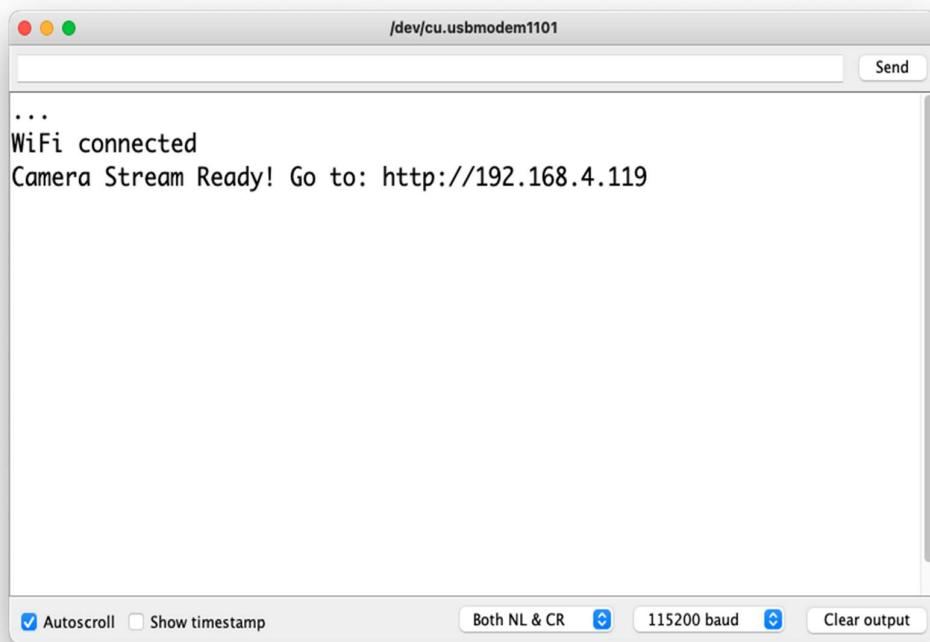
### Streaming video sul Web

Ora che sappiamo che si possono inviare comandi dalla pagina Web al dispositivo, facciamo il contrario. Prendiamo l'immagine catturata dalla telecamera e la trasmettiamo in streaming su una pagina Web:

Scaricare da GitHub la [cartella](#) che contiene il codice: XIAO-ESP32S3-Streaming\_Video.ino.

Ricordarsi che la cartella contiene il file .ino e un paio di file .h necessari per gestire la telecamera.

Inserire le credenziali ed eseguire lo sketch. Sul monitor seriale, si può trovare l'indirizzo della pagina da inserire nel browser:

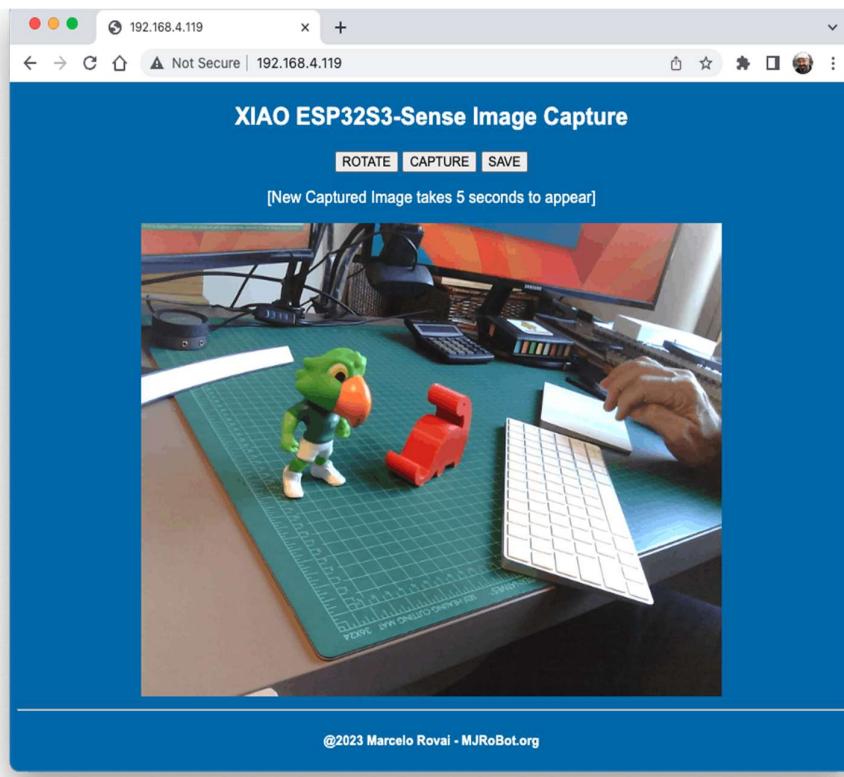


Apri la pagina sul browser (attendere qualche secondo per avviare lo streaming). Tutto qui.



Mettere in streaming ciò che la telecamera “vede” può essere importante quando la si posiziona per catturare un set di dati per un progetto ML (ad esempio, usando il codice “take\_photos\_commands.ino”).

Naturalmente, possiamo fare entrambe le cose contemporaneamente: mostrare ciò che la telecamera vede sulla pagina e inviare un comando per catturare e salvare l’immagine sulla scheda SD. Per questo, si può usare il codice Camera\_HTTP\_Server\_STA, scaricabile da GitHub.



Il programma eseguirà le seguenti attività:

- Imposta la fotocamera in modalità di output JPEG.
- Creare una pagina Web (ad esempio ==> <http://192.168.4.119/>). L'indirizzo corretto verrà visualizzato sul Serial Monitor.
- Se server.on (“/capture”, HTTP\_GET, serverCapture), il programma scatta una foto e la invia al Web.
- È possibile ruotare l'immagine sulla pagina Web utilizzando il pulsante [ROTATE]
- Il comando [CAPTURE] visualizzerà solo l'anteprima dell'immagine sulla pagina Web, mostrandone le dimensioni sul Serial Monitor
- Il comando [SAVE] salverà un'immagine sulla scheda SD e la mostrerà sul browser.
- Le immagini salvate seguiranno una denominazione sequenziale (image1.jpg, image2.jpg).

```
/dev/cu.usbmodem1101
...
WiFi connected..!
Got IP: 192.168.4.119
HTTP server started
Capturing Image for view only
The picture has a size of 143360 bytes
Saving Image to SD Card
Photo saved to file
Saved picture: /image1.jpg

Saving Image to SD Card
Photo saved to file
Saved picture: /image2.jpg

Autoscroll  Show timestamp  Both NL & CR  115200 baud  Clear output
```

Questo programma è utilizzabile per l'acquisizione di un dataset di immagini con un progetto di classificazione delle immagini.

Esamine il codice; sarà più facile capire come funziona la fotocamera. Questo codice è stato sviluppato sulla base del fantastico tutorial di Rui Santos [ESP32-CAM Take Photo and Display in Web Server](#), che invito tutti a visitare.

### Uso di CameraWebServer

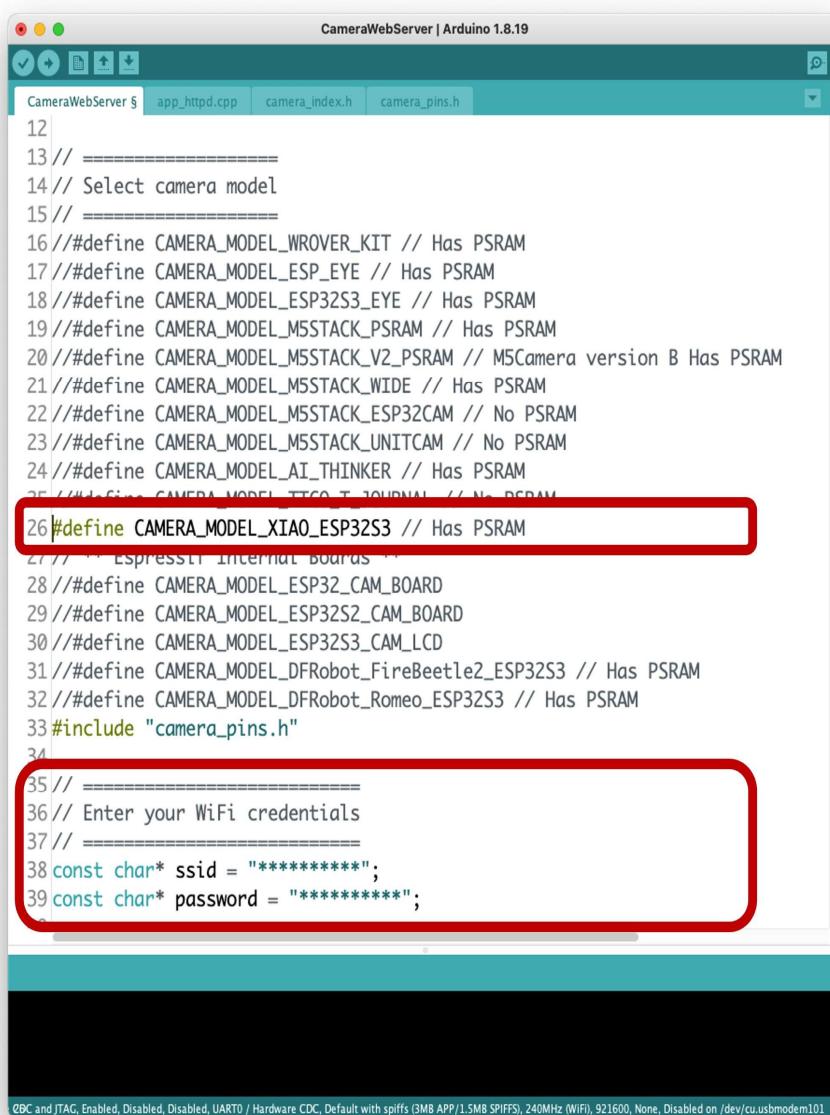
Nell'IDE Arduino, si va su `File > Examples > ESP32 > Camera` e si seleziona `CameraWebServer`.

Si devono anche commentare tutti i modelli di fotocamere, eccetto i pin del modello XIAO:

```
#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
```

E non dimenticare `Tools` per abilitare la PSRAM.

Inserire le credenziali wifi e caricare il codice sul dispositivo:



```
12
13 //=====
14 // Select camera model
15 //=====
16 //#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
17 //#define CAMERA_MODEL_ESP_EYE // Has PSRAM
18 //#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
19 //#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
20 //#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
21 //#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
22 //#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
23 //#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
24 //#define CAMERA_MODEL_AI_THINKER // Has PSRAM
25 //##define CAMERA_MODEL_TTC2_T_JOURNAL // No PSRAM
26 #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
27 // Espressif Internal Boards
28 //#define CAMERA_MODEL_ESP32_CAM_BOARD
29 //#define CAMERA_MODEL_ESP32S2_CAM_BOARD
30 //#define CAMERA_MODEL_ESP32S3_CAM_LCD
31 //#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
32 //#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
33 #include "camera_pins.h"
34
35 //=====
36 // Enter your WiFi credentials
37 //=====
38 const char* ssid = "*****";
39 const char* password = "*****";
```

Se il codice viene eseguito correttamente, si vedrà l'indirizzo sul monitor seriale:

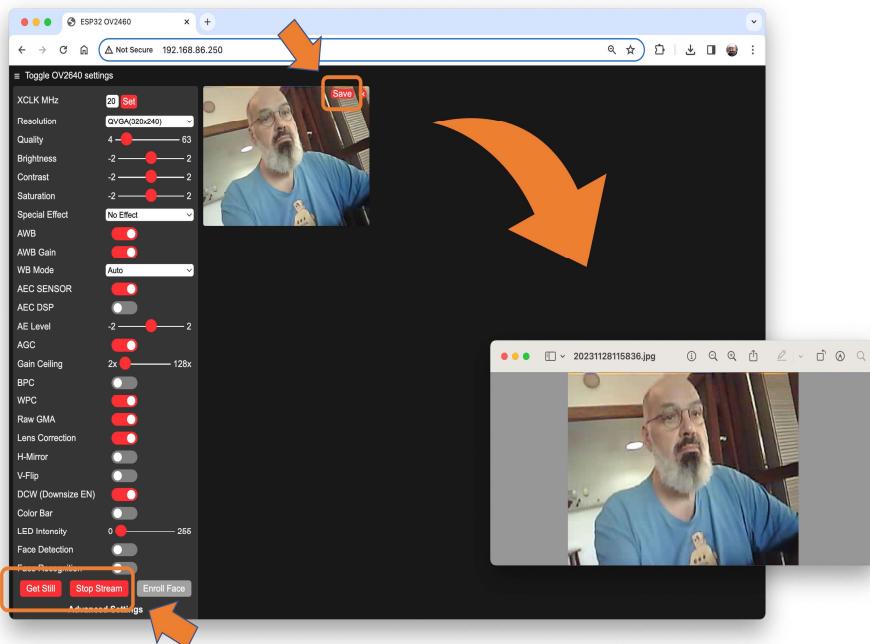
```

* WiFi connected
[ 1946][I][app_httpd.cpp:1361] startCameraServer(): Starting web server on port: '80'
[ 1948][I][app_httpd.cpp:13791 startCameraServer(): Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.86.250' to connect

```

*image-20240214163034559*

Copiare l'indirizzo sul browser e attendere che la pagina venga caricata. Selezionare la risoluzione della telecamera (ad esempio, QVGA) e selezionare [START STREAM]. Attendi qualche secondo/minuto, a seconda della connessione. Si può salvare un'immagine nell'area download del computer usando il pulsante [Save].



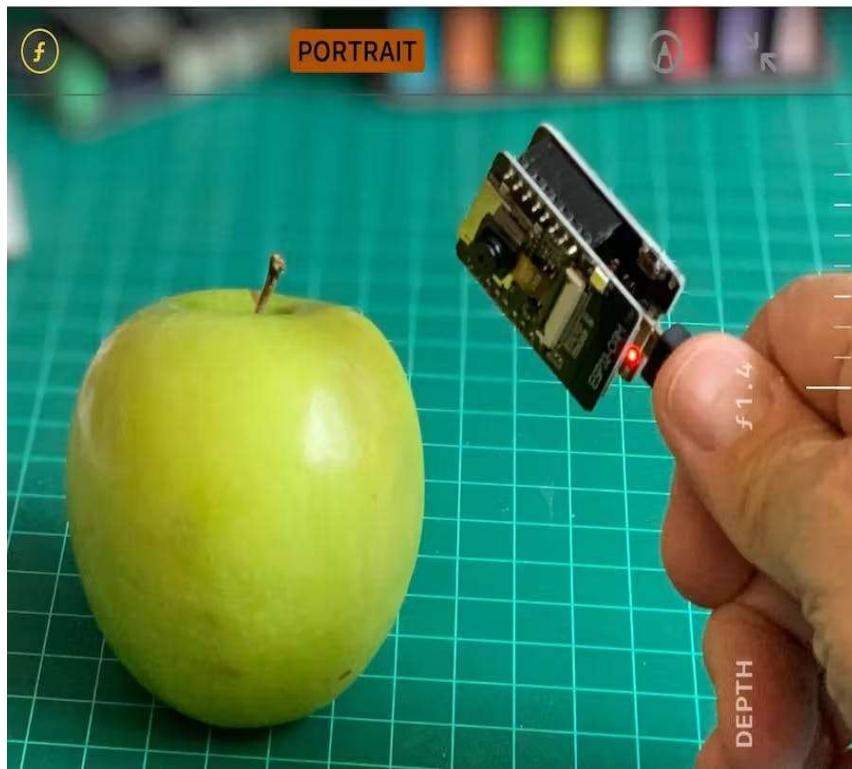
Ecco fatto! Si possono salvare le immagini direttamente sul computer per utilizzarle nei progetti.

#### 4.4.9 Frutta contro Verdura - Un progetto di Classificazione delle Immagini TinyML



Ora che abbiamo una fotocamera embedded in esecuzione, è il momento di provare la classificazione delle immagini. Per motivi comparativi, replicheremo lo stesso progetto di classificazione delle immagini sviluppato per essere utilizzato con una vecchia ESP2-CAM:

#### ESP32-CAM: Classificazione delle Immagini TinyML - Frutta contro Verdura



L'idea di base del nostro progetto sarà quella di addestrare un modello e procedere con l'inferenza su XIAO ESP32S3 Sense. Per l'addestramento, dovremmo trovare alcuni dati (**in effetti, tonnellate di dati!**).

*Ma prima di tutto, abbiamo bisogno di un obiettivo! Cosa vogliamo classificare?*

Con TinyML, un set di tecniche associate all'inferenza di apprendimento automatico su dispositivi embedded, dovremmo limitare la classificazione a tre o quattro categorie a causa di limitazioni (principalmente della memoria). Differenziamo **mele** da **banane** e **patate** (si possono provare altre categorie).

Quindi, cerchiamo un set di dati specifico che includa immagini da quelle categorie. Kaggle è un buon inizio:

<https://www.kaggle.com/kritikseth/fruit-and-vegetable-image-recognition>

Questo set di dati contiene immagini dei seguenti alimenti:

- **Frutta** - banana, mela, pera, uva, arancia, kiwi, anguria, melograno, ananas, mango.
- **Verdura** - cetriolo, carota, peperone, cipolla, patata, limone, pomodoro, ravanello, barbabietola, cavolo, lattuga, spinaci, soia, cavolfiore, peperone, peperoncino, rapa, mais, mais dolce, patata dolce, paprika, jalepeño, zenzero, aglio, piselli, melanzane.

Ogni categoria è suddivisa in **train** [addestramento] (100 immagini), **test** (10 immagini) e **validation** (10 immagini).

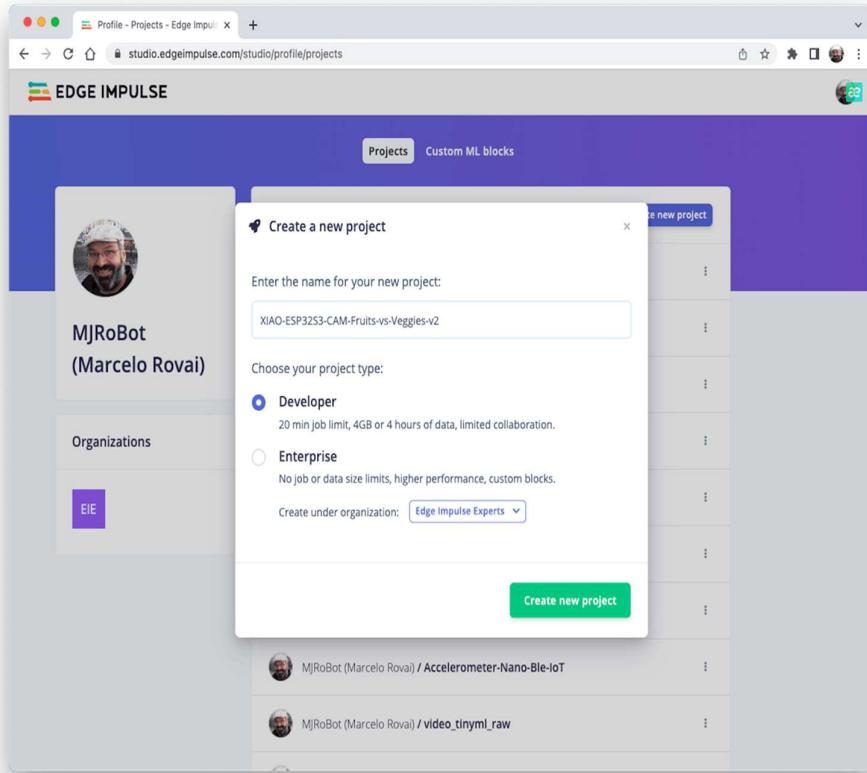
- Scaricare il dataset dal sito web di Kaggle sul computer.

Facoltativamente, si possono aggiungere alcune foto fresche di banane, mele e patate dalla propria cucina di casa, utilizzando, ad esempio, i codici discussi nell'ultima sezione.

#### 4.4.10 Addestramento del modello con Edge Impulse Studio

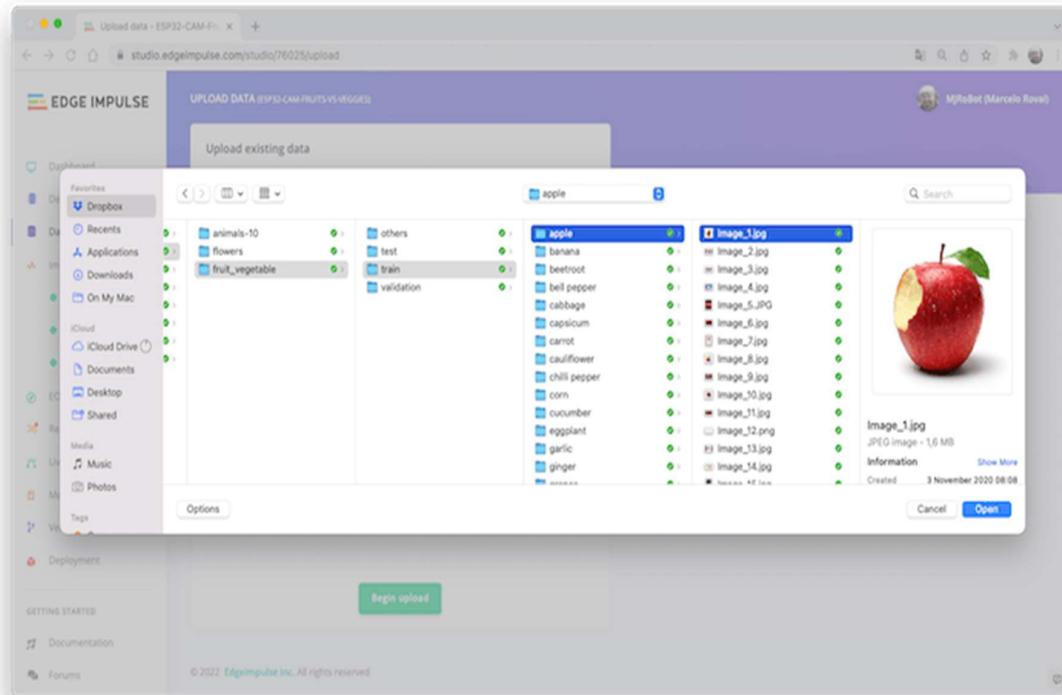
Utilizzeremo Edge Impulse Studio per addestrare il modello. Come si sa, [Edge Impulse](#) è una piattaforma di sviluppo leader per l'apprendimento automatico su dispositivi edge.

Inserire le credenziali del proprio account (o creare uno gratuito) su Edge Impulse. Quindi, creare un nuovo progetto:



#### Acquisizione Dati

Successivamente, nella sezione UPLOAD DATA, si caricano dal computer i file delle categorie scelte:



Sarebbe meglio se ora si avesse il proprio set di dati di training suddiviso in tre classi di dati:

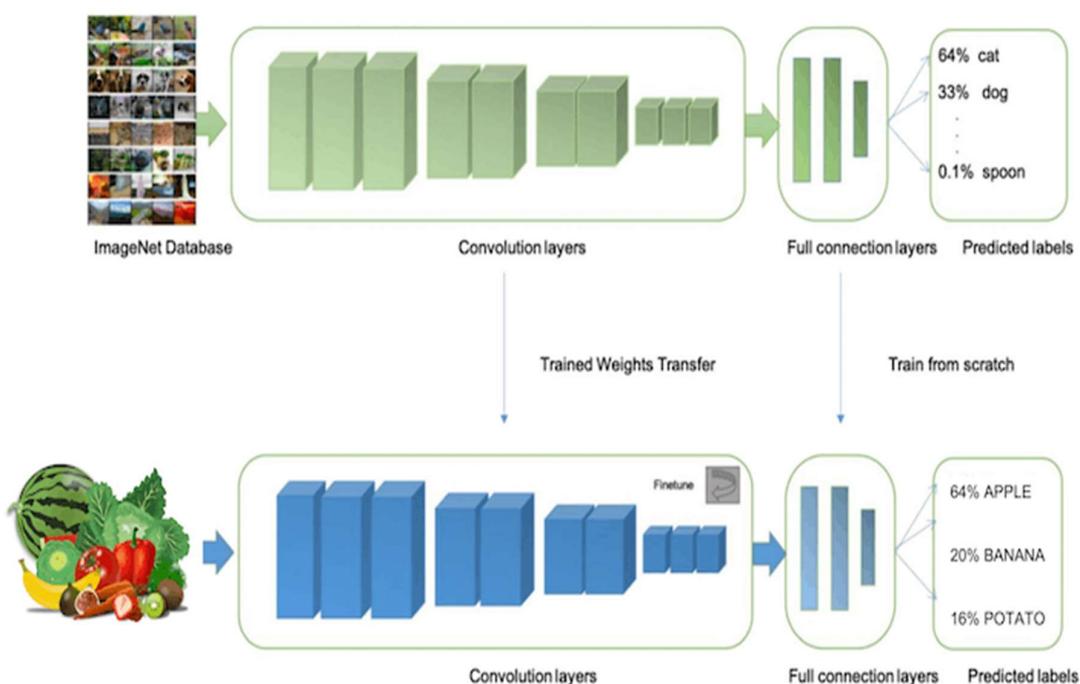
A screenshot of the Edge Impulse Studio interface showing the 'Data acquisition' section. The left sidebar includes 'Dashboard', 'Devices', 'Data acquisition', 'Impulse design', 'EON Tuner', 'Retrain model', 'Live classification', 'Model testing', 'Versioning', 'Deployment', 'Documentation', and 'Forums'. The main area shows a table titled 'Collected data' with columns 'SAMPLE NAME', 'LABEL', 'ADDED', and 'LENGTH'. All samples listed are labeled as 'banana'. A preview of a banana image is shown under the heading 'RAW DATA' with the caption 'Image\_89.jpg.2p4ed0vt'. A message at the top says 'Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options'.

Si possono caricare dati extra per ulteriori test del modello o dividere i dati di training. Lo lasceremo così com'è per utilizzare più dati possibili.

### Impulse Design

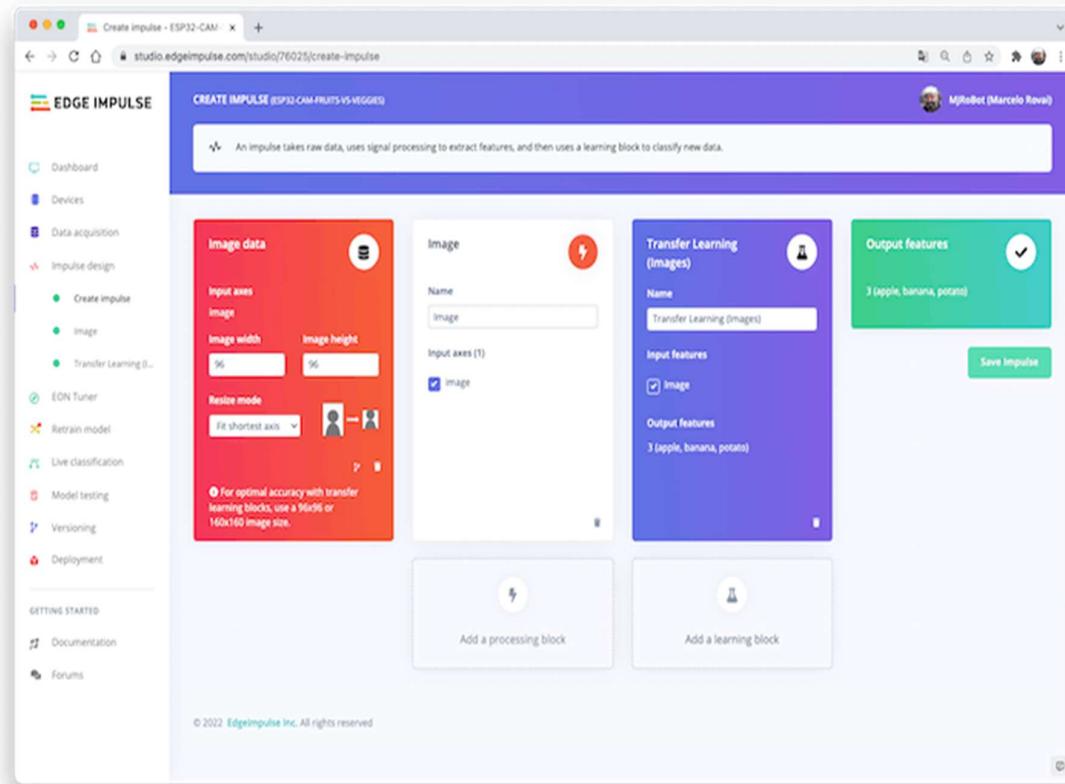
Un impulso prende dati grezzi (in questo caso, immagini), estrae feature (ridimensiona le immagini) e poi utilizza un blocco di apprendimento per classificare nuovi dati.

La classificazione delle immagini è l'uso più comune del deep learning, ma per portare a termine questo compito dovrebbero essere utilizzati molti dati. Abbiamo circa 90 immagini per ogni categoria. Questo numero è sufficiente? No, per niente! Avremo bisogno di migliaia di immagini per "insegnare o modellare" per distinguere una mela da una banana. Ma possiamo risolvere questo problema riaddestrando un modello precedentemente addestrato con migliaia di immagini. Chiamiamo questa tecnica "Transfer Learning" (TL).



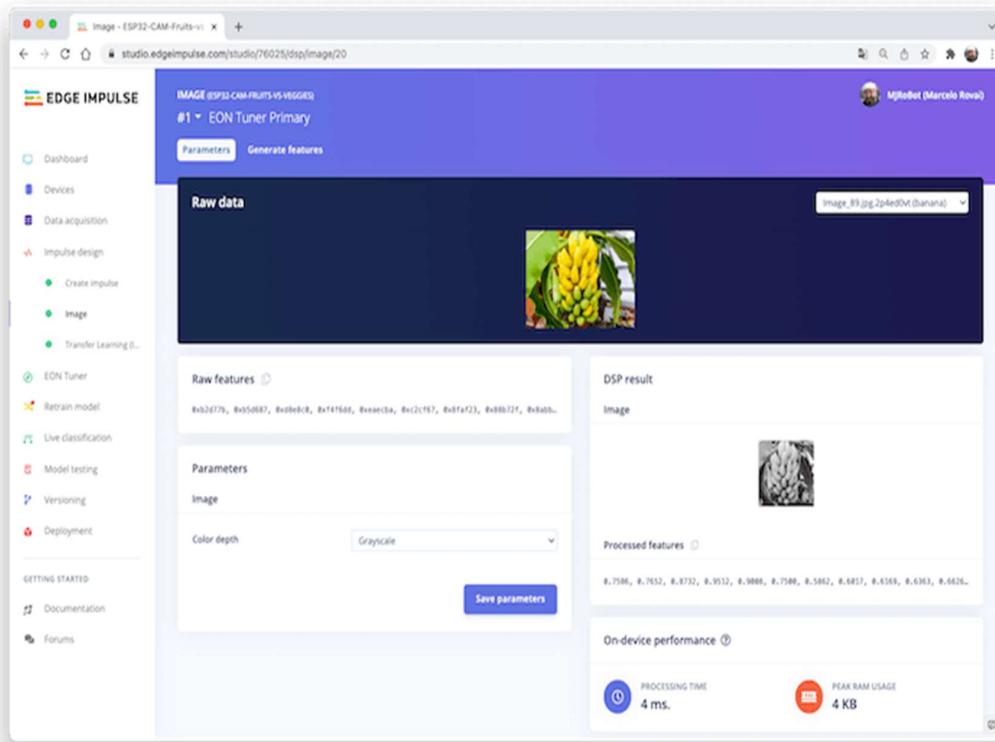
Col TL, possiamo mettere a punto un modello di classificazione delle immagini pre-addestrato sui nostri dati, che funziona bene anche con set di dati di immagini relativamente piccoli (il nostro caso).

