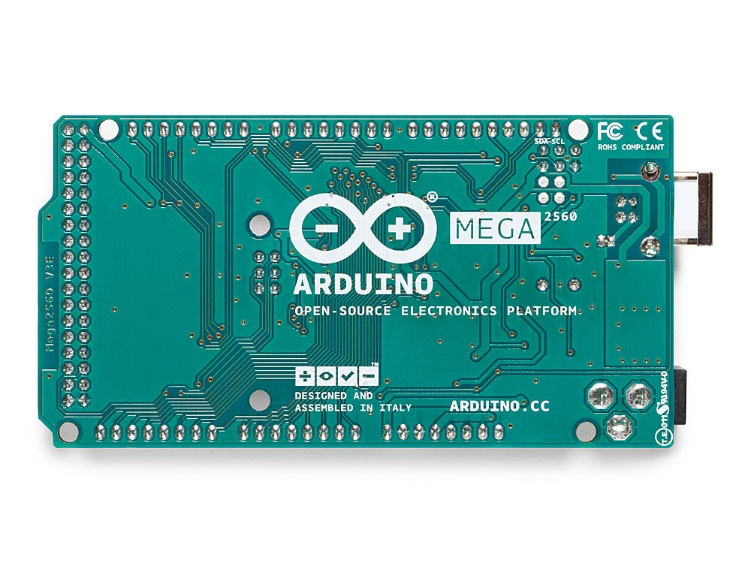
# **COMPONENTTI PROGETTO**

## **Arduino Mega 2560 rev3**

Immagine che contiene elettronica, Componente elettrico, Componente di circuito, Componente di circuito passivo

