



Istituto Istruzione Superiore "LUDOVICO GEYMONAT"

ESAMI DI STATO 2020 – 2021

Elaborato per l'avvio del colloquio ai sensi della O.M. 53 del 3/03/2021

CANDIDATO: Federico Vittorio Chiodo

CLASSE: 5E

Casa domotica con ESP32



federicochiodo.it@gmail.com

Indice

INTRODUZIONE E SPIEGAZIONE DEL PROGETTO	3
INTRODUZIONE:	3
SCHEMA A BLOCCHI:	3
FUNZIONI CARATTERIZZANTI:	4
FUNZIONAMENTO:	5
LISTA COMPONENTI:	9
COMPONENTI UTILIZZATI	10
ESP32:	10
<i>Gestione dei consumi</i>	10
<i>Periferiche</i>	11
<i>Pinout ESP32 utilizzato</i>	12
ESP 8266:	13
<i>Gestione dei consumi</i>	13
<i>Periferiche</i>	14
<i>Pinout ESP 8266 NodeMCU</i>	15
DIFFERENZE FRA ARDUINO UNO E UN ESP32	15
DHT 11:	15
DISPLAY LCD IPS:	16
MODULO RELÈ:	17
TRASMETTITORE IR:	18
ROTARY ENCODER:	19
SPIEGAZIONE DELLE TECNOLOGIE DI COMUNICAZIONE UTILIZZATE	21
WiFi:	21
ESP-NOW:	22
CODICE	23
LIBRERIE UTILIZZATE:	23
PROBLEMATICHE RISCONTRATE:	24
BIBLIOGRAFIA:	24
ALLEGATI:	25

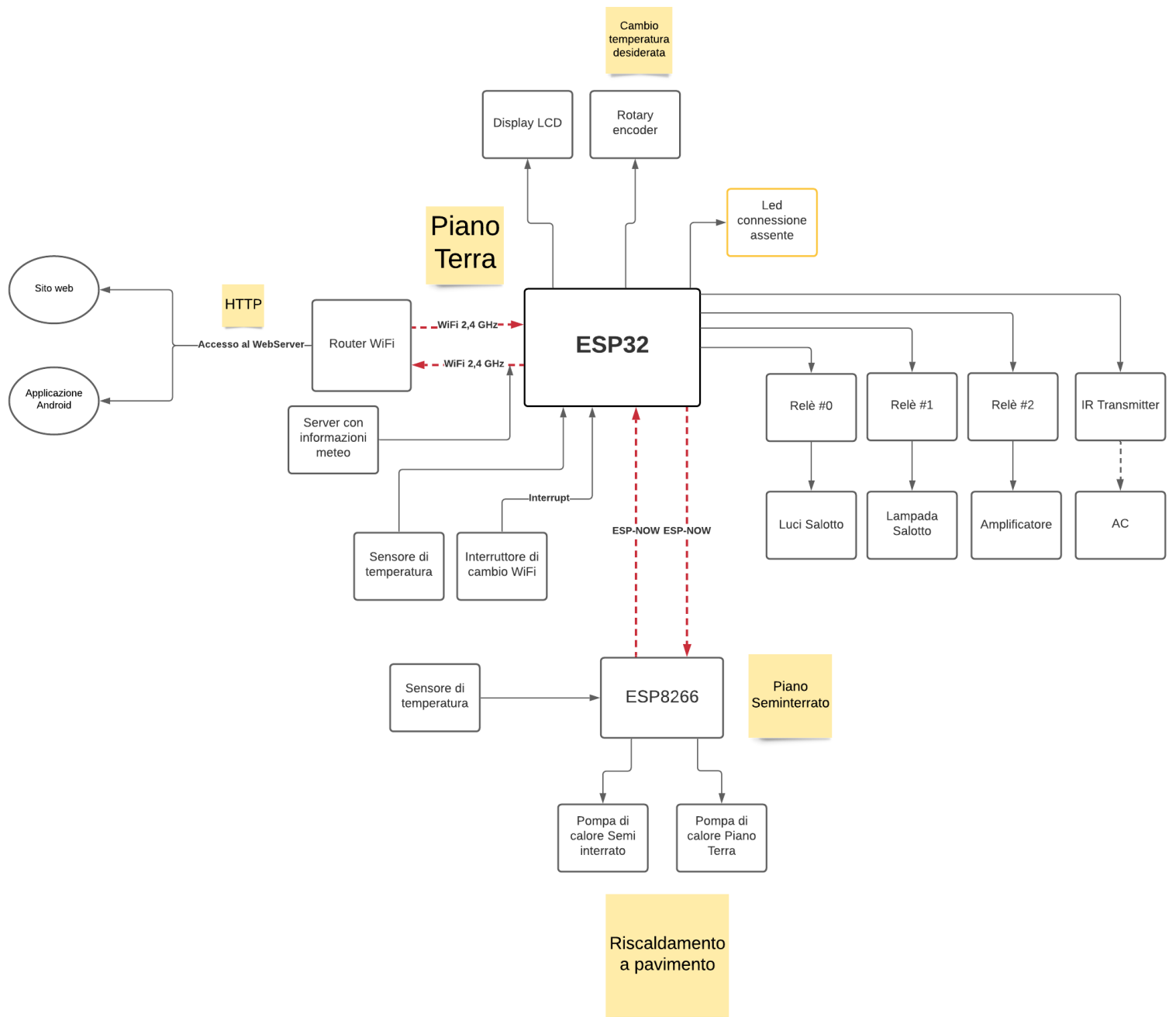
Introduzione e spiegazione del progetto

Introduzione:

Il progetto ha l'obiettivo di creare una casa intelligente, cioè di rendere smart la propria casa; quindi sarà possibile gestire automaticamente l'accensione del riscaldamento di una casa a due piani (Piano terra e piano seminterrato) tramite una unità di elaborazione che si interfacerà con le pompe di calore, che verrà comandato a sua volta da una centralina posta nel salotto del piano terra, dove, da essa sarà possibile visualizzare, tramite un display LCD, la temperatura corrente del piano e se in quel momento il riscaldamento è acceso. Sarà anche possibile accedere a più funzioni tramite la pagina web (connettendo la centralina alla rete WiFi di casa) o tramite l'applicazione Android, dove sarà possibile visualizzare le temperature di entrambi i piani, il meteo corrente e accendere o spegnere dei dispositivi domestici collegati ad essa.

Schema a blocchi:

Dall'immagine sottostante si potrà notare uno schema a blocchi per identificare come verranno collegati i sensori o attuatori e come i microcontrollori comunicano fra di loro.



Funzionamento a blocchi

Funzioni caratterizzanti:

Centralina (piano terra):

- **Web Server** per gestire le richieste dall'utente di pagine web utilizzando il protocollo HTTP.
- **DHT11** per la lettura della temperatura.

- **Display LCD** per la possibilità di visualizzare la temperatura corrente del piano e il relativo riscaldamento.
- Utilizzo di un **rotary encoder** per la modifica della temperatura desiderata nel piano corrente.
- Connessione WiFi alla rete domestica per visualizzare, tramite pagina web o tramite l'applicazione, le temperature dei piani, il meteo corrente e l'accensione o spegnimento di dispositivi domestici collegati ad essa.
- **Led giallo** per segnalare un errore nella connessione alla rete WiFi (Connessione assente).
- **Pulsante** per la modifica della rete WiFi: dopo 10 secondi dall'azionamento del bottone la centralina si disconetterà dalla rete wireless e ne creerà una, dove l'utente potrà collegarsi e, tramite una pagina web si potrà selezionare e immettere le credenziali della propria rete. Le credenziali, se corrette, verranno salvate nella memoria FLASH, e quindi anche se la centralina verrà spenta, le credenziali rimarranno salvate.
- **Modulo relè** per l'accensione o lo spegnimento di elettrodomestici collegati ad essa.
- **Diodo IR** per l'accensione o lo spegnimento di elettrodomestici controllabili via infrarossi.
- **ESP-NOW** per lo scambio di dati (temperatura e accensione del riscaldamento) con la centralina del piano seminterrato.

