

CORSO DI ORDINAMENTO

Indirizzo: TECNICO DELLE INDUSTRIE ELETTRONICHE

Tema di: SISTEMI, AUTOMAZIONE E ORGANIZZAZIONE DELLA PRODUZIONE

Tema n. 2

Si vuole realizzare un sistema di controllo automatico a microcontrollore o a microprocessore per l'irrigazione del parco di una villa di campagna . Quest'ultimo è suddiviso in quattro zone che debbono essere annaffiate in ore e con tempi diversi secondo la seguente tabella:

Zona	Ora	Tempo in minuti
1	20.00	20'
2	21.00	30'
3	22.00	30'
4	23.00	40'

La durata dei tempi dell'irrigazione è anche legata alla temperatura media T_m della giornata, monitorata ogni ora tra le ore 7.00 e 19.00, come indicato nella tabella sottostante:

$T_m \leq 15^\circ\text{C}$	Il sistema non si avvia
$15^\circ\text{C} < T_m \leq 25^\circ\text{C}$	1 tempi sono quelli indicati in tabella
$25^\circ\text{C} < T_m \leq 30^\circ\text{C}$	1 tempi debbono essere aumentati del 30%.
$T_m > 30^\circ\text{C}$	1 tempi debbono essere aumentati del 50%.

Il sistema di irrigazione, che è azionato da una pompa ad immersione sempre sotto tensione, posizionata sul fondo di un pozzo di 30 m, invia l'acqua alle zone tramite l'apertura e la chiusura programmata di 4 valvole; se il livello dell'acqua si abbassa fino a raggiungere il livello di 2 m rispetto al fondo, il sistema d'irrigazione si deve arrestare. Tale condizione viene segnalata con l'invio di un segnale digitale proveniente da un sensore di livello. Per acquisire la temperatura si utilizza un sensore che dà in uscita una tensione proporzionale alla temperatura assoluta $V = KT$ con $K = 10 \text{ mV}/^\circ\text{K}$.

Il candidato, formulate le ipotesi aggiuntive che ritiene opportune, e scelto un dispositivo programmabile di sua conoscenza:

- 1 . Descriva tramite schema a blocchi la struttura del controllo.
2. Illustri la funzione dei singoli blocchi.
3. Determini le caratteristiche di ciascun blocco in funzione dei segnali elettrici di ingresso e di uscita.
4. Disegni il flow-chart del programma di gestione.
5. Traduca un segmento del programma in un linguaggio di sua conoscenza.

Durata massima della prova: 6 ore.

È consentito soltanto l'uso di manuali tecnici e di calcolatrici tascabili non programmabili. Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

POSSIBILE SOLUZIONE

Intendo utilizzare il microcontrollore ST62T65B capace di gestire al suo interno un orologio, in grado di acquisire il valore della temperatura attraverso un INPUT ADC ed equipaggiato di una comunicazione SPI (Serial Peripheral Interface) per la comunicazione digitale con il sensore di livello.

Ipotesizzo pertanto che il sensore di livello con uscita digitale possa comunicare su linea a due fili secondo un protocollo SPI da implementare con una adeguata routines.

Leggendo il testo del tema emerge che la pompa è sempre sotto tensione in contrasto con il fatto che l'irrigazione si debba fermare quando il sensore rilevi un livello dell'acqua minore o uguale a 2 metri.

Pertanto prevedo che anche la pompa sia comandata dal dispositivo e accesa quando si richiede l'irrigazione e spenta quando non deve irrigare, rispettando le specifiche imposte dal tema.

Il tema non specifica le caratteristiche delle valvole pertanto ritengo siano equipaggiate da un motore funzionante a 24Vcc e che in un tempo di 10 secondi apra o chiuda in base alle polarità della tensione di alimentazione.

Prevedo anche che il microcontrollore gestisca un tastierino a 12 tasti per l'impostazione dei parametri per l'orologio e per i tempi di intervento e durata dell'irrigazione qualora si debbano cambiare.

Per quanto concerne le temperature da rilevare reputo queste comprese in un range tra 0 °C minimo e un massimo di 50 °C visto che la media delle temperature da considerare vanno da 15 a 30 °C.

1 - SCHEMA A BLOCCHI

Il disegno dello schema a blocchi qui sotto riportato mette in evidenza come sarà gestito l'impianto descritto nel tema.

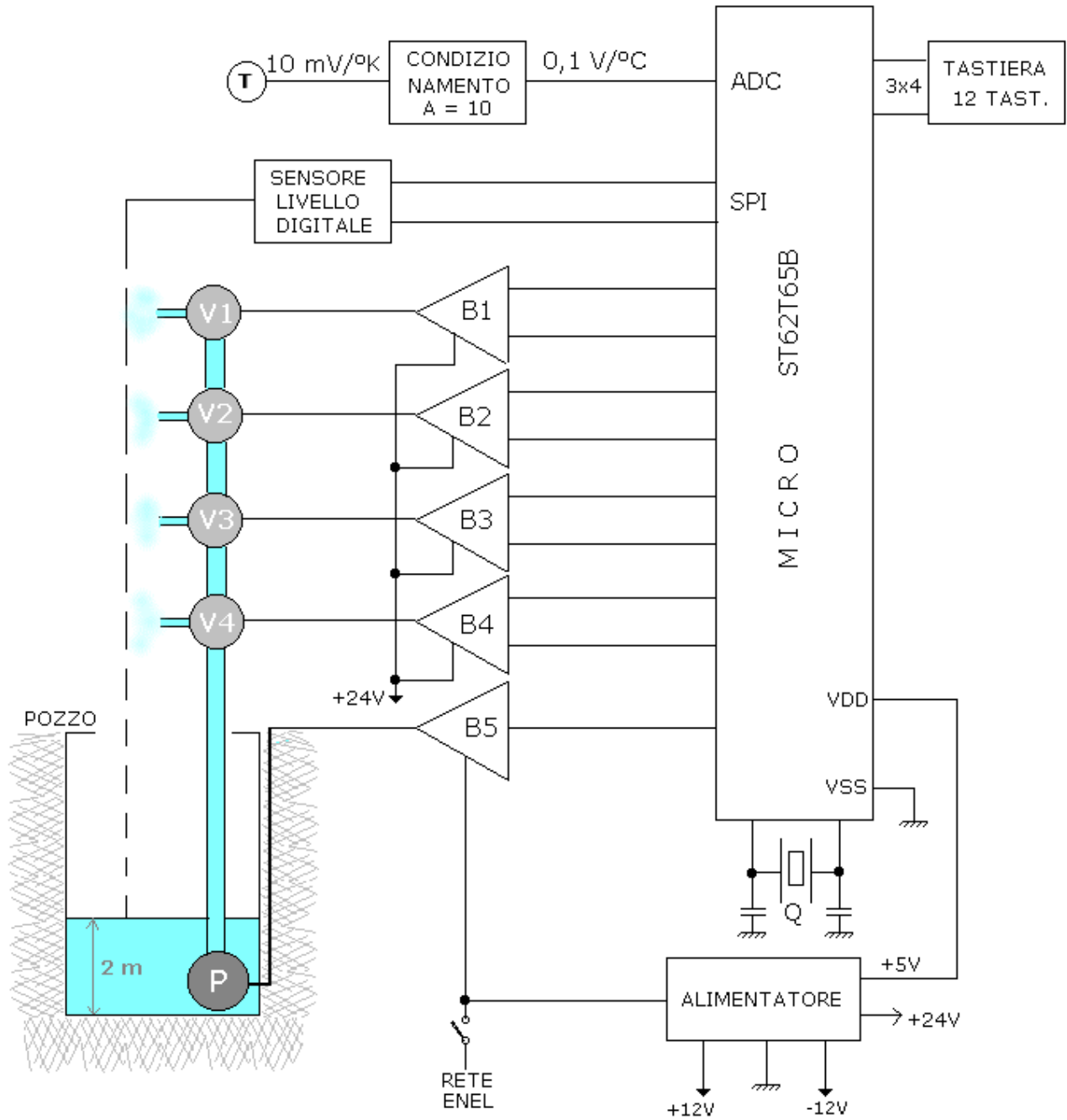
Partendo dal sensore di temperatura si arriva al blocco di condizionamento per la conversione da gradi Kelvin a gradi Centigradi e amplificazione del segnale affinché all'ADC arrivi 0,1 V/°C.

