



Ministero della Pubblica Istruzione

YAM1 - ESAMI DI MATURITÀ TECNICA INDUSTRIALE SPERIMENTALE PROGETTO "AMBRA"

INDIRIZZO: ELETTRONICA INDUSTRIALE

Una azienda automobilistica, nel corso dei test su pista di un nuovo tipo di motore, intende monitorare in tempo reale la temperatura di funzionamento in otto punti del motore stesso. A tal fine vengono impiegate delle termoresistenze, in cui la dipendenza della resistenza elettrica dalla temperatura si può supporre data dalla seguente relazione:

$$R(T) = R(0) * (1 + a * T), \quad \text{con } 0 < T < 400 \text{ } ^\circ\text{C}$$

dove:	(T)	Temperatura in gradi centigradi;
	R(T)	Resistenza elettrica a T °C;
	R(0) = 100 Ω	Resistenza a 0 °C;
	a = 3,675 * 10 ⁻³ °C ⁻¹	Costante di temperatura media nel range di impiego.

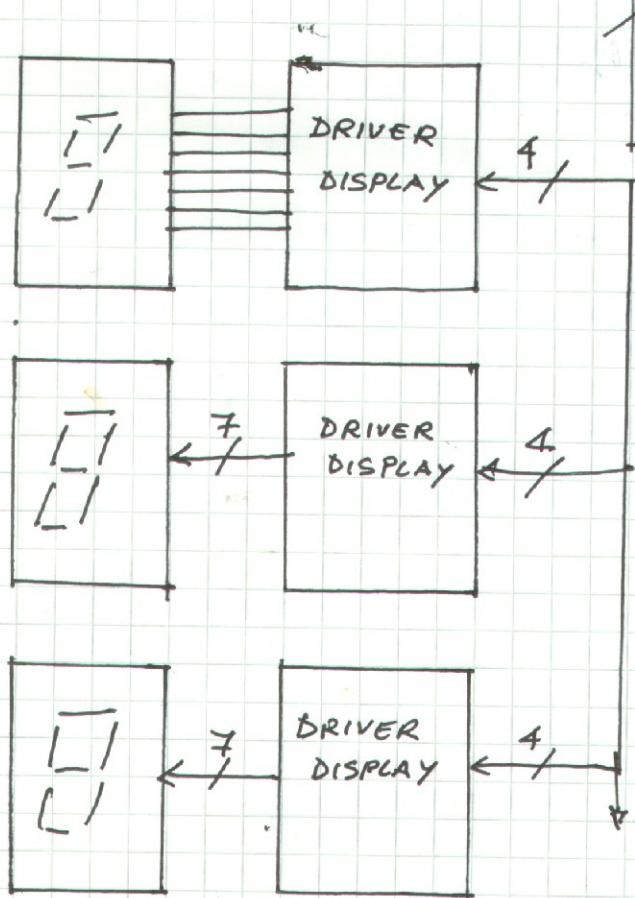
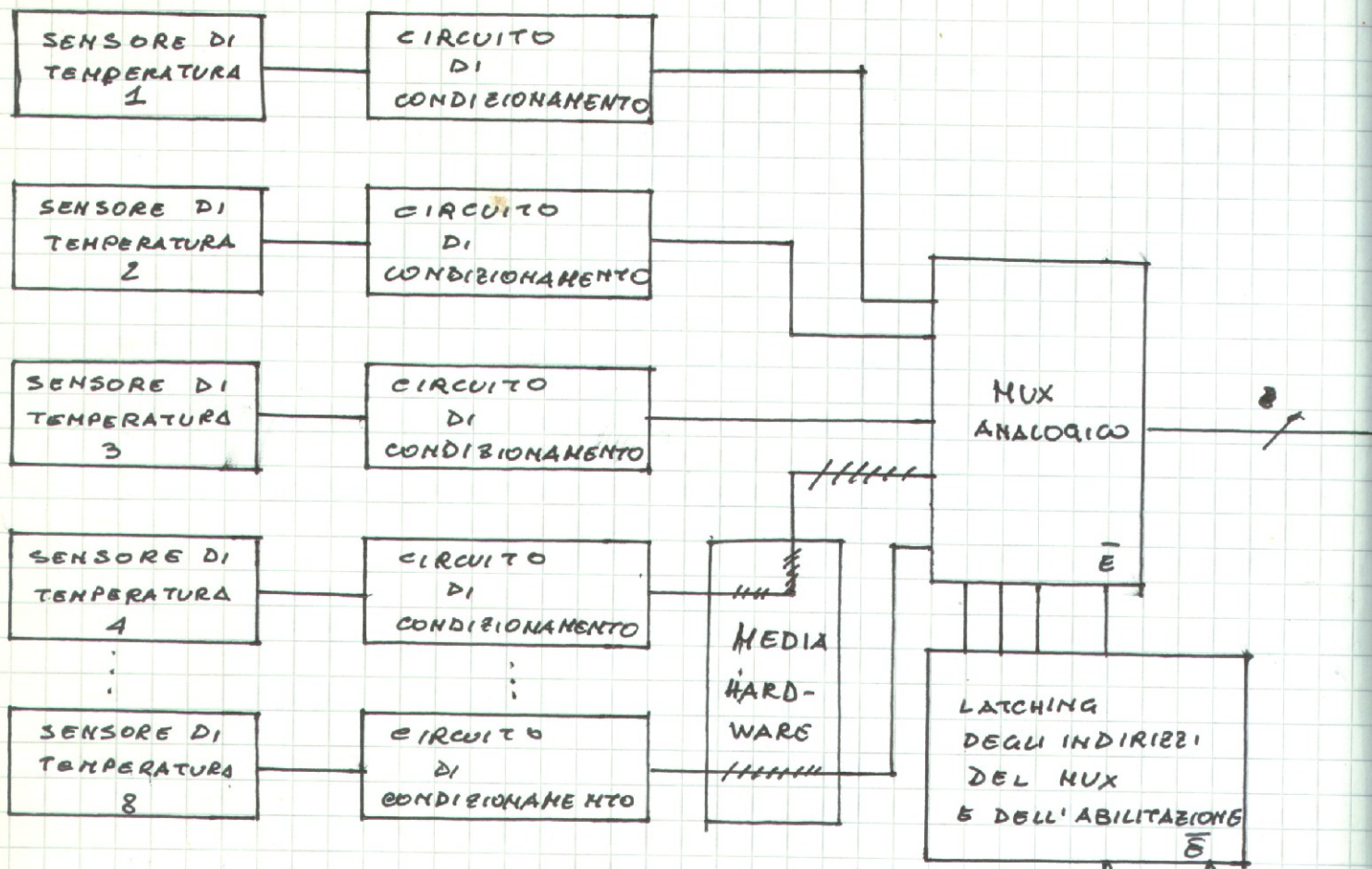
I punti del motore in cui si fa il rilevamento sono i seguenti:

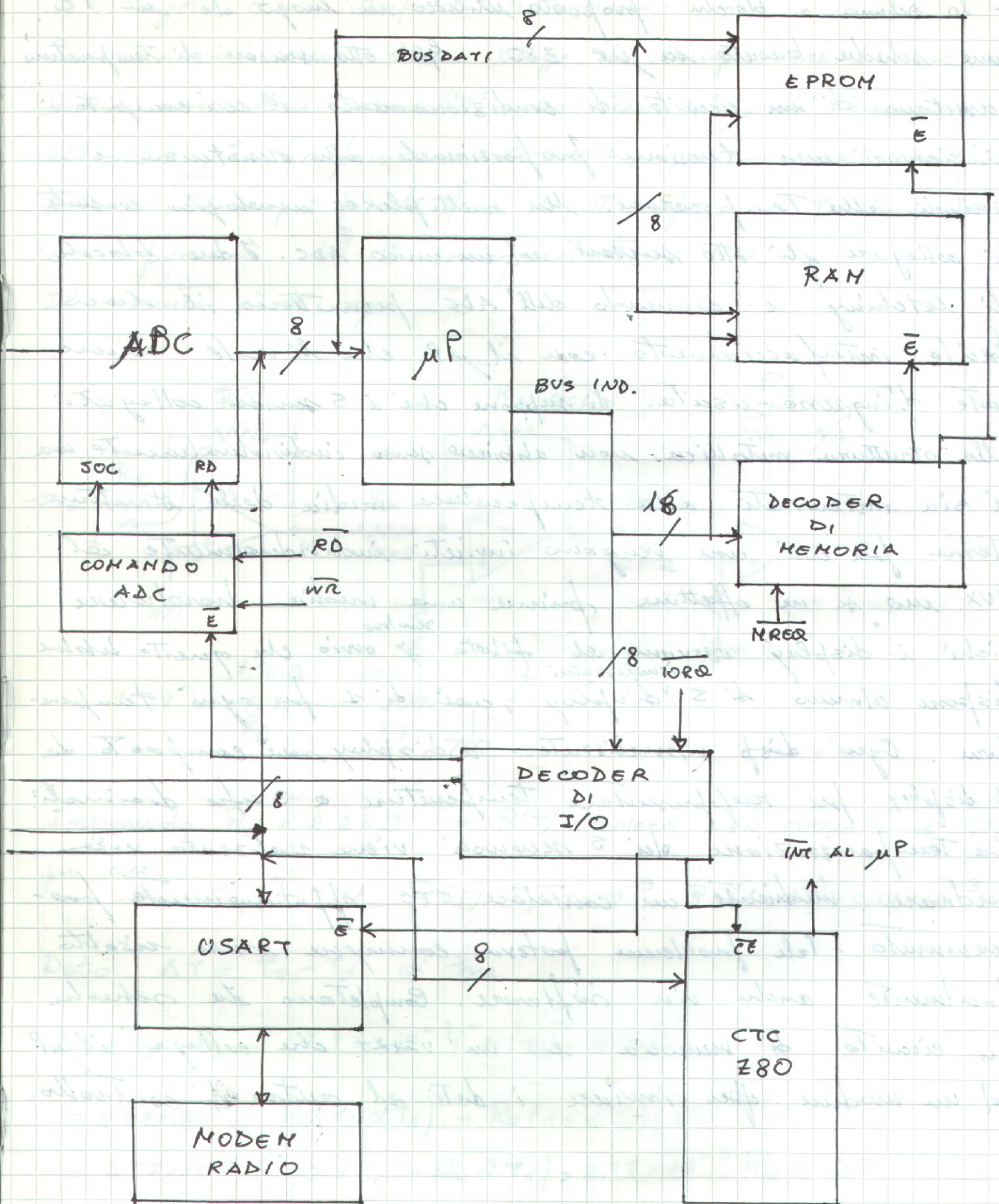
- 1) Liquido di raffreddamento, con range 50 ... 200 °C;
- 2) Liquido lubrificante, con range 50 ... 250 °C;
- 3) Collettore di scarico, con range 200 ... 400 °C;
- 4) N° 5 punti localizzati sulla struttura metallica, con range 150 ... 300 °C.

Un sistema basato su P.C. deve acquisire i dati di temperatura ogni 5 secondi con risoluzione pari a 8 bit, deve visualizzarli su display alfanumerico e inviarli ad una stazione fissa di controllo posta ai margini della pista, a una distanza massima dal veicolo di 1 Km.

Il candidato, formulate le necessarie ipotesi aggiuntive:

- a) elabori lo schema a blocchi dettagliato di un sistema di acquisizione dei dati adeguato alle problematiche proposte, illustrando le soluzioni adottate e le interazioni fra i vari componenti;
- b) progetti, dimensionandola in almeno un caso, l'elettronica di interfacciamento dei sensori;
- c) rappresenti, nel modo che ritiene più opportuno, le procedure relative alla gestione generale del sistema;
- d) progetti e codifichi in un linguaggio di propria conoscenza, sulla base dell'hardware proposto, la procedura di acquisizione di un valore di temperatura;
- e) illustri un sistema idoneo al collegamento con la stazione di controllo.

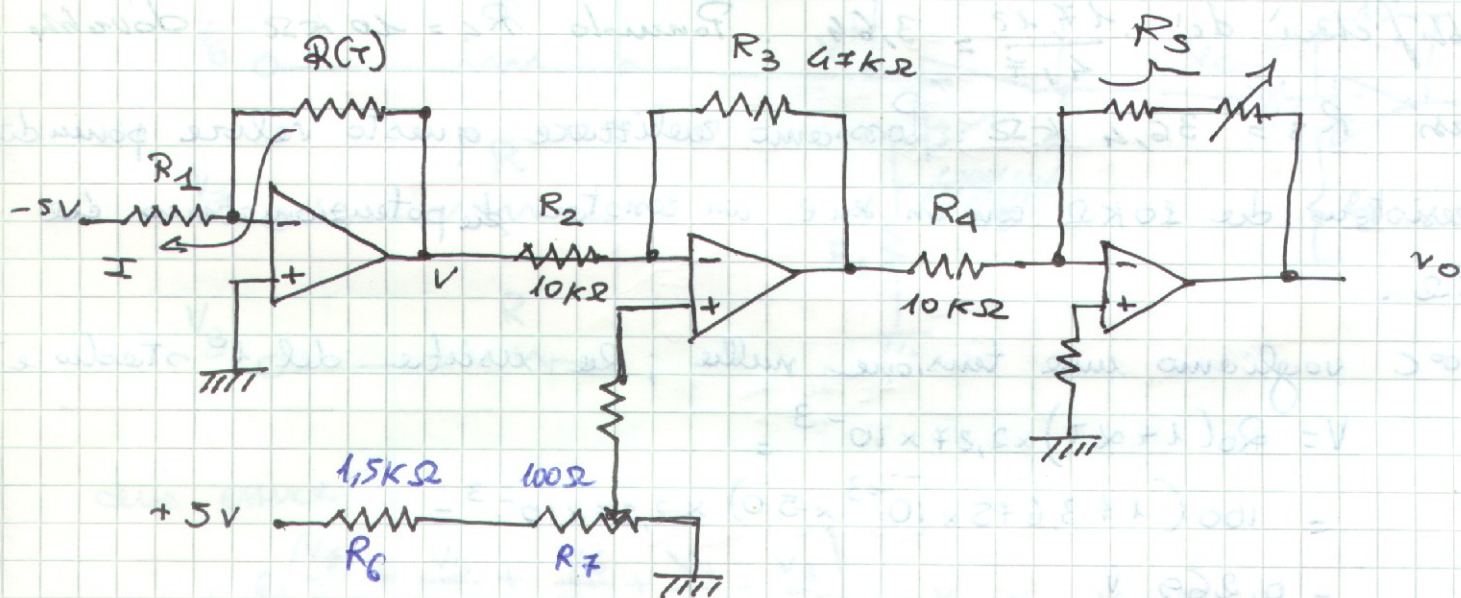




1- Lo schema a blocchi proposto utilizza un luogo di un
una scheda basata su μP Z80. Gli otto sensori di tempera-
necessitano di un circuito di condizionamento il cui compito
di ricevere una tensione proporzionale alla resistenza e
quindi, alla temperatura. Un multiplexer analogico consente
di collegare gli otto sensori ad un unico ADC. I due blocchi
di latching e comando dell'ADC permettono il loro
facile interfacciamento con il μP che li vuole come
parte d'ingresso-uscita. Si suppone che i 5 sensori collegati
alla struttura metallica non abbiano senso individualmente.
Si sia interessati alla temperatura media delle strutture
sterne per cui non vengono inviati individualmente al
MUX ma se ne effettua prima una media hardware.
Poiché i display servono al pilota ^{sempre} è ovvio che questo debba
disporre almeno di 5 ^{visualizzatori} display, cioè di 1 per ogni tempera-
tura. Ogni disp visualizzatore a display non composto da
3 display per rappresentare temperature a 3 cifre decimali.
La temperizzazione dei 5 sensori viene realizzata via
hardware mediante un contatore etc opportunamente pro-
grammato. Tali problemi poteva comunque essere risolti
facilmente anche via software. Completano il schema
un circuito di memoria e un USART che collega il μP
ad un modulo per inviare i dati al centro di controllo.

2 - Progetto delle interfacce elettroniche.

