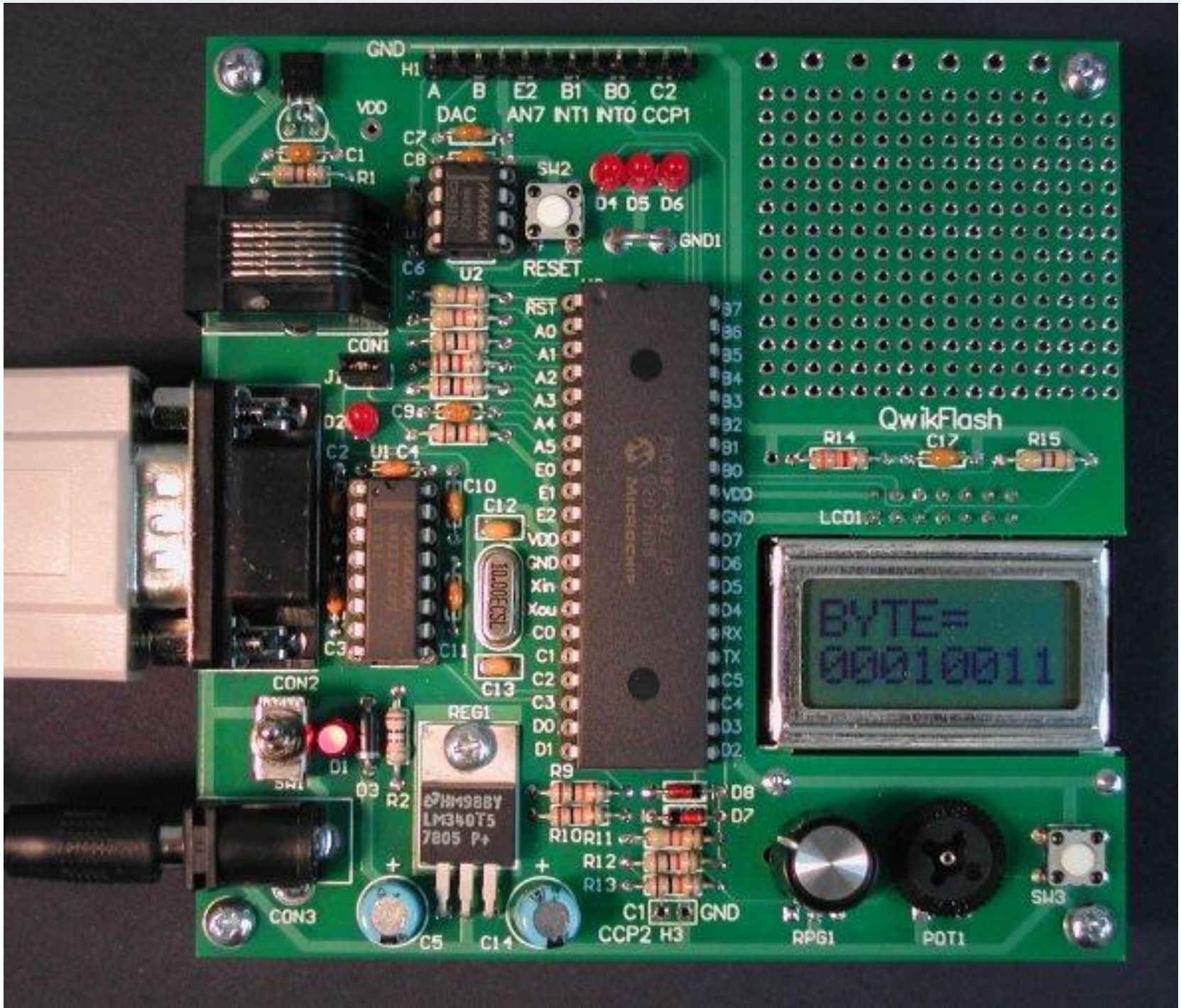
Sulautetut prosessorijärjestelmät

(4 op)