Quindi, partendo dalle immagini grezze, le ridimensioneremo (96x96) pixel e le invieremo al nostro blocco Transfer Learning:



### Pre-processing (generazione delle feature)

Oltre a ridimensionare le immagini, possiamo cambiarle in scala di grigi o mantenere la profondità di colore RGB effettiva. Iniziamo selezionando **Grayscale**. In questo modo, ognuno dei nostri campioni di dati avrà dimensione 9.216 feature (96x96x1). Con RGB, questa dimensione sarebbe tre volte più grande. Lavorare con la scala di grigi aiuta a ridurre la quantità di memoria finale necessaria per l'inferenza.



Non dimenticare il `[save parameters]`. Questo genererà le funzionalità da utilizzare nella formazione.

### Training (Transfer Learning & Data Augmentation)

Nel 2007, Google ha introdotto **MobileNetV1**, una famiglia di reti neurali di visione artificiale di uso generale progettate pensando ai dispositivi mobili per supportare classificazione, rilevamento e altro. Le MobileNet sono modelli piccoli, a bassa latenza e a basso consumo, parametrizzati per soddisfare i vincoli di risorse di vari casi d'uso.

Sebbene l'architettura di base di MobileNet sia già minuscola e abbia una bassa latenza, spesso un caso d'uso o un'applicazione specifica potrebbe richiedere che il modello sia più piccolo e veloce. MobileNet introduce un parametro semplice  $\alpha$  (alfa) chiamato moltiplicatore di larghezza per costruire questi modelli più piccoli e meno costosi dal punto di vista computazionale. Il ruolo del moltiplicatore di larghezza  $\alpha$  è di assottigliare una rete in modo uniforme a ogni livello.

Edge Impulse Studio ha a disposizione MobileNet V1 (immagini 96x96) e V2 (immagini 96x96 e 160x160), con diversi valori  $\alpha$  (da 0.05 a 1.0). Ad esempio, si otterrà la massima accuratezza con V2, immagini 160x160 e  $\alpha=1.0$ . Naturalmente, c'è un compromesso. Maggiore è la precisione, maggiore sarà la memoria (circa 1,3 M di RAM e 2,6 M di ROM) necessaria per eseguire il modello, il che implica una maggiore latenza.

L'ingombro più piccolo sarà ottenuto all'altro estremo con **MobileNet V1** e  $\alpha=0.10$  (circa 53,2 K di RAM e 101 K di ROM).

Quando abbiamo pubblicato per la prima volta questo progetto per l'esecuzione su un ESP32-CAM, ci siamo mantenuti sul lato inferiore delle possibilità, il che ha garantito l'inferenza con una latenza ridotta ma non con un'accuratezza elevata. Per questo primo passaggio, manterremo questo modello di progettazione (**MobileNet V1** e  $\alpha=0.10$ ).

Un'altra tecnica necessaria da usare con il deep learning è il **data augmentation**. Il data augmentation è un metodo che può aiutare a migliorare l'accuratezza dei modelli di machine learning, creando dati artificiali aggiuntivi. Un sistema di Data Augmentation apporta piccole modifiche casuali ai dati di training (ad esempio capovolgendo, ritagliando o ruotando le immagini).

Sotto, qui si può vedere come Edge Impulse implementa una policy di data Augmentation sui dati:

```
# Implements the data augmentation policy
def augment_image(image, label):
    # Flips the image randomly
    image = tf.image.random_flip_left_right(image)

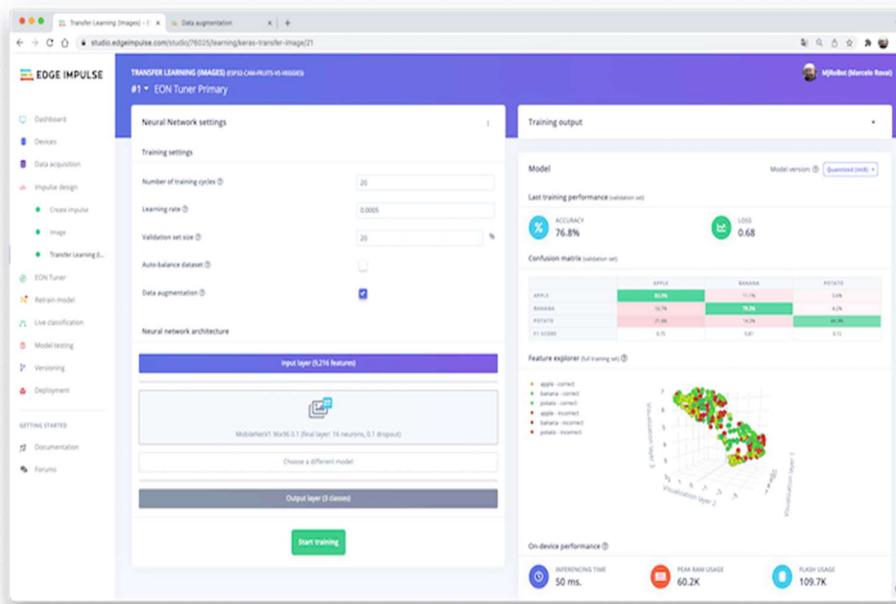
    # Increase the image size, then randomly crop it down to
    # the original dimensions
    resize_factor = random.uniform(1, 1.2)
    new_height = math.floor(resize_factor * INPUT_SHAPE[0])
    new_width = math.floor(resize_factor * INPUT_SHAPE[1])
    image = tf.image.resize_with_crop_or_pad(image, new_height, new_width)
    image = tf.image.random_crop(image, size=INPUT_SHAPE)

    # Vary the brightness of the image
    image = tf.image.random_brightness(image, max_delta=0.2)

    return image, label
```

L'esposizione a queste variazioni durante l'addestramento può aiutare a impedire al modello di prendere scorciatoie "memorizzando" indizi superficiali nei dati di addestramento, il che significa che potrebbe riflettere meglio i pattern profondi in esame nel set di dati.

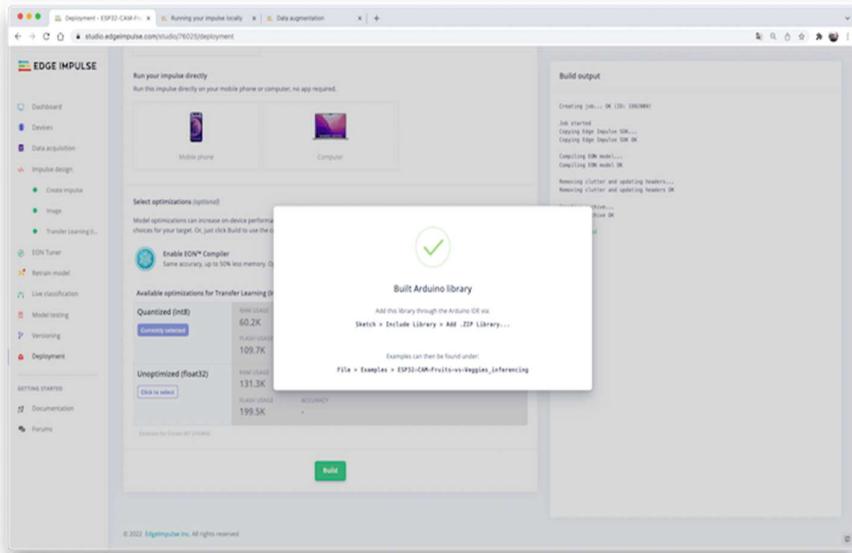
L'ultimo layer del nostro modello avrà 16 neuroni con un dropout del 10% per prevenire l'overfitting. Ecco l'output del Training:



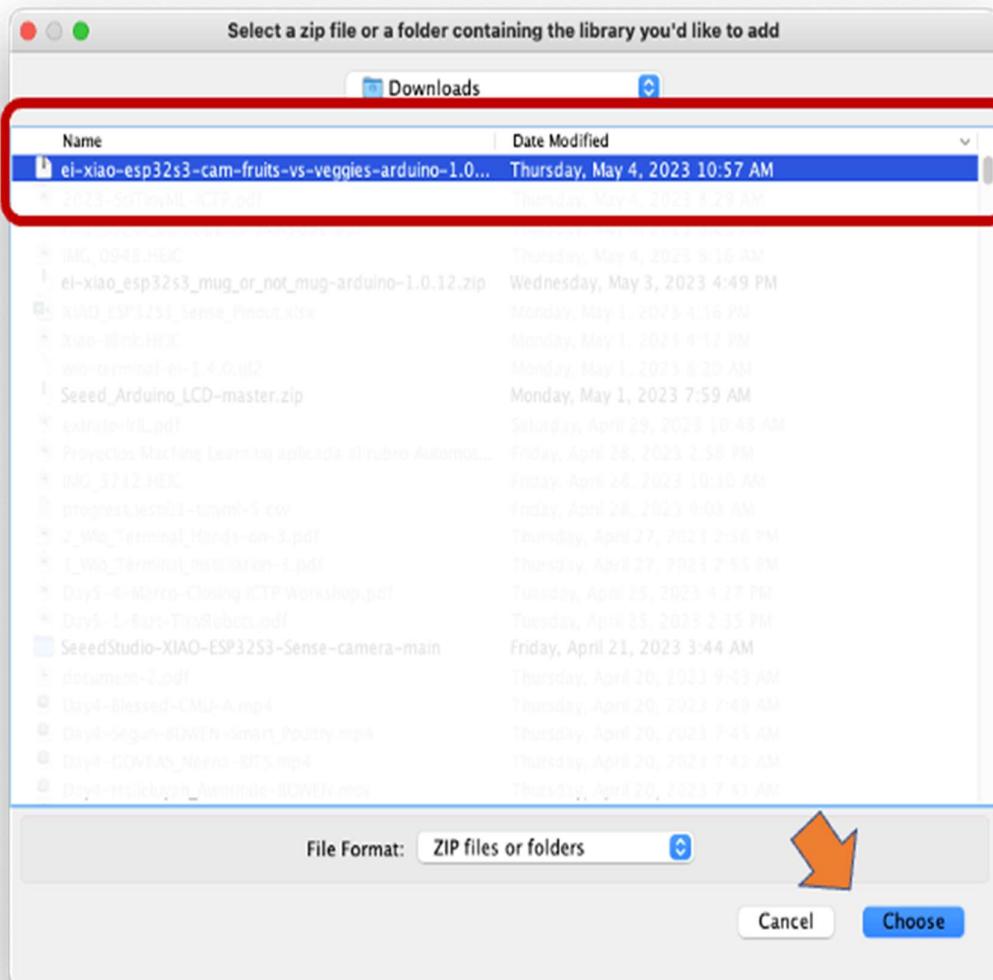
Il risultato potrebbe essere migliore. Il modello ha raggiunto circa il 77% di accuratezza, ma la quantità di RAM prevista per essere utilizzata durante l'inferenza è relativamente piccola (circa 60 KByte), il che è molto buono.

## Deployment

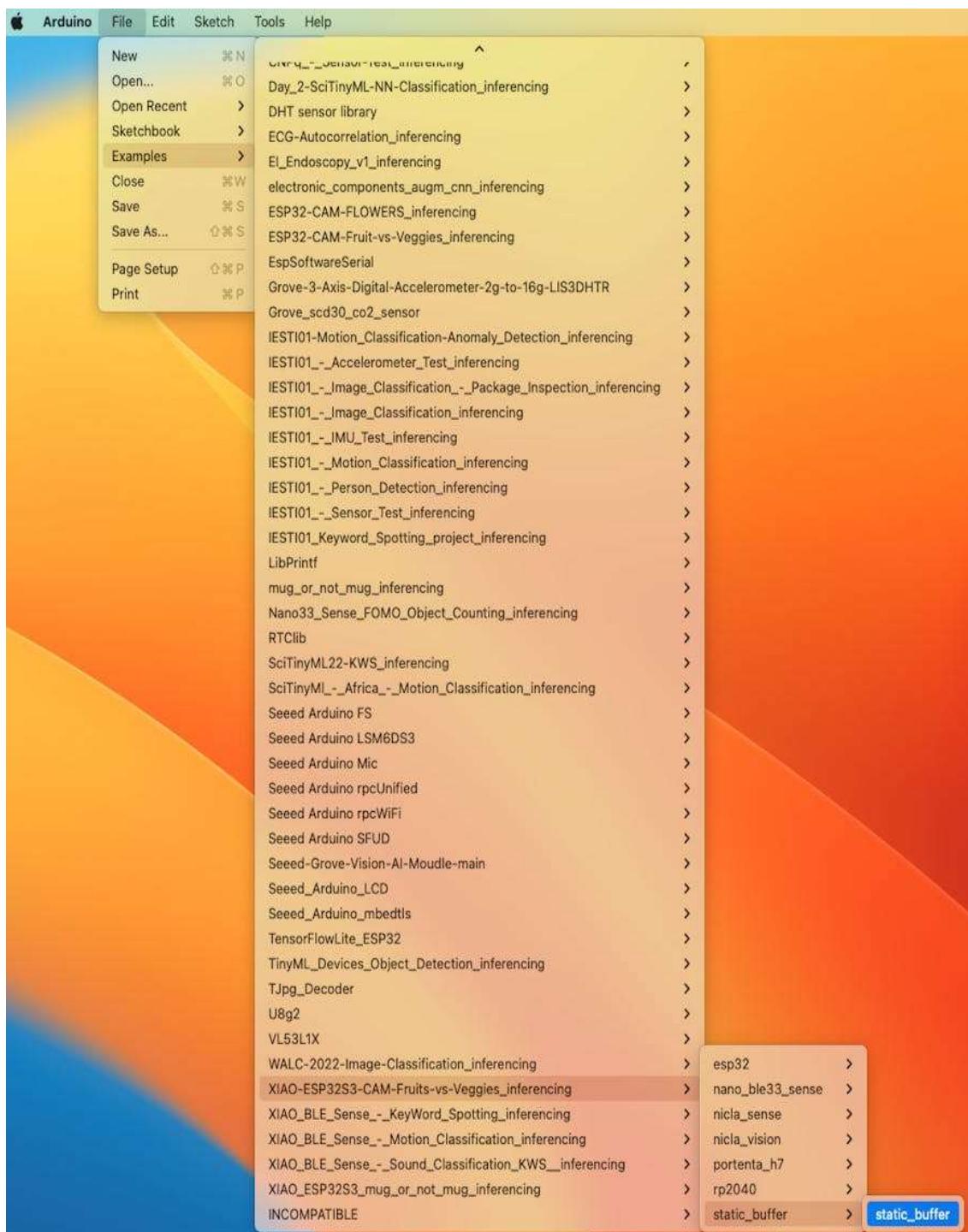
Il modello addestrato verrà distribuito come libreria Arduino .zip:



Apri l'IDE Arduino e in **Sketch**, si va su **Include Library** e **add.ZIP Library**. Selezionare il file che scaricato da Edge Impulse Studio e il gioco è fatto!



Nella scheda **Examples** su Arduino IDE, si trova un codice sketch sotto il nome del proprio progetto.



Aprire l'esempio Static Buffer:

```

static_buffer | Arduino 1.8.19

static_buffer

15 */
16
17 /* Includes ----- */
18 #include <XIAO-ESP32S3-CAM-Fruits-vs-Veggies_inferencing.h>
19
20 static const float features[] = {
21     // copy raw features here (for example from the 'Live classification' page)
22     // see https://docs.edgeimpulse.com/docs/running-your-impulse-arduino
23 };
24
25 /**
26 * @brief      Copy raw feature data in out_ptr
27 *             Function called by inference library
28 *
29 * @param[in]  offset    The offset
30 * @param[in]  length   The length
31 * @param      out_ptr   The out pointer
32 *
33 * @return     0

```

Mb), Core 1, Core 1, Hardware CDC and JTAG, Enabled, Disabled, Disabled, UART0 / Hardware CDC, Default with spiffs (3MB APP/1.5MB SPIFFS), 240MHz (WiFi), 921600, None, Disabled on /dev/cu.usbmodem1101

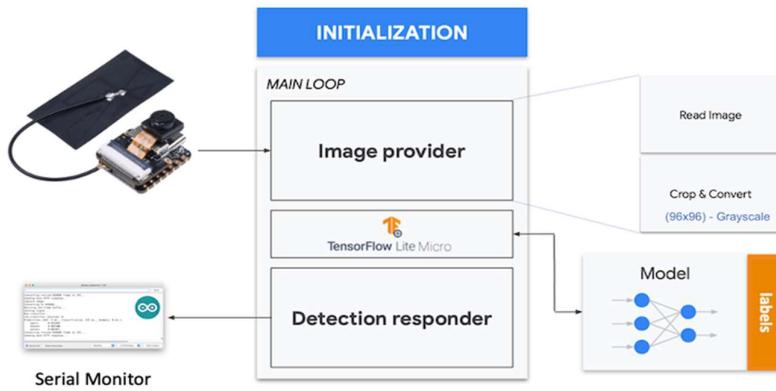
La prima riga di codice è esattamente la chiamata di una libreria con tutto il necessario per eseguire l'inferenza sul dispositivo.

---

```
#include <XIAO-ESP32S3-CAM-Fruits-vs-Veggies_inferencing.h>
```

Ovviamente, questo è un codice generico (un “template”) che ottiene solo un campione di dati grezzi (memorizzati nella variabile: `features = {}`) ed esegue il classificatore, eseguendo l'inferenza. Il risultato viene mostrato sul monitor seriale.

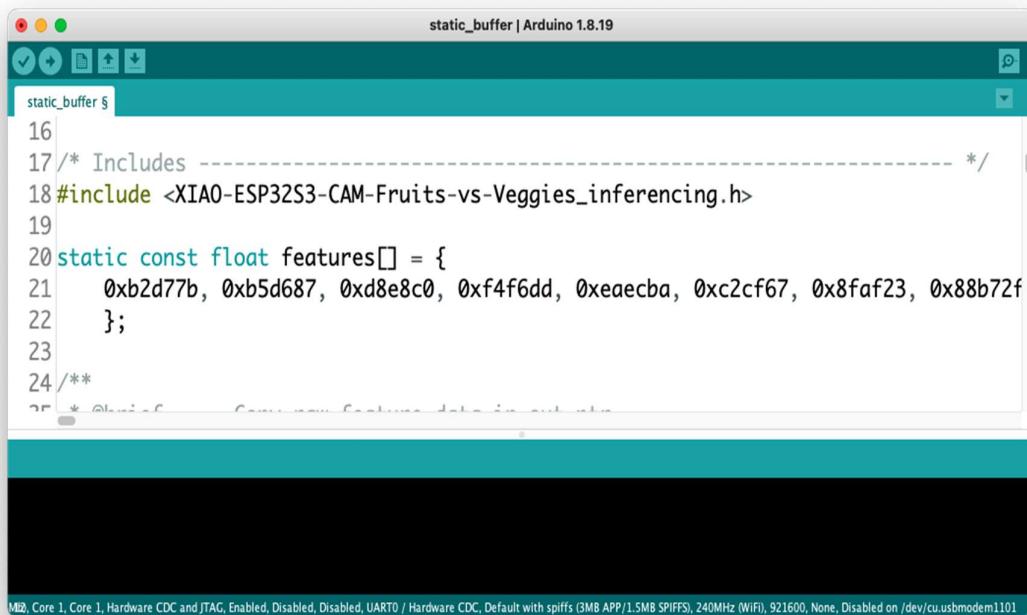
Dovremmo ottenere il campione (immagine) dalla fotocamera e pre-elaborarlo (ridimensionandolo a 96x96, convertendolo in scala di grigi e appiattendolo). Questo sarà il tensore di input del nostro modello. Il tensore di output sarà un vettore con tre valori (etichette), che mostrano le probabilità di ciascuna delle classi.



Tornando al progetto (Tab Image), copiare uno dei Raw Data Sample:

A screenshot of the Edge Impulse studio interface. The left sidebar shows project navigation with sections like Dashboard, Devices, Data acquisition, EON Tuner, Retrain model, Live classification, Model testing, Versioning, Deployment, and Getting Started. The main area is titled '#1 ▾ EON Tuner Primary'. It displays a 'Raw data' section with a preview image of a bunch of bananas and a button labeled 'Copy 9216 features to clipboard'. Below this are sections for 'Raw features' (with a dropdown menu), 'Parameters' (with a 'Save parameters' button), 'DSP result' (showing an image of a banana), 'Processed features' (listing values like 0.7586, 0.7652, etc.), and 'On-device performance' (showing processing time as 15 ms and peak RAM usage as 4 KB). The URL at the bottom is <https://studio.edgeimpulse.com/studio/221775/dsp/image/20#>.

9.216 feature verranno copiate negli appunti. Questo è il tensore di input (un'immagine appiattita di 96x96x1), in questo caso, banane. Oltre questo tensore di input sulle feature[] = {0xb2d77b, 0xb5d687, 0xd8e8c0, 0xeaecba, 0xc2cf67, ...}



```

static_buffer | Arduino 1.8.19

static_buffer §

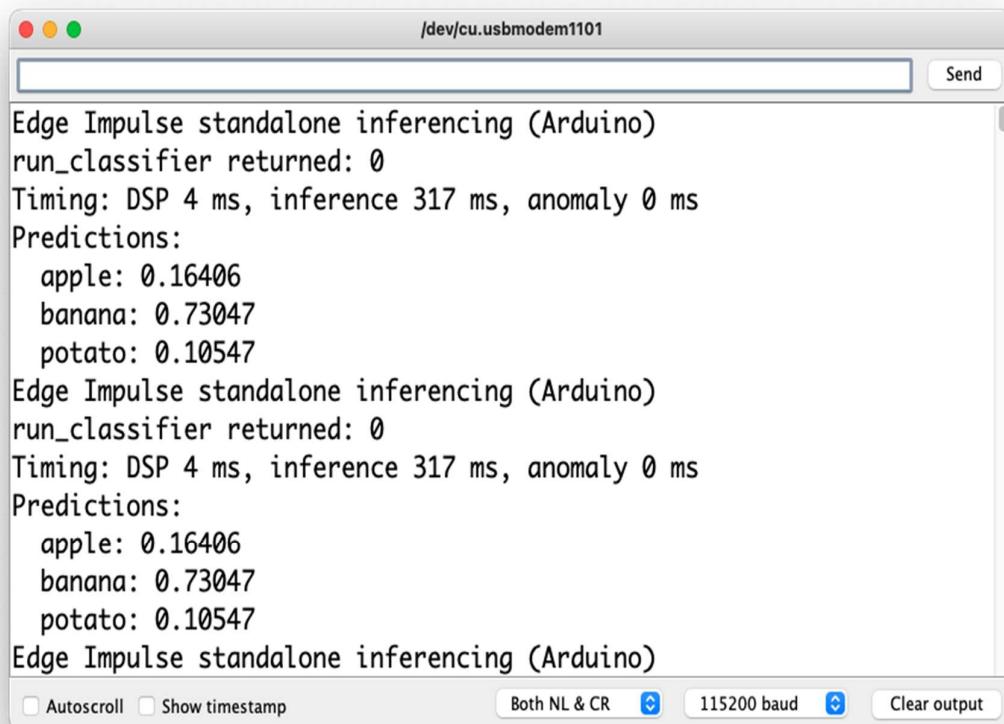
16
17 /* Includes -----
18 #include <XIAO-ESP32S3-CAM-Fruits-vs-Veggies_inferencing.h>
19
20 static const float features[] = {
21     0xb2d77b, 0xb5d687, 0xd8e8c0, 0xf4f6dd, 0xeaecba, 0xc2cf67, 0x8faf23, 0x88b72f
22 };
23
24 /**
25 * -----
26 */

```

M2, Core 1, Core 1, Hardware CDC and JTAG, Enabled, Disabled, UART0 / Hardware CDC, Default with spiffs (3MB APP/1.5MB SPIFFS), 240MHz (WiFi), 921600, None, Disabled on /dev/cu.usbmodem1101

Edge Impulse ha incluso la [libreria ESP NN](#) nel suo SDK, che contiene funzioni NN (Neural Network) ottimizzate per vari chip Espressif, tra cui ESP32S3 (in esecuzione su Arduino IDE).

Quando si esegue l'inferenza, si deve ottenere il punteggio più alto per "banana".



```

/dev/cu.usbmodem1101
Send

Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Timing: DSP 4 ms, inference 317 ms, anomaly 0 ms
Predictions:
apple: 0.16406
banana: 0.73047
potato: 0.10547
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Timing: DSP 4 ms, inference 317 ms, anomaly 0 ms
Predictions:
apple: 0.16406
banana: 0.73047
potato: 0.10547
Edge Impulse standalone inferencing (Arduino)

 Autoscroll  Show timestamp Both NL & CR 115200 baud Clear output

```

Ottime notizie! Il nostro dispositivo gestisce un'inferenza, scoprendo che l'immagine in ingresso è una banana. Notare inoltre che il tempo di inferenza è stato di circa 317 ms, con

un massimo di 3 fps se si è provato a classificare le immagini da un video. È un risultato migliore della ESP32 CAM (525 ms di latenza).

Ora, dovremmo incorporare la telecamera e classificare le immagini in tempo reale.

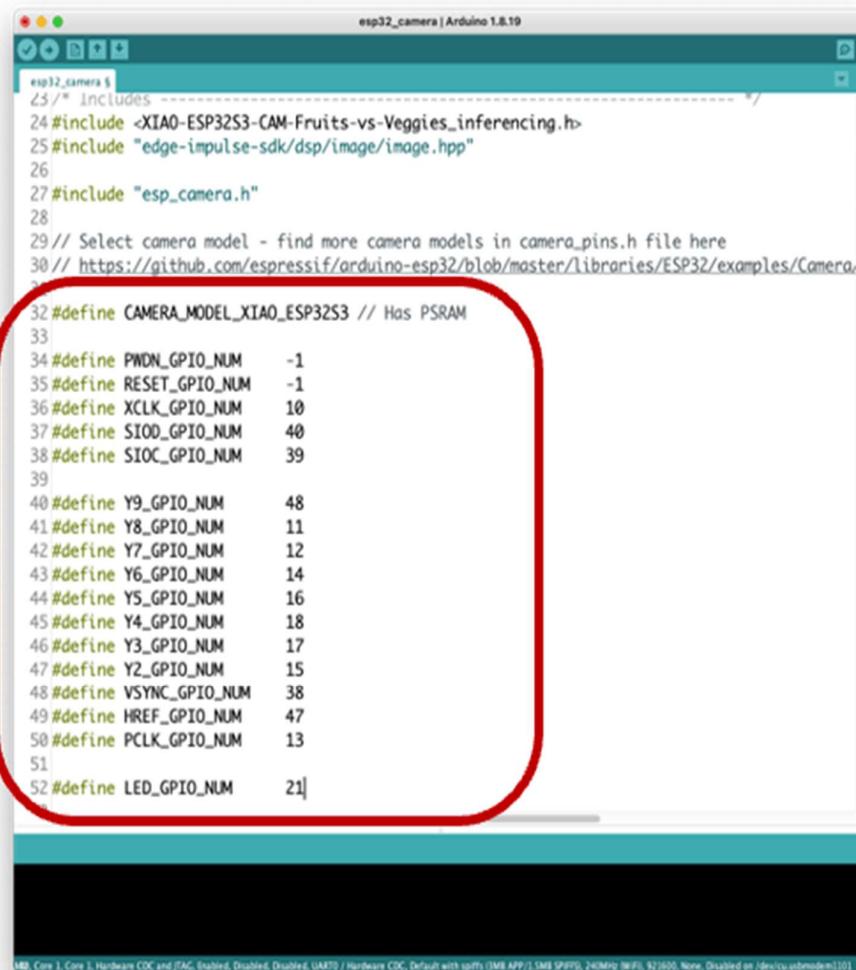
Si va su Arduino IDE Examples e si scarica dal progetto lo sketch esp32\_camera:



Si devono cambiare le righe dalla 32 alla 75, che definiscono il modello e i pin della telecamera, utilizzando i dati relativi al nostro modello. Copiare e incollare le righe seguenti, sostituendo le righe 32-75:

```
#define PWDN_GPIO_NUM      -1
#define RESET_GPIO_NUM     -1
#define XCLK_GPIO_NUM       10
#define SIOD_GPIO_NUM        40
#define SIOC_GPIO_NUM        39
#define Y9_GPIO_NUM          48
#define Y8_GPIO_NUM          11
#define Y7_GPIO_NUM          12
#define Y6_GPIO_NUM          14
#define Y5_GPIO_NUM          16
#define Y4_GPIO_NUM          18
#define Y3_GPIO_NUM          17
#define Y2_GPIO_NUM          15
#define VSYNC_GPIO_NUM       38
#define HREF_GPIO_NUM        47
#define PCLK_GPIO_NUM        13
```

Ecco il codice risultante:



```
esp32_camera | Arduino 1.8.19

3 /* includes */
4 #include <XIAO-ESP32S3-CAM-Fruits-vs-Veggies_inferencing.h>
5 #include "edge-impulse-sdk/dsp/image/image.hpp"
6
7 #include "esp_camera.h"
8
9 // Select camera model - find more camera models in camera_pins.h file here
10 // https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP32/examples/Camera/
11
12 #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
13
14 #define PWDN_GPIO_NUM      -1
15 #define RESET_GPIO_NUM    -1
16 #define XCLK_GPIO_NUM     10
17 #define SIOD_GPIO_NUM     40
18 #define SIOC_GPIO_NUM     39
19
20 #define Y9_GPIO_NUM        48
21 #define Y8_GPIO_NUM        11
22 #define Y7_GPIO_NUM        12
23 #define Y6_GPIO_NUM        14
24 #define Y5_GPIO_NUM        16
25 #define Y4_GPIO_NUM        18
26 #define Y3_GPIO_NUM        17
27 #define Y2_GPIO_NUM        15
28 #define VSYNC_GPIO_NUM     38
29 #define HREF_GPIO_NUM      47
30 #define PCLK_GPIO_NUM      13
31
32 #define LED_GPIO_NUM       21|
```

Lo sketch modificato è scaricabile da GitHub: [xiao\\_esp32s3\\_camera](#).

Notare che è possibile, facoltativamente, conservare i pin come file .h, come abbiamo fatto nelle sezioni precedenti.

Caricare il codice sullo XIAO ESP32S3 Sense e si sarà pronti per iniziare a classificare la frutta e la verdura! Si può controllare il risultato su Serial Monitor.