Descrizione generata automaticamente

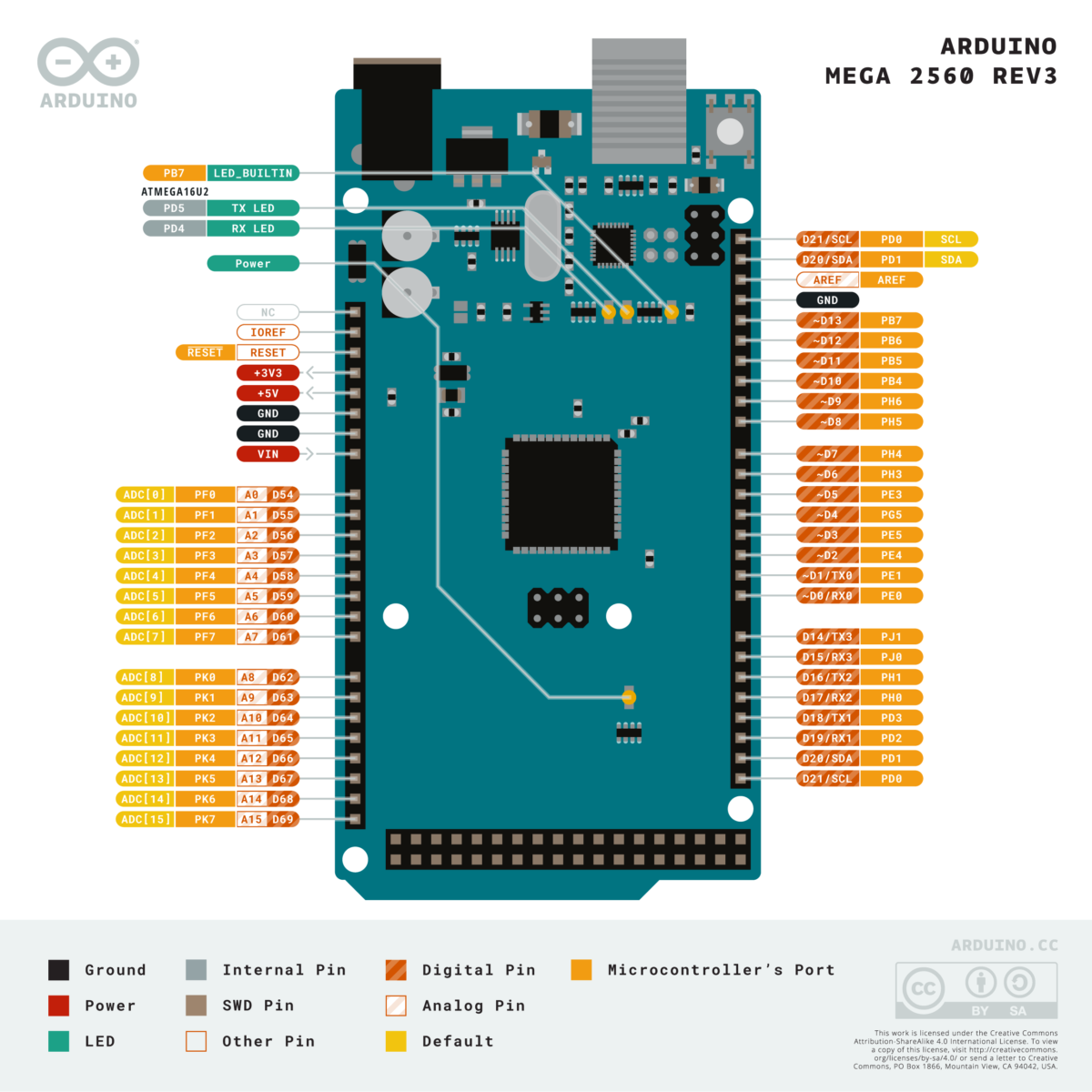
**CARATERISTICHE SCHEDA:**

|  |  |  |
| --- | --- | --- |
|  | **CARATERISTICA** | **VALORE** |
| **1** | **Microcontrollore** | **ATmega 2560** |
| **2** | **Tensione Operativa** | **5V** |
| **3** | **Input Voltage(consigliata)** | **7-12V** |
| **4** | **Input Voltage(limiti)** | **6-20V** |
| **5** | **Pin I/O digitali** | **54 (14 sono di uscita segnali PWM)** |
| **6** | **Pin imput Analogici** | **16** |
| **7** | **DC current per I/O** | **40mA** |
| **8** | **DC current per pin allimentati a 3.3V** | **50mA** |
| **9** | **Flash Memory** | **256KB** |
| **10** | **SRAM** | **8KB** |
| **11** | **EEPROM** | **4KB** |
| **12** | **Frequenza Clock** | **16MHz** |

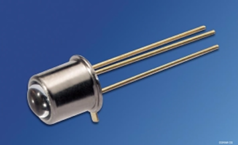
**DESCRIZIONE:**

La Arduino Mega 2560 rev3 è una scheda basata sul microcontrollore ATmega2560, dotata di 54 pin di input/output digitali (14 di questi possono essere usati come segnali PWM), 16 input analogici. Fornisce tutto ciò che è necessario per supportare il funzionamento del microcontrollore, permettendo di controllare dispositivi elettronici, leggere sensori e utomatizzare processi tramite un codice programmato dall'utente.

**INOUT Arduino Mega 2560:**

****

## **Phototransistor**

Il BPX 43 è un fototransistor al silicio NPN ad alta sensibilità progettato per rilevare la luce nell'intervallo spettrale del visibile e del vicino infrarosso, con una risposta spettrale che va da circa 400 nm a 1100 nm. Ciò lo rende ideale per applicazioni quali rilevatori ottici, interruttori fotoelettrici e sistemi di controllo a infrarossi. Il suo contenitore in plastica nera aiuta a filtrare la luce visibile e migliora la risposta agli infrarossi.

Grazie all'elevata corrente di collettore (fino a 50 mA in condizioni di illuminazione ottimali) e alla tensione operativa fino a 70 V, il BPX 43 funziona in modo affidabile in una varietà di ambienti, come telecomando, sensore di prossimità e tachimetro. La sua struttura consente tempi di risposta rapidi, il che è fondamentale per le applicazioni che richiedono un rilevamento della luce rapido e accurato.

|  |
| --- |
| **Caratteristiche** |
|  |
| **Particolarmente adatto per applicazioni da Da 450 nm a 1100 nm** |
| **Elevata linearità** |
| **Confezione metallica ermeticamente sigillata (TO-18) con collegamento alla base adatto fino a 125 °C** |
| **Disponibile in gruppi** |
|  |
|  |
| **Applicazioni** |
|  |
| **Foto-interruttori** |
| **Elettronica industriale** |
| **Per circuiti di controllo e azionamento** |