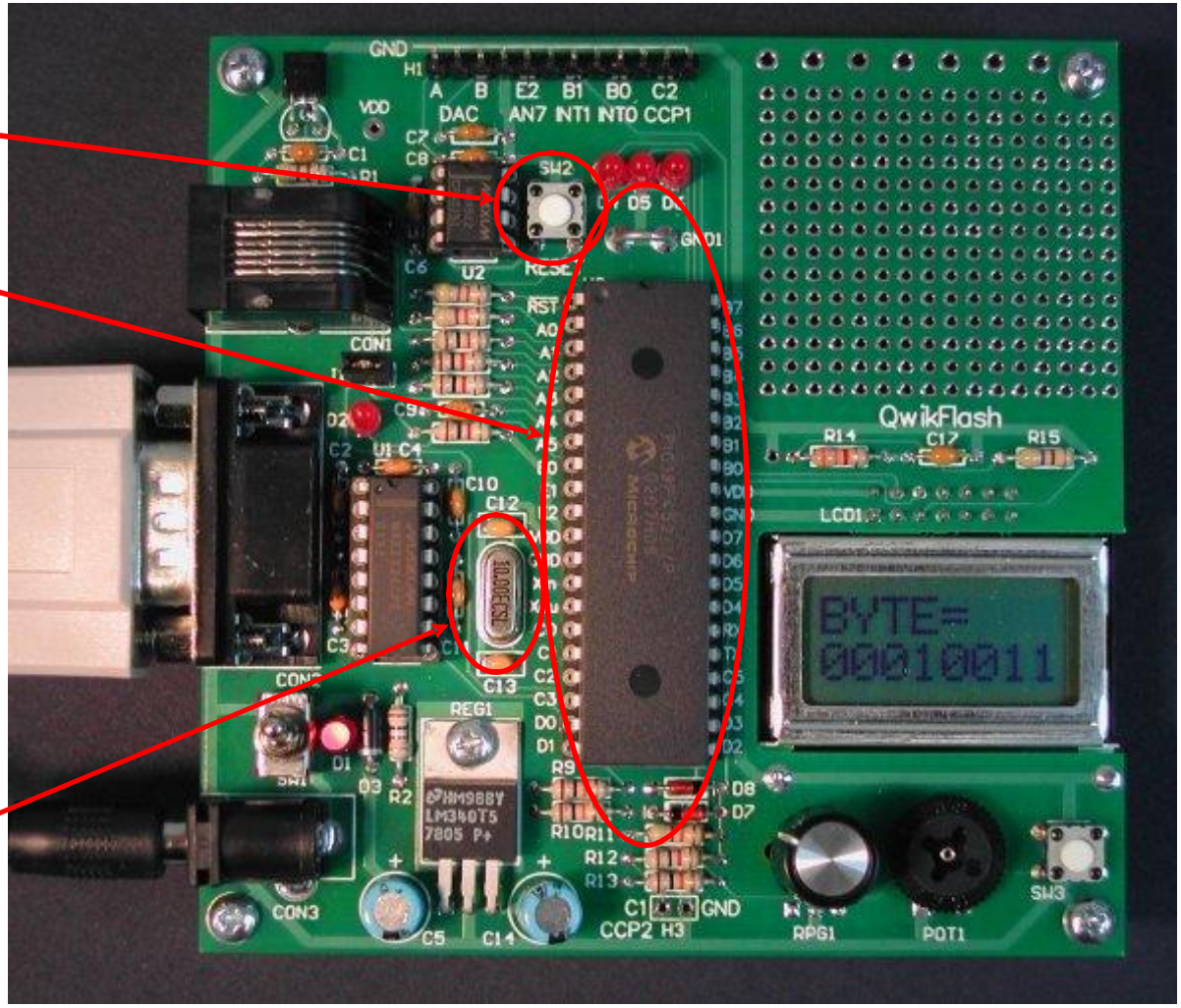
QwikFlash –laudan rakenne, komponentit ja QwikBug –ohjelma

- Käydään läpi laudalla olevat komponentit, niiden sijainti ja käyttö
- QwikBug –ohjelman käyttö
  - Ohjelmointi
  - Debuggaus
- Tässä esitetyt asia voidaan yleistää koskemaan muitakin systeemejä, kuin vain QwikFlash:iä
  - Prosessori ja virransyöttö tarvitaan aina
  - Näyttö- ja syöttölaitteet toimivat samoilla periaatteilla alustasta riippumatta
  - Samoin esim. D/A muunnin, jos käytössä on SPI-väylä


GND
H1
A B E2 B1 B0 C2
DAC AN7 INT1 INT0 CCP1
VDD
C1
R1
C7
C8
SW2
D4 D5 D6
GND1
C6
U2
RESET
CON1
J1
D2
C9
C2
U1
C4
C10
C12
10.000MHz
C3
C11
C13
CON2
SW1
D1
D3
R2
REG1
R9
D8
D7
R10
R11
R12
R13
C1
GND
CCP2 H3
CON3
C5
C14
RPG1
POT1
SW3
RST
A0
A1
A2
A3
A4
A5
E0
E1
E2
VDD
GND
Xin
Xou
C0
C1
C2
C3
D0
D1
B7
B6
B5
B4
B3
B2
B1
B0
VDD
GND
D7
D6
D5
D4
RX
TX
C5
C4
D3
D2
QwikFlash
R14
C17
R15
LCD1
BYTE=
00010011