Le 4 valvole di zona indicate con V1,2,3,4 sono comandate dai blocchi B1,2,3,4 contenenti i buffer che adattano i segnali del micro con quelli di potenza per i motori a 24V.

Il blocco B5 riguarda il buffer di potenza per il comando della pompa immersa.

Il micro è collegato anche al blocco della tastiera e in fine abbiamo il blocco per le alimentazioni del +5V per il micro, ±12V per gli operazionali e i +24V per i motori delle valvole di zona.

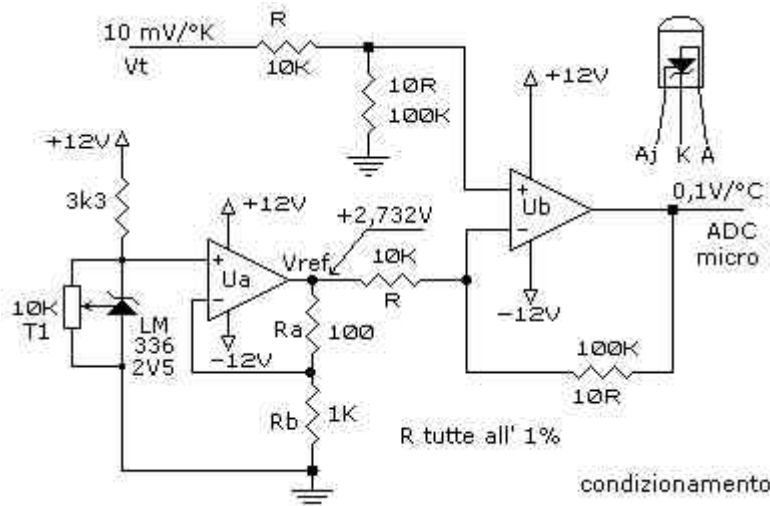
Il blocco del sensore digitale di livello rileva l'altezza dell'acqua e sarà connesso al micro con una linea bifilare dove su un filo viaggeranno i dati seriali e sull'altro gli impulsi di sincronismo.



2-3 - DESCRIZIONE BLOCCHI E FUNZIONAMENTO CIRCUITI

CIRCUITO DI CONDIZIONAMENTO

Siccome il sensore fornisce $10 \text{ mV}/^\circ\text{K}$ e considerato che 0°C corrispondono a $273,2$ gradi Kelvin significa che a 0°C all'uscita del sensore saranno presenti $2,732\text{V}$. Per avere un segnale di $0,1 \text{ V}/^\circ\text{C}$, all'uscita del blocco di condizionamento, è necessario realizzare un circuito amplificatore differenziale così da poter sottrarre una tensione di riferimento V_{ref} di $2,732\text{V}$ e amplificare con un guadagno di 10. Il circuito che intendo adottare è il seguente:



FUNZIONAMENTO

L'integrato LM336-2V5 (super zener con soglia regolabile) fornisce nominalmente una tensione stabile di $2,5\text{V}$ regolabile entro un range di $\pm 0,12\text{V}$ attraverso T1 da 10K . Questa tensione si presenta sul non invertente dell'operazionale configurato come amplificatore con guadagno $A=1 + R_a/R_b$.

In uscita pertanto avremo $2,5 + 2,5 \times R_a/R_b$ essendo $R_a = 100 \text{ Ohm}$ e $R_b = 1\text{K}$ sarà $V_{\text{ref}} = 2,5\text{V} + 0,25\text{V} = 2,75\text{V}$ superiore a quanto noi volgiamo ma facilmente regolabile con il trimmer T1. Basterà regolare T1 per avere in ingresso $2,484\text{V}$ per avere $V_{\text{ref}} = 2,484\text{V} + 0,2484\text{V} = 2,7324 \text{ V}$ come richiesto.

Il micro dispone di un ADC (Analogic Digital Converter) a 8 bit con range di tensione d'ingresso compreso tra 0V e 5 V pertanto con un segnale di $0,1 \text{ V}/^\circ\text{C}$ copro comodamente l'escursione delle temperature da rilevare comprese tra 0°C e 50°C ; così dopo la conversione a 8 bit si avranno numeri compresi tra 0 e 255.

Come già anticipato l'amplificatore differenziale dovrà amplificare 10 volte il segnale differenza $V_t - V_{\text{ref}}$. Per comprendere il funzionamento dell'amplificatore applichiamo la sovrapposizione degli effetti e vediamo come risponde considerando prima nullo il segnale V_t e poi viceversa V_{ref} .