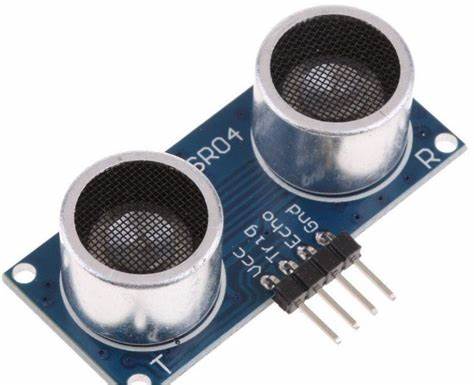
L'angolo di rilevamento del BPX 43 lo rende adatto a specifiche direzioni della luce ed è particolarmente utile nei sistemi di misurazione e controllo, dove combina elevata sensibilità, velocità di risposta e facile integrazione nei circuiti elettronici.

Caratteristiche:

Altri Modelli:

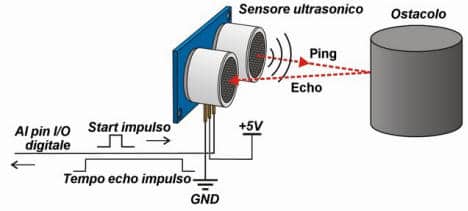
|  |  |
| --- | --- |
| **TIPO** | **CODICE D'ORDINE** |
| **BPX 43** | **Q62702-P16** |
| **BPX 43-2/3** | **Q62702-P3580** |
| **BPX 43-3** | **Q62702-P16-S3** |
| **BPX 43-3/4** | **Q62702-P3581** |
| **BPX 43-4** | **Q62702-P16-S4** |
| **BPX 43-4/5** | **Q62702-P3582** |
| **BPX 43-5** | **Q62702-P16-S5** |

## **Sensore ad Ultra Suoni**

Il sensore ad ultrasuoni (HC-SR04) è un dispositivo che serve a rilevare la presenza di oggetti/ostacoli nelle sue immediate vicinanze senza bisogno di contatto fisico, in particolare usa il principio del sonar.

Il HC-SR04 è dotato di un trasmettitore e di un ricevitore.

Il sensore emette impulsi sonori ultrasonici che, se riflessi da un oggetto/ostacolo, generano un burst ultrasonico (40KHz). In seguito, utilizzando il tempo impiegato dal burst per tornare, il dispositivo può determinare la presenza e la distanza di un oggetto entro il raggio di azione.

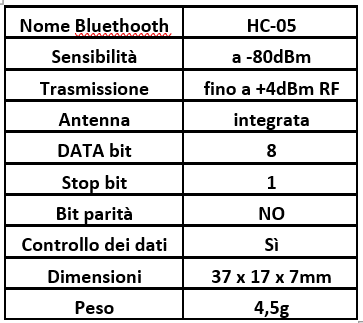
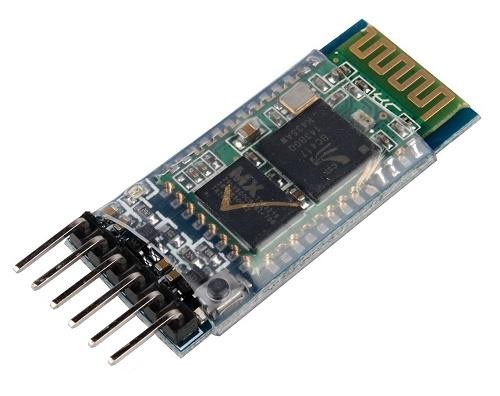
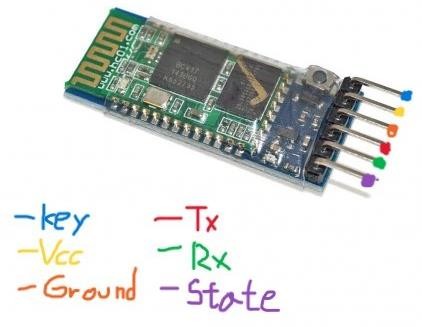


**LIMITI:**

* Distanza dell’oggetto superiore a quella massima ammessa dal sensore
* Angolo di incidenza del tono ultrasonico con l’oggetto troppo piccolo per permettere una sufficiente riflessione
* Oggetto troppo piccolo
* Insufficiente riflessione del suono da parte delle superfici fonoassorbenti e irregolari