Poiché i vari sensori hanno range di temperature diverse che devono essere tutti ricondotti allo stesso range di tensioni dell'ADC che imponiamo 0-5 volt, poniamo 0 volt corrispondente al limite più piccolo fra i limiti inferiori di T che è 50°C e imponiamo 5 volt corrispondente a 400°C .
Il circuito è il seguente



Imponendo $R_1 = 2,2\text{k}\Omega$ in $R(T)$ circolerà una corrente di $2,27\text{mA}$ in cui

$$V = R(T) \cdot 2,27 \times 10^{-3} = R_0(1 + \alpha T) \times 2,27 \times 10^{-3}$$

Dato $\Delta T = T_2 - T_1$ si ha

$$\Delta V = V_2 - V_1 = 2,27 \times 10^{-3} (R(T_2) - R(T_1)) =$$

$$= 2,27 \times 10^{-3} (R_0(1 + \alpha T_2) - R_0(1 + \alpha T_1)) =$$

$$= 2,27 \times 10^{-3} (R_0 \alpha T_2 - R_0 \alpha T_1) = 2,27 \times 10^{-3} R_0 \alpha \Delta T$$

$$\frac{\Delta V}{\Delta T} = 2,27 \times 10^{-3} \times 100 \times 3,675 \times 10^{-3} = 8,34 \times 10^{-4}$$

Noi vogliamo che $\Delta V = 5 - 0 = 5$ corrisponda a $\Delta T = 400 - 50 = 350$

$$\text{cioè } \frac{\Delta V}{\Delta T} = \frac{5}{350} = 0,014$$

cioè l'uscita del 1° stadio va amplificata di

$$A_v = \frac{0,1428 \times 10^{-3}}{8,34 \times 10^{-4}} = 17,12$$

Imponendo $R_2 = 10 \text{ k}\Omega$ e $R_3 = 47 \text{ k}\Omega$, il 2° stadio amplifica in valore assoluto di 4,7 volte. Quindi il 3° stadio amplificherà di $\frac{17,12}{4,7} = 3,64$. Ponendo $R_4 = 10 \text{ k}\Omega$ dovremo

avere $R_5 = 36,4 \text{ k}\Omega$. Possiamo realizzare questo valore ponendo un resistore da $20 \text{ k}\Omega$ con in serie un resistore di potenziometro da $50 \text{ k}\Omega$.

A 50°C vogliamo una tensione nulla; la uscita del 1° stadio

$$\begin{aligned} V &= R_0(1 + \alpha T) \times 2,27 \times 10^{-3} = \\ &= 100(1 + 3,675 \times 10^{-3} \times 50) \times 2,27 \times 10^{-3} = \\ &= 0,269 \text{ V} \end{aligned}$$

Dobbiamo eliminare questo offset di Tensione. Sul secondo stadio esso verrebbe amplificato del fattore $\frac{R_3}{R_2}$; se applichiamo una Tensione V_+ al morsetto positivo del 2° operazionale, in uscita, per la sovrapposizione degli effetti, avremo

$$V' = V_+ \left(1 + \frac{R_3}{R_2}\right) - \frac{R_3}{R_2} V$$

che vogliamo $V' = 0$ per cui deve essere

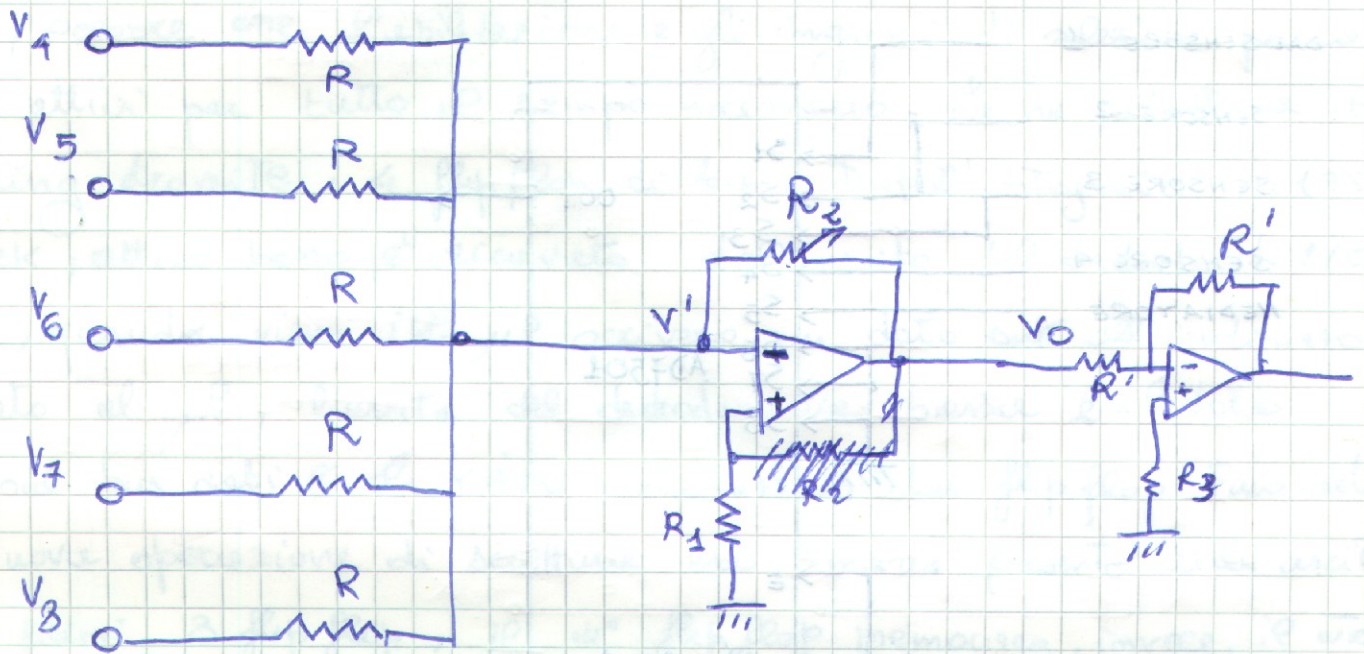
$$V_+ = \frac{\frac{R_3}{R_2}}{1 + \frac{R_3}{R_2}} V = \frac{4,7}{5,7} \times 0,269 = 0,22 \text{ V}$$

$$V_+ = \frac{5 \cdot R_7}{R_6 + R_7}$$

$$\frac{R_6 + R_7}{R_7} = \frac{0,22}{0,22} \frac{5}{0,22} = 22,72$$

Con un potenziometro da ~~100~~ $10\text{ k}\Omega$ e una resistenza da $1\text{ k}\Omega$ possiamo ottenere il risultato voluto.

I 5 sensori collegati alle ^{strutture} testine metalliche vanno poi inviati ad una media hardware



deve essere

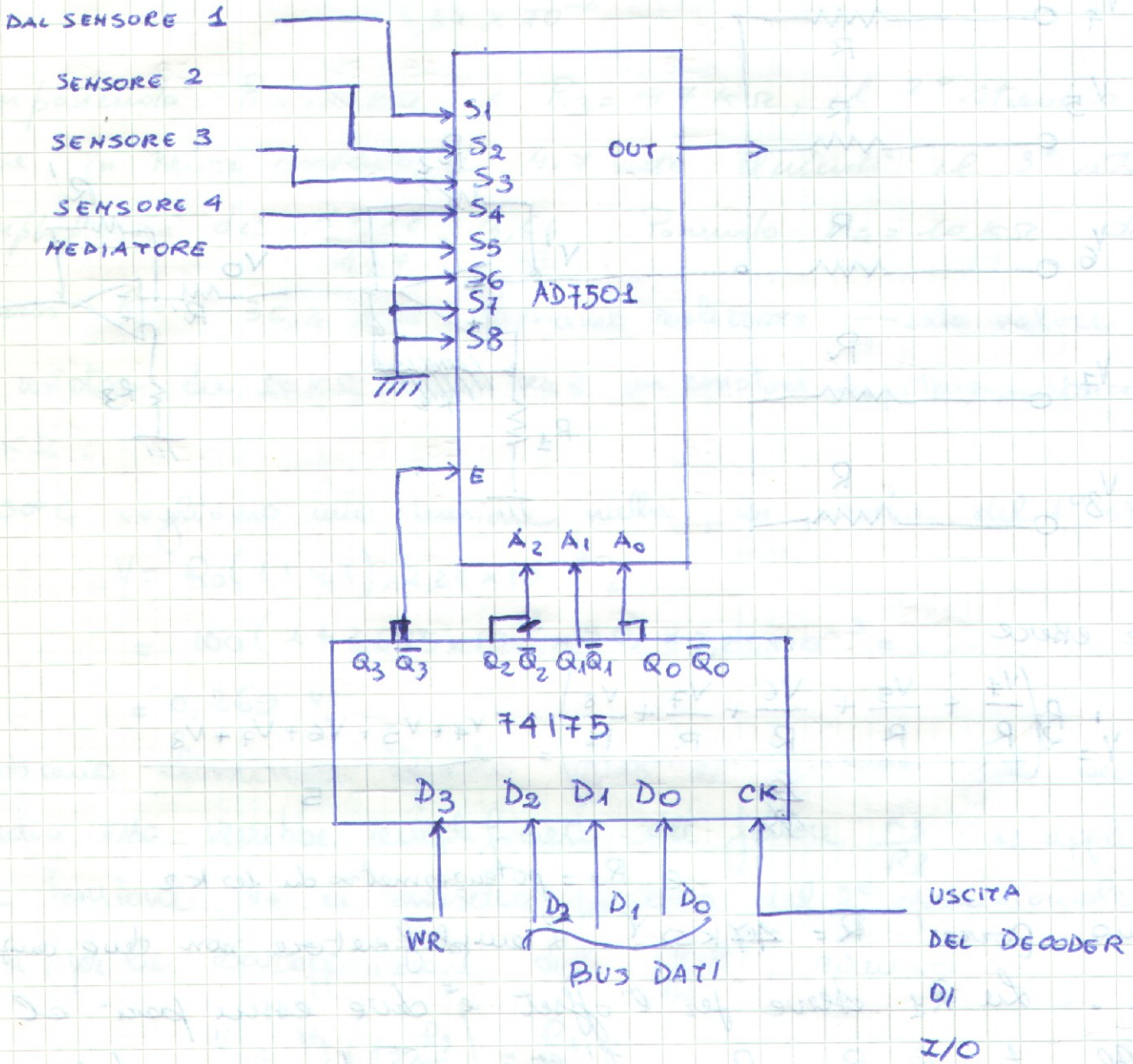
$$V_0 = \frac{R_2 \left(\frac{V_4}{R} + \frac{V_5}{R} + \frac{V_6}{R} + \frac{V_7}{R} + \frac{V_8}{R} \right)}{\frac{5}{R}} = \frac{V_4 + V_5 + V_6 + V_7 + V_8}{5}$$