$V_{u1} = -10R/R \times V_{\text{ref}} = -10V_{\text{ref}}$ secondo caso $V^+ = V_t \times 10R/10R+R = V_t \times 10/11$ ma

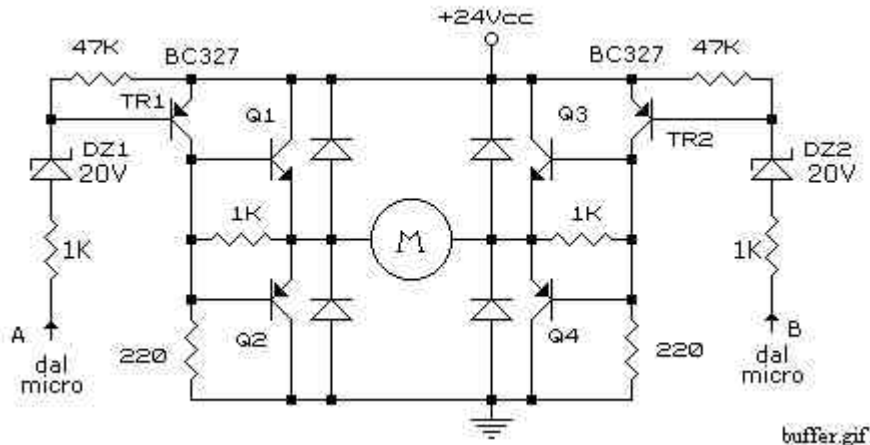
$V_{u2} = V^+ \times (1 + 10R/R) = V^+ \times 11$ sostituendo $V_{u2} = V_t \times 10/11 \times 11 = 10 V_t$

Sommando i due effetti sarà $V_u = V_{u1} + V_{u2} = -10V_{\text{ref}} + 10 V_t = \mathbf{10 (V_t - V_{\text{ref}})}$ come previsto.

BUFFER PER IL COMANDO MOTORE VALVOLE

Le uscite dell'ST6 possono essere di tipo alto = 5V o basso = 0V ["1" o "0"] oppure ad alta impedenza cioè ininfluenti sul circuito (configurazione output Tree State). In questo caso intendo configurare il micro con uscite di tipo soltanto alto / basso.

Avendo ipotizzato che le valvole sono equipaggiate con motori in continua a 24V si dovrà interporre tra micro e valvole una opportuna interfaccia per l'adattamento dei segnali sia per quanto riguarda i valori tensione/corrente sia per quanto concerne l'inversione di polarità per avere in un caso l'apertura e nell'altro la chiusura della valvola. Utilizzando due uscite dal micro per ogni valvola il circuito buffer diventa:



FUNZIONAMENTO

A motore spento (valvola ferma aperta o chiusa) non dovrà condurre né Q1 né Q3 e di conseguenza interdetti pure TR1 né TR2. Per avere questa condizione i livelli di tensione provenienti dal micro dovranno essere alti a 5V in maniera da avere i diodi zener DZ1,2 da 20V interdetti e pertanto V_{be1} e V_{be2} nulle.

Quando una delle due uscite del micro va a livello basso (0V) abilita o TR1 o TR2. Supponiamo che il punto A sia basso e il punto B alto; va in saturazione TR1 e Q1 mentre sono interdetti TR2 e Q3; Q4 PNP ha la base a massa con la 220 Ohm pertanto pure lui in conduzione. Il motore viene alimentato con il + sul lato sinistro e il meno su quello destro e girerà nel modo che potremmo attribuire all'APERTURA.

Viceversa con il punto B basso e il punto A alto va in saturazione TR2 e Q3 mentre sono interdetti TR1 e Q1; Q2 PNP ha la base a massa con la 220 Ohm pertanto pure lui in conduzione. Il motore viene alimentato con il meno sul lato sinistro e il + su quello destro e girerà in un modo contrario a prima che potremmo attribuire alla CHIUSURA.

I diodi servono a smorzare ed annullare le extra tensioni inverse che nascono ai capi del motore al momento dell'interruzione della sua alimentazione; dovranno essere scelti con velocità di commutazione veloce e di corrente anche solo di 1A.

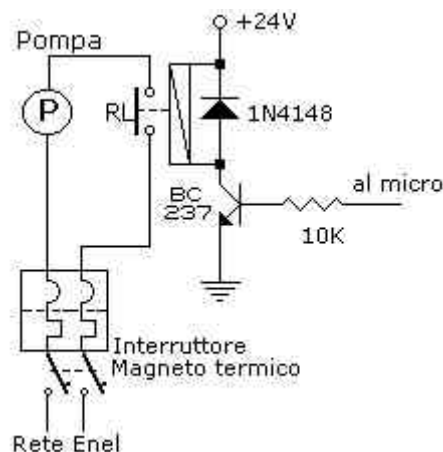
Tabella riassuntiva delle condizioni:

A	B	MOTORE	VALVOLA
1	1	OFF	indifferente
0	1	ON	APRE
1	0	ON	CHIUDE
0	0	NON AMMES.	NON AMMES.

Il motore di queste valvole di piccola e media portata in genere assorbe correnti dell'ordine di 100/200 mA e devono girare per compiere la totale apertura o chiusura una decina di secondi. Considerato il consumo relativamente basso i transistori Q1,2,3,4 possono essere dei normali BDxxx (PNP e NPN) con un guadagno β possibilmente $>$ di 50 così da poter garantire una buona saturazione.

BUFFER PER IL COMANDO POMPA

La pompa immersa funzionerà a tensione di rete Enel e un relè a 24V comandato da un transistore pilotato dal micro interrompe o chiude il circuito di alimentazione di linea della pompa.



In serie alla linea è bene prevedere ai fini protettivi un interruttore magneto-termico capace di interrompere energia verso la pompa qualora ci siano assorbimenti anomali.