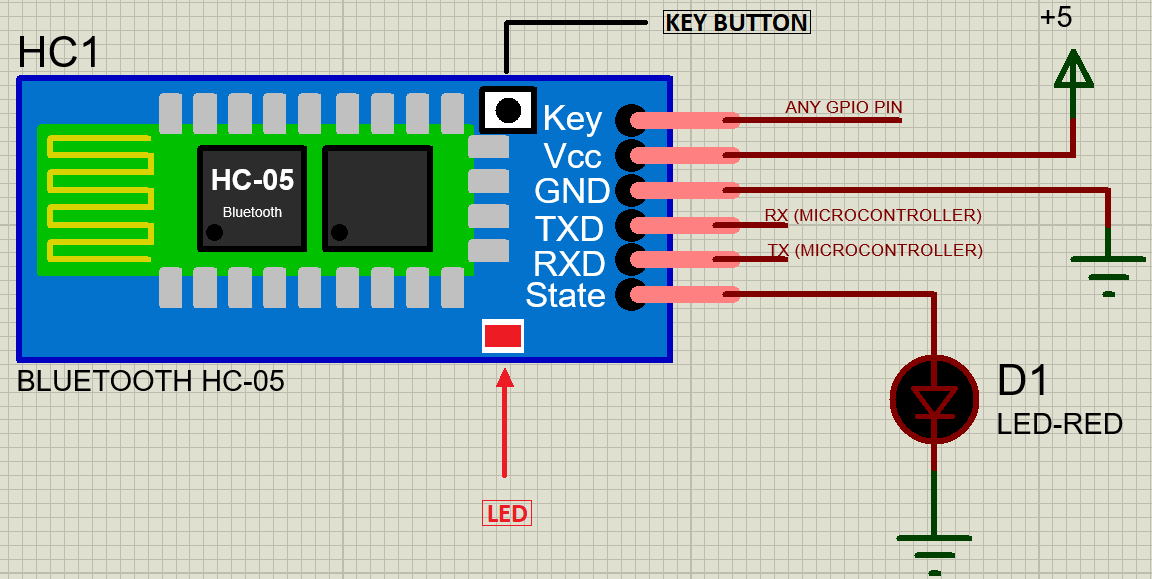
|  |  |
| --- | --- |
| **CARATERISTICHE** | |
| **ALIMENTAZIONE** | **+5 V DC** |
| **ANGOLO DI MISURA** | **< 30°** |
| **DISTANZA DI RILEVAMNETO** | **2 cm A 40 cm** |
| **RISOLUZIONE** | **1 cm** |
| **FREQUENZA** | **40 kHz** |
| **4 PIN** | |
| **VCC** | **+5 V DC** |
| **TRIGGER** | **Genera l'impulso ultrasonico** |
| **ECHO** | **Rileva il segnale ultrasonico di ritorno** |
| **GND** | **0V Ground** |

## **HC-05 (Dispositivo Bluetooth)**

Il modulo HC-05 è un modulo in gradi di convertire una porta seriale UART in una porta Bluetooth; questo dispositivo può essere utilizzato per consentire ad un microprocessore (MCU) installato su una scheda Arduino di comunicare con un dispositivo dotato di Bluetooth (come un PC, smartphone o tablet).

Il modulo si basa sul chip CSR Bluecore 04 External, utilizza la tecnologia CMOS e AFH (adptive frequency hopping function), ed è compatibile con il protocollo Bluetooth V2.0 3Mbps + EDR (Enhanced Data Rate).

La caratteristica principale del modulo HC-05 è che può essere impostato in due modalità tramite comandi AT a seconda delle esigenze dell'utente: come dispositivo master o come dispositivo slave; questo permetterà di utilizzare lo stesso tipo di modulo per creare una rete di dispositivi.