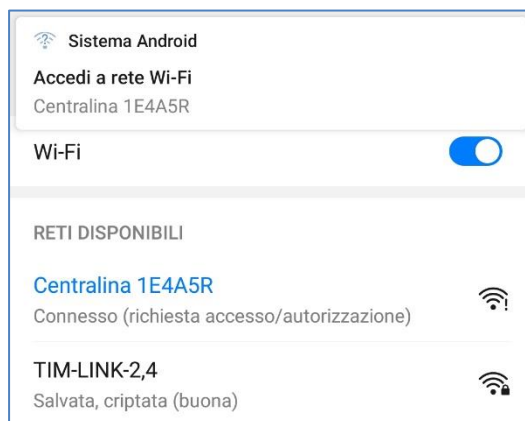
Centralina (piano seminterrato):

- **DHT11** per la lettura della temperatura.
- **ESP-NOW** per lo scambio di dati (temperatura e accensione del riscaldamento) con la centralina del piano terra.
- Gestione delle **pompe di calore**.

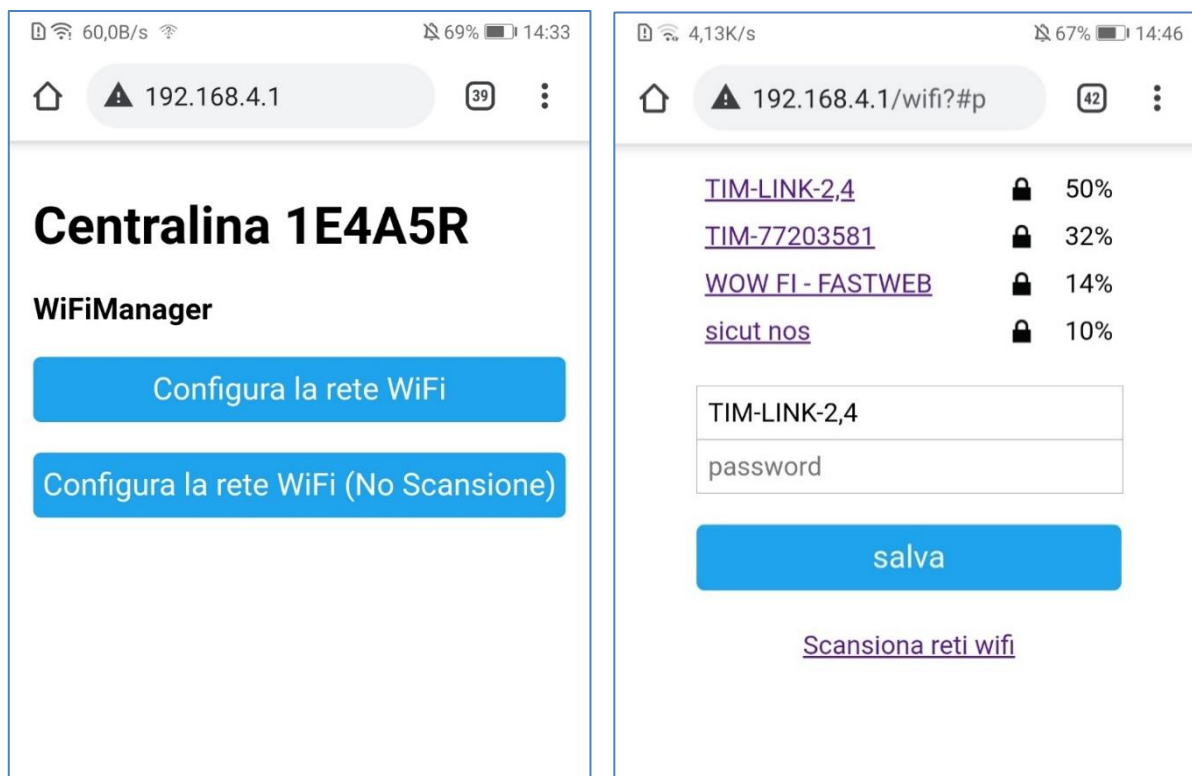
Funzionamento:

Al primo avvio della centralina del piano terra sarà necessario configurare le credenziali per la connessione alla rete WiFi di casa, come richiesto sul display LCD. Per configurare le credenziali sarà necessario accedere via smartphone alla

rete WiFi “Centralina 1E4A5R”. A connessione avvenuta apparirà un popup nel dispositivo connesso per configurare l’accesso; il popup sarà il seguente:

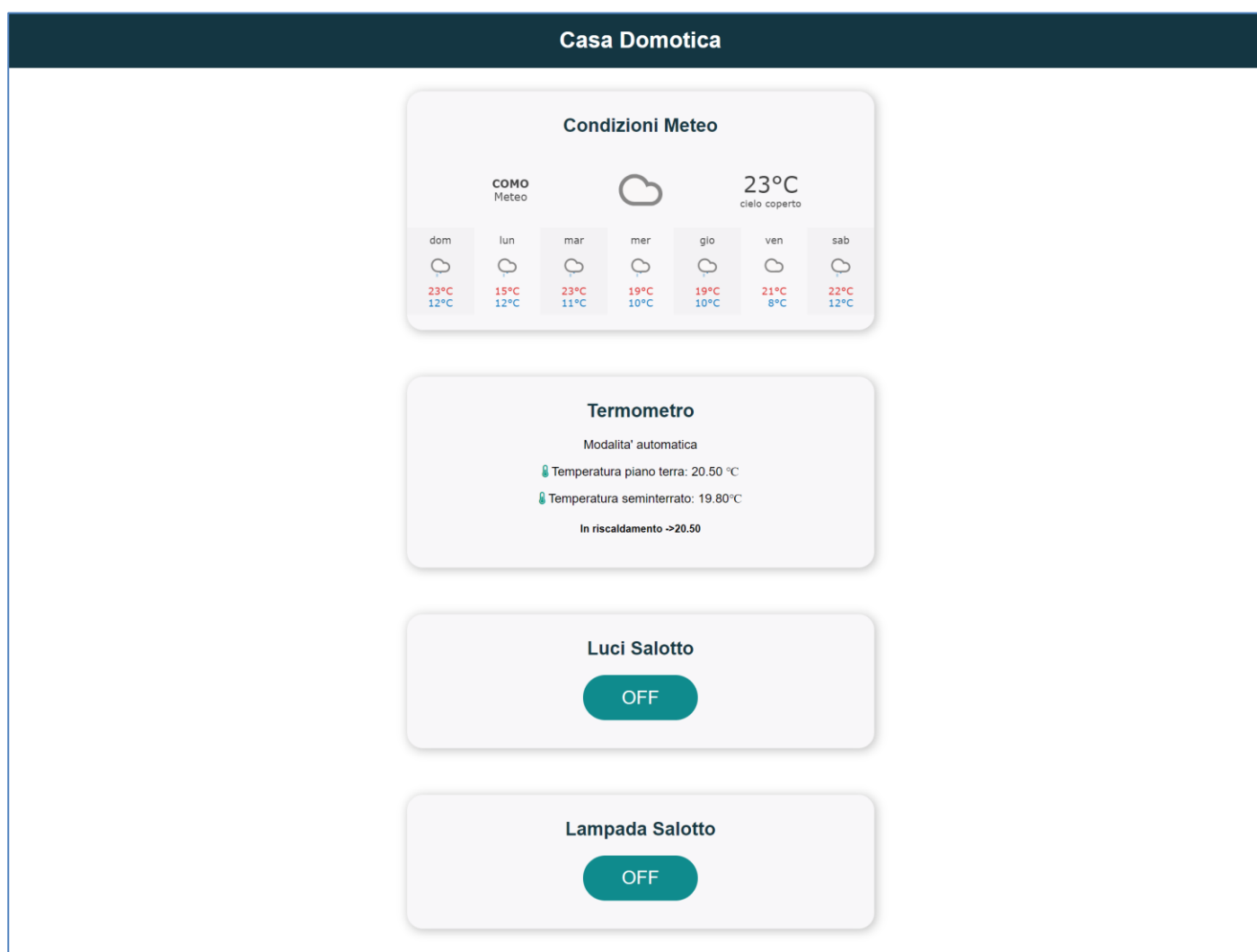


Sarà necessario cliccare su di esso per essere portati alla pagina web di configurazione della centralina, chiamata anche pagina web Captive portal; essa sarà necessaria per l'immissione delle credenziali della rete su cui dovrà connettersi.



Ora si presenterà una schermata come l’immagine di sinistra, allora, schiacciando sul pulsante “Configura la rete WiFi”, la centralina farà una scansione delle reti WiFi nelle vicinanze, come si può vedere nella figura di destra. Adesso, selezionando

uno dei seguenti access point e immettendo la password sarà possibile effettuare la connessione alla rete wireless desiderata. Se la connessione è avvenuta con successo verrà stampato sul display l'indirizzo IP ottenuto, che servirà all'applicazione per poter connettersi ad esso. Per vedere la pagina web via PC o laptop è possibile utilizzare i cosiddetti "Domini locali" con il protocollo mDNS, cioè dei domini disponibili soltanto localmente, cioè che invece di inserire un determinato IP è possibile inserire un dominio, che in questo caso sarà "casadomotica.local". Collegandosi all'indirizzo sarà possibile trovare questa schermata:



Dalla figura si può osservare la sezione meteo, sempre aggiornata e localizzata in base alla posizione, la sezione per la gestione del riscaldamento e la sezione dedicata alla gestione dei dispositivi collegati alla centralina tramite il modulo relè o anche i dispositivi controllabili tramite infrarossi, quali per esempio il condizionatore e l'amplificatore stereo.

La gestione del riscaldamento, essendo automatica, non richiede nessuna interazione da parte dell'utente e, tramite un controllo della temperatura a doppia soglia, riesce ad ottimizzare l'accensione delle pompe di calore, evitando, in caso di correnti di aria fredde a intermittenza, di azionare il riscaldamento.

Per collegarsi invece dall'applicazione sarà necessario installare l'APK fornito

(Funziona solo per Android) e localizzare l'applicazione (



). Alla prima

apertura sarà necessario immettere l'indirizzo IP premendo il tasto "indietro". Dopo l'immissione dell'indirizzo sarà possibile accedere alla pagina web.



Se si ha la necessità di cambiare la rete su cui connettere la centralina è possibile tener premuto per più di 10 secondi il pulsante presente all'interno della scatola da

incasso a muro per eliminare le credenziali correnti e avviare il processo di immissione delle credenziali descritto prima.

Lista componenti:

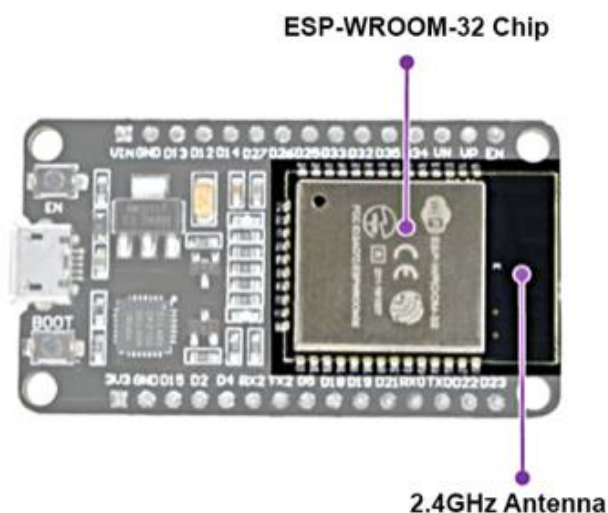
Componente	Caratteristiche	Quantità
ESP-32 (Centralina piano terra)	Dual core, WiFi 2.4GHz, Bluetooth 4.2 Versione WROVER	x1
ESP-8266 (Centralina seminterrato)	WiFi 2.4GHz, Bluetooth 4.2 Versione NodeMCU	x1
DHT-11	Accuratezza $\pm 2^{\circ}\text{C}$	x2
Bottone		x1
Display LCD	Modulo display 1.28" IPS TFT 240x240 Rotondo	x1
Rotary Encoder		x1
Modulo relè	4 Relè integrati	x1
Pompa di calore	220V	x2
Led Giallo	2,2V	x1
Trasmittitore IR	~3,5V	x1
Resistenza	220 Ω	x1

Componenti utilizzati

ESP32:

ESP-WROOM-32 Chip

- Xtensa® Dual-Core LX6 a 32 bit
- Freq CPU fino a 240 MHz.
- SRAM interna da 520 kb
- Flash esterno da 4 MB
- Ricetrasmittitore Wi-Fi 802.11b/g/n
- Bluetooth 4.2/BLE



La scheda di sviluppo equipaggia il modulo ESP-WROOM-32 contenente un microprocessore Dual-Core LX6 a 32 bit Tensilica Xtensa. Questo processore è simile all'ESP8266 ma ha un core aggiuntivo (possono essere controllati individualmente) e funziona a una frequenza di clock regolabile da 80 a 240 MHz; presenta anche 448 KB di ROM, 520 KB di SRAM e 4 MB di memoria Flash.

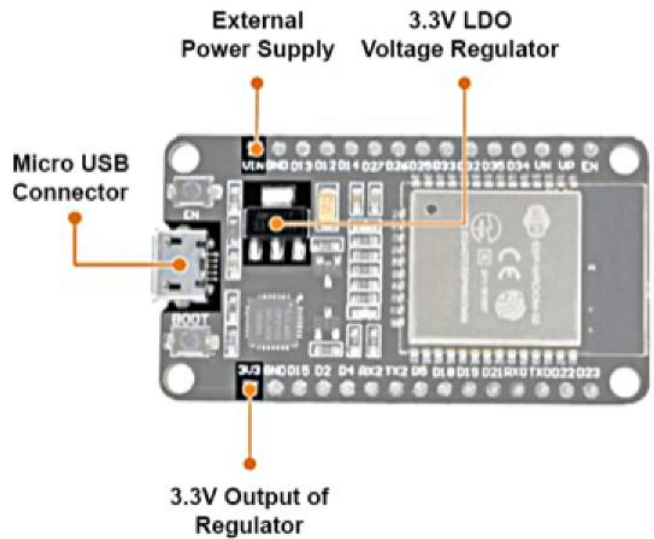
Esso integra il ricetrasmittitore HT40 con Wi-Fi 802.11 b/g/n, in modo che non solo possa connettersi a una rete WiFi e interagire con Internet, ma anche configurare una propria rete, consentendo ad altri dispositivi di connettersi direttamente ad essa. L'ESP32 supporta anche WiFi Direct, che è una buona opzione per la connessione peer-to-peer senza la necessità di un punto di accesso. Il chip ha anche funzionalità Bluetooth dual mode, il che significa anche il Bluetooth Low-Energy (Bluetooth LE), rendendolo ancora più versatile nel mondo IoT.

