

PICLOCK

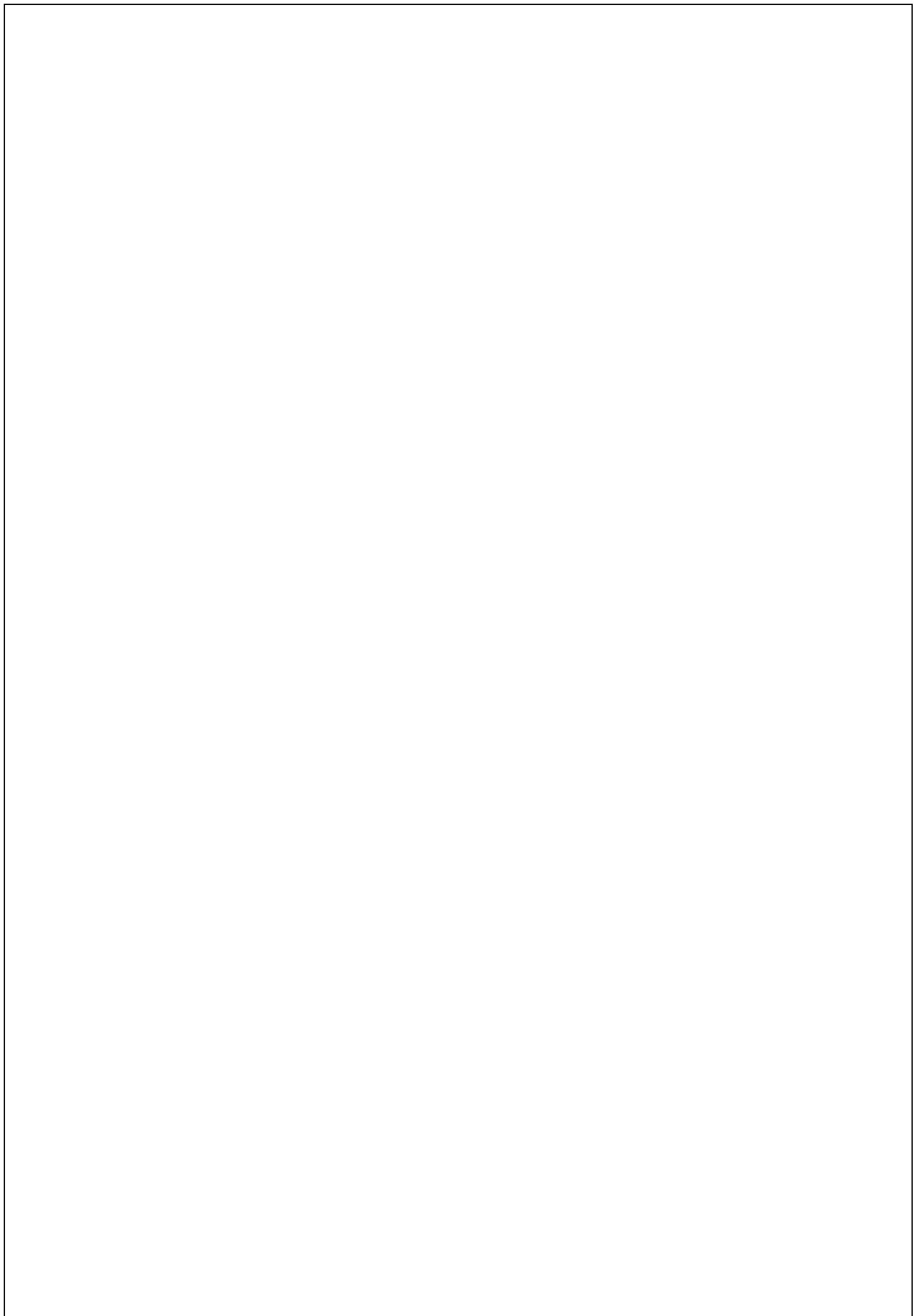
Classe 4 EA indirizzo elettronica ed elettrotecnica

Ivancich Stefano

Tecnologie e progettazione di sistemi elettrici ed elettronici

Istituto tecnico C. Zuccante – Mestre

A.S. 2013/2014 progetto n° 2

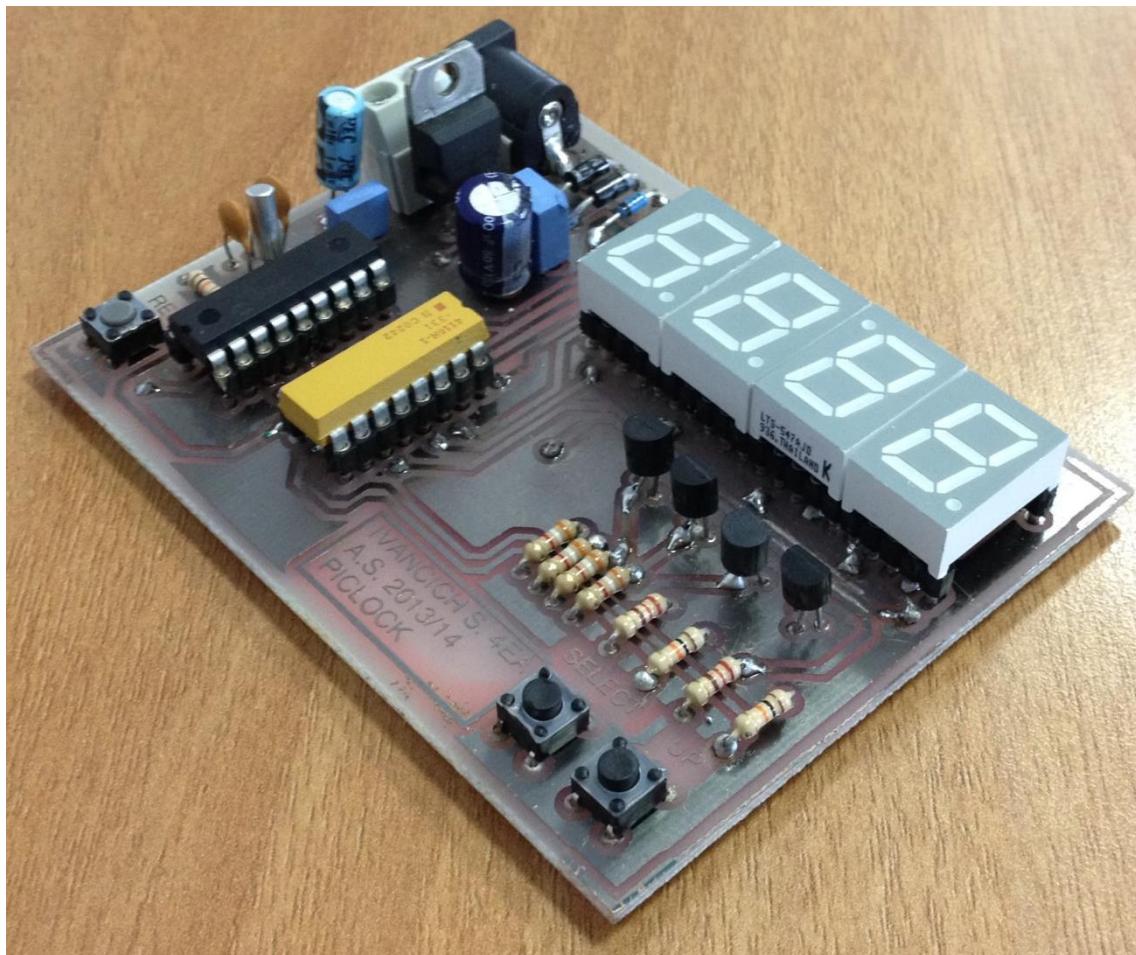


SOMMARIO

Di seguito si illustra il progetto di un orologio digitale.

Il progetto è molto utile per capire il funzionamento delle dinamiche interne di un circuito gestito da un microcontrollore, il che diventa utilissimo per moltissime applicazioni dove la logica cablata non è sufficiente a soddisfare le specifiche di progetto.

Pur essendo un progetto artigianale e scolastico, l'orologio non è molto ingombrante (7 cm di larghezza x 9cm di lunghezza).



INDICE

| | |
|---------------------------------------|----|
| SOMMARIO..... | 1 |
| INTRODUZIONE..... | 7 |
| 1. SCHEMA ELETTRICO | 9 |
| 1.1. ALIMENTAZIONE | 9 |
| 1.2. OROLOGIO..... | 10 |
| 2. FIRMWARE..... | 11 |
| 2.1 FLOW CHART ALIMENTAZIONE | 11 |
| 2.2 FLOW CHART MICROCONTROLLORE | 13 |
| 3. DIMENSIONAMENTO COMPONENTI | 14 |
| 3.1. ALIMENTAZIONE | 14 |
| 3.2. OROLOGIO..... | 15 |
| 4. LISTA COMPONENTI..... | 16 |
| 5. CONCLUSIONI..... | 17 |

TAVOLE

| | |
|--|---------------|
| DISEGNI TECNICI | 19 |
| 2014-1.1 schema elettrico alimentazione..... | foglio 1 di 5 |
| 2014-1.1 schema elettrico orologio..... | foglio 2 di 5 |
| 2014-1.1 serigrafia master | foglio 3 di 5 |
| 2014-1.1 master bottom..... | foglio 4 di 5 |
| 2014-1.1 master top..... | foglio 5 di 5 |
| | |
| FIRMWARE | 31 |
| | |
| DATASHEETS | 41 |
| AN5090 | |
| BC337 npn transistor | |
| BZX79C4V7 Zener diode | |
| Chrystal 32.768Hz | |
| LM7805 voltage regulators | |

INTRODUZIONE

Lo scopo del progetto è di realizzare un circuito in grado di mostrare l'ora in ore e minuti, di impostarla tramite dei pulsanti e di gestire il risparmio energetico a seconda che venga alimentato da rete o da batteria. Questo progetto non è quindi solo un puro e semplice esercizio, ma si tratta di un progetto con un'applicazione pratica e funzionale.

L'orologio ha le seguenti specifiche:

1. Display a 4 cifre con due punti lampeggianti che separano le ore dai minuti.
2. Due tipi di alimentazione:
 - 2.1. Da rete elettrica: serve un trasformatore/raddrizzatore esterno con tensione raddrizzata e livellata 11 - 15 V.
 - 2.2. Con batteria da 7 - 9 V DC.
3. Tre pulsanti di controllo: reset, select e set.
4. Componenti del magazzino della scuola C. Zuccante laboratorio OEN2.

1. SCHEMA ELETTRICO

Lo schema a blocchi, illustrato nella figura 1.1, suddivide in due parti principali il progetto: l'alimentazione e l'orologio.

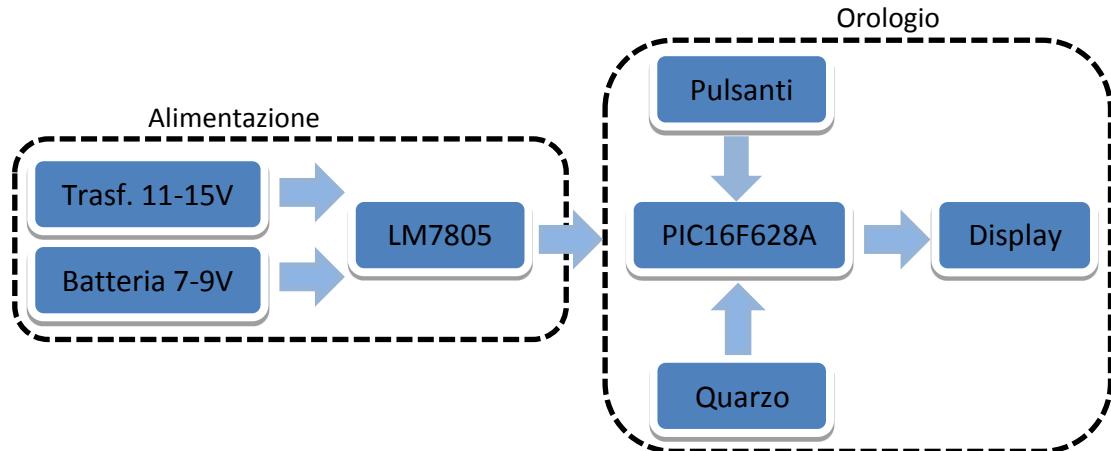


Figura 1.1

1.1. ALIMENTAZIONE

Nel primo blocco, illustrato nella figura 1.1.1, la tensione in ingresso al connettore a due poli X1-1 e X1-2 è continua di ampiezza compresa tra +7V e +9V, invece la tensione in ingresso al connettore jack J1 è anch'essa continua compresa tra +11V e +15V (si è utilizzato un trasformatore che trasforma i 230 V AC in 12 V DC).

Sono stati utilizzati due diodi 1N4007 in modo tale che in ingresso allo stabilizzatore LM7805 sia applicata la tensione del trasformatore se presente, altrimenti viene applicata quella della batteria, questo per far risparmiare energia alla batteria.

Al pin RA4 del microcontrollore viene inviata la tensione d'alimentazione del trasformatore tramite un apposito diodo Zener (BZX79C4V7) per segnalare al PIC se questa alimentazione è o meno presente per gestire la modalità a risparmio energetico se è presente solo l'alimentazione a batteria.

L'integrato stabilizzatore LM7805 fornisce in uscita 5V utilizzati per alimentare l'intero circuito.

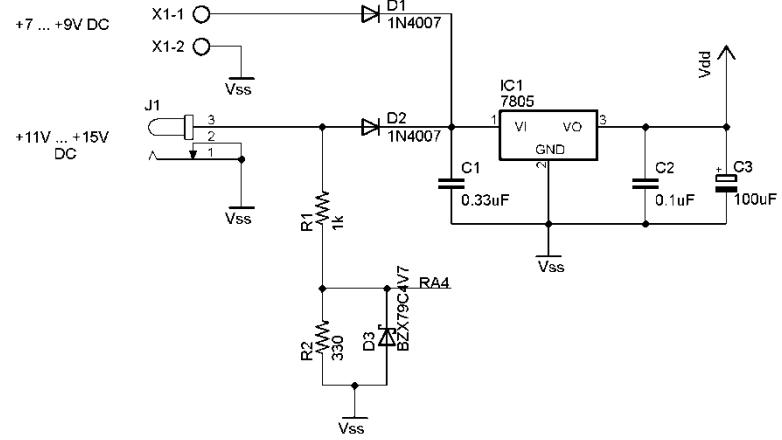


Figura 1.1.1

1.2. OROLOGIO

Il primo pulsante è connesso al pin RA5, il master clear, questo ha funzione di RESET, ovvero se premuto riavvia il sistema e tutti i led del display si accendono e si spengono in modo intermittente per segnalare che si deve impostare l'ora.

Il secondo pulsante è connesso al pin RB0 ed ha funzione di SELECT, se premuto per 4 volte consecutive, seleziona quale cifra del display deve essere regolata, quando viene premuto per la quinta volta fa partire l'orologio, funziona anche nella modalità basso consumo.

Il terzo pulsante è connesso al pin RB1 ed ha funzione di SET, premuto successivamente, permette di impostare la cifra desiderata.

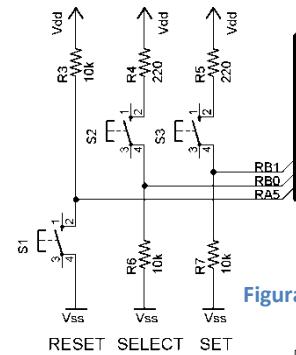


Figura 1.2.1

L'oscillatore al quarzo di frequenza 32.768Hz è connesso ai pin RB6 e RB7, fornisce il clock al Timer1, periferica interna del PIC16F628A.