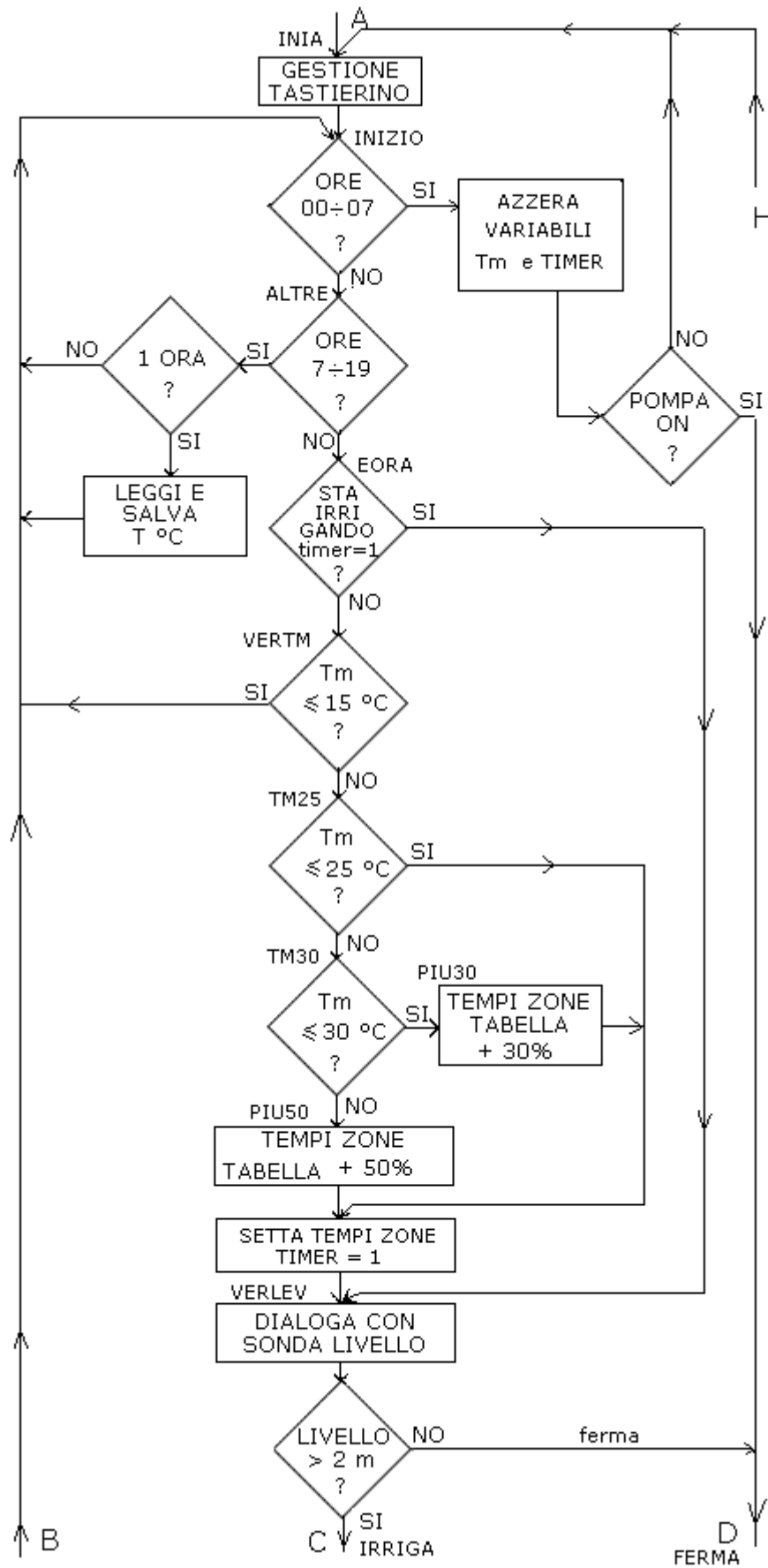
Per far funzionare la pompa il micro dovrà mandare alto l'uscita che polarizza il transistore così da eccitare il relè che chiudendo i suoi contatti fornirà energia al motore.

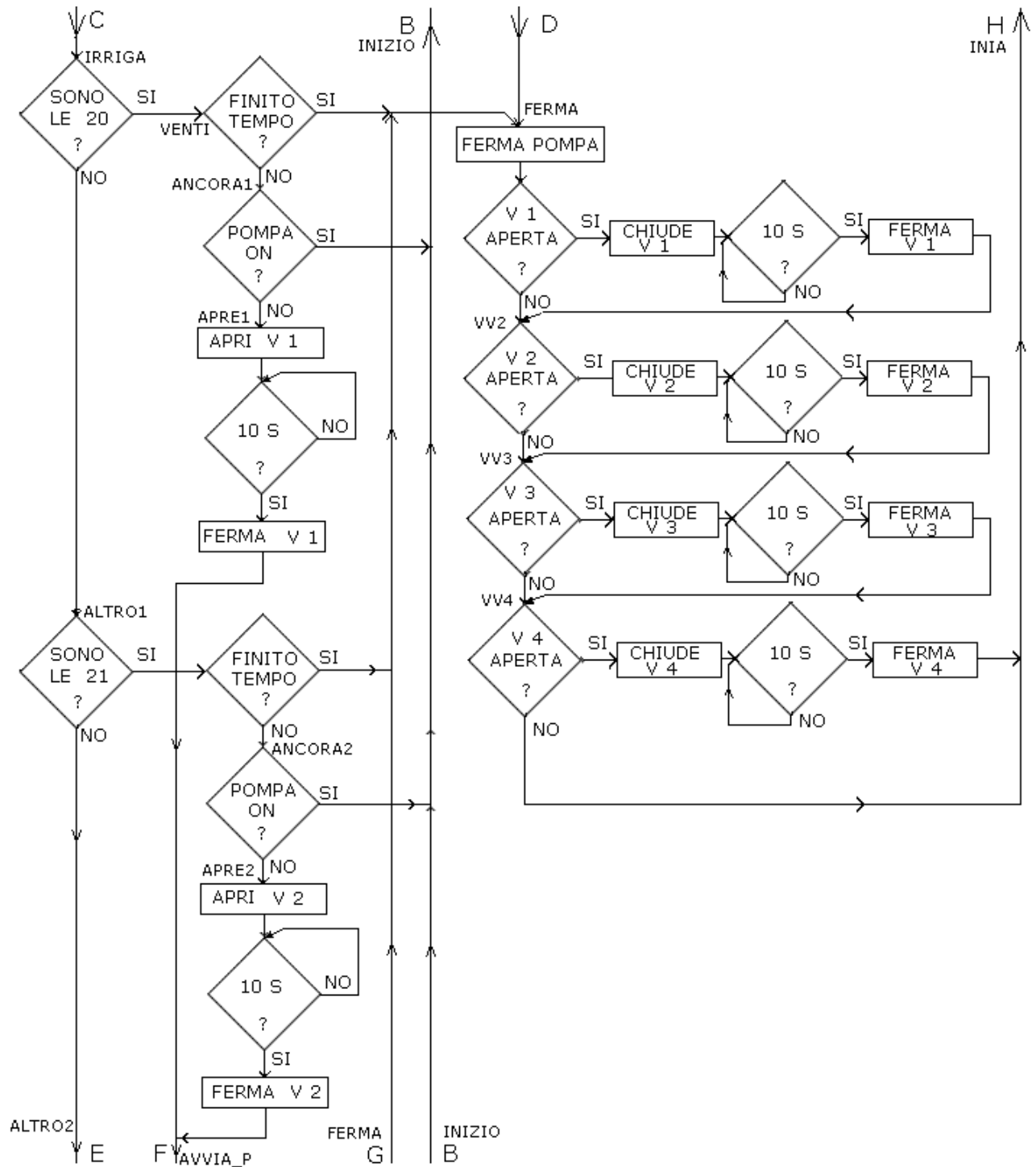
In parallelo alla bobina del relè ho previsto un diodo per lo smorzamento delle extra tensioni generate all'apertura del transistore.

