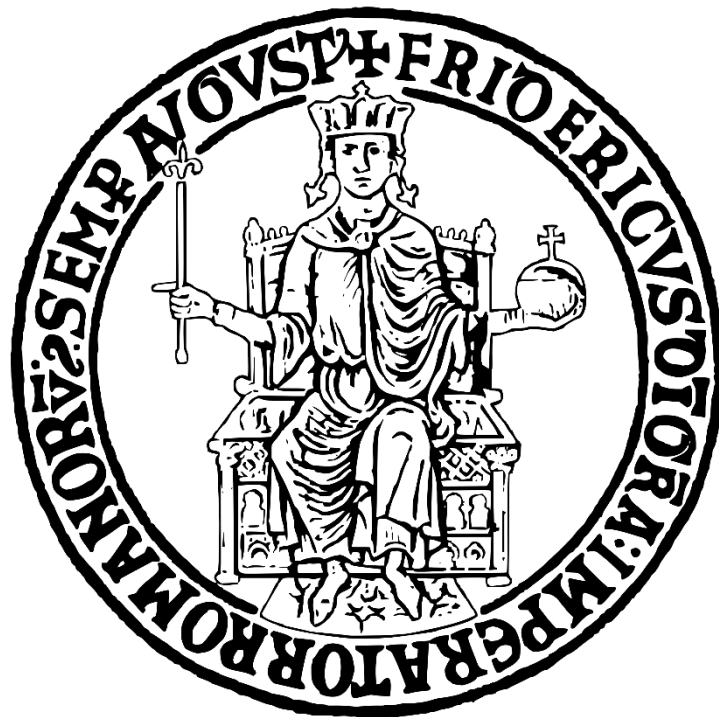


Progetto di sistemi embedded

Computer System Design, Prof. Nicola Mazzocca

Iorio Domenico M63001251

Maddaluno Manuel M63001208



Indice

Computer System Design, Prof. Nicola Mazzocca	1
1. Esercizio Prova Intercorso	3
1.1 Descrizione Generale.....	3
1.2 Protocolli di comunicazione	5
1.3 Configurazione Periferiche e Pin	6
1.3.1 Periferica PIA	6
1.3.2 Periferica UART.....	7
2. Progetto Automatic Irrigation	7
2.1 Descrizione Generale.....	7
2.2 Protocolli di comunicazione	9
2.3 Board STM32F3	9
2.3.1 Comunicazione seriale tramite UART	9
2.3.2 Sensore di umidità e ADC	10
2.3.2 Digital Relay e Pompa	11
2.4 Board STM32L4.....	12
2.4.1 Comunicazione seriale tramite UART con altra board	12
2.4.2 Comunicazione seriale tramite UART con PC.....	12
3. Riferimenti	13

1. Esercizio Prova Intercorso

1.1 Descrizione Generale

Il primo esercizio proposto riguarda la riproduzione della prova intercorso del 25/05/21 su due discovery board della famiglia STM, in particolare la STM32F3 e la STM32L475, le quali realizzano rispettivamente un nodo ricevitore e un nodo trasmettitore. L'architettura complessiva è mostrata nelle seguenti immagini in due viste differenti. In figura 1.1 è rappresentata una vista logica, mentre in figura 1.2 una vista fisica.

