

ALPHA

Acquisizione Locale di Parametri con Hardware Avanzato

Alessio Tommasi

17 febbraio 2025

Indice

1	Introduzione	2
1.1	Dipendenze	2
1.2	Configurazione dell'Arduino IDE	3
2	Hardware	4
2.1	Funzionamento	4
2.1.1	Alimentazione	4
2.1.2	Ingressi Digitali	6
2.1.3	Multiplex	8
2.2	Hardware esterno	12
2.2.1	Winzet W5500 Ethernet module	12
2.2.2	MAX485 Modbus RTU	14
2.2.3	ADS1115 ADC	16
2.2.4	ESP32 38 Pin	17
3	Software	18
3.1	Diagramma UML	18
3.1.1	Pattern	18
3.2	performace	19
3.3	WebServer	19
3.4	Modbus	19
4	Attività	21
5	Conclusioni	22
5.1	Sviluppi futuri	22
5.2	Ringraziamenti	22

Capitolo 1

Introduzione

Il progetto *ALPHA* è stato sviluppato nel corso di IoT del Master in Informatica presso SUPSI. Il focus principale è sull'ESP32 e il protocollo Modbus.

La documentazione ufficiale del progetto disponibile al seguente link:
[ALPHA](#).

1.1 Dipendenze

Driver Per gli utenti Windows, è necessario installare **CP210xDriver**

Compiler Per compilare tale progetto e' stato utilizzato Arduino IDE 2.3.3. disponibile al seguente link: [Arduino IDE](#).

Public library

AsyncTCP ulteriori informazioni sono disponibili a questo link

ESPAsyncTCP ulteriori informazioni sono disponibili a questo link

ESPAsyncWebServer ulteriori informazioni sono disponibili a questo link

UIPEthernet ulteriori informazioni sono disponibili a questo link

Modbus Sono state testate diverse librerie, ma nessuna di esse ha funzionato in modo ottimale.

1.2 Configurazione dell'Arduino IDE

Link repo ufficiale: [iotProject](#).

Per compilare i file nelle sottocartelle, è necessario aggiungerli come librerie (.zip) all'Arduino IDE. Ho creato una cartella specifica per le librerie dove posizionare o sostituire i file zip. Per una corretta compilazione, importa tutte le cartelle zip presenti in /Library.

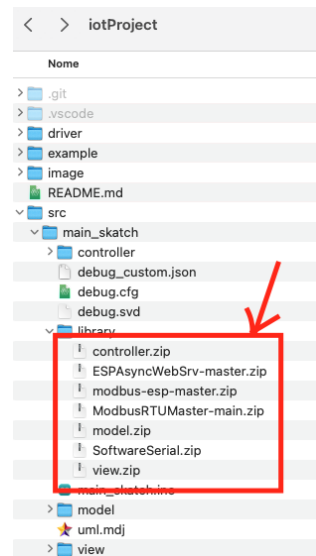


Figura 1.1: Importazione delle librerie nell'Arduino IDE

Altrimenti clonare la versione Portable del progetto disponibile al seguente link: [iotProject-portable](#).

Capitolo 2

Hardware

2.1 Funzionamento

2.1.1 Alimentazione

E' possibile alimentare la board tramite un alimentatore esterno da 24V DC che fornisca almeno 0.5A.

In particolare e' stato utilizzato SIMENS SITOP PSU100C 24V 0.5A.

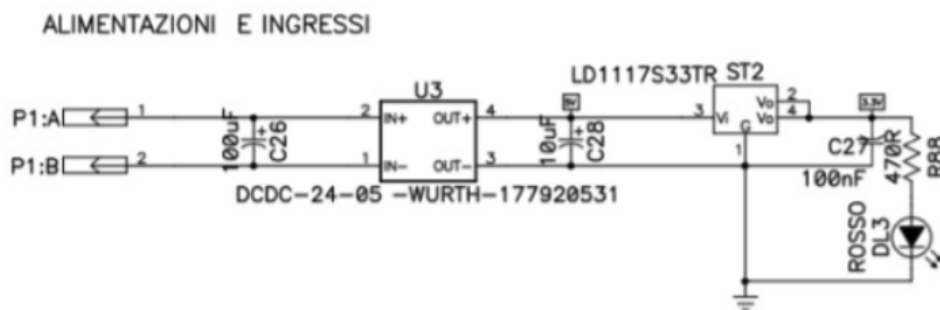


Figura 2.1:

Descrizione Lo schema in Figura 2.1 rappresenta il sistema di alimentazione con regolatori di tensione e relativi condensatori di filtraggio. I connettori P1:A e P1:B forniscono l'ingresso principale della tensione al circuito.