e $R_2 =$ potenziometro da $10\text{ k}\Omega$

Possiamo porre $R = 10\text{ k}\Omega$. L'amplificatore non deve amplificare. La R_1 serve per l'offset e deve essere pari al parallelo tra le R e R_2 . L'ultimo stadio non deve amplificare ma invertire soltanto per cui $R' = R_1 = 10\text{ k}\Omega$
 $R_3 = \frac{R'}{2}$ per l'offset.

INTERFACCIA PER LA CONVERSIONE A/D.

Per collegare tutti i sensori all'unico ADC abbiamo bisogno di un multiplexer analogico, ad esempio l'AD7501, multiplexer analogico CMOS ad 8 ingressi $S_1 + S_8$ cui collegare le uscite dei sensori e



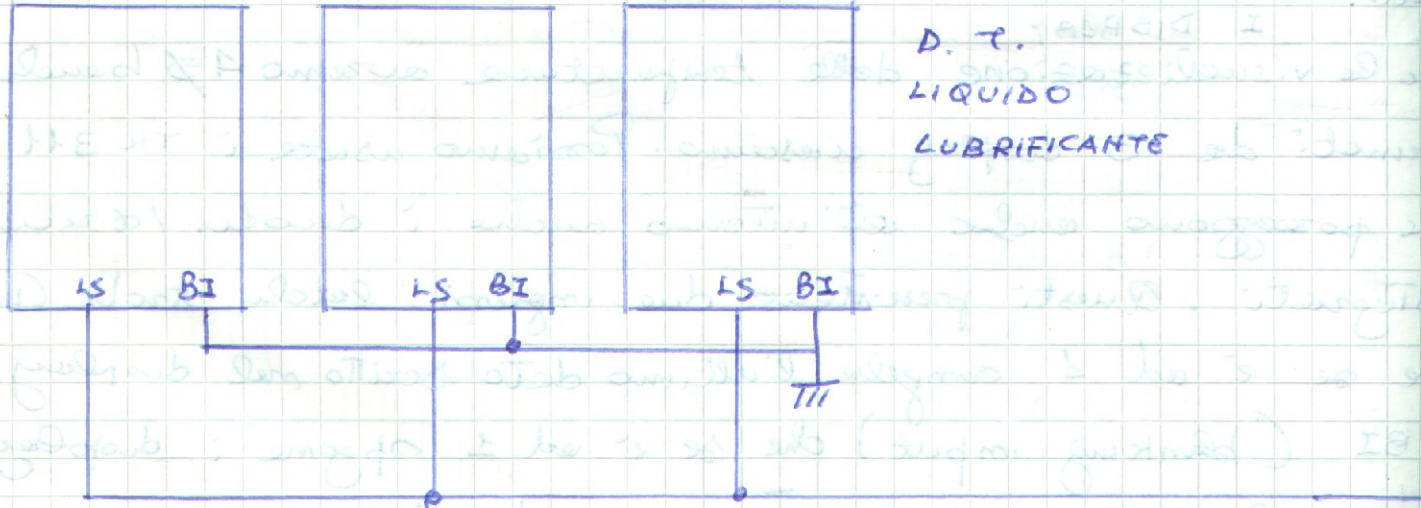
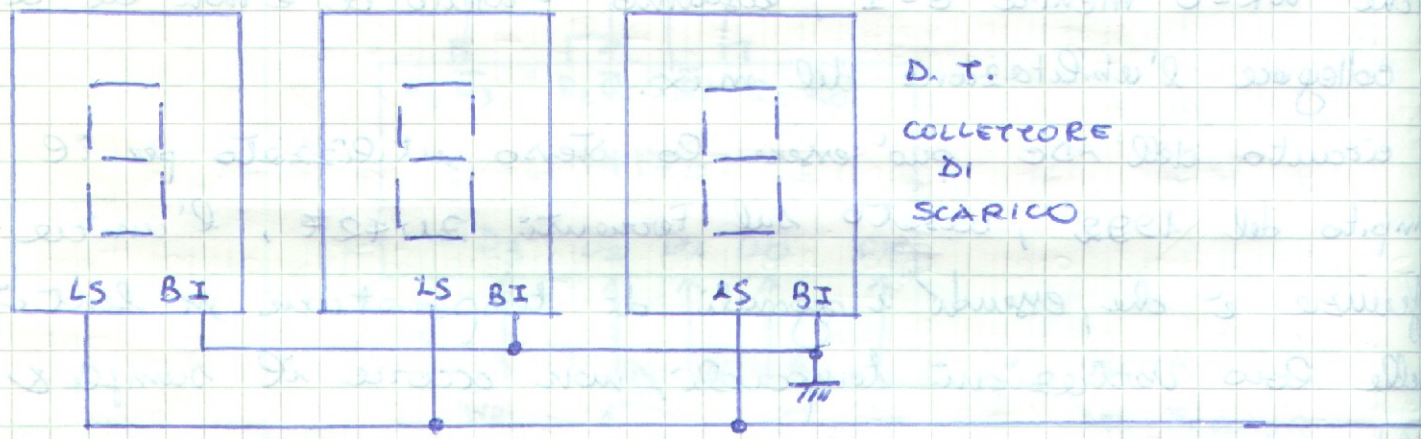
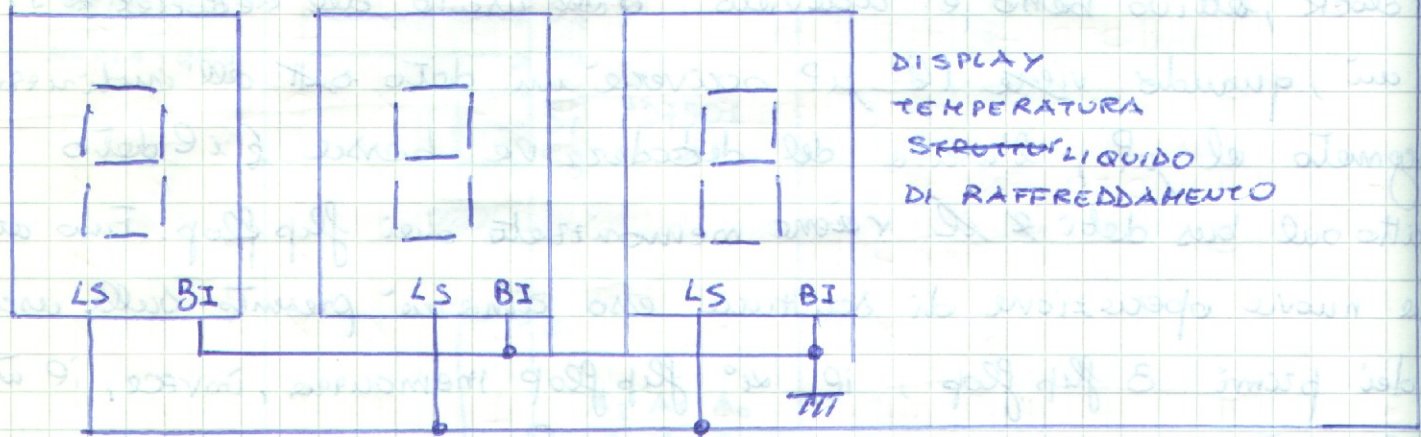
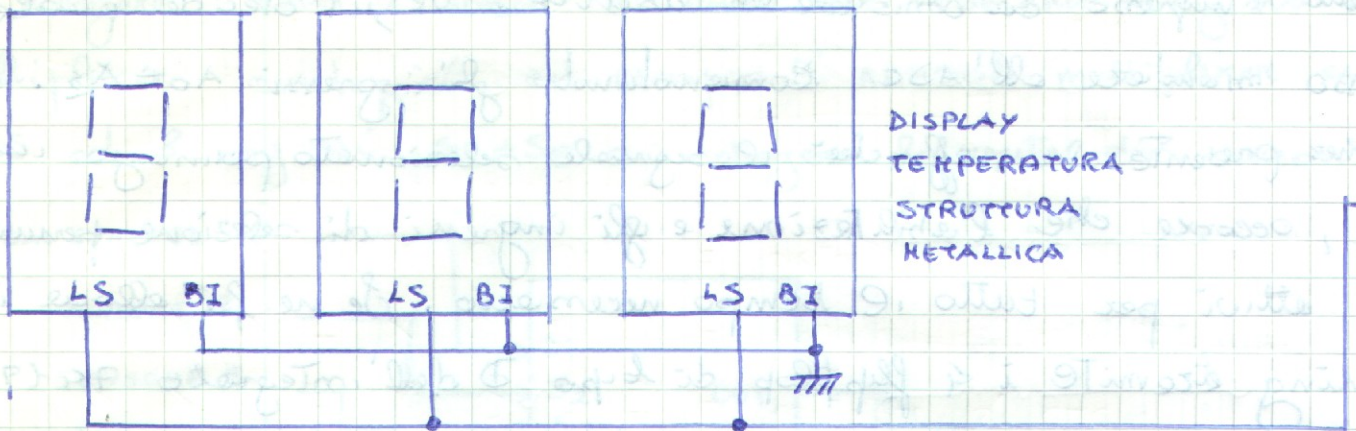
l'uscita del mediatore. L'abilitazione è attiva alta, A_0, A_1, A_2 abilitano l'impresso da smistare in uscita. Il μP decide quale impresso mandare all'ADC domandando gli impressi $A_0 \div A_2$.
Da tener presente che, affinché il segnale selezionato permanga in uscita, occorre che l'abilitazione e gli ingressi di selezione permangano attivi per tutto il tempo necessario. Se ne fa^o allora il latching tramite i 4 flip flop di tipo D dell'integrato 74175. Il clock, attivo basso, è ricevuto dall'uscita del decoder di 2/10 per cui, quando viene il μP scrivere un dato sull'indirizzo assegnato al μP , l'uscita del decoder va bassa & il dato scivola sul bus dati & è viene memorizzato dai flip flop. Fatto ciò una nuova operazione di scrittura esso emerge presente sulle uscite a dei primi 3 flip flop, il 4° flip flop memorizza, invece, il \overline{WR} . Poiché $\overline{WR} = 0$ mentre $E = 1$, avremo l'uscita \overline{a} e non a & per collegare l'abilitazione del max.

