UNIVERSITÀ DEGLI STUDI
DI GENOVA

DIPARTIMENTO DI INGEGNERIA NAVALE, ELETTRICA, ELETTRONICA E DELLE TELECOMUNICAZIONI
Corso di Laurea in Ingegneria elettronica e tecnologie dell'informazione

Sistema embedded per controllo e movimentazione di osservatorio astronomico

Tesi di laurea triennale

18 LUGLIO 2024

Relatore:

Prof. Rodolfo Zunino

Correlatore:

Prof. Andrea Marenco

Candidato: *Emanuele Castagno*

Sommario

La seguente tesi ha come obiettivo quello di documentare il processo di realizzazione di un sistema elettronico embedded, ovvero un dispositivo integrato o “incastonato” dedicato a compiere elaborazioni per uno specifico scopo. In questo caso il sistema è stato sviluppato per la gestione e il controllo del movimento della cupola dell’osservatorio astronomico “LISA” del Liceo Scientifico Gian Domenico Cassini, situato a Genova.

In particolare, dopo una fase iniziale di sopralluogo, si è deciso di sostituire la scheda di controllo già presente nel suddetto osservatorio con una nuova, riprogettata, più moderna e compatibile con i nuovi protocolli di comunicazione. Per consentire futuri miglioramenti ed aggiunte di funzionalità si è prodotta una buona documentazione. Successivamente, come per ogni sistema embedded, è stato necessario passare al lato software, ovvero lo sviluppo del firmware che abilita il microcontrollore ad apprendere ed elaborare le informazioni del sistema che lo circonda, per poter poi impartire i controlli agli attuatori, che convertono il codice in movimento. La necessità di controllare e soprattutto automatizzare il controllo della cupola di un osservatorio deriva da due peculiari caratteristiche di una struttura di questo tipo, ovvero l’orario di utilizzo, poco pratico per effettuare lezioni in presenza, ma soprattutto la necessità di seguire oggetti in movimento costante, non solo con il telescopio stesso, ma anche con la sottile feritoia che consente a quest’ultimo di osservare l’universo.

Indice

Introduzione

Situazione trovata

Il Liceo Scientifico Gian Domenico Cassini è dotato di un osservatorio astronomico, posto sul terrazzo della sede stessa; esso è composto principalmente da due parti: il telescopio e la struttura che lo circonda, che, a sua volta, è divisa in due parti:

- Una parete circolare fissa, su sono fissati il motore a corrente continua e il sistema di controllo, responsabili del movimento della cupola.
- Una cupola, dotata di una feritoia. La cupola, quando l'osservatorio non è in funzione, è chiusa, ed ha lo scopo di protezione dagli agenti atmosferici, mentre, quando è aperta, grazie alla feritoria consente la vista della volta celeste. La cupola è libera di ruotare sulla parete circolare fissa grazie ad un sistema di ruote.

Figura 1: Osservatorio LISA

Per gestire il pilotaggio del motore, è stata trovata una scheda di controllo basata su un microprocessore PIC16F877A-I/P, con diversi relè a stato solido e altre numerose porte per la gestione della sensoristica e della comunicazione con il telescopio, di cui si parlerà più avanti. Un qualche problema circuitale aveva da tempo reso inoperativa questa scheda che poteva pilotare il motore solo in una direzione. Considerando quindi questo problema e il fatto che fosse arretrata e priva di una buona documentazione, è stato convenuto fin da subito di riprogettare l'intero sistema di controllo.

Obiettivo del progetto

Una volta analizzata la situazione, si è deciso l'obiettivo per il progetto, ovvero quello di rimettere in funzione e modernizzare la struttura trovata con un microcontrollore open-source e facilmente collegabile in rete. Questo aspetto ha il grande vantaggio di rendere quest'ultimo integrabile con ulteriori sistemi di controllo, per esempio quello per automatizzare l'apertura della feritoia posta sulla cupola, oppure l'aggiunta di sensori per rilevare la pioggia e le condizioni atmosferiche, ma soprattutto per rendere possibile in futuro il controllo da remoto.

È dunque necessario realizzare il progetto in maniera modulare e documentata, in modo da rendere lo stesso comprensibile ed espandibile anche da altre persone che in futuro potranno proseguire il progetto.

Il controllore dovrà dunque soddisfare le seguenti specifiche:

- **Abilità di pilotaggio del motore DC** mediante l'interfacciamento con una scheda realizzata appositamente basata su un ponte H di relè,
- **Gestione real-time** del sensore di movimento della cupola e dell'interruttore di fine corsa.
- **Comunicazione con il telescopio** tramite interfaccia seriale, usando il protocollo specifico del produttore dello stesso per ottenere le informazioni necessarie sulla sua posizione.
- **Gestione di dispositivi I/O** per il controllo manuale e la visualizzazione delle informazioni riguardanti lo stato del sistema.
- **Supporto alla comunicazione WiFi/Bluetooth** per implementazioni future.
- **Interfaccia seriale aggiuntiva** per la comunicazione con un pc (per utilizzo software planetario)

Una volta definite le specifiche, si è passati alla realizzazione del progetto, in primis del ponte H di relè e successivamente della scheda per interfacciare il microcontrollore.

Navigazione stellare e punti di riferimento

Un aspetto che occorre precisare per la spiegazione del progetto è quello della navigazione e del puntamento ai corpi celestiali, specificando soprattutto la gestione dei punti di riferimento. In ogni sistema di riferimento è infatti necessario stabilire quali siano i punti cardine fissi e il criterio per il conteggio delle distanze dagli assi.

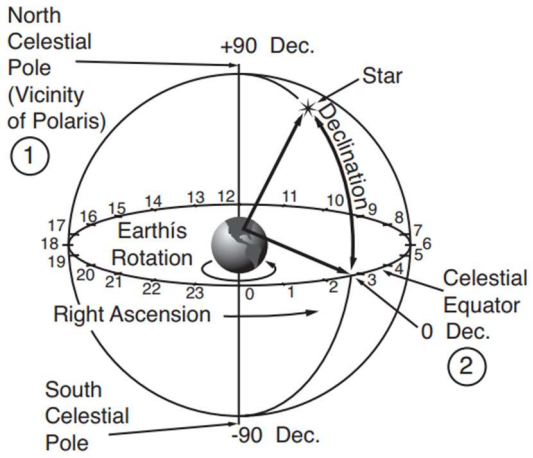

Figura 2: Sistema di riferimento celestiale

Il sistema di riferimento celestiale ha molte analogie con quello terrestre: il punto di riferimento per la mappatura stellare è la stella polare, che coincide (quasi) con il prolungamento dell'asse terrestre passante per il polo nord e il polo sud.

Per esprimere la posizione di un corpo celeste si utilizzano i dati su azimut ed altezza, ovvero gli equivalenti di latitudine e longitudine per la posizione terrestre. L'altezza è un angolo misurato a partire dall'orizzonte e diretto verso l'asse polare terrestre, mentre l'azimut è l'angolo sotteso tra l'oggetto di cui stiamo esprimendo la posizione e la linea dello "zero", che è stata decisa arbitrariamente come quella passante per la costellazione di Pegaso (l'equivalente del meridiano di Greenwich). Inoltre è bene sapere che questa misura viene espressa in ore, minuti e secondi, con un intervallo da 0 a 24 ore, mentre i minuti e i secondi sono sessagesimali.