Conversione DC-DC Il primo stadio dell'alimentazione è rappresentato dal modulo DC-DC Wurth 177920531, (Data Sheet Here) che si occupa

della conversione della tensione di ingresso a un livello adeguato per il regolatore successivo.

La stabilizzazione della tensione è garantita dai condensatori C26 e C28 entrambi da 100 μ F, i quali riducono eventuali ripple presenti nella tensione di alimentazione.

Regolazione di tensione Dopo la conversione DC-DC, il regolatore lineare LD1117S33TR (U3 sulla board) fornisce una tensione stabile di 3.3V, necessaria per l'alimentazione dei componenti successivi. Questo regolatore include un condensatore di ingresso C28 di 10 μ F e un condensatore di uscita C27 di 100nF per migliorare la stabilità del segnale. Un LED rosso (D3) con la resistenza R88 fornisce un'indicazione visiva della corretta alimentazione del sistema.

Regolatore Switching LM7660 Un ulteriore regolatore di tensione, l'LM7660 (U10), viene utilizzato per generare una tensione negativa o fornire una conversione di tensione specifica.

Questo componente opera con i condensatori C29 e C30 (entrambi da 10 μ F), i quali servono per stabilizzare la tensione e ridurre le oscillazioni indesiderate.

Uscite Le alimentazioni ottenute distribuiti tramite i connettori P20:1, P20:2, P21:1, P21:2, P22:1 e P22:2, che permettono l'integrazione del sistema con altri moduli elettronici.

P20:1 e P20:2 forniscono la tensione di alimentazione principale a 5V, mentre P21:1 e P21:2 forniscono la tensione negativa a -5V.

P22:1 e P22:2 forniscono la tensione di alimentazione a 3.3V.

2.1.2 Ingressi Digitali

Questa sezione descrive l'implementazione degli ingressi digitali nel circuito, illustrando il principio di funzionamento, la protezione e il filtraggio dei segnali.

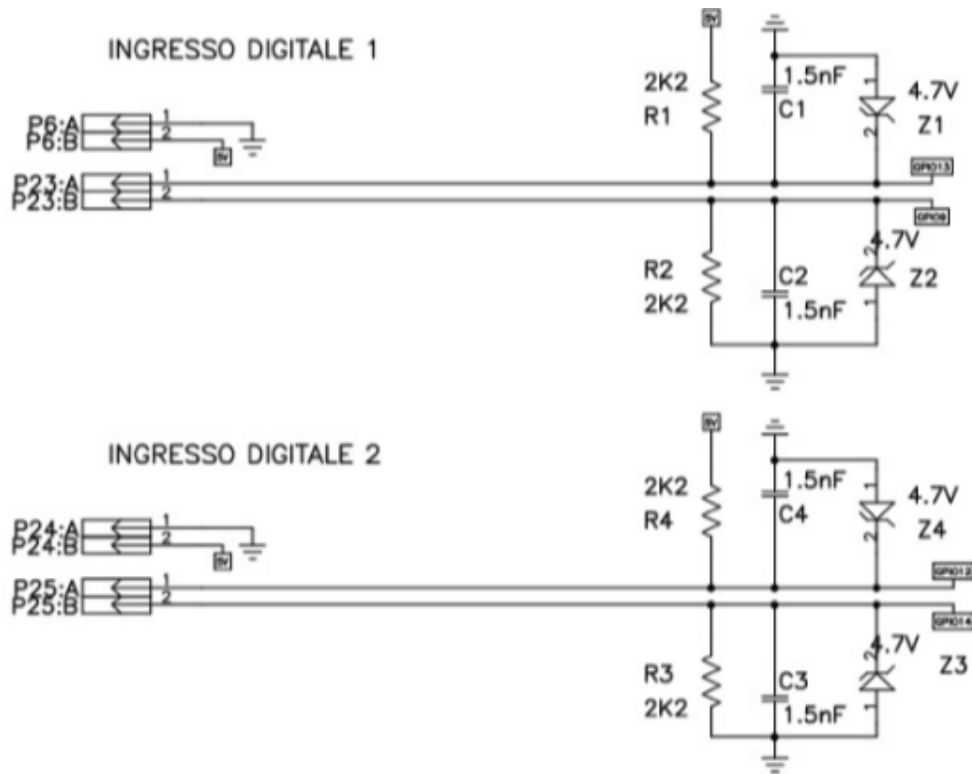


Figura 2.2:

Descrizione Il circuito degli ingressi digitali è progettato per accettare segnali in ingresso e condizionarli adeguatamente prima di inviarli alla logica di elaborazione.

Ogni ingresso è dotato di:

- **Resistenze** da 2.2 k Ω rispettivamente (R1, R2) per ingresso digitale 1 e (R3, R4) per ingresso digitale 2: il loro scopo è limitare la corrente di ingresso e formare un partitore resistivo.
- **Condensatori** da 1.5 nF (C1, C2, C3, C4): implementati per ridurre il rumore ad alta frequenza e migliorare l'integrità del segnale.
- **Diodi Zener** da 4.7V (Z1, Z2, Z3, Z4): impiegati per proteggere l'ingresso da sovratensioni accidentali che potrebbero danneggiare i componenti a valle.

Conclusioni Grazie all'uso combinato di resistenze, condensatori e diodi di protezione, il circuito è in grado di accettare segnali digitali provenienti da diversi dispositivi, mantenendo una buona immunità al rumore. Inoltre, la configurazione utilizzata permette di garantire livelli logici stabili e ben definiti.

Dunque il sistema di ingressi digitali descritto rappresenta una soluzione efficace per la gestione di segnali binari, garantendo protezione, stabilità e robustezza. Questa configurazione assicura un'interfaccia affidabile tra il mondo fisico e il sistema di elaborazione.

2.1.3 Multiplex

Il dispositivo di multiplexing risulta essere il **CD405xB**, sviluppato da Texas Instruments, link alla documentazione ufficiale: [CD405xB](#).

Siccome successivamente servira diporto di seguito la tabella di verita di tale dispositivo. Figura 5.1

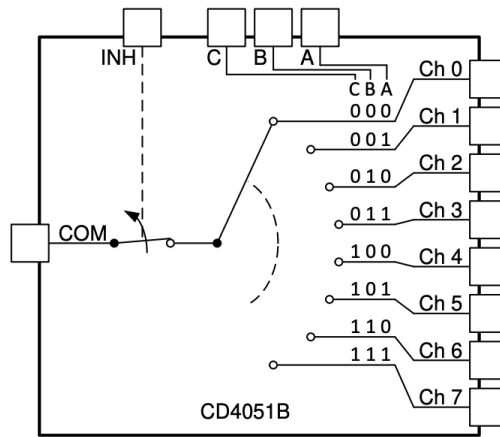


Figura 2.3: Tabella di verita del CD405xB

Collegamenti Multiplexer

I pin di ingresso A, B, C del multiplexer **CD405xB** sono collegati rispettivamente ai pin GPIO 12, 13, 14 dell'ESP32. La selezione dei canali del multiplexer avviene impostando i pin A, B, C come segue:

Canale	Pin A	Pin B	Pin C
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1

Tabella 2.1: Configurazione dei pin per la selezione dei canali del multiplexer

Canali Multiplexer

Nella figura nella pagina seguente seguito vennogno riportati i collegamenti dei canali del multiplexer con i collegamenti esterni.