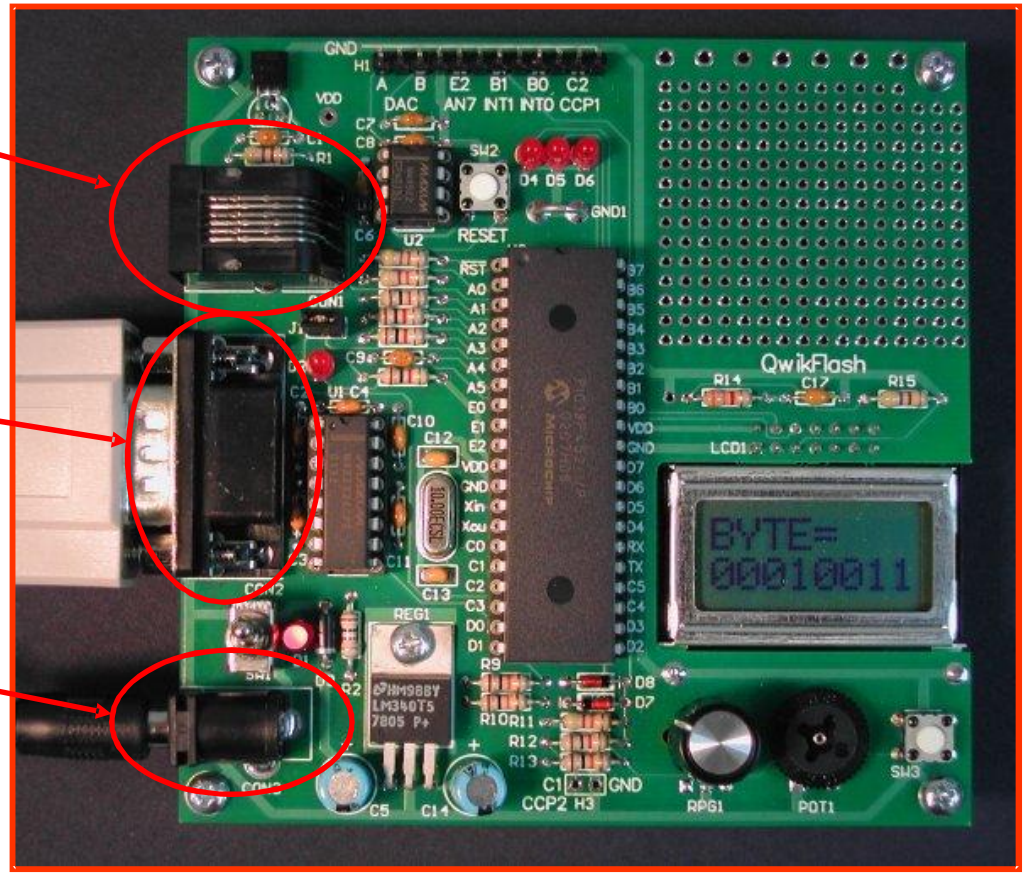
Prosessori

- Reset -kytkin
- PIC18F452
  - Ominaisuudet esitelty aiemmin
  - Ohjelmointi joko ICD2 –modulilla tai sarjaportin kautta terminaali-emulaattorilla
- 10 MHz kide
  - =>2.5 MHz sisäinen taajuus