D'ora in avanti verrà usata solo la coordinata dell'azimut, in quanto il controllo dell'altezza è gestito completamente dal telescopio. Infatti la feritoia presente sulla cupola consente il libero movimento del telescopio in altezza, e il controllore progettato si deve solo assicurare che questa feritoia sia allineata con il corpo celeste che il telescopio stesso sta osservando.

Metodi e strumenti utilizzati

Sistema da controllare per il movimento meccanico della cupola

Per procedere è necessaria una breve ma più precisa descrizione del sistema già presente nell'osservatorio, ovvero la parte da gestire per controllare la rotazione della cupola

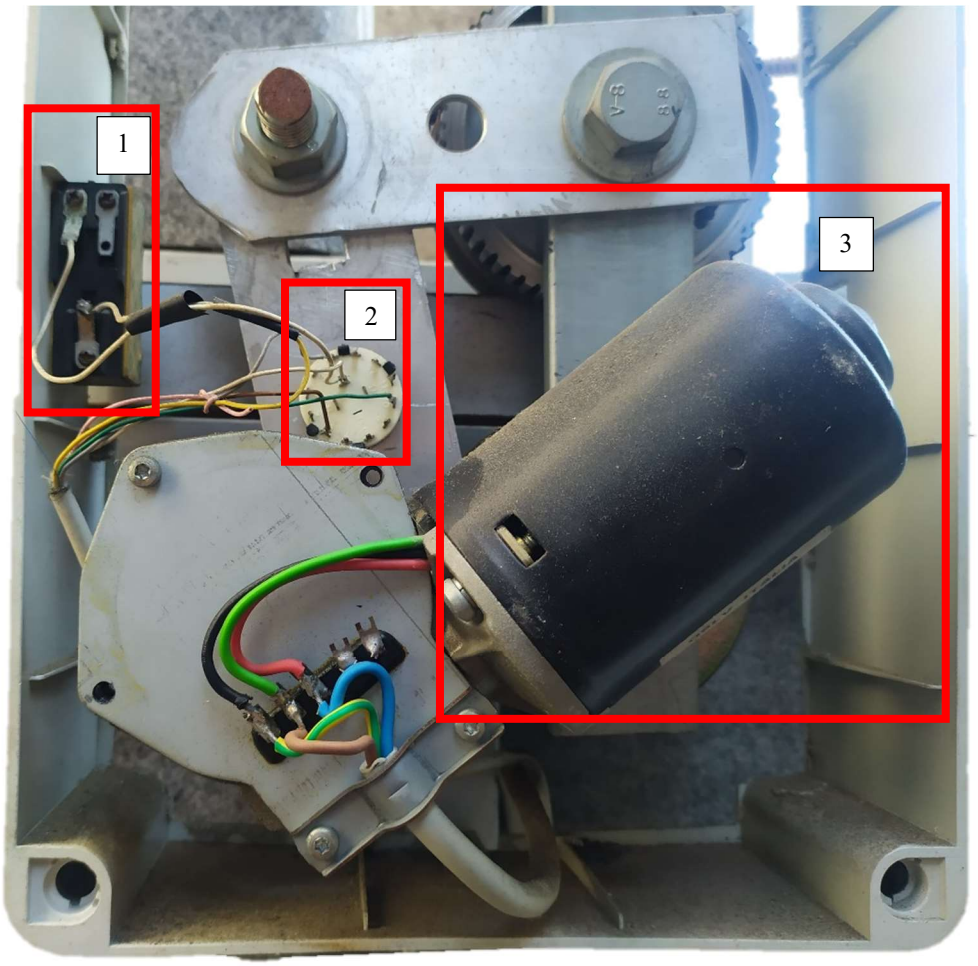

Figura 3: apparato di movimentazione della cupola

1. **Interruttore di "posizione zero":** si tratta appunto di un interruttore che chiude il suo contatto al passaggio di una tacchetta presente sulla rotaia. Questa tacchetta denota la "posizione zero", ovvero il punto di riferimento che consente al microcontrollore di capire a quale grado di rotazione sia la feritoria.
2. **Encoder rotativo:** questo dispositivo è dotato di quattro pin: uno comune di "ingresso" e tre di "uscita". A seconda del verso di rotazione del motore l'encoder metterà in corto il pin centrale con uno dei tre di uscita (per esempio 1 → 2 → 3 → 1 per il verso orario e 1 → 3 → 2 → 1 in senso antiorario).
3. **Motore DC 12V:** il motore da pilotare è un motore a corrente continua con assorbimento massimo sui 5A, che, grazie ad un riduttore di giri, riesce a mettere in movimento la cupola.

Relè di potenza

Figura 4: relè RT114005

Il relè di potenza scelto per il progetto è il RT114005 prodotto dalla SHRACK. Si tratta di un relè SPDT (single pole dual throw) realizzato per montaggio su PCB e con valore massimo di operazione di 12A, ben al di sopra del valore massimo necessario per il pilotaggio del motore.

L'utilizzo di relè consente di isolare galvanicamente il circuito a "bassa potenza" di controllo da quello ad "alta potenza" del motore, mantenendo allo stesso tempo una condizione di sicurezza dai cortocircuiti della parte ad alta potenza anche nel caso di guasto di uno dei due relè.

ESP32-S2 Mini

Il microcontrollore scelto per questa applicazione è un ESP32-S2 mini prodotto da WEMOS. Questo supporta il protocollo wi-fi 802.11 b/g/n, dispone di connessioni ADC, DAC, I2C, SPI, UART e monta un Chip ESP32-S2FN4R2. Le specifiche di questo microprocessore sono riportate nella seguente tabella:

	PIC16F877A-I/P	ESP32-S2FN4R2
Cpu Arch	8 Bit	32 Bit
Cpu clock	20 MHz	240 MHz
RAM	368 B	320 KB
ROM	256 B	128 KB

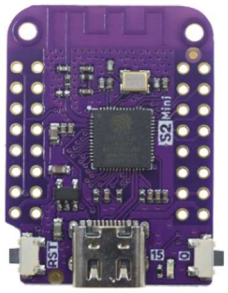

Figura 5: ESP32-S2 mini

Come si può facilmente notare il nuovo chip, che ha anche un costo più contenuto, è decisamente più performante in termini di potenza computazionale rispetto al precedente e dispone inoltre di protocolli di comunicazione non supportati dal circuito integrato sostituito.

DC-DC buck converter

Per l'alimentazione del microcontrollore è stato deciso, poiché si disponeva già di un alimentatore AC-DC da 12V nel quadro elettrico, di utilizzare un convertitore buck DC-DC per abbassare la tensione a 5V per il microcontrollore e la scheda dei relè.

In questo caso la scelta è ricaduta sul MP1584EN, un modulo buck con tensione di uscita variabile e con corrente di uscita fino a 3A, che è stato impostato ad una tensione di uscita di 5V.

Figura 6: MP1584EN buck converter

ESP32-WROVER DEV KIT come simulatore della cupola

La programmazione del microcontrollore principale presentava la difficoltà di non poter provare il firmware durante lo sviluppo, in quanto, ad ogni cambiamento di quest'ultimo, non era conveniente visitare l'osservatorio per effettuare i test sul funzionamento. Al fine di risolvere il problema, è stato deciso di utilizzare un ulteriore microcontrollore per "simulare" il comportamento della cupola durante i test. Questo microcontrollore è stato programmato in modo da emulare il funzionamento dell'encoder, dell'interruttore di "posizione zero" e alcune limitate funzioni della seriale con il telescopio.