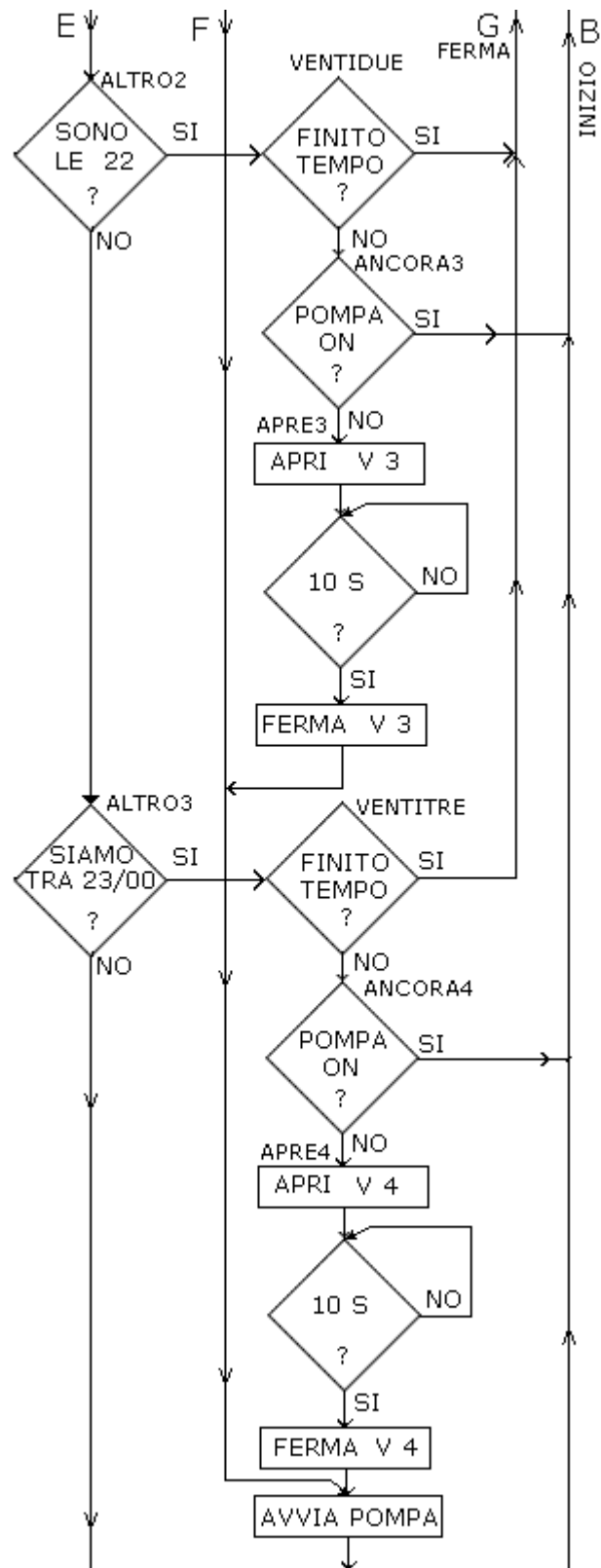
ALIMENTATORE

Per i 24V si utilizza un trasformatore da circa 10VA con secondario a 18V ac che raddrizzati e livellati con un elettrolitico da 2200 uF meno la caduta dei diodi del ponte si ottiene appunto 24Vcc. Per il \pm 12V si utilizza un trasformatore da 5 o 6 VA con doppio avvolgimento a 12Vac che raddrizzati e livellati danno circa 16Vcc e che stabilizzati da un regolatore 7812 e un 7912 danno quanto necessario. Partendo dal +16V con un regolatore 7805 si ottiene il + 5V per il micro.

4 – FLOW-CHART







5 – PROGRAMMAZIONE IN ASSEMBLER ST62XX

Considero qui solo le routines principali ma è ovvio che ci sono anche quelle relative alla definizione delle variabili, all'orologio, alla tastiera, alla comunicazione SPI.