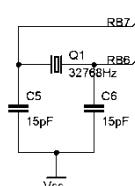


Figura 1.2.2

Sono stati utilizzati 4 display 7 segmenti a catodo comune, uno di essi, le decine dei minuti, è girato così da posizionare il suo decimal point in alto per separare le ore dai minuti. Non è stato utilizzato un decoder BCD/7 segmenti, quindi la decodifica viene gestita via software, dunque è stato utilizzato un integrato contenente 8 resistenze, questo per risparmiare spazio e collegamenti nel pcb. Inoltre per risparmiare energia i display sono multiplexati, ovvero è acceso un solo display alla volta, questo tramite 4 transistor BJT BC337 col collettore connesso al catodo comune di ciascun display, l'emettitore connesso a massa e la base comandata dai pin RA0, RA1, RA2, RA3 del microcontrollore PIC16F628A.

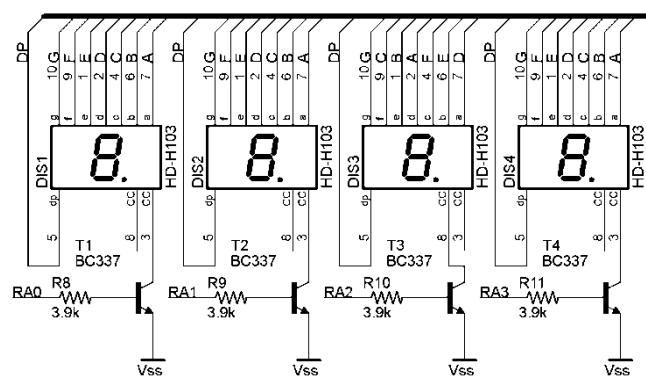


Figura 1.2.3

Per il funzionamento del PIC16F628A si è optato per l'oscillatore interno a 4 MHz per risparmiare componenti, il ciclo di una istruzione dura 1 microsecondo. Tra i pin di VDD e VSS è stato aggiunto un condensatore elettronitico di disaccoppiamento da 1uF.

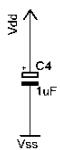


Figura 1.2.4

Riassumendo nella tabella 1.2.2 la definizione dei pin:

| Pin | IN/OUT | Funzione |
|-----|--------|---|
| RA0 | OUTPUT | BJT di comando display unita dei minuti |
| RA1 | OUTPUT | BJT di comando display decine dei minuti |
| RA2 | OUTPUT | BJT di comando display unita delle ore |
| RA3 | OUTPUT | BJT di comando display decine delle ore |
| RA4 | INPUT | Funzionamento da rete (HIGH) / batteria (LOW) |
| RA5 | INPUT | Master clear esterno, pulsante RESET |
| RA6 | OUTPUT | Segmento "A" dei display a 7 segmenti |
| RA7 | OUTPUT | Segmento "punto" dei display a 7 segmenti |

| | | |
|-----|--------|---------------------------------------|
| RB0 | INPUT | Pulsante SELECT |
| | OUTPUT | Segmento "G" dei display a 7 segmenti |
| RB1 | INPUT | Pulsante SET |
| | OUTPUT | Segmento "F" dei display a 7 segmenti |
| RB2 | OUTPUT | Segmento "E" dei display a 7 segmenti |
| RB3 | OUTPUT | Segmento "D" dei display a 7 segmenti |
| RB1 | OUTPUT | Segmento "C" dei display a 7 segmenti |
| RB5 | OUTPUT | Segmento "B" dei display a 7 segmenti |
| RB6 | INPUT | Timer1 Oscillatore Output |
| RB7 | INPUT | Timer1 Oscillatore Input |

2. FIRMWARE

2.1 FLOW CHART ALIMENTAZIONE

Il diagramma di flusso relativo all'alimentazione del progetto (figura 2.1) si suddivide principalmente in due parti:

alimentazione da rete e da batteria.

Per prima cosa viene controllato il pulsante SELECT: se non è premuto lascia il display nella modalità blinking. Se invece è premuto permette di impostare l'ora tramite il pulsante SET, l'ordine di impostazione dell'ora è decine delle ore, unità delle ore, decine dei minuti e unità dei minuti.

A questo punto parte il conteggio dei secondi tramite il Timer1. Adesso viene controllato il tipo di alimentazione:

1. ALIMENTAZIONE DA RETE: Se il circuito è alimentato tramite la rete elettrica non c'è bisogno di risparmiare energia quindi si può selezionare tramite il pulsante SELECT se spegnere i display o lasciarli sempre attivi
2. ALIMENTAZIONE TRAMITE BATTERIA: Se il pulsante SELECT non è premuto il microcontrollore si trova in modalità basso consumo (SLEEP). Se invece è premuto, vengono eseguiti due blink e vengono spenti i display. Si passa così in modalità basso consumo. Successivamente si controlla se c'è una richiesta di interrupt, si esce dalla modalità basso consumo e si esegue il programma principale descritto dal flow chart in figura 2.2.

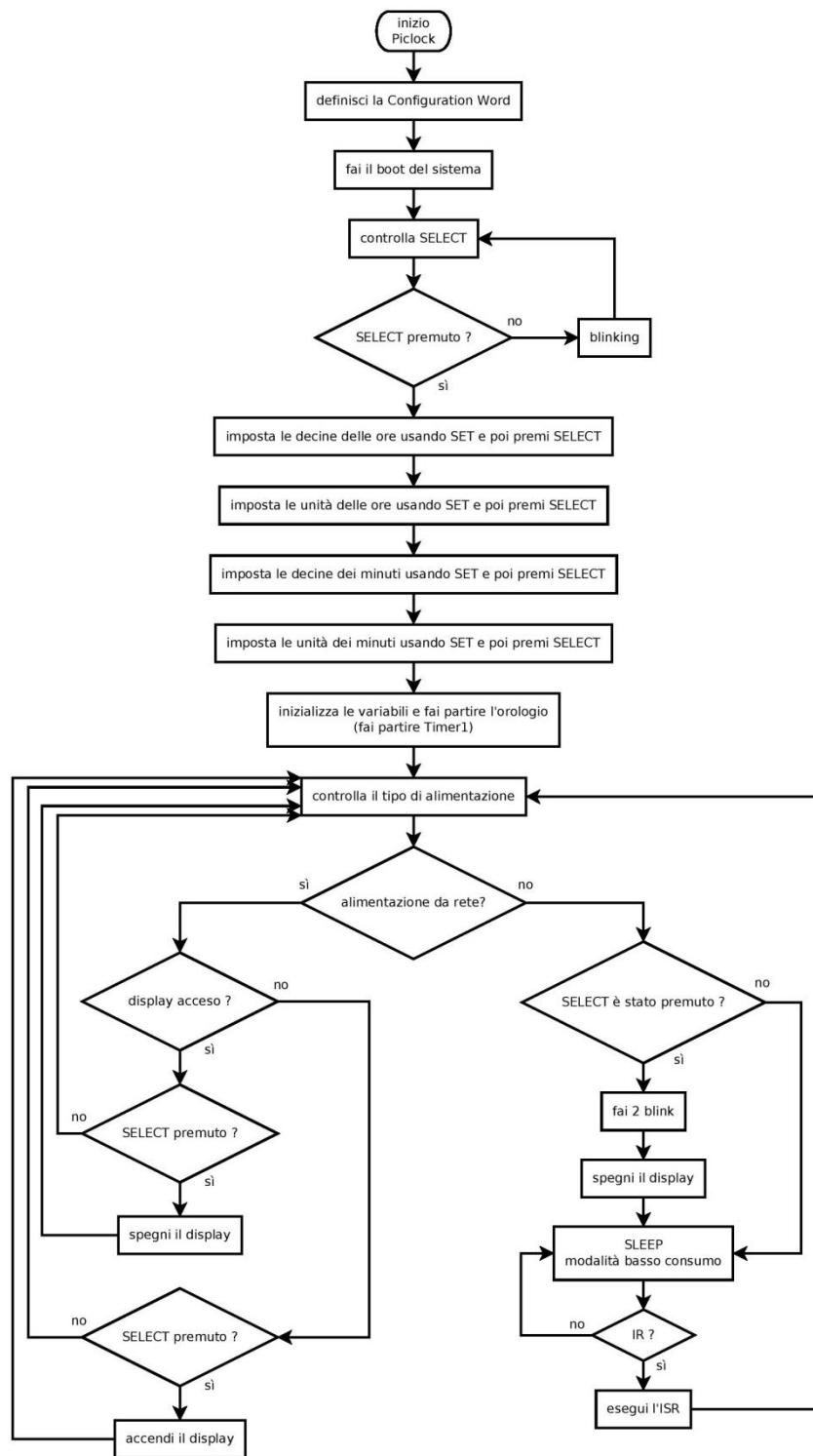


Figura 2.1.1: flow chart alimentazione

2.2 FLOW CHART MICROCONTROLLORE

Il diagramma di flusso in figura 2.2 descrive il funzionamento dell'ISR (interrupt service routine).

Per prima cosa viene testato il Timer1, se il Timer non si è azzerato maschera l'interruzione RB0INT, "manda" GIE a 1 abilitando così le interruzioni, accende i display per otto secondi, azzerà la richiesta di interruzione RB0INT e termina l'ISR. Se Timer1 si è azzerato e TMR1IF = 1, per prima cosa salva il contesto del microcontrollore, poi azzerà la richiesta di interruzione di Timer1 e comincia il ciclo per il conteggio dei secondi, dei minuti e delle ore.

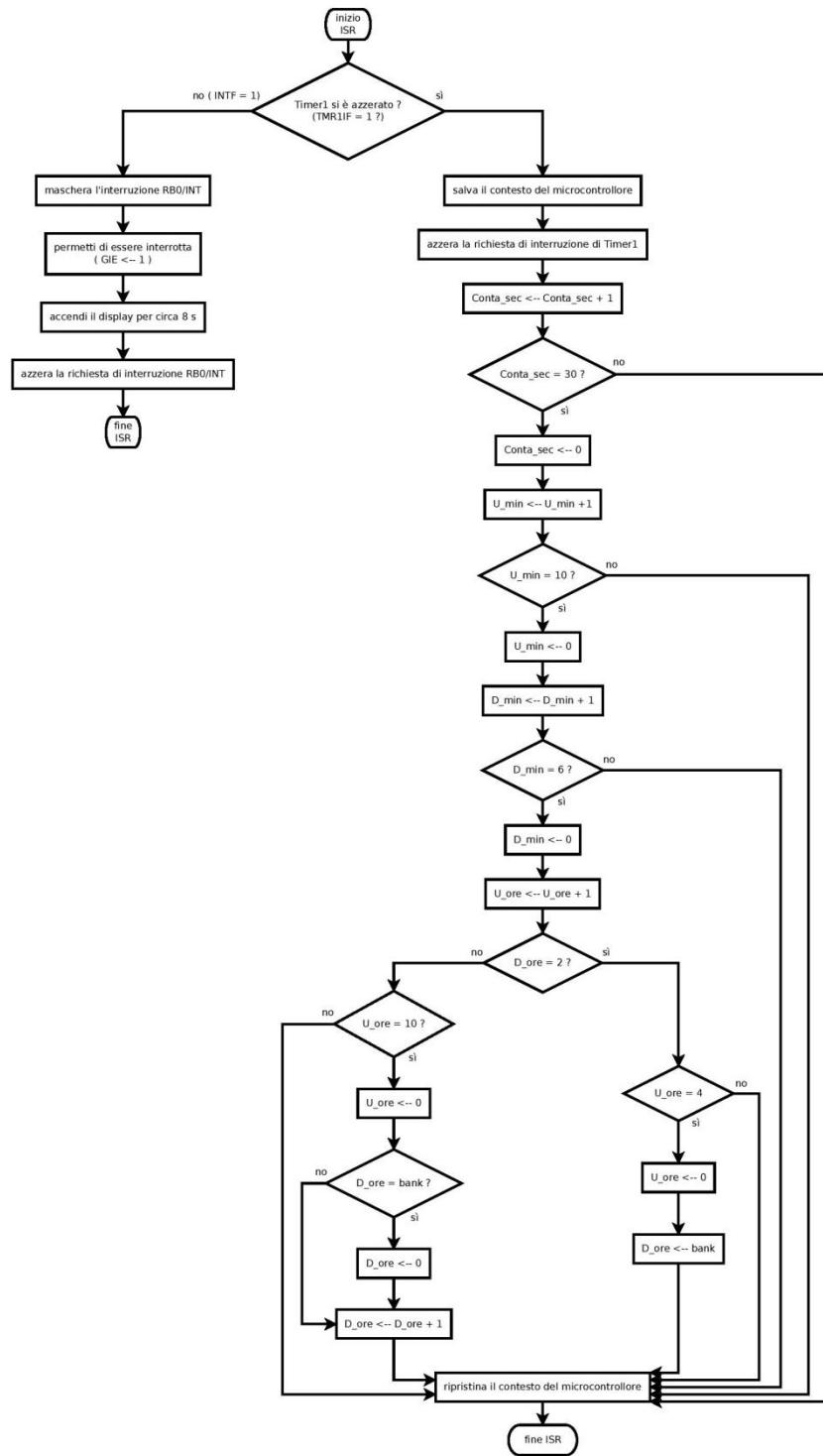
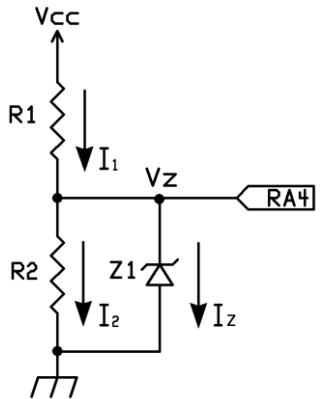


Figura 2.2.1: flow chart microcontrollore

3. DIMENSIONAMENTO COMPONENTI

3.1. ALIMENTAZIONE

RESISTENZE DEL DIODO ZENER



$$V_{min} \leq V \leq V_{max}$$

$$V_{Zmin} \leq V_Z \leq V_{Zmax}$$

$$I_{Zmin} \leq I_Z \leq I_{Zmax}$$

$$I_2 = \frac{V_Z}{R_2} \quad I_{2min} \leq I_2 \leq I_{2max}$$

$$I_Z = I_1 - I_2 = \frac{V - V_Z}{R_1} - I_2$$

$$\text{Valore più piccolo di } I_Z: V = V_{min} \quad V_Z = V_{Zmax} \quad I_2 = I_{2max}$$

Per garantire che lo zener sia in conduzione occorre:

$$\frac{V_{min} - V_{Zmax}}{R_1} - I_{2max} > I_{Zmin} \quad R_1 < \frac{V_{min} - V_{Zmax}}{I_{Zmin} + I_{2max}} = R_{max}$$

$$\text{Valore più grande di } I_Z: V = V_{max} \quad V_Z = V_{Zmin} \quad I_2 = I_{2min}$$

Per garantire che lo zener non si danneggi occorre:

$$\frac{V_{max} - V_{Zmin}}{R_1} - I_{2min} < I_{Zmax} \quad R_1 > \frac{V_{max} - V_{Zmin}}{I_{Zmax} + I_{2min}} = R_{min}$$

$$V_{min} = 11V \quad V_{max} = 15V \quad \text{scelte di progetto}$$

Diodo zener: BZX9C4V7 (vedi datasheet)

$$V_{Zmin} = 4,4V \quad V_{Zmax} = 5,0V \quad I_{Zmin} = 5,0mA$$

Potenza dissipata: $P_{max} = 500mW$ $\vartheta 25^\circ C$

$$4,0mW/^\circ C$$

$$\Delta T = 45^\circ C - 25^\circ C = 20^\circ C \quad \Delta P = \frac{4,0mW}{^\circ C} * 20^\circ C = 80mW$$

Potenza massima dissipata:

$$P_{max} = 500mW - 80mW = 420mW$$

$$I_{Zmax} < \frac{P_{max}}{V_Z} = \frac{420mW}{4,7V} = 89,36mA$$

Si sceglie: $I_{Zmax} = 50\text{mA}$

Per avere I_2 con valore di qualche mA si sceglie:

$$R_2 = 1K\Omega \quad I_{2min} = \frac{V_{Zmin}}{R_2} = \frac{4,4V}{1K\Omega} = 4,400mA$$

$$I_{2max} = \frac{V_{Zmax}}{R_2} = \frac{5,0V}{1K\Omega} = 5,000mA$$

$$R_{max} = \frac{11V - 5,0V}{5,0mA + 5,00mA} = 600,0\Omega$$

$$R_{min} = \frac{15V - 4,4V}{50mA + 4,4mA} = 194,9\Omega$$

$R_1 = 330\Omega$ Scelgo un valore compreso tra R_{max} e R_{min} .