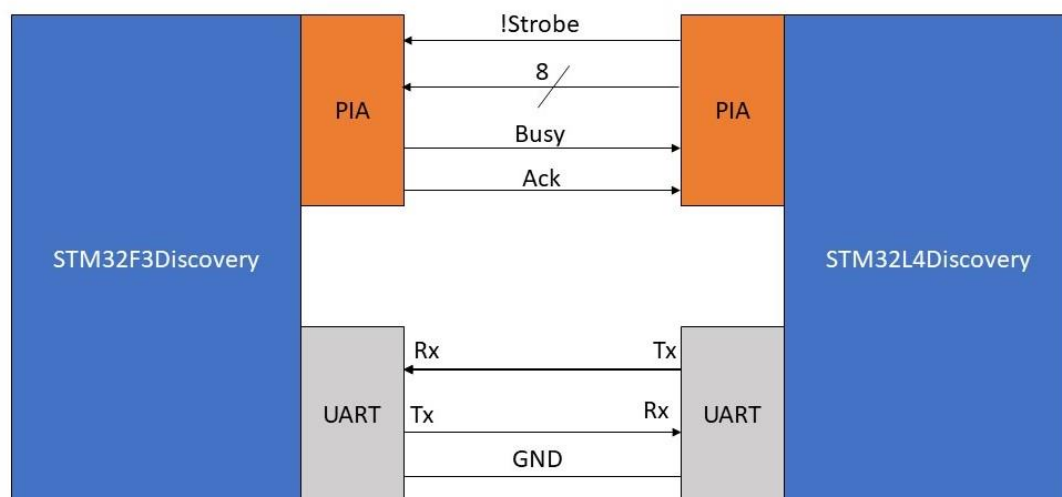


Figura 1.1.1 - Architettura Logica

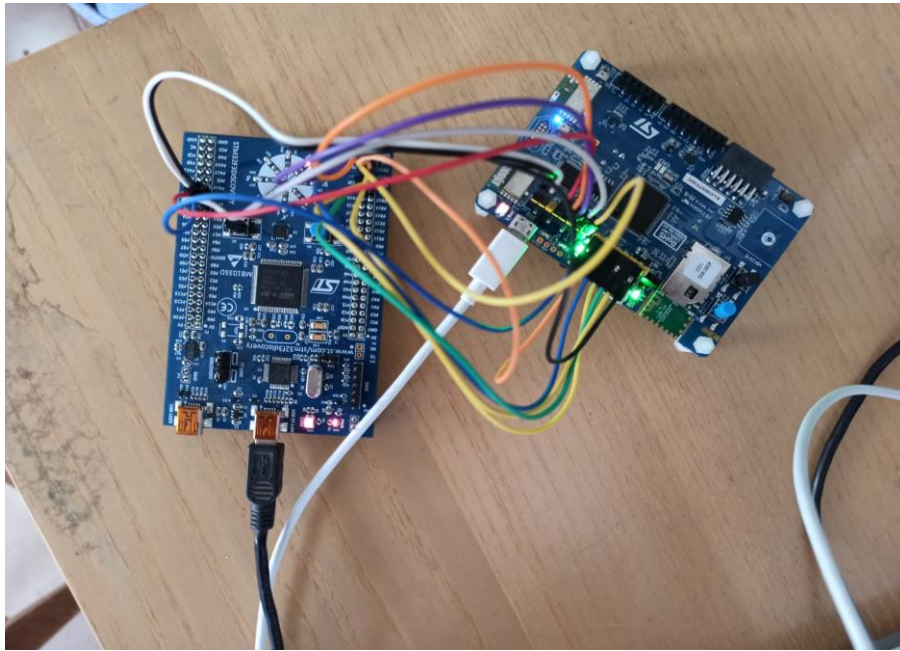


Figura 1.1.2 - Architettura Fisica

Sul nodo ricevitore (STM32F3) entrambe le periferiche gestiscono la ricezione dei messaggi mediante il meccanismo delle interruzioni; i livelli di priorità scelti sono **1** per la PIA e **2** per l'UART. Sul nodo trasmettitore (STM32L4) l'UART utilizza la propria interruzione per gestire l'invio dei messaggi, mentre la comunicazione mediante la periferica PIA è avviata dall'interruzione legata alla pressione del PUSHBUTTON. Quest'ultima è anche configurata per essere l'interruzione più prioritaria all'interno del nodo.