Gestione dei consumi

Poiché l'intervallo di tensione operativa di ESP32 è da 2,2 V a 3,6V, la scheda è dotata di un regolatore di tensione LDO (Low-dropout regulator) per mantenere la tensione costante a 3,3V. L'uscita del regolatore è anche suddivisa su uno dei lati della scheda ed etichettata come 3V3 e può fornire in modo affidabile fino a 600 mA e quindi può essere utilizzato per fornire alimentazione a componenti esterni.

Fabbisogno energetico

- Tensione operativa: da 2,2 V a 3,6V
- Regolatore di tensione da 3,3 V 600mA
- 5 μ A durante la modalità sospensione
- 250mA durante le trasmissioni RF



L'alimentazione della scheda di sviluppo ESP32 viene fornita tramite il connettore USB MicroB integrato. In alternativa, se si dispone di una sorgente di tensione 5V regolata, il pin VIN può essere utilizzato per fornire direttamente l'ESP32 e le sue periferiche.

La corrente di idle del chip è inferiore a 5 μ A, rendendolo adatto per qualsiasi applicazione elettronica alimentate a batteria.

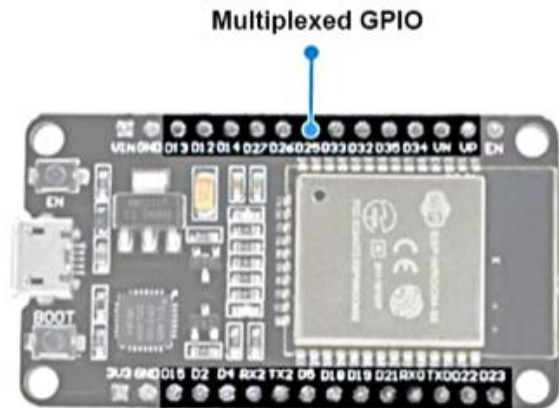
Periferiche

Esso presenta un totale di 48 pin GPIO, a cui possono essere assegnati:

- 15 canali ADC - 15 canali di ADC SAR a 12 bit. L'intervallo ADC può essere impostato, in firmware, su 0-1V, 0-1.4V, 0-2V o 0-4V
- 2 interfacce UART - 2 interfacce UART. Uno viene utilizzato per caricare il codice in serie. Sono dotati di controllo del flusso e supportano anche IrDA (Comunicazione a infrarossi).
- 25 uscite PWM - 25 canali che supportano la modulazione PWM.
- 2 canali DAC - DAC a 8 bit per produrre tensioni analogiche.
- Interfaccia SPI, I2C e I2S (Per audio).
- 9 Touch Pad - 9 GPIO sono dotati di rilevamento tattile capacitivo.

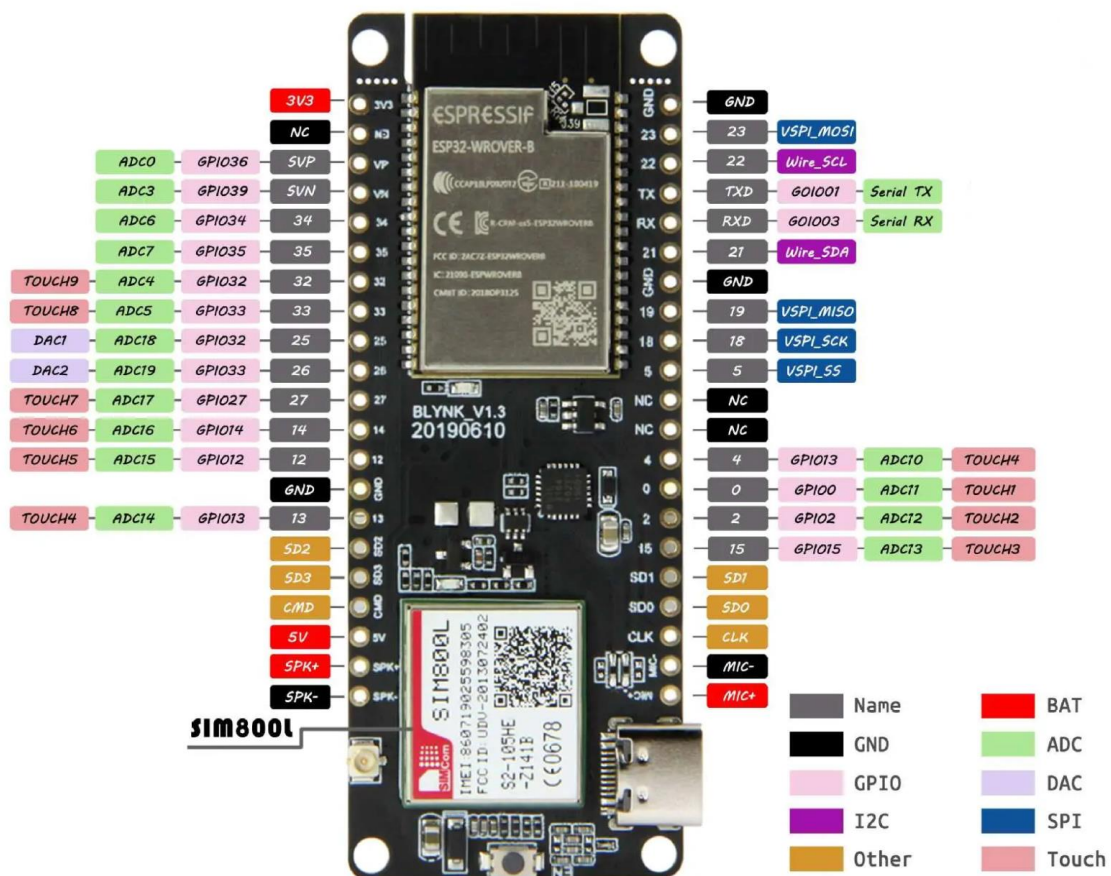
I/O multiplexati

- 15 canali ADC
- 2 interfacce UART
- 25 uscite PWM
- 2 canali DAC
- Interfaccia SPI, I2C e I2S
- 9 Touch Pad



Grazie alla funzione di multiplexing pin dell'ESP32 (periferiche multiple multiplexate su un singolo pin GPIO), ogni singolo pin GPIO, può fungere da ingresso ADC, uscita DAC o Touch pad.

