

iotProject

Alessio Tommasi

6 febbraio 2025

Indice

1	Introduzione	2
2	Dipendenze	3
3	Configurazione dell'Arduino IDE	4
4	Diagramma UML	6
5	Hardware	7
5.1	Board Esam	7
5.1.1	Funzionamento	7
5.1.2	hardware esterno posizionabile sulla board	8
6	Software	10
6.1	WebServer	10
6.2	Modbus	10
7	Attività	11

Capitolo 1

Introduzione

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

Capitolo 2

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](#).

Capitolo 3

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.

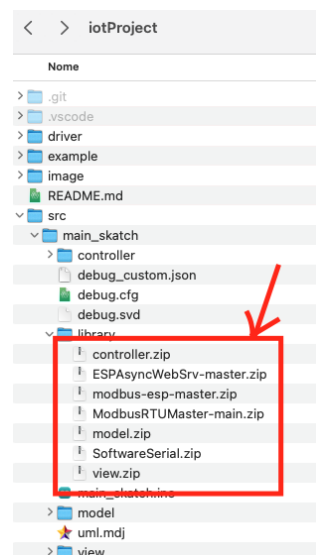


Figura 3.1: Importazione delle librerie nell'Arduino IDE

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

Capitolo 4

Diagramma UML

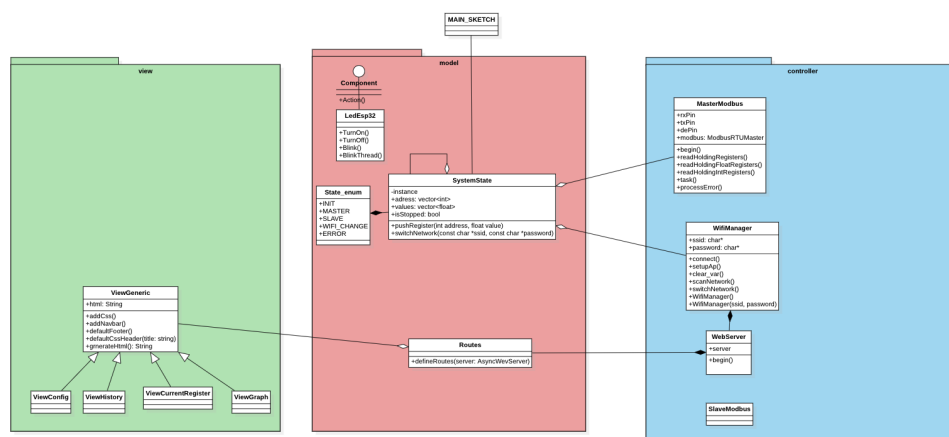


Figura 4.1: Diagramma UML del sistema

Capitolo 5

Hardware

5.1 Board Esam

5.1.1 Funzionamento

Multiplex

Il dispositivo di multiplexing risulta essere il **CD4051B**,
sviluppato da Texas Instruments, link alla documentazione ufficiale:
[CD4051B](#).

Siccome successivamente servirà di seguito la tabella di verità di
tale dispositivo. Figura 5.1

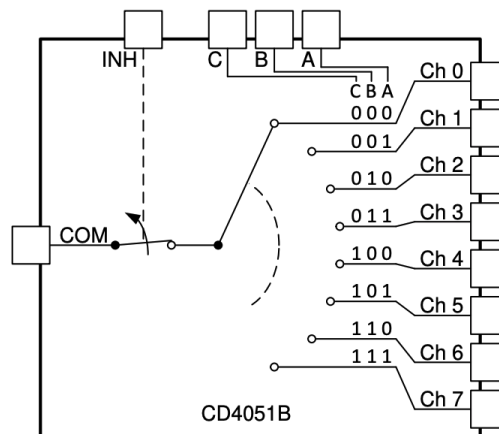


Figura 5.1: Tabella di verità del CD4051B

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 5.1: Configurazione dei pin per la selezione dei canali del multiplexer

5.1.2 hardware esterno posizionabile sulla board

MAX31865

asd

ESP32 38 Pin

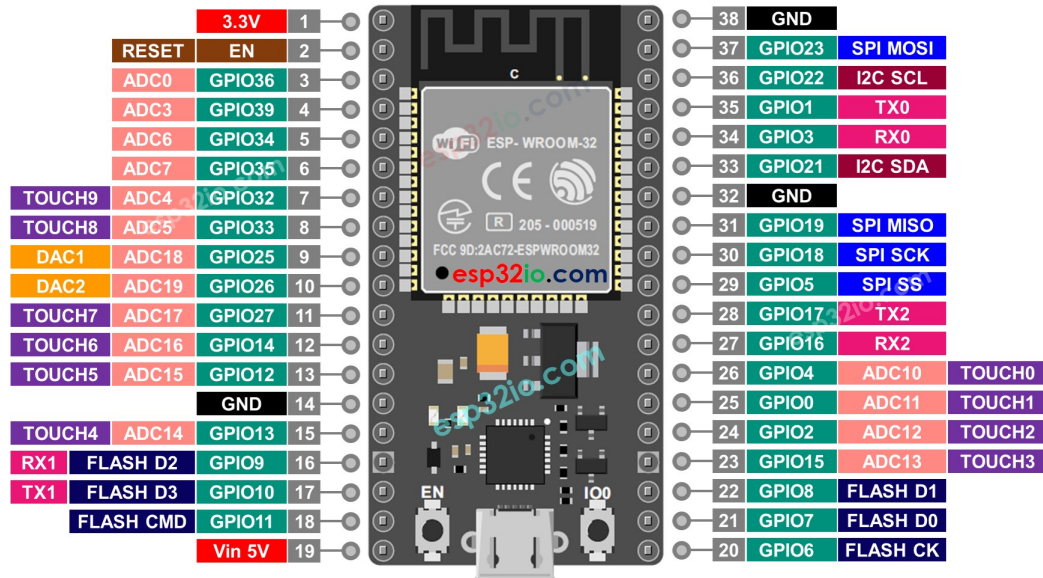


Figura 5.2: Pinout dell'ESP32-DOIT-DEV-KIT v1

Capitolo 6

Software

6.1 WebServer

6.2 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 6.1: Pinout proposto per il dispositivo slave Modbus

Capitolo 7

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.