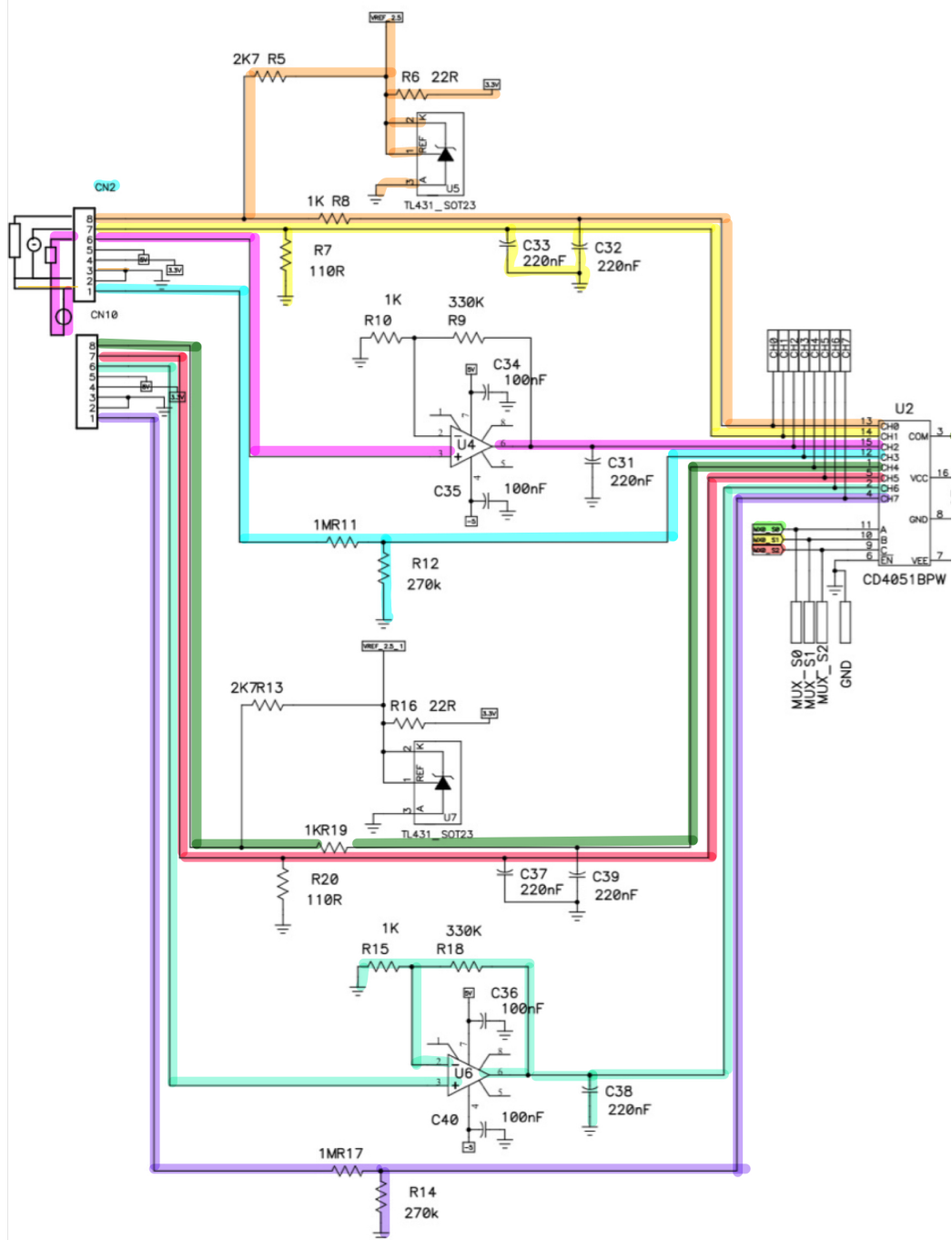


Figura 2.4: Collegamenti dei canali del multiplexer

Ch0 Selezionando Ch0 dall'apposita pagina dedicata segue il percorso evidenziato in arancione nella Figura 5.2. E' collegato a pin 8 della morsettiera CN2. Su tale morsetto ci aspettiamo in ingresso un segnale di corrente che il componente TL431 si occuperà di regolare e stabilizzare la tensione in ingresso entro dei range indicati. a questo link potete trovare il datasheet di tale componente per maggiori info TL431.

Ch1 percorso evidenziato in giallo nella Figura 5.2.
Possibili funzioni del segnale sul segnale in ingresso al pin 7 della morsettiera CN2.