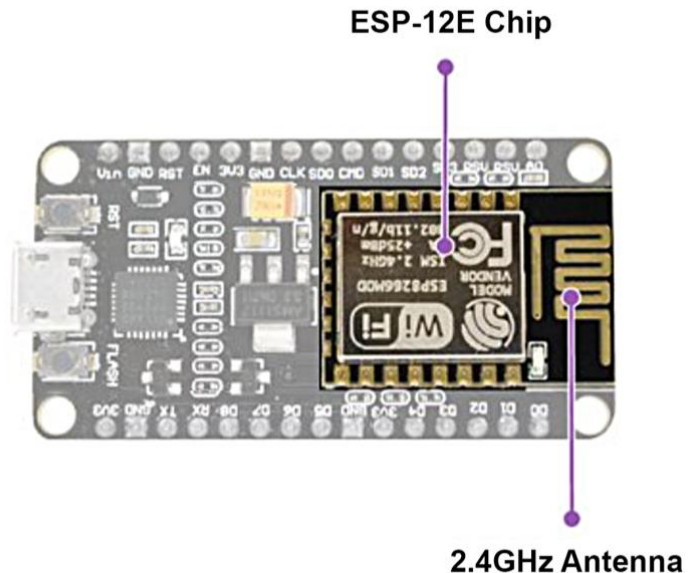
Pinout ESP32 utilizzato



ESP 8266:

ESP-12E Chip

- Tensilica Xtensa® LX106 a 32 bit
- Freq CPU da 80 a 160 MHz.
- 128 kb RAM interna
- Flash esterno da 4 MB
- Ricetrasmittitore Wi-Fi 802.11b/g/n



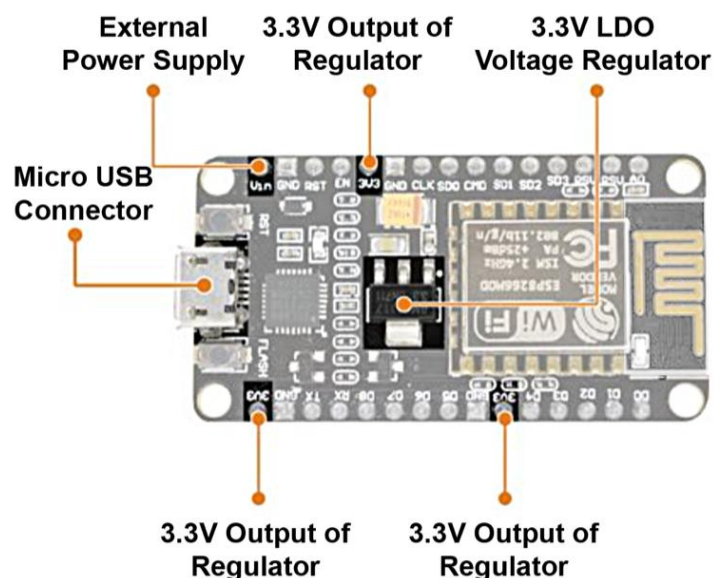
La scheda di sviluppo equipaggia un chip ESP-12E con un microprocessore Tensilica Xtensa RISC LX106 a 32 bit ad un singolo core che funziona a una frequenza di clock regolabile da 80 a 160 MHz e supporta RTOS (real-time operating system), insieme a 128 KB di RAM e 4 MB di memoria Flash.

Gestione dei consumi

Entrambi i microcontrollori lavorano fino a 3.6V, ma si differenziano per la loro corrente consumata in modalità sleep o di sospensione pari a 20 μ A, che risulta molto maggiore rispetto alla sua versione aggiornata. Inoltre una maggior tensione minima richiesta dall'ESP 8266 lo rende meno predisposto all'uso con batterie.

Fabbisogno energetico

- Tensione operativa: da 2,5 V a 3,6V
- Regolatore di tensione da 3,3 V 600mA
- Corrente operativa di 80 mA
- 20 μ A durante la modalità sospensione



L'alimentazione del NodeMCU ESP8266 viene fornita tramite il connettore USB Micro B integrato. In alternativa, se si dispone di una sorgente di tensione 5V regolata, il pin VIN può essere utilizzato per alimentare il controllore direttamente.

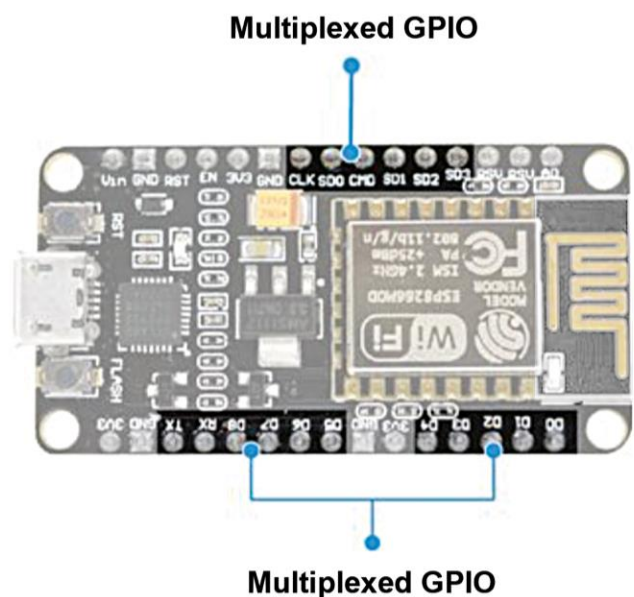
Periferiche

Esso ha un totale di 17 pin GPIO suddivisi per le intestazioni pin su entrambi i lati della scheda di sviluppo. Questi pin possono essere assegnati:

- Canale ADC - Canale ADC a 10 bit.
- Interfaccia UART utilizzata per caricare il codice in serie.
- Uscite PWM - Pin in modulazione PWM.
- Interfaccia SPI, I2C e I2S (Per audio).

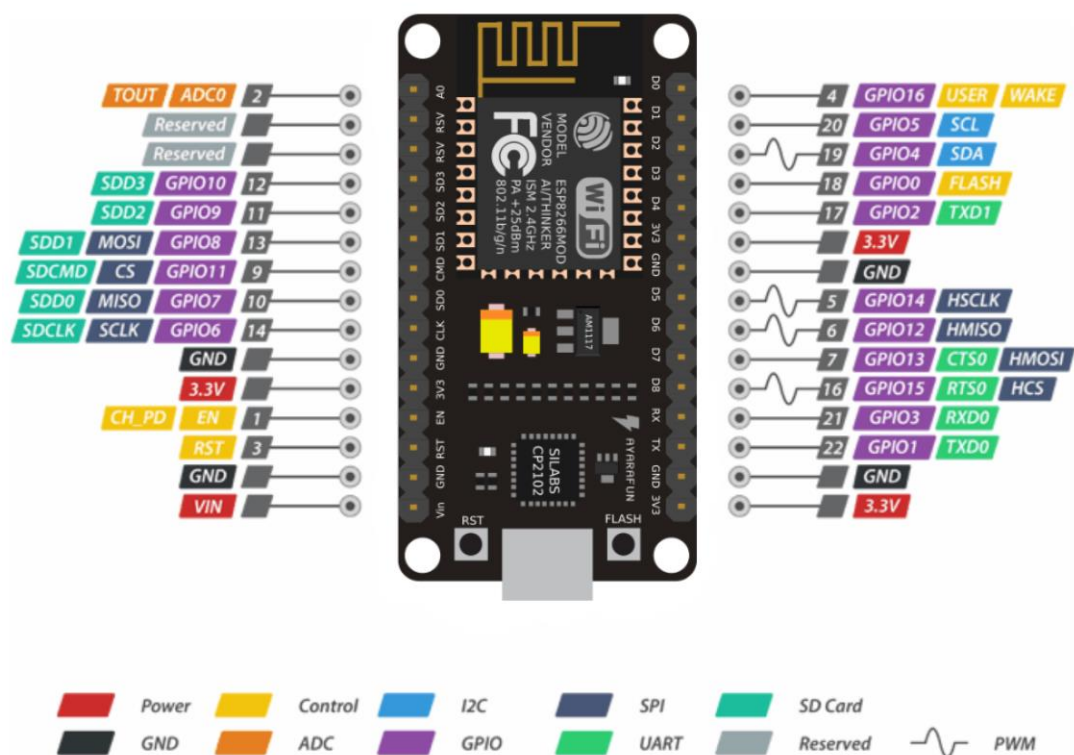
I/O multiplexati

- 1 canali ADC
- 2 interfacce UART
- 4 uscite PWM
- Interfaccia SPI, I2C e I2S



Grazie alla funzione di multiplexing (periferiche multiple multiplexate su un singolo pin GPIO) i pin GPIO dell'ESP8266, possono fungere da uscita PWM, UART o SPI.