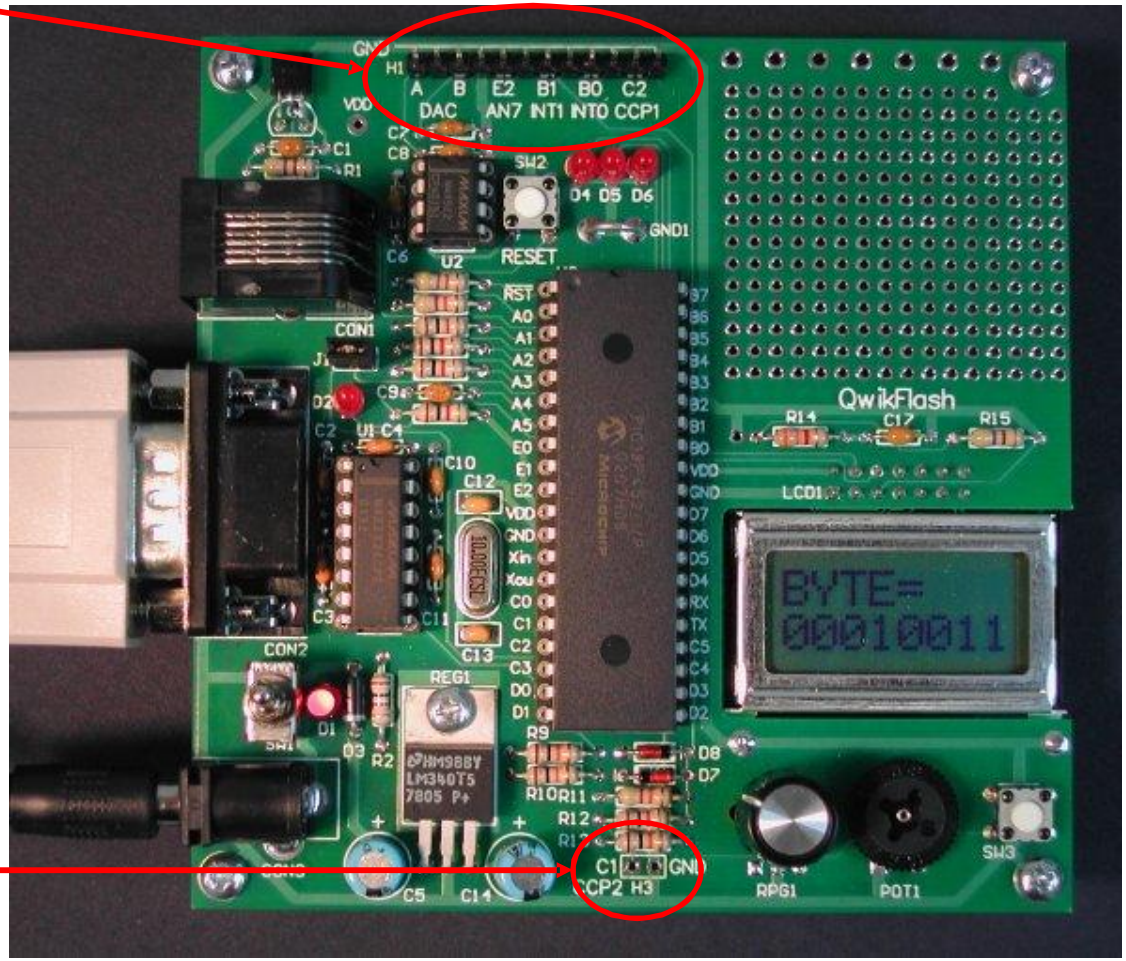
Liittimet

- Ohjelmointi / debuggaus
  - ICD2 –modulilla
- Ohjelmointi / debuggaus
  - Terminaaliemulaattorilla
- Virtaliitin
  - 9 V DC
  - Meidän muuntajiin vihreä liitin ja ”TIP” ja ’+’ –merkit samalle puolelle



