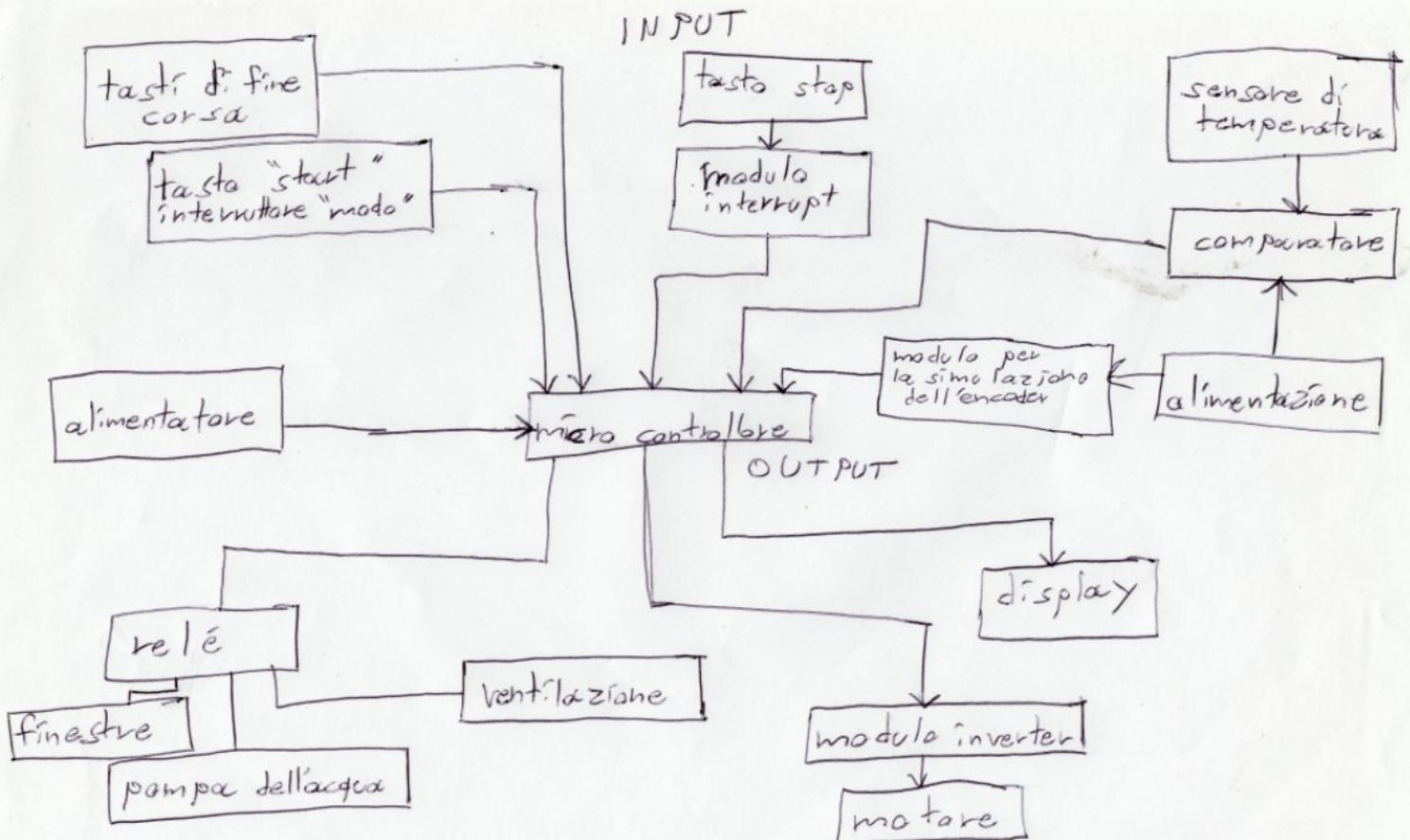


Schema a blocchi con descrizione:



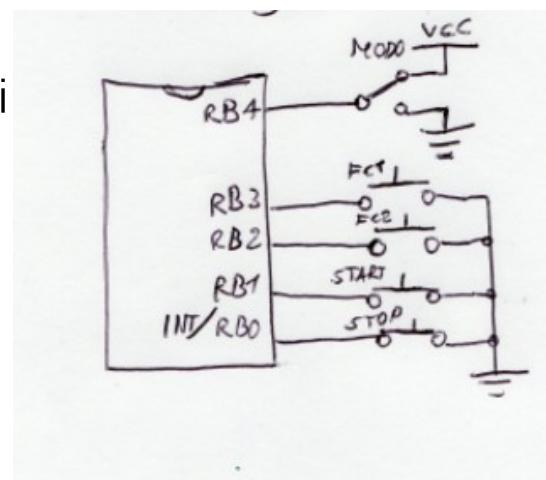
INPUT

Tasti di fine e inizio corsa

I fine corsavengono rappresentati da dei pulsanti collegati a massa e al registro B del PIC per sfruttare le resistenze di PULL-UP interne.

```
#define FC1 RB3
```

```
#define FC2 RB2
```



Tasti di “Start” “Stop”

Il tasto di “start” viene collegato come i tasti di finecorsa, ovvero al registro B.

Il tasto di “stop” viene collegato al pin “INT” per sfruttare l’interrupt, così che il tasto di stop può interrompere il programma principale.

```
#define T_START RB1  
#define MODO RB4
```

Interruttore di MODO

Per l’interruttore di modo viene usato un deviatore collegato a vcc e gnd così da poter avere un livello logico rispettivamente alto o basso. Sarebbe preferibile mettere dei diodi per evitare un corto quando si muove l’interruttore.

```
#define MODO RB4
```

Modulo per la gestione di temperatura

per gestire la temperatura viene usato un comparatore a finestra fatto con due amplificatori operazionali per gestire su due pin di uscita le due soglie, quella superiore e quella inferiore.

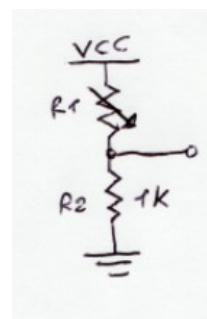
Calcoli:

Sapendo che: $R(T) = 10^3 + 10T$

Calcoliamo la resistenza nelle due temperature:

$$R(30^\circ C) = 10^3 + 10 \cdot 30 = 1300 \Omega$$

$$R(27^\circ C) = 10^3 + 10 \cdot 27 = 1270 \Omega$$



Immaginando che $R_2 = 1\text{k}\Omega$ calcoliamo le tensioni di fascia:

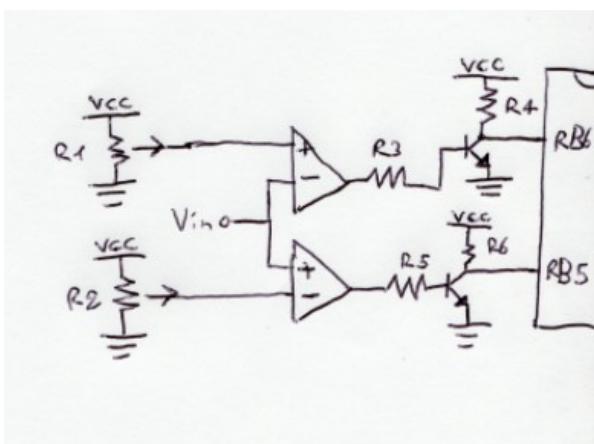
$$V_{sup}(27^\circ C) = \frac{R_2}{R_1 - R_2} \cdot 5 = \frac{1k}{1,27k - 1k} \cdot 5 = 2,2V$$

$$V_{inf}(30^\circ C) = \frac{R_2}{R_1 - R_2} \cdot 5 = \frac{1k}{1,3k - 1k} \cdot 5 = 2,17V$$

Per raggiungere le tensioni in modo preciso usiamo due trimmer.