Figura 7: ESP32-WROVER

Display Oled I2C

Per rendere il sistema più accessibile e facile da interpretare, si è deciso di aggiungere un piccolo display economico che rendesse agevole consultare le informazioni di posizione della cupola e del telescopio. Lo stesso display inoltre consente la segnalazione di possibili errori che avvengono durante l'utilizzo, come problemi di comunicazione con il telescopio o sulla rete.

Figura 8: Display Oled SSD-1306

EasyEda designer

Il software utilizzato per la realizzazione degli schematici e della fase di layout e routing è EasyEda designer, uno strumento di progettazione gratuito che offre la possibilità di realizzare PCB, ovvero circuiti stampati, a partire dallo schematico fino alla realizzazione fisica di questi mediante l'azienda partner JLCPCB.

Visual Studio Code

Il programma usato per lo sviluppo del firmware è stato Visual Studio Code, un editor di codice sorgente gratuito e molto apprezzato in generale dai programmatori per la leggerezza e la versatilità che lo caratterizzano. Infatti dispone di supporto a numerose estensioni per soddisfare le esigenze di diversi tipi di progetti. Nel nostro caso sono state usate le estensioni di PlatformIO ed ESP-IDF per la programmazione di microcontrollori basati sulle architetture di Espressif.

Connessione I2C

La connessione I2C, acronimo di Inter Integrated Circuit, è un sistema di comunicazione bifilare sviluppato da Philips nel 1982. Viene usato l'omonimo protocollo per la comunicazione su una linea seriale, detta bus, che necessita di solo due linee di trasmissione:

- **SDA** (Serial Data Line): linea di trasmissione dei dati
- **SCL** (Serial Clock Line): linea di trasmissione del segnale di clock

Oltre a queste due linee è necessario collegare anche il riferimento a GND e talvolta è anche obbligatoria la linea di alimentazione con resistori di pull-up opportuni (solitamente tra i 5 e i 10 $k\Omega$) tra quest'ultima e le linee di comunicazione. Grazie a questa interfaccia sarà possibile collegare diversi dispositivi, ognuno con il proprio indirizzo, sul bus. I dispositivi potranno poi svolgere la funzione di master, colui che gestisce lo scambio di dati iniziando la comunicazione, o di slave, ovvero quello che risponde alle richieste del master.

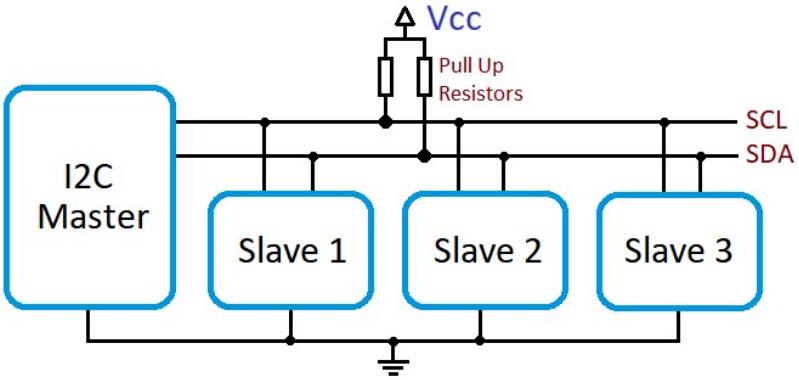

Figura 9: Esempio di connessione I2C

Comunicazione Seriale

La comunicazione seriale utilizza una linea di trasmissione per inviare dati, sotto forma di bit, sequenzialmente. Un dispositivo UART (Universal Asynchronous Receiver-Transmitter), presente sul microcontrollore, ha la funzione di gestire il flusso dei dati, l'impostazione della velocità e la gestione di un buffer.

Nel nostro caso verranno utilizzate due connessioni seriali, una predisposta per il telescopio e una per eventuali connessioni ad un computer; tutto questo per le operazioni di diagnostica e di test, ma anche per il caricamento del firmware. La connessione con il telescopio è effettuata mediante interfaccia RS-232 e necessita di un traslatore di livello apposito per interfacciare la logica a bassa tensione dell'ESP-32 con quella a livelli più alti del telescopio.

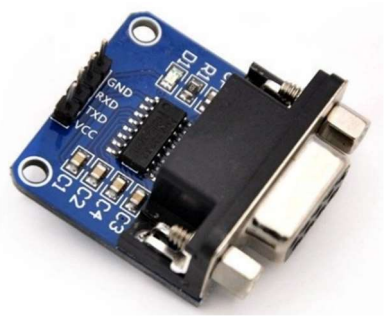

Figura 10: traslatore di livello RS-232

Telescopio Meade LX200ACF

Il telescopio presente nell'osservatorio è un LX200ACF prodotto dalla MEADE INSTRUMENTS, con ottica aplanatica ACF di diametro da 10 pollici e apertura F/10. Dispone inoltre di un montaggio motorizzato di tipologia altazimutale, ovvero con due gradi di libertà: altezza ed azimut.

Viene gestito tramite un telecomando con 145000 oggetti celesti in memoria, che controlla la parte di puntamento del telescopio e la fase di calibrazione iniziale. Sono presenti inoltre numerose porte per il collegamento con dispositivi esterni, tra cui due RS232 che possono essere utilizzate per connettere un computer o un microcontrollore.

Figura 11: Telescopio LX200ACF

Sperimentazione e risultati

Realizzazione del ponte H con relè

La prima fase del progetto si è concentrata sulla realizzazione di una scheda che potesse consentire il controllo di un motore DC 12v con un assorbimento massimo di 5 Ampere. Prima di procedere alla realizzazione su PCB, è stato necessario realizzare lo schematico delle connessioni elettriche.

Schematico

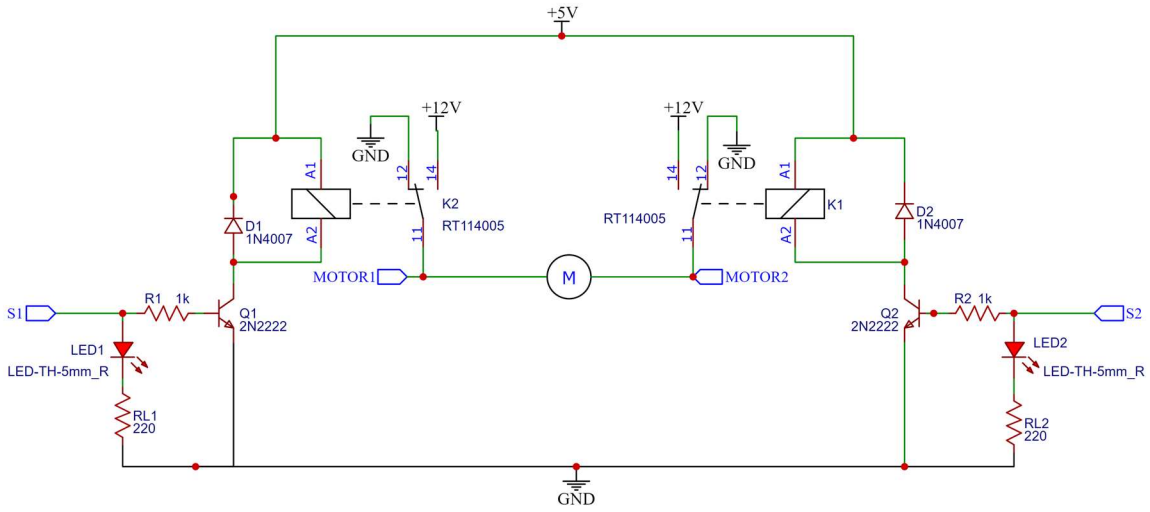

Figura 12: Schematico scheda ponte H di relè

I relè scelti sono dotati di bobine che hanno un assorbimento di potenza intorno al mezzo watt ciascuna e una tale corrente non può essere erogata dai pin GPIO del microcontrollore. Perciò è stato deciso di connettere ad ogni bobina dei relè un transistor che fosse in grado di far passare la corrente necessaria sull'induttore. Tale transistor è collegato con una connessione a emettitore comune e viene portato in zona attiva diretta quando i pin corrispondenti alle porte logiche S1 o S2 saranno su un livello alto; solo a quel punto avverrà la commutazione dello stato del relè. Poiché queste bobine rappresentano un carico induttivo non trascurabile per il nostro circuito, è stato aggiunto un diodo di flyback per ogni bobina in modo da impedire possibili danneggiamenti dovuti ai picchi di tensione che avvengono quando la corrente che passa negli induttori cambia rapidamente. La connessione a ponte H dei relè con il motore consente di avere sulla rotazione della cupola l'effetto schematizzato nella seguente tabella:

S1	S2	
OFF	OFF	Motore spento
OFF	ON	Motore acceso in senso orario
ON	OFF	Motore acceso in senso antiorario
ON	ON	Motore spento

Infine, per rendere più facilmente visualizzabile lo stato di funzionamento del circuito, sono stati aggiunti due led in serie con due resistori (RL1 ed RL2), con l'anodo connesso alle porte S1 ed S2. In questo modo sarà facilmente comprensibile quali siano i relè attivi e quali no.

Prototipo su breadboard

Nella fase successiva si è passati alla realizzazione dei collegamenti fisici su breadboard, al fine di testare i componenti prima di realizzare una vera e propria PCB. La breadboard, infatti, è una tipologia di scheda che consente di connettere rapidamente numerose componenti elettroniche, senza il bisogno della saldatura. Questo aspetto è fondamentale durante la fase di prototipazione in quanto errori e cambi di componenti renderebbero difficile e molto lento lo sviluppo su una scheda millefori con saldature. Ovviamente questa tipologia di scheda non potrà essere usata per il prodotto finale, poiché non è robusta e le componenti, non essendo saldate, potrebbero uscire di posizione durante l'utilizzo sul campo.

Progettazione PCB

Considerando la necessità di avere una scheda robusta e soprattutto in grado di gestire correnti molto alte, è stato deciso di realizzare una scheda PCB sfruttando i servizi offerti da EasyEda e JLCPCB, un produttore di schede PCB custom. Dopo una fase iniziale di realizzazione dello schematico, è stato possibile procedere alla fase di layout, che supporta 2 layer per poter facilitare il passaggio delle piste. Per la parte a bassa potenza sono state scelte piste con una larghezza di $1mm$, mentre per la parte ad alta potenza, quella che mette in collegamento il motore con l'alimentazione, è stata scelta la larghezza massima consentita dalla disposizione dei pin dei relè, che è risultata essere $4{,}7mm$ millimetri. Inoltre, per migliorare ulteriormente la sezione trasversale, è stato deciso di applicare due once di rame per metro quadro anziché la quantità standard di 1 oncia per metro quadro: in questo modo è stato accertata una capacità di conduzione più che sufficiente, mediante l'utilizzo della seguente tabella, presa dallo standard IPC-2152:

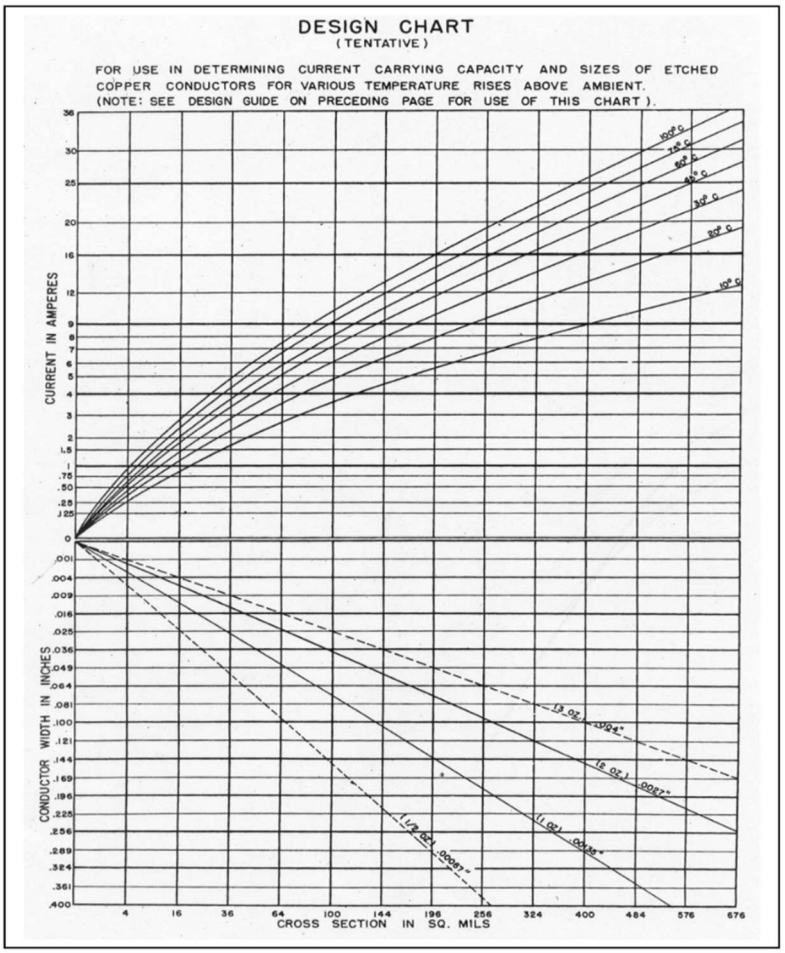

Figura 13: Diagramma della relazione tra sezione trasversale con corrente

Infatti, considerando un riscaldamento delle piste massimo di $10°C$, risulterebbe una corrente massima teorica compresa tra i 9 e 12 A, ben al sopra di quella raggiungibile con l'alimentatore in uso.

Una volta definite le dimensioni delle piste, si è passati alla fase di routing, sfruttando i due layer presenti sulla PCB e ottenendo così la seguente configurazione:

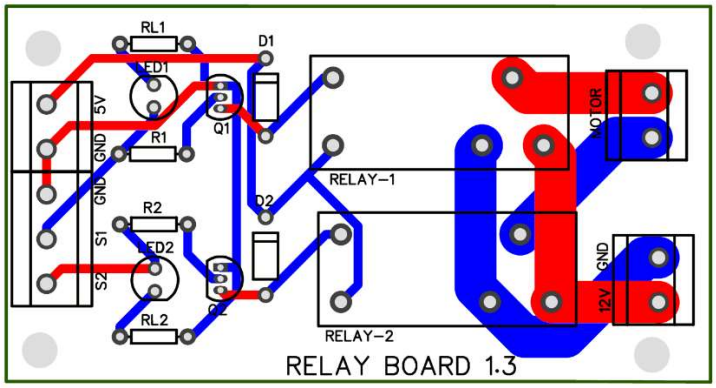

Figura 14: Routing della scheda ponte H di relè

Realizzazione della PCB

Una volta concluso il design della scheda, è stato generato il file gerber da mandare alla fabbrica per la realizzazione fisica della scheda stessa. Una volta approvato, il design verrà mandato in produzione con i seguenti passaggi:

- **Deposizione del rame**: il substrato, ovvero la base per la realizzazione della PCB, realizzata in fibra di vetro impregnata di resina epossidica, passa in un macchinario che depone uno strato di rame, nella quantità desiderata (nel nostro caso due once per metro quadro) su ambo i lati del substrato.
- **Realizzazione e applicazione dei film fotosensibili**: Un plotter stampa il design delle piste di rame su un film trasparente, che viene deposto sulle facciate di rame e fissato mediante un processo di fotoesposizione.
- **Rimozione del rame in eccesso**: tutto il rame che non fa parte del circuito disegnato dal plotter viene rimosso, lasciando dunque solo le piste necessarie alla realizzazione dei collegamenti tra le varie componenti della scheda.
- **Ispezione visiva automatica**: un macchinario apposito controlla se, visivamente, siano presenti sulle piste imperfezioni che potrebbero causare mancati collegamenti o cortocircuiti.
- **Foratura**: la scheda passa in un macchinario che mediante trapani a colonna appositi realizza i fori necessari per l'installazione dei componenti e per i fori passanti, fondamentali per garantire i collegamenti elettrici da una parte all'altra del circuito.
- **Deposizione della vernice isolante e della legenda**. La fase conclusiva prevede la deposizione di due strati di vernice finali: nel nostro caso la mascheratura verde è deposta per isolare il circuito, ad eccezione dei pad usati per collegare i componenti, mentre la vernice bianca ha la funzione di legenda, per il posizionamento dei componenti stessi.

Il risultato ottenuto è il seguente:

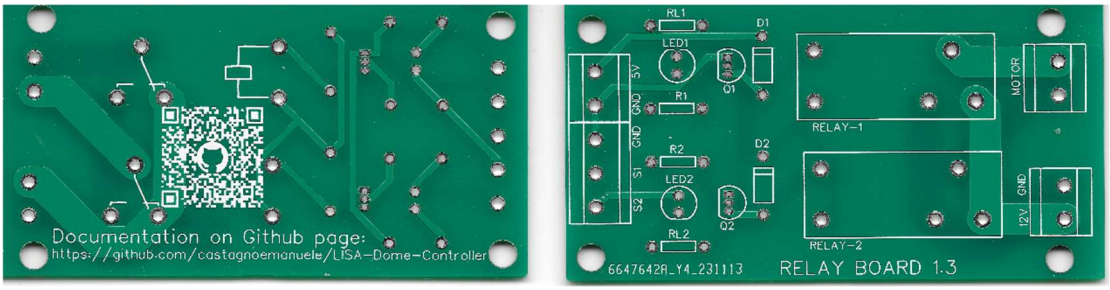

Figura 15:scheda PCB stampata

Progettazione della scheda di supporto al microcontrollore

Dopo aver concluso lo sviluppo della scheda dei relè, si è dovuta realizzare la circuiteria necessaria per interfacciare il microcontrollore con la sensoristica e con l'interfaccia utente. La scelta questa volta non è ricaduta sulla creazione di una PCB, poiché non necessitava di portare alte correnti e soprattutto perché, a differenza della scheda dei relè, quella di supporto al microcontrollore non è un prodotto "fatto e finito", ma probabilmente in futuro potrà dover essere modificato seguendo l'evoluzione del progetto, per esempio con l'aggiunta di altri sensori o dispositivi di interfaccia. Pertanto la decisone è stata quella di costruirla su una scheda millefori, in modo da garantire sia una buona robustezza durante l'utilizzo, che una flessibilità alle modifiche. Considerando le specifiche sopra definite, possiamo stabilire quali siano le connessioni che la scheda di supporto al microcontrollore deve avere:

- Interfaccia I^2C per display oled.
- Interfaccia seriale per comunicazione con il telescopio.
- Connessione con encoder e interruttore di fine corsa.
- Connessione con ponte H.
- Connessione con pulsantiera esterna.
- Gestione dell'alimentazione.
- Gestione con resistori di pull-up della pulsantiera integrata sulla scheda.

Le connessioni con i sensori posti all'esterno saranno fisicamente effettuate con morsetti a vite, che consentono un collegamento sicuro, ma allo stesso tempo non permanente. Oltre al microcontrollore saranno presenti altri dispositivi, ovvero il convertitore buck DC-DC e il display I2C, che è stato predisposto anche con una connessione esterna. In base a queste considerazioni si è realizzato il seguente schematico della scheda:

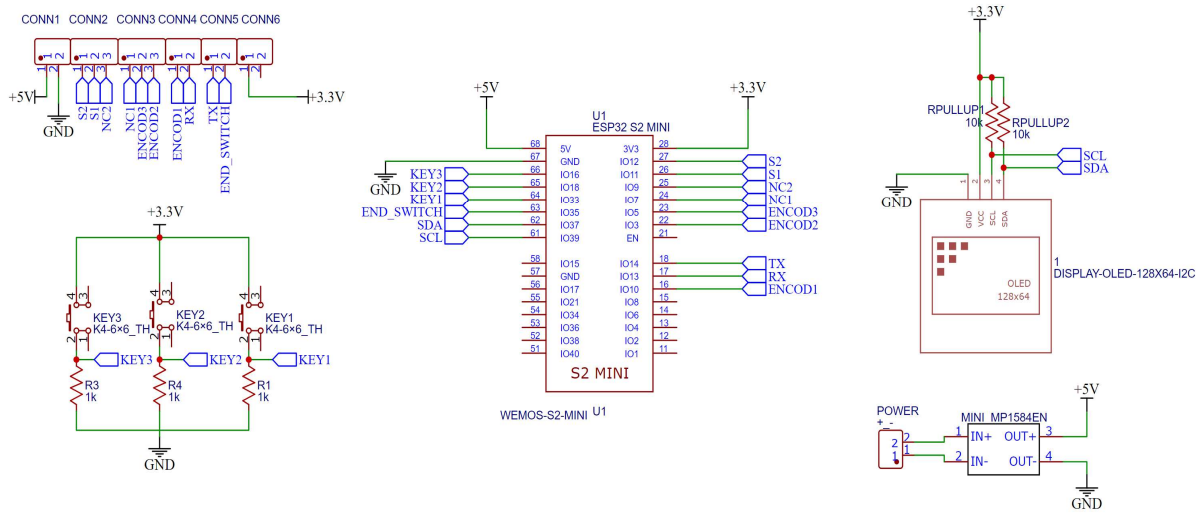

Figura 16: Schematico della scheda di supporto al microcontrollore

Prototipo su breadbord e risultato finale

Nella prima fase di prototipazione, come detto precedentemente, è stato usato un supporto breadboard per realizzare le connessioni e provare il circuito prima di passare alla saldatura. Il primo sistema realizzato è il seguente:

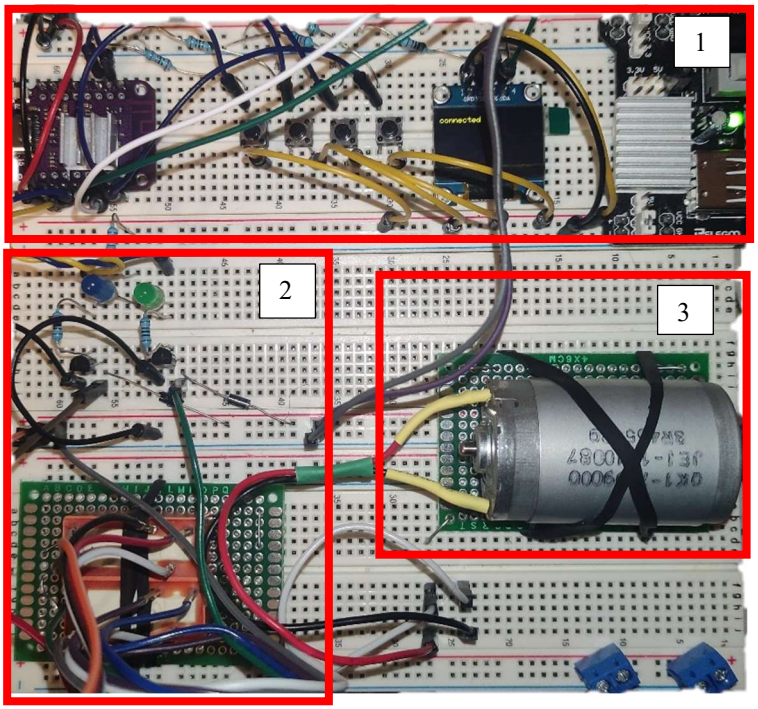

Figura 17: Breadboard usata per la sperimentazione

Come si può notare, è composto da tre parti:

1. Sezione di prova per la scheda del microcontrollore, composta da microcontrollore, pulsantiera, schermo oled e alimentatore dc.
2. Sezione di prova della gestione dei relè, con i transistor per pilotarli e i led per segnalare il funzionamento.
3. Motore dc $12V$ a bassa potenza per testare il funzionamento dei relè.

Una volta sviluppata la parte di firmware per i test e confermato che il circuito stesse funzionando, si è passati alla saldatura dei componenti sulle schede, seguendo gli schematici delle figure 12 e 16.

La scheda dei relè di potenza realizzata è la seguente:

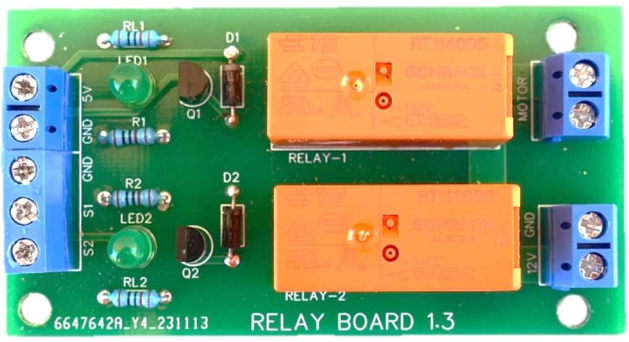

Figura 18: Scheda dei relè di potenza

La scheda di supporto al microcontrollore realizzata è la seguente:

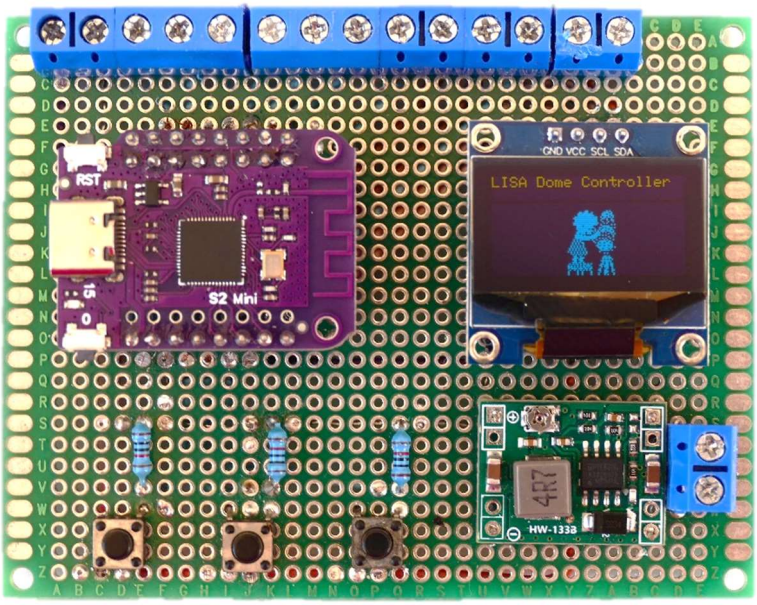

Figura 19: scheda di supporto al microcontrollore

Come si può notare, sono presenti i resistori da 10 $K\Omega$, usati come resistenze di pull-down. Tale resistenza viene applicata nei casi in cui si deve misurare un livello logico, per esempio su un pin di ingresso di un microcontrollore, che rimarrebbe altrimenti "floating", ovvero sospeso, non connesso elettricamente. La resistenza viene scelta con un valore abbastanza alto da consentire un basso consumo di corrente, ma permettendo sempre di far rimanere l'ingresso logico ad un livello stabilito anche quando l'interruttore è in uno stato aperto. Gli interruttori e l'interfaccia per il display I2C saranno inoltre disponibili mediante un connettore JST, per facilitare il collegamento con la pulsantiera e il display esterni alla scatola di derivazione elettrica in cui sono ospitati tutti i componenti elettronici. Il supporto per il display è stato realizzato mediante stampa in 3D.

Infine si sono realizzati i collegamenti tra le schede ed è stato connesso il secondo microcontrollore, l'ESP-32, con funzione di simulatore della cupola, per poter passare alla fase di programmazione e sviluppo del firmware.

Sviluppo del firmware per il microcontrollore.

Simulatore della cupola

Lo sviluppo del firmware per controllare un sistema complesso come quello di un osservatorio astronomico, presenta la difficoltà di non poter essere sempre fisicamente nei pressi di quest'ultimo, sia perché si tratta di un ambiente poco predisposto per lo sviluppo di software, sia perché, essendo un edificio scolastico, accedervi dopo ogni cambiamento e fase di sperimentazione sarebbe stato impraticabile. Per questa ragione è stato deciso di creare un sistema che simulasse il funzionamento della cupola, specialmente dell'encoder rotativo, al fine di poter programmare tranquillamente da remoto. Questo simulatore avrà come input due segnali, quelli che il microcontrollore principale invierà alla scheda a ponte H dei relè. Lo stesso simulatore, monitorando questi due pin, potrà decidere in che ordine attivare tre degli output, ovvero i contatti corrispondenti a quelli dell'encoder rotativo. Quindi, a seconda del comando di attuazione del motore in senso orario o antiorario, questo microcontrollore simulerà la chiusura di uno dei tre contatti alla volta in modo circolare e periodico.

Inoltre il sistema, se rileva una prolungata attivazione del motore in una direzione, invierà un segnale di "posizione zero", per simulare il corretto funzionamento della funzione di reset, che verrà spiegata in seguito.

Una volta sviluppata la base per poter programmare il sistema da remoto, si è potuto sviluppare il firmware per il microcontrollore principale.

Gestione della pulsantiera

La gestione della pulsantiera consente all'utente di poter controllare manualmente la posizione della cupola quando questo sarà necessario. Attualmente il sistema prevede tre tasti, due per muovere la cupola in senso orario e antiorario e uno per avviare la procedura di reset alla posizione zero. Tale procedura consiste semplicemente nel controllare lo stato attuale della cupola, per poi farla ruotare nella direzione corretta al fine di riportarla alla posizione di partenza, corrispondente al punto in cui l'interruttore di posizione zero viene chiuso da una tacca metallica posizionata sulla rotaia che scorre concordemente alla cupola, visibile nella figura 20 a sinistra.