CONDENSATORI

$C_1=330\text{nF}$ e $C_2=100\text{nF}$ sono stati scelti consultando il datasheets del regolatore di tensione.

100 microF per stabilizzare la tensione di alimentazione durante la commutazione dell'alimentazione da rete a batteria e viceversa.

3.2. OROLOGIO

RESISTENZE DEI TRANSISTOR

8 Resistenze integrate dual in-line package – DIP 330Ω valore nominale

$$V_F = 2,0V \quad V_{CESat} = 0,2V \quad V_H = 5,0V$$

$$I_F = \frac{V_H - V_F - V_{CESat}}{R} = \frac{(5,0 - 2,0 - 0,2)V}{330\Omega} = 8,5mA \quad (\text{accettabile})$$

BC337-25 $h_{FEmin} = 160$ (vedi datasheet)

Per garantire la saturazione del transistor:

$$I_B > \frac{I_F}{h_{FEmin}} = \frac{8,5mA}{160} = 53,13\mu A = I_{Bmin}$$

$$V_{BEsat} = 0,8V \quad I_B = 20I_{Bmin} = 1,063mA$$

Calcolo il valore della resistenza di base del transistor:

$$R_B = \frac{V_H - V_{BEsat}}{I_B} = \frac{(5 - 0,8)V}{1,063mA} = 3,951K\Omega$$

$R_B = 3,9K\Omega$ Valore appartenente alla serie E12 normalizzata.

RESISTENZE DEI PULSANTI

$R_3=R_6=R_7=10k\Omega$ scelte consultando la documentazione Microchip AN590.

$R_4=R_5=220\Omega$ scelte consultando la documentazione Microchip AN590: 820 ohm, valore ridotto per adattarlo alle esigenze del progetto Piclock.

QUARZO

$$C_L = \frac{C_2 C_3}{C_2 + C_3} + C_S \quad (\text{formula reperibile dal datasheet del costruttore del quarzo})$$

Capacità parassita $C_S (2 \div 5)pF$ (range di valori fornito dal costruttore)

$$C_2 = C_3 = C$$

Load capacitance $C_L = 12,5pF$

$$C_L = \frac{C}{2} + C_S \quad C = 2(C_L - C_S)$$

$$C_{min} = 2 * (12,5 - 5)pF = 15pF$$

$$C_{max} = 2 * (12,5 - 2)pF = 21pF$$

$$15pF \ll C \ll 21pF$$

$C = 15pF$ valore suggerito nei datasheet del pic 16F628A.

4. LISTA COMPONENTI

| MATERIALE | SIGLA | Q/TA' |
|-----------------------------|---------------|-------|
| MICROCONTROLLORE | PIC16F628A | 1 |
| PULSANTE N.A. | | 3 |
| QUARZO | f = 32768 Hz | 1 |
| TRANSISTOR BJT NPN | BC337-25 | 4 |
| DISPLAY 7 SEGMENTI | CATODO COMUNE | 4 |
| CONDENSATORI CERAMICI | 15pF | 2 |
| PACCHETTO RESISTIVO | 330Ω 8+8 | 1 |
| CONNETTORE 2 POLI | | 1 |
| REGOLATORE DI TENSIONE | LM 7805 | 1 |
| CONDENSATORE PLASTICO | 0,33μF | 1 |
| CONDENSATORE PLASTICO | 0,1μF | 1 |
| DIODO ZENER | BZX79C4V7 | 1 |
| DIODO | 1N4007 | 2 |
| CONNETTORE JACK | | 1 |
| CONDENSATORE ELETTRROLITICO | 1μF | 1 |
| TULIPANO 10 PIN | | 8 |

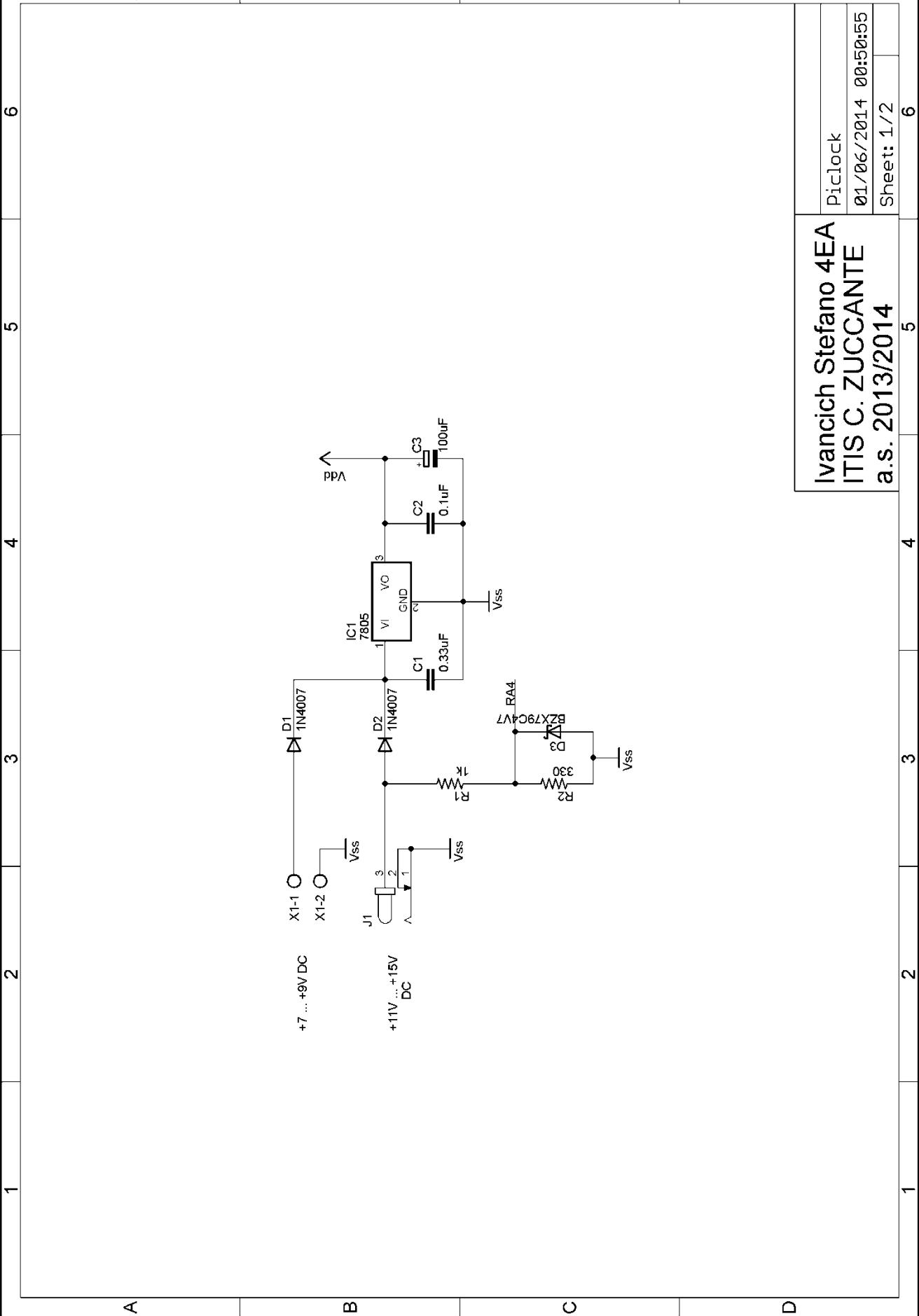
5. CONCLUSIONI

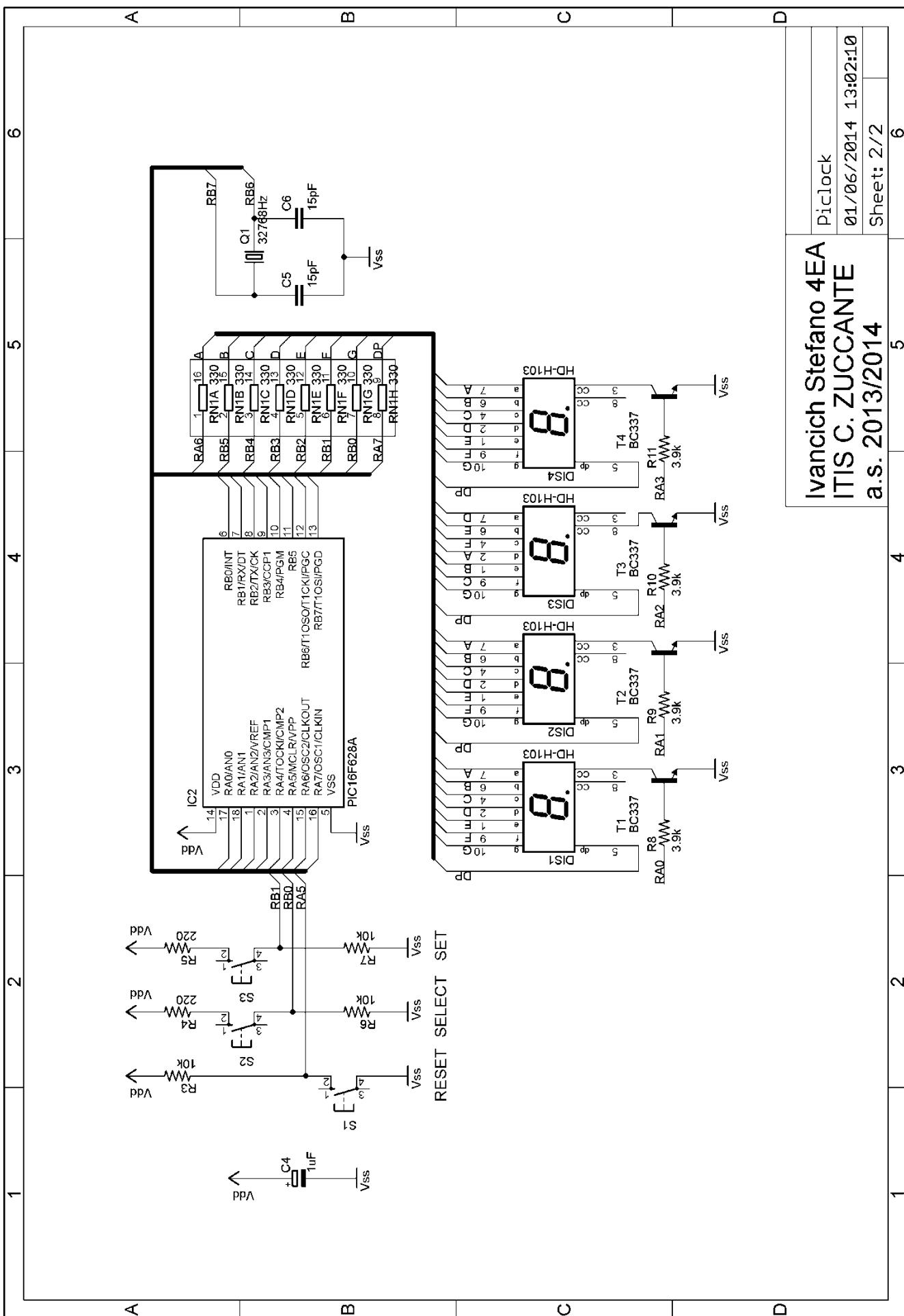
Il progetto è stato concluso con successo, anche se non si era partiti nel modo migliore, visto la scarsità di componenti reperibili nel magazzino del laboratorio OEN2 dell'istituto.

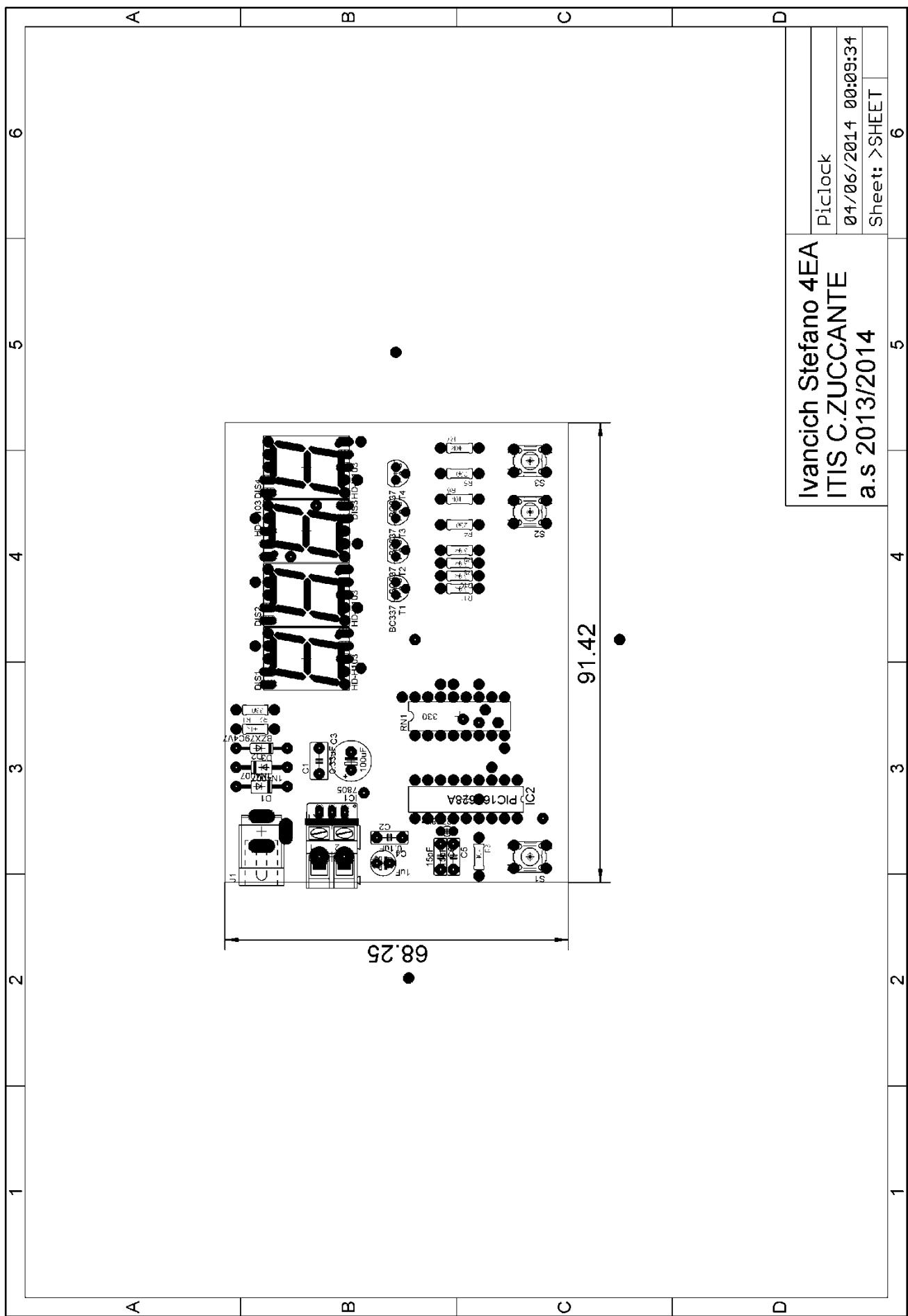
Sicuramente per questo tipo di progetto e per la larga memoria di programma si poteva optare per la programmazione in C del PIC16F628A.