Pinout ESP 8266 NodeMCU



Il DHT11, presenta due sensori, uno per la temperatura (un termistore resistivo di tipo NTC) e l'altro per l'umidità, anch'esso di tipo resistivo.

Questo sensore fornisce un'uscita in formato digitale, grazie all'ADC (Analogic-Digital converter) che converte il valore analogico fornito dai sensori in formato a 8 bit, il sensore ha una precisione della temperatura piuttosto accurata ($\pm 2^{\circ}\text{C}$) e una precisione discreta dell'umidità ($\pm 5\%$), utilizzando nel contempo solo 3 pin, Vcc (Da 3V a 5,5V), GND e il pin Dati. Se si ha però un cavo dati lungo qualche decina di metri, è consigliabile aggiungere una resistenza da $5\text{K}\Omega$, tra Vcc e il pin dati per compensare la dispersione della corrente causata dal mezzo trasmissivo.

Come detto in precedenza il DHT11 utilizza i termistori, che sono dei trasduttori di temperatura, la cui resistenza varia al variare della temperatura.

Si dividono in:

- NTC (Negative Temperature Coefficient): la cui resistenza diminuisce all'aumentare della temperatura.
- PTC (Positive Temperature Coefficient): la cui resistenza aumenta all'aumentare della temperatura.

Per il calcolo della loro resistenza vale la seguente formula:

$$R_f = R_i \cdot e^{B\left(\frac{1}{T_f} - \frac{1}{T_i}\right)}$$

Dove R_i è la resistenza a zero gradi, B è la costante di temperatura che varia da 2000°K a 5000°K , T_f è la temperatura finale, T_i è la temperatura iniziale.

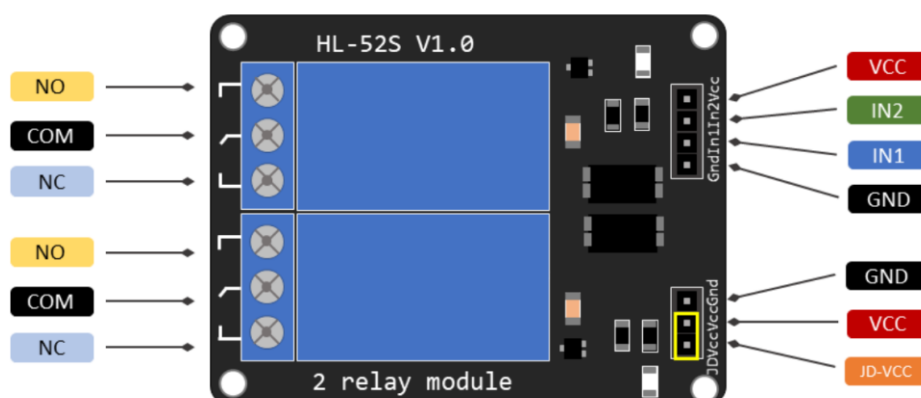
Display LCD IPS:



Il display LCD ha una risoluzione di 240x240 pixel e ha un diametro di 1.28". Utilizza un pannello IPS a colori, che, grazie al controller GC9401, è facilmente controllabile tramite un'interfaccia SPI a 4 fili (Vcc, GND, SCL e SDA), sono disponibili altri 4 connettori per il reset e il controllo del backlight. Richiede una tensione massima di 3.3V e ha una luminosità massima di 400 Cd/m². Per la visualizzazione di testo o immagini saranno necessarie librerie supplementari, quali:

- Arduino GFX → Per la visualizzazione di testo
- PNGLE → Per la visualizzazione di immagini
- JPEGDEC → Per ottimizzare la codifica delle immagini JPEG.

Modulo Relè:



Un relè è un interruttore azionato elettricamente e, come qualsiasi altro interruttore, può essere acceso o spento, lasciando passare o meno la corrente. Può essere controllato con basse tensioni, come i 3.3V forniti dall'ESP32 e ci permette di controllare alte tensioni come 12V, 24V o 220V AC. Quindi esso ci permette di gestire con sicurezza un carico di tensioni non sopportabili da un qualsiasi microcontrollore; per azionarlo però necessita di una corrente di circa 80 mA e un qualsiasi GPIO non riesce a soddisfare questo requisito, quindi nel modulo si è ricorso all'uso di transistor, così tramite i pin di comando si potrà controllare il transistor (S8050 Versione SMD) che, con il suo metodo di conduzione NPN, è perfetto per gestire come un pulsante, il passaggio di corrente all'interno della bobina del relè. Utilizzandolo come interruttore ON-OFF si dovrà portare il transistor in saturazione. Essendo comandato in corrente, bisognerà prima sapere quanta

corrente richiede la bobina del relè, che è possibile trovarla nel suo datasheet e sapendo l'HFE si riuscirà a calcolare la corrente di base necessaria per condurre elettricità secondo la formula: $I_C = h_{FE} I_B$.

Il modulo relè implementa anche funzioni di sicurezza, quali l'utilizzo in parallelo della bobina di un diodo (a verso invertito e diodo di tipo Schottky o a commutazione rapida), perché un induttore (bobina del relè) non può perdere istantaneamente la sua corrente, allora il diodo flyback fornisce un percorso per la corrente quando la bobina è spenta. Altrimenti, si verificherebbe un picco di tensione che causerà un arco sui contatti di commutazione e che, eventualmente distruggerà i transistor di commutazione. Altre feature di sicurezza sono la presenza di un fotoaccoppiatore, utilizzato per trasferire un segnale, digitale o analogico, da un apparato ad un altro, tenendoli elettricamente isolati l'uno dall'altro. Questo fotoaccoppiatore sarà utilizzato per separare il circuito del relè dal microcontrollore e prevenire, in caso di malfunzionamento del modulo, una corrente inversa che potrebbe distruggere anche il microcontrollore.

Trasmittitore IR:

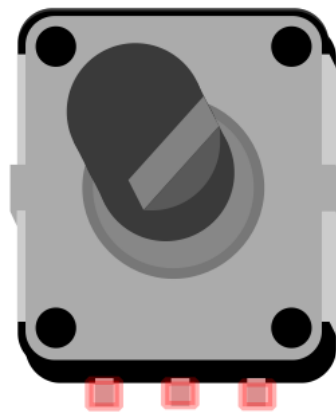


Il trasmettitore a infrarossi è comunemente usato per controllare i dispositivi elettronici (televisori, AC) in modalità wireless, principalmente attraverso un telecomando; esso emette luce a infrarossi, significa che emette luce nell'intervallo di frequenza infrarossa ($>3 \cdot 10^{12}$ Hz). Non possiamo vedere la luce infrarossa attraverso i nostri occhi, e quindi è invisibile agli occhi umani. La lunghezza d'onda dell'infrarosso (700nm – 1mm) è appena oltre la normale luce visibile. L'infrarosso

ha le stesse proprietà della luce visibile, come può essere focalizzato, riflesso e polarizzato come la luce visibile.