- **Filtro Passa Basso** Se il segnale applicato su R7 è una tensione alternata (AC), il circuito potrebbe attenuare le alte frequenze, lasciando passare solo le frequenze più basse
- **Stabilizzazione del segnale**

Ch2 percorso evidenziato in viola nella Figura 5.2.
Leggeremo il segnale in ingresso al pin 6 della morsettiera CN2.
Il segnale verrà amplificato dall'operazionale U4 che è collegato come amplificatore non invertente (maggiori info al link LM358).
Il guadagno di U4 è dato dalla formula $G = 1 + \frac{R9}{R10} = 1 + 330 = 331$.
Inoltre tale alimentatore è alimentato tra 0 e +5v quindi clamera il segnale in ingresso tra 0 e +5v. Il condensatore C31 si occupa di stabilizzare il segnale in uscita.

Ch3 percorso evidenziato in azzurro nella Figura 5.2.
Leggeremo il segnale in ingresso al pin 1 della morsettiera CN2.
tra il segnale sul pin 1 ed ESP è presente un partitore di tensione composto da $R11 = 1M$ e $R12 = 270K$ dunque
$$V_{esp} = V_{pin1} * \frac{R12}{R11+R12} = V_{pin1} * \frac{270}{1270} = V_{pin1} * 0.2126.$$

Ch4 Connessione equivalente a CH0

Ch5 Connessione equivalente a CH1

Ch6 Connessione equivalente a CH3
Tutte le connessioni equivalenti hanno un circuito hw separato per garantire la massima indipendenza tra i canali.

Per una documentazione dettagliata sulla ricerca di tale modulo e sul codice e librerie utilizzate da esso consultare il seguente link: W5500.

Si è optato per l'utilizzo di un modulo Ethernet W5500 per la comunicazione tramite Modbus TCP/IP.
- documentazione dettagliata modbus tcp/ip Modbus TCP/IP

funzionamento modbus TCP/IP

Modbus TCP è stato sviluppato per sfruttare le infrastrutture di rete LAN esistenti, permettendo la comunicazione attraverso reti Ethernet.

Attualmente viene utilizzato per qualsiasi connessione tra dispositivi connessi a internet.

Questo protocollo incapsula i messaggi Modbus RTU in pacchetti TCP, il che facilita la loro trasmissione su reti Ethernet standard. Uno dei principali vantaggi del Modbus TCP è la sua capacità di connettere un numero illimitato di dispositivi, grazie all'uso di indirizzi IP invece delle limitazioni di indirizzamento dei protocolli seriali.

In questo contesto, Modbus TCP ridefinisce la relazione master-slave in termini di client-server, permettendo una comunicazione più flessibile e scalabile. I dispositivi possono agire come client o server, facilitando l'integrazione di più sistemi e migliorando l'efficienza delle comunicazioni nelle reti industriali.

link bibliografia Modbus TCP/IP

Vantaggi W5500

- **Alta velocità e prestazioni:** Supporta fino a 80 Mbps grazie al buffer hardware dedicato.
- **Stack TCP/IP hardware integrato:** Libera risorse sul microcontrollore ESP32.
- **Consumo energetico ridotto:** Più efficiente rispetto all'ENC28J60, ideale per applicazioni a basso consumo.
- **Compatibilità ampia:** Supportato dalla libreria Ethernet ufficiale di Arduino.

Per visualizzare una ricerca più dettagliata sul confronto di tale modulo con altri moduli presenti sul mercato consultare il seguente link:

ResearchEthModule.

2.2.2 MAX485 Modbus RTU

Tale moduli sono utilizzati per la comunicazione seriale tra ESP32 e altri dispositivi che supportano il protocollo Modbus RTU.

il dispositivo deve essere alimentato con 5V e collegato ai pin RX e TX dell'ESP32. per far si che la comunicazione funzioni correttamente collegare cavo trasporto dati dell'usb e collegare al dispositivo Master o slave.