Figura 20: tacca di posizione zero

I tasti vengono monitorati dal microcontrollore per mezzo di interrupt, un segnale asincrono che interrompe la normale esecuzione del programma per gestire eventi real-time, nel nostro caso la pressione di uno dei tasti. Questa scelta è stata fatta per consentire al microcontrollore di poter continuare a monitorare l'encoder e lo stato del telescopio durante la rotazione della cupola, senza il bisogno di verificare se il tasto sia ancora premuto o meno ad ogni ciclo del loop. Una volta ricevuto il segnale di interrupt, il microcontrollore interrompe l'esecuzione del codice e passa ad eseguire un programma ISR (Interrupt service routine), che attiverà opportuni flag, realizzati nel nostro caso con variabili bool, poi gestiti dal normale loop quando possibile. Per esempio, se la variabile corrispondente al comando di movimento orario della cupola risultasse "vera", verrà iniziata la rotazione oraria mediante l'attivazione a livello alto del pin corrispondente.

Gestione dell'encoder

L'encoder rotativo è l'unico elemento presente nella cupola che ci consente di avere un'idea di quanto questa si stia effettivamente spostando. Esso è connesso al motore mediante la cinghia che porta la trasmissione dal motore all'ingranaggio che a sua volta mette in movimento la cupola. La rotazione dell'albero dell'encoder chiude il collegamento tra il pin comune e uno dei tre contatti alla volta. Grazie a questo sistema possiamo dunque non solo capire quanti giri stia facendo il motore e quindi quanto si stia spostando la cupola, ma anche comprendere in che direzione quest'ultima si stia muovendo. Sarebbe stato infatti sufficiente un solo contatto per ottenere la velocità di spostamento della cupola, mentre i due pin aggiuntivi danno informazioni sulla direzione.

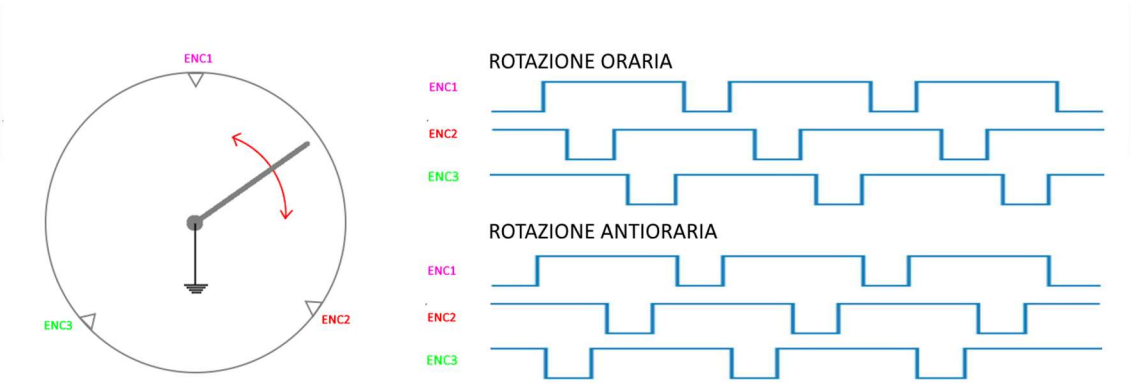

Figura 21: Funzionamento dell'encoder rotativo

Dal punto di vista software, è stata creata una classe apposita per gestire questo oggetto; è infatti necessario poter salvare le variabili relative alla posizione e altre funzioni della cupola, ovvero:

```
void checkEncoder(Encoder& encoder1);
```

Chiamata dopo un interrupt, controlla quale pin dell'encoder si sia attivato e capisce, a seconda dell'impulso ricevuto precedentemente, in che direzione questo si stia muovendo.

```
void updatePosition(Encoder& encoder1);
```

Chiamata dopo un cambiamento rilevato da checkEncoder, aggiorna lo stato della posizione e stampa sulla seriale le informazioni relative al movimento

```
void resetPosition(Encoder& encoder1, Button& resetButton, Button& limitSwitch);
```

Avvia la procedura di reset della posizione a quella denominata con zero, controllando prima in quale direzione sia più opportuno far ruotare la cupola

```
long countTicksFullRotation (Encoder& encoder1, Button& resetButton, Button& limitSwitch,
Preferences& preferences);
```

Avvia la procedura di conteggio del numero di "tick" dell'encoder per fare una rotazione completa, dopodiché salva i valori sulla EPROM dell'ESP32, affinché siano disponibili all'avvio successivo.

```
int convertTicksToDegrees(int ticks, Encoder& encoder1);
```

Grazie ai dati sulla rotazione completa ottenuti precedentemente, ricava la posizione della cupola in gradi, partendo dal numero di tick dell'encoder.

```
void saveData (int data, const char *address, Preferences& preferences);
```

Questa funzione gestisce, mediante la libreria preferences, il salvataggio dei dati di posizione sulla EPROM dell'ESP32, per consentire al microcontrollore di avere le informazioni, anche dopo lo spegnimento, su posizione e numero di tick per un giro completo.

Gestione del display

Per rendere lo stato del sistema più facilmente interpretabile, senza il bisogno di un computer collegato alla seriale, è stato deciso di aggiungere un piccolo display. Si è optato di usare un display SSD_1306, dotato di un piccolo pannello oled da 0.96 pollici, e una risoluzione di 128×64 pixel. Esso viene gestito tramite interfaccia I2C e le librerie usate sono state sviluppate da Adafruit.

Il particolare display è diviso in due sezioni: una barra superiore con pixel gialli, che verrà usata come barra di stato, contenente le informazioni su connessione wifi, posizione della cupola e posizione del telescopio, mentre la parte inferiore, con pixel azzurri, mostrerà il log dei vari messaggi sui comandi che sta eseguendo e dei possibili errori rilevati. Sono state scritte delle funzioni, appartenenti alla classe relativa al display, per gestire la stampa dei messaggi e dello stato del sistema; in particolare:

```
void initDisplay(Adafruit_SSD1306& display, Encoder& encoder, Telescope& LISA;
```

La funzione inizializza il display, avviando la connessione I2C sui pin corretti e mostrando il logo personalizzato, ispirato al nome dell'osservatorio "LISA".

```
void printWifiStatus(Adafruit_SSD1306& display);
```

Questa funzione è stata predisposta per quando verrà implementato l'uso del wi-fi e verrà usata per stampare l'indirizzo IP del dispositivo nonché altre possibili informazioni relative alla connessione wi-fi sulla prima riga dello schermo.

```
void printPositionStatus(Adafruit_SSD1306& display, Encoder& encoder, Telescope& LISA);
```

Questa funzione si occupa di stampare e aggiornare le informazioni relative allo stato attuale della posizione della cupola e del telescopio.

Gestione del telescopio

La gestione della comunicazione del telescopio è una parte molto importante del progetto, in quanto è ciò che consente di ottenere informazioni sulla posizione del telescopio, per poi poter muovere la cupola e seguire l'oggetto che il telescopio sta puntando attraverso la feritoia. Il telescopio Meade LX200ACF è dotato di due porte seriali con interfaccia proprietaria ma adattabile alla porta RS-232; esso utilizza queste porte per comunicare normalmente con un computer, ma in questo caso una è stata dedicata al microcontrollore.