Il sistema realizzato prevede che il nodo ricevitore continui a ricevere messaggi su entrambe le periferiche fino al momento in cui si verifica almeno una delle seguenti condizioni:

- **Il nodo ricevitore riceve due messaggi sulla PIA:** all'arrivo del secondo messaggio verrà alzato un flag che segnala la fine della comunicazione con entrambe le periferiche. Se il secondo messaggio inviato sulla PIA arriva mentre la ISR di ricezione dall'UART è in corso, viene letto anche il carattere elaborato dalla ISR prima di terminare la comunicazione.
- **Il nodo ricevitore rileva che l'ultimo carattere di un messaggio ricevuto sull'UART contiene tutti 0:** in questo caso verrà interrotta la comunicazione con l'UART, mentre continuerà quella con la PIA.
- **Il nodo ricevitore rileva che sono stati inviati un numero di messaggi pari ad una soglia prefissata:** anche in questo caso terminerà la comunicazione con l'UART ma non con la PIA.

1.2 Protocolli di comunicazione

Poiché nessuna delle due board è equipaggiata con un'interfaccia parallela, la comunicazione tra le due PIA è realizzata mediante l'utilizzo delle GPIO. In particolare, ne vengono utilizzate 8 per simulare l'area dati e 3 per i segnali di !STROBE, BUSY e ACK, necessari per realizzare il protocollo di handshake tra le due periferiche. L'esecuzione del protocollo è la seguente:

- Il nodo trasmettitore:
 - Aspetta che BUSY non sia asserito tramite polling
 - Trasmette il dato e asserisce !STROBE
 - Aspetta che ACK sia asserito tramite polling
- Il nodo ricevitore:
 - Aspetta che !STROBE sia asserito. In questo caso la variazione sul fronte di discesa di questo segnale genera un'interruzione
 - Asserisce BUSY e inizia a ricevere il dato
 - De-asserisce BUSY e asserisce ACK

Nel seguito è riportato il diagramma temporale.

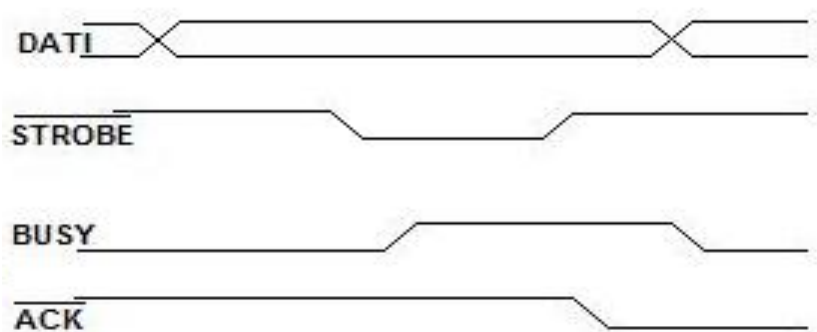


Figura 1.2.1 - Diagramma temporale PIA

Anche il protocollo di comunicazione relativo alle due periferiche UART presenta una particolarità. In questo caso è stato implementato un meccanismo di conferma di ricezione del messaggio da parte del nodo ricevitore per impedire al trasmettitore di inviare un nuovo messaggio se quello precedente è ancora in fase di ricezione, impedendo così il verificarsi di problematiche legate all'overrun. Dunque, i due sistemi si scambiano messaggi in modo alternato, il messaggio dalla board STM32L4 alla STM32F3 è per i dati veri e propri, mentre quello che manda l'STM32F3 all'STM32L4 è un messaggio di Ack.

1.3 Configurazione Periferiche e Pin

Si analizzano adesso nel dettaglio i collegamenti delle periferiche e i relativi Pin utilizzati.

1.3.1 Periferica PIA

La periferica di comunicazione parallela è stata configurata nel seguente modo.

Dal punto di vista del trasmettitore (STM32L4) sono stati configurati i pin da PA2 a PA7, PA15 e PB4 in modalità output per la trasmissione del dato parallelo (byte). Sono stati configurati poi i pin per i segnali di handshake. In particolare, il pin PB2 in output per il segnale di !Strobe, il pin PB1 in input per il segnale Busy (il quale ci informa che il ricevitore è occupato) e il pin PB0 in input per l' Ack del ricevitore. Inoltre per questo nodo è stato impostato il pin PC13 (collegato al pulsante) come fonte di interruzione esterna (EXTI13) ed è stato assegnato il livello di priorità 1. In questo modo alla pressione del pulsante vengono inviati i dati tramite PIA.