(immagine da: Micricontroller Lab immagine da: Micricontroller Lab)

**Comandi**

|  |  |
| --- | --- |
| **Comandi AT** | **Funzione** |
| **AT** | **Testa la connessione all’interfaccia AT** |
| **AT + reset** | **Resetta il nostro componente** |
| **AT + version** | **Comunica la versione del firmware** |
| **AT + orgl** | **Riporta le impostazioni correnti alle predefinite** |
| **AT + addr** | **Il dispositivo ci comunica il proprio indirizzo** |
| **AT + name** | **Domandiamo o impostiamo il nome del dispositivo** |
| **AT + rname** | **Domandiamo il nome di un dispositivo bluetooth a cui siamo collegati (remoto)** |
| **AT + role** | **Domandiamo o impostiamo il ruolo del dispositivo (1=Master / 0=Slave)** |
| **AT + class** | **Domandiamo o impostiamo la classe del dispositivo (Class of Device CoD)** |
| **AT + iac** | **Domandiamo o impostiamo le richieste del codice di accesso** |
| **AT + inqm** | **Domandiamo o impostiamo le richieste della modalità di accesso** |
| **AT + pswd** | **Domandiamo la password o ne impostiamo l’associazione** |
| **AT + uart** | **Domandiamo i parametri UART o li reimpostiamo** |
| **AT + cmode** | **Domandiamo la modalità di connessione o la impostiamo** |
| **AT + bind** | **Domandiamo o impostiamo l’associazione all’indirizzo del bluetooth** |
| **AT + polar** | **Domandiamo o impostiamo la polarità di output del LED** |
| **AT + pio** | **Impostiamo o resettiamo un pin di Input/Output (I/O) utente** |
| **AT + mpio** | **Impostiamo o resettiamo più pin di I/O utente** |
| **AT + mpio?** | **Domandiamo l’I/O di un pin utente** |
| **AT + ipscan** | **Domandiamo o impostiamo i parametri di scansione** |
| **AT + sniff** | **Domandiamo o impostiamo i parametri di risparmio energetico SNIFF** |
| **AT + senm** | **Domandiamo o impostiamo i modi di sicurezza e crittografia** |
| **AT + rmsad** | **Elimina un dispositivo autenticato dalla lista** |
| **AT + fsad** | **Trova un dispositivo dalla lista dei dispositivi autenticati** |
| **AT + adcn** | **Domandiamo il numero totale dei dispositivi dalla lista dei dispositivi autenticati** |
| **AT + mrad** | **Domandiamo quale è il dispositivo autenticato utilizzato più di recente** |
| **AT + state** | **Domandiamo lo stato corrente del dispositivo** |
| **AT + init** | **Inizializziamo il profilo SPP (ricerca)** |
| **AT + inq** | **Domandiamo quale sia il dispositivo individuabile più vicino** |
| **AT + inqc** | **Cancelliamo la ricerca dei dispositivi individuabili** |
| **AT + pair** | **Associazione al dispositivo** |
| **AT + link** | **Connessione ad un dispositivo remoto** |
| **AT + disc** | **Disconnessione da un dispositivo remoto** |
| **AT + ensniff** | **Entriamo in modalità risparmio energetico** |
| **AT + exsniff** | **Usciamo dalla modalità di risparmio energetico** |

## **Arduino nano every**

**(Usato solo nei test e non nel progettato finale)**

Immagine che contiene elettronica, Componente elettrico, Componente di circuito, Ingegneria elettronica

Descrizione generata automaticamente

La scheda Arduino Nano Every è compatibile con i 5V nella forma più piccola: 45\*18 mm; è una scheda usata per molti progetti che richiedono un microcontrollore piccolo e facile da usare, adatto per invenzioni indossabili, robotica a basso costo, strumenti musicali elettronici e per controllare le parti piccole di progetti grandi.

La scheda presenta 14 pin digitali, di cui 5 supportano il PWM, e 8 ingressi analogici, più un'uscita analogica con DAC a 10 bit. Supporta una tensione di funzionamento a 5V e può essere alimentato tramite USB o attraverso il pin VIN, accettando tensioni da 5V fino a 21V, rendendola molto flessibile per progetti alimentati esternamente. Include inoltre interfacce di comunicazione come UART, SPI e I²C, ampliando la gamma di dispositivi con cui può interagire.