Punto A del flow-chart

```
INIA      ldi    piu1,8      ; valore nella variabile usata per ogni ora
          call   TASTI     ; va a gestire la tastiera qualora ci siano richieste
INIZIO    ld     a,ore      ; legge ore dall'orologio
          jrnz   ALTRE     ; se non sono le ore 00 salta va ad ALTRE
          clr    tm        ; azzerava variabile dove tiene temperatura media
          clr    timer     ; azzerava la variabile timer =1 se irriga = 0 no
          clr    restT     ; prepara azzerate le variabili restT
          clr    restM     ; e restM
          jrs    0,port_b,FERMA ; se il bit 0 è 1 = pompa ON va a fermare
          jp     INIA      ; altrimenti va ad INIZIO
ALTRE     cpi    a,7        ; verifica se le ore sono < delle 7
          jrc    INIZIO    ; se si salta ad INIZIO
          cpi    a,19      ; se > 19 (= > ore 20 )
          jrnc   EORA      ; salta a controllare dalle 20 alle 23
          cp     a,piu1    ; verifica se passato 1 ora (prima volta alle 8)
          jrnz   JINI      ; non ancora passata 1 ora e va a JINI
          inc    piu1      ; è passata un ora predisporre piu1 per la prossima
          call   ADC       ; ogni ora legge temperatura con ADC e calcola Tm
JINI      jp     INIZIO
EORA      ld     a,timer    ; verifica se sta temporizzando per l'irrigazione
          jrz    VERTM     ; se 0 non temporizza e salta
VERLEV    call   LIVELLO   ; comunica con sonda attraverso SPI
          ld     a,level    ; temporizza e sta irrigando e controlla livello
          cpi    a,2        ; verifica se > di 2 metri
          jrz    JPFERM    ; se è 2 m va a fermare
          jrnc   JIRRI     ; > a 2 metri salta
JPFERM    jp     FERMA     ; livello <= 2m va e ferma pompa
JIRRI     jp     IRRIGA    ; va a controllare tra le ore 20 e 23
VERTM     ld     a,tm      ; prende media temperatura
          cpi    a,15      ; verifica se Tm <= di 15
          jrz    JPIN      ; se = 15 salta e ricomincia dall'inizio
          jrnc   TM25      ; se > 15 salta a verificare se è 25 o 30 °C
JPIN      jp     INIZIO    ; se <= 15 non fa nulla e ritorna a INIZIO
TM25     cpi    a,25      ; verifica se Tm <= 25
          jrz    NORMAL   ; se Tm = 25 salta e mette valori standard
          jrnc   TM30      ; salta a verificare se >30
NORMAL    ld     a,t1      ; gestisce settaggio
          ld     tz1,a      ; tz1-tz2-tz3-tz4 tempi irrigazione = t1,t2,t3,t4
          ld     a,t2      ; con tempi standard
          ld     tz2,a
          ld     tz3,a
          ld     a,t4
          ld     tz4,a
          jp     TIME      ; va a verifica livello
TM30     cpi    a,30      ; se Tm >= di 30 salta
          jrz    CALPIU    ; se = 30 salta per call piu30
          jrnc   P50       ; se > 30 va +50%
CALPIU    call   PIU30     ; aggiunge + 30% ai valori standad di temporiz.
          jp     TIME      ; va a verifica livello
P50      call   PIU50     ; aggiunge + 50% ai valori standad di temporiz.
TIME     ldi    timer,1   ; mette a 1 per ricordare che è tempo di irrigazione
          jp     VERLEV    ; va a verifica livello
```

I valori standard della tabella dei tempi di irrigazione verranno messi in 4 variabili chiamate T1=20, t2=30, t3=30, t4=40 dove le seguenti routines attingeranno per poi settare tz1, tz2, tz3 e tz4 in funzione della temperatura media tm.

Minuti standard	+ 30 %	+ 50 %
20	26	30
30	39	45
40	52	60

```

PIU30  ld  a,t1      ; prende standard t1
       ldi  x,2     ; 2 = 30% vedi add sotto
       call TROVA   ; trova %
       ld  a,t1     ; prende standard t1
       add a,w      ; somma %
       ld  tz1,a    ; setta tz1 con nuovo valore per tempo irrigazione zona 1
       ld  a,t2     ; prende secondo standard = al terzo
       ldi  x,2     ; 2 = 30% vedi add sotto
       call TROVA   ; trova %
       ld  a,t2     ; prende standard t2
       add a,w      ; somma %
       ld  tz2,a    ; salva in tz2 per il tempo di irrigazione zona 2
       ld  tz3,a    ; salva in tz3 per il tempo di irrigazione zona 3
       ld  a,t4     ; prende standard t4
       ldi  x,2     ; 2 = 30% vedi add sotto
       call TROVA   ; trova %
       ld  a,t4     ; prende standard t4
       add a,w      ; somma % trova nuovo valore per irrigare
       ld  tz4,a    ; salva in tz4 per il tempo di irrigazione zona 4
       ret          ; ritorna da dove chiamata

PIU50  ld  a,t1     ; prende standard t1
       ldi  x,4     ; 4 = 50%
       call TROVA   ; trova %
       ld  a,t1     ; prende standard t1
       add a,w      ; somma % trova nuovo valore per irrigare
       ld  tz1,a    ; setta tz1 con nuovo valore per tempo irrigazione zona 1
       ld  a,t2     ; prende secondo standard = al terzo
       ldi  x,4     ; 4 = 50%
       call TROVA   ; trova %
       ld  a,t2     ; prende standard t2
       add a,w      ; somma % trova nuovo valore per irrigare
       ld  tz2,a    ; salva in tz2 per il tempo di irrigazione zona 2
       ld  tz3,a    ; salva in tz3 per il tempo di irrigazione zona 3
       ld  a,t4     ; prende standard t4
       ldi  x,4     ; 4 = 50%
       call TROVA   ; trova %
       ld  a,t4     ; prende standard t4
       add a,w      ; somma % trova nuovo valore per irrigare
       ld  tz4,a    ; salva in tz4 per il tempo di irrigazione zona 4
       ret

TROVA  ldi  y,10    ; prepara variabile y , va a dividere x 10
       call DIVIDI

ADDA   add  a,w     ; a+a = a x 2 perciò x arriva con 1 in meno
       dec  x       ; valore per il calcolo della %
       jrnz ADDA   ; se x = 0 in a = risultato
       ld  w,a     ; lo salva in w per poi sommarlo ai vari t1,2,3,4
       ret          ; al ritorno

```

Punto C del flow-chart:

arriva qui da quando riconosce un orario di irrigazione compreso tra 20 e 23;