Oltre a emettere luce infrarossa invisibile, il LED IR consuma una corrente di 20mA e 3V come un normale LED; inoltre hanno un angolo di emissione della luce di circa 20-60 gradi e un intervallo da circa pochi centimetri a diversi metri, dipende dal tipo di trasmettitore IR e dal produttore. Alcuni trasmettitori professionali hanno un raggio d'azione di qualche chilometro.

Rotary Encoder:

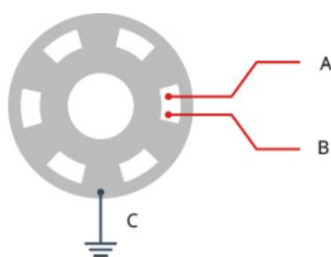


Un Rotary Encoder è un tipo di sensore di posizione che converte la posizione angolare (rotazione) di una manopola in un segnale di uscita utilizzato per determinare la direzione in cui la manopola viene ruotata.

Grazie alla loro robustezza e al fine controllo digitale; sono utilizzati in molte applicazioni tra cui nella robotica, nelle macchine CNC e nelle stampanti.

Esistono due tipi di codificatore rotativo: assoluto e incrementale. L'encoder assoluto ci fornisce la posizione esatta della manopola in gradi mentre l'encoder incrementale segnala quanti incrementi la manopola si è spostata.

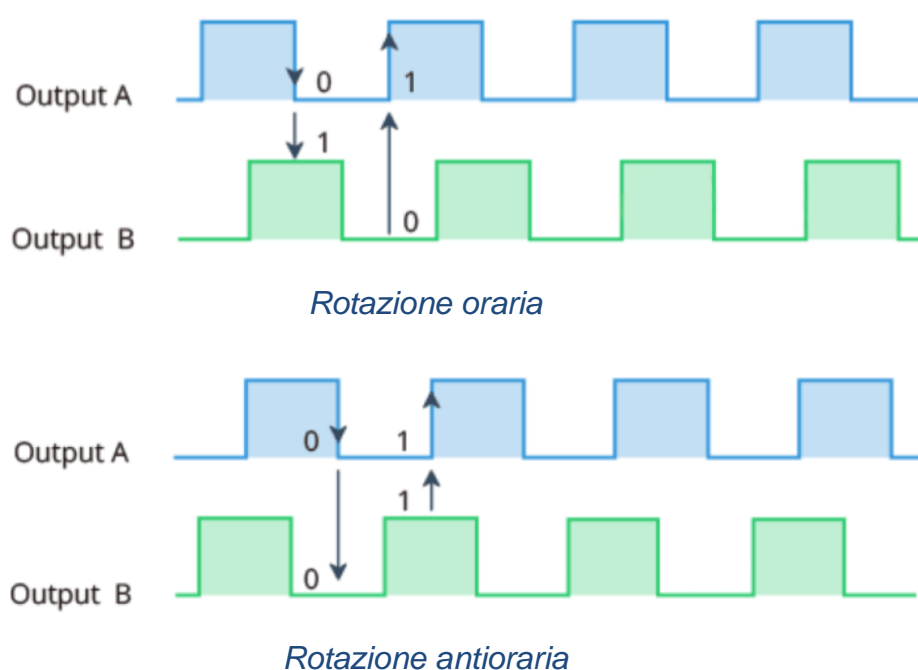
All'interno dell'encoder c'è un disco fessurato collegato al pin di terra comune C, e due pin di contatto A e B, come illustrato di seguito.



Quando si ruota la manopola, A e B entrano in contatto con il perno di terra comune C, in un ordine particolare in base alla direzione in cui si sta ruotando la manopola.

Quando entrano in contatto, emanano un segnale. Questi segnali vengono spostati di 90° fuori fase l'uno con l'altro quando un perno entra in contatto prima dell'altro perno. Questa è chiamata codifica a quadratura.

Come esempio si può vedere da questa figura che rappresenta come un microcontrollore può capire il verso di rotazione della manopola:



Spiegazione delle tecnologie di comunicazione utilizzate

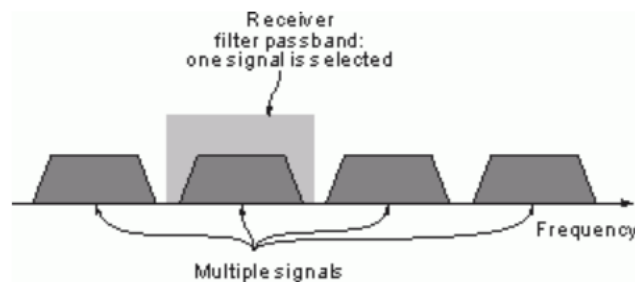
WiFi:

La connessione alla rete di casa avverrà tramite il protocollo WiFi 802.11n (2.4 GHz) che sfrutta la modulazione OFDM.

Sarà quindi necessario un Access Point per far accedere alla rete l'ESP-32, ovvero la centralina del piano terra. Essendo del secondo livello del modello ISO OSI, l'autenticazione alla rete non avverrà tramite un indirizzo IP, ma tramite l'indirizzo MAC della scheda di rete (Codice univoco assegnato a una interfaccia di rete). La prima fase per un dispositivo che si vuole connettere ad una rete wireless riguarda la richiesta di collegamento ad una specifica "stazione", se questa "stazione" è protetta da password si passa alla fase di autenticazione, dove il client chiederà l'associazione inviando le credenziali, quindi la password e se riceverà una risposta positiva sarà connesso alla rete. Dopo aver stabilito la connessione, il client, necessiterà di un indirizzo IP, allora avvierà la ricerca di un server DHCP trasmettendo a tutti gli host della rete la richiesta chiamata anche *DHCP Discovery*. Se un server DHCP è disponibile risponderà con un indirizzo IP libero (*DHCP Offer*). Come fattore di sicurezza, il client manderà una richiesta ARP (Address Resolution Protocol) all'indirizzo fornito per vedere se effettivamente quell'indirizzo è inutilizzato, se nessun host risponde, il client invierà la conferma dell'indirizzo fornito (*DHCP Request*) a questo punto la richiesta è finita e il client avrà un indirizzo IP.

Il protocollo 802.11n sfrutta, come detto in precedenza, la modulazione OFDM (Orthogonal Frequency Division Multiplexing) che presenta più portanti modulate e vicine fra di loro. Quando bisogna modulare un segnale vocale o un qualsiasi dato, viene assegnato a una delle portanti e le bande laterali si creeranno. Prima, quando più segnali venivano trasmessi, uno vicino all'altro, era necessario distanziarli per far in modo che il ricevitore, tramite l'uso di filtri e tramite la banda di guardia, riesca ad effettuare la demodulazione. Adesso con la OFDM anche se le bande laterali delle portanti si sovrappongono, i dati possono essere sempre ricevuti anche se

sono ortogonali tra di loro e senza interferenze, grazie alla spaziatura della portante che equivale al reciproco del simbolo del periodo.



Segnale OFDM

Un requisito fondamentale di questa modulazione è che i sistemi di ricezione e di trasmissione devono essere lineari. Se si ha una minima non-linearità potrebbe portare a interferenze fra le portanti e quindi a errori di trasmissione e ricezione.