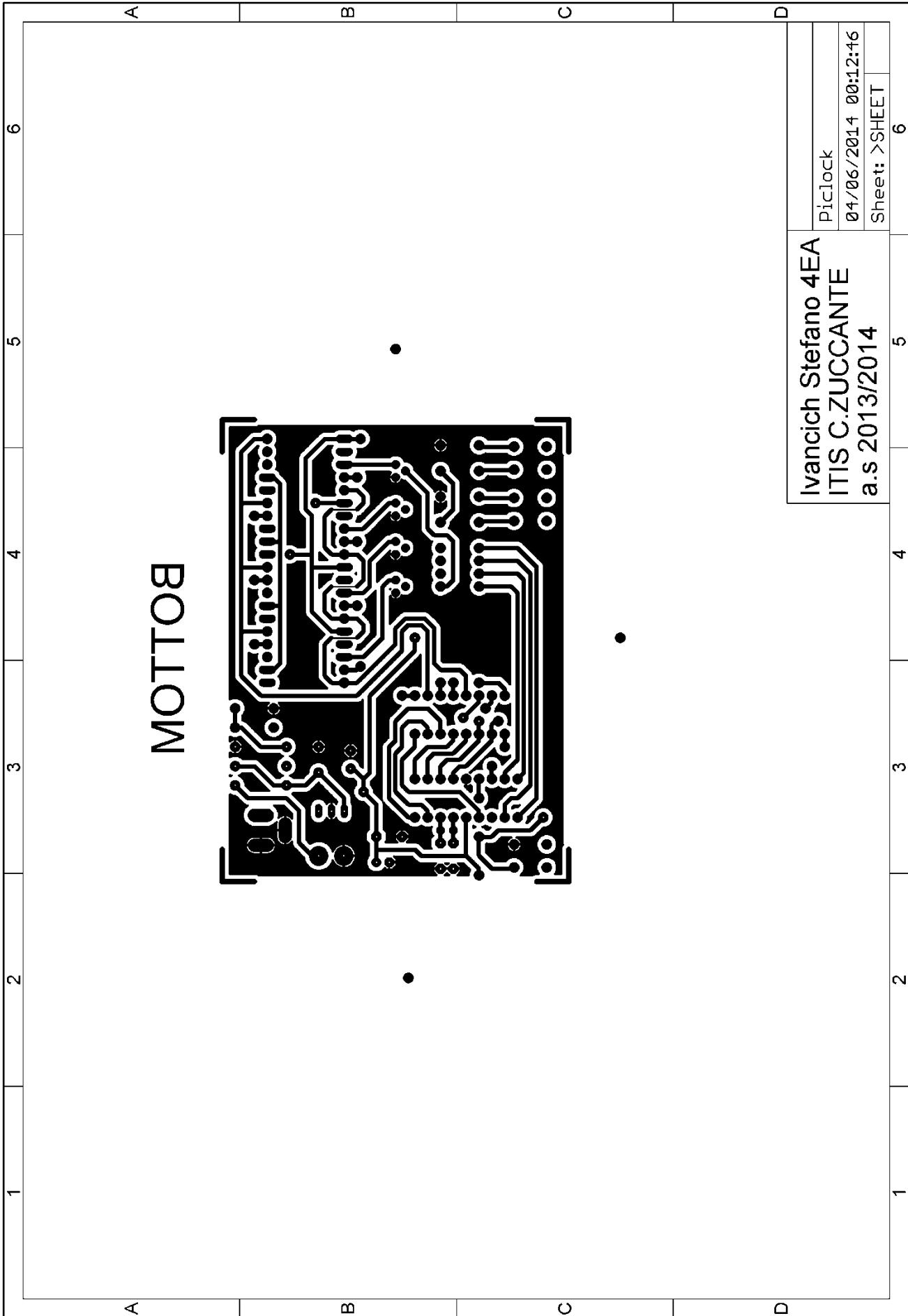
Per diminuire le dimensioni del PCB si potevano utilizzare componenti SMD, non presenti però nel magazzino della scuola.

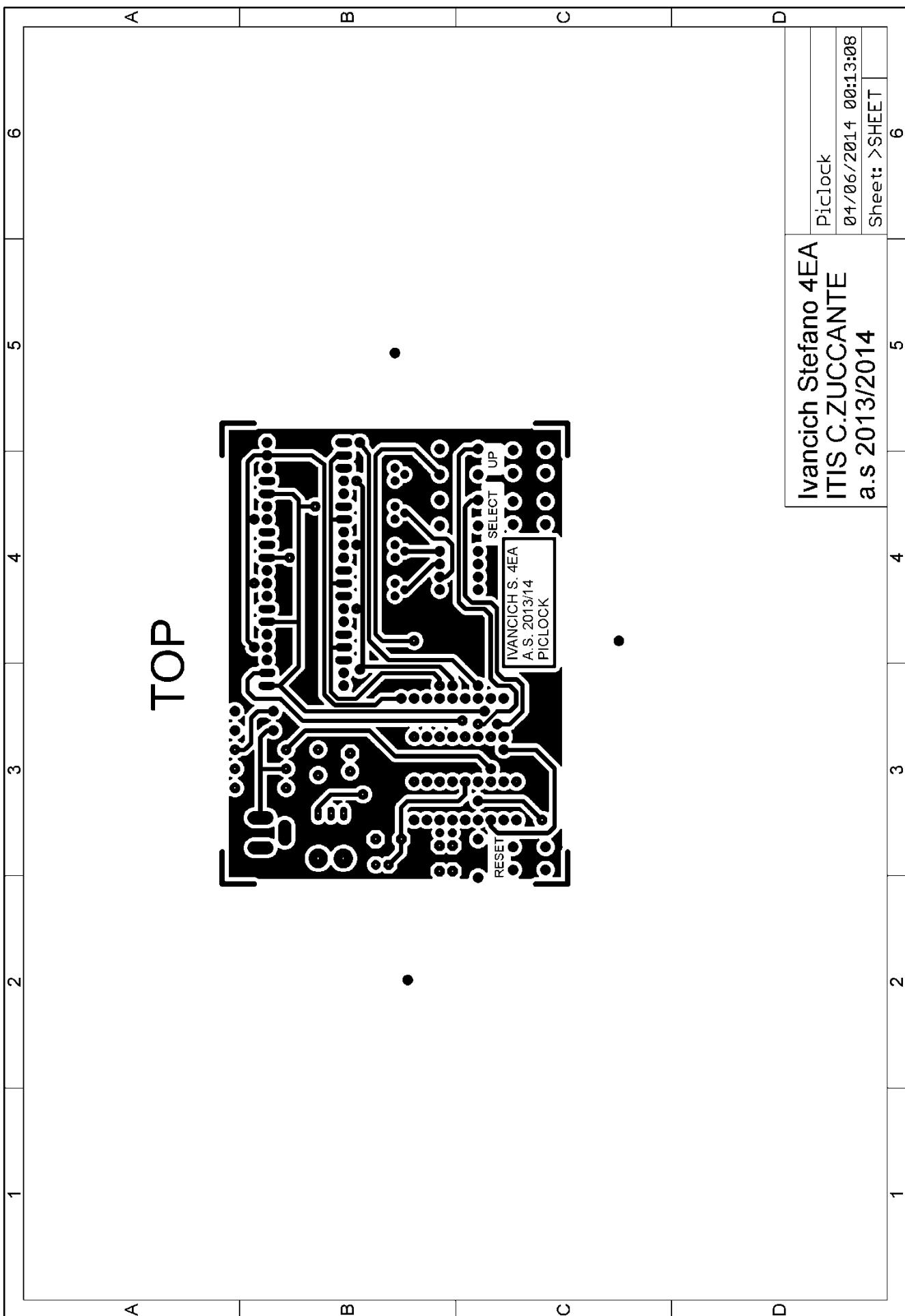
DISEGNI TECNICI











FIRMWARE


```

; IT Zuccante - Mestre - A.S. 2013/2014
; Indirizzo Elettronica ed Elettrotecnica - Articolazione Elettronica - classe 4 EA
; TECNOLOGIE E PROGETTAZIONE DI SISTEMI ELETTRICI ED ELETTRONICI - E. Minosso, R. Bardelle
;
; 26 maggio 2014
;
; Progetto " PICLOCK "
;
;*****=====
;
;=====
processor 16F628A
#include P16F628A.INC

; imposta la CPU - Configuration Word
;
; CW<13> = CP      = 1   Code Protection OFF          _CP_OFF = 11 1111 1111 1111
; CW<8>  = CPD     = 1   Data Code Protection OFF      _CPD_OFF = 11 1111 1111 1111
; CW<7>  = LVP      = 0   Low-Voltage Programming disabled _LVP_OFF = 11 1111 0111 1111
; CW<6>  = BOREN    = 0   Brown-Out Reset disabled       _BOREN_OFF = 11 1111 1011 1111
;
; CW<5>  = MCLRE    = 1   pin RA5/MCLR/Vpp function is MCLR _MCLRE_ON = 11 1111 1111 1111
;
; CW<3>  = PWRTE    = 0   Power-up Timer enabled        _PWRTE_ON = 11 1111 1111 0111
; CW<2>  = WDTE     = 0   Watchdog Timer disabled       _WDTE_OFF = 11 1111 1111 1011
; CW<4,1-0> = FOSC<2:0> = 100 INTOSC oscillator:           _INTOSC_OSC_NOCLKOUT = 11 1111 1111 1100
;                               I/O function on RA6/OSC2/CLKOUT RA7/OSC1/CLKIN pins
;
; Configuration Word Register = 1x xxx1 0011 0000
; Configuration Word Register = 2130h
;
_config _CP_OFF & _CPD_OFF & _LVP_OFF & _BOREN_OFF & _MCLRE_ON & _PWRTE_ON & _WDTE_OFF & _INTOSC_OSC_NOCLKOUT
 radix DEC

; COSTANTI =====
Blank equ 10 ; utilizzata nella decodifica BCD/7seg per tenere spenti tutti i led del display

; VARIABILI =====
cblock 0x20 ; definisce gli indirizzi delle variabili seguenti partendo dall'indirizzo 20h

U_min ; unità dei minuti --> display n° 0
D_min ; decine dei minuti --> display n° 1
U_ore ; unità delle ore --> display n° 2
D_ore ; decine dedelle ore --> display n° 3
Due_punti ; memorizza l'accensione e lo spegnimento dei due punti --> display n° 1 e n° 2
Conta_sec ; contatore che va da 0 a 29 --> si incrementa ogni 2 secondi di tempo, per cui quando arriva a 30 significa
; che sono passati 30 x 2 secondi = 60 secondi
Contatore1 ; variabile contatore
Contatore2 ; variabile contatore
Acceso ; variabile booleana: Acceso = 1 <-> display acceso; Acceso = 0 <-> display spento
RB0_INT ; variabile booleana: RB0 INT = 1 <-> eseguita l'ISR dell'interruzione RB0/INT
; RB0_INT = 0 <-> non è stata eseguita l'ISR dell'interruzione RB0/INT

; INTERRUPT SERVICE ROUTINE
STATUS_temp ; salva temporaneamente il contesto del uC (registro STATUS)
endc ; fine della direttiva cblock
W_temp equ 0x70 ; salva temporaneamente il contesto del uC (W deve essere salvato negli
; ultimi 16 GPRs del banco 00, che sono replicati anche negli altri tre
; banchi di memoria, così per accedere a W_temp non occorre modificare
; RPL ed RPO del registro STATUS)

; INIZIO PROGRAMMA =====
org 0
GOTO boot ; avvia il programma principale

; gestore dell'interruzione =====
org 4
BTFS S PIR1, TMRIIF ; SE TMRIIF = 1 esegui l'ISR "rtc" che aggiorna i secondi, i minuti e le ore
GOTO wake_up ; ALTRIMENTI esegui l'ISR "wake-up" che accende nuovamente il display

; ISR rtc (real-time clock) =====
; salva il contesto del uC
rtc MOVWF W_temp ; salva il contenuto di W per poter usare subito W nell'ISR
SWAPP STATUS, W ; copia temporaneamente in W i nibble scambiati di STATUS
BCF STATUS, RP1 ; seleziona
BCF STATUS, RP0 ; il banco 00
MOVWF STATUS_temp ; salva i nibble scambiati di STATUS

BCF PIR1, TMRIIF ; azzerza il flag di interruzione di Timer1

; aggiorna il contatore dei secondi
INCFS Conta_sec, F ; Conta_sec = Conta_sec + 1
MOVLW 30 ; controllo
XORWF Conta_sec, W ; se Conta_sec = 30
BTFS STATUS, Z ; SE Conta_sec = 30 ALLORA azzerà Conta_sec e incrementa U_min
GOTO fine_ISR ; ALTRIMENTI finisci l'ISR

; aggiorna le unità dei minuti
CLRF Conta_sec ; Conta_sec = 0
INCFS U_min, F ; U_min = U_min + 1
MOVLW 10 ; controllo