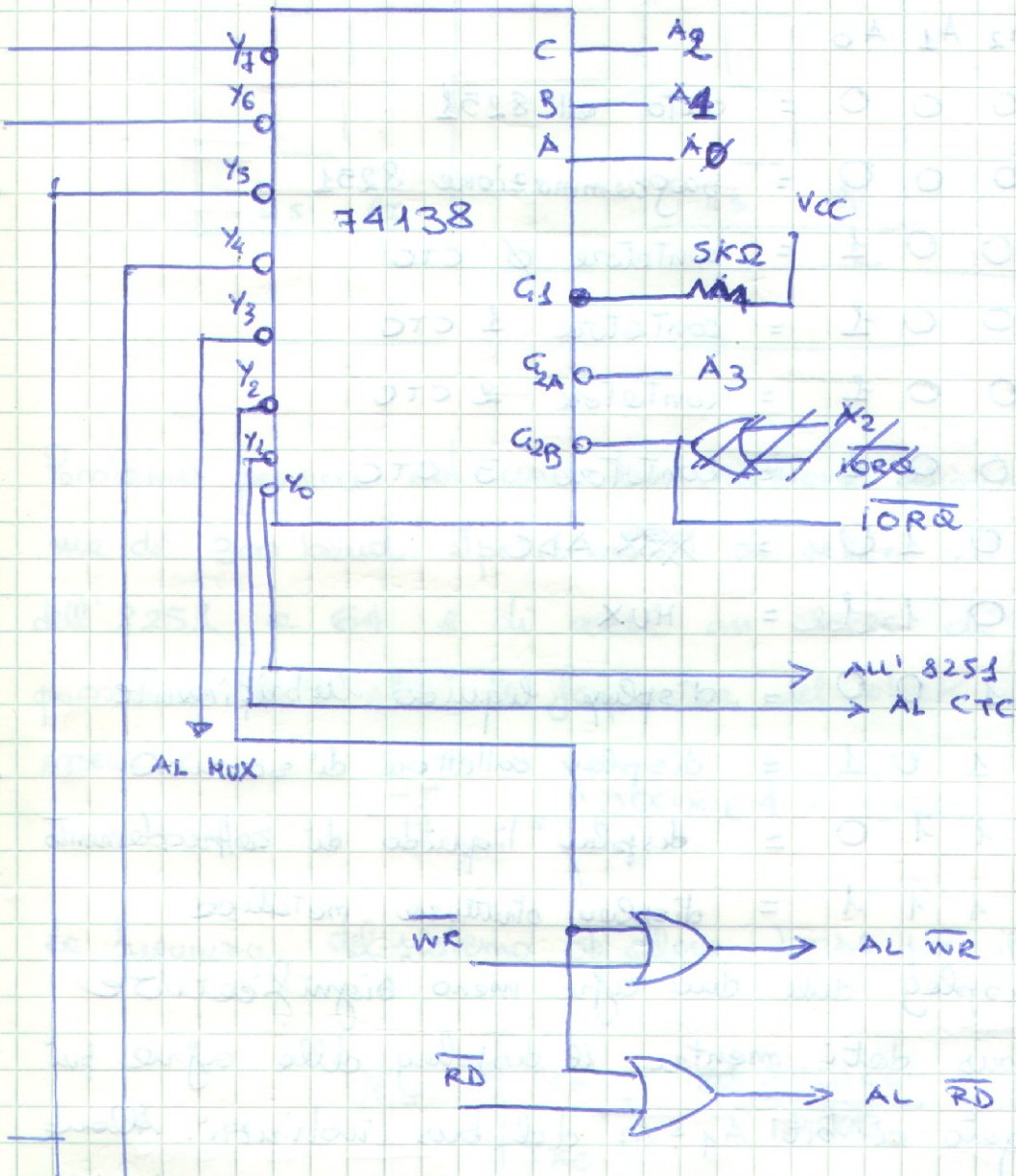
Il circuito dell'ADC può essere lo stesso utilizzato per il compito del 1992, basato sul Ferranti 20427, l'unica differenza è che, essendo i segnali di temperatura μ lenti nelle loro variazioni temporali, non occorre il sample & hold.