IRRIGA	ld	a,ore	; legge l'orologio, mette ore nel registro a
	cpi	a, 20	; compara a con il valore 20
	jrz	VENTI	; corrisponde pertanto salta a VENTI
	jp	ALTRO1	; non sono le venti ma altro pertanto salta
VENTI	ld	a,minuti	; legge i minuti dall'orologio
	cp	a,tz1	; confronta minuti tz1 zon1 con i minuti orolog.
	jrnz	ANCORA1	; non è 0 pertanto non ha finito di irrigare
ANCORA1	jp	FERMA	; sono finiti i minuti tz1=0 pertanto va a fermare
	ld	a,pompa	; mette in a il valore della variabile pompa
APRE1	jrz	APRE1	; pompa=0 cioè ancora ferma, apre prima V1
	jp	INIZIO	; pompa=1=già ON, ricomincia da inizio
	ldi	port_a,11111110b	; bit0=0 cioè basso il punto A del buffer B1 apre V1
	call	RITARDO	; va a fare un ritardo di 10 secondi per apertura V1
ALTRO1	ldi	port_a,11111111b	; tutti bit a 1 buffer disabilitati, motori valvole fermi
	ldi	v1,1	; V1 = 1 ricorda valvola V1 aperta
	jp	AVVIA_P	; va ad avviare la pompa
	cpi	a,21	; compara a con il valore 21
VENTUNO	jrz	VENTUNO	; corrisponde pertanto salta a VENTUNO
	jp	ALTRO2	; non sono le venti ma altro pertanto salta
	ld	a,minuti	; legge i minuti dall'orologio
ANCORA2	cp	a,tz2	; confronta minuti tz2 zon2 con i minuti orolog.
	jrnz	ANCORA2	; non sono uguali pertanto non ha finito di irrigare
	jp	FERMA	; sono finiti i minuti tz2=0 pertanto va a fermare
	ld	a,pompa	; mette in a il valore della variabile pompa
APRE2	jrz	APRE2	; pompa=0 cioè ancora ferma, apre prima V2
	jp	INIZIO	; pompa=1=già ON, ricomincia da inizio
	ldi	port_a,11111011b	; bit0=2 cioè basso il punto A del buffer B2 apre V2
	call	RITARDO	; va a fare un ritardo di 10 secondi per apertura V2
ALTRO2	ldi	port_a,11111111b	; tutti bit a 1 buffer disabilitati, motori valvole fermi
	ldi	v2,1	; v2 =1 ricorda valvola V2 aperta
	jp	AVVIA_P	; va ad avviare la pompa
	cpi	a,22	; compara a con il valore 22
VENTIDUE	jrz	VENTIDUE	; corrisponde pertanto salta a VENTIDUE
	jp	ALTRO3	; non sono le venti ma altro pertanto salta
	ld	a,minuti	; legge i minuti dall'orologio
ANCORA3	cp	a,tz3	; confronta minuti tz3 zon3 con i minuti orolog.
	jrnz	ANCORA3	; non è 0 pertanto non ha finito di irrigare
	jp	FERMA	; sono finiti i minuti tz3=0 pertanto va a fermare
	ld	a,pompa	; mette in a il valore della variabile pompa
APRE3	jrz	APRE3	; pompa=0 cioè ancora ferma, apre prima V3
	jp	INIZIO	; pompa=1=già ON, ricomincia da inizio
	ldi	port_a,11101111b	; bit0=4 cioè basso il punto A del buffer B3 apre V3
	call	RITARDO	; va a fare un ritardo di 10 secondi per apertura V3
ALTRO3	ldi	port_a,11111111b	; tutti bit a 1 buffer disabilitati, motori valvole fermi
	ldi	v3,1	; v3 =1 ricorda valvola V3 aperta
	jp	AVVIA_P	; va ad avviare la pompa
	cpi	a,23	; compara a con il valore 23
VENTITRE	jrz	VENTITRE	; corrisponde pertanto salta a VENTIDUE
	jp	ALTRO3	; non sono le venti ma altro pertanto salta
	ld	a,minuti	; legge i minuti dall'orologio
	cp	a,tz4	; confronta minuti tz4 zon4 con i minuti orolog.
	jrnz	ANCORA4	; non è 0 pertanto non ha finito di irrigare
	jp	FERMA	; sono finiti i minuti tz4=0 pertanto va a fermare

ANCORA4	ld	a,pompa	; mette in a il valore della variabile pompa
	jrz	APRE4	; pompa=0 cioè ancora ferma, apre prima V4
	jp	INIZIO	; pompa=1=gia ON, ricomincia da inizio
APRE4	ldi	port_a,10111111b	; bit0=6 cioè basso il punto A del buffer B1 apre V4
	call	RITARDO	; va a fare un ritardo di 10 secondi per apertura V4
	ldi	port_a,11111111b	; tutti bit a 1 buffer disabilitati, motori valvole fermi
	ldi	v4,1	; v4 = 1 ricorda valvola V4 aperta
AVVIA_P	set	0,port_b	; bit0=1 avviare la pompa
	jp	INIZIO	; ricomincia dall'inizio
RITARDO	ld	a,secondi	; legge orologio parte dei secondi
	addi	a,10	; aggiunge 10
	cpi	a,60	; verifica se > = 60
	jrnc	DIFF	; se >= 60 salta a fare la differenza
LDXA	ld	x,a	; salva a in x, sono i secondi da raggiungere
LDAS	ld	a, secondi	; rilegge orologio parte secondi
	cp	a,x	; verifica se sono trascorsi 10 sec
	jrnz	LDAS	; non ancora trascorsi 10 sec pertanto ricontrolla
	ret		; sono 10 s e ritorna da dove è partita la chiamata
DIFF	subi	a,60	; sottrae 60 ad a
	jp	LDXA	; ha trovato il valore dei secondi da raggiungere

Si arriva qui quando si finisce irrigazione o si arriva alle ore 00 nel caso di irrigazione di un'ora dalle 23 alle 00; **punto D del flow-chart**

FERMA	res	0,port_b	; a 0 l'uscita per buffer 5, relè OFF pompa ferma
	ld	a,v1	; verifica se V1 era aperta 1=aperta 0=chiusa
	jrz	VV2	; V1 = chiusa pertanto va a V2
	ldi	port_a,11111101b	; mette a 0 bit punto B buffer B1 chiude V1
	call	RITARDO	; 10 secondi di ritardo per la chiusura
	ldi	port_a,11111111b	; tutti punti A e B buffer a 1 motori valvole = OFF
	clr	v1	; ricorda che V1 è chiusa
VV2	ld	a,v2	; verifica se V2 era aperta 1=aperta 0=chiusa
	jrz	VV3	; V2 = chiusa pertanto va a V3
	ldi	port_a,11110111b	; mette a 0 bit punto B buffer B2 chiude V2
	call	RITARDO	; 10 secondi di ritardo per la chiusura
	ldi	port_a,11111111b	; tutti punti A e B buffer a 1 motori valvole = OFF
	clr	v2	; ricorda che V2 è chiusa
VV3	ld	a,v3	; verifica se V3 era aperta 1=aperta 0=chiusa
	jrz	VV4	; V3 = chiusa pertanto va a V4
	ldi	port_a,11011111b	; mette a 0 bit punto B buffer B3 chiude V3
	call	RITARDO	; 10 secondi di ritardo per la chiusura
	ldi	port_a,11111111b	; tutti punti A e B buffer a 1 motori valvole = OFF
	clr	v3	; ricorda che V3 è chiusa
VV4	ld	a,v4	; verifica se V4 era aperta 1=aperta 0=chiusa
	jrz	JPIN	; V4 = chiusa pertanto torna ad IN
	ldi	port_a,01111111b	; mette a 0 bit punto B buffer B4 chiude V4
	call	RITARDO	; 10 secondi di ritardo per la chiusura
	ldi	port_a,11111111b	; tutti punti A e B buffer a 1 motori valvole = OFF
	clr	v4	; ricorda che V4 è chiusa
JPIN	jp	INIZIO	; ricomincia dall'inizio in A del flow-chart