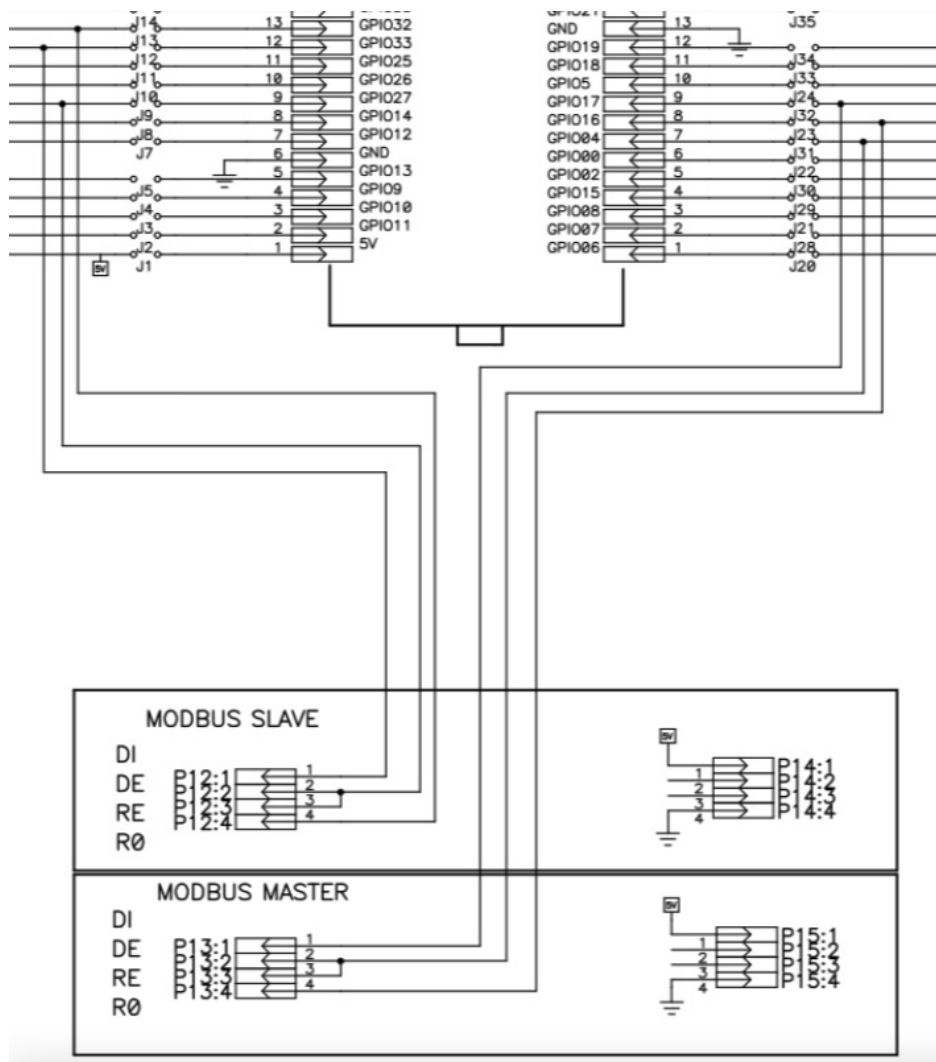


Figura 2.6: MAX485 Modbus RTU

Descrizione Protocollo Modbus RTU

Modbus RTU è basato su una comunicazione seriale asincrona, che consente la trasmissione di dati su cavi.

Per una corretta comunicazione il dispositivo MAX485 SLAVE deve essere collegato su P12 mentre il dispositivo MASTER su P13.
i collegamenti ai pin dell'esp sono rappresentati in figura 2.6.

- **Caratteristiche**

- Formato di messaggio compatto: I messaggi sono compatti, il che consente una trasmissione più rapida ed efficiente.
- Modalità binaria: I dati vengono trasmessi in un formato binario, il che significa che si usano bit (0 e 1).

- **Vantaggi del Modbus RTU**

- Ideale per distanze brevi e medie.
- Semplice ed economico per installazioni piccole o medie.

- **Limitazioni del Modbus RTU**

- Limitazioni di distanza e velocità.
- Numero limitato di dispositivi che possono essere collegati sulla stessa linea di comunicazione, ovvero 32 dispositivi.

2.2.3 ADS1115 ADC

Il modulo ADS1115 è un convertitore analogico-digitale (ADC) a 16 bit con quattro canali di ingresso.