#### 4.4.11 Test del modello (inferenza)



Scattando una foto con la fotocamera, il risultato della classificazione apparirà su Serial Monitor:

```
banana: 0.90234
potato: 0.03906
Predictions (DSP: 4 ms., Classification: 318 ms., Anomaly: 0 ms.):
apple: 0.03906
banana: 0.93359
potato: 0.02734
Predictions (DSP: 4 ms., Classification: 317 ms., Anomaly: 0 ms.):
apple: 0.05469
banana: 0.90625
potato: 0.03906
Predictions (DSP: 4 ms., Classification: 318 ms., Anomaly: 0 ms.):
apple: 0.04297
banana: 0.92578
potato: 0.03125

 Autoscroll  Show timestamp Both NL & CR 115200 baud Clear output
```

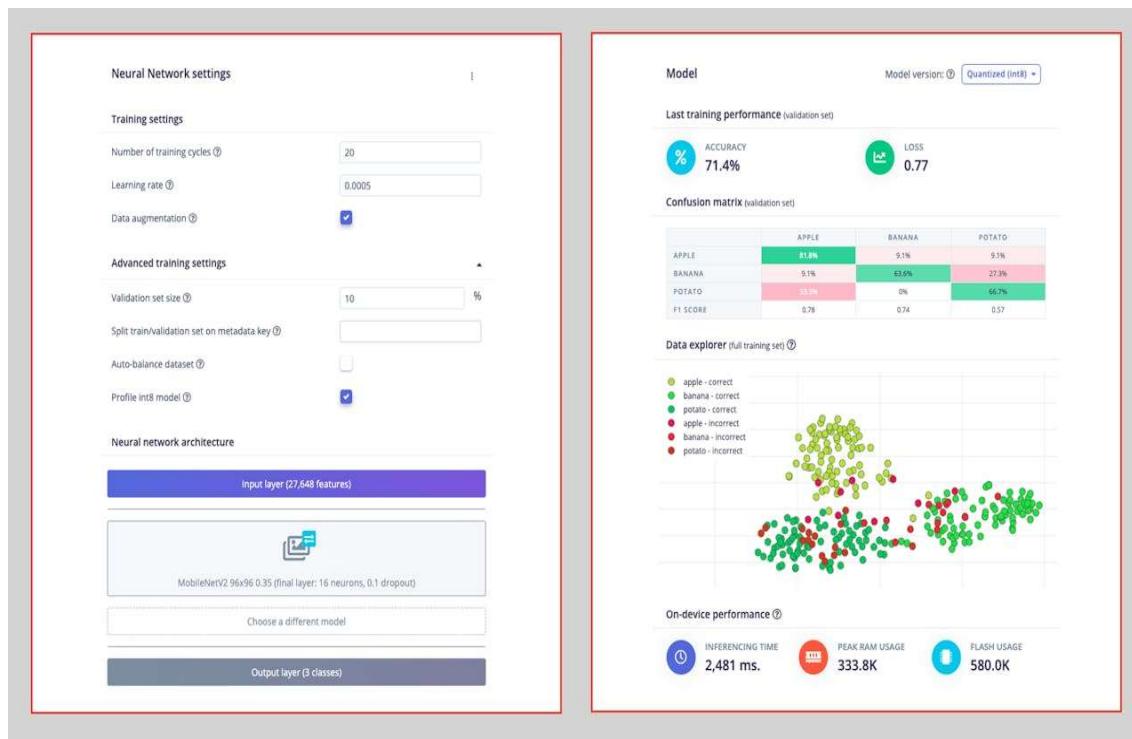
Altri test:

```
/dev/cu.usbmodem1101
Send
banana: 0.14844
potato: 0.12891
Predictions (DSP: 4 ms., Classification: 318 ms., Anomaly: 0 ms.):
apple: 0.78906
banana: 0.06641
potato: 0.14453
Predictions (DSP: 4 ms., Classification: 317 ms., Anomaly: 0 ms.):
apple: 0.71484
banana: 0.06641
potato: 0.21875
Predictions (DSP: 4 ms., Classification: 318 ms., Anomaly: 0 ms.):
apple: 0.79297
banana: 0.05469
potato: 0.14844
Autoscroll Show timestamp Both NL & CR 115200 baud Clear output
```



#### 4.4.12 Test con un Modello Più Grande

Ora, passiamo all'altro lato delle dimensioni del modello. Selezioniamo un MobilinetV2 96x96 0.35, con immagini RGB in input.



Anche con un modello più grande, la precisione potrebbe essere migliore e la quantità di memoria necessaria per eseguire il modello aumenta di cinque volte, con una latenza che aumenta di sette volte.

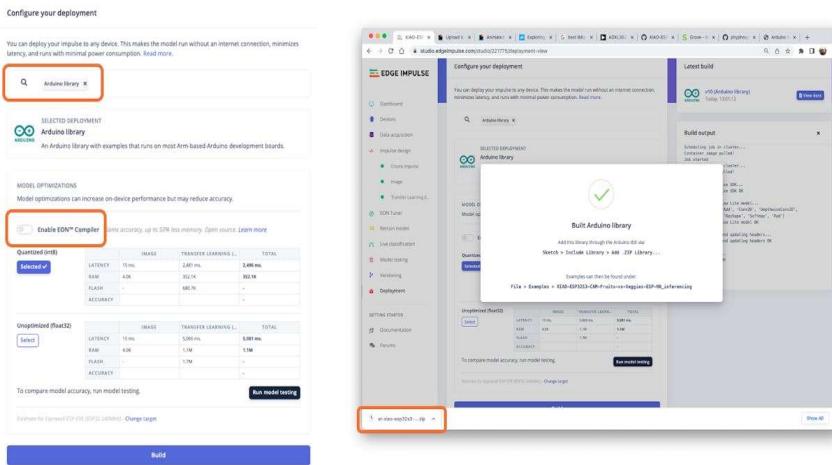
Notare che le prestazioni qui sono stimate con un dispositivo più piccolo, l'ESP-EYE. L'inferenza effettiva con l'ESP32S3 dovrebbe essere migliore.

Per migliorare il nostro modello, dovremo addestrare più immagini.

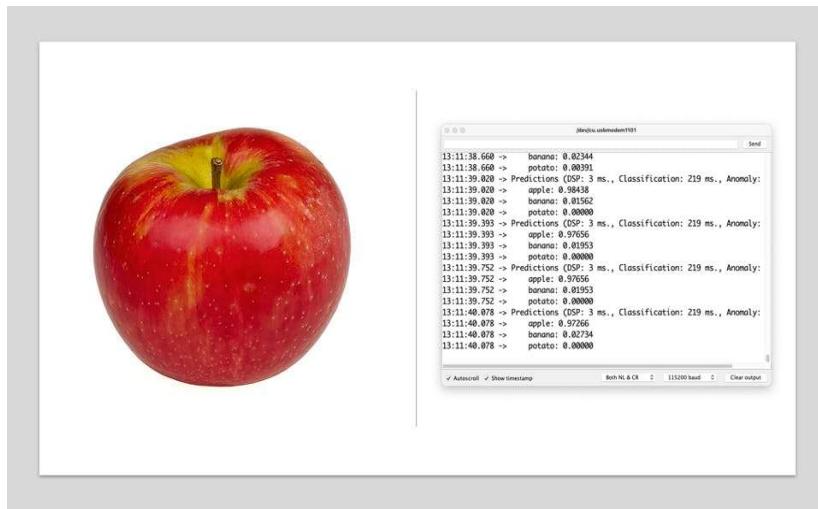
Anche se il nostro modello non ha migliorato la precisione, testiamo se l'XIAO può gestire un modello così grande. Faremo un semplice test di inferenza con lo sketch Static Buffer.

Ridistribuiamo il modello. Se il compilatore EON è abilitato quando si genera la libreria, la memoria totale necessaria per l'inferenza dovrebbe essere ridotta, ma ciò non influisce sulla precisione.

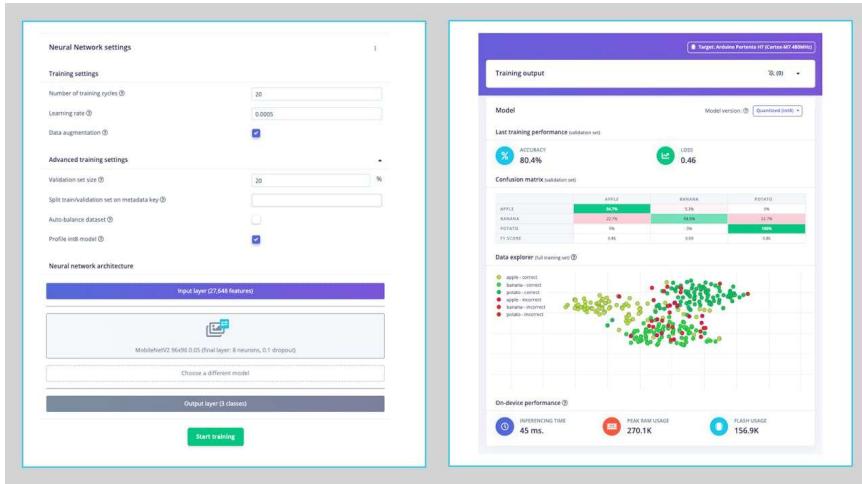
**⚠️ Attenzione** - Xiao ESP32S3 con PSRAM abilitata ha memoria sufficiente per eseguire l'inferenza, anche in un modello così grande. Mantenere il Compilatore EON **NOT ENABLED**.



Facendo un'inferenza con MobilinetV2 96x96 0.35, con immagini RGB in input, la latenza è stata di 219 ms, il che è ottimo per un modello così grande.



Per il test, è stato addestrato nuovamente il modello, utilizzando la versione più piccola di MobileNet V2, con un valore alfa di 0.05. È interessante che il risultato in accuratezza sia stato più alto.



Notare che la latenza stimata per un Arduino Portenta (o Nicla), in esecuzione con un clock di 480 MHz è di 45 ms.

Col deploy del modello, si è ottenuta un'inferenza di soli 135 ms, ricordando che l'XIAO funziona con metà del clock utilizzato da Portenta/Nicla (240 MHz):

```

/dev/cu.usbmodem1101
Send
10:44:47.849 ->    banana: 0.01953
10:44:47.849 ->    potato: 0.12891
10:44:48.103 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.103 ->    apple: 0.86328
10:44:48.103 ->    banana: 0.03906
10:44:48.103 ->    potato: 0.10156
10:44:48.356 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.356 ->    apple: 0.90234
10:44:48.356 ->    banana: 0.02344
10:44:48.356 ->    potato: 0.07422
10:44:48.612 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.612 ->    apple: 0.91797
10:44:48.612 ->    banana: 0.02344
10:44:48.612 ->    potato: 0.05859
10:44:48.861 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.861 ->    apple: 0.88281
10:44:48.861 ->    banana: 0.03516
10:44:48.861 ->    potato: 0.08203
10:44:49.114 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):

```

Both NL & CR 115200 baud Clear output

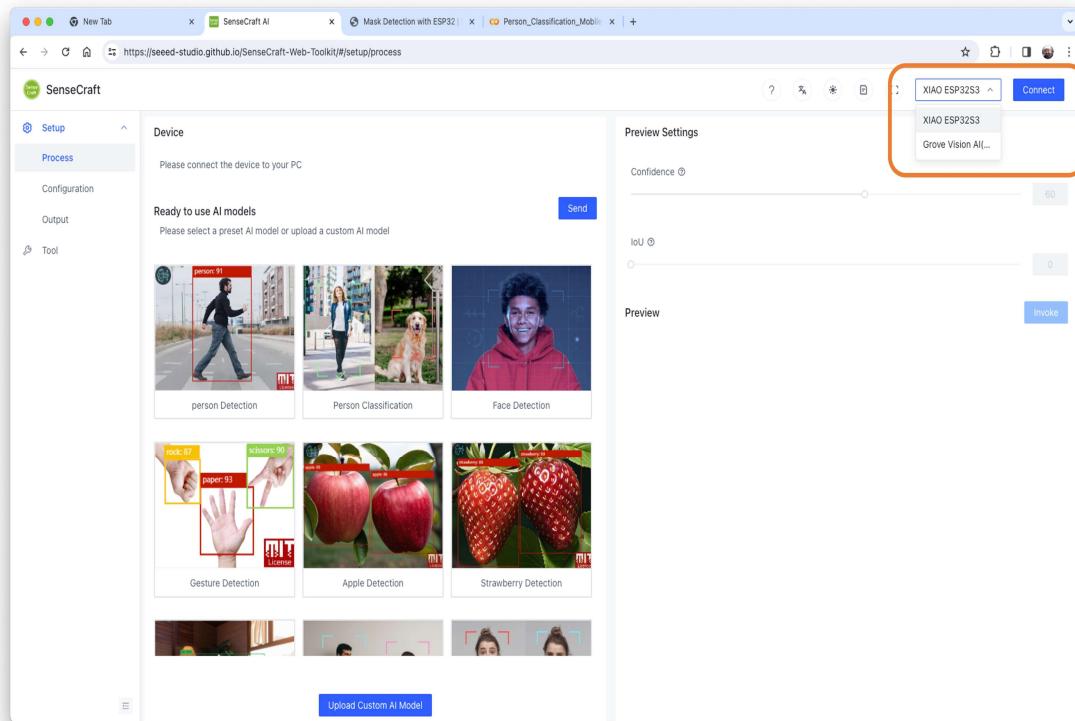
#### 4.4.13 Esecuzione dell'inferenza su SenseCraft-Web-Toolkit

Una limitazione significativa della visualizzazione dell'inferenza su Arduino IDE è che non possiamo vedere su cosa punta la telecamera. Una buona alternativa è **SenseCraft-Web-Toolkit**, uno strumento di distribuzione del modello visivo fornito da **SSCMA**(Seeed SenseCraft Model Assistant). Questo strumento consente di distribuire facilmente modelli su varie piattaforme tramite semplici operazioni. Lo strumento offre un'interfaccia intuitiva e non richiede alcuna codifica.

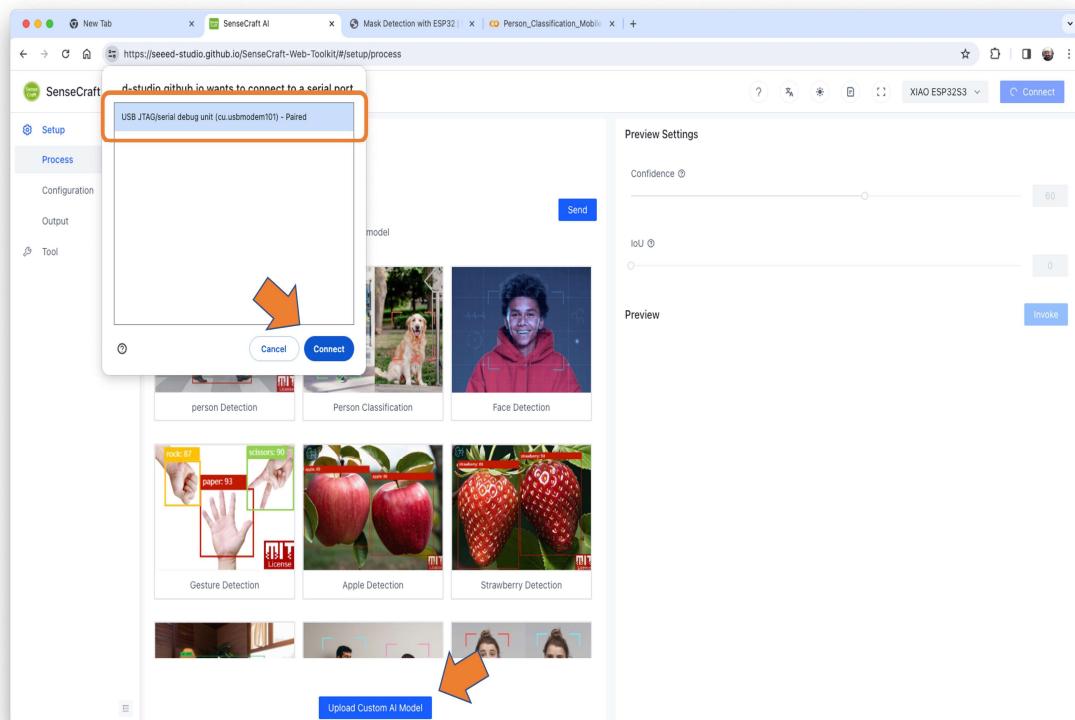
Seguire i seguenti passaggi per avviare SenseCraft-Web-Toolkit:

1. Aprire il sito web di SenseCraft-Web-Toolkit.
2. Collega XIAO al computer:

- Dopo aver collegato XIAO, selezionarlo come di seguito:



- Selezionare il dispositivo/Porta e premere [connect]:



Si possono provare diversi modelli di Computer Vision caricati in precedenza da Seeed Studio. Da provare e verificarli!

Nel nostro caso, useremo il pulsante blu in fondo alla pagina: [Upload Custom AI Model].

Ma prima, dobbiamo scaricare da Edge Impulse Studio il modello **quantized .tflite**.

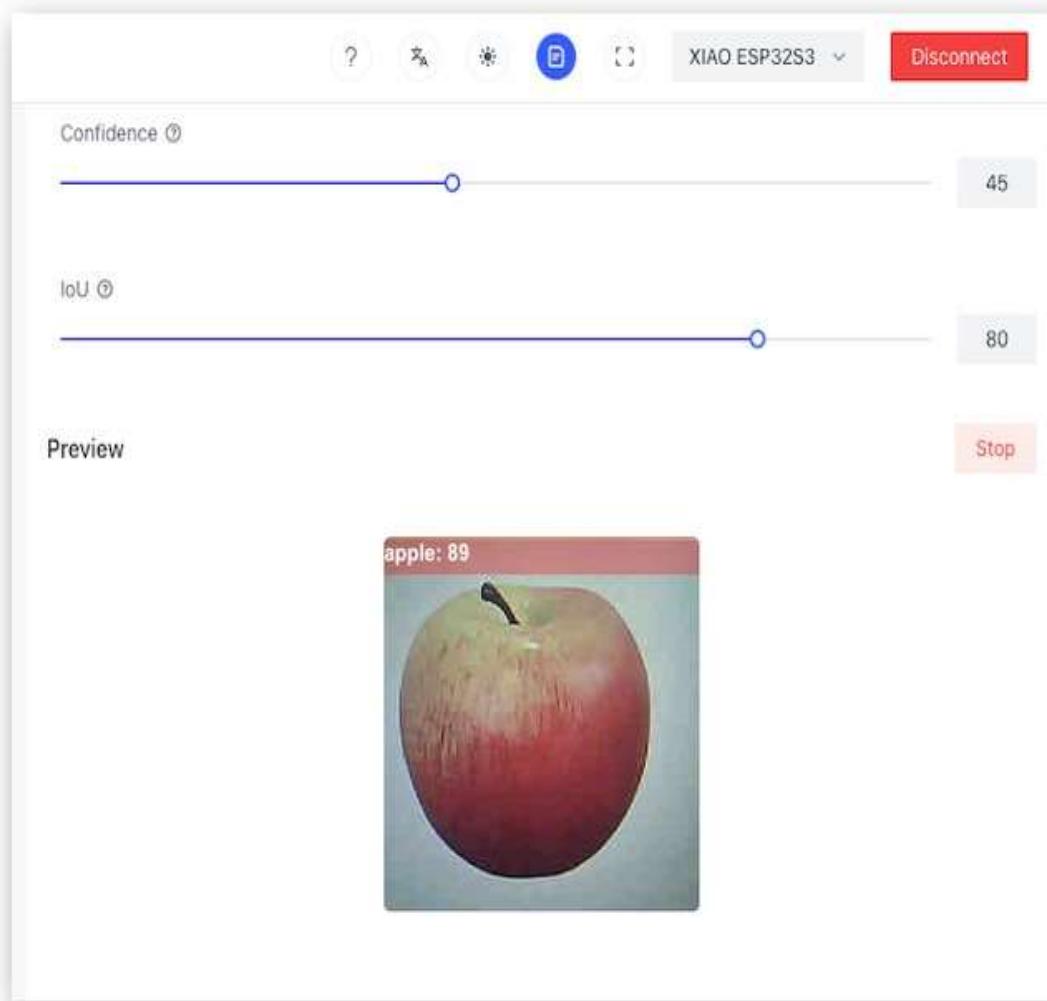
3. Si va sul proprio progetto su Edge Impulse Studio, oppure si clona questo:
  - [XIAO-ESP32S3-CAM-Fruits-vs-Veggies-v1-ESP-NN](#)
4. Nella Dashboard, scaricare il modello (“block output”): **Transfer learning model - TensorFlow Lite (int8 quantized)**.

| Download block output          |                                  |             |
|--------------------------------|----------------------------------|-------------|
| TITLE                          | TYPE                             | SIZE        |
| Image training data            | NPY file                         | 279 windows |
| Image training labels          | NPY file                         | 279 windows |
| Image testing data             | NPY file                         | 15 windows  |
| Image testing labels           | NPY file                         | 15 windows  |
| <b>Transfer learning model</b> | TensorFlow Lite (float32)        | 245 KB      |
| <b>Transfer learning model</b> | TensorFlow Lite (int8 quantized) | 171 KB      |
| Transfer learning model        | TensorFlow SavedModel            | 314 KB      |
| Transfer learning model        | Keras h5 model                   | 248 KB      |

5. Su SenseCraft-Web-Toolkit, usare il pulsante blu in fondo alla pagina: [Upload Custom AI Model]. Si aprirà una finestra. Inserire il file del Modello scaricato sul computer da Edge Impulse Studio, scegliere un nome del modello e inserirlo con le etichette (ID: Object):

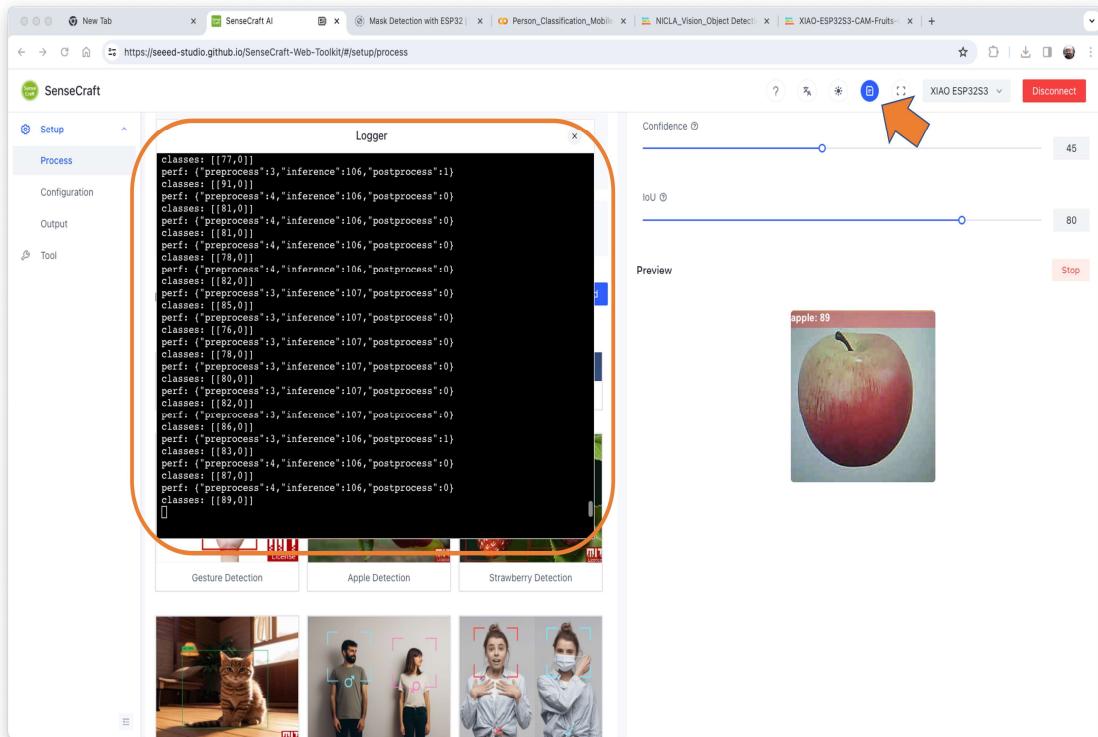
Notare che si devono usare le etichette addestrate su El Studio, inserendole in ordine alfabetico (nel nostro caso: apple, banana, potato).

Dopo alcuni secondi (o minuti), il modello verrà caricato sul dispositivo e l'immagine della telecamera apparirà in tempo reale nel Preview Sector:



Il risultato della classificazione sarà in cima all'immagine. Si può anche selezionare la “Confidence” del cursore di inferenza `confidence`.

Cliccando sul pulsante in alto (Device Log), si può aprire un Serial Monitor per seguire l'inferenza, come fatto con l'IDE Arduino:



Su Device Log, si otterranno informazioni come:

```
perf: {"preprocess":4,"inference":106,"postprocess":0}
classes: [[89,0]]
[]
```

- Tempo di pre-elaborazione (cattura immagine e Crop): 4ms,
- Tempo di inferenza (latenza modello): 106ms,
- Postprocess time (display of the image and inclusion of data): 0ms,
- Tensore di output (classi), ad esempio: [[89,0]]; dove 0 è Apple (e 1 è banana e 2 è patata).

Ecco altri screenshot:



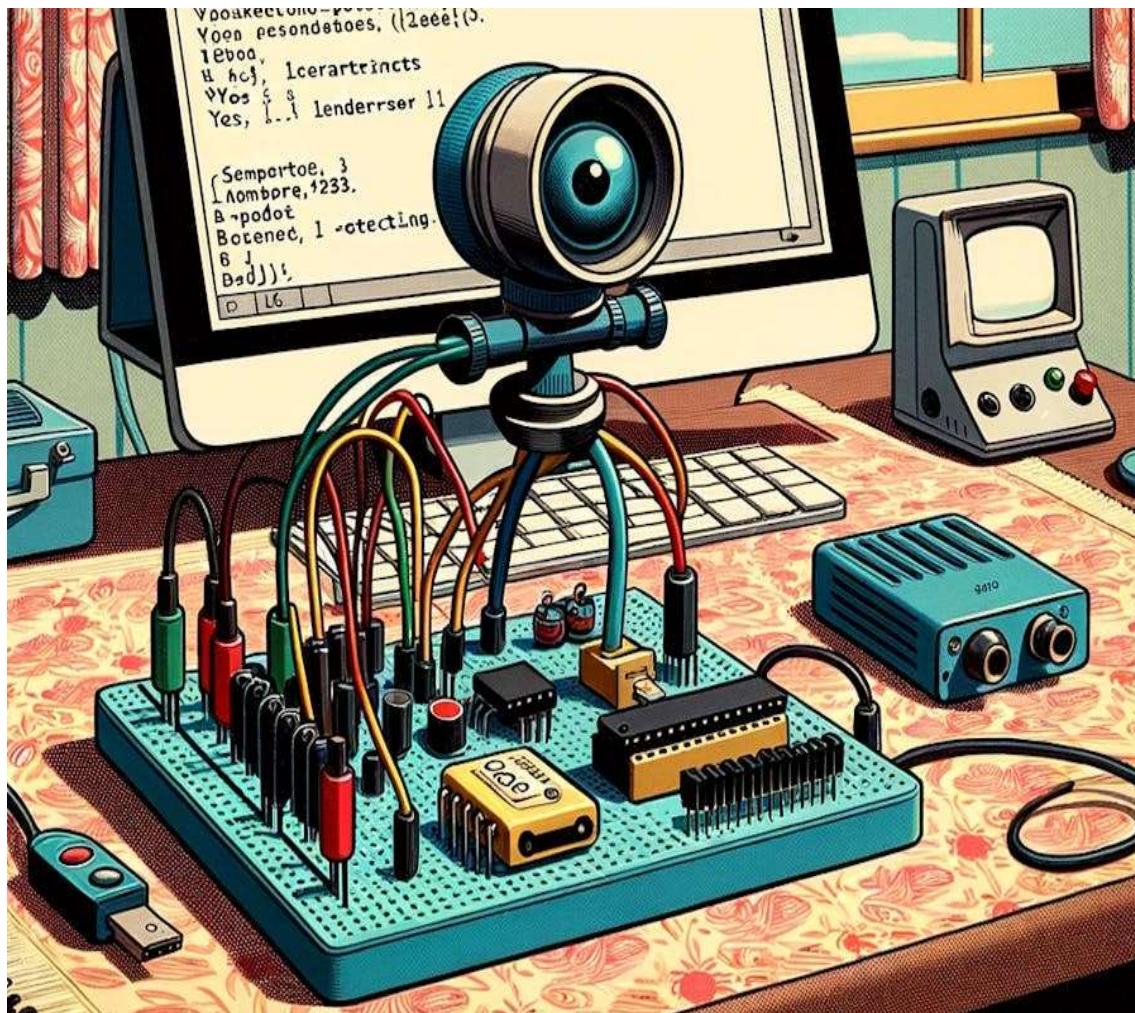
#### 4.4.14 Conclusioni

XIAO ESP32S3 Sense è molto flessibile, poco costoso e facile da programmare. Il progetto dimostra il potenziale di TinyML. La memoria non è un problema; il dispositivo può gestire molte attività di post-elaborazione, tra cui la comunicazione.

L'ultima versione del codice si trova nel repository GitHub: [XIAO-ESP32S3-Sense](#).

### 4.5 Rilevamento degli Oggetti

Questa sezione tratterà altre applicazioni critiche di computer vision, come il rilevamento di oggetti tramite XIAO ESP32S3 Sense, Edge Impulse Studio e Arduino IDE.



#### 4.5.1 Cose utilizzate in questo progetto

##### 4.5.1.1 Componenti hardware

[Seeed Studio Seeed XIAO ESP32S3 Sense x 1](#)

##### 4.5.2 App software e servizi online



- 

Arduino IDE



- Edge Impulse Studio

## 4.5.2 Introduzione

Nell'ultima sezione riguardante Computer Vision (CV) e XIAO ESP32S3, *Classificazione delle immagini*, abbiamo imparato come impostare e classificare le immagini con questa straordinaria scheda di sviluppo. Continuando il nostro viaggio con CV, esploreremo il **Rilevamento degli oggetti** sui microcontrollori.

### 4.5.2.1 Object Detection e Image Classification

Il compito principale con i modelli di Classificazione delle immagini è identificare la categoria di oggetti più probabile presente su un'immagine, ad esempio, per classificare tra un gatto o un cane, gli “oggetti” dominanti in un’immagine:



Cat: 70%



Dog: 80%

Ma cosa succede se non c’è una categoria dominante nell’immagine?