ESP-NOW:



ESP-NOW è un protocollo sviluppato da Espressif, che consente a più dispositivi di comunicare tra loro senza utilizzare una rete WiFi comune e permette la creazione di una rete di microcontrollori interconnessi. Il protocollo è simile alla connettività wireless a bassa potenza a 2,4 GHz, che viene spesso distribuita nei mouse wireless. Quindi, l'associazione tra i dispositivi è necessaria prima della loro comunicazione. Al termine dell'associazione, la connessione è peer-to-peer, senza che sia necessario alcun hand-shake.

Questo protocollo permette anche uno scambio, chiamato connessione a due vie, dove due o più dispositivi si scambiano dati, comandandosi a vicenda e, in base al codice, permette uno scambio di informazioni sicuro, crittografando i dati trasmessi. Delle limitazioni di questo protocollo sono riguardanti al payload (carico dei dati)

massimo di 250 byte e riguardante i peer massimi con nessuna crittografia dei dati è di massimo 19 peer e con dati crittografati pari alla metà.

Come la comunicazione Bluetooth, si ha la presenza delle modalità di Master e Slave, in altri termini il “Capo” e lo “Schiavo” e quindi il “Capo” invierà dei dati allo “Schiavo” che poi li processerà. Esistono configurazioni multiple che presentano un master e più slave, più master e uno slave o, come detto prima, una configurazione dove più microcontrollori possono sia ricevere che inviare dati.

Per effettuare la comunicazione sarà necessario conoscere l'indirizzo MAC della scheda di rete del microcontrollore essendo una comunicazione del secondo livello del modello ISO OSI. È possibile, nel lato ricezione, filtrare i pacchetti per indirizzo MAC, così è possibile evitare che un dispositivo malevolo possa interferire.

Codice

Librerie utilizzate:

Tutte le librerie sono Open-Source (codice sorgente aperto) e possono essere trovate su [GitHub](#). Le librerie che sono state utilizzate sono le seguenti:

- [WiFiManager](#) → Libreria per la gestione delle credenziali della rete WiFi; se la rete su cui collegarsi non è stata definita, creerà una rete WiFi, dove si potrà, tramite una pagina web configurare le credenziali della rete.
- [WebServer](#) → Libreria necessaria dalla libreria appena descritta sopra, essa sarà necessaria per creare la pagina web di configurazione del WiFi.
- [DNSServer](#) → Libreria necessaria per la libreria WiFiManager, essa servirà per ridirezionare tutte le richieste alla pagina di configurazione.
- [WiFi](#) → Libreria per far funzionare le funzioni riguardanti alla connessione WiFi.
- [IRremote](#) → Libreria che permette, tramite un diodo IR, la codifica e l'invio di un segnale ad infrarossi.
- [ESP_NOW](#) → Libreria per le funzioni della comunicazione senza rete tramite il protocollo ESP-NOW.
- [DHT](#) → Libreria per le funzioni di
- [ESPmDNS](#) → Libreria per funzioni per l'implementazione dell'mDNS.

- [Preferences](#) → Libreria per funzioni di salvataggio di variabili su flash (variabili non volatili).
- [ArduinoGFX](#) → Libreria grafica con funzioni riguardanti l'inizializzazione del display e la visualizzazione di testo personalizzato.
- [Pngle](#) → Libreria per poter caricare su un display immagini png.

Problematiche riscontrate:

Sono state riscontrate delle limitazioni tecniche nella trasmissione dati tramite il protocollo ESP-NOW, quali una non perfetta retrocompatibilità fra ESP-32 e ESP-8266 e riguardo una limitazione dovuta alla trasmissione WiFi, dove, se un microcontrollore è connesso ad un access-point, la trasmissione con il protocollo ESP-NOW deve avvenire sullo stesso canale, altrimenti i dati non arriveranno al destinatario.

Altre limitazioni riguardano l'applicazione Android, dove sarà necessario inserire l'indirizzo IP della centralina principale perché il sistema operativo non risolve indirizzi tramite il protocollo mDNS, invece sarà possibile visualizzare la pagina web via Windows o MacOS tramite il dominio locale.

Bibliografia:

Fonti utilizzate:

- [ESP32: ESP-NOW and Wi-Fi Web Server Dashboard \(Arduino\) | Random Nerd Tutorials](#)
- [Getting Started with ESP-NOW \(ESP32 with Arduino IDE\) | Random Nerd Tutorials](#)
- [How to connect ESP32 and ESP8266 using ESP-Now protocol \(survivingwithandroid.com\)](#)
- [mDNS: come funziona il DNS Multicast? - IONOS](#)
- [mDNS Service - ESP32 - — ESP-IDF Programming Guide latest documentation \(espressif.com\)](#)
- <https://lastminuteengineers.com/>

- [OFDM and OFDMA Subcarriers - What Are the Differences? \(extremenetworks.com\)](http://extremenetworks.com)
- [OFDM \(Orthogonal Frequency Division Multiplexing\) - CableFree](http://CableFree)
- [Orthogonal frequency-division multiplexing - Wikipedia](http://Wikipedia)
- [adafruit/DHT-sensor-library: Arduino library for DHT11, DHT22, etc Temperature & Humidity Sensors \(github.com\)](https://github.com/adafruit/DHT-sensor-library)
- [arduino-esp32/libraries/ESPmDNS at master · espressif/arduino-esp32 \(github.com\)](https://github.com/espressif/arduino-esp32)
- [moononournation/Arduino GFX: Arduino GFX developing for various color displays and various data bus interfaces \(github.com\)](https://github.com/moononournation/Arduino-GFX)
- [kikuchan/pngle: Pngle - PNG Loader for Embedding \(github.com\)](https://github.com/kikuchan/pngle)
- [zhouhan0126/DNSServer---esp32 \(github.com\)](https://github.com/zhouhan0126/DNSServer-esp32)
- [zhouhan0126/WebServer-esp32: WebServer-esp32 \(github.com\)](https://github.com/zhouhan0126/WebServer-esp32)
- [HTTP Server - ESP32 - — ESP-IDF Programming Guide latest documentation \(espressif.com\)](http://ESP-IDF-Programming-Guide-latest-documentation.espressif.com)
- [SPI Master Driver - ESP32 - — ESP-IDF Programming Guide latest documentation \(espressif.com\)](http://ESP-IDF-Programming-Guide-latest-documentation.espressif.com)
- [Wi-Fi - ESP32 - — ESP-IDF Programming Guide latest documentation \(espressif.com\)](http://ESP-IDF-Programming-Guide-latest-documentation.espressif.com)
- [ESP-NOW - ESP32 - — ESP-IDF Programming Guide latest documentation \(espressif.com\)](http://ESP-IDF-Programming-Guide-latest-documentation.espressif.com)
- [ER-TFTM1.28-1 Datasheet \(buydisplay.com\)](http://buydisplay.com)
- [GC9A01A Datasheet \(buydisplay.com\)](http://buydisplay.com)
- [HTML Tutorial \(w3schools.com\)](http://w3schools.com)
- [CSS Tutorial \(w3schools.com\)](http://w3schools.com)
- [CCNA: Introduction to Networks | Networking Academy \(netacad.com\)](http://netacad.com)
- *Manuale Cremonese di Informatica e Telecomunicazioni (Zanichelli)*

Allegati:

È possibile trovare l'applicazione per Android e il file CAD contenente il supporto per il display LCD come allegato nella mail inviata.



CasaDomotica.apk



Porta-LCD-Centralina
-ESP32.f3d