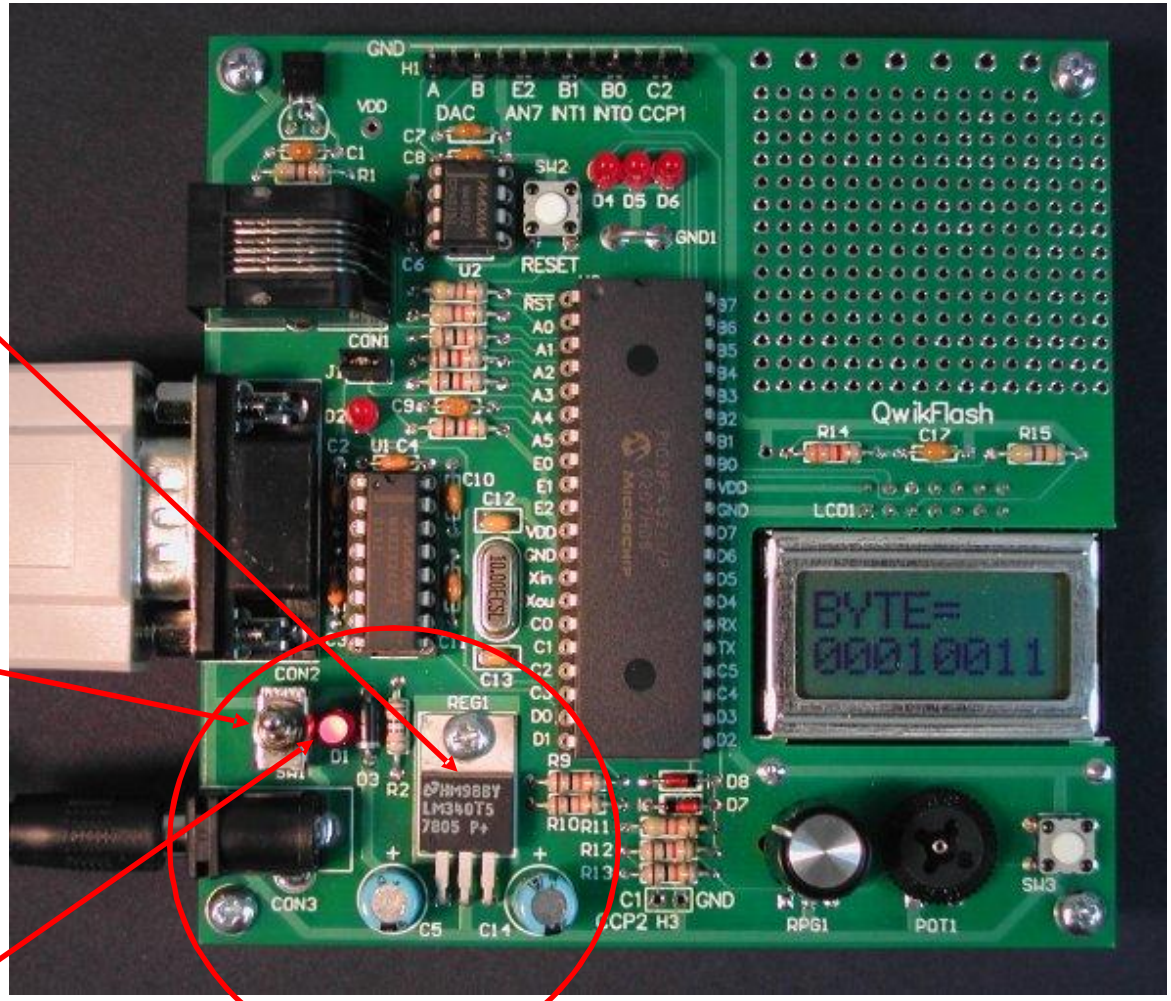
Liittimet

- Liitinrima 1
  - 2 kanavaa D/A -muuntimelta
  - E2 / AN7
  - B1 / INT1
  - B0 / INT0
  - C2 / CCP1
- Liitinrima 2
  - Laudan pohjassa, näissä laudoissa sitä ei ole
- Liitinrima 3
  - C1 / CCP2