## [PREDICTION]

## [Prob]

|              |       |
|--------------|-------|
| ashcan       | : 27% |
| Egyptian cat | : 19% |
| hamper       | : 13% |

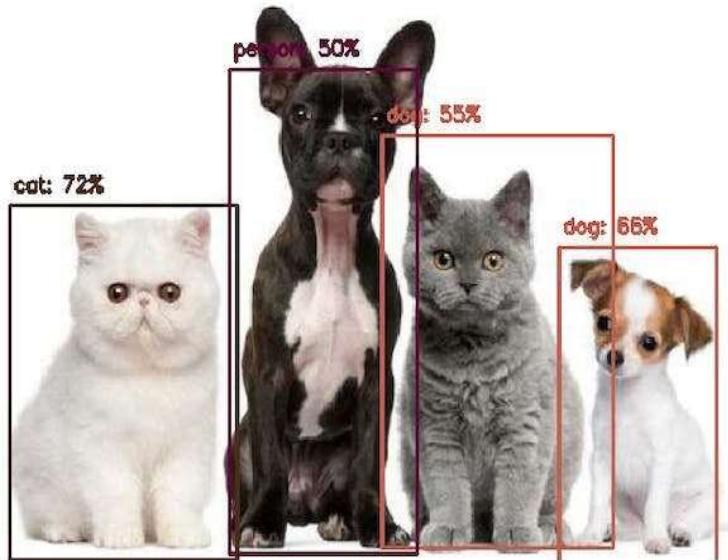


Un modello di classificazione delle immagini identifica l'immagine soprastante in modo completamente sbagliato come un “ashcan”, probabilmente a causa delle tonalità di colore.

Il modello utilizzato nell'esempio precedente è *MobileNet*, addestrato con un ampio set di dati, *ImageNet*, in esecuzione su un Raspberry Pi.

Per risolvere questo problema, abbiamo bisogno di un altro tipo di modello, in cui non solo possono essere trovate **più categorie** (o etichette), ma anche **dove** si trovano gli oggetti in una determinata immagine.

Come possiamo immaginare, tali modelli sono molto più complicati e più grandi, ad esempio, **MobileNetV2 SSD FPN-Lite 320x320, addestrato con il set di dati COCO**. Questo modello di rilevamento degli oggetti pre-addestrato è progettato per individuare fino a 10 oggetti all'interno di un'immagine, generando un riquadro di delimitazione per ogni oggetto rilevato. L'immagine sottostante è il risultato di un tale modello in esecuzione su un Raspberry Pi:



I modelli utilizzati per il rilevamento di oggetti (come MobileNet SSD o YOLO) hanno solitamente dimensioni di diversi MB, il che è accettabile per l'uso con Raspberry Pi ma non è adatto per l'uso con dispositivi embedded, in cui la RAM è solitamente inferiore a 1 Mbyte o almeno di alcuni MB, come nel caso di XIAO ESP32S3.

#### 4.5.2.2 Una soluzione innovativa per il Rilevamento di Oggetti: FOMO

Edge Impulse ha lanciato nel 2022, [FOMO \(Faster Objects, More Objects\)](#), una nuova soluzione per eseguire il rilevamento di oggetti su dispositivi embedded, come Nicla Vision e Portenta (Cortex M7), su CPU Cortex M4F (serie Arduino Nano33 e OpenMV M4) e sui dispositivi Espressif ESP32 (ESP-CAM, ESP-EYE e XIAO ESP32S3 Sense).

In questo progetto pratico, esploreremo l'Object Detection utilizzando FOMO.

Per saperne di più sulla FOMO, si può leggere l'[annuncio ufficiale su FOMO](#) di Edge Impulse, dove Louis Moreau e Mat Kelcey spiegano in dettaglio come funziona.

#### 4.5.3 Obiettivo del Progetto di Object Detection

Tutti i progetti di apprendimento automatico devono iniziare con un obiettivo dettagliato. Supponiamo di trovarci in una struttura industriale o rurale e di dover smistare e contare arance (frutti) e in particolare rane (insetti).



In altre parole, dovremmo eseguire una classificazione multi-etichetta, in cui ogni immagine può avere tre classi:

- Background [Sfondo] (nessun oggetto)
- Fruit
- Bug

Ecco alcuni esempi di immagini non etichettate che dovremmo utilizzare per rilevare gli oggetti (frutti e insetti):



Siamo interessati a quale oggetto è presente nell'immagine, alla sua posizione (centroide) e a quanti ne possiamo trovare su di essa. La dimensione dell'oggetto non viene rilevata con

FOMO, come con MobileNet SSD o YOLO, in cui il Bounding Box è uno degli output del modello.

Svilupperemo il progetto utilizzando XIAO ESP32S3 per l'acquisizione di immagini e l'inferenza del modello. Il progetto ML verrà sviluppato utilizzando Edge Impulse Studio. Ma prima di iniziare il progetto di "object detection" in Studio, creiamo un *dataset grezzo* (non etichettato) con immagini che contengono gli oggetti da rilevare.

#### 4.5.4 Raccolta Dati

Si può usare XIAO, il telefono o altri dispositivi per catturare l'immagine. Qui, useremo XIAO con un codice nella libreria ESP32.

##### 4.5.4.1 Raccolta di Dataset con XIAO ESP32S3

Aprire Arduino IDE e selezionare la scheda XIAO\_ESP32S3 (e la porta a cui è collegata). Su

**File > Examples > ESP32 > Camera, Scegliere CameraWebServer.**

Nel pannello BOARDS MANAGER, confermare di aver installato l'ultimo pacchetto "stable".

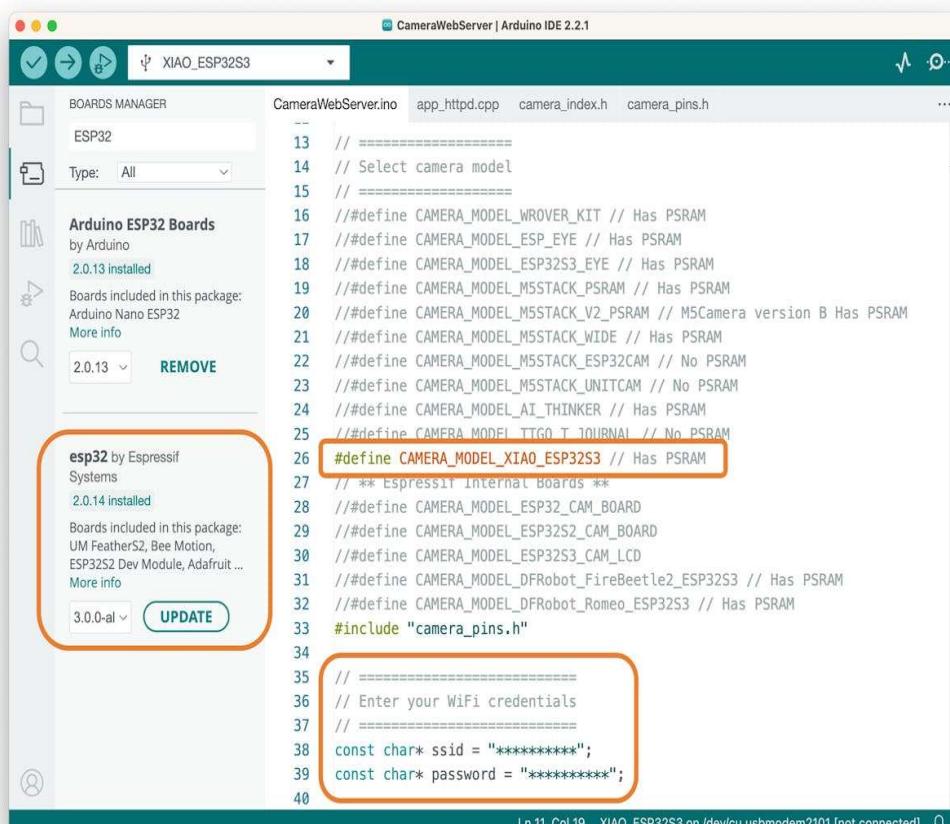
##### ⚠ Attenzione

Le versioni Alpha (ad esempio, 3.x-alpha) non funzionano correttamente con XIAO ed Edge Impulse. Utilizzare invece l'ultima versione stabile (ad esempio, 2.0.11).

Si devono anche commentare tutti i modelli di fotocamere, eccetto i pin del modello XIAO:

```
#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
```

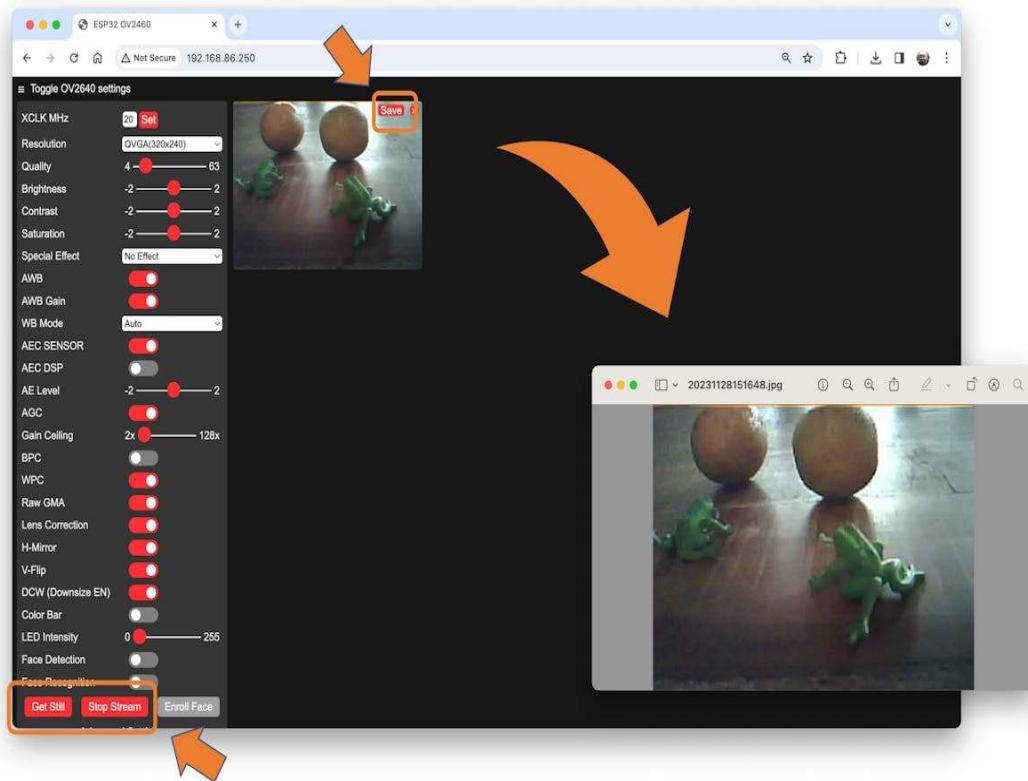
E su tools, abilitare la PSRAM. Inserisci le credenziali wifi e caricare il codice sul dispositivo:



Se il codice viene eseguito correttamente, si vedrà l'indirizzo sul monitor seriale:



Copiare l'indirizzo sul browser e attendere che la pagina venga caricata. Selezionare la risoluzione della telecamera (ad esempio, QVGA) e selezionare [START STREAM]. Attendi qualche secondo/minuto, a seconda della connessione. Si può salvare un'immagine nell'area download del computer usando il pulsante [Save].



Edge impulse suggerisce che gli oggetti dovrebbero essere di dimensioni simili e non sovrapposti per prestazioni migliori. Questo va bene in una struttura industriale, dove la telecamera dovrebbe essere fissa, mantenendo la stessa distanza dagli oggetti da rilevare. Nonostante ciò, proveremo anche a usare dimensioni e posizioni miste per vedere il risultato.

Non abbiamo bisogno di creare cartelle separate per le nostre immagini perché ognuna contiene più etichette.

Suggeriamo circa 50 immagini che mescolano gli oggetti e variano il numero di ciascuno che appare sulla scena. Provare ad acquisire con diverse angolazioni, sfondi e condizioni di luce.

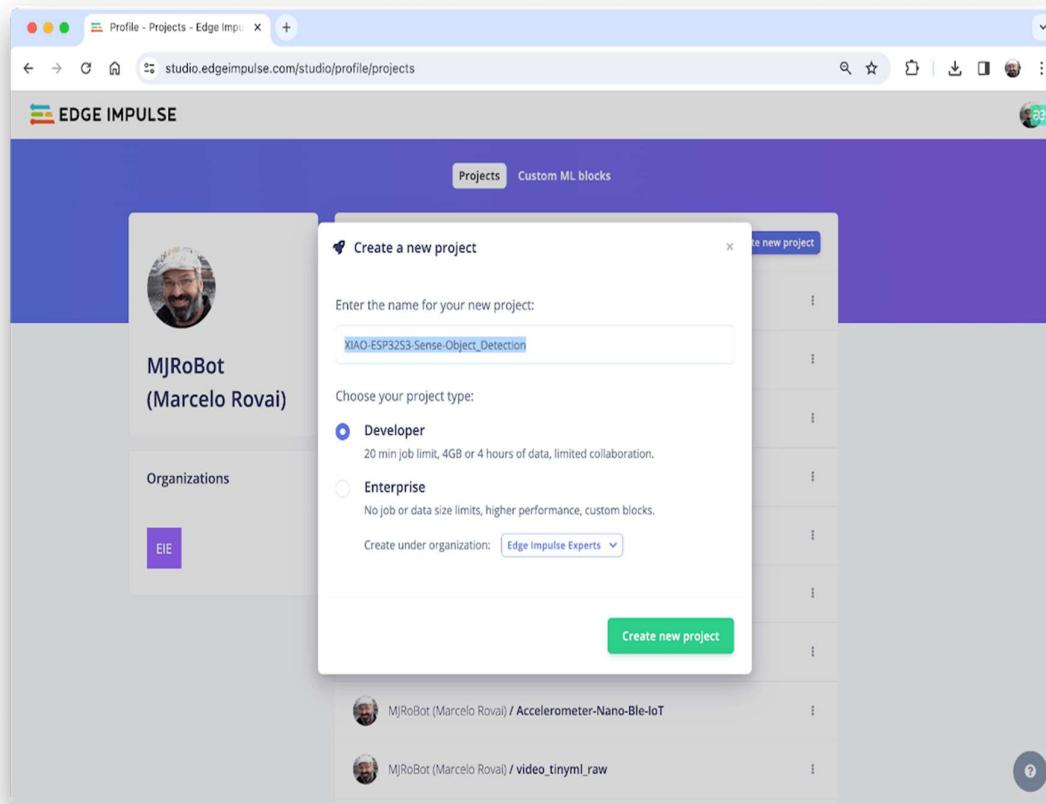
Le immagini memorizzate utilizzano una dimensione del fotogramma QVGA 320x240 e RGB565 (formato pixel a colori).

Dopo aver acquisito il set di dati, [Stop Stream] e spostare le immagini in una cartella.

## 4.5.4.2 Edge Impulse Studio

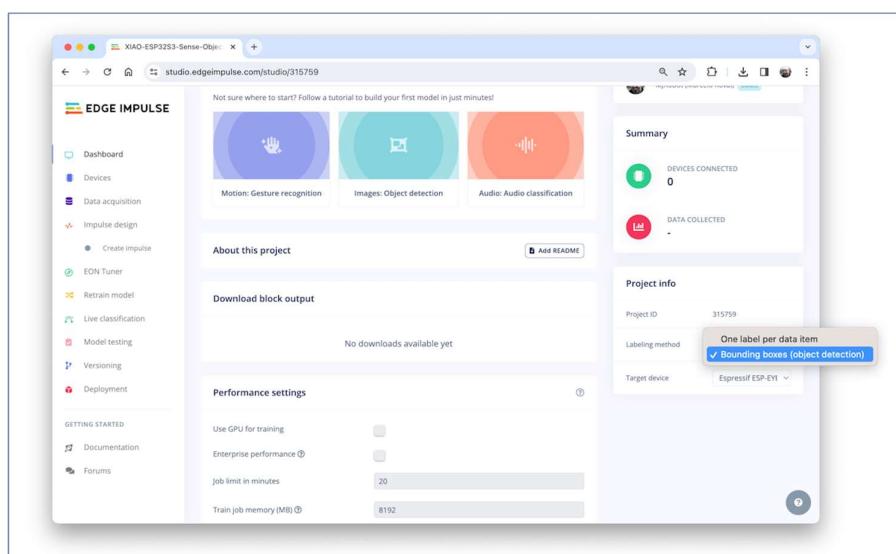
### 4.5.4.2.1 Setup del progetto

Si va su [Edge Impulse Studio](#), si inseriscono le proprie credenziali in **Login** (o si crea un account) e si avvia un nuovo progetto.



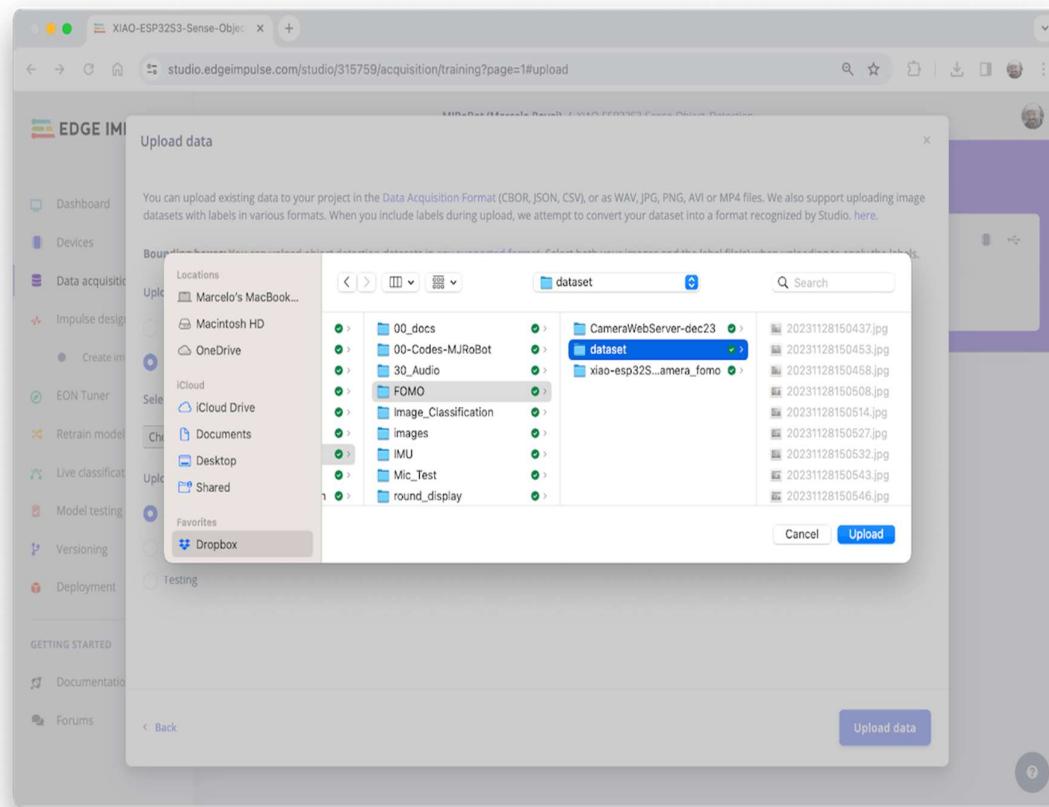
Qui, è possibile clonare il progetto sviluppato per questa esercitazione pratica:  
[XIAO-ESP32S3-Sense-Object\\_Detection](#)

Nella dashboard del progetto, andare in basso e su **Project info** e selezionare **Bounding boxes (object detection)** e **Espressif ESP-EYE** (il più simile alla nostra scheda) come Target Device:



### 4.5.4.3 Caricamento dei dati non etichettati

In Studio, si va alla scheda **Data acquisition** e nella sezione **UPLOAD DATA** caricare i file acquisiti come cartella dal computer.



Si può lasciare che Studio divida automaticamente i dati tra “Train” e “Test” o farlo manualmente. Caricheremo tutti come training.

| SAMPLE NAME    | LABELS | ADDED           | LENGTH |
|----------------|--------|-----------------|--------|
| 20231128151645 | -      | Today, 15:27:09 | 1      |
| 20231128150613 | -      | Today, 15:27:09 | 1      |
| 20231128150604 | -      | Today, 15:27:09 | 1      |
| 20231128150833 | -      | Today, 15:27:09 | 1      |
| 20231128150600 | -      | Today, 15:27:09 | 1      |
| 20231128150855 | -      | Today, 15:27:09 | 1      |
| 20231128150458 | -      | Today, 15:27:09 | 1      |
| 20231128150713 | -      | Today, 15:27:09 | 1      |
| 20231128150908 | -      | Today, 15:27:09 | 1      |

Tutte le immagini non etichettate (47) sono state caricate, ma devono essere etichettate in modo appropriato prima di essere utilizzate come dataset del progetto. Studio ha uno strumento per questo scopo, che si trova nel link Labeling queue (47).

Ci sono due modi per eseguire l'etichettatura assistita dall'IA su Edge Impulse Studio (versione gratuita):

- Utilizzando yolov5
- Tracciando di oggetti tra i frame

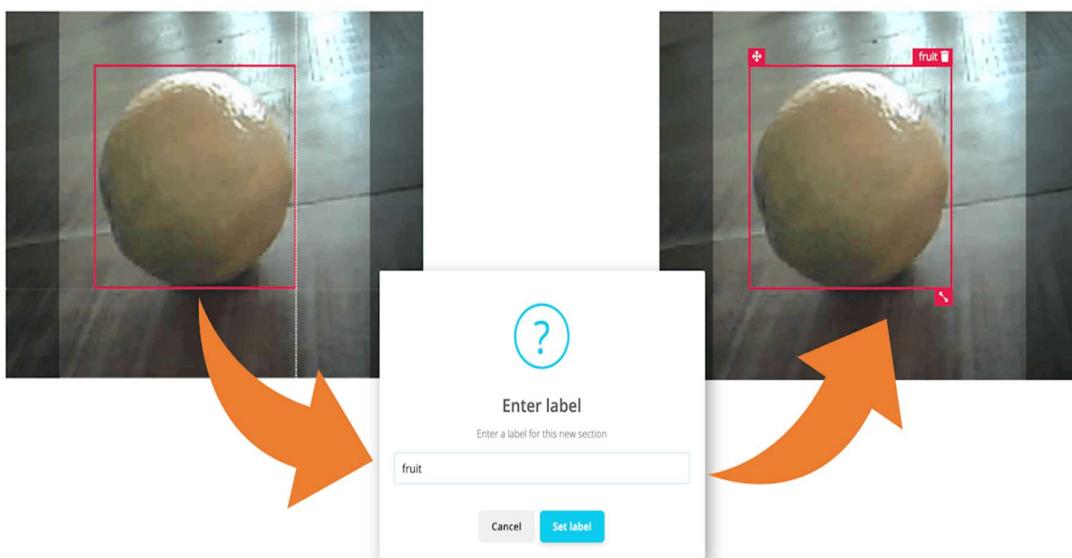
Edge Impulse ha lanciato una [funzione di auto-labeling](#) per i clienti Enterprise, semplificando le attività di etichettatura nei progetti di rilevamento degli oggetti.

Gli oggetti ordinari possono essere rapidamente identificati ed etichettati utilizzando una libreria esistente di modelli di rilevamento degli oggetti pre-addestrati da YOLOv5 (addestrati con il set di dati COCO). Ma poiché, nel nostro caso, gli oggetti non fanno parte dei dataset COCO, dovremmo selezionare l'opzione di “tracking” [tracciamento] degli oggetti. Con questa opzione, una volta disegnati i bounding box ed etichettate le immagini in un frame, gli oggetti verranno tracciati automaticamente da un frame all'altro, etichettando *partially* quelli nuovi (non tutti sono etichettati correttamente).

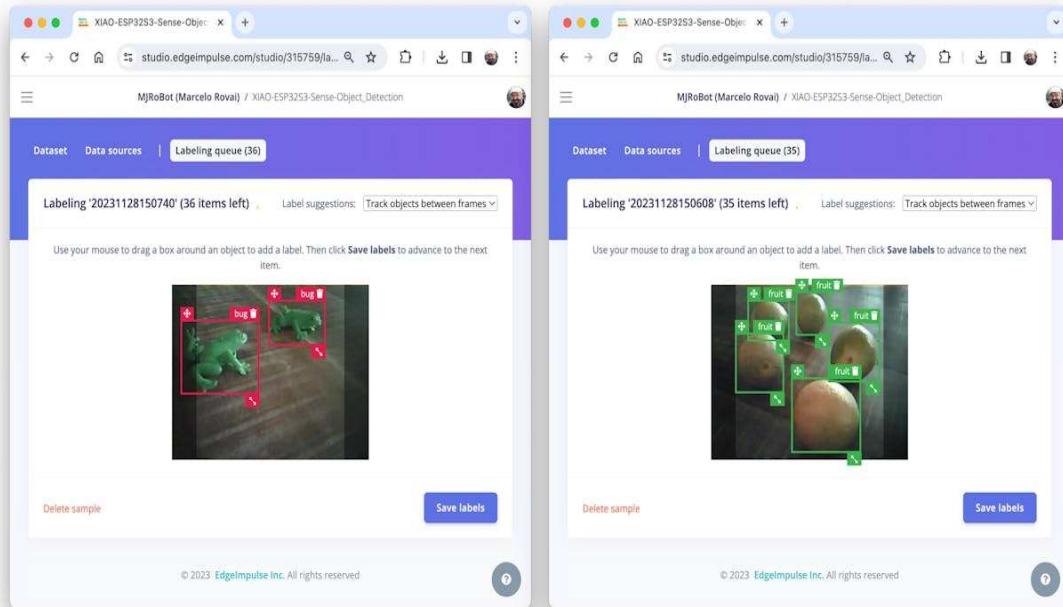
Si può usare [EI uploader](#) per importare i dati se si ha già un dataset etichettato contenente dei “bounding box”.

#### 4.5.4.4 Etichettatura del Dataset

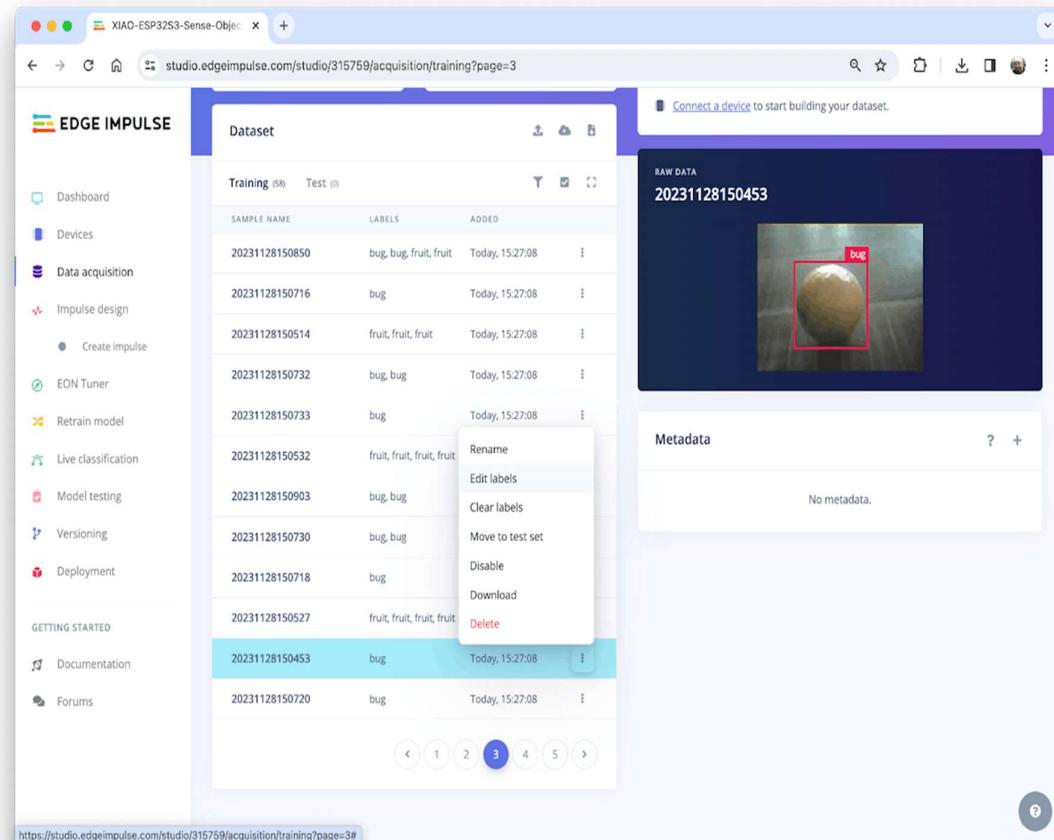
Iniziando dalla prima immagine dei dati non etichettati, si usa il mouse per trascinare una casella attorno a un oggetto per aggiungere un'etichetta. Poi si clicca su **Save labels** per passare all'elemento successivo.



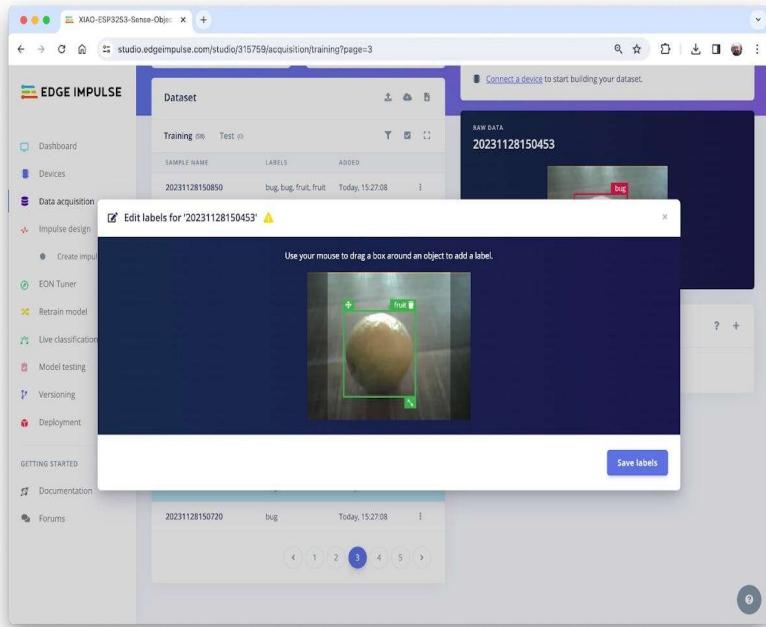
Si continua con questo processo finché la coda non è vuota. Alla fine, tutte le immagini dovrebbero avere gli oggetti etichettati come i campioni sottostanti:



Poi, si esaminano i campioni etichettati nella scheda **Data acquisition**. Se una delle etichette è errata, la si può modificarla utilizzando il menù *tre puntini* dopo il nome del campione:

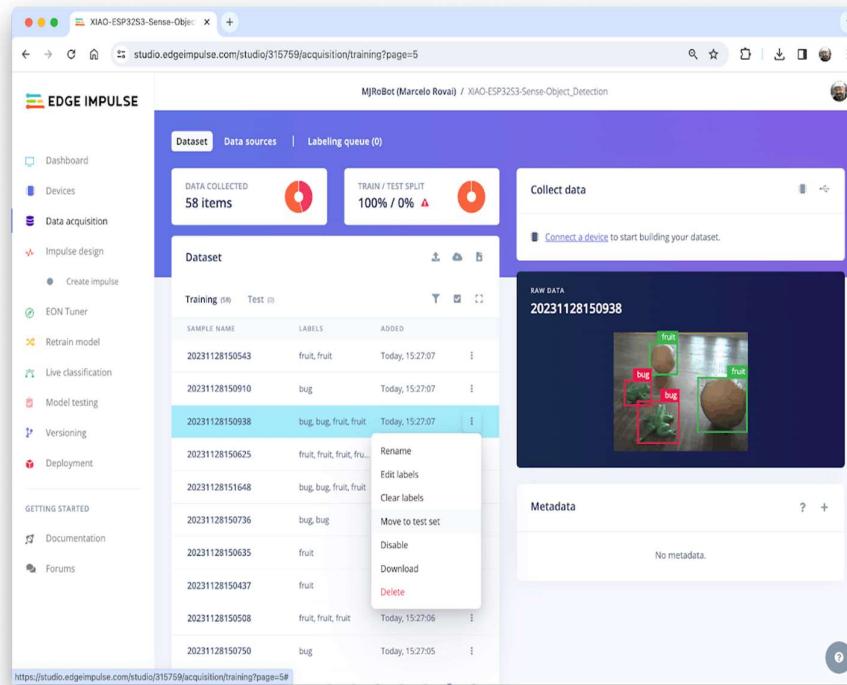


Si verrà guidati a sostituire l'etichetta errata e a correggere il dataset.