Mentre per quanto riguarda il ricevitore (STM32F3) sono stati configurati i pin da PD8 a PD15 in input per la ricezione parallela del dato. Il pin PD1 in input sensibile sul fronte di discesa e inizializzato come fonte di interruzione esterna (EXTI1) per il segnale di !Strobe il quale segnala che il ricevitore è pronto a inviare dati. Anche qui il livello di priorità è 1. Il pin PD2 in output per il segnale Busy e il pin PD0 sempre in output per il segnale Ack.

Nella tabella seguente sono riportati i collegamenti.

STM32L4 Pin		Direzione	STM32F3 Pin	
PA2	Data	->	PD8	Data
PA3	Data	->	PD9	Data
PA4	Data	->	PD10	Data
PA5	Data	->	PD11	Data
PA6	Data	->	PD12	Data
PA7	Data	->	PD13	Data
PA15	Data	->	PD14	Data
PB4	Data	->	PD14	Data
PB2	!Strobe	->	PD1	!Strobe
PB1	Busy	<-	PD2	Busy
PB0	Ack	<-	PD0	Ack

Figura 1.3.1.1 - Tabella Pin PIA

1.3.2 Periferica UART

La periferica di comunicazione seriale è stata configurata nel seguente modo.

Sia per trasmettitore che per ricevitore sono state utilizzate le UART4 e sono state configurate in modalità asincrona, 8 bit di messaggio senza bit di parità, 1 bit di stop e un baud rate di 115200 bit/sec.

Dal punto di vista del trasmettitore (STM32L4) è stato configurato il pin PA0 per la trasmissione e il pin PA1 per la ricezione.

Mentre per quanto riguarda il ricevitore (STM32F3) è stato configurato il pin PC10 per la trasmissione e il pin PC11 per la ricezione.

In entrambi i nodi sono state abilitate le interruzioni tramite NVIC ed è stato impostato con livello di priorità 2.

Come per la PIA, nella seguente tabella sono riportati i collegamenti.

STM32L4 Pin		Direzione	STM32F3 Pin	
PA0	DataTX	->	PC11	DataRX
PA1	DataRX	<-	PC10	DataTX

Figura 1.3.2.1 - Tabella Pin UART

2. Progetto Automatic Irrigation

2.1 Descrizione Generale

Il secondo esercizio proposto rappresenta un sistema di irrigazione automatico realizzato tramite due discovery board (anche in questo caso STM32F3 e la STM32L475), un sensore di rilevamento dell'umidità, un interruttore e una pompa automatica. L'architettura realizzata è la seguente:

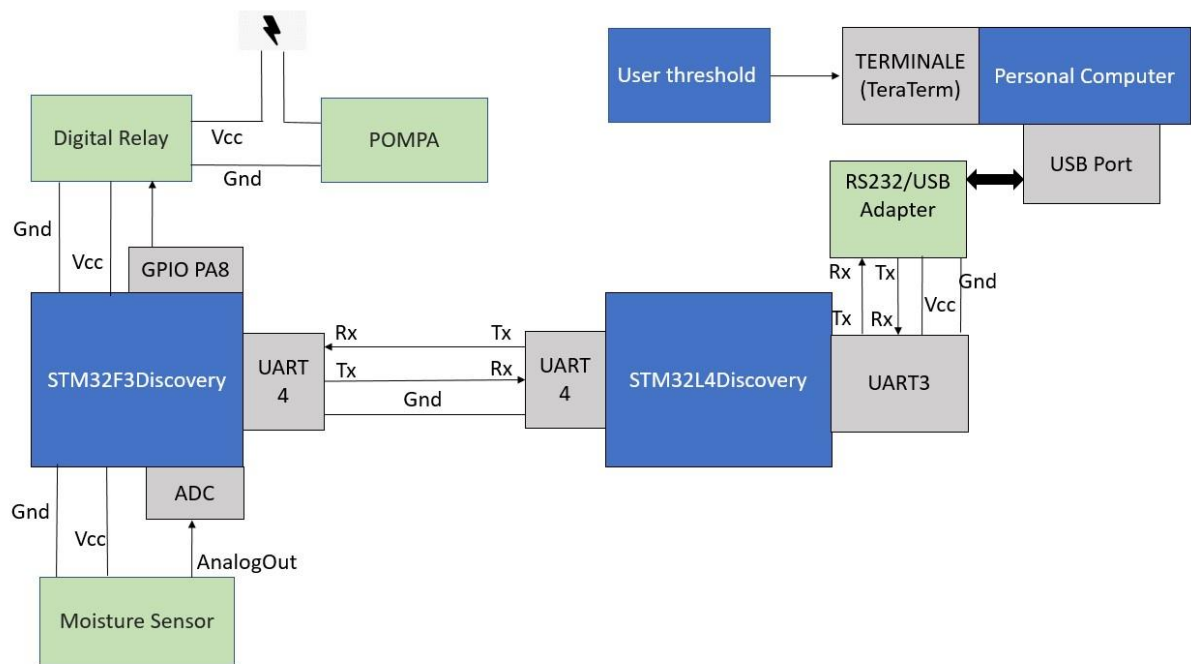


Figura 2.1.1 - Architettura Logica

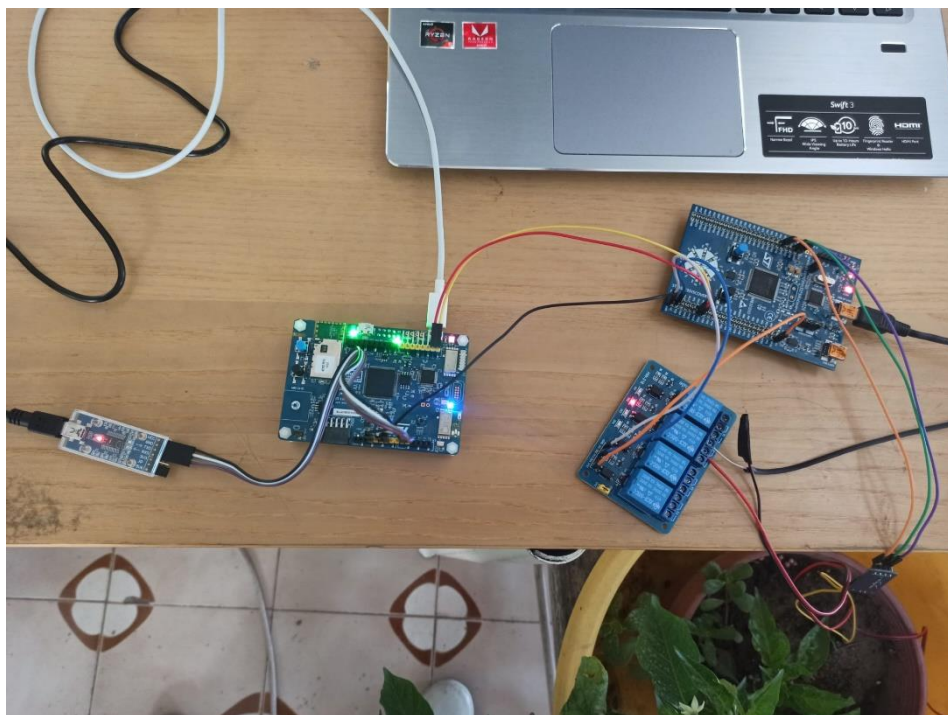


Figura 2.1.2 - Architettura Fisica

Il sistema funziona nel modo seguente: la board STM32L475 è collegata al terminale del PC mediante interfaccia seriale e da questa riceve un valore che rappresenta una soglia di umidità percentuale, la quale può variare tra 0 e 100. Il valore ottenuto equivale alla codifica ASCII del carattere ricevuto e qualsiasi valore superiore al 100 viene visto come la soglia massima. Il messaggio ricevuto, dopo questa prima elaborazione, viene inviato alla board STM32F3 tramite

UART in seguito alla pressione del PUSHBUTTON. La seconda board confronta il valore ricevuto sull'UART con quello ottenuto dal sensore di umidità sulla periferica ADC e, se l'umidità rilevata è inferiore alla soglia, attiva per un secondo l'interruttore al quale è collegata la pompa attraverso un Pin di GPIO. In questo modo la pompa viene azionata e inizia l'irrigazione.