I DISPLAY

Per la visualizzazione delle temperature avremo 4 μ canali formati da 3 display ciascuno. Possiamo usare i TL311 che possiedono anche all'interno anche i decoder / driver integrati. Questi presentano due ingressi: latch strobe (LS) che se è ad 1 congela l'ultimo dato scritto nel display & BI (blanking input) che se è ad 1 spegne i display. Potremo BI a zero per mantenere i display sempre accesi & collegheremo di LS di ogni banco collegati ^{ad una} all'uscita

porta di I/O





tenendo presente che invieremo A_4 e A_5 agli ingressi CS_0 e CS_1 del CTC e A_4 all'ingresso C/D dell'8251 ovvero la seguente mappa di I/O

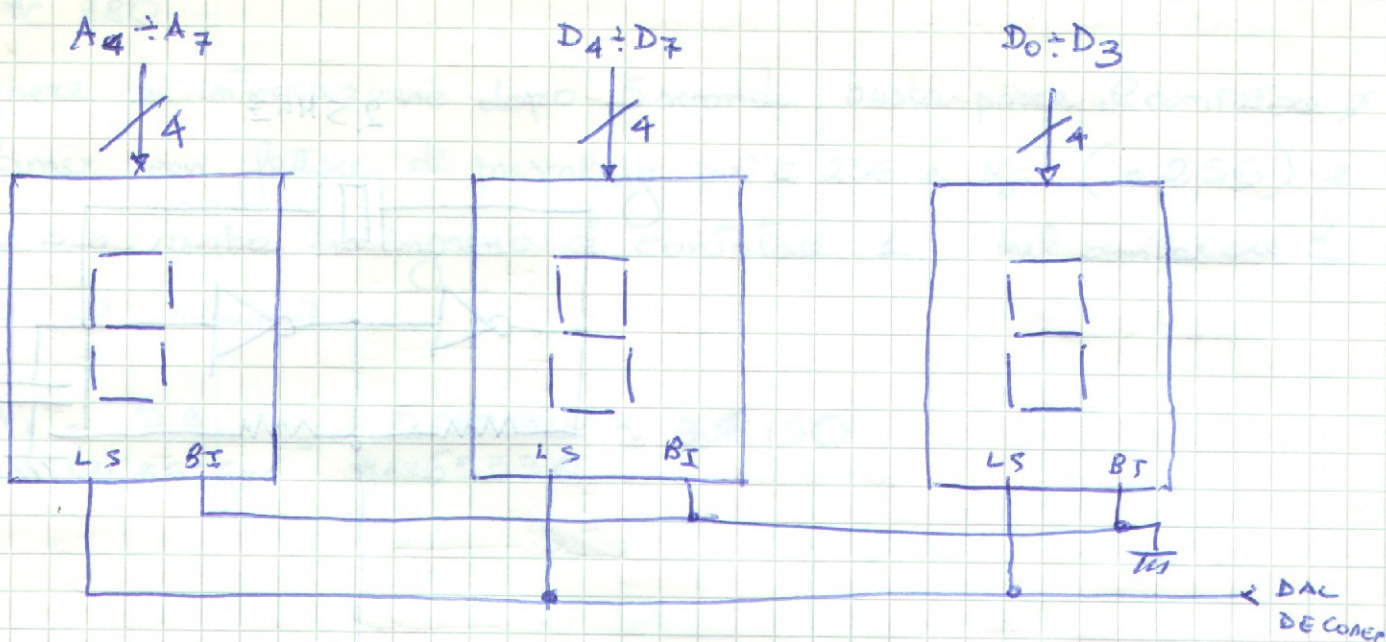
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
X	X	X	0	X	0	0	0	= dato all'8251
X	X	X	1	X	0	0	0	= programmazione 8251
X	X	0	0	X	0	0	1	= contatore \emptyset CTC
X	X	0	1	X	0	0	1	= contatore 1 CTC
X	X	1	0	X	0	0	1	= contatore 2 CTC
X	X	1	1	X	0	0	1	= contatore 3 CTC
X	X	X	X	X	0	1	0	= MUX ADC
X	X	X	X	X	0	1	1	= MUX
X	X	X	X	X	1	0	0	= display liquido lubrificante
X	X	X	X	X	1	0	1	= display collettore di scarico
X	X	X	X	X	1	1	0	= display liquido di raffreddamento
X	X	X	X	X	1	1	1	= display struttura metallica

In ogni banco i 2 display delle due cifre meno significative vengono collegati al bus dati mentre il display delle cifre più significative viene collegato ai bit $A_4 \div A_7$ del bus indirizzi. Allora per far apparire, ad esempio, le cifre 350 sul display del liquido lubrificante si deve il μP deve scrivere all'indirizzo $00110100 = 34H$

il dato $50H$.

CONNESSIONE CON IL MODEM

Ogni 5 secondi vengono prelevati e inviati 4 dati a 8 bit. Nelle nostre ipotesi, decidiamo di inviare i dati trasformati in BCD al centro di controllo per cui diventano 10 byte.



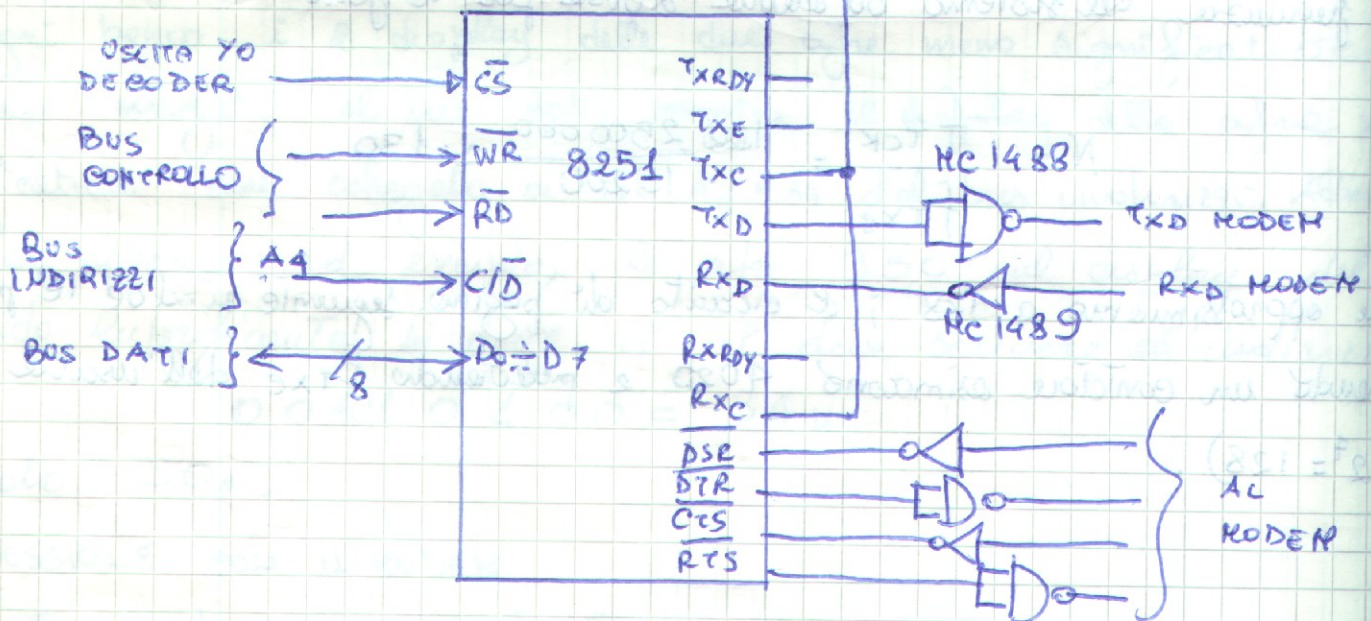
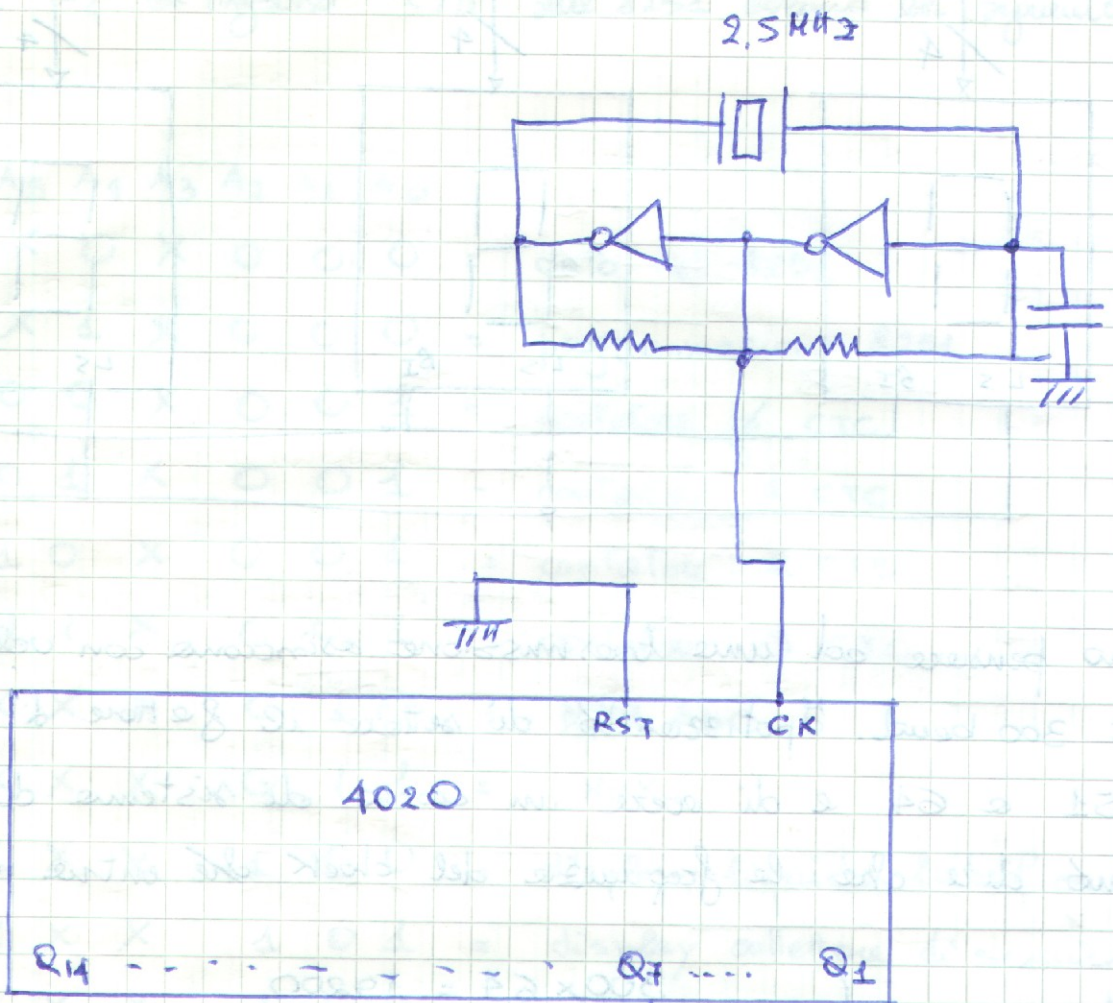
Possiamo pensare ad una trasmissione asincrona con velocità minima di 300 baud. Spostando di sette il fattore di baud rate dell'8251 a 64 e di avere un clock di sistema di 2.5 MHz possiamo dire che la frequenza del clock che entra nell'8251 deve essere