Il protocollo di comunicazione è disponibile nella documentazione del telescopio ed è una lista di comandi che, una volta inviati sulla seriale dal nostro microcontrollore/computer, verranno gestiti dal telescopio stesso che avvierà particolari azioni, per esempio il movimento verso la stella polare, oppure richiederà informazioni, come nel nostro caso, per capire la posizione attuale del telescopio, ottenibile con la stringa ":GR#". Una volta inviato il comando, il telescopio risponderà con la sua posizione attuale, espressa come "$HH:MM:SS$#", ovvero ore, minuti e secondi, con un carattere terminatore "#". Questa posizione, come spiegato nell'introduzione, è calcolata rispetto ad un punto di riferimento coincidente con la costellazione di Pegaso. In questa fase il microcontrollore dovrà convertire mediante l'apposita funzione il valore ricevuto in gradi, affinché si possa calcolare lo spostamento necessario per allineare la feritoia della cupola con la posizione di puntamento del telescopio.

A livello fisico il microcontrollore è stato predisposto per avere una seconda porta seriale dedicata al telescopio, la quale si collegherà ad un traslatore di livello per poter mettere in comunicazione l'ESP32 che ha livelli logici a 3.3V, con il telescopio che invece funziona a 12V.

Per gestire con il firmware questo elemento, è stata creata una classe apposita con i metodi per controllare la posizione e per ottenerla. La chiamata di queste funzioni verrà effettuata automaticamente da un timer, che attiva un segnale di interrupt ogni dieci secondi per avviare la comunicazione con il telescopio e ottenere dati aggiornati sulla sua posizione.

Il firmware è stato testato e trovato funzionante quando il comportamento del telescopio è stato simulato dal microcontrollore "simulatore" o da un computer, ma non con il telescopio stesso.

Debouncing dell'interruttore di posizione zero

Un problema riscontrato durante la fase di sperimentazione è stato quello del rumore elettrico che causava notevoli disturbi sull'interruttore di posizione zero. Tale situazione era particolarmente problematica perché una lettura scorretta dell'interruttore causa una completa perdita di calibrazione della posizione, in quanto questa viene resettata a zero quando invece non dovrebbe esserlo. Questo fenomeno, probabilmente causato dagli impulsi di corrente provocati dall'encoder o dall'interferenza elettrica dovuta al motore molto vicino, sono stati risolti collegando un circuito RC che andasse ad aggiungere una costante di tempo al fronte di salita dell'interruttore, in modo da filtrare i disturbi ad alta frequenza causati dal sistema circostante.

Ai capi dell'interruttore è stato allora collegato un condensatore in serie con un resistore, che consente di scaricare

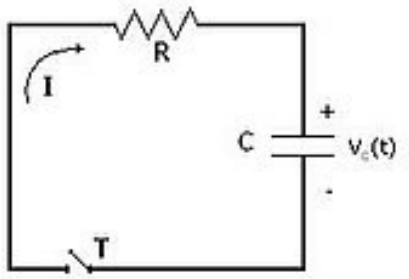

Figura 22: schema del Circuito RC

la corrente quando l'interruttore viene chiuso. Il sistema allora si comporterà mediante la legge:

$$\tau = RC$$

È stata scelto un condensatore da 33 μF accompagnato da una resistenza da 100 Ω, per cui avremo che $\tau = 3{,}3ms$. Infine si è verificato che la resistenza potesse dissipare la potenza necessaria e, una volta saldato il circuito RC all'interruttore, si è appurato che il problema fosse risolto.

Contributo personale e considerazioni conclusive

Questa tesi rappresenta il primo passo per poter rendere controllabile da remoto un osservatorio astronomico, elemento importante per la ricerca scientifica, l'osservazione dell'universo e in questo caso anche uno strumento educativo efficace.

Il progetto, seppur meno sviluppato in confronto a quello che si potrebbe vedere in altri ambiti professionali, ha comunque seguito tutti i passaggi necessari per la progettazione di un generico sistema embedded, partendo dalle specifiche per passare allo schematico, la prototipazione, la progettazione di due schede, di cui una stampata, e infine lo sviluppo del firmware. Grazie a questa varietà di fasi ho potuto mettere a frutto numerose competenze acquisite durante gli anni degli studi triennali, iniziando dalla teoria dei circuiti, passando poi all'elettronica e ai sistemi embedded e infine arrivando alla programmazione. Questo percorso mi ha insegnato molto, poiché ha richiesto un'ampia varietà di conoscenze su materie diverse.

Il lavoro comunque non è concluso, ma quel che è stato svolto è fondamentale per poter consentire lo sviluppo futuro; per questo il codice è stato predisposto per essere ampliabile ed è stato documentato per consentire ulteriori miglioramenti, mentre il sistema che era presente non era operativo, non si disponeva del codice sorgente e non supportava moderni protocolli di comunicazione, oltre al fatto di essere antiquato dal punto di vista computazionale. Infine, il costo del sistema complessivo si aggira sulle poche decine di euro, mentre un sistema apposito già realizzato, come quello che è stato sostituito, si trova in commercio con un prezzo mediamente al di sopra di uno o due centinaia di euro.

L'obiettivo finale sarà quello di sviluppare un sistema più completo e automatico, e, sfruttando le risorse disponibili della rete, renderlo controllabile da remoto per facilitare l'insegnamento a distanza, dando la possibilità a tutti gli utenti, specie gli studenti, di usufruire di questa piattaforma anche in orari serali o notturni. Molti altri osservatori più grandi e famosi non supportano ancora certe funzionalità di automatizzazione, rendendo così necessaria la presenza di un operatore che controlli manualmente la rotazione della cupola. Questo può essere un ostacolo all'acquisizione dei dati in maniera automatica; basti pensare alle possibilità che si avrebbero con una automatizzazione completa dell'osservazione di corpi celesti su periodi di tempo prolungati, come un'intera notte o diversi giorni consecutivi.

In conclusione si può affermare che questa tesi mi ha consentito sia di approfondire le competenze acquisite durante gli studi universitari, sia di mettere a frutto le capacità personali derivanti dalla mia passione per i dispositivi elettronici coltivata negli anni precedenti. Spero che questo progetto contribuisca in maniera positiva all'osservazione dell'universo ed al suo studio a livello scolastico.

Riferimenti Bibliografici

1. Dispense di Sistemi Elettronici Embedded – Rodolfo Zunino
2. Coordinate astronomiche
   https://en.wikipedia.org/wiki/Astronomical_coordinate_systems
3. Manuale di istruzioni del telescopio
   https://www.meade.com/downloadEntityFile/assets/product_files/instructions/LX200ACF_REV5.pdf
4. Datasheet relè di potenza
   https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FRT1%7F4%7Fpdf%7FEnglish%7FENG_DS_RT1_4.pdf%7F1393239-7
5. Informazioni relè
   https://en.wikipedia.org/wiki/Relay
6. Modello per stampa 3d del supporto al display
   https://www.thingiverse.com/thing:3080488
7. Informazioni sul software di progettazione utilizzato
   https://easyeda.com/it
8. The Value of IPC-2152 - Michael R. Jouppi
   https://www.ipc.org/system/files/technical_resource/E7%26S22_03.pdf
9. Processo di produzione PCB
   https://www.pcbcart.com/article/content/PCB-manufacturing-process.html
10. Informazioni sull'utilizzo dei transistori di pull-up
    https://en.wikipedia.org/wiki/Pull-up_resistor
11. Libreria Preferences (Utilizzo EPROM ESP-32)
    https://github.com/vshymanskyy/Preferences
12. Libreria Adafruit GFX
    https://github.com/adafruit/Adafruit-GFX-Library
13. Libreria Adafruit SSD1306
    https://github.com/adafruit/Adafruit_SSD1306
14. Informazioni circuiti RC
    https://en.wikipedia.org/wiki/RC_circuit