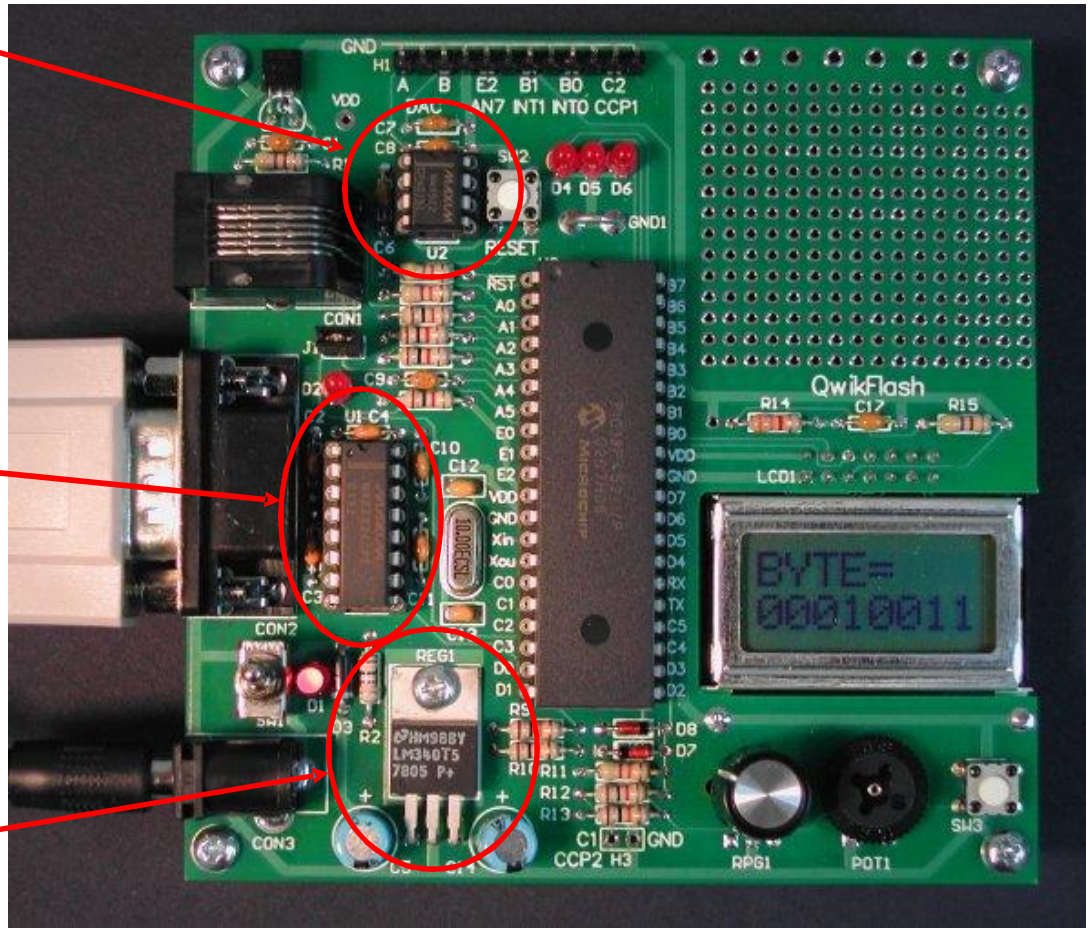
Virransyöttö

- Sisään 9 V DC
  - Ei tarvitse olla reguloitu
- Regulaattori vakaa jännitteen 5 V:hen
- Kondensaattorit ennen ja jälkeen regulaattoria
- Virtakytkin vasta regulaattorin ja kondensaattoreiden jälkeen
  - Kytkee VDD:n joko maahan tai 5 V:hen
- Power-on LED



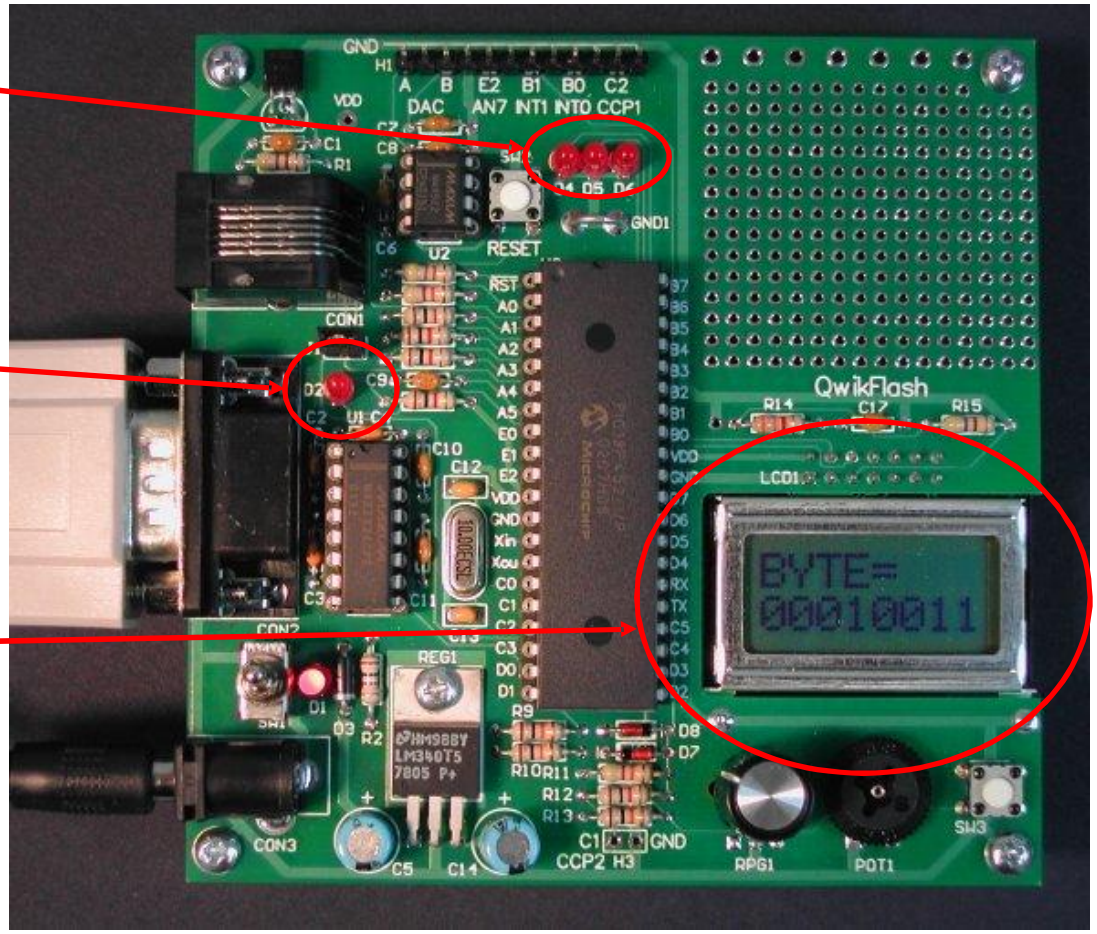
Oheispiirit

- D/A –muunnin
  - 2 kanavaa
  - 8 bittinen
  - SPI-väylässä
- MAX232A
  - Muuntaa 0 – 5 V signaalit RS232 mukaisiksi, eli noin -12 V – +12 V
- Regulaattori
  - 5 V