Il microcontrollore ATmega4809 introduce anche supporto per Core Independent Peripherals (CIPs), che consentono alle periferiche interne di operare in modo autonomo, alleggerendo il carico di lavoro della CPU. Inoltre, la scheda supporta modalità di risparmio energetico avanzate, risultando particolarmente utile per progetti che devono essere alimentati a batteria o che richiedono bassi consumi energetici.

**Specifiche Tecniche**

|  |  |
| --- | --- |
| **Microcontrollore** | **ATMega4809** |
| **Tensione di esercizio** | **5V** |
| **Vin min-Max** | **7-21V** |
| **Corrente CC per pin I/O** | **20mA** |
| **Corrente CC per pin da 3,3 V** | **50mA** |
| **Velocità di clock** | **20MHz** |
| **Memoria Flash CPU** | **48KB** |
| **SRAM** | **6KB** |
| **Memoria EEPROM** | **256byte** |
| **Perni PWM** | **5 (D3, D5, D6, D9, D10)** |
| **UART** | **1** |
| **SPI** | **1** |
| **I2C** | **1** |
| **Pin di ingresso analogici** | **8 (ADC 10 bit)** |
| **Pin di uscita analogici** | **Solo tramite PWM** |
| **Interruzioni esterne** | **tutti i pin digitali** |
| **LED\_INCORPORATO** | **13** |
| **USB** | **Utilizza ATSAMD11D14A** |
| **Lunghezza** | **45 millimetri** |
| **Larghezza** | **18 millimetri** |
| **Peso** | **5 gr (con intestazioni)** |

**Porte PIN**

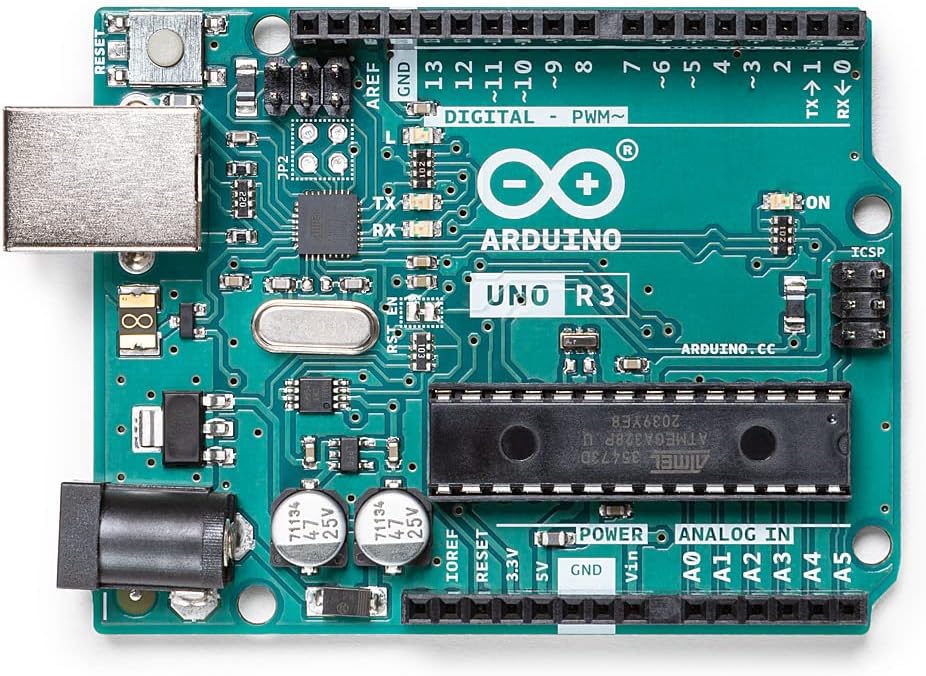
Immagine che contiene testo, schermata, diagramma, design

Descrizione generata automaticamente

* **Ground**
* **Power**
* **LED**
* **Internal Pin**
* **SWD Pin**
* **Other Pin (Reset)**
* **Digital Pin (D2, D3...)**
* **Analog Pin (A0, A1…)**
* **Default**
* **Porta microcontrollore**

## **Arduino Uno**

**(Usato solo nei test del progetto e non nel progettato finale)**

L'**Arduino** **Uno** è una scheda microcontrollore che utilizza il chip **ATmega328P**. Dispone di 14 pin digitali per input/output, 6 dei quali supportano l'output PWM, oltre a 6 ingressi analogici. La scheda include un risonatore ceramico da 16 MHz (modello CSTCE16M0V53-R0), una porta USB per la connessione al computer, un jack per l'alimentazione esterna, un header ICSP e un pulsante di reset. Fornisce tutto il necessario per gestire il microcontrollore: è sufficiente collegarla al PC tramite USB o alimentarla con una batteria o un adattatore AC-DC per iniziare ad utilizzarla.