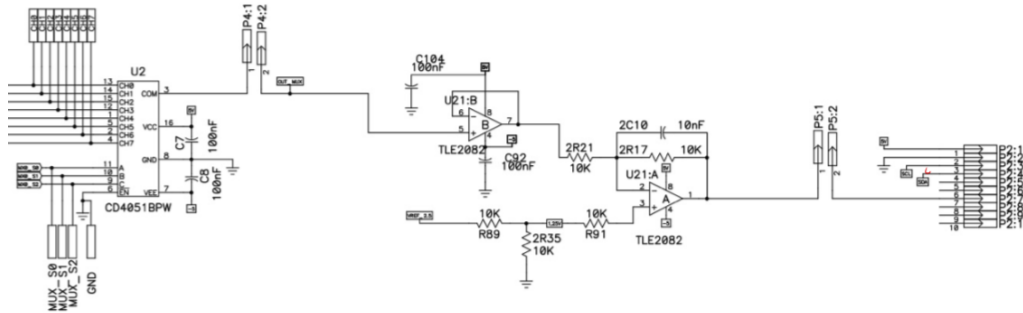


Figura 2.7: ADS1115 ADC

Il dispositivo utilizza un multiplexer descritto nella sezione 2.1.3 per selezionare uno dei canali di ingresso, amplifica il segnale tramite il PGA e poi lo converte in un valore digitale.

Il risultato della conversione viene inviato al microcontrollore tramite l'interfaccia I2C.

il datasheet dettagliato di tale dispositivo e' disponibile al seguente link:
[ADS1115](#).

2.2.4 ESP32 38 Pin

Descrizione

L'ESP32 è un potente microcontrollore prodotto da Espressif Systems che integra funzionalità Wi-Fi e Bluetooth, rendendolo ideale per applicazioni

IoT (Internet of Things), automazione domestica, sistemi indossabili, monitoraggio remoto e molti altri progetti. La versione a 38 pin offre una vasta gamma di GPIO (General Purpose Input/Output) utilizzabili per diversi scopi.

Caratteristiche Principali Dual-core Tensilica LX6: Processore dual-core con velocità di clock fino a 240 MHz.

Wi-Fi e Bluetooth: Supporto per Wi-Fi 802.11b/g/n e Bluetooth v4.2 BR/EDR e BLE (Bluetooth Low Energy).

Interfacce Multiple: Include SPI, I2C, I2S, UART, ADC, DAC, PWM, e touch sensor.

Basso Consumo Energetico: Modalità di basso consumo per applicazioni a batteria.

Memoria Integrata: SRAM da 520 KB e supporto per memoria esterna.