$$f_{TXC} = 300 \times 64 = 19200$$

La frequenza del sistema o allora divisa per il fattore

$$N = \frac{f_{CLK}}{f_{TXC}} = \frac{250000}{19200} \approx 130$$

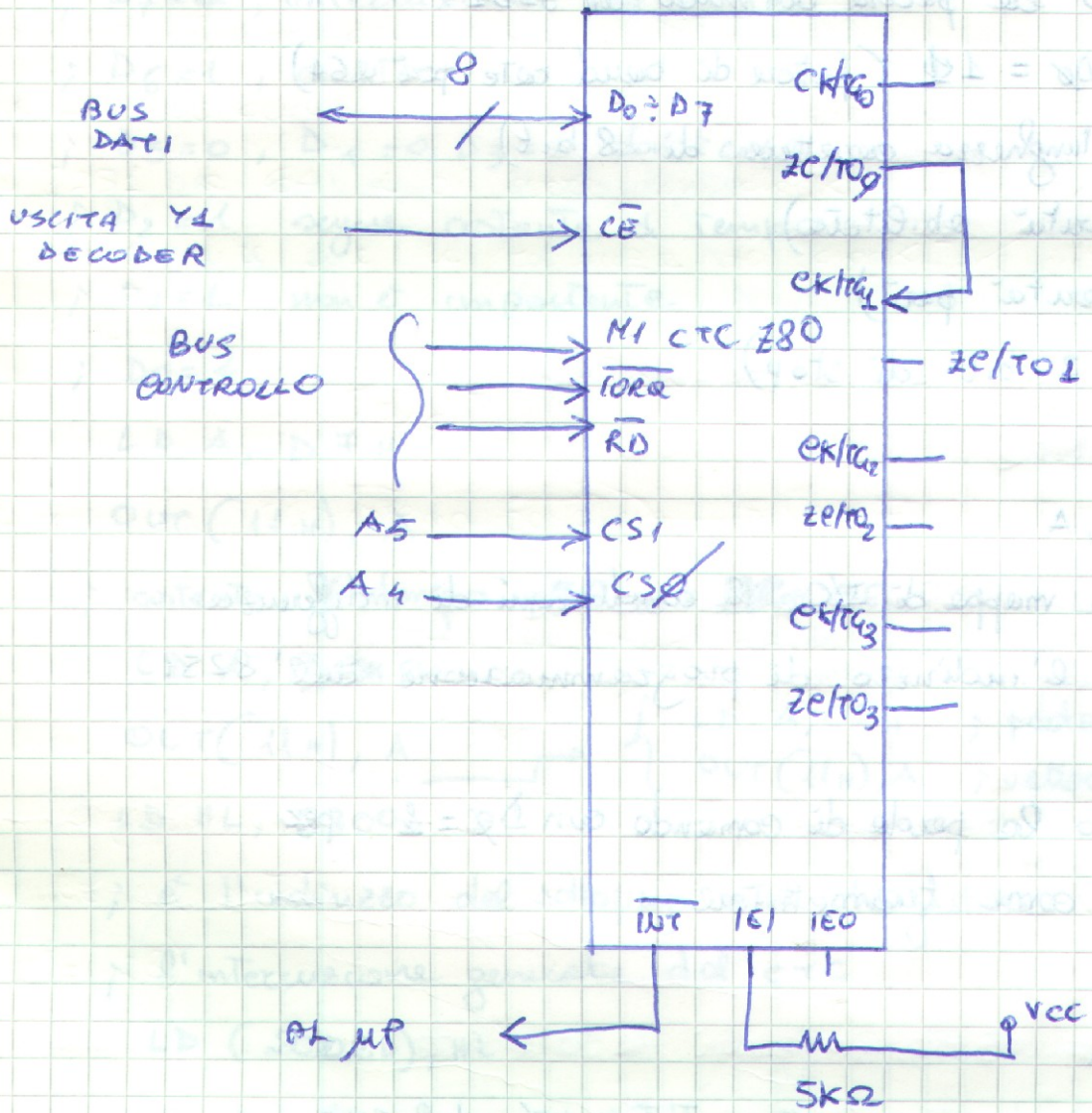
che approssimiamo a 128; il circuito di pagina seguente risolve il problema usando un contatore asincrono 4020 e prelevando f_{TXC} dall'uscita Q_7 ($2^7 = 128$).



CTC 280

Per ottenere un'interruzione dopo 5 secondi basta porre il contatore \emptyset come timer con fattore di prescaling pari a 256 e $N_0 = 0 (= 256)$ e porre le sue uscite in ingresso al contatore 1; nel contatore 1 porremo il numero

$$N_1 = \frac{5s}{0,465536\mu s} = \frac{5000000}{0,465536} \approx 1074190$$



Soluzione identica a quella del compito precedente.

SOFTWARE

Cominciamo con il software di inizializzazione della scheda

```
ORG 0000
```

```
LD SP, 2800H
```

- ; inizializziamo lo stack pointer in modo
- che la 1^a locazione libera dello stack sia
- l'ultima disponibile della RAM
- ; ora inviamo la parola di modo all'8251
- ; ponendo $D_7D_6 = 1\Phi$ (fattore di baud rate pari a 64)
- ; $D_3D_2 = 11$ (lunghezza carattere di 8 bit)
- ; $D_4 = 1$ (parità abilitata)
- ; $D_5 = 1$ (parità pari)
- ; $D_7D_6 = 11$ (2 bit di stop)

```
LD A, FFH
```

```
OUT (10H), A
```

- ; ponendo, nelle mappe di I/O, le condizioni di indifferenza
- pari a zero l'indirizzo di programmazione dell'8251
- ; zero 10H

- ; ora inviamo la parola di comando con $D_7 = 1$ per
- porre l'8251 come trasmettitore

```
LD A, 01H
```

```
OUT (10H), A
```

- ; parola di programmazione controllore ϕ del CRC
- ; $D_7 = 0$ interruzioni disabilitate
- ; $D_6 = \phi$ timer

Soluzione identica a quella del compito precedente.

SOFTWARE

Cominciamo con il software di inizializzazione della scheda

```
ORG 0000
```

```
LD SP, 2800H
```

- ; inizializziamo lo stack pointer in modo
- che la 1^a locazione libera dello stack sia
- l'ultima disponibile della RAM
- ; ora inviamo la parola di modo all'8251
- ; ponendo $D_7D_6 = 10$ (fattore di baud rate pari a 64)
- ; $D_3D_2 = 11$ (lunghezza carattere di 8 bit)
- ; $D_4 = 1$ (parità abilitata)
- ; $D_5 = 1$ (parità pari)
- ; $D_7D_6 = 11$ (2 bit di stop)

```
LD A, FFH
```

```
OUT (10H), A
```

- ; ponendo, nelle mappe di I/O, le condizioni di indifferenza
- pari a zero e l'indirizzo di programmazione dell'8251
- zero 10H

- ; ora inviamo la parola di comando con $D_7 = 1$ per
- porre l'8251 come trasmettitore

```
LD A, 01H
```

```
OUT (10H), A
```

- ; parola di programmazione controllore ϕ del CRC
- ; $D_7 = 0$ interruzioni disabilitate
- ; $D_6 = 0$ timer

- ; D₃=0 periferice software
- ; D₂=1 segue costante di tempo
- ; D₁=1 non importa all'inizio
- ; D₀=1 parole di controllo

LD A, 27H

OUT(04H), A

LD A, 00H ; costante di tempo nulla

OUT(04H), A

; parola di programmazione controller 1 CRC

; D₇=1, interruzioni abilitate

; D₆=1, counter

; D₅=0, D₄=0, D₃=0 don't care

; D₂=1 segue costante di tempo

; D₁=1 non è importante


; D₀=1

LD A, D7H

OUT(11H), A

; costante di tempo 190/10 = ~~19~~ BEH

LD A, ~~19~~ BEH

OUT(11H), A  $\left\{ \begin{array}{l} \text{LD A, 02H} \quad ; \text{ parte bassa} \\ \text{OUT(11H), A} \quad ; \text{ vettore interruzione} \end{array} \right.$

LD HL, 0300H

; è l'indirizzo del sottoprogramma che gestisce

; l'interruzione generata dal CRC

LD (2002H), HL

; la tabella dei vettori di interruzione

; contiene solo l'indirizzo del sottoprogramma

; che gestisce il CRC, da ricordare che per

il CRC il contatore è dal CRC al momento

; D₇=1 non impatta all'interno

; D₀=1 parola di controllo

LD A, 27H

OUT(04H), A

LD A, 00H ; costante di tempo nulla

OUT(0AH), A

; parola di programmazione contiene 1 CRC

; D₇=1, interruzioni abilitate

; D₆=1, counter

; D₅=0, D₄=0, D₃=0 don't care

; D₂=1 segue costante di tempo

; D₁=1 non è importante

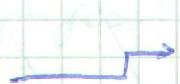
; D₀=1

LD A, D7H

OUT(11H), A

; costante di tempo 190/10 = ~~19~~ BEH

LD A, ~~19~~ BEH

OUT(11H), A  $\left\{ \begin{array}{l} \text{LD A, 02H} \quad ; \text{parte base} \\ \text{OUT(11H), A} \quad ; \text{veloce interruzione} \end{array} \right.$

LD HL, 0300H

; è l'indirizzo del sottoprogramma che gestisce

; l'interruzione generata dal CRC

LD (2002H), HL

; la tabella dei vettori di interruzione

; contiene solo l'indirizzo del sottoprogramma

; che gestisce il CRC, da ricordare che per

; il CRC il contatore 1 del CRC ci sono

; 2 bit D₂D₁ delle interrupt vector word

; riservati a 01

• IIR2 → $\left\{ \begin{array}{l} \text{LD A, 20H} \\ \text{LD I, A} \end{array} \right.$; parte alta in I

; impostiamo il modo 2

• EI

; abilitiamo le interruzioni

HALT

; restiamo in stato di attesa

ORG 0300H

LD B, 04H ; serve da contatore

; subroutine di gestione dell'interruzione

LD E, 00H

LOOP: LD A, 00H

OUT (03H), A

; selezioniamo il sensore da inviare

; attraverso il MUX dell'ADC

CALL RITARDO

; attendiamo circa 15 μ s perché la conversione

; venga completata

IN A, (02H)

; leggiamo il risultato della conversione

~~; ora lo convertiamo in BCD su 16 bit~~

~~ADD A, 00H~~

~~; l'addizione è fittizia, serve solo a permettere~~

~~; l'utilizzo dell'istruzione DAA~~

~~DAA~~

; dobbiamo ora convertirlo in gradi da 50 a 400

; le combinazioni da 00 a FF sono 256 mentre i

; gradi da 50 a 400 sono 351.

; $351/256 = 1,37$

; il dato in gradi sarebbe allora dato da

; N * 137/100 LDA, B
LD B, 89H EX AF, AF' ; selvo B

; 89H = 137/100

LD L, A

LD H, 00H

; spostato il risultato della conversione in HL

LOOP1: ADD HL, HL

DJNZ; LOOP1

; effettua la moltiplicazione attraverso un ciclo

; di somme

LD D, H

LD E, L

; spostato il risultato in DE per iniziare la divisione

LD H, 00H

LD L, 00H

; HL dovrà contenere il risultato

LD A, 16H

; A farà da contatore

; BC conterrà il divisore 64H = 100/10

LD C, 64H

DIV RLE

RLD

ADC HL, HL

; con queste 3 istruzioni il blocco di 32 bit formato

; da HL e DE viene shiftato a sinistra di una posizione

; abbiamo "abbassato una cifra"

SAL HL, BC

ADD HL, BC

; altrimenti: a partire da HL

DIV1 CCF

; se siamo arrivati a questo punto attraverso il salto

; il carry era a zero, ma nel risultato parziale va messo

; 1 e viceversa per cui il carry va complementato

; in modo che, quando andremo a motore DE, tale bit

; del risultato finisca in coda

DECA

JR NZ, DVO

EX DE; HL

; metti il risultato in DE HL

ADC HL, HL

; prendi l'ultimo bit del risultato che era nel carry

; a questo punto HL contiene il risultato della

; divisione e DE contiene il resto

; decidiamo di trascurare il resto e

; trasformiamo HL in un dato BCD

~~ADD HL~~

LD A, L

ADD A, 00H

; serve solo a poter utilizzare DAA

DAA

LD L, A

; ora il carry contiene un eventuale riporto

~~ADC A, 00H~~

LD A, H

ADC A, 00H

~~MOV~~
EX AF, AF'

LD B, A

; recupero il contatore

~~INC~~ C

~~; incremento C in modo che al~~

~~; ciclo successivo si incrementa il numero del~~

~~; sensore che smistare all'ADC~~

~~LD A, C~~

ADD A, 04H

LD C, A

; sommando A e C otteniamo l'indirizzo di

; I/O del display giusto

~~LD A, L~~

~~OUT (C), A~~

~~LD A, H~~

; ora modifichiamo il nibble superiore dell'indirizzo

; 20 per cambiare il display più significativo

LD A, H

CP 0FH

JP NZ, AVANTI1

OR 00010000B

; e alle cifre superiori e 1 modifica il nibble

; superiore dell'indirizzo in modo che contenga 1

AVANTI1 LD RCP 02H

AVANTI2 JP NZ, AVANTI2

OR 00100000B

AVANTI2 CP 03H

OR 0011 0000B

AVANTI3 CP 04H

JP NZ, AVANTI G

OPOR 01000000B

AVANTI G LD C, A

; composto all'indirizzo giusto si invia
; il dato al display

~~OUT (C), A~~ LDA, L

OUT (C), A

; inviamo ora il dato all'8251

LD A, 03H

OUT (10H), A

; ~~si continua a provare se bit 7 = 0~~

; facciamo uno zero in uscita al DTR

LOOP2 IN A, (10H)

; attendiamo che DSR vada a zero

BIT 7, A

JPNZ, LOOP2; ~~si continua~~

; si continua a provare se bit 7 = 0

LD A, 2FH

; se si, si attende RTS

~~LD A, 0~~ OUT (10H), A

LOOP3 IN A, (10H)

BIT 0, A

; si controlla che TXRDY = 1 per verificare che il
buffer di trasmissione è vuoto

JP Z, ~~LOOP3~~ LOOP3

LD A, 1

; si ripete il procedimento per il 2° byte

LD A, 03H

OUT (10H), A

LOOP4 IN A, (10H)

BIT 7, A

JPNZ, LOOP4

LD A, 21H

OUT (10H), A

LOOP5 IN A, (10H)

BIT 0, A

JPZ, LOOP5

LD A, H

OUT (09H), A

~~LD~~ LD A, C

SUB 04H

~~AND~~ AND 00011111B

LD C, A

; si ripristina in C il valore corrispondente al

; prossimo sensore da inviare ~~il valore~~ all'ADC
INC C

DJNZ, LOOP

; si continua fino a che tutti i sensori sono stati ~~chiesti~~

; richiesti

EI

; abbiamo le interruzioni

RET

; e usciamo.