```

```

        XORWF U_min, W          ; se U_min = 10
        BTFSS STATUS, Z          ; SE U_min = 10 ALLORA azzera U_min e incrementa D_min
        GOTO fine_ISR            ; ALTRIMENTI finisci l'ISR

; aggiorna le decine dei minuti

        CLRF U_min              ; U_min = 0
        INCF D_min, F            ; D_min = D_min + 1
        MOVLW 6                  ; controlla
        XORWF D_min, W          ; se D_min = 6
        BTFSS STATUS, Z          ; SE D_min = 6 ALLORA azzera D_min e incrementa U_ore
        GOTO fine_ISR            ; ALTRIMENTI finisci l'ISR

; aggiorna le unità delle ore e le decine delle ore

        CLRF D_min              ; D_min = 0
        INCF U_ore, F             ; U_ore = U_ore + 1
        MOVLW 2                  ; controlla
        XORWF D_ore, W           ; se D_ore <> 2
        BTFSC STATUS, Z           ; SE D_ore <> 2 ALLORA controlla se U_ore = 10
        GOTO azzera               ; ALTRIMENTI controlla se U_ore = 4
        MOVLW 10                 ; controlla
        XORWF U_ore, W           ; se U_ore = 10
        BTFSS STATUS, Z           ; SE U_ore = 10 ALLORA azzera U_ore e incrementa D_ore
        GOTO fine_ISR            ; ALTRIMENTI finisci l'ISR

        CLRF U_ore              ; U_ore = 0
        MOVLW Blank               ; controlla
        XORWF D_ore, W           ; se D_ore <> blank
        BTFSC STATUS, Z           ; SE D_ore <> blank ALLORA incrementa D_ore
        CLRF D_ore                ; ALTRIMENTI D_ore = 0
        INCF D_ore, F             ; D_ore = D_ore + 1
        GOTO fine_ISR            ; finisci l'ISR

azzera
        MOVLW 4                  ; controlla
        XORWF U_ore, W           ; se U_ore = 4
        BTFSS STATUS, Z           ; SE U_ore = 4 ALLORA azzera e finisci l'ISR
        CLRF U_ore                ; U_ore = 0
        MOVLW Blank               ; spegni
        MOVWF D_ore                ; il display n° 3

; ripristina il contesto del uC

fine_ISR      SWAPF STATUS_temp, W    ; scambia e copia in W i nibble salvati in STATUS_temp
        MOVWF STATUS               ; ripristina il contenuto di STATUS
        SWAPF W_temp, F            ; scambia i nibble salvati in W_temp
        SWAPF W_temp, W            ; scambia e ripristina il contenuto di W
        RETFIE                     ; riprendi il processo interrotto

; FINE ISR rtc =====

; ISR wake_up =====

wake_up       BCF INTCON, INTTE     ; maschera l'interruzione RBO/INT
        BSF INTCON, GIE            ; abilita le interruzioni

        BSF RBO_INT, 0             ; RBO_INT = 1 per segnalare che è stata fatta l'ISR wake-up

        MOVLW 2                  ; inizializza
        MOVWF Contatore2           ; il Contatore2

iniz_contl    MOVLW 200               ; inizializza
        MOVWF Contatore1           ; il Contatore1

due_punti_on3 BTFSS TMR1H, 6          ; SE TMR1H<6> = 1 ALLORA accendi i due punti
        GOTO due_punti_off3         ; ALTRIMENTI spegni i due punti
        BSF Due_punti, 7            ; accendi i due punti
        GOTO fai_ref                ; fai un refresh
        due_punti_off3 BCF Due_punti, 7      ; spegni i due punti

fai_ref        CALL refresh            ; fai il refresh del display
        DECFSZ Contatore1, F          ; Contatore1 = Contatore1 - 1, SE Contatore1 = 0 ALLORA decrementa Contatore 2
        GOTO due_punti_on3           ; ALTRIMENTI fai il refresh del display

down mode      DECFSZ Contatore2, F      ; Contatore2 = Contatore2 - 1, SE Contatore2 = 0 ALLORA finisci ISR e vai in power-
        GOTO iniz_contl               ; ATRIMENTI inizializza Contatore 1
        BCF INTCON, INTF             ; azzera il flag dell'interruzione RBO/INT
        RETFIE                      ; riprendi il processo interrotto

; FINE ISR wake_up =====

; SUBROUTINE refresh =====
; Legge i valori delle ore, dei minuti e dei 2 punti e accende in successione i 4 display a 7 segmenti, nell'ordine che
; va dal display n° 3 al display n° 0.
; Ogni display rimane acceso circa 5 ms, per cui tutta la subroutine viene eseguita in circa 20 ms.
;

; VARIABILI DI INGRESSO:
; U_min   unità dei minuti --> display n° 0
; D_min   decine dei minuti --> display n° 1
; U_ore   unità delle ore --> display n° 2
; D_ore   decine dei minuti --> display n° 3
; Due_punti memorizza l'accensione e lo spegnimento dei due punti --> display n° 1 e n° 2
;

; VARIABILI DI USCITA:
; RA7    RA6    RA5    RA4    RA3    RA2    RA1    RA0    <- PORTA
; DP     a      RESET   BAT    disp3   disp2   disp1   disp0
; RB7    RB6    RB5    RB4    RB3    RB2    RB1    RB0    <- PORTB
; T1OSI  T1OSO   b      c      d      e      f      g
;

refresh       BSF STATUS, RP0            ; seleziona il banco 01

```

```

BCF    TRISB, TRISB0      ; imposta RB0 come pin di output
BCF    TRISB, TRISB1      ; imposta RB1 come pin di output
BCF    STATUS, RP0         ; ritorna al banco 00

CLRF   PORTA              ; spegni i quattro display
MOVF   D_ore, W            ; decodifica i segmenti da accendere
CALL   decod_BCD_7seg     ; per il display n° 3 delle DECINE DELLE ORE
MOVWF  PORTB              ; accendi/spegni i segmenti b, c, d, e, f, g del display n° 3
ANDLW  B'01000000'          ; cattura il valore acceso/spento del segmento a
IORWF  PORTA, F            ; accendi/spegni il segmento a
BSF    PORTA, RA3          ; accendi il display n° 3
CALL   aspetta_5ms         ; lascia acceso il display n° 3 per 5 ms

CLRF   PORTA              ; spegni i quattro display
MOVF   U_ore, W            ; decodifica i segmenti da accendere
CALL   decod_BCD_7seg     ; per il display n° 2 delle UNITÀ DELLE ORE
MOVWF  PORTB              ; accendi/spegni i segmenti b, c, d, e, f, g del display n° 2
ANDLW  B'01000000'          ; cattura il valore acceso/spento del segmento a
IORWF  PORTA, F            ; accendi/spegni il segmento a
MOVF   Due_punti, W        ; cattura il valore acceso/spento dei due punti
IORWF  PORTA, F            ; accendi/spegni i due punti
BSF    PORTA, RA2          ; accendi il display n° 2
CALL   aspetta_5ms         ; lascia acceso il display n° 2 per 5 ms

CLRF   PORTA              ; spegni i quattro display
MOVF   D_min, W            ; decodifica i segmenti da accendere
CALL   decod_BCD_7seg     ; per il display n° 1 delle DECINE DEI MINUTI
MOVWF  PORTB              ; accendi/spegni i segmenti b, c, d, e, f, g del display n° 1
ANDLW  B'01000000'          ; cattura il valore acceso/spento del segmento a
IORWF  PORTA, F            ; accendi/spegni il segmento a
MOVF   Due_punti, W        ; cattura il valore acceso/spento dei due punti
IORWF  PORTA, F            ; accendi/spegni i due punti
BSF    PORTA, RA1          ; accendi il display n° 1
CALL   aspetta_5ms         ; lascia acceso il display n° 1 per 5 ms

CLRF   PORTA              ; spegni i quattro display
MOVF   U_min, W            ; decodifica i segmenti da accendere
CALL   decod_BCD_7seg     ; per il display n° 0 delle UNITÀ DEI MINUTI
MOVWF  PORTB              ; accendi/spegni i segmenti b, c, d, e, f, g del display n° 0
ANDLW  B'01000000'          ; cattura il valore acceso/spento del segmento a
IORWF  PORTA, F            ; accendi/spegni il segmento a
BSF    PORTA, RA0          ; accendi il display n° 0
CALL   aspetta_5ms         ; lascia acceso il display n° 0 per 5 ms

CLRF   PORTA              ; spegni i quattro display

BSF    STATUS, RP0         ; seleziona il banco 01
BSF    TRISB, TRISB0       ; imposta RB0 come pin di input
BSF    TRISB, TRISB1       ; imposta RB1 come pin di input
BCF    STATUS, RP0         ; ritorna al banco 00

RETURN

; FINE SUBROUTINE refresh =====

; SUBROUTINE decod_BCD_7seg =====
; Esegue la decodifica BCD/7 segmenti con la seguente tavola di verità:
;

; N | D C B A || a b c d e f g           a
; ---+-----+-----+-----+-----+-----+-----+
; 0 | 0 0 0 0 || 1 1 1 1 1 1 0             f |   | b
; 1 | 0 0 0 1 || 0 1 0 0 0 0 0             |   g |
; 2 | 0 0 1 0 || 1 1 0 1 0 1 1             -----
; 3 | 0 0 1 1 || 1 1 1 1 0 0 1             -----
; 4 | 0 1 0 0 || 0 1 1 0 0 1 1             e |   | c
; 5 | 0 1 0 1 || 1 0 1 1 0 1 1             |
; 6 | 0 1 1 0 || 1 0 1 1 1 1 1             |
; 7 | 0 1 1 1 || 1 1 0 0 0 0 0             -----
; 8 | 1 0 0 0 || 1 1 1 1 1 1 1             d
; 9 | 1 0 0 1 || 1 1 1 1 0 0 1             |
; 10| 1 0 1 0 || 0 0 0 0 0 0 0             <- blank

; VARIABILE DI INGRESSO:
;   W: codice BCD (numero da 0 a 10)
; VARIABILE DI USCITA:
;   W: segmenti da accendere 1 <-> acceso 0 <-> spento
;

decod_BCD_7seg ADDWF  PCL, F      ; seleziona il codice corrispondente al numero contenuto in W
RETLW  B'01111110'      ; segmenti del numero 0
RETLW  B'00110000'      ; segmenti del numero 1
RETLW  B'01010101'      ; segmenti del numero 2
RETLW  B'01110001'      ; segmenti del numero 3
RETLW  B'00110011'      ; segmenti del numero 4
RETLW  B'01010101'      ; segmenti del numero 5
RETLW  B'01011111'      ; segmenti del numero 6
RETLW  B'01110000'      ; segmenti del numero 7
RETLW  B'01111111'      ; segmenti del numero 8
RETLW  B'01111011'      ; segmenti del numero 9
RETLW  B'00000000'      ; blank

; FINE SUBROUTINE decod_BCD_7seg =====

; SUBROUTINE aspetta_5ms =====
; Utilizza Timer0 per tenere impegnata la CPU per un intervallo di tempo che dura circa 5 ms:
;

; Fosc = 4 MHz                  oscillatore interno
; Fosc/4 = 1 MHz                 frequenza base del segnale di clock
; Tck = 1 / 1 MHz = 1 microsec  periodo base del segnale di clock
; D = 128                        divisore del prescaler
; N = 39                          numero di impulsi contati da Timer0
; T = N x D x Tck = 39 x 128 x 1 microsec = 4,992 ms tempo impiegato da Timer0 per generare l'overflow
; N_iniz = 256 - 39 = 217         numero di partenza di Timer0
;
```

```

aspetta_5ms      BCF      INTCON, TOIF          ; azzerza il flag di interruzione dell'overflow di Timer0
                  MOVLW    217                ; inizializza e
                  MOVWF    TMRO               ; fai partire Timer0
test_TOIF        BTFS     INTCON, TOIF          ; SE TOIF = 1 ALLORA Timer0 si è azzerato, quindi ritorna
                  GOTO    test_TOIF           ; ALTRIMENTI controlla Timer0
                  RETURN
; FINE SUBROUTINE aspetta_5ms =====

; SUBROUTINE antirimbalzo =====
; Tiene impegnata la CPU per circa 40 ms.
; =====
antirimbalzo    CALL    aspetta_5ms
                  RETURN
; FINE SUBROUTINE antirimbalzo =====

; INIZIO PROGRAMMA PRINCIPALE =====
; imposta PORTA e PORTB -----
; RA7   RA6   RA5   RA4   RA3   RA2   RA1   RA0   <- OUTPUT
; DP     a      RESET  BAT    disp3  disp2  disp1  disp0  <- INPUT
; RB7   RB6   RB5   RB4   RB3   RB2   RB1   RB0
; b      c      d      e      f      g      <- OUTPUT
; TIOSI TIOSO          SET    SELECT <- INPUT
;
boot            CLRF    PORTA              ; azzerza i latch di uscita di PORTA
CLRF    PORTB              ; azzerza i latch di uscita di PORTB
MOVLW  B'00000111'          ; spegni i comparatori e abilita la funzione
MOVWF  CMCON             ; di I/O digitale dei pin di PORTA
BSF    STATUS, RP0          ; seleziona il banco 01
MOVLW  B'00110000'          ; RA7 RA6   RA3 RA2 RA1 RA0  pin di output digitale
MOVWF  TRISA              ;          RA5 RA4   pin di input digitale
MOVLW  B'11000011'          ; RB5 RB4   RB3 RB2  pin di output digitale
MOVWF  TRISB              ; RB7 RB6   RB1 RB0  pin di input (Timer1 e pulsanti)
;
; imposta Timer0 -----
; OPTION_REG<7> = RPB0 = 1    resistenze interne di pull-up disattivate
; OPTION_REG<6> = INTEDG = 0   interrupt sul fronte di discesa del pin RB0/INT
; OPTION_REG<5> = TOCS = 0    segnale di clock interno
; OPTION_REG<4> = TOSE = 1    incrementa Timer0 sul fronte di salita
; OPTION_REG<3> = PSA = 0     assegna il prescaler a Timer0
; OPTION_REG<210> = PSC210> = 110  divisore del prescaler D = 128
MOVLW  B'10010110'
MOVWF  OPTION_REG
BCF    STATUS, RP0          ; seleziona il banco 00
;
; imposta Timer1 -----
; T1CON<76> = unimplemented = 00  read as '0'
; T1CON<54> = TICKPS<1:0> = 00  nessun divisore del prescaler
; T1CON<3> = TIOSCEN = 1       abilita l'oscillatore di Timer1
;
; T1CON<2> = T1SYNC = 1       non sincronizzare il clock esterno con la CPU
; T1CON<1> = TMRICS = 1       sorgente esterna del segnale di clock (in questo caso l'oscillatore di Timer1)
; T1CON<0> = TMR1ON = 1       abilita Timer1
MOVLW  B'00001111'
MOVWF  T1CON
;
; imposta l'interruzione diooverflow di Timer1 -----
; PIE Peripheral Interrupt Enable register
BSF    STATUS, RP0          ; seleziona il banco 01
BSF    PIE1, TMR1IE         ; abilita l'interruzione di overflow di Timer1
BCF    STATUS, RP0          ; seleziona il banco 00
;
; INTCON INTerrupt CONtrol Register
BSF    INTCON, PEIE         ; abilita le interruzioni non mascherate delle periferiche
;
; blinking iniziale -----
; fino a quando non viene premuto e rilasciato il pulsante SELECT, fa lampeggiare i 4 display
; contemporaneamente, accendendo tutti i led; i display stanno accesi e spenti per circa
; 300 ms = 15 refresh x 20 ms/refresh.
;
; prepara i led da accendere
MOVLW  B'10000000'          ; accendi