2.2 Protocolli di comunicazione

I protocolli adoperati sono due comunicazioni seriali tramite UART (una per la comunicazione tra le due schede e una per la comunicazione tra la scheda STM32L4 e il PC), mentre per il prelievo dell'umidità dal sensore è stata utilizzato il convertitore analogico/digitale ADC.

Per scopi descrittivi è stato scelto di continuare la trattazione suddividendo i prossimi capitoli in base al nodo preso in considerazione così da suddividere il problema in moduli.

2.3 Board STM32F3

Nel seguito è descritto tutto ciò che riguarda il nodo STM32F3

2.3.1 Comunicazione seriale tramite UART

Come detto per la comunicazione tra i due nodi si è deciso di utilizzare la periferica seriale UART. Per quanto riguarda il nodo di interesse la periferica (UART4) è stata configurata nel modo classico come anche visto nell'esercizio precedente. Modalità asincrona, 8 bit per il dato senza bit di parità, 1 bit di stop e il baud rate di 115200 bit/sec.

Anche in questo caso è stata abilitata l'interruzione tramite NVIC. Viene avviata la periferica per ricevere il dato, quando questo arriva si scatena l'interrupt e il dato viene memorizzato, dunque la periferica viene messa di nuovo in "ascolto".

I pin utilizzati sono il pin PC10 per la trasmissione (anche se qui la trasmissione non avviene mai) e il pin PC11 per la ricezione.

Questa volta (a differenza del primo esercizio) la periferica UART ha come priorità di interruzione **1**, più prioritaria dell'interruzione dovuta all'ADC. Questo perché l'ADC riceve dati continuamente, se fosse più prioritario non darebbe la possibilità a UART di interrompere e completare la propria ISR. Inoltre, è anche logico pensare di dare maggiore priorità ad UART dal momento che viene prima ricevuta la soglia di umidità da rispettare e poi viene confrontata con il valore di umidità attuale del terreno.

2.3.2 Sensore di umidità e ADC

Per rilevare la percentuale di umidità è stato utilizzato un sensore apposito applicabile direttamente nel terreno mostrato in figura 2.3.2.1.