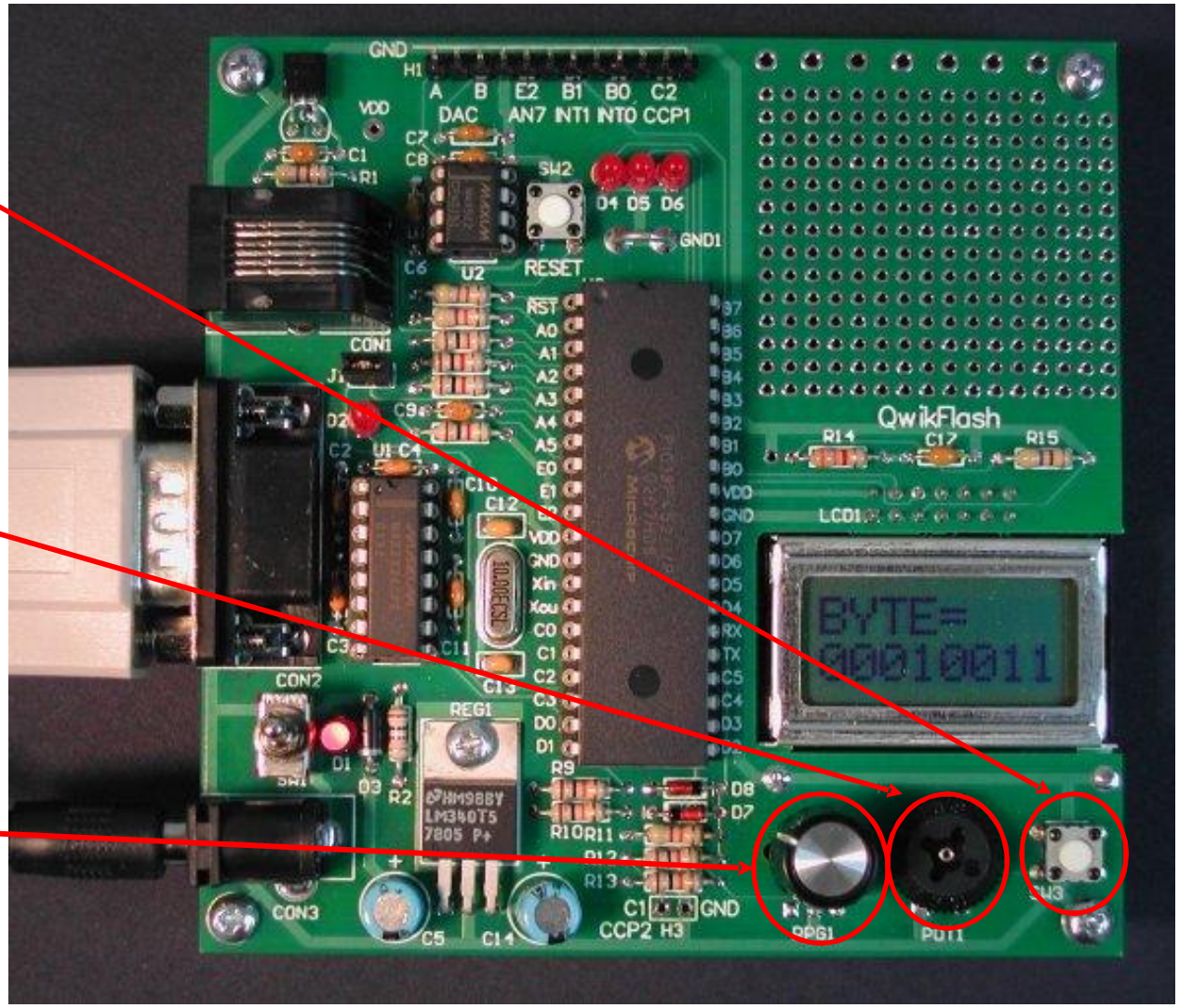
Näyttölaitteet

- 3 LEDiä
  - RA<1:3>
  - Palaa, kun ’1’
- Alive LED
  - RA<4>
  - Palaa, kun ’0’
- 8x2 LCD –näyttö
  - ASCII merkit
  - 8 itse koodattavaa merkkiä