```

```

MOVWF Due_punti      ; i due punti
MOVLW 8               ; accendi tutti
MOVWF D_ore           ; i led del display n° 3
MOVLW 8               ; accendi tutti
MOVWF U_ore           ; i led del display n° 2
MOVLW 8               ; accendi tutti
MOVWF D_min           ; i led del display n° 1
MOVLW 8               ; accendi tutti
MOVWF U_min           ; i led del display n° 0

; blinking

        CLRF PORTA      ; disabilita i 4 display

test_RB0_0
        BCF  PORTE, RB0   ; azzerza il latch di ingresso di RB0
        BTFSC PORTE, RB0   ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai il blinking
        GOTO fine_blink   ; ALTRIMENTI esci dal blinking (SELECT premuto)

        MOVLW 15            ; inizializza
        MOVWF Contatorel    ; il Contatorel

display_on
        CALL refresh        ; fai il refresh del display
        DECF SZ, Contatorel, F ; Contatorel = Contatorel - 1, SE Contatorel = 0 ALLORA lascia spento il display
        GOTO display_on     ; ALTRIMENTI fai il refresh del display

        MOVLW 15            ; inizializza
        MOVWF Contatorel    ; il Contatorel

display_off
        CALL aspetta_5ms    ; fai
        CALL aspetta_5ms    ; passare
        CALL aspetta_5ms    ; circa
        CALL aspetta_5ms    ; 20 ms
        DECF SZ, Contatorel, F ; Contatorel = Contatorel - 1, SE Contatorel = 0 ALLORA accendi il display
        GOTO display_off    ; ALTRIMENTI tieni spento il display

        GOTO test_RB0_0      ; ripeti blink

; imposta le DECINE DELLE ORE ----

fine_blink
        CALL antirimbalzo  ; pausa antirimbalzo
test_RB0_1
        BTFSC PORTB, RB0   ; SE RB0 = 0 (SELECT rilasciato) ALLORA imposta le DECINE DELLE ORE
        GOTO test_RB0_1      ; ALTRIMENTI aspetta perché SELECT è ancora premuto
        CALL antirimbalzo  ; pausa antirimbalzo

; prepara i led da accendere

        BCF Due_punti, 7    ; spegni i due punti
        MOVLW 0              ; D_ore = 0
        MOVWF D_ore           ; U_ore = blank
        MOVLW Blank           ; D_min = blank
        MOVWF D_min           ; U_min = blank
        MOVLW Blank           ; U_min = blank

; controlla RB1 (tasto SET) per l'avanzamento delle DECINE DELLE ORE: 0, 1, 2, 0, 1, 2, ...

test_RB1_0
        BCF  PORTB, RB1   ; azzerza il latch di ingresso di RB1
        BTFSS PORTB, RB1   ; SE RB1 = 1 (SET premuto) ALLORA aspetta che SET venga rilasciato
        GOTO test_RB0_2     ; ALTRIMENTI controlla RB0

        CALL antirimbalzo  ; pausa antirimbalzo
test_RB1_1
        BTFSC PORTB, RB1   ; SE RB1 = 0 (SET rilasciato) ALLORA incrementa le DECINE DELLE ORE
        GOTO test_RB1_1     ; ALTRIMENTI aspetta perché SET è ancora premuto

        CALL antirimbalzo  ; pausa antirimbalzo
        INCF D_ore, F       ; D_ore = D_ore + 1
        MOVLW 3              ; controllo
        XORWF D_ore, W       ; se D_ore = 3
        BTFSC STATUS, Z     ; SE D_ore <> 3 ALLORA fai il refresh
        CLR D_ore             ; ALTRIMENTI D_ore = 0
        GOTO ref_D_ore       ; fai il refresh

test_RB0_2
        BTFSC PORTB, RB0   ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai il refresh
        GOTO fine_D_ore     ; ALTRIMENTI imposta le UNITA' DELLE ORE (SELECT premuto)

ref_D_ore
        CALL refresh        ; fai un refresh
        GOTO test_RB1_0      ; controlla RB1

; imposta le UNITA' DELLE ORE ----

fine_D_ore
        CALL antirimbalzo  ; pausa antirimbalzo
test_RB0_3
        BTFSC PORTB, RB0   ; SE RB0 = 0 (SELECT rilasciato) ALLORA imposta le UNITA' DELLE ORE
        GOTO test_RB0_3      ; ALTRIMENTI aspetta perché SELECT è ancora premuto
        CALL antirimbalzo  ; pausa antirimbalzo

; prepara i led da accendere

        MOVLW 0              ; U_ore = 0
        MOVWF U_ore           ; D_min = blank
        MOVLW Blank           ; U_min = blank
        MOVWF U_min           ; U_min = blank

; controlla RB1 (tasto SET) per l'avanzamento delle UNITA' DELLE ORE: 0, 1, 2, 3, ..., 8, 9, 0, 1, 2, ...

test_RB1_2
        BCF  PORTB, RB1   ; azzerza il latch di ingresso di RB1
        BTFSS PORTB, RB1   ; SE RB1 = 1 (SET premuto) ALLORA aspetta che SET venga rilasciato
        GOTO test_RB0_4     ; ALTRIMENTI controlla RB0

        CALL antirimbalzo  ; pausa antirimbalzo
test_RB1_3
        BTFSC PORTB, RB1   ; SE RB1 = 0 (SET rilasciato) ALLORA incrementa le UNITA' DELLE ORE
        GOTO test_RB1_3     ; ALTRIMENTI aspetta perché SET è ancora premuto

        CALL antirimbalzo  ; pausa antirimbalzo

```

```

INCF    U_ore, F           ; U_ore = U_ore + 1
MOVLW   4                 ; controlla
XORWF   U_ore, W           ; se U_ore = 4
BTFSZ   STATUS, Z           ; SE U_ore = 4 ALLORA controlla se D_ore = 2
GOTO    controlla_10         ; ALTRIMENTI controlla se U_ore = 10

MOVLW   2                 ; controlla
XORWF   D_ore, W           ; se D_ore = 2
BTFSZ   STATUS, Z           ; SE D_ore <> 2 ALLORA continua a incrementare
GOTO    azzera_U_ore         ; ALTRIMENTI U_ore = 0

controlla_10
MOVLW   10                ; controlla
XORWF   U_ore, W           ; se U_ore = 10
BTFSZ   STATUS, Z           ; SE U_ore <> 10 ALLORA fai il refresh
CLRF    U_ore               ; ALTRIMENTI U_ore = 0
GOTO    ref_U_ore            ; fai il refresh

test_RB0_4
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai il refresh
GOTO    fine_U_ore           ; ALTRIMENTI imposta le DECINE DEI MINUTI (SELECT premuto)

ref_U_ore
CALL    refresh              ; fai un refresh
GOTO    test_RB1_2            ; controlla RB1

; imposta le DECINE DEI MINUTI ----

fine_U_ore
CALL    antiribalzo          ; pausa antiribalzo
test_RB0_5
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA imposta le DECINE DEI MINUTI
GOTO    test_RB0_5            ; ALTRIMENTI aspetta perché SELECT è ancora premuto
CALL    antiribalzo          ; pausa antiribalzo

; prepara i led da accendere

BSF     Due_punti, 7          ; accendi i due punti
MOVLW   0
MOVWF   D_min                ; D_min = 0
MOVLW   Blank
MOVWF   U_min                ; U_min = blank

; controlla RB1 (tasto SET) per l'avanzamento delle DECINE DEI MINUTI: 0, 1, 2, 3, 4, 5, 0, 1, 2, ...
test_RB1_4
BCF     PORTB, RB1          ; azzera il latch di ingresso di RB1
BTFSZ   PORTB, RB1          ; SE RB1 = 1 (SET premuto) ALLORA aspetta che SET venga rilasciato
GOTO    test_RB0_6            ; ALTRIMENTI controlla RB0

test_RB1_5
CALL    antiribalzo          ; pausa antiribalzo
BTFSC  PORTB, RB1          ; SE RB1 = 0 (SET rilasciato) ALLORA incrementa le DECINE DEI MINUTI
GOTO    test_RB1_5            ; ALTRIMENTI aspetta perché SET è ancora premuto

CALL    antiribalzo          ; pausa antiribalzo
INCF    D_min, F             ; D_min = D_min + 1
MOVLW   6
XORWF   D_min, W             ; controlla
BTFSZ   STATUS, Z           ; se D_min = 6
CLRF    D_min               ; ALTRIMENTI D_min = 0
GOTO    ref_D_min             ; fai il refresh

test_RB0_6
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai il refresh
GOTO    fine_D_min            ; ALTRIMENTI imposta le UNITA' DEI MINUTI

ref_D_min
CALL    refresh              ; fai un refresh
GOTO    test_RB1_4            ; controlla RB1

; imposta le UNITA' DEI MINUTI ----

fine_D_min
CALL    antiribalzo          ; pausa antiribalzo
test_RB0_7
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA imposta le UNITA' DEI MINUTI
GOTO    test_RB0_7            ; ALTRIMENTI aspetta perché SELECT è ancora premuto
CALL    antiribalzo          ; pausa antiribalzo

; prepara i led da accendere

MOVLW   0
MOVWF   U_min                ; U_min = 0

; controlla RB1 (tasto SET) per l'avanzamento delle UNITA' DEI MINUTI: 0, 1, 2, 3, ..., 8, 9, 0, 1, 2, ...
test_RB1_6
BCF     PORTB, RB1          ; azzera il latch di ingresso di RB1
BTFSZ   PORTB, RB1          ; SE RB1 = 1 (SET premuto) ALLORA aspetta che SET venga rilasciato
GOTO    test_RB0_8            ; ALTRIMENTI controlla RB0

test_RB1_7
CALL    antiribalzo          ; pausa antiribalzo
BTFSC  PORTB, RB1          ; SE RB1 = 0 (SET rilasciato) ALLORA incrementa le UNITA' DEI MINUTI
GOTO    test_RB1_7            ; ALTRIMENTI aspetta perché SET è ancora premuto

CALL    antiribalzo          ; pausa antiribalzo
INCF    U_min, F             ; U_min = U_min + 1
MOVLW   10
XORWF   U_min, W             ; controlla
BTFSZ   STATUS, Z           ; se U_min <> 10 ALLORA fai il refresh
CLRF    U_min               ; ALTRIMENTI U_min = 0
GOTO    ref_U_min             ; fai il refresh

test_RB0_8
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai il refresh
GOTO    fine_U_min            ; ALTRIMENTI fa partire l'orologio

ref_U_min
CALL    refresh              ; fai un refresh
GOTO    test_RB1_6            ; controlla RB1

; fai partire l'orologio ----

fine_U_min
CALL    antiribalzo          ; pausa antiribalzo
test_RB0_9
BTFSC  PORTB, RB0          ; SE RB0 = 0 (SELECT rilasciato) ALLORA fai partire l'orologio
GOTO    test_RB0_9            ; ALTRIMENTI aspetta perché SELECT è ancora premuto
CALL    antiribalzo          ; pausa antiribalzo

CLRW                           ; controlla

```

```

XORWF D_ore, W           ; se D_ore = 0
BTFSW STATUS, Z          ; SE D_ore = 0 ALLORA spegni il display n° 3
GOTO inizializza          ; ALTRIMENTI lascia acceso il display n° 3 e continua
MOVWF Blank               ; spegni
MOVWF D_ore               ; il display n° 3

inizializza CLRFB Conta_sec ; azzerà il contatore dei secondi
CLRFB Due_punti           ; spegni i due punti

MOVLW 1                  ; inizializza
MOVWF Acceso              ; la variabile booleana Acceso

MOVLW 1                  ; inizializza
MOVWF RB0_INT             ; la variabile booleana RB0_INT

BCF T1CON, TMRION         ; ferma Timer1
CLRF TMRIH                ; inizializza
CLRF TMRL                ; Timer1

BCF INTCON, INTE           ; maschera l'interruzione RB0/INT
BSF INTCON, GIE            ; abilita le interruzioni

BSF T1CON, TMRION         ; fai partire Timer1

; controllo il tipo di alimentazione ----

test_alim BTFSW PORTA, RA4 ; SE RA4 = 1 ALLORA alimentazione da rete
GOTO da_batteria          ; ALTRIMENTI alimentazione da batteria

; alimentazione da rete ----

        BTFSW Acceso, 0      ; SE Acceso = 1 ALLORA visualizza l'ora e controlla RB0
        GOTO test_RB0_10       ; ALTRIMENTI non visualizzare l'ora e controlla RB0

        BTFSW TMRIH, 6        ; SE TMRIH<6> = 1 ALLORA accendi i due punti
        GOTO due_punti_off     ; ALTRIMENTI spegni i due punti
        BSF Due_punti, 7        ; accendi i due punti
        GOTO fai_refresh        ; fai un refresh
        BCF Due_punti, 7        ; spegni i due punti

fai_refresh CALL refresh      ; fai un refresh

test_RB0_10 BTFSW PORTB, RB0 ; SE RB0 = 1 (SELECT premuto) ALLORA aspetta che SELECT venga rilasciato
GOTO test_alim              ; ALTRIMENTI controlla il tipo di alimentazione
CALL antiribalzo            ; pausa antiribalzo

test_RB0_11 BTFSW PORTB, RB0 ; SE RB0 = 0 (SELECT rilasciato) ALLORA modifica la variabile Acceso
GOTO test_RB0_11              ; ALTRIMENTI aspetta perché SELECT è ancora premuto
CALL antiribalzo            ; pausa antiribalzo

        BTFSW Acceso, 0      ; SE Acceso = 1 ALLORA spegni (Acceso = 0)
        GOTO accendi          ; ALTRIMENTI accendi
        BCF Acceso, 0           ; Acceso = 0
        GOTO test_alim          ; controlla il tipo di alimentazione
accendi    BSF Acceso, 0           ; Acceso = 1
        GOTO test_alim          ; controlla il tipo di alimentazione

; alimentazione da batteria ----

; fai due bblink

da_batteria BTFSW RB0_INT, 0 ; SE RB0_INT = 1 ALLORA fai blink e poi va in power-down mode
GOTO power_down          ; ALTRIMENTI non fare blink e vai subito in power-down mode

        CLRF RB0_INT           ; azzerà la variabile booleana RB0_INT

        MOVLW 2                  ; inizializza
        MOVWF Contatore2          ; il Contatore2

fai_blink MOVLW 15             ; inizializza
MOVWF Contatore1          ; il Contatore1

disp_off CALL aspetta_5ms      ; fai
CALL aspetta_5ms          ; passare
CALL aspetta_5ms          ; circa
CALL aspetta_5ms          ; 25 ms
DECFSZ Contatore1, F        ; Contatore1 = Contatore1 - 1, SE Contatore1 = 0 ALLORA accendi il display
GOTO disp_off              ; ALTRIMENTI tieni spento il display

        MOVLW 15                  ; inizializza
        MOVWF Contatore1          ; il Contatore1

        BTFSW TMRIH, 6        ; SE TMRIH<6> = 1 ALLORA accendi i due punti
        GOTO due_punti_off2      ; ALTRIMENTI spegni i due punti
        BSF Due_punti, 7        ; accendi i due punti
        GOTO disp_on              ; fai un refresh
        BCF Due_punti, 7        ; spegni i due punti

disp_on    CALL refresh          ; fai il refresh del display
DECFSZ Contatore1, F        ; Contatore1 = Contatore1 - 1, SE Contatore1 = 0 ALLORA lascia spento il display
GOTO disp_on              ; ALTRIMENTI fai il refresh del display

        DECFSZ Contatore2, F        ; Contatore2 = Contatore2 - 1, SE Contatore2 = 0 ALLORA power-down mode
GOTO fai_blink              ; ALTRIMENTI fai un blink

; power-down mode

power_down BCF INTCON, INTF   ; azzerà il flag dell'interruzione INT/RB0
BSF INTCON, INTE             ; abilita l'interruzione INT/RB0

        SLEEP                   ; power-down mode
        NOP                     ; istruzione nulla eseguita sempre dopo il wake-up
GOTO test_alim              ; controlla il tipo di alimentazione

; FINE PROGRAMMA PRINCIPALE =====

```

end

DATASHEET



A Clock Design Using the PIC16C54 for LED Displays and Switch Inputs

Author: Dan Matthews
Microchip Technology Inc.

INTRODUCTION

The purpose of this application note is to design a clock while multiplexing the features as much as possible, allowing the circuit to use the 18-pin PIC16C54. Other devices in the Microchip line expand on this part, making it a good starting point for learning the basics. This design is useful because it utilizes every pin for output and switches some of them to inputs briefly to read the keys. For a more extensive clock design, consult application note AN529.

THE DESIGN

This design is a simple time-of-day clock incorporating four seven-segment LED displays and three input switches. There is also an additional reset switch that would not normally be incorporated into the final design. The schematic is illustrated in Figure 1.