#### 4.5.4.5 Bilanciamento del dataset e suddivisione Train/Test

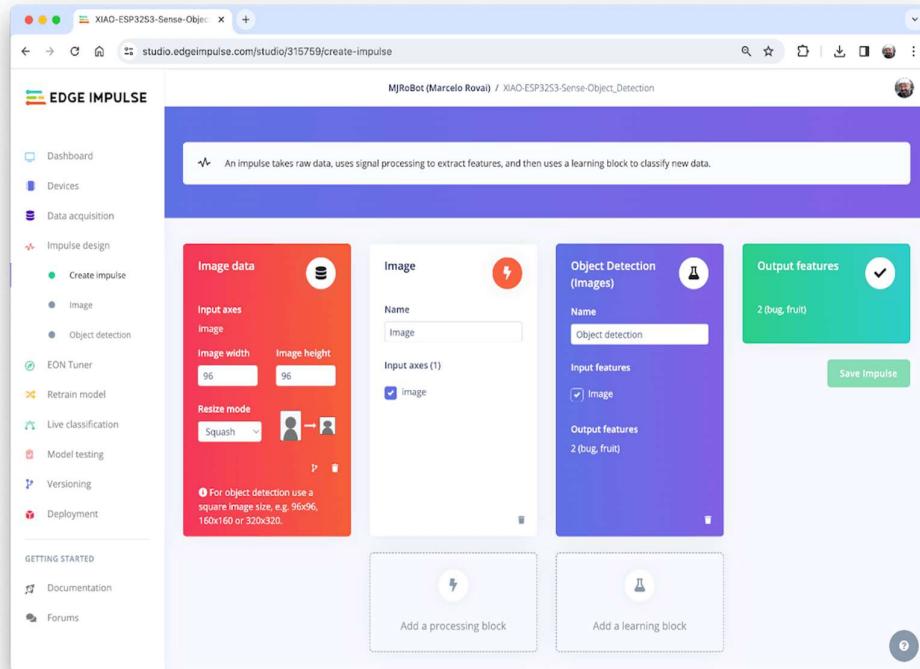
Dopo aver etichettato tutti i dati, ci siamo resi conto che la classe fruit aveva molti più campioni di bug. Quindi, sono state raccolte 11 immagini di bug nuove e aggiuntive (per un totale di 58 immagini). Dopo averle etichettate, è il momento di selezionare alcune immagini e spostarle nel dataset di test. Per farlo si usa il menu a tre punti dopo il nome dell'immagine. Sono state selezionate sei immagini, che rappresentano il 13% del set di dati totale.



#### 4.5.5 Impulse Design

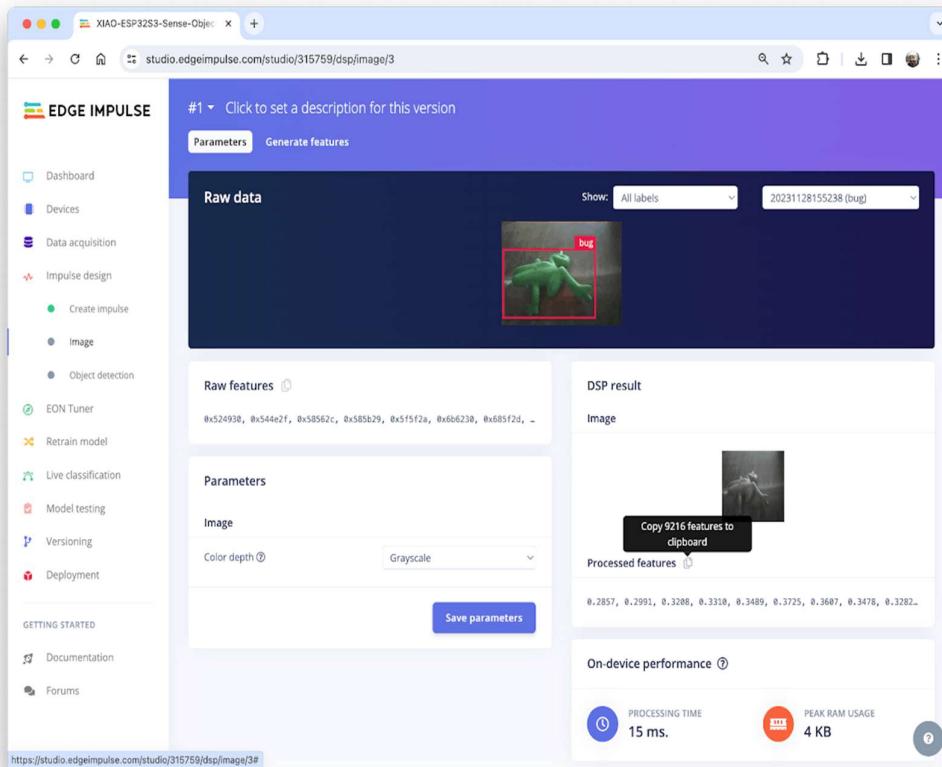
In questa fase, si deve definire come:

- Il **Pre-processing** consiste nel ridimensionare le singole immagini da 320 x 240 a 96 x 96 e nel ridurle (forma quadrata, senza ritaglio). In seguito, le immagini vengono convertite da RGB a scala di grigi.
- **Design a Model**, in questo caso, “Object Detection”.

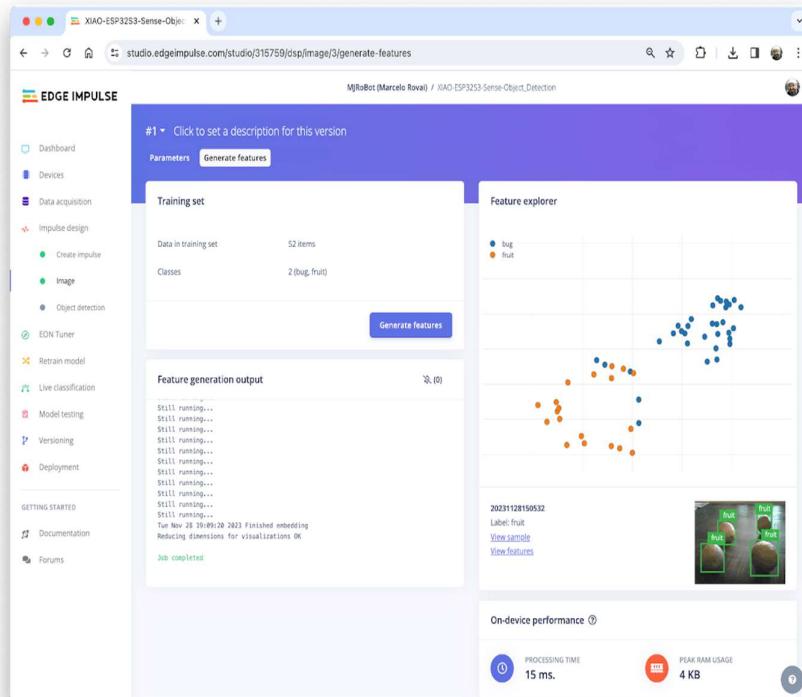


#### 4.5.5.1 Pre-elaborazione di tutti i dataset

In questa sezione, selezionare **Color depth** come Grayscale, adatta per l'uso con modelli FOMO ed eseguire il “Save” dei parametri.



Studio passa automaticamente alla sezione successiva, “Generate features”, in cui tutti i campioni verranno pre-elaborati, generando un set di dati con singole immagini 96x96x1 o 9.216 feature.



L'esploratore di feature mostra che tutti i campioni evidenziano una buona separazione dopo la generazione delle feature.

Alcuni campioni sembrano stare nello spazio sbagliato, ma cliccandoci sopra si conferma la corretta etichettatura.

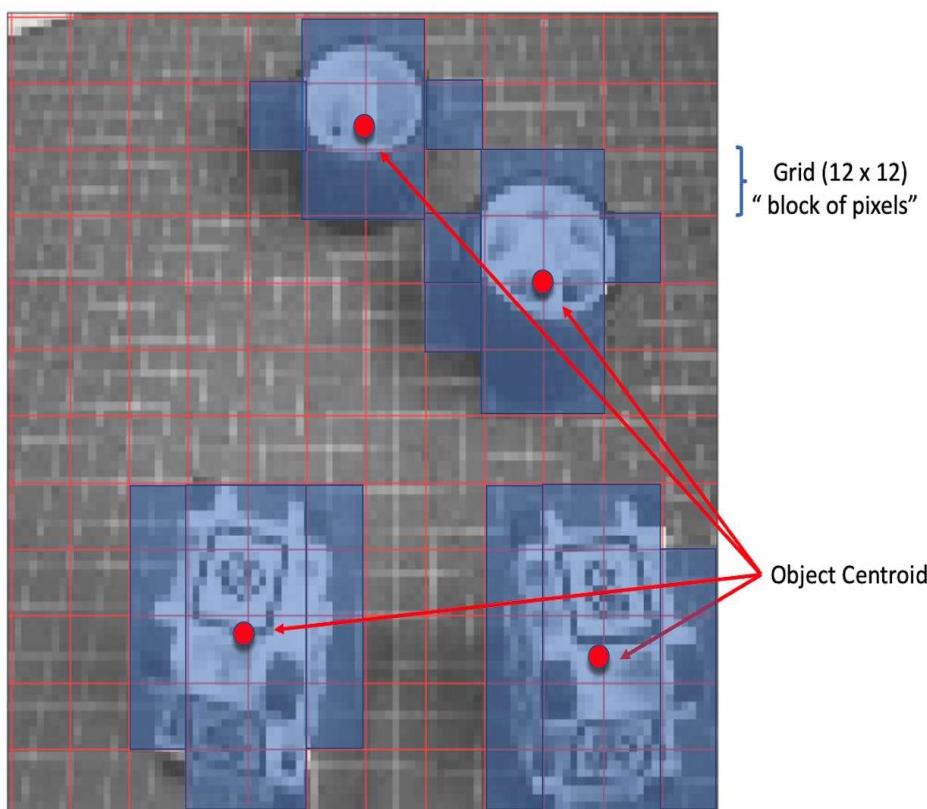
#### 4.5.6 Progettazione, Addestramento e Test del Modello

Useremo FOMO, un modello di rilevamento degli oggetti basato su MobileNetV2 (alpha 0.35) progettato per segmentare grossolanamente un'immagine in una griglia di **background** rispetto a **oggetti di interesse** (in questo caso, scatole e ruote).

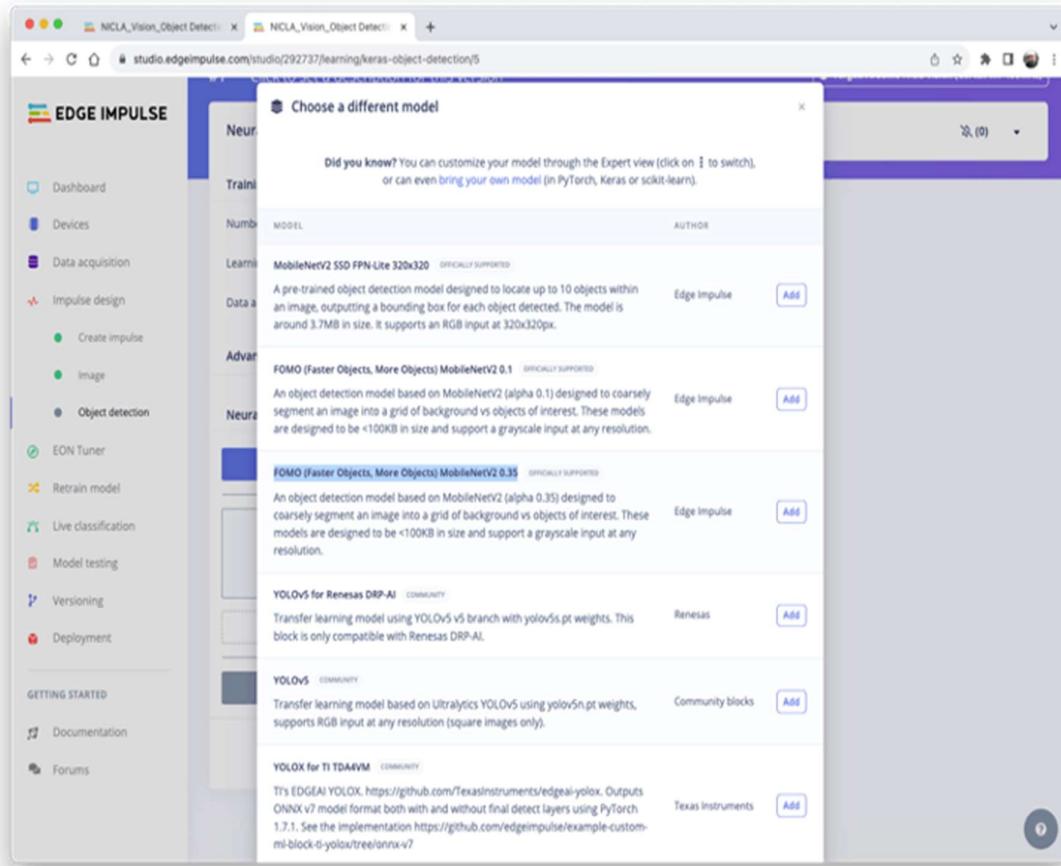
FOMO è un modello di apprendimento automatico innovativo per il rilevamento degli oggetti, che può utilizzare fino a 30 volte meno energia e memoria rispetto ai modelli tradizionali come Mobilenet SSD e YOLOv5. FOMO può funzionare su microcontrollori con meno di 200 KB di RAM. Il motivo principale per cui ciò è possibile è che mentre altri modelli calcolano le dimensioni dell'oggetto disegnando un quadrato attorno ad esso (bounding box), FOMO ignora le dimensioni dell'immagine, fornendo solo le informazioni su dove si trova l'oggetto nell'immagine tramite le sue coordinate del centroide.

##### Come funziona FOMO?

FOMO prende l'immagine in scala di grigi e la divide in blocchi di pixel usando un fattore di 8. Per l'input di 96x96, la griglia è 12x12 ( $96/8=12$ ). Successivamente, FOMO eseguirà un classificatore attraverso ogni blocco di pixel per calcolare la probabilità che ci sia un box o una ruota in ognuno di essi e, successivamente, determinerà le regioni che hanno la più alta probabilità di contenere l'oggetto (se un blocco di pixel non ha oggetti, verrà classificato come *background*). Dalla sovrapposizione della regione finale, FOMO fornisce le coordinate (relative alle dimensioni dell'immagine) del centroide di questa regione.



Per l'addestramento, dovremmo selezionare un modello pre-addestrato. Usiamo **FOMO (Faster Objects, More Objects) MobileNetV2 0.35**. Questo modello utilizza circa 250 KB di RAM e 80 KB di ROM (Flash), che si adatta bene alla nostra scheda.



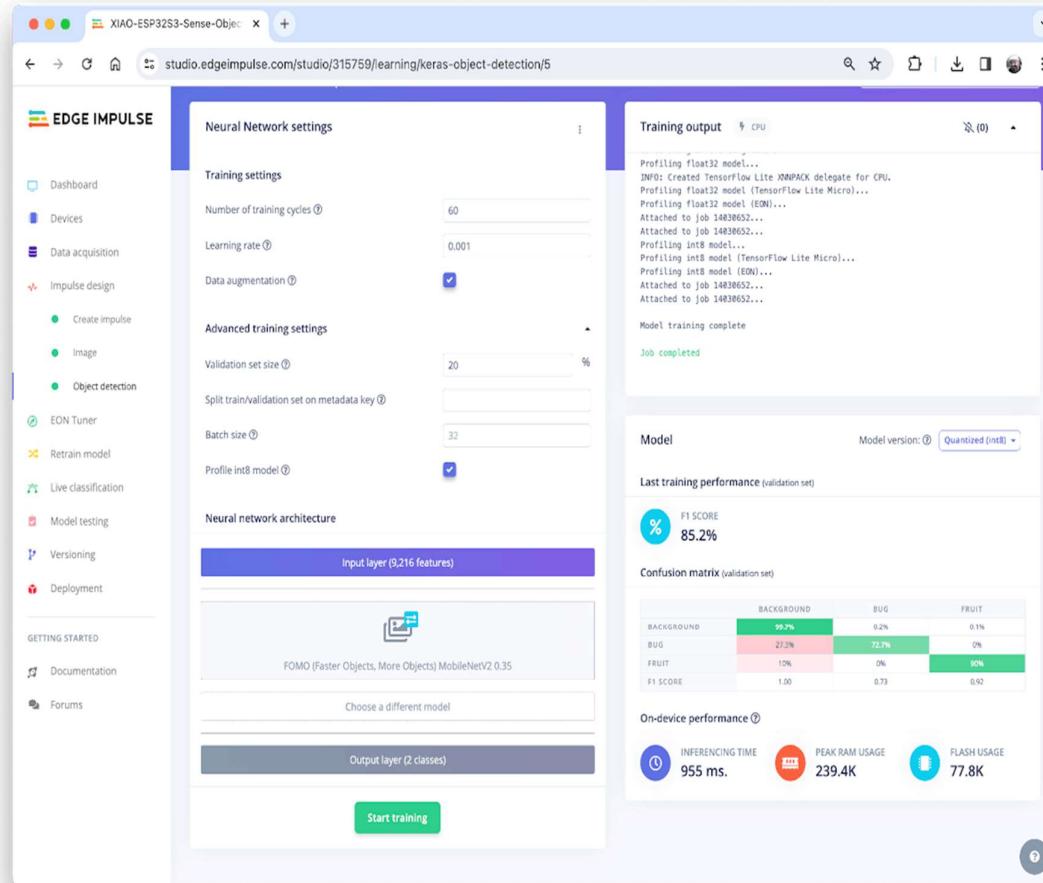
Per quanto riguarda gli iperparametri di training, il modello verrà addestrato con:

- Epochs: 60
- Batch size: 32
- Learning Rate: 0.001.

Per la convalida durante l'addestramento, il 20% del set di dati (*validation\_dataset*) verrà risparmiato. Per il restante 80% (*train\_dataset*), applicheremo il “Data Augmentation”, che capovolgerà casualmente, cambierà le dimensioni e la luminosità dell'immagine e le ritaglierà, aumentando artificialmente il numero di campioni sul set di dati per l'addestramento.

Di conseguenza, il modello termina con un punteggio F1 complessivo dell'85%, simile al risultato ottenuto utilizzando i dati di prova (83%).

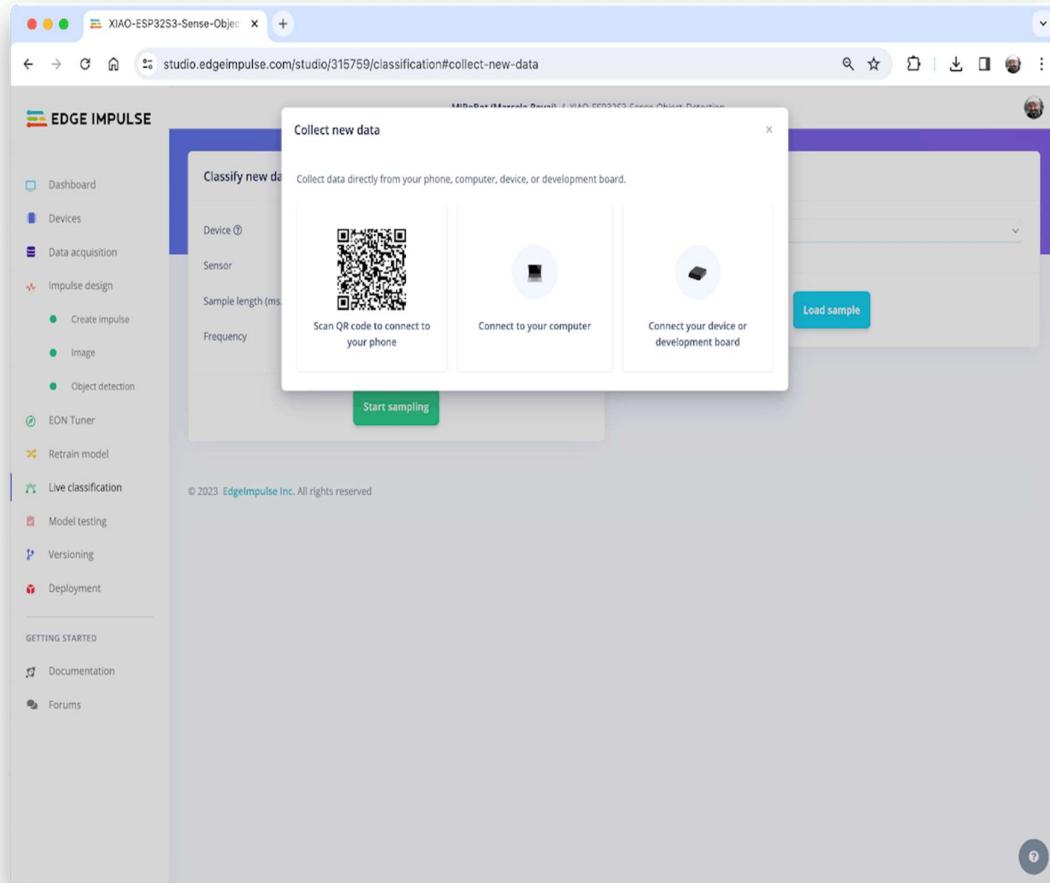
Notare che FOMO ha aggiunto automaticamente una terza etichetta di background [sfondo] ai due precedentemente definiti (*box* e *wheel*).



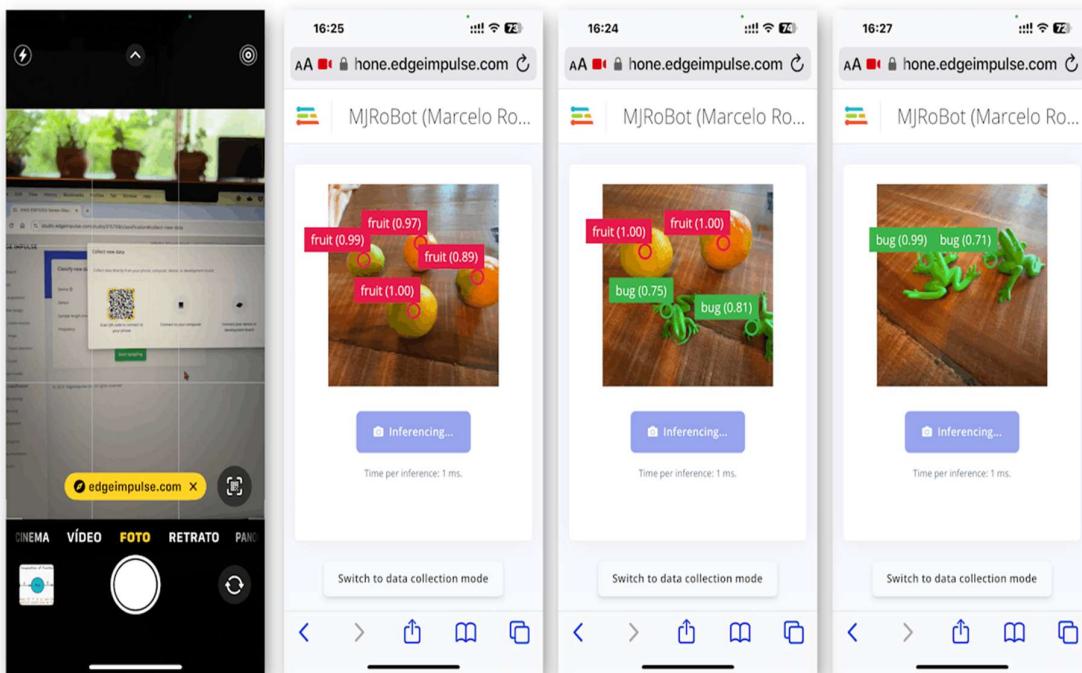
Nelle attività di rilevamento di oggetti, l'accuratezza non è in genere la **metrica di valutazione primaria**. Il rilevamento di oggetti comporta la classificazione degli oggetti e la definizione di riquadri di delimitazione attorno a essi, il che lo rende un problema più complesso della semplice classificazione. Il problema è che non abbiamo il riquadro di delimitazione, solo i centroidi. In breve, usare l'accuratezza come metrica potrebbe essere fuorviante e potrebbe non fornire una comprensione completa delle prestazioni del modello. Per questo motivo, useremo il punteggio F1.

#### 4.5.6.1 Modello di test con “Live Classification”

Una volta addestrato il nostro modello, possiamo testarlo utilizzando lo strumento Live Classification. Nella sezione corrispondente, cliccare sull’icona “Connect a development board” (una piccola MCU) e scansionare il codice QR col telefono.



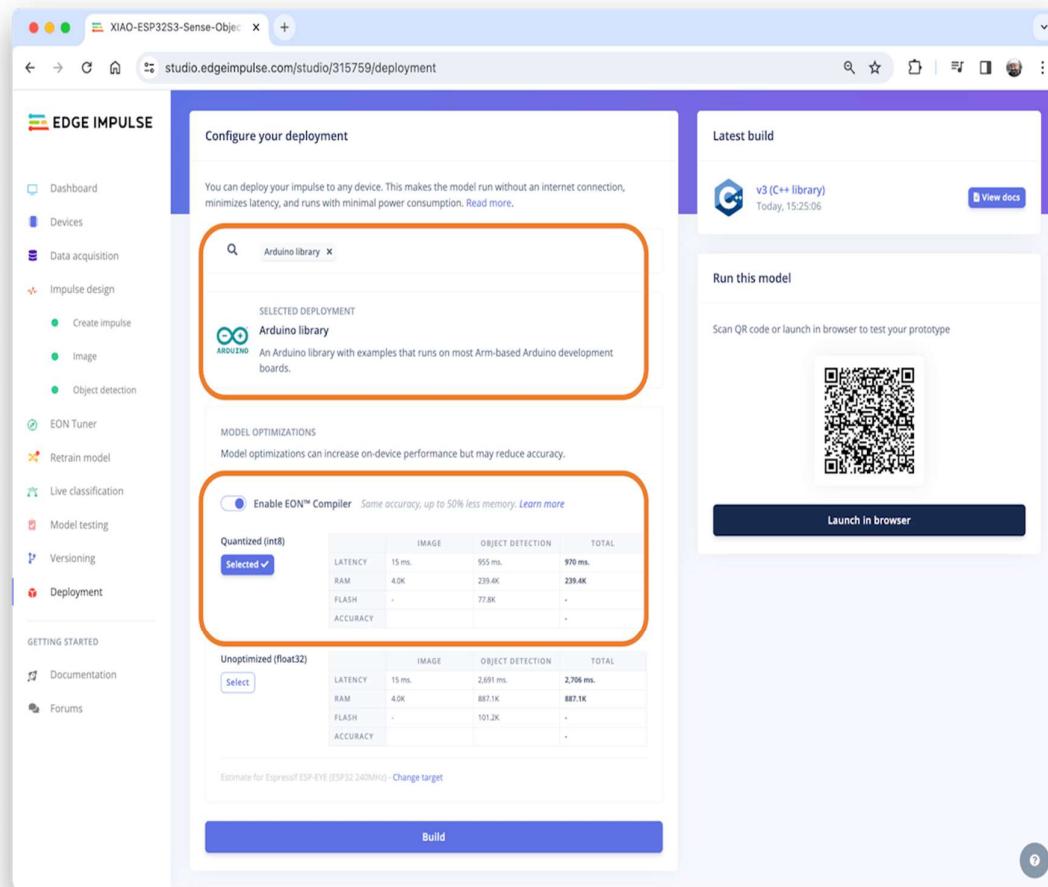
Una volta connesso, si può usare lo smartphone per catturare immagini reali da testare col modello addestrato su Edge Impulse Studio.



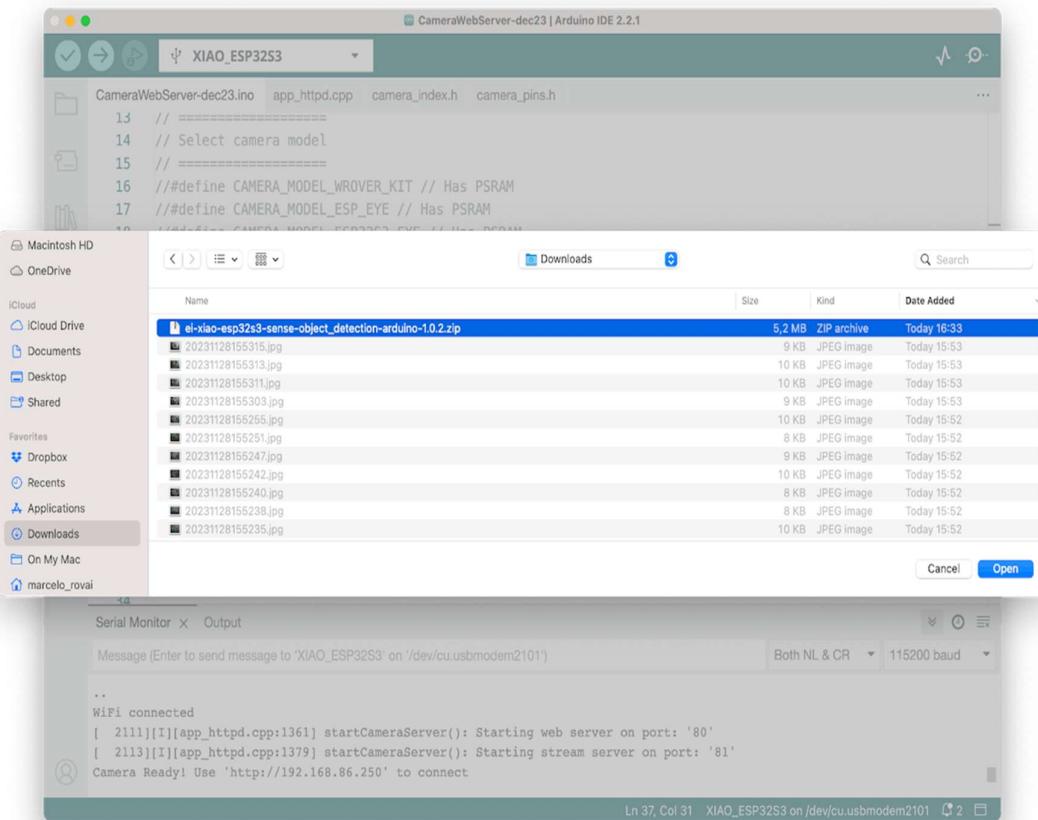
Una cosa da notare è che il modello può produrre falsi positivi e falsi negativi. Questo può essere ridotto al minimo definendo una “Confidence Threshold” appropriata (usare il menu “Tre puntini” per la configurazione). Provare con 0,8 o più.