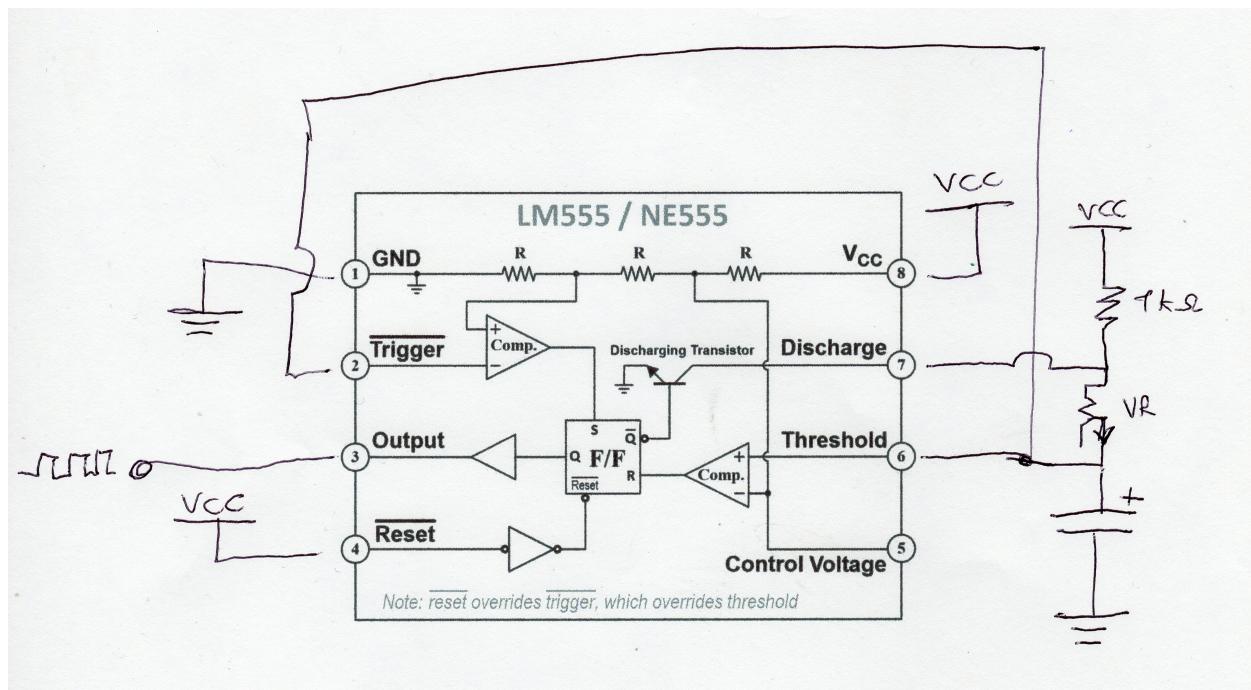
Mentre per simulare l'ingresso come tensione variabile viene usato un potenziometro.

```
#define V_SUP RB6
#define V_INF RB5
```



Modulo per la simulazione dell'encoder

Per simulare l'uscita a onda quadra con frequenza variabile dell'encoder usiamo un generatore di impulsi a frequenza modulabile usando un NE555.



$$R1=1k\Omega$$

$$R2=470k\Omega$$

$$C=1\mu F$$

Sapendo che la formula per trovare la frequenza è:

$$f = \frac{1,44}{(R1+2 \cdot VR) \cdot C}$$

Troviamo le frequenze con i componenti:

Quando il potenziometro vale 0Ω :

$$f = \frac{1,44}{(1k+2 \cdot 0) \cdot 1\mu F} = 1,44 \text{ kHz}$$

Quando il potenziometro vale $470k\Omega$:

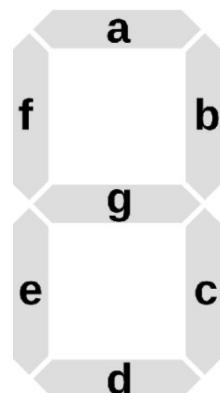
$$f = \frac{1,44}{(1k+2 \cdot 470k) \cdot 1\mu F} = 1,53 \text{ Hz}$$

```
#define
#define
```

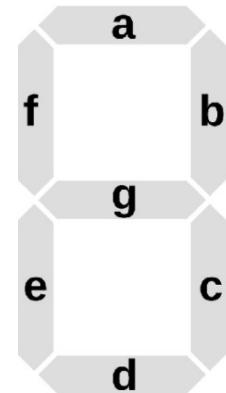
OUTPUT

Gestione Display

Per gestire i display si usano i registri C e A e per il circuito viene usata solo una resistenza da $1k\Omega$ per una questione di tempo, normalmente si dovrebbe usare una resistenza per ogni led di un singolo segmento.



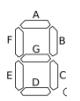
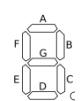
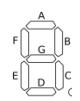
DISPLAY1



DISPLAY2

Collegamento Display1:

RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
	g	f	e	d	c	b	a

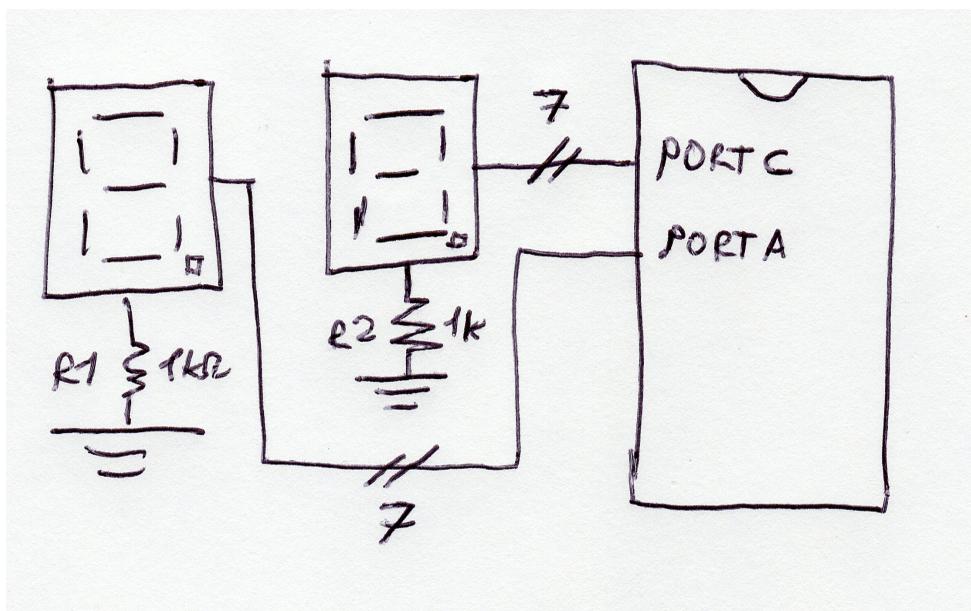
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0		
0	0	1	1	1	1	1	1	0	
0	1	1	1	0	0	0	1	F	
0	1	1	1	1	0	0	1	E	

Collegamento Display2:

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
g	f	e	d	c	b	a	g

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0		
0	1	1	1	0	0	0	1	F	
0	1	0	1	0	1	0	0	n	
0	1	0	1	1	0	0	0	c	

La parte di visualizzazione viene gestita nel programma tramite una funzione che assegna ai registri i valori secondo la tabella per visualizzare una determinata cifra.