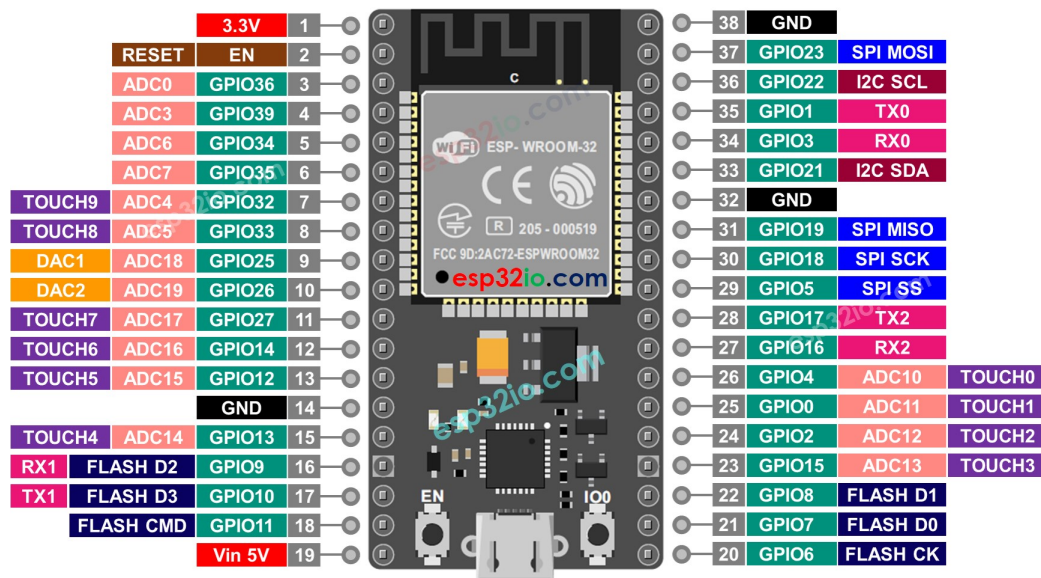


Figura 2.8: Pinout dell'ESP32-DOIT-DEV-KIT v1

Capitolo 3

Software

3.1 Diagramma UML

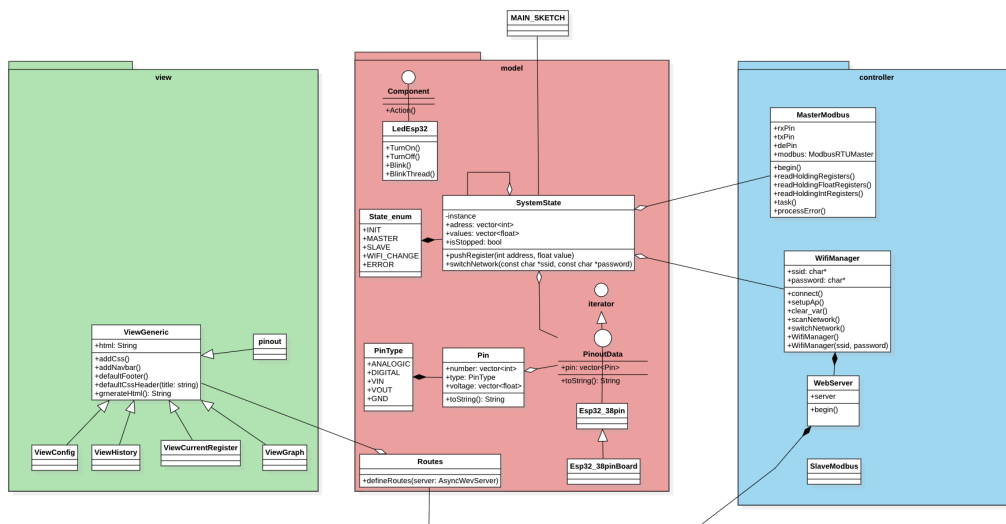


Figura 3.1: Diagramma UML del sistema

3.1.1 Pattern

Model-View-Controller (MVC)

Model Contiene la struttura dei dati e le funzioni per accedere e modificarli. Le classi coinvolte sono **SystemState**, **Pin**, **PinoutData**, **Routes**.

View Si occupa di visualizzare i dati e di interagire con l'utente. principalmente sono classi che si occupano della generazione dei componenti delle pagine web visualizzate dall utente.

Controller Gestisce le richieste dell'utente e aggiorna il modello di conseguenza.

Singleton

utilizzato per garantire unicità e atomicità dei dati **SystemState** e' la classe che implementa tale pattern. Sarà particolarmente utile in futuro per l'implementazione di un sistema di data logging e persistenza dei dati su scheda SD.

3.2 performace

3.3 WebServer

3.4 Modbus

Il file utilizzato per testare lo slave è disponibile qui: `modbusSlave2.ino`.

TX (GPIO32)	--> DI
RX (GPIO33)	--> RO
GPIO27	--> DE, RE
3.3V	--> VCC
GND	--> GND

Figura 3.2: Pinout proposto per il dispositivo slave Modbus

Capitolo 4

Attività

Attività	Descrizione
Configurazione sensori di temperatura	Configurare e integrare sensori di temperatura PT100 , PT1000 e termocoppie utilizzando moduli come MAX31865 e MAX31855 .
Lettura segnali analogici	Implementare la lettura di segnali analogici tramite gli ingressi ADC dell'ESP32 e eventuali moduli esterni.
Gestione uscite digitali e analogiche	Sviluppare la gestione delle uscite digitali e analogiche tramite l'ESP32.
Comunicazione RS485 (Modbus RTU)	Integrare la comunicazione RS485 utilizzando il protocollo Modbus RTU per interfacciarsi con altri dispositivi.
Server Web (Ethernet TCP/IP)	Sviluppare un server Web basato su Ethernet TCP/IP per il monitoraggio e controllo remoto dei dati acquisiti.
Datalogging	Implementare un sistema di datalogging per salvare e storicizzare i dati raccolti dai sensori.
Test e validazione	Testare e validare il sistema attraverso simulazioni e test su hardware reale.

Capitolo 5

Conclusioni

5.1 Sviluppi futuri

persistenza e datalogging su scheda SD

backup dati su cloud implementazione backend etc

5.2 Ringraziamenti