### 4.5.7 Deploying del Modello (Arduino IDE)

Selezionare la Libreria Arduino e il modello Quantized (int8), abilitare il compilatore EON nella scheda Deploy e premere [Build].



Apri l'Arduino IDE e, in Sketch, andare su Include Library e aggiungere .ZIP Library..  
Selezionare il file che scaricato da Edge Impulse Studio e il gioco è fatto!



Nella scheda Examples su Arduino IDE, si trova il codice di uno sketch (`esp32 > esp32_camera`) sotto il nome del progetto.

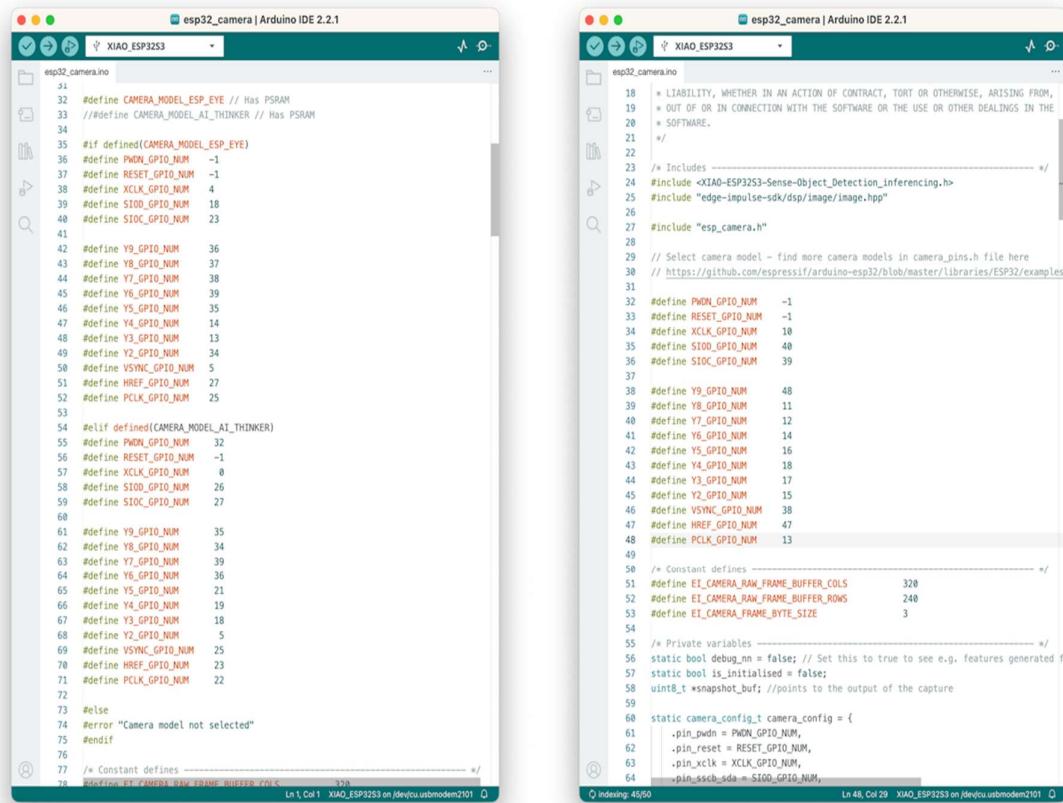


Si devono cambiare le righe dalla 32 alla 75, che definiscono il modello e i pin della telecamera, utilizzando i dati relativi al nostro modello. Copiare e incollare le righe seguenti, sostituendo le righe 32-75:

```
#define PWDN_GPIO_NUM      -1
#define RESET_GPIO_NUM     -1
#define XCLK_GPIO_NUM       10
#define SIOD_GPIO_NUM       40
#define SIOC_GPIO_NUM        39
#define Y9_GPIO_NUM          48
#define Y8_GPIO_NUM          11
#define Y7_GPIO_NUM          12
#define Y6_GPIO_NUM          14
#define Y5_GPIO_NUM          16
#define Y4_GPIO_NUM          18
#define Y3_GPIO_NUM          17
#define Y2_GPIO_NUM          15
```

```
#define VSYNC_GPIO_NUM    38
#define HREF_GPIO_NUM     47
#define PCLK_GPIO_NUM     13
```

Ecco il codice risultante:



```
esp32_camera.ino | XIAO_ESP32S3 | Arduino IDE 2.2.1
esp32_camera.ino
31
32 #define CAMERA_MODEL_ESP_EYE // Has PSRAM
33 //#define CAMERA_MODEL_AI_THINKER // Has PSRAM
34
35 #if defined(CAMERA_MODEL_ESP_EYE)
36 #define PWDN_GPIO_NUM -1
37 #define RESET_GPIO_NUM -1
38 #define XCLK_GPIO_NUM 4
39 #define SIO0_GPIO_NUM 18
40 #define SIO2_GPIO_NUM 23
41
42 #define Y9_GPIO_NUM 36
43 #define Y8_GPIO_NUM 37
44 #define Y7_GPIO_NUM 38
45 #define Y6_GPIO_NUM 39
46 #define Y5_GPIO_NUM 35
47 #define Y4_GPIO_NUM 14
48 #define Y3_GPIO_NUM 13
49 #define Y2_GPIO_NUM 34
50 #define VSYNC_GPIO_NUM 5
51 #define HREF_GPIO_NUM 27
52 #define PCLK_GPIO_NUM 25
53
54 #elif defined(CAMERA_MODEL_AI_THINKER)
55 #define PWDN_GPIO_NUM 32
56 #define RESET_GPIO_NUM -1
57 #define XCLK_GPIO_NUM 0
58 #define SIO0_GPIO_NUM 26
59 #define SIO2_GPIO_NUM 27
60
61 #define Y9_GPIO_NUM 35
62 #define Y8_GPIO_NUM 34
63 #define Y7_GPIO_NUM 39
64 #define Y6_GPIO_NUM 36
65 #define Y5_GPIO_NUM 21
66 #define Y4_GPIO_NUM 19
67 #define Y3_GPIO_NUM 18
68 #define Y2_GPIO_NUM 5
69 #define VSYNC_GPIO_NUM 25
70 #define HREF_GPIO_NUM 23
71 #define PCLK_GPIO_NUM 22
72
73 #else
74 #error "Camera model not selected"
75 #endif
76
77 /* Constant defines */
78 #define EL_CAMERA_RAW_FRAME_BUFFER_COLS 320
79 #define EL_CAMERA_RAW_FRAME_BUFFER_ROWS 240
80 #define EL_CAMERA_FRAME_BYTE_SIZE 3
81
82 /* Private variables */
83 static bool debug_on = false; // Set this to true to see e.g. features generated for
84 static bool is_initialised = false;
85 uint8_t *snapshot_buf; //points to the output of the capture
86
87 static camera_config_t camera_config = {
88     .pin_pdn = PWDN_GPIO_NUM,
89     .pin_reset = RESET_GPIO_NUM,
90     .pin_xclk = XCLK_GPIO_NUM,
91     .pin_sccb_sda = SIO0_GPIO_NUM,
```

Ln 1, Col 1 XIAO\_ESP32S3 on /dev/cu.usbmodem2101 Q

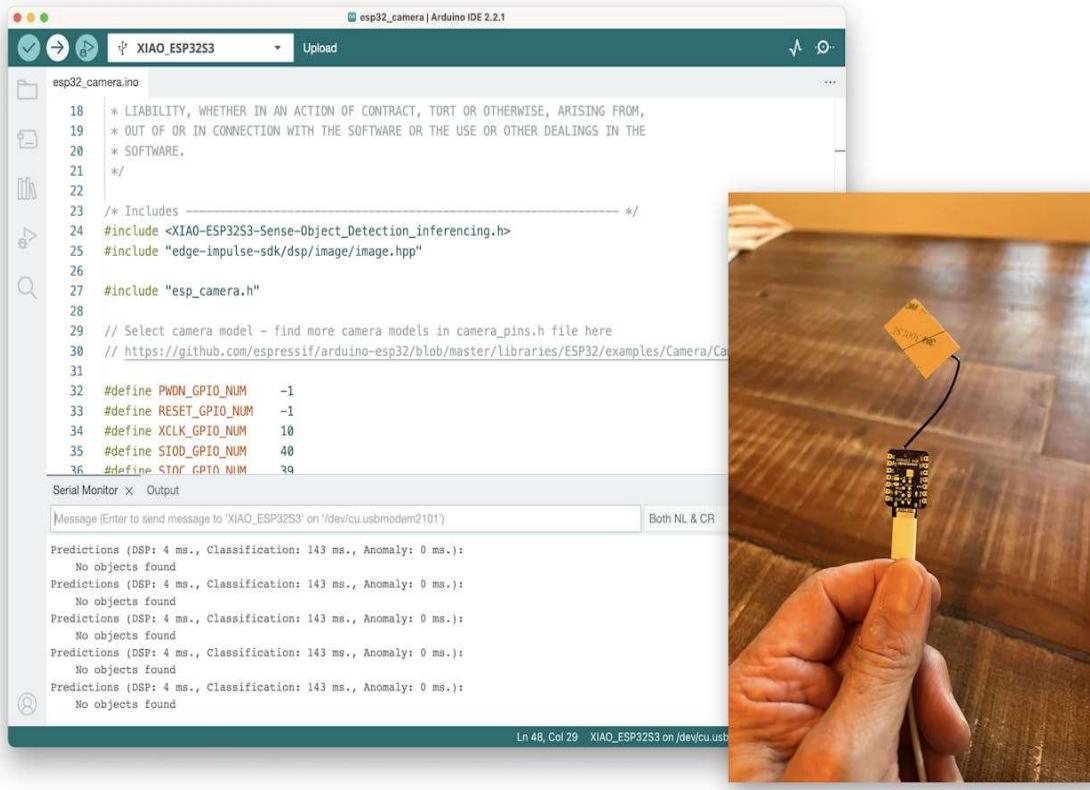
  

```
esp32_camera.ino | XIAO_ESP32S3 | Arduino IDE 2.2.1
esp32_camera.ino
18 /* LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
19 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
20 * SOFTWARE.
21 */
22
23 /* Includes */
24 #include <XIAO-ESP32S3-Sense-Object_Detection_inferencing.h>
25 #include "edge-impulse-sdk/dsp/image/image.hpp"
26
27 #include "esp_camera.h"
28
29 // Select camera model - find more camera models in camera_pins.h file here
30 // https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP32/examples
31
32 #define PWDN_GPIO_NUM -1
33 #define RESET_GPIO_NUM -1
34 #define XCLK_GPIO_NUM 10
35 #define SIO0_GPIO_NUM 40
36 #define SIO2_GPIO_NUM 39
37
38 #define Y9_GPIO_NUM 48
39 #define Y8_GPIO_NUM 11
40 #define Y7_GPIO_NUM 12
41 #define Y6_GPIO_NUM 14
42 #define Y5_GPIO_NUM 16
43 #define Y4_GPIO_NUM 18
44 #define Y3_GPIO_NUM 17
45 #define Y2_GPIO_NUM 15
46 #define VSYNC_GPIO_NUM 38
47 #define HREF_GPIO_NUM 47
48 #define PCLK_GPIO_NUM 13
49
50 /* Constant defines */
51 #define EL_CAMERA_RAW_FRAME_BUFFER_COLS 320
52 #define EL_CAMERA_RAW_FRAME_BUFFER_ROWS 240
53 #define EL_CAMERA_FRAME_BYTE_SIZE 3
54
55 /* Private variables */
56 static bool debug_on = false; // Set this to true to see e.g. features generated for
57 static bool is_initialised = false;
58 uint8_t *snapshot_buf; //points to the output of the capture
59
60 static camera_config_t camera_config = {
61     .pin_pdn = PWDN_GPIO_NUM,
62     .pin_reset = RESET_GPIO_NUM,
63     .pin_xclk = XCLK_GPIO_NUM,
64     .pin_sccb_sda = SIO0_GPIO_NUM,
```

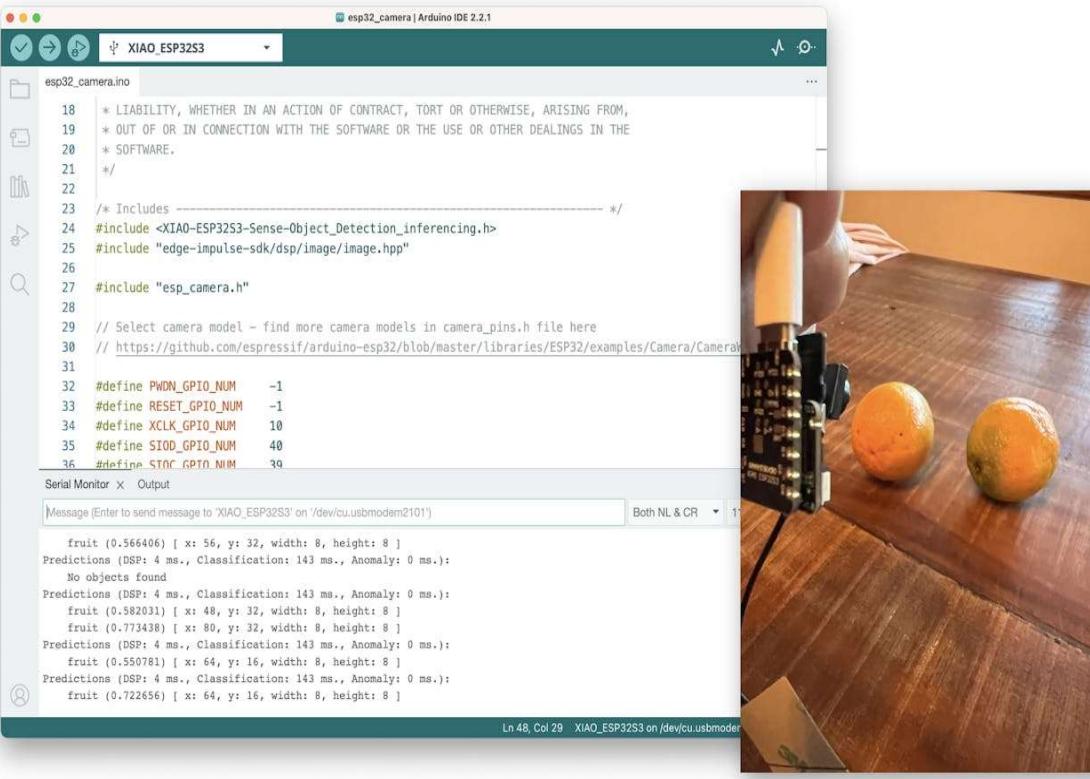
Ln 48, Col 29 XIAO\_ESP32S3 on /dev/cu.usbmodem2101 Q

Caricare il codice sul XIAO ESP32S3 Sense e si è pronti a iniziare a rilevare frutta e insetti. Si può controllare il risultato su Serial Monitor.

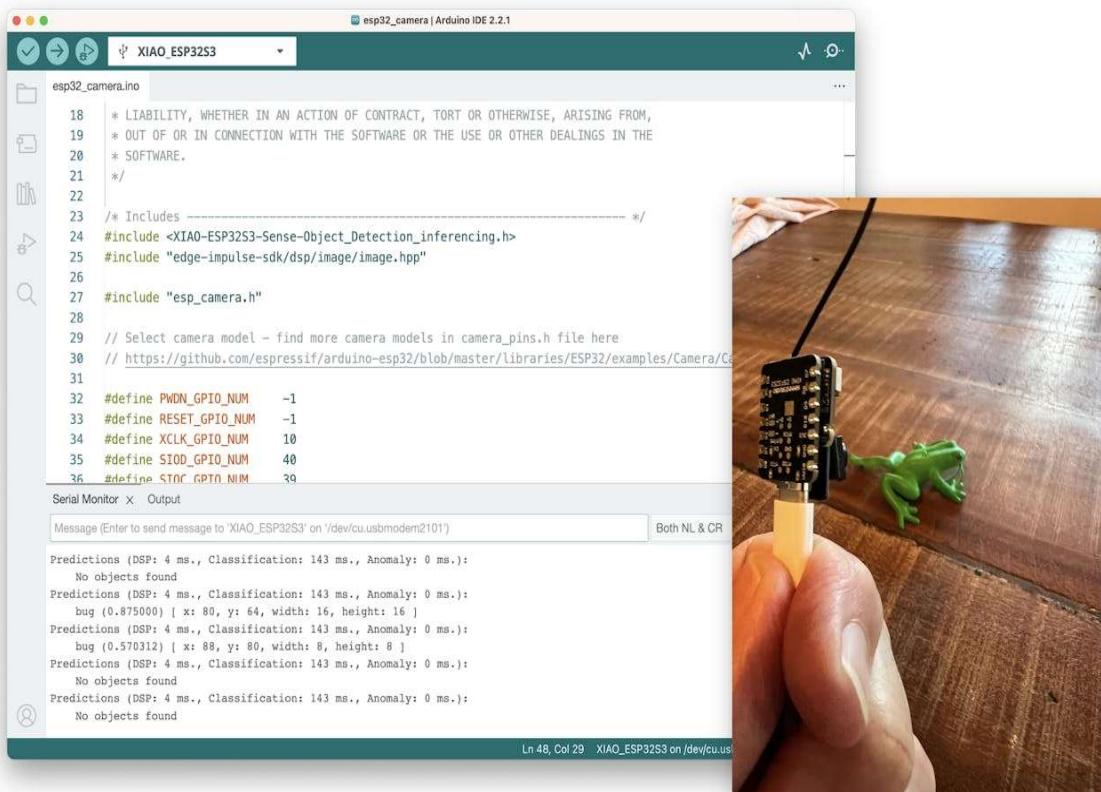
## Background



## Fruits



## Bugs



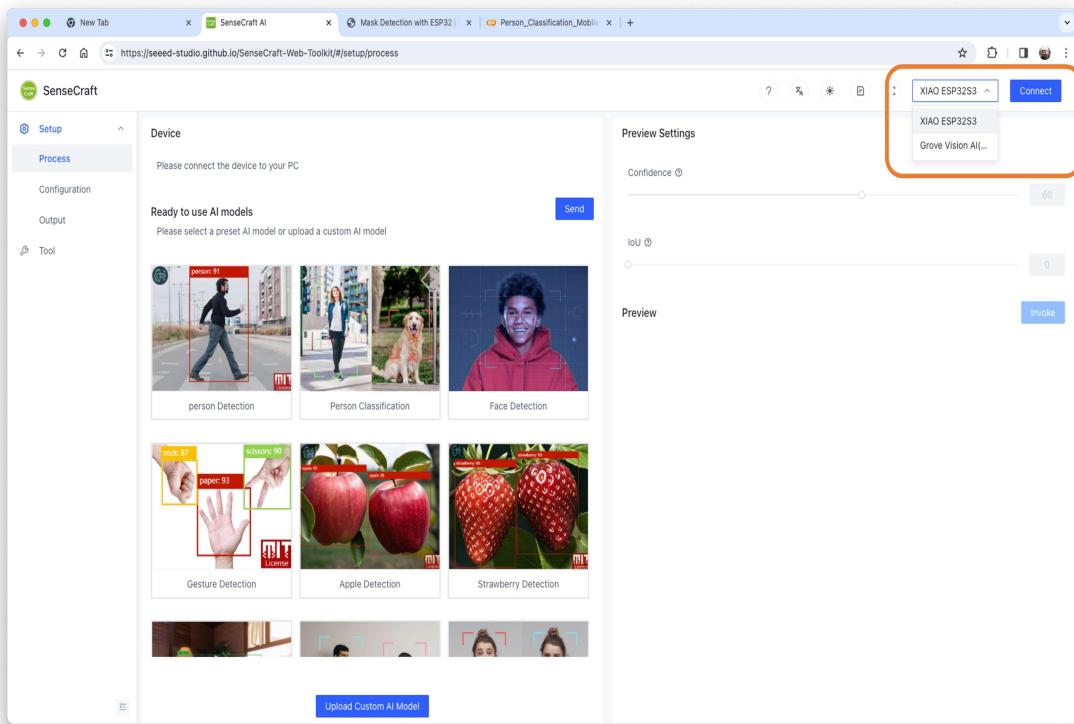
Si noti che la latenza del modello è di 143ms e il frame rate al secondo è di circa 7 fps (simile a quanto ottenuto con il progetto Image Classification). Ciò accade perché FOMO è intelligentemente costruito su un modello CNN, non con un modello di rilevamento degli oggetti come SSD MobileNet. Ad esempio, quando si esegue un modello MobileNetV2 SSD FPN-Lite 320x320 su un Raspberry Pi 4, la latenza è circa cinque volte superiore (circa 1,5 fps).

### 4.5.8 Distribuzione del Modello (SenseCraft-Web-Toolkit)

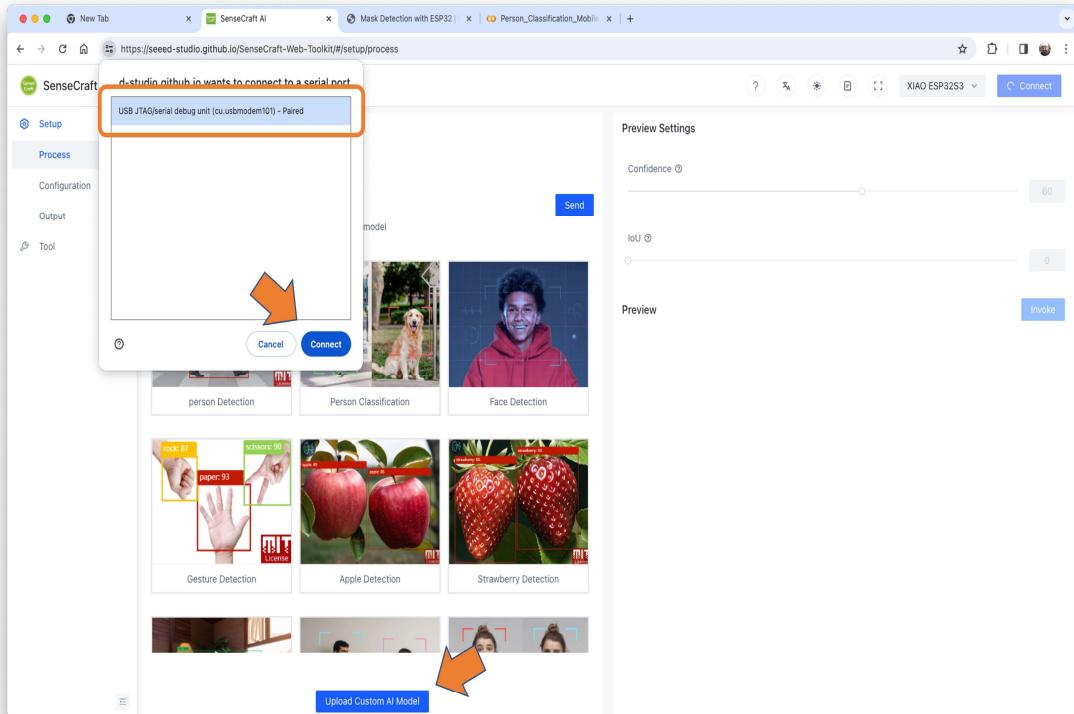
Come discusso nel capitolo Image Classification, verificare l'inferenza con i modelli di immagine su Arduino IDE è molto impegnativo perché non possiamo vedere su cosa punta la telecamera. Di nuovo, utilizziamo **SenseCraft-Web Toolkit**.

Seguire i seguenti passaggi per avviare SenseCraft-Web-Toolkit:

1. Aprire il [sito web di SenseCraft-Web-Toolkit](#).
2. Collega XIAO al computer:
  - Dopo aver collegato XIAO, selezionarlo come di seguito:



- Selezionare il dispositivo/Porta e premere [Connect]:



Si possono provare diversi modelli di Computer Vision caricati in precedenza da Seeed Studio. Da provare e verificarli!

Nel nostro caso, useremo il pulsante blu in fondo alla pagina: [Upload Custom AI Model].

Ma prima, dobbiamo scaricare da Edge Impulse Studio il modello **quantized .tflite**.

3. Si va sul proprio progetto su Edge Impulse Studio, oppure si clona questo:

- **XIAO-ESP32S3-CAM-Fruits-vs-Veggies-v1-ESP-NN**

4. Su Dashboard, scaricare il modello (“block output”): **Object Detection model - TensorFlow Lite (int8 quantized)**

**Download block output**

| TITLE                         | TYPE                                    | SIZE         |
|-------------------------------|-----------------------------------------|--------------|
| Image training data           | NPY file                                | 52 windows   |
| Image training labels         | JSON file                               | 52 windows   |
| Image testing data            | NPY file                                | 6 windows    |
| Image testing labels          | JSON file                               | 6 windows    |
| Object detection model        | TensorFlow Lite (float32)               | 82 KB        |
| <b>Object detection model</b> | <b>TensorFlow Lite (int8 quantized)</b> | <b>55 KB</b> |
| Object detection model        | TensorFlow SavedModel                   | 186 KB       |
| Object detection model        | Keras h5 model                          | 88 KB        |

**Collaborators (1/4)**  
MJRoBot (Marcelo Rovai) OWNER

**Summary**  
DEVICES CONNECTED 1  
DATA COLLECTED 59 items

**Project info**  
Project ID 315759

5. Su SenseCraft-Web-Toolkit, usare il pulsante blu in fondo alla pagina: **[Upload Custom AI Model]**. Si aprirà una finestra. Inserire il file del Modello scaricato sul computer da Edge Impulse Studio, scegliere un nome del modello e inserirlo con le etichette (ID: Object):

**SenseCraft**

**Custom AI Model**

Model Name: Bug and Fruit: Object Detection (FOMO)

Model File: bug\_and\_fruit-object-detection-tensorflow-int8.tflite

ID: Object: background x: 1bug x: 2fruit x:

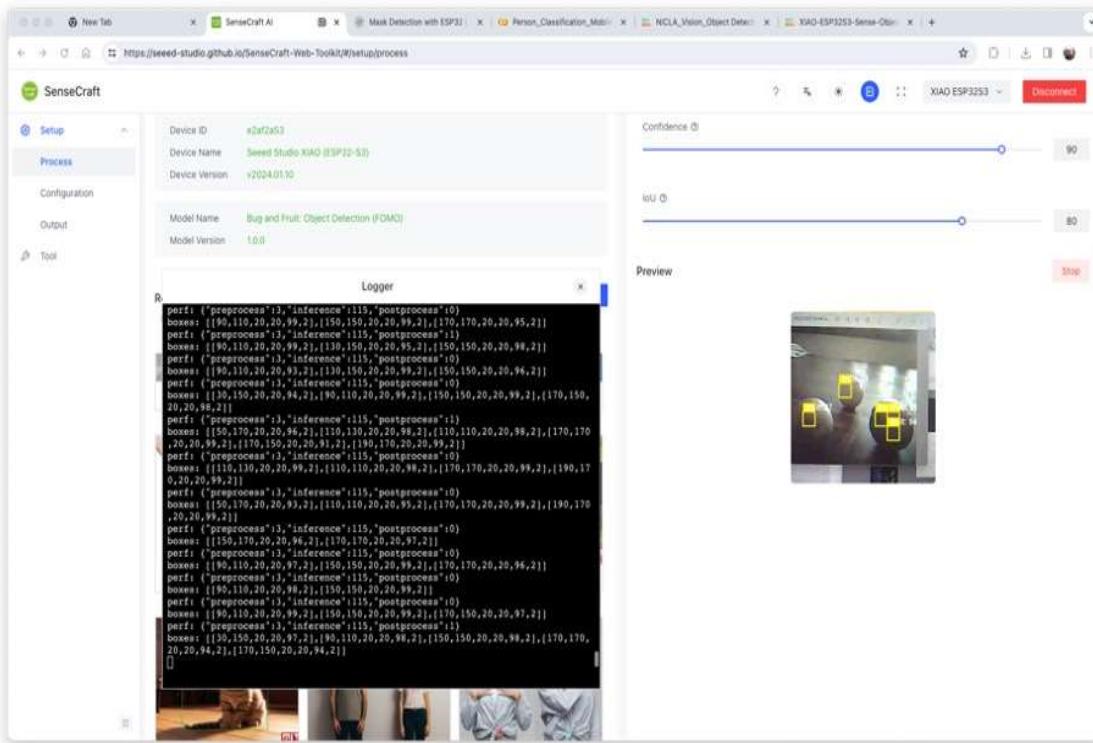
**Send Model**

**Upload Custom AI Model**

**seeed studio**

Notare che si devono utilizzare le etichette apprese su El Studio, inserendole in ordine alfabetico (nel nostro caso: background, bug, fruit).

Dopo alcuni secondi (o minuti), il modello verrà caricato sul dispositivo e l'immagine della telecamera apparirà in tempo reale nel Preview Sector:



Gli oggetti rilevati saranno contrassegnati (il centroide). Si può selezionare la Confidence del cursore di inferenza `confidence`. e `IoU`, che viene utilizzato per valutare l'accuratezza delle bounding box previste rispetto alle vere bounding box.

Cliccando sul pulsante in alto (Device Log), si può aprire un Serial Monitor per seguire l'inferenza, come abbiamo fatto con l'IDE Arduino.

```

perf: {"preprocess":3,"inference":115,"postprocess":1}
boxes: [[30,150,20,20,97,2],[90,110,20,20,98,2],[150,150,20,20,98,2],[170,170,20,20,94,2],[170,150,20,20,94,2]]
  
```

Su Device Log, si otterranno informazioni come:

- Tempo di pre-elaborazione (acquisizione dell'immagine e Crop): 3 ms,
- Tempo di inferenza (latenza del modello): 115 ms,
- Tempo di post-elaborazione (visualizzazione dell'immagine e marcatura degli oggetti): 1 ms.
- Tensore di output (caselle), ad esempio, una delle caselle: [[30,150,20,20,97,2]]; dove 30,150,20,20 sono le coordinate della casella (intorno al centroide); 97 è il risultato dell'inferenza e 2 è la classe (in questo caso 2: frutto).