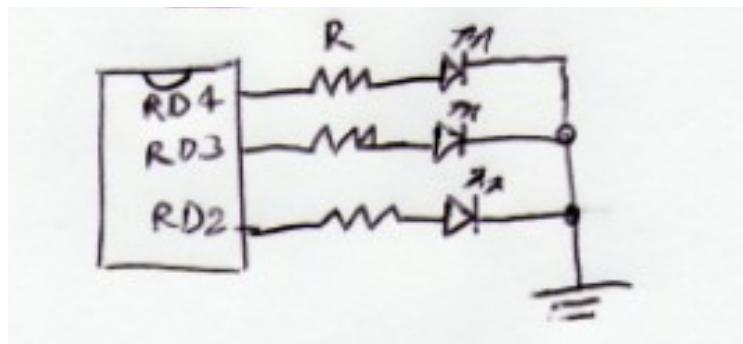
Gestione del modulo inverter per il motore

Per gestire il motore si usa un modulo inverter idealizzato che viene controllato da 3 bit forniti dal PIC

bit	Livello logico	Azione
	0	Start
1	1	Stop
	0	Dietro
2	1	Avanti
	0	Minima velocità
3	1	Massima velocità

```
#define POWER_MOT RD4
#define VEL_MOT RD3
#define DIR_MOT RD2
```

```
#define START 1
#define STOP 0
#define AVANTI 1
#define DIETRO 0
#define LENTO 0
#define VELOCE 1
```



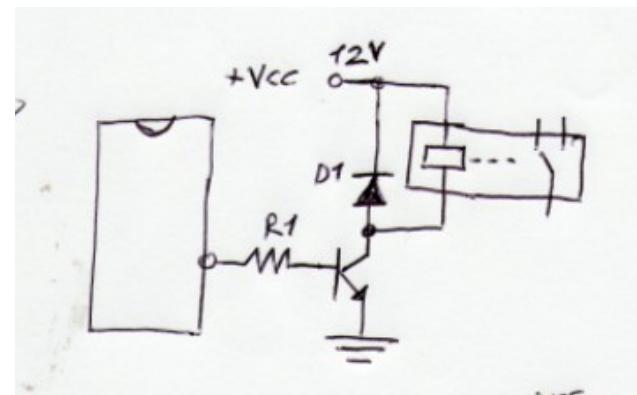
La parte di irrigazione viene gestita nel programma tramite una funzione che accetta 4 comandi, tre per il motore uno per la gestione del relè dell'acqua.

Gestione dei relè

Per gestire i relè si usa PORTD; nel circuito reale sono simulati come led, nel caso serva usare un relè reale bisogna far uso di un transistor come interruttore che riesce a gestire le potenze dei relè senza farle passare per il microcontrollore

```
#define RELE_ACQUA RD7
#define FINESTRE RD5

#define ECCITATO 1
#define DISECCITATO 0
```

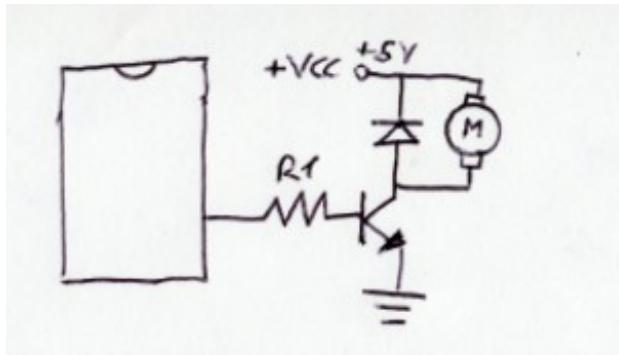


il relè dell'acqua viene gestito nella funzione di irrigazione che può essere quindi eccitato o disecxitato

relè	Livello logico	Azione
	0	chiuse
Finestre	1	aperte
	0	spenta
Pompa	1	accesa
	0	spenta
ventilazione	1	accesa

Gestione del motorino dc per simulare i ventilatori

Per gestire il motorino si usa PORTD; bisogna far uso di un transistor come interruttore che riesce a gestire la potenza del motore senza farle passare per il microcontrollore



```
#define VENTILATORI RD6
```

lo schema del programma sta in [schema-programma.pdf](#)