Arriva qui quando viene eseguito call ADC per la lettura della temperatura ogni ora.
 Per una accurata lettura immune da errori questa routines esegue 256 volte la lettura della temperatura ed esce con il valore mediato molto preciso (il tutto in circa 18 mS).
 Per avere i gradi °C basta dividere il numero x 5 infatti per 20°C l'ADC legge 2V considerato che a 5 V ho 255 a 2 V avrò $2 \times 255/5 = 102$ (numero dell'ADC a 20°C) $20^\circ\text{C} = T = 102/5$
Per evitare questa divisione si sarebbe potuto amplificare 5 anziché 10.

```

ADC      res    4,ior          ; disabilita interrupt, fa solo la conversione
         clr    resto        ; predisporre resto a zero per il conteggio
         clr    risul        ; predisporre risul a zero per il conteggio
         ldi    aa,255        ; numero di campionamenti da effettuare
AVVER1   ldi    adcr,00110000b ; abilita convertitore ADC
ATT      jrr    6,adcr,ATT    ; attende fine conversione, circa 70 micro secondi
         nop                    ; non opera
         ld     a,addr        ; mette in a il valore della conversione
         add   a,resto
         jrnc  AVVER2        ; se non ce riporto salta
         inc   risul          ; dopo 255 riporti qui ci sarà il valore mediato
AVVER2   ld     resto,a       ; salva il contenuto di a nella var. resto
         ld     a,aa          ; prende aa per verificare fine conversione
         jrz   AVVER4        ; se 0 allora ha finito e salta a AVVER4
         dec   aa            ; altrimenti una conversione in meno
AVVER4   jp     AVVER1        ; e ripete altra conversione
         ld     a,resto        ; prende resto per valutare arrotondamento
         cpi   a,127          ; compara con il valore 127
         jrc   AVVER3        ; se a < 127 approssima per difetto
         inc   risul          ; se a > 127 approssima per eccesso
AVVER3   ld     a,risol        ; mette risultato in a e
         add   a,restT        ; somma eventuale resto precedente
         ldi   y,5            ; trova la T in gradi °C
         call  DIVIDI         ; ritorna con il risultato in w= temperatura letta
         ld     a,rest        ; prende il resto della divisione
         ld     restT,a       ; e lo ricorda per sommarlo alla prossima
         ld     a,ore         ; prende le ore
         subi  a,7
         cpi   a,1            ; se a=1 è la prima T misurata
         jrnz  MED           ; se non è la prima allora va a mediare
         ld     a,w           ; prende la prima T letta
         ld     tm           ; e la salva in tm
         ret                    ; torna indietro
MED      ld     y,a           ; divisore per media
         ld     a,w           ; prende lettura temperatura ultima
         ld     temper,a     ; salva ultima letta
         cp    a,tm          ; verifica quale è maggiore
         jrnc  SUBBA        ; se a >= tm salta e fa a-tm
         ld     a,tm          ; altrimenti
         sub   a,temper      ; fa tm-a
         jp    CADIV
SUBBA    sub   a,tm          ; sottrae attuale con precedente
CADIV    call  DIVIDI         ; calcola la media progressiva delle T già misurate
         ld     a,restM      ; prende resto precedente
         add   a,rest        ; somma all'attuale
         cpi   a,10          ; se >=10 aggiunge 1 al risultato w=w+1
         jrnc  SUB10        ; se a >= 10 salta
         ld     restM,a      ; altrimenti ricorda nuovo resto
         jp    SAVE          ; va a salvare nuova media

```

```

SUB10    subi    a,10          ; sottrae 10
         ld      restM,a      ; ricorda nuovo resto, eccedenza oltre il 10
         inc     w            ; la somma dei resti da un riporto pertanto w = w+1
SAVE     ld      a,w          ; recupera risultato della divisione
         add     a,temper     ; aggiunge temper e trova nuova Tm
         ld      tm,a         ; memorizzare la tm ed esce
         ld      a,y          ; verifica se siamo a 12 misure
         cpi    a,12         ; se < allora salta e ritorna
         jrz    RITO         ; altrimenti
         ld      a,restM     ; valuta se il resto > di 5 aggiunge 1 a Tm
         cpi    a,5
         jrc    RITO         ; se < ritorna senza arrotondare
         inc     tm          ; arrotonda tm+1
RITO     ret

```

Per trovare la media usiamo solitamente sommare i valori delle 12 misure e poi dividere per 12, un metodo che semplifica il problema dei riporti per non utilizzare più byte, è quello sopra esposto che consiste nel fare subito la media applicando questo algoritmo:

$T_{mu} = [(T_u - T_{mp}) / (Nr.L)] + T_{mp}$ dove:

T_{mu} = ultima da trovare (alla prossima misura diventa T_{mp})

T_{mp} = media precedente

T_u = ultima T letta

Nr.L = numero delle letture fatte

Facciamo un esempio:

prima T misurata = 10 °C seconda 14°C media $T_m = [(14-10)/2] + 10 = 12$

Terza misura 15 °C $T_m = [(15-T_{mp})/3] + T_{mp} = [(15-12)/3] + 12 = 13$

Quarta misura 21 °C $T_m = [(21-13)/4] + 13 = 15$ come si vede si può continuare e l'ultimo valore trovato è la media di tutte le temperature precedentemente misurate.

Come si può osservare non è necessario ricordare le precedenti letture ma basta ricordare la media ultima ricavata con questo algoritmo.

```

DIVIDI   ; entra con a= dividendo e y= divisore
         clr    w            ; cancella w = 0 per ricordare risultato
         cp     a,y          ; verifica se divisibile
         jrc    FINE        ; se a non è divisibile fine
SOTTR    sub    a,y          ; sottrae divisore
         inc     w            ; conta quante volte ci sta y
         cp     a,y          ; verifica se a è ancora divisibile
         jrnc  SOTTR        ; sottrae ancora se a >= y
         ld     rest,a      ; ricorda il resto che sommerà alla prossima
FINE     ret                ; ed esce con il risultato in w e resto

```