Notare che nell'esempio precedente, abbiamo ottenuto 5 caselle perché nessuno dei frutti ha ottenuto 3 centroidi. Una soluzione sarà la post-elaborazione, dove possiamo aggregare centroidi vicini in uno.

Ecco altri screenshot:



### 4.5.9 Conclusione

FOMO è un salto significativo nello spazio di elaborazione delle immagini, come hanno affermato Louis Moreau e Mat Kelcey durante il suo lancio nel 2022:

FOMO è un algoritmo rivoluzionario che porta per la prima volta il rilevamento, il tracciamento e il conteggio degli oggetti in tempo reale sui microcontrollori.

Esistono molteplici possibilità per esplorare il rilevamento di oggetti (e, più precisamente, il loro conteggio) su dispositivi embedded.

## 4.6 Per saperne di più

Questa sezione contiene link a corsi, libri e progetti per saperne di più sulle applicazioni di Machine Learning e TinyML.

### Corsi Online

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
- UNIFEI-ESTI01 TinyML: “Machine Learning for Embedding Devices”

### Libri

- “Python for Data Analysis by Wes McKinney”
- “Deep Learning with Python” by François Chollet - GitHub Notebooks
- “TinyML” by Pete Warden, Daniel Situnayake
- “TinyML Cookbook” by Gian Marco Iodice
- “Technical Strategy for AI Engineers, In the Era of Deep Learning” by Andrew Ng
- “AI at the Edge” book by Daniel Situnayake, Jenny Plunkett
- “MACHINE LEARNING SYSTEMS for TinyML” Collaborative effort

### Repository di Progetti

- Edge Impulse Expert Network
- MRovai XIAO ESP32S3 Movement/Sound/Image

## 5.1 Progetti XIAO creativi e utili

Dopo aver letto le sezioni di questo libro, potrebbero venire molte idee innovative che non si vede l'ora di implementare. Ma prima di buttarsi, diamo un'occhiata alle cose interessanti che altri hanno realizzato con XIAO. Per questo, abbiamo raccolto alcuni casi di progetti

realizzati dagli utenti che utilizzano XIAO principalmente dalle comunità di innovatori di fama mondiale come [hackster.io](#) e [instructables](#), per vedere più possibilità di XIAO.

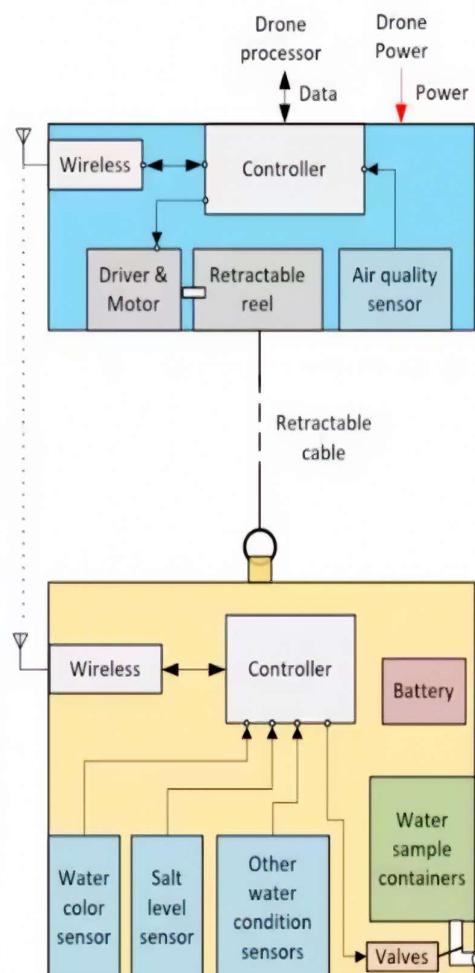
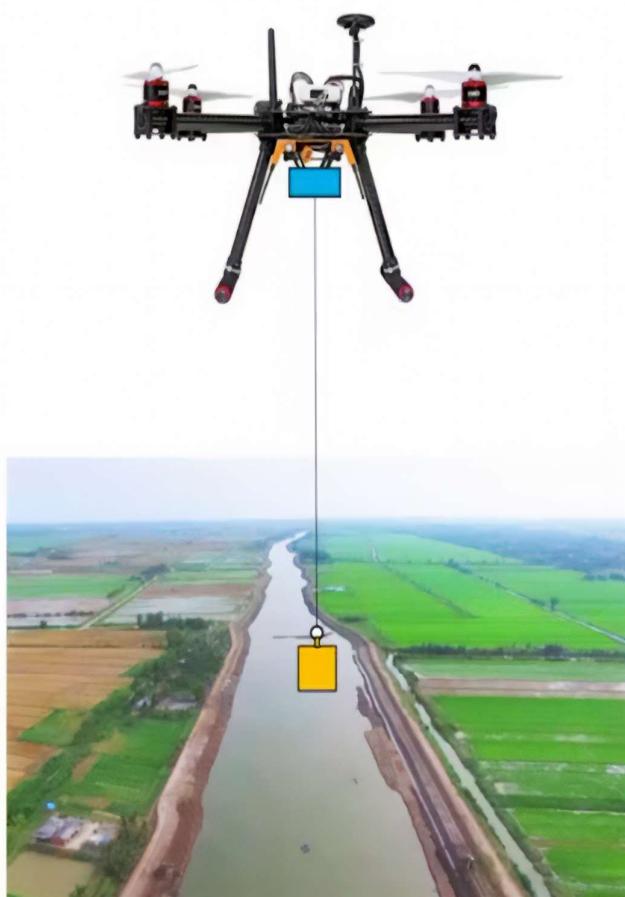
### 5.1.1 Salt Water Tracker (SWT) Trasportato da Drone

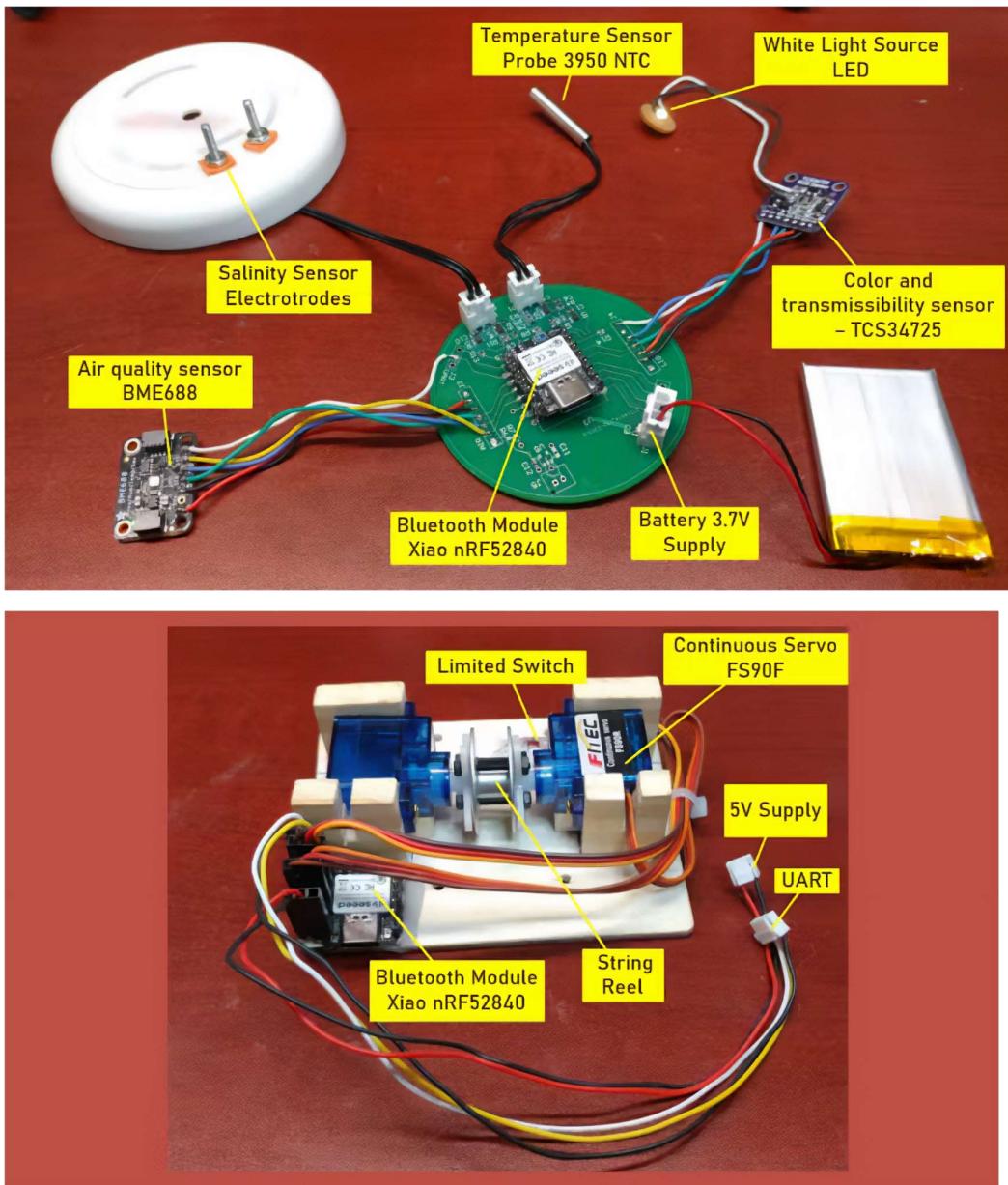
<https://www.hackster.io/txnghia/salt-water-tracker-swt-cb68be>

Autore: Nghia Tran

Il progetto 'Salt Water Tracker', che usa [XIAO BLE nRF52840 Sense](#), affronta il problema dell'erosione provocata dall'acqua marina nelle risaie in aree come il delta del Mekong in Vietnam. Il progetto integra un sistema di sensori di acqua salata su un drone Hovergames, trasformando il drone in un efficiente strumento di tracciamento dell'acqua salata. Questo progetto aiuta gli agricoltori a monitorare la salinità dei fiumi e delle grandi reti idriche in tempo reale per garantire la sicurezza dell'acqua e guidare l'assegnazione dell'acqua di riserva. Il sistema è inoltre dotato di sensori di temperatura, qualità dell'acqua, qualità dell'aria e una funzione fotocamera per scattare foto o registrare video dell'area acquatica e aiutare a determinare il tipo e le condizioni dell'acqua.

Salt Water Tracker





### 5.1.2 Progetto SAJAC: Giacca Intelligente per Avventure Speleologiche

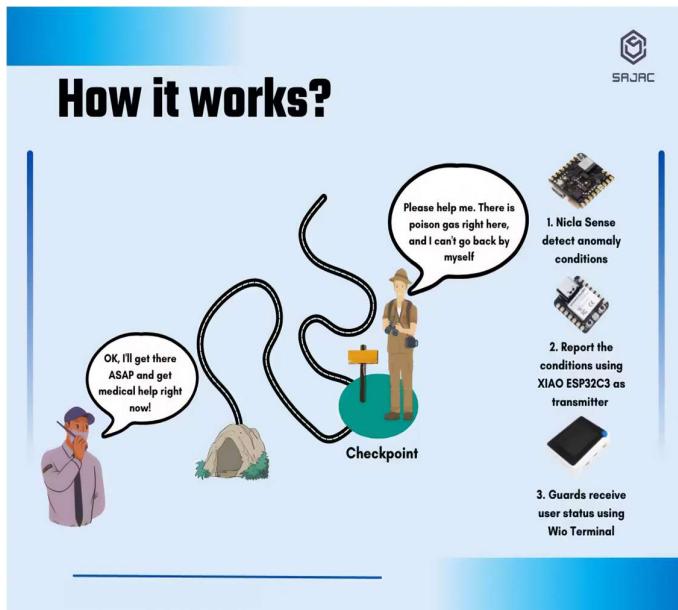
<https://www.hackster.io/rifqiabdillah/sajac-project-smart-jacket-for-caving-0e6a19>

Autore: [Rifqi Abdillah](#)

La speleologia è diventata sempre più popolare negli ultimi anni. Tuttavia, gli speleologi possono trovarsi ad affrontare una serie di pericoli per la sicurezza, tra cui temperature estreme, aria umida, bassa pressione atmosferica, scarsa qualità dell'aria e gas tossici. Per risolvere questo problema, è stato sviluppato il progetto SAJAC, un sistema di monitoraggio intelligente progettato per osservare le condizioni ambientali all'interno di una grotta. Il sistema utilizza Nicla Sense ME per misurare la qualità ambientale attorno all'utente e invia i risultati all'app SAJAC sullo smartphone dell'utente. Se le condizioni della grotta non sono adatte all'esplorazione, Nicla Sense ME o lo smartphone dell'utente riceveranno un promemoria di notifica. Nel frattempo, a ogni punto di controllo all'interno della grotta, ci sarà un trasmettitore direttamente collegato alla guardia all'esterno della grotta. L'utente può rapidamente cercare aiuto tramite il trasmettitore in caso di pericolo.

Considerando che non c'è una connessione Internet nella grotta, utilizziamo un sistema di comunicazione LoRa basato su XIAO ESP32C3 per trasmettere i dati del checkpoint. Quando l'utente raggiunge il checkpoint, deve solo connettersi al trasmettitore e premere il pulsante "Invia". Se l'utente si imbatte in una situazione in cui non può continuare l'esplorazione, può decidere se tornare da solo o attendere la risposta della guardia.

La guardia del posto principale utilizzerà LoRa per ricevere i dati trasmessi dall'interno della grotta. C'è un terminale Wio dotato di Grove Wio E5 all'avamposto per ricevere i dati dai trasmettitori all'interno della grotta. Il terminale Wio necessita solo di un alimentatore da 5 volt, adatto per luoghi con potenza limitata.



### 5.1.3 Computer da Bicicletta su Spresense

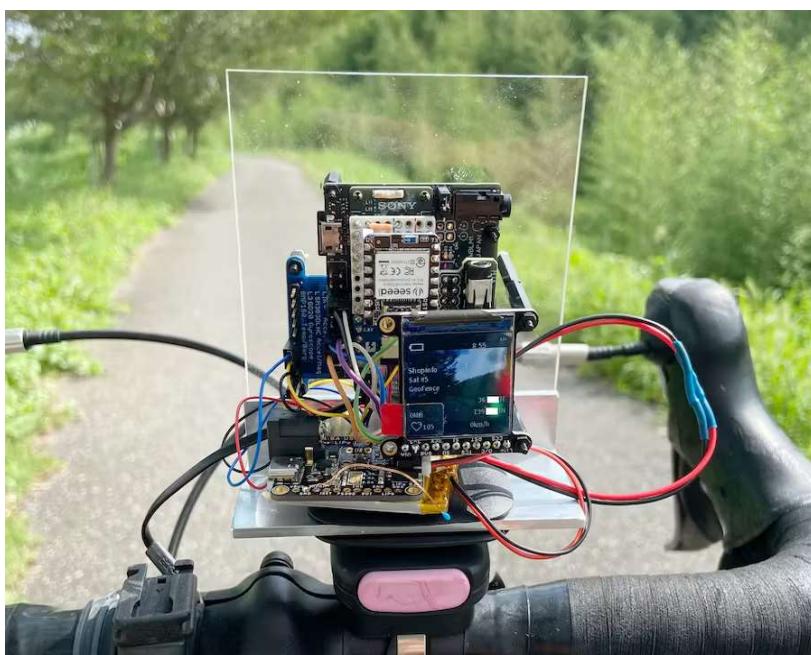
<https://www.hackster.io/jens6151/bicycle-computer-on-spresense-b0e332>

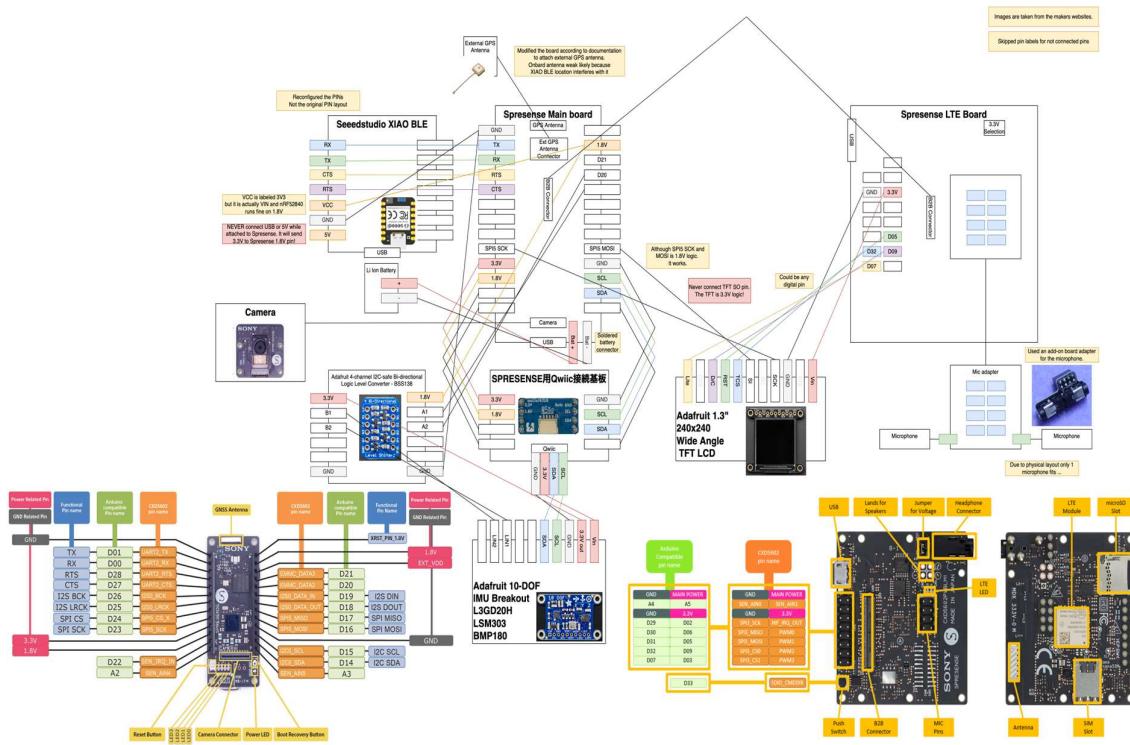
Autore: Jens

L'obiettivo di questo progetto è quello di costruire un computer da bicicletta utilizzando la scheda madre Sony Spresense, la scheda di espansione LTE, XIAO e altre periferiche. Le caratteristiche principali includono:

1. Cattura di un flusso video a bassa risoluzione e visualizzazione su un monitor. Opzione per scattare foto ad alta risoluzione e salvarle su una scheda SD.
2. Cattura audio mono, utilizzando il codec OPUS e il formato contenitore OGG per un'elevata compressione, da inviare o registrare sulla scheda SD tramite una connessione LTE-M.
3. Traccia della posizione tramite GNSS, combinando la posizione con i dati meteo e i dati dei punti di interesse (POI) ricevuti dai servizi cloud tramite una connessione LTE.
4. Collegamento dei sensori della bici (attualmente la frequenza cardiaca) tramite Bluetooth Low Energy, visualizzando i dati sul monitor e registrandoli.
5. Accesso remoto alla telecamera, flusso audio in tempo reale e vari dati (inclusa la posizione) tramite MQTT.
6. Rilevamento e notifica furti tramite geofencing GNSS, accelerometro e monitoraggio per smartphone nelle vicinanze.

Questo progetto di Jens dimostra la sorprendente complessità di un progetto prototipo hardcore, come si può vedere dallo schema a destra.



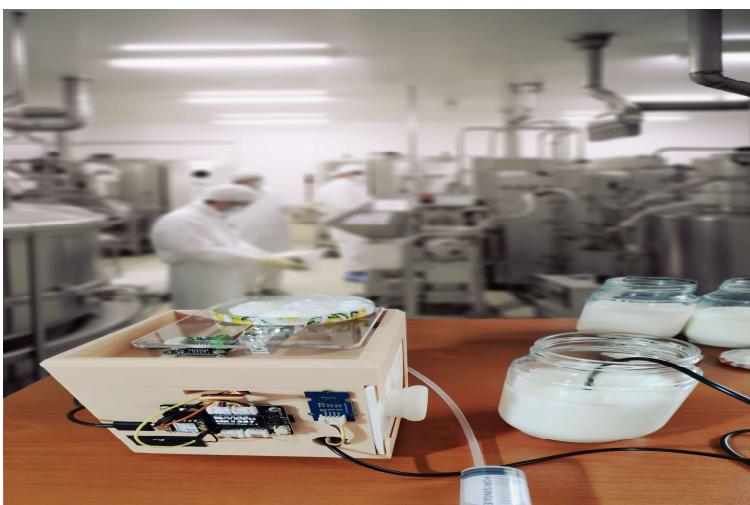


## 5.1.4 Elaborazione dello Yogurt Basata su IoT AI e Previsione della Consistenza con Blynk

<https://www.instructables.com/IoT-AI-driven-Yogurt-Processing-Texture-Prediction/>

Autore: Kutluhan Aktar

L'obiettivo di questo progetto è fornire una previsione della consistenza per l'elaborazione dello yogurt utilizzando la tecnologia IoT e l'IA. Utilizzando la scheda di sviluppo XIAO ESP32C3, insieme a un sensore di temperatura e umidità, un kit di sensori di pressione integrati, un kit di sensori di peso I2C e un sensore di temperatura impermeabile DS18B20, il creatore del progetto ha creato un modello di rete neurale artificiale e lo ha addestrato con Edge Impulse per prevedere la consistenza dello yogurt senza l'aggiunta di additivi chimici. Gli utenti possono visualizzare da remoto le letture dei sensori e controllare i dispositivi tramite l'app Blynk. Infine, l'autore ha progettato un involucro resistente adatto a un ambiente caseario. Questo progetto ha il potenziale per aiutare i produttori caseari a ridurre i costi e migliorare la qualità del prodotto.

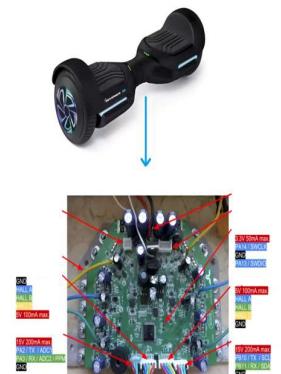
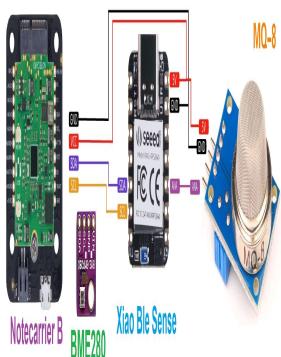


### 5.1.5 Robot azionato da browser Web per il rilevamento di perdite di gas

<https://www.hackster.io/ivan-arakistain/web-browser-operated-robot-for-gas-leak-detection-4cbe1b>

Autore: [Ivan Arakistain](#)

Questo progetto riutilizza un vecchio hoverboard trasformandolo in un robot telecomandato dotato di un sensore di idrogeno per il rilevamento precoce di perdite di idrogeno. Utilizza il Bluetooth per collegare Seedstudio Xiao Ble Sense, il sensore di gas MQ-8 e altri dispositivi e usa Edge Impulse Studio per addestrare un modello di apprendimento automatico. Il robot utilizza anche la tecnologia di connessione cellulare Blues Wireless Notecard NBGL per caricare i dati sul cloud. Con Remo.TV, può essere azionato da remoto per guidare il robot e visualizzare i feed della telecamera in tempo reale tramite un browser.

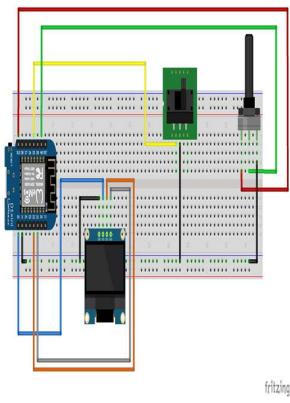
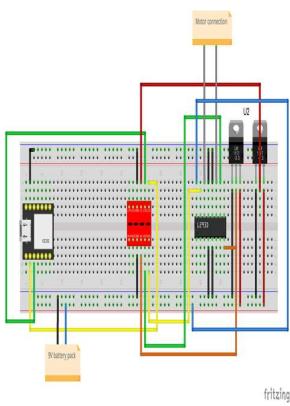


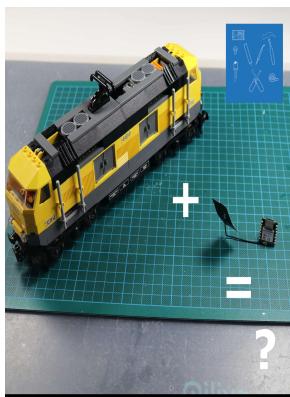
### 5.1.6 Controller del Treno con Seeed Studio XIAO ESP32C3

<https://www.instructables.com/Train-Controller-With-Seeed-Studio-XIAO-ESP32C3/>

Autore: [Tiago Santos](#)

Questo è il progetto di un controller del treno utilizzando il modulo XIAO ESP32C3 di Seeed Studio. Il progetto è diviso in una parte treno e una parte controller. La parte treno utilizza il modulo XIAO ESP32C3 per connettersi al treno e controlla il motore del treno tramite il driver del motore L293D. La parte controller utilizza Wemos D1 Mini per ricevere informazioni su velocità e direzione e visualizza la velocità effettiva su uno schermo ssd1306 da 0.96 pollici. Il controller comunica con la parte treno tramite Wi-Fi e un server MQTT. Il progetto semplifica la complessità dei tradizionali sistemi di controllo remoto dei treni Lego e migliora l'efficienza del controllo.



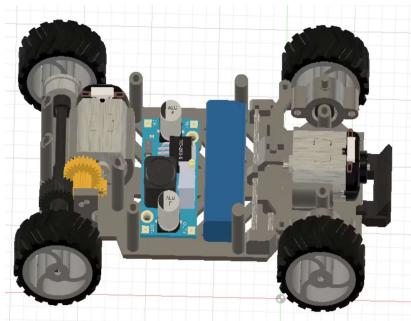
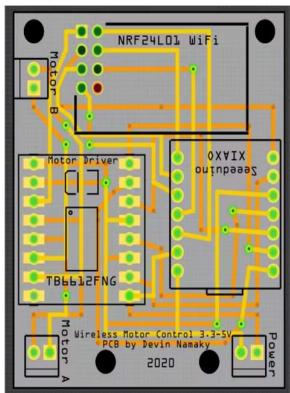


### 5.1.7 Auto Radio Comandata (Stampata in resina 3D Basata su Arduino) RC\_Car\_RP

<https://www.hackster.io/devinnamaky/rc-car-arduino-based-3d-resin-printed-rc-car-rp-9b4dce>

Autore: Devin Namaky

Questo progetto è un'auto telecomandata stampata in 3D basata su Arduino Nano e Seeeduino XIAO, denominata RC\_Car\_RP. Il progetto utilizza due motori DC standard di tipo 130 come azionamento e sterzo e il sistema di sterzo utilizza la trasmissione ad ingranaggi. Il modulo Seeeduino XIAO viene utilizzato per controllare il driver del motore TB6612FNG, realizzando il controllo della velocità e della direzione dell'auto. La comunicazione tra il telecomando e l'auto avviene tramite il modulo wireless nRF24L01. Il progetto è di piccole dimensioni, semplice nel design, facile da costruire e può soddisfare le esigenze dell'auto telecomandata in diversi scenari.



## 5.1.8 Tracker per Attività di Animali Domestici con XIAO BLE Sense e Edge Impulse

<https://www.hackster.io/mithun-das/pet-activity-tracker-using-xiao-ble-sense-edge-impulse-858d73>

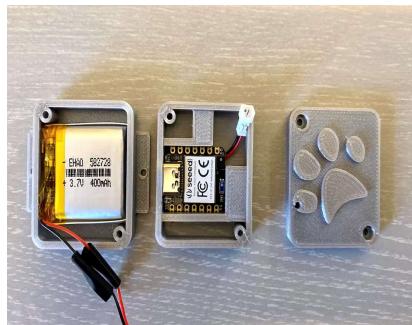


Autore: [Mithun Das](#)

Questo progetto è un dispositivo indossabile che traccia le attività degli animali domestici con XIAO BLE Sense e Edge Impulse, con l'obiettivo di aiutare i nostri animali domestici a rimanere attivi. XIAO BLE Sense è un mini controller dotato di un potente MCU Nordic nRF52840, modulo Bluetooth 5.0 integrato e progettato attorno a una CPU ARM® Cortex™-M4 a 32 bit. È dotato di un'IMU a 6 assi che può essere utilizzata per prevedere attività come riposo, camminata e corsa.

Con l'app per smartphone in dotazione, gli utenti possono connettersi al dispositivo tramite Bluetooth e ottenere dati di previsione minuto per minuto. I dati vengono archiviati nella memoria locale dello smartphone e presentati graficamente per fornire informazioni significative.

Il progetto raccoglie dati tramite l'app mobile EI Blue, crea modelli di apprendimento automatico tramite Edge Impulse Studio e crea un'app iOS tramite Google Flutter. L'intero sistema può monitorare lo stato di attività dell'animale in tempo reale e visualizzare i dati tramite l'app mobile.





### 5.1.9 H.E.D.S. Al polso, la nuova scheda Seeeduino XIAO

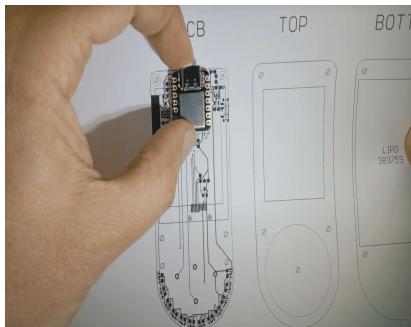
<https://www.hackster.io/ihayri1/h-e-d-s-on-your-wrist-new-seeeduino-xiao-board-7d8f74>

<https://youtu.be/ql2wnFtSQqQ>

Autore: [Hayri Uygur](#)

Hayri ha realizzato un orologio da polso multifunzionale in stile Maker, H.E.D.S., tramite XIAO. Fornisce un set di piccoli e pratici strumenti con molte funzioni e varianti ed è dotato di un bellissimo e nitido display IPS da 240x240 pixel.





### 5.1.10 Monitor del Battito Cardiaco con XIAO NRF52840

<https://www.instructables.com/Hearbeat-Monitor-With-XIAO-NRF52840/>

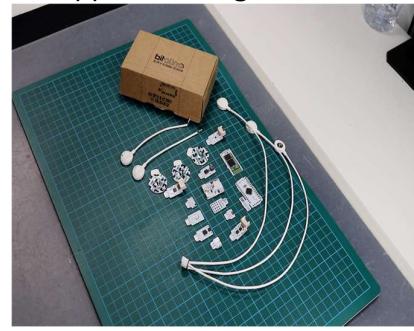
Autore: TiagoSantos

Questo progetto utilizza un microcontrollore XIAO NRF52840, basato sulla CPU Nordic nRF52840, per realizzare un monitor del battito cardiaco. Questo microcontrollore supporta Bluetooth 5.0 e NFC e ha dimensioni super ridotte, il che lo rende ideale per dispositivi indossabili e altri progetti con spazio limitato. Il progetto utilizza un altro microcontrollore biomedico chiamato Bitalino per monitorare il battito cardiaco. XIAO NRF52840 riceve informazioni dal sensore ECG (elettrocardiogramma) e poi le trasmette a un set di LED. Attraverso questo progetto, possiamo visualizzare la frequenza cardiaca in tempo reale e osservare i dati dell'attività cardiaca.