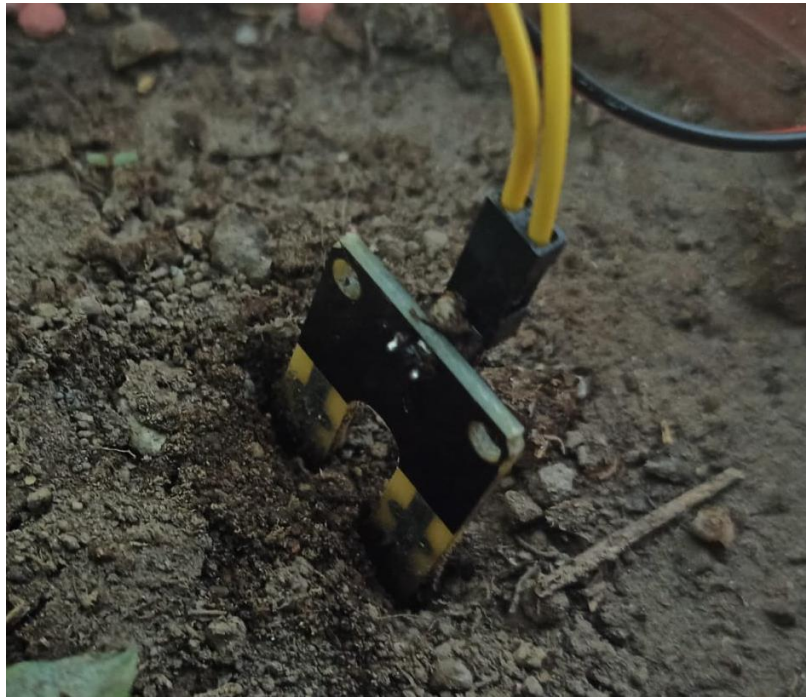


Figura 2.3.2.1 - Moisture Sensor

Il sensore è dotato di un'uscita analogica e una digitale, si è scelto di utilizzare l'analogica e di collegarla dunque al convertitore analogico/digitale presente a bordo della scheda.

Il convertitore è stato configurato in modo standard ovvero:

- Rappresentazione del dato convertito su 12 bit (risoluzione)
- Conversione single-ended (conversione rispetto alla tensione Vcc)
- Dato allineato a destra
- Conversione continua (occorre monitorare continuamente l'umidità)
- Interruzione abilitata su ogni conversione avvenuta
- Priorità di interruzione 2

È stato utilizzato il canale 2 dell'ADC1 (il canale 1 era in conflitto con altre periferiche). Il pin collegato è il PA1.

C'è da fare una precisazione sul dato ricevuto. Quello che viene ottenuto infatti è un codice (binario) relativo al valore di tensione rilevato dal convertitore e va dunque convertito in umidità percentuale.

```
return (uint8_t)100-(volt*((3.0/4096.0))*(100.0/3.0));
```

Figura 2.3.2.2 - Conversione Tensione/Umidità

Per effettuare questa conversione è stata utilizzata la formula in figura 2.3.2.2. Viene moltiplicato il codice ottenuto dal convertitore per il valore di tensione massima (3V) e diviso per il codice massimo (4096, perché stiamo lavorando con una risoluzione a 12 bit). Inoltre, questo valore di tensione (in volt) viene reso valore percentuale (rispetto ai 3V) moltiplicando per il fattore 100/3. Infine, è stato trovato empiricamente che il valore ottenuto è indice del grado di aridità del terreno e non di umidità per cui è bastato prendere il complemento rispetto a 100 del valore trovato (100-*).

2.3.2 Digital Relay e Pompa

Come attuatore per eseguire l'innaffiatura è stata utilizzata una pompa sommergibile la cui tensione di ingresso è di 12V, superiore ai 5V che al massimo eroga la board.

Per cui è stato utilizzato un interruttore digitale (relè) al quale è collegata la pompa da un lato e dall'altro il pin PA8 della board (in output). Così facendo per attivare la pompa è bastato scrivere opportunamente il registro di quel pin.