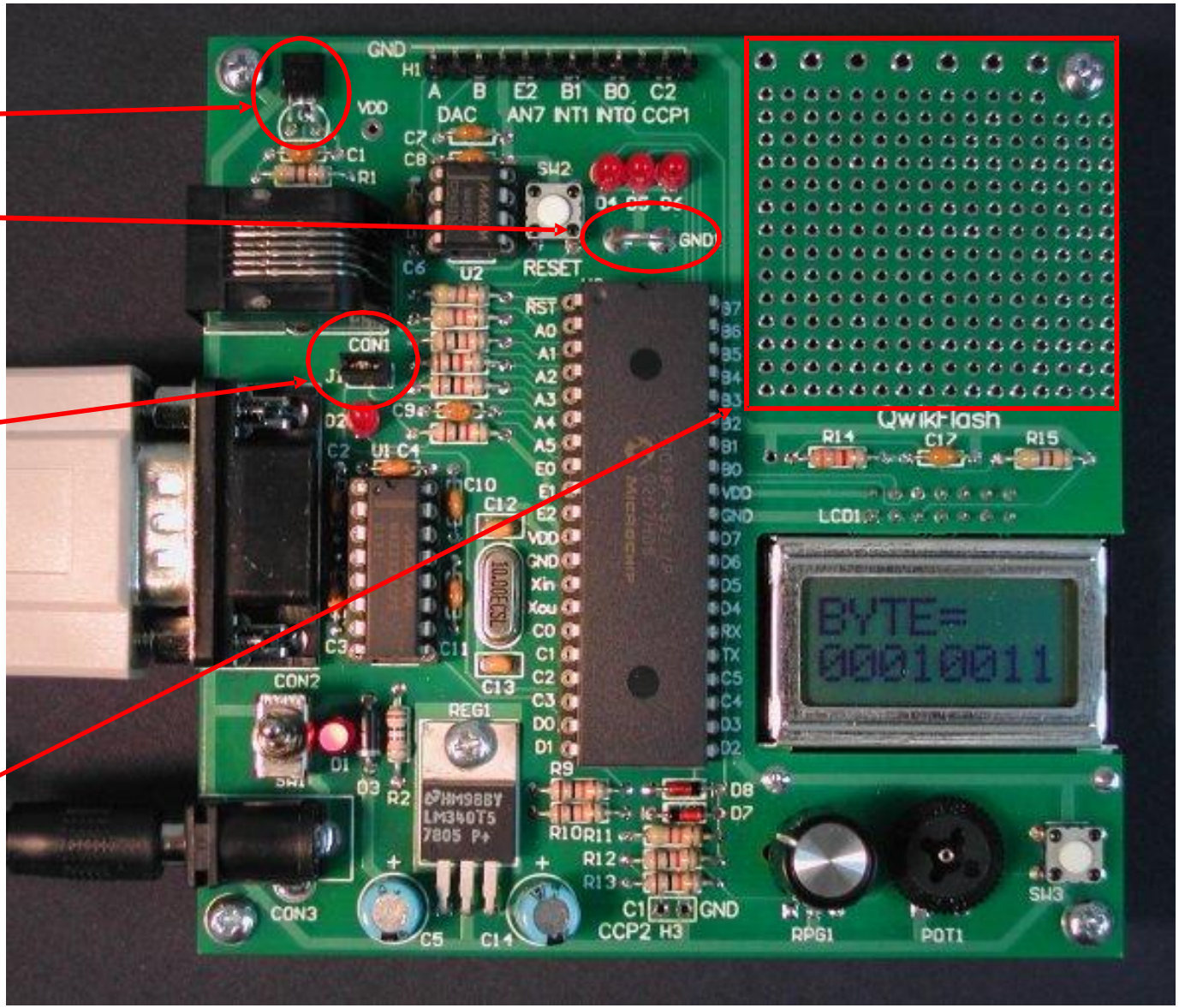
Syöttölaitteet

- Painokytkin
  - 10k ylösvetovastus tai suora kytkentä maahan
  - RD<3>
- Potentiometri
  - Säätövastus VDD:n ja maan välillä
  - RA<5> / AN4
- Kiertosäädin
  - 24 asentoa / kierros
  - RD<1:0>



Lisäksi

- Lämpötila-anturi
- Maakontakti
  - Mittauksia varten
- Jumpperi 1
  - Valitsee, käytetäänkö ICD2- vai sarjaliitintä
- Kytkentä alue
  - Oheiskompo-nenteille



D/A -muunnin

- MAX522CPA
  - 8 bittinen, 2 kanavainen, molemmat kanavat liitinrimassa 1
- Kommunikaatio SPI väylällä
  - Paluukanavaa ei kytketty QwikFlash –laudalla
  - RC<5>=SDO
  - RC<4>=SDI (ei kytketty)
  - RC<3>=SCK
  - RC<0>=#Chip Select

D/A -muunnin

- Oletus: siirrettävä data muuttujassa ”BYTE”

```
;;;;;;; SPItransfer subroutine
SPItransfer
        bcf    PIR1,SSPIF              ; Clear PIR1,SSPIF to ready for
                                       ; transfer.
        movff  BYTE,SSPBUF             ; Initiates write when anything is
                                       