**Specifiche Tecniche**

|  |  |
| --- | --- |
| **Microcontrollore** | **ATmega328p** |
| **Voltaggio Operativo** | **5V** |
| **Voltaggio Imput** | **7-12V** |
| **Voltaggio Imput (limite)** | **6-20V** |
| **Pin Digitale I/O** | **14** |
| **Pin PWM Digitale I/O** | **6** |
| **Pin Input Analogici** | **6** |
| **Corrente DC per I/O** | **20 mA** |
| **Corrente DC per 3.3V** | **50 Ma** |
| **Memoria Flash** | **32 KB** |
| **SRAM** | **2 KB** |
| **EEPROM** | **1 KB** |
| **Velocita Clock** | **16 MHz** |
| **LED\_BUILTIN** | **13** |
| **Lunghezza** | **68.6 mm** |
| **Larghezza** | **53.4 mm** |
| **Peso** | **25 g** |

## **Modulo NFC V3**

Immagine che contiene testo, rosso, interno, muro

Descrizione generata automaticamenteImmagine che contiene testo, rosso, muro, interno

Descrizione generata automaticamente

Il Modulo NFC v3 è un dispositivo hardware che consente la comunicazione tramite tecnologia NFC (Near Field Communication), un tipo di connessione wireless che funziona su brevi distanze, solitamente entro 4 cm. È ampiamente utilizzato per progetti elettronici che coinvolgono microcontrollori come Arduino, Raspberry Pi e altre piattaforme simili.

**CARATERISTICHE**

* **Chip:** L’NFC V3 è spesso basato su chip NXP PN532, uno dei più diffusi
* **Interfaccia di Comunicazione:**

1. **I2C** (modalità di default)
2. **SPI**
3. **UART**

* **Distanza Lettura:** Solitamente intorno a 2-4 cm, ma può variare in base al tag utilizzato
* **Velocità di comunicazione:**

1. **I2C:** Velocita standard (100 kHz) e veloce (400kHz)
2. **SPI:** Supporto fino a 5 Mbps
3. **UART:** Baud rate configurabile

* **Tensione Operativa:** 3,3 V e 5 V, adatto sia per progetti Arduino e Raspberry
* **Protocolli NFC supportati:**

1. **ISO/IEC 14443-A/B**
2. **ISO/IEC 18092**

* **Dimensioni:** Solitamente attorno ai 40 mm x 40mm

**Modalità di Funzionamento**

* **Reader/writer Mode:** In questa modalità, il modulo può leggere e scrivere su tag NFC.
* **Peer-to-Peer Mode:** Consente la comunicazione diretta tra due dispositivi NFC, ad esempio tra un telefono NFC e il modulo stesso.
* **Card Emulation Mode:** Il modulo può simulare una smart card NFC, permettendo ad altri lettori di leggere i dati del modulo come se fosse un tag NFC.

**Esempio Codice**

#include <Wire.h> // Libreria per la comunicazione I2C

#include <Adafruit\_PN532.h> // Libreria Adafruit per PN532

// Definisci i pin I2C

#define SDA\_PIN A4

#define SCL\_PIN A5

// Crea un'istanza dell'oggetto PN532 usando l'I2C

Adafruit\_PN532 nfc(SDA\_PIN, SCL\_PIN);

void setup(void) {

**Serial**.begin(115200); // Inizializza la comunicazione seriale

**Serial**.println("Inizializzazione del modulo NFC...");

nfc.begin(); // Avvia la comunicazione con il modulo NFC

uint32\_t versiondata = nfc.getFirmwareVersion();

if (!versiondata) {

**Serial**.println("Modulo NFC non trovato!");

while (1); // Se non trova il modulo NFC, blocca l'esecuzione

}

// Mostra la versione del firmware del modulo NFC

**Serial**.print("Firmware trovato: 0x");

**Serial**.println(versiondata, HEX);

// Configura il modulo NFC per la lettura di tag

nfc.SAMConfig();