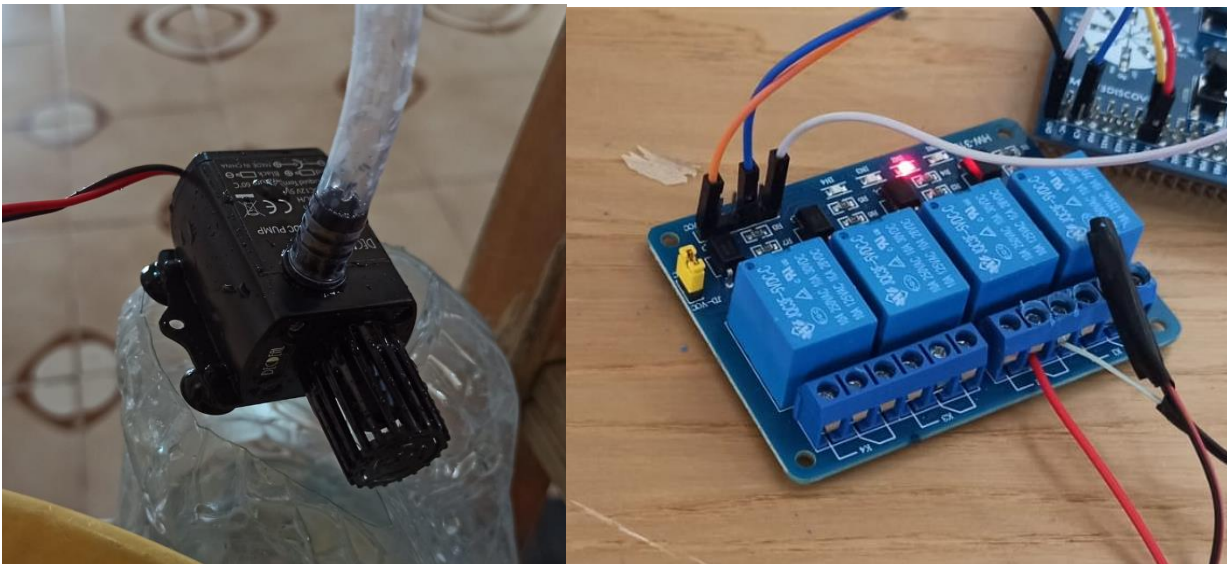


Figura 2.3.2.1 - A destra i relè, a sinistra la pompa

2.4 Board STM32L4

Nel seguito è descritto tutto ciò che riguarda il nodo STM32L4

2.4.1 Comunicazione seriale tramite UART con altra board

La periferica UART ha la stessa configurazione vista per l'altra board.

I pin utilizzati sono il pin PA0 per la trasmissione e il pin PA1 per la ricezione. In questo caso la trasmissione non avviene sotto interrupt bensì avviene in modo bloccante alla pressione del pulsante PC13 (il quale opera sotto interrupt). La priorità di interruzione è **1**. La periferica UART adoperata è la UART4.

2.4.2 Comunicazione seriale tramite UART con PC

L'ultimo problema da affrontare era trovare un modo per permettere all'utente di inserire una soglia di umidità desiderata dinamicamente.

Ovviamente per fare ciò esistono molteplici soluzioni, ad esempio sarebbe stato possibile connettere sempre tramite UART un'altra scheda (Raspberry Pi) dotata di modulo WiFi (o anche Bluetooth) in modo da poter far accedere tramite la rete l'utente.

Il modo più semplice e veloce però è quello di collegare la scheda direttamente ad un Personal Computer (sempre tramite comunicazione seriale) e di inserire la soglia dal PC stesso.

Dunque, è stata configurata un'altra periferica seriale sull'STM32L4 (UART3) collegata poi ad un adattatore RS232/USB per poter collegare il tutto alla porta USB del PC.

La configurazione della periferica è la medesima menzionata fino ad ora, il livello di priorità è **2**. I pin utilizzati sono il pin PC4 per la trasmissione e il pin PC5 per la ricezione.

Infine, sul PC è stato utilizzato il programma di emulatore di terminale "TeraTerm" di modo da connetterci alla linea di comunicazione. A questo punto l'utente può digitare un carattere sulla tastiera il cui valore esadecimale ASCII corrispondente sarà il valore di soglia inviato all'STM32L4.

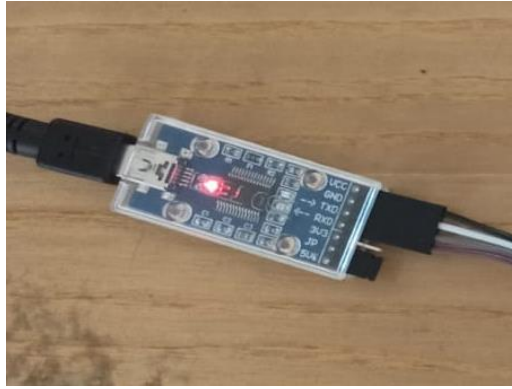


Figura 2.4.2.1 - Adattatore RS232/USB

3. Riferimenti

Di seguito sono riportati rispettivamente i link alla repository GitHub prodotta e ad un video dimostrativo dell'esecuzione del secondo progetto.

<https://github.com/DOTTORM5/CSD-project.git>

https://drive.google.com/file/d/1xWDZbisT1xNbUwKjPxWCwKAhKr_T-vbS/view?usp=sharing