CONNECTIONS

The individual segments of each display are connected together, A-A-A-A, B-B-B-B, etc. The displays are numbered from the right, or least significant digit. The second display from the right is flipped upside down to align its decimal with the third display, creating the center clock colon. Therefore the segments are not tied together evenly straight across on the board, but must compensate for the change in one display's orientation. The common cathode for each display is turned on with transistors connected to the four I/O lines of PORTA. The connections are RA0-CC4/Digit4, RA1-CC3/Digit3, RA2-CC2/Digit2, RA3-CC1/Digit1. A low output turns on the PNP transistor for the selected display. The PORTB pins activate the LED segments. For this design only the center colon decimal points were connected. The connections are RB0-dp, RB1-A, RB2-B, RB3-C...RB7-G.

The switches are also connected to PORTB I/O pins. PORTB pins RB1, RB2, and RB3 are pulled low with 10 kΩ resistors. This value is high enough to not draw current away from the LEDs when they are being driven on. Inputs are detected by pulling the pins high with a switch to VDD through 820Ω resistors. This value is low enough to pull the pin high quickly when the outputs have been turned off, and to create a 90% of VDD high input.

OPERATION

Switches

When no buttons are pressed, the circuit will display the current time, starting at 12:00 on reset. Pressing SW1 will cause seconds to be displayed. The time is set by pressing SW2 to advance minutes, and SW3 to advance hours. Since each of the segments are tied together across all displays, only one display should be turned on at a time, or all displays turned on would display duplicate data. The displays are turned on right to left, with each display's value being output in its turn. This is done fast enough so that there is no perceived flicker. The switches are read between display cycles.

Timing

The PIC16C5X prescaler is assigned to TMR0 as a 1:16 divide. The T0CKI pin is tied low since it is not used. The OPTION Register is loaded with 03h to initialize this prescaler setup. The software is written with timing based on a 4.000 MHz crystal. The instruction clock is 1.000 MHz after the internal divide by four. The 8-bit TMR0 register rolls over every 256 cycles, for a final frequency of 244.1406 Hz. (exactly a 4.096 ms period) A variable named sec_nth is used to count 244 roll-overs of TMR0 for one second. The benefit of keeping time with a nth variable is that it can be written to as needed to adjust time in "nths" of a second, allowing almost any odd crystal frequency to be used. Simply determine the best prescale and "nth" divider, and compute the "nth" adjustment needed for each minute, hour, twelve hour roll-over. Time can be kept accurately to two "nths" a day (an "nth" is 1/244 of a second in this case). In this circuit, 9 "nths" are subtracted each minute, 34 "nths" are added each hour, and 6 "nths" subtracted every twelve hour roll-over. This leaves a computed error of 1.5 seconds/year except for crystal frequency drift. Another possible solution is to initialize TMR0 to some value that causes a roll-over at a predetermined time interval. Writing to TMR0 causes two clock cycles to be missed while clock edges realign, which would have to be accounted for. This is described in the *Microchip Data Book*.

AN590

Displays

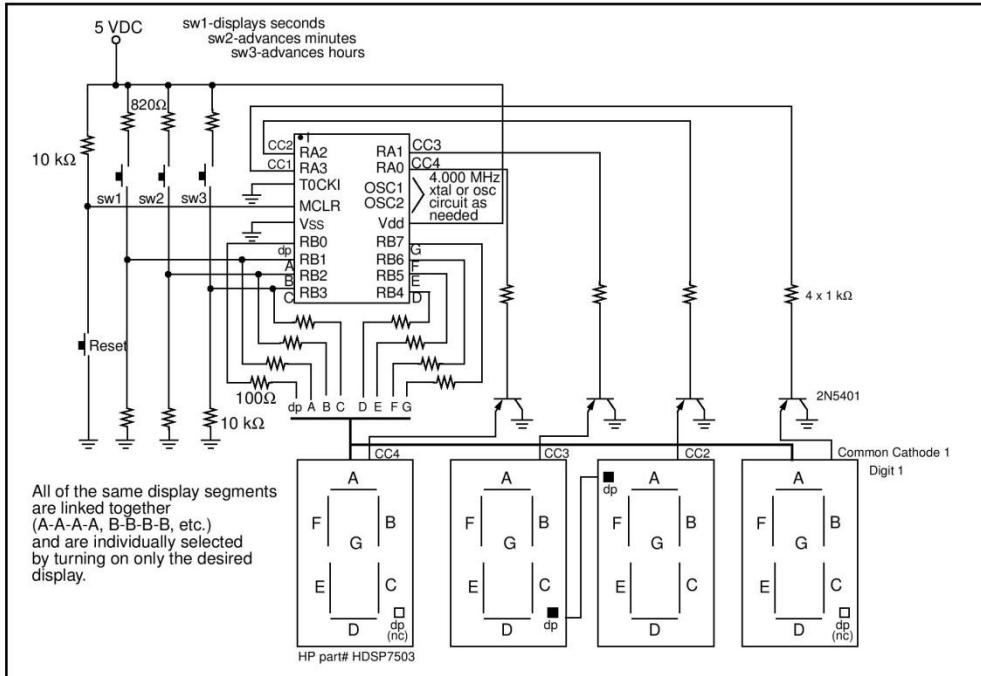
The program contains portions of code that act as a display driver. A variable exists for each of the four displays. A hex value from 0 to 9 can be written to these variables and they will be converted to display code and output to the displays. Only one display is actually on at a time, and its code is output into it in its turn. Another section of code takes the seconds, minutes, or hours value and separates it into the two digits needed for each display. In other words, 48 seconds would be separated into a "4" and an "8" and written to the appropriate display variable. The displays used were common cathode and turned on with transistors to avoid trying to sink too much current into the PIC16C5X. A display is enabled with a zero at the appropriate pin. 100 Ω resistors were used in series with the segments

to obtain the desired brightness. Different values may be required if different displays are used. Since each display is on less than one fourth of the time, the resistor value must be low enough to compensate for the needed forward current.

CONCLUSION

The instruction execution speed of the PIC16C54 (and the rest of the PICmicro™ series) allows many functions to be implemented with only a few pins, by multiplexing them in software. User inputs, Timer0 Counter, and multiple LED displays are all accommodated with little or no sacrifice in functionality.

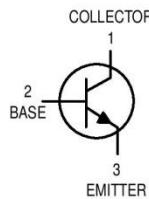
FIGURE 1: TIME OF DAY CLOCK USING A PIC16C54



Amplifier Transistors

NPN Silicon

BC337,-16,-25,-40
BC338,-16,-25,-40



MAXIMUM RATINGS

| Rating | Symbol | BC337 | BC338 | Unit |
|--|----------------|-------------|-------|------------------------------|
| Collector-Emitter Voltage | V_{CEO} | 45 | 25 | Vdc |
| Collector-Base Voltage | V_{CBO} | 50 | 30 | Vdc |
| Emitter-Base Voltage | V_{EBO} | 5.0 | | Vdc |
| Collector Current — Continuous | I_C | 800 | | mAdc |
| Total Device Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C | P_D | 625 5.0 | | mW mW/ $^\circ\text{C}$ |
| Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C | P_D | 1.5 12 | | Watt mW/ $^\circ\text{C}$ |
| Operating and Storage Junction Temperature Range | T_J, T_{stg} | -55 to +150 | | $^\circ\text{C}$ |

CASE 29-04, STYLE 17
TO-92 (TO-226AA)



THERMAL CHARACTERISTICS

| Characteristic | Symbol | Max | Unit |
|---|-----------------|------|---------------------------|
| Thermal Resistance, Junction to Ambient | $R_{\theta JA}$ | 200 | $^\circ\text{C}/\text{W}$ |
| Thermal Resistance, Junction to Case | $R_{\theta JC}$ | 83.3 | $^\circ\text{C}/\text{W}$ |

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

| Characteristic | Symbol | Min | Typ | Max | Unit |
|----------------|--------|-----|-----|-----|------|
|----------------|--------|-----|-----|-----|------|

OFF CHARACTERISTICS

| | | | | | | |
|--|----------------|---------------|----------|--------|------------|------|
| Collector-Emitter Breakdown Voltage ($I_C = 10 \text{ mA}, I_B = 0$) | BC337 BC338 | $V_{(BR)CEO}$ | 45 25 | — — | — — | Vdc |
| Collector-Emitter Breakdown Voltage ($I_C = 100 \mu\text{A}, I_E = 0$) | BC337 BC338 | $V_{(BR)CES}$ | 50 30 | — — | — — | Vdc |
| Emitter-Base Breakdown Voltage ($I_E = 10 \mu\text{A}, I_C = 0$) | | $V_{(BR)EBO}$ | 5.0 | — | — | Vdc |
| Collector Cutoff Current ($V_{CB} = 30 \text{ V}, I_E = 0$) ($V_{CB} = 20 \text{ V}, I_E = 0$) | BC337 BC338 | I_{CBO} | — — | — — | 100 100 | nAdc |
| Collector Cutoff Current ($V_{CE} = 45 \text{ V}, V_{BE} = 0$) ($V_{CE} = 25 \text{ V}, V_{BE} = 0$) | BC337 BC338 | I_{CES} | — — | — — | 100 100 | nAdc |
| Emitter Cutoff Current ($V_{EB} = 4.0 \text{ V}, I_C = 0$) | | I_{EBO} | — | — | 100 | nAdc |

BC337,-16,-25,-40 BC338,-16,-25,-40

 ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted) (Continued)

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|-----------------|--------------------------------|-----------------------|-------------------------------|-----------------|
| ON CHARACTERISTICS | | | | | |
| DC Current Gain ($I_C = 100 \text{ mA}$, $V_{CE} = 1.0 \text{ V}$) ($I_C = 300 \text{ mA}$, $V_{CE} = 1.0 \text{ V}$) | h_{FE} | 100 100 160 250 60 | — — — — — | 630 250 400 630 — | — |
| Base-Emitter On Voltage ($I_C = 300 \text{ mA}$, $V_{CE} = 1.0 \text{ V}$) | $V_{BE(on)}$ | — | — | 1.2 | V_{dc} |
| Collector-Emitter Saturation Voltage ($I_C = 500 \text{ mA}$, $I_B = 50 \text{ mA}$) | $V_{CE(sat)}$ | — | — | 0.7 | V_{dc} |
| SMALL-SIGNAL CHARACTERISTICS | | | | | |
| Output Capacitance ($V_{CB} = 10 \text{ V}$, $I_E = 0$, $f = 1.0 \text{ MHz}$) | C_{ob} | — | 15 | — | pF |
| Current-Gain — Bandwidth Product ($I_C = 10 \text{ mA}$, $V_{CE} = 5.0 \text{ V}$, $f = 100 \text{ MHz}$) | f_T | — | 210 | — | MHz |

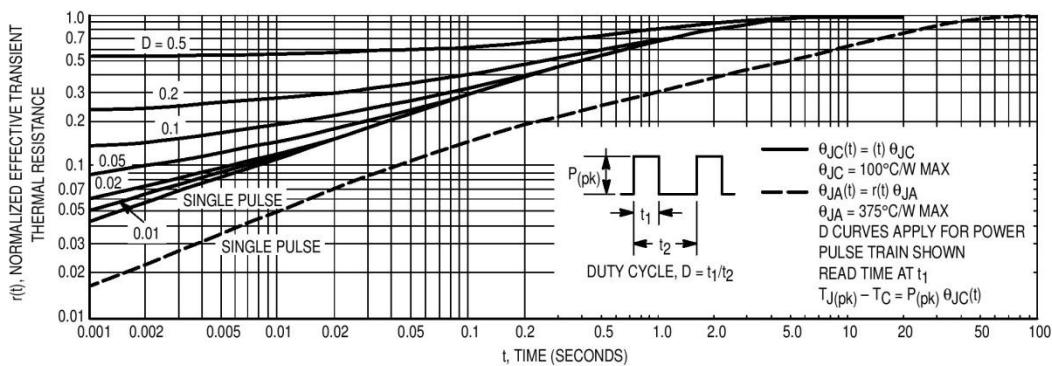


Figure 1. Thermal Response

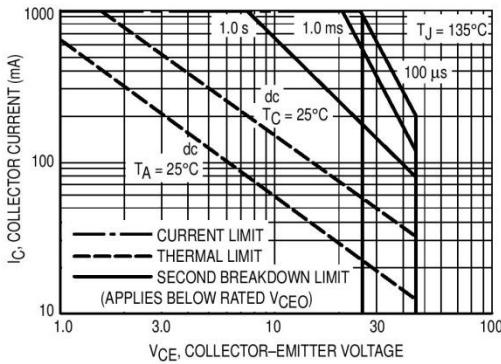


Figure 2. Active Region — Safe Operating Area

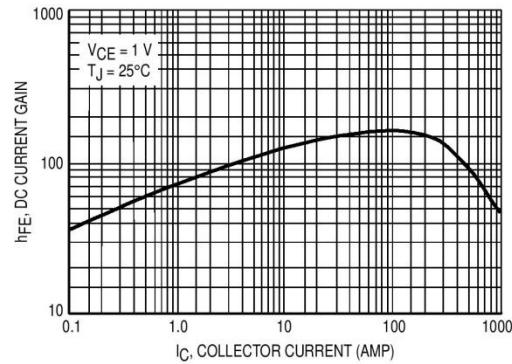


Figure 3. DC Current Gain

BZX79C 3V3 - 33 Series Half Watt Zeners

Absolute Maximum Ratings*

TA = 25°C unless otherwise noted

| Parameter | Value | Units |
|---|-------------|-------------|
| Storage Temperature Range | -65 to +200 | °C |
| Maximum Junction Operating Temperature | + 200 | °C |
| Lead Temperature (1/16" from case for 10 seconds) | + 230 | °C |
| Total Device Dissipation Derate above 25°C | 500 4.0 | mW mW/°C |
| Surge Power** | 30 | W |

*These ratings are limiting values above which the serviceability of the diode may be impaired.