**Serial**.println("Modulo NFC pronto per leggere tag.");

}

void loop(void) {

**Serial**.println("Posiziona un tag NFC vicino al lettore...");

uint8\_t success;

uint8\_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Array per memorizzare l'UID

uint8\_t uidLength; // Variabile per la lunghezza dell'UID

// Prova a leggere un tag NFC (protocollo ISO14443A come MIFARE)

success = nfc.readPassiveTargetID(PN532\_MIFARE\_ISO14443A, uid, &uidLength);

if (success) {

// Se viene rilevato un tag, stampa l'UID del tag

**Serial**.println("Tag NFC trovato!");

**Serial**.print("UID del tag: ");

for (uint8\_t i = 0; i < uidLength; i++) {

**Serial**.print(" 0x");

**Serial**.print(uid[i], HEX); // Stampa l'UID in formato esadecimale

}

**Serial**.println();

delay(1000); // Aspetta 1 secondo prima di cercare di nuovo

}

else {

**Serial**.println("Nessun tag trovato.");

}

delay(1000); // Aspetta un po' prima di riprovare

}

**Vantaggi e Limitazioni**

* **Vantaggi:**

1. Facile da usare con piattaforme come Arduino e Raspberry Pi.
2. Compatibile con una vasta gamma di protocolli NFC.
3. Supporto per lettura, scrittura ed emulazione di tag NFC.

* **Limitazioni:**

1. Portata limitata a pochi centimetri.
2. Richiede la libreria e una configurazione corretta per l'interfaccia desiderata.
3. Prestazioni variabili a seconda del tipo di tag utilizzato.

## **RFId**

RFId (Radio Frequency Identification) viene utilizzata per l'identificazione automatica di oggetti animali e persone;

basato sulla lettura di informazioni nel Tag.

**Componenti:**

* Tag o transponder
* Reader o tranceiver
* Antenna
* Sistema elaborazione

**TAG** (dipende dal tipo di alimentazione):

* **trasponder passivo:** utilizzano l'onda a radio frequenza generato dal lettore sai come fonte di energia per alimentazione che per ricevere dati, Basso costo tempi di vita più lunghi, distanza limitati;
* **trasponder semi passivo:** batteria incorporata, grande distanza di comunicazione, costosi, impossibile stabilire stato batteria;
* **trasponder attivo:** completamente autonomi, autoalimentati da batteria e possiedono un trasmettitore attivo bordo, grande distanza di comunicazione, costosi, impossibile stabilire stato batteria;

**Distanza(tag):** da pochi centimetri a qualche metro;

**Contenuto(tag):** antenna, chip che svolge semplici elaborazioni, piccola memoria;

**Assemblaggio(tag):** substrato, antenna, chip, overlay;

Campi che utilizzano RFId:

* Industriale
* medico
* automotivo

PN532 --> basato su corrente

Distanza: 5-6 cm

reader che writer

alimentazione: 3,3V a 5V

SDA --> serial data

SCL --> serial connector

## **Sensore IR Sharp**

Il sensore IR sharp è un tipo di sensore a infrarossi usato per il rilevamento di distanze di oggetti senza andare in contatto.

Il sensore emette un fascio di luce infrarossi attraverso un LED IR. La luce viene riflessa dall’oggetto e rilevata da un fotodiodo all’interno del sensore e basandosi sull’angolo di riflessione e l’intensità della luca riflessa, il sensore calcola la distanza dall’oggetto.

**Specifiche Tecniche**

|  |  |
| --- | --- |
| **Nome** | **Sensore IR Sharp** |
| **Portata di misura** | **10 cm - 80 cm** |
| **Tensione di Alimentazione** | **4.5 V - 5.5 V** |
| **Corrente Operativa** | **Circa 30 mA** |
| **Tempo di risposta** | **Circa 38 ms** |
| **Segnale di uscita** | **0.4 V a 3.1 V** |
| **Uscita** | **Analogica** |
| **Angolo di rilevamento** | **10 °** |
| **Dimensioni** | **29.5mm x 13mm x 21.6mm** |
| **Peso** | **Circa 3.5 g** |