1. Preparazione della versione Bluetooth di XIAO nRF52840. Le sue dimensioni ridotte sono molto adatte ai dispositivi indossabili.

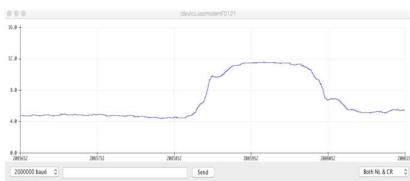
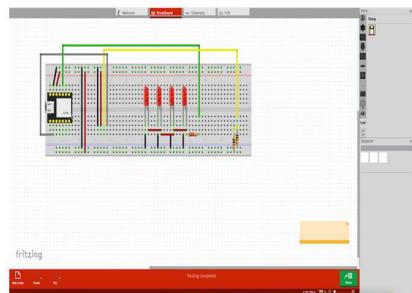


2. Bitalino è un kit biomedico simile ad Arduino sviluppato da Hugo Silva in Portogallo.

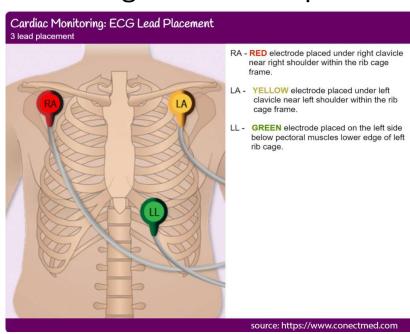


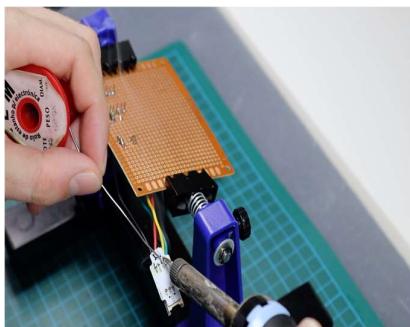
Questo progetto ne utilizzerà alcuni moduli.

3. Schema del circuito: XIAO riceve informazioni sulla frequenza cardiaca dal sensore ECG, le converte e le invia. Il LED lampeggia con la frequenza cardiaca e il plotter della porta seriale Arduino visualizza le informazioni grafiche della frequenza cardiaca.

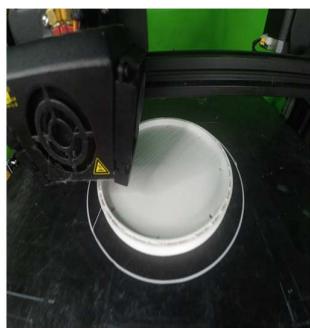
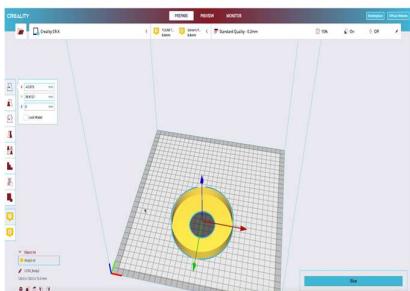


4. Si usa una scheda perforata per posizionare i componenti e saldarli. Innanzitutto, si posizionano i resistori e i pin femmina di XIAO, poi si salda il sensore ECG. Infine, si taglia la scheda perforata alla dimensione richiesta.

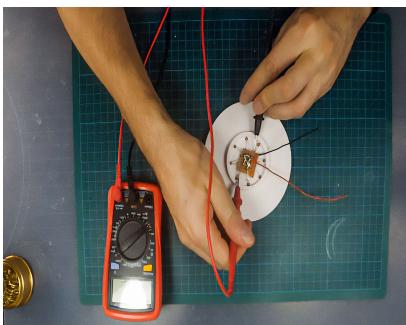




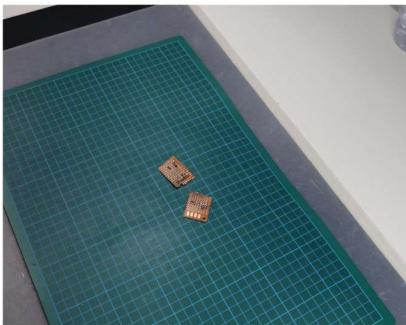
5. Si usa Fusion 360 per progettare la calotta LED, la calotta principale e la struttura della parte toracica. Si usa Creality Slicer per transcodificare e inviare alla stampante 3D per ottenere parti strutturali.



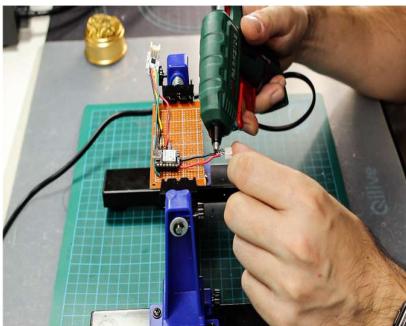
6. Quando si collegano il LED, si usa una scheda perforata per collegare tutti i catodi e posizionare un connettore di massa. Dopo aver completato tutti i collegamenti, è necessario verificare se VCC è isolato da terra ed eseguire un test.



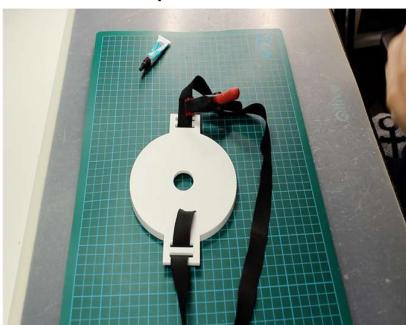
7. Non tutto può andare come previsto. Durante il controllo del collegamento, l'apparecchio ha esercitato troppa forza, causando la rottura della scheda perforata. Ha dovuto essere rifatta.



8. Finalmente, è il momento di collegare la batteria e isolare tutti i circuiti per evitare cortocircuiti. Di solito, qui si usa un tubo termorestringente, ma se non c'è una dimensione adatta, può funzionare anche la colla a caldo.



9. Posizionare tutti i componenti sulle parti stampate in 3D ed eseguire un test, poi usare la super colla per unire le parti. La parte fissata sul torace è stata incollata con un elastico. Infine, si sostituisce il LED e si rimuove la resistenza per ottenere effetti di luce più evidenti.



10. L'effetto finale.

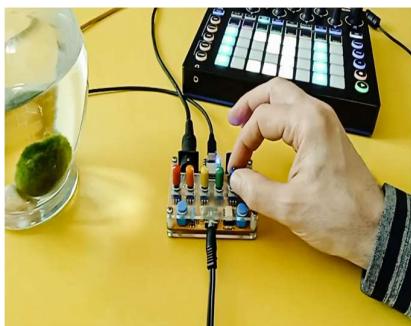


### 5.1.11 Controller Multi MIDI, Filtro, Router e Generatore di Suoni

<https://www.synthtopia.com/content/2022/03/29/multi-midi-controller-filter-router-sound-generator/> <https://github.com/pangrus/multi>

Autore: Pangrus

Multi è un controller MIDI multifunzionale, utilizzato principalmente per la sintesi audio, con dimensioni molto ridotte. Rispetto all'ultima generazione di controller commerciali, ha una porta USB e due interfacce DIN. Il controller Multi è completamente programmabile, consentendo alcune funzionalità in una configurazione senza computer. Inoltre, può anche essere utilizzato come generatore di suoni in quanto è dotato di un convertitore DAC a 10 bit, rendendolo ideale per esplorare la tecnologia di sintesi digitale. Il controller Multi è alimentato dal robusto Seeeduino XIAO, dotato di 6 manopole, 2 pulsanti, 2 interfacce Midi DIN e un'interfaccia audio da 1/8 di pollice. Il suo ingresso MIDI ha un isolamento ottico per evitare loop di massa, conforme alle specifiche ufficiali.

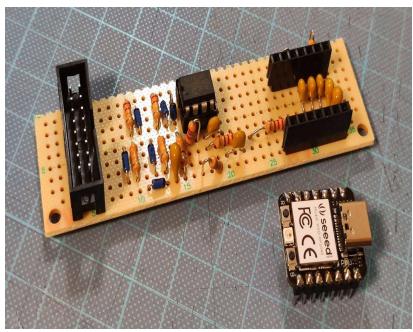


### 5.1.12 Sintetizzatore Modulare eurorack fai da te Raspberry Pi VCO con Seeed XIAO

<https://www.hackster.io/hagiwo/diy-eurorack-modular-synth-raspberry-pi-vco-with-seeed-xiao-133ac0>

Autore: HAGIWO/ハギヲ

Un maker giapponese, [HAGIWO/ハギヲ](#), ha utilizzato la scheda di sviluppo Seeed XIAO RP2040 per creare un modulo Voltage-Controlled Oscillator (VCO) per un sintetizzatore modulare Eurorack. Questa scheda ha un microcontrollore Raspberry Pi RP2040, 4 convertitori AD ed è più facile da usare rispetto al Raspberry Pi Pico. Il modulo VCO ha tre modalità: Wavefold, FM e AM, con otto forme d'onda integrate, al costo di soli 1100 yen (6.8 Euro circa).



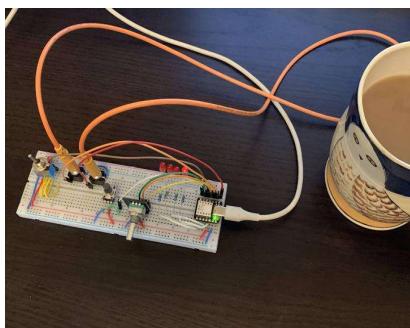
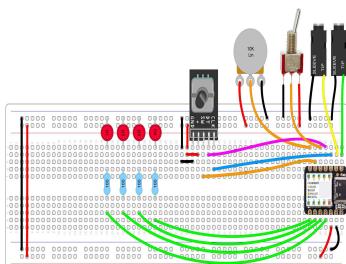
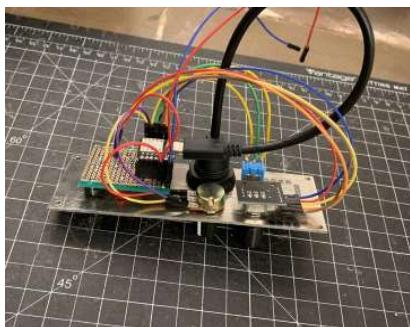
### 5.1.13 Sequencer Xiao CV

<https://www.instructables.com/Xiao-CV-Sequencer/>

Autore: [analogsketchbook](#)

Utilizzando il microcontrollore Seeduino Xiao e alcune parti, è stato creato un discreto sintetizzatore CV, principalmente per sistemi di sintetizzatori modulari. Il ruolo di Xiao in questo progetto è quello di emettere segnali di Control Voltage (CV) tramite i suoi pin di uscita analogici per passare informazioni sulle note tra i moduli. Controlla anche altre funzioni come la regolazione della velocità, la commutazione della modalità e la selezione della sequenza.



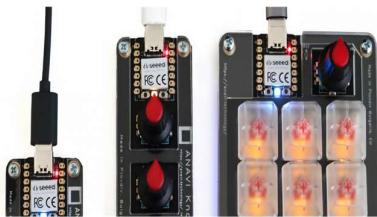


### 5.1.14 ANAVI Macro Pad 10 e Manopole

<https://www.crowdsupply.com/anavi-technology/anavi-macro-pad-10>

Autore: Crowd Supply

Un'azienda ha progettato e prodotto tre piccoli dispositivi di input meccanici open source programmabili tramite crowdfunding: la tastiera ANAVI Macro Pad 10, ANAVI Knob 3 e ANAVI Knob 1. Tutti sono gestiti dal potente microcontrollore Raspberry Pi RP2040 all'interno di Seeed XIAO RP2040, supportano USB Type-C ed eseguono il firmware KMK basato su CircuitPython. Questi dispositivi personalizzabili sono adatti per l'editing video o audio, la trasmissione di intrattenimento, i giochi, la programmazione, ecc., fornendo un controllo preciso ed effetti di illuminazione pratici. Sono semplici da usare e i loro piani e schemi si trovano su GitHub.



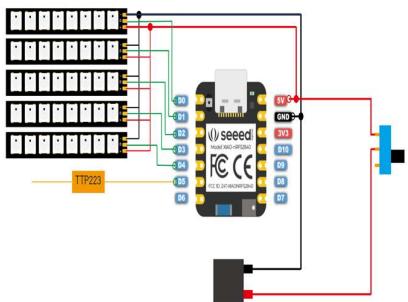
### 5.1.15 Lampada da Scrivania Death Stranding

<https://www.hackster.io/wyx269263336/death-stranding-desk-lamp-ae5f71>

Autore: [Pinkman](#)

Questa lampada intelligente, basata sul dispositivo di scansione multifunzionale Odradek nel gioco Death Stranding, è composta da cinque lame luminose separate, ciascuna con tre gradi di libertà, così da poter regolare l'angolazione desiderata in qualsiasi momento. Integra la scheda di controllo principale Bluetooth XIAO nRF52840 Sense e la striscia luminosa magic WS2812, e si può controllarne il colore e la luminosità tramite un'app mobile.







### 5.1.17 DISCIPLINE - Un timer per l'allenamento

<https://www.hackster.io/rw2493/discipline-a-workout-timer-6b5614>

Autore: [Rui Wang](#)

**DISCIPLINE:** Questo è un timer fatto in casa che aiuta a controllare rigorosamente gli intervalli di riposo durante l'allenamento muscolare. Il progetto utilizza il microcontrollore Seeeduino XIAO, insieme a due pulsanti, uno schermo di visualizzazione, una batteria e altri componenti per ottenere un'interfaccia utente semplice e un design portatile. XIAO è responsabile della funzione di controllo principale del timer nel progetto, fornendo servizi di temporizzazione accurati agli utenti

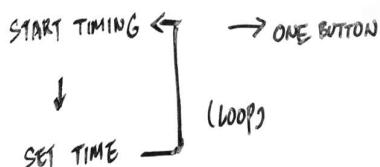


**Gli obiettivi di progettazione includono:**

- Piccolo, portatile e compatto
- Funzioni complete del timer
- Design semplice dell'interfaccia utente
- Flusso di interazione chiaro
- Aspetto accattivante

L'interazione è progettata per essere il più semplice possibile per ridurre al minimo i passaggi operativi.

SET A TIME (SETUP) → ONE BUTTON.



**Descrizione dell'interazione della luce del pulsante giallo e blu:** Dopo alcuni test di prova, si usa il pulsante giallo per controllare l'impostazione del tempo e il pulsante blu per avviare il conteggio. Per fornire una buona indicazione, si sono diverse cose fatte per i LED. (Y per giallo, B per blu) Quando lo si accende: Y -> Dissolvenza; B -> ON, indica di selezionare un periodo di tempo.

- Quando lo si accende: Y -> Fade; B -> ON, indica di selezionare un periodo di tempo.

Premere Y per cambiare le opzioni di temporizzazione: 30s, 60s, 90s, 120s.

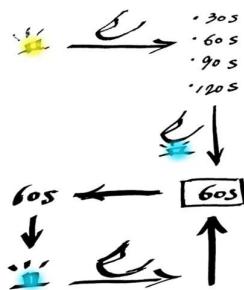
- Premere Y per cambiare le opzioni di temporizzazione: 30s, 60s, 90s, 120s.

Premere B per confermare la scelta, il timer inizia il conto alla rovescia. Y -> OFF; B -> OFF.

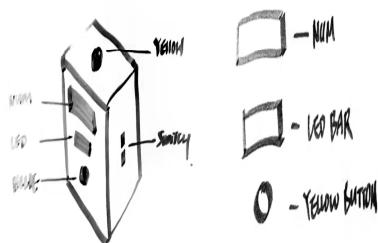
- Premere B per confermare la scelta, il timer inizia il conto alla rovescia. Y -> OFF; B -> OFF.

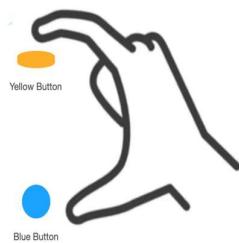
Il timer termina il conteggio, B -> ON; Y -> OFF per sempre.

- Il timer termina il conteggio, B -> ON; Y -> OFF per sempre.

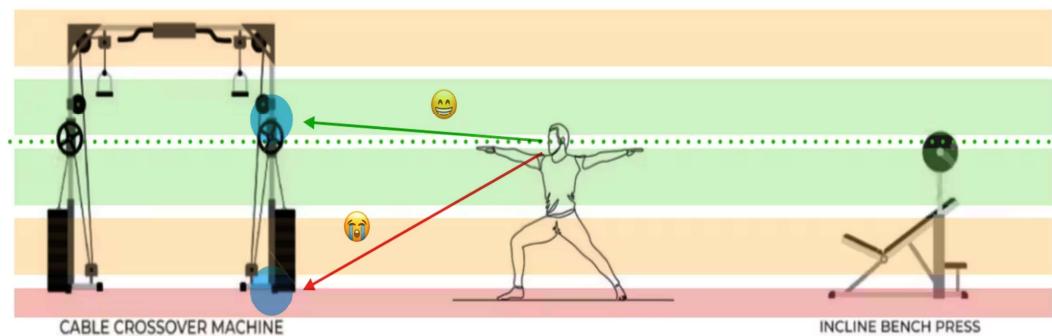
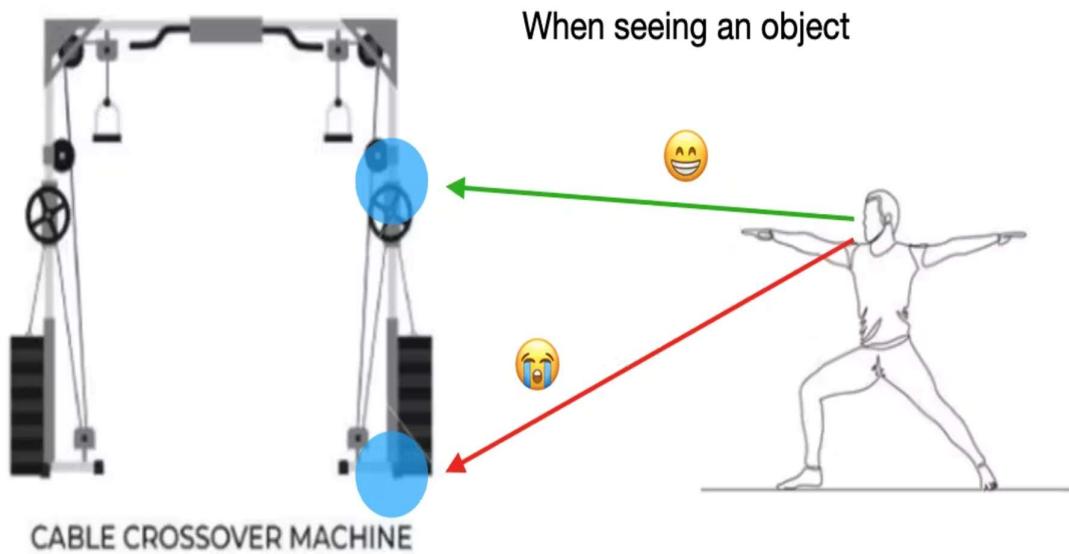


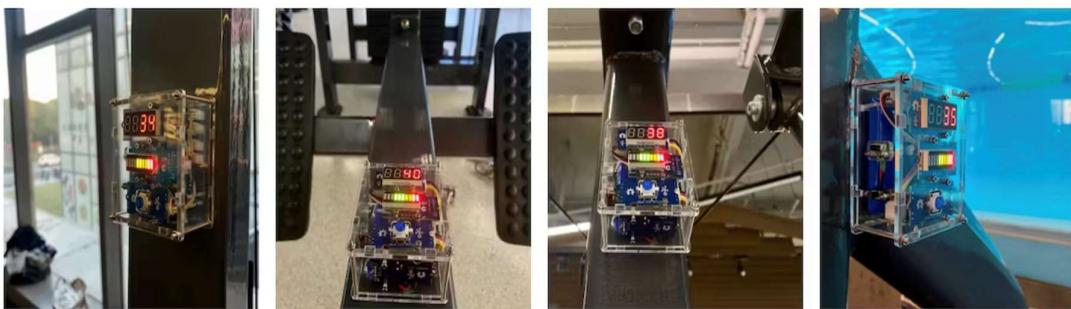
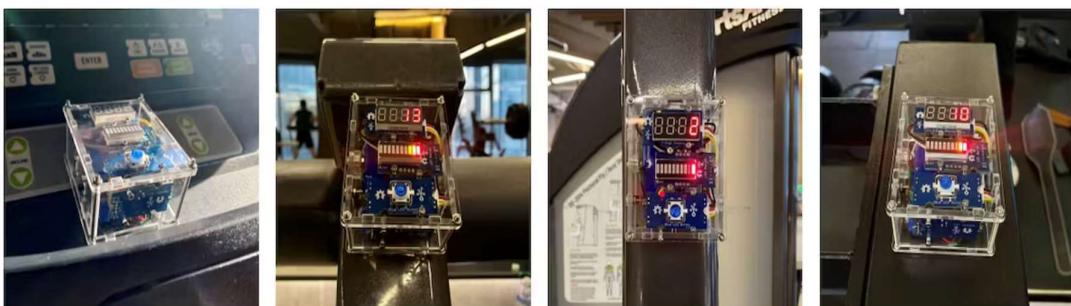
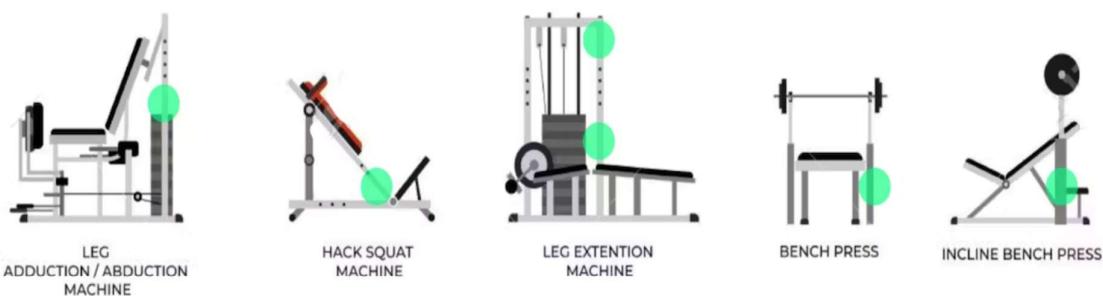
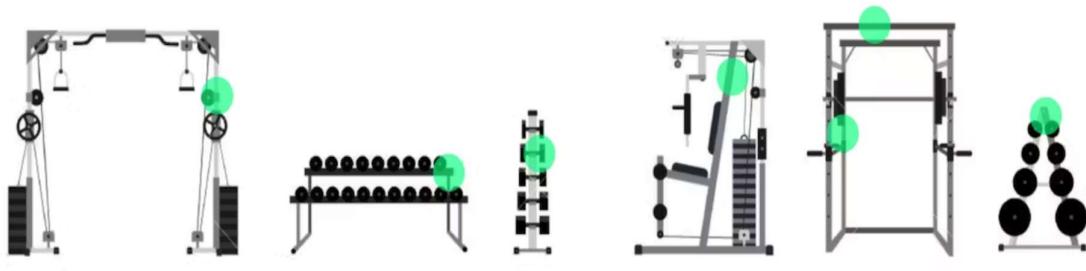
**Funzionamento a due dita:** La scelta di progettazione finale è stata quella di consentire agli utenti di tenerlo facilmente con una mano e di azionarlo con due dita.





**Fissaggio magnetico:** Dopo aver analizzato i punti critici, si è deciso di utilizzare magneti per fissare il prodotto nei punti in cui l'interazione e il funzionamento sono più facilmente realizzabili.





### 5.1.18 Seeed Fusion DIY XIAO Mecha

<https://www.seeedstudio.com/seeed-fusion-diy-xiao-mechanical-keyboard-contest.html>

XIAO 的小巧尺寸与其强悍的性能，没想到在 DIY 键盘与控制器玩家中得到认可，为此 Seeed 在2022年7月至10月，组织了一次 Fusion XIAO

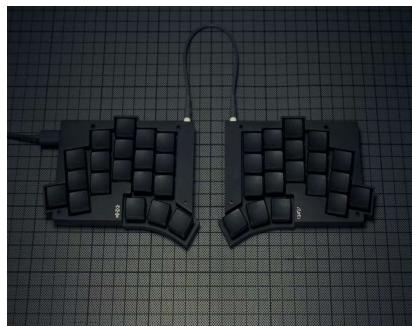
机器键盘大赛，下面我们展示了此次比赛的一些获奖项目，以帮助对 DIY 键盘有兴趣的读者。

### 1° premio: TOTEM | una minuscola tastiera divisa con splay

(2x)19 tasti ergo split: cluster di 3 tasti per il pollice, splay per il mignolo, profilo basso. Repository utile e custodia elegante e unica. Ben documentata e open source. Ed è una tastiera utilizzabile, che potrebbe essere usata quotidianamente. A parte questo, Marc ha fatto un grande sforzo per presentare il suo design esteticamente

<https://www.hackster.io/geist/totem-a-tiny-splitkeyboard-with-splay-cb2e43>

Autore: Marc Rühl



### 2° premio: Beyblock20 | un MacroPad magnetico e modulare

<https://github.com/ChrisChrisLoLo/beyblock20>

Autore: Christian Lo



**2° premio: Purple Owl | una tastiera al 60% con tecnologia Seeed XIAO RP2040**

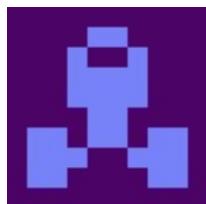
<https://www.hackster.io/sonalpinto/purple-owl-a-60-keyboard-powered-by-seeed-xiao-rp2040-f73604>



Autore: Sonal Pinto

**3° premio: KLEIN | una tastiera ergonomicica wireless**

<https://www.hackster.io/nosnk/klein-a-wireless-ergonomical-keyboard-b4cd9a>



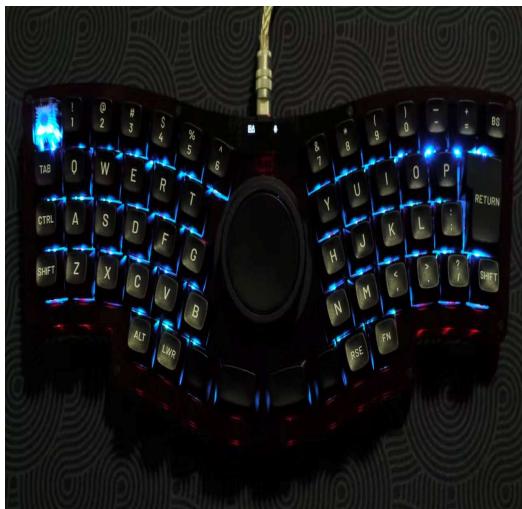
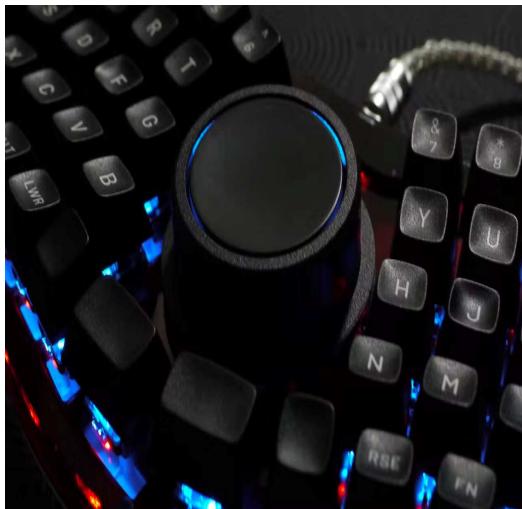
Autore: Shashank

**3° premio: GRIN Quern | una tastiera ergonomicica sul trackpad centrale**

<https://www.hackster.io/policium/grin-quern-ergonomic-keyboard-on-center-trackpad-8b58c3>



Autore: policum



### 3° premio: Kidoairaku Swallowtail | una simpatica tastiera a forma di farfalla

Autore: yswallow



## Riferimenti

### Informazioni sugli autori

**Lei Feng** è il responsabile del gruppo di supporto tecnico e del curriculum di prodotto presso Seeed Studio. Autore esperto nei settori dell'hardware open source e dell'edge computing, ha pubblicato diversi libri in Cina, tra cui "GameGo Beginner Programming Course for Arcade 《做游戏，玩编程——零基础开发微软 Arcade 掌机游戏》," "Grove Beginner Kit For Arduino - Codecraft Graphical Programming Course 《Arduino 图形化编程轻松学》," e la traduzione cinese di "IoT for Beginners 《深入浅出 IoT : 完整项目通关实战》" col supporto di Microsoft Cina.

Lei Feng ha creato numerosi tutorial e documentazione open source in cinese e inglese con il suo team. La sua esperienza pratica nello sviluppo di progetti IoT ed edge computing gli offre spunti unici per semplificare concetti complessi per i principianti. Come scrittore coinvolgente e insegnante paziente, Lei Feng è la guida ideale per rendere Arduino e TinyML accessibili ai nuovi arrivati in tutto il mondo.

Profilo LinkedIn: <https://www.linkedin.com/in/leon-feng-a029bb1/>

**Marcelo Rovai** è una figura riconosciuta nell'ambito dell'ingegneria e dell'istruzione tecnologica, insignito del titolo di Professore Honoris Causa dall'Università Federale di Itajubá, Brasile. Il suo background formativo comprende una laurea in Ingegneria presso UNIFEI e una specializzazione avanzata presso la Scuola Politecnica dell'Università di San Paolo. Per migliorare ulteriormente la sua competenza, ha conseguito un MBA presso IBMEC (INSPER) e un Master in Data Science presso l'Universidad del Desarrollo in Cile.

Con una carriera che ha toccato diverse aziende tecnologiche di alto profilo come AVIBRAS Airspace, ATT, NCR e IGT, dove ha ricoperto il ruolo di Vicepresidente per l'America Latina, porta con sé una vasta esperienza nel settore nei suoi sforzi accademici. È uno scrittore prolifico su argomenti correlati all'elettronica e condivide le sue conoscenze attraverso piattaforme aperte come Hackster.io.

Oltre ai suoi interessi professionali, si dedica alla divulgazione educativa, prestando servizio come professore volontario presso l'UNIFEI e collaborando con il gruppo TinyML4D come co-

presidente, promuovendo l'istruzione TinyML nei paesi in via di sviluppo. Il suo lavoro sottolinea l'impegno a sfruttare la tecnologia per il progresso della società.

*Profilo LinkedIn:* <https://www.linkedin.com/in/marcelo-jose-rovai-brazil-chile/>