**Non-recurrent square wave PW= 8.3 ms, TA= 50 degrees C.

Tolerance: C = 5%



Electrical Characteristics

TA = 25°C unless otherwise noted

| Device | V_z^* (V) | | Z_Z (Ω) | @ I_{ZT} (mA) | Z_{ZK} (Ω) | @ I_{ZT} (mA) | V_R (V) | @ I_R (μA) | T_C (mV/°C) | |
|------------|----------------|------|--------------|-----------------------|-----------------|-----------------------|--------------|--------------------|------------------|--------|
| | MIN | MAX | | | | | | | MIN | MAX |
| BZX79C 3V3 | 3.1 | 3.5 | 95 | 5.0 | 600 | 1.0 | 1.0 | 25 | - 3.5 | 0.0 |
| BZX79C 3V6 | 3.4 | 3.8 | 90 | 5.0 | 600 | 1.0 | 1.0 | 15 | - 3.5 | 0.0 |
| BZX79C 3V9 | 3.7 | 4.1 | 90 | 5.0 | 600 | 1.0 | 1.0 | 10 | - 3.5 | + 0.3 |
| BZX79C 4V3 | 4.0 | 4.6 | 90 | 5.0 | 600 | 1.0 | 1.0 | 5.0 | - 3.5 | + 1.0 |
| BZX79C 4V7 | 4.4 | 5.0 | 80 | 5.0 | 500 | 1.0 | 2.0 | 3.0 | - 3.5 | + 0.2 |
| BZX79C 5V1 | 4.8 | 5.4 | 60 | 5.0 | 480 | 1.0 | 2.0 | 2.0 | - 2.7 | + 1.2 |
| BZX79C 5V6 | 5.2 | 6.0 | 40 | 5.0 | 400 | 1.0 | 2.0 | 1.0 | - 2.0 | + 2.5 |
| BZX79C 6V2 | 5.8 | 6.6 | 10 | 5.0 | 150 | 1.0 | 4.0 | 3.0 | + 0.4 | + 3.7 |
| BZX79C 6V8 | 6.4 | 7.2 | 15 | 5.0 | 80 | 1.0 | 4.0 | 2.0 | + 1.2 | + 4.5 |
| BZX79C 7V5 | 7.0 | 7.9 | 15 | 5.0 | 80 | 1.0 | 5.0 | 1.0 | + 2.5 | + 5.3 |
| BZX79C 8V2 | 7.7 | 8.7 | 15 | 5.0 | 80 | 1.0 | 5.0 | 0.7 | + 3.2 | + 6.2 |
| BZX79C 9V1 | 8.5 | 9.6 | 15 | 5.0 | 100 | 1.0 | 6.0 | 0.5 | + 3.8 | + 7.0 |
| BZX79C 10 | 9.4 | 10.6 | 20 | 5.0 | 150 | 1.0 | 7.0 | 0.2 | + 4.5 | + 8.0 |
| BZX79C 11 | 10.4 | 11.6 | 20 | 5.0 | 150 | 1.0 | 8.0 | 0.1 | + 5.4 | + 9.0 |
| BZX79C 12 | 11.4 | 12.7 | 25 | 5.0 | 150 | 1.0 | 8.0 | 0.1 | + 6.0 | + 10 |
| BZX79C 13 | 12.4 | 14.1 | 30 | 5.0 | 170 | 1.0 | 8.0 | 0.10 | - 7.0 | + 11 |
| BZX79C 15 | 13.8 | 15.6 | 30 | 5.0 | 200 | 1.0 | 10.5 | 0.05 | - 9.2 | + 13 |
| BZX79C 16 | 15.3 | 17.1 | 40 | 5.0 | 200 | 1.0 | 11.2 | 0.05 | + 10.4 | + 14 |
| BZX79C 18 | 16.8 | 19.1 | 45 | 5.0 | 225 | 1.0 | 12.6 | 0.05 | + 12.4 | + 16 |
| BZX79C 20 | 18.8 | 21.2 | 55 | 5.0 | 225 | 1.0 | 14 | 0.05 | + 14.4 | + 18 |
| BZX79C 22 | 20.8 | 23.3 | 55 | 5.0 | 250 | 1.0 | 15.4 | 0.05 | + 16.4 | + 20 |
| BZX79C 24 | 22.8 | 25.6 | 70 | 5.0 | 250 | 1.0 | 16.8 | 0.05 | + 18.4 | + 22 |
| BZX79C 27 | 25.1 | 28.9 | 80 | 2.0 | 300 | 0.5 | 18.9 | 0.05 | + 21.4 | + 25.3 |
| BZX79C 30 | 28 | 32 | 80 | 2.0 | 300 | 0.5 | 21 | 0.05 | + 24.4 | + 29.4 |
| BZX79C 33 | 31 | 35 | 80 | 2.0 | 325 | 0.5 | 23.1 | 0.05 | + 27.4 | + 33.4 |

V_F Forward Voltage = 1.5 V Maximum @ I_F = 100 mA for all BZX 79 series

*Pulse Test: Pulse Width ± 300 ms, Duty Cycle ± 2.0%

© 1997 Fairchild Semiconductor Corporation

SPECIFICATION FOR CRYSTAL UNITS

| | |
|--------------------------------|--|
| 1.TYPE | SM-26F |
| 2.NOMINAL FREQUENCY | 32.768 kHz |
| 3.LOADING CAPACITANCE | 12.5 pF |
| 4.FREQUENCY TOLERANCE | $\pm 20 \times 10^{-6}$ at 25 ± 2 °C |
| 5.SERIES RESISTANCE | 50 kΩ max. |
| 6.Q FACTOR | 40000 min. |
| 7.MEASUREMENT DRIVE LEVEL | 1.0 ± 0.2 μW |
| 8.TURNOVER TEMPERATURE | 25 ± 5 °C |
| 9.PARABOLIC COEFFICIENT | $-0.04 \times 10^{-6}/^{\circ}\text{C}^2$ max. |
| 10.INSULATION RESISTANCE | 500 MΩ min./DC100V |
| 11.PARALLEL CAPACITANCE | 1.1 pF typ. |
| 12.OPERATING TEMPERATURE RANGE | -40 ~ +85 °C |
| 13.STORAGE TEMPERATURE RANGE | -40 ~ +85 °C |
| 14.TAPING | 2500pcs/reel |
| 15.PART NUMBER | 0410-4581 |
| 16.DIMENSION AND MARKING | :Refer to Fig-1 and Table-1 |

Fig-1

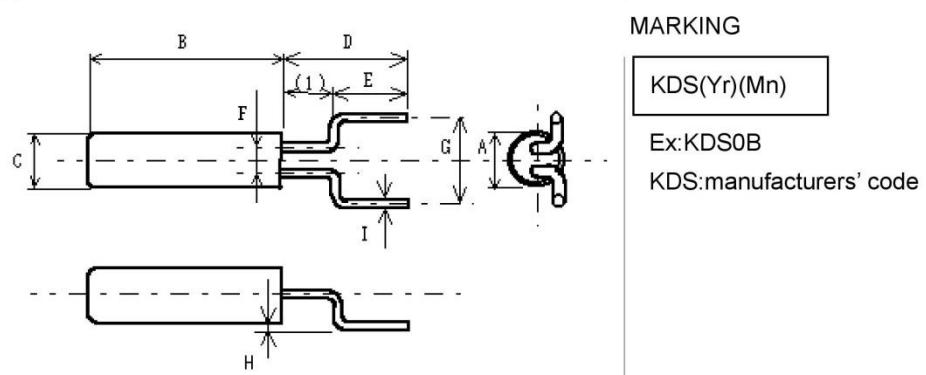


Table-1

| A | B | C | D | E | F | G | H | I |
|-------------------|------------------|-------------|-------------|-------------|-------------|--------------|--------------|-----------------|
| Φ2.0 +0.1/-0.2 | 6.0 +0.1/-0.2 | 1.9 ±0.2 | 3.0 ±0.2 | 2.0 ±0.2 | 0.7 ±0.2 | 2.54 ±0.2 | 0 ~ +0.25 | Φ0.275 ±0.03 |

DAISHINKU CORPORATION

LM78XX / LM78XXA 3-Terminal 1 A Positive Voltage Regulator

Features

- Output Current up to 1 A
- Output Voltages: 5, 6, 8, 9, 10, 12, 15, 18, 24 V
- Thermal Overload Protection
- Short-Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The LM78XX series of three-terminal positive regulators is available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down, and safe operating area protection. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed-voltage regulators, these devices can be used with external components for adjustable voltages and currents.



Ordering Information⁽¹⁾

| Product Number | Output Voltage Tolerance | Package | Operating Temperature | Packing Method |
|----------------|--------------------------|--------------------------|---|----------------|
| LM7805CT | $\pm 4\%$ | TO-220 (Single Gauge) | -40°C to $+125^{\circ}\text{C}$ | Rail |
| LM7806CT | | | | |
| LM7808CT | | | | |
| LM7809CT | | | | |
| LM7810CT | | | | |
| LM7812CT | | | | |
| LM7815CT | | | | |
| LM7818CT | | | | |
| LM7824CT | | | | |
| LM7805ACT | $\pm 2\%$ | | 0°C to $+125^{\circ}\text{C}$ | |
| LM7809ACT | | | | |
| LM7810ACT | | | | |
| LM7812ACT | | | | |
| LM7815ACT | | | | |

Note:

1. Above output voltage tolerance is available at 25°C .

Block Diagram

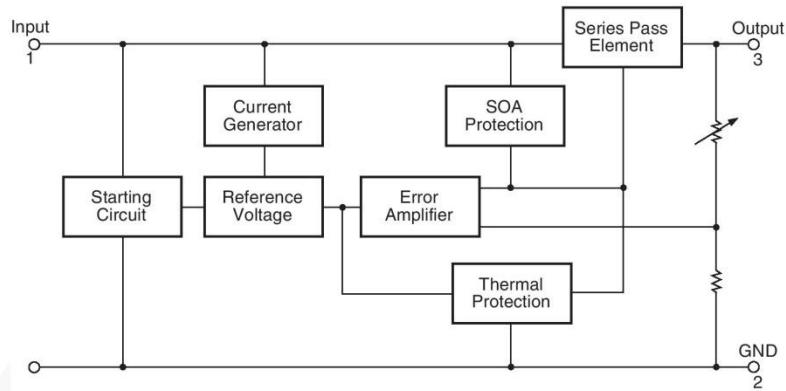


Figure 1. Block Diagram

Absolute Maximum Ratings

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only. Values are at $T_A = 25^\circ\text{C}$ unless otherwise noted.

| Symbol | Parameter | Value | Unit |
|-----------------|--|--------------------------------------|--------------------|
| V_I | Input Voltage | $V_O = 5 \text{ V to } 18 \text{ V}$ | V |
| | | $V_O = 24 \text{ V}$ | |
| $R_{\theta JC}$ | Thermal Resistance, Junction-Case (TO-220) | 5 | $^\circ\text{C/W}$ |
| $R_{\theta JA}$ | Thermal Resistance, Junction-Air (TO-220) | 65 | $^\circ\text{C/W}$ |
| T_{OPR} | Operating Temperature Range | -40 to +125 | $^\circ\text{C}$ |
| | | 0 to +125 | |
| T_{STG} | Storage Temperature Range | -65 to +150 | $^\circ\text{C}$ |

Electrical Characteristics (LM7805)

Refer to the test circuit, $-40^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$, $I_O = 500 \text{ mA}$, $V_I = 10 \text{ V}$, $C_I = 0.1 \mu\text{F}$, unless otherwise specified.

| Symbol | Parameter | Conditions | | Min. | Typ. | Max. | Unit |
|-----------------------|-------------------------------------|--|---|------|------|-------|------------------------|
| V_O | Output Voltage | $T_J = +25^{\circ}\text{C}$ | | 4.80 | 5.00 | 5.20 | V |
| | | $I_O = 5 \text{ mA to } 1 \text{ A}$, $P_O \leq 15 \text{ W}$, $V_I = 7 \text{ V to } 20 \text{ V}$ | | 4.75 | 5.00 | 5.25 | |
| Regline | Line Regulation ⁽²⁾ | $T_J = +25^{\circ}\text{C}$ | $V_I = 7 \text{ V to } 25 \text{ V}$ | | 4.0 | 100.0 | mV |
| | | | $V_I = 8 \text{ V to } 12 \text{ V}$ | | 1.6 | 50.0 | |
| Regload | Load Regulation ⁽²⁾ | $T_J = +25^{\circ}\text{C}$ | $I_O = 5 \text{ mA to } 1.5 \text{ A}$ | | 9.0 | 100.0 | mV |
| | | | $I_O = 250 \text{ mA to } 750 \text{ mA}$ | | 4.0 | 50.0 | |
| I_Q | Quiescent Current | $T_J = +25^{\circ}\text{C}$ | | | 5.0 | 8.0 | mA |
| ΔI_Q | Quiescent Current Change | $I_O = 5 \text{ mA to } 1 \text{ A}$ | | | 0.03 | 0.50 | mA |
| | | $V_I = 7 \text{ V to } 25 \text{ V}$ | | | 0.30 | 1.30 | |
| $\Delta V_O/\Delta T$ | Output Voltage Drift ⁽³⁾ | $I_O = 5 \text{ mA}$ | | | -0.8 | | mV/ $^{\circ}\text{C}$ |
| V_N | Output Noise Voltage | $f = 10 \text{ Hz to } 100 \text{ kHz}$, $T_A = +25^{\circ}\text{C}$ | | | 42.0 | | $\mu\text{V}/V_O$ |
| RR | Ripple Rejection ⁽³⁾ | $f = 120 \text{ Hz}$, $V_I = 8 \text{ V to } 18 \text{ V}$ | | 62.0 | 73.0 | | dB |
| V_{DROP} | Dropout Voltage | $T_J = +25^{\circ}\text{C}$, $I_O = 1 \text{ A}$ | | | 2.0 | | V |
| R_O | Output Resistance ⁽³⁾ | $f = 1 \text{ kHz}$ | | | 15.0 | | $\text{m}\Omega$ |
| I_{SC} | Short-Circuit Current | $T_J = +25^{\circ}\text{C}$, $V_I = 35 \text{ V}$ | | | 230 | | mA |
| I_{PK} | Peak Current ⁽³⁾ | $T_J = +25^{\circ}\text{C}$ | | | 2.2 | | A |

Notes:

2. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

3. These parameters, although guaranteed, are not 100% tested in production.

LM78XX / LM78XXA — 3-Terminal 1 A Positive Voltage Regulator

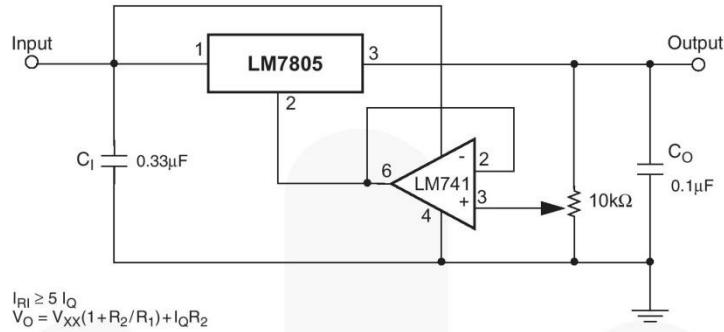


Figure 12. Adjustable Output Regulator (7 V to 30 V)

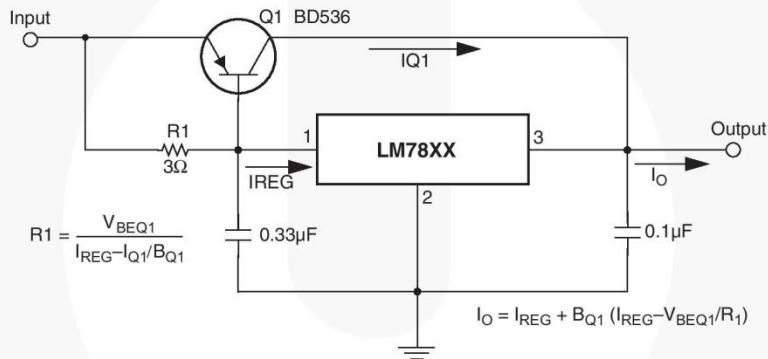


Figure 13. High-Current Voltage Regulator

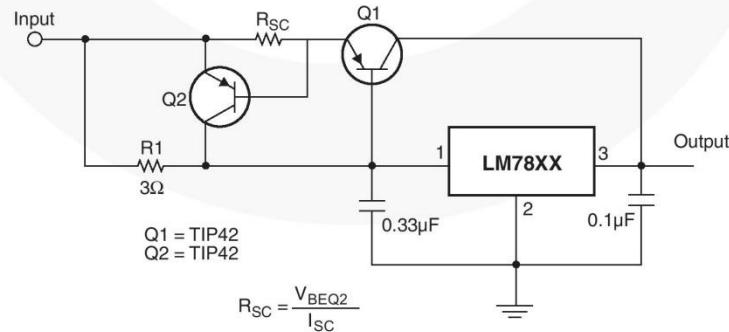


Figure 14. High Output Current with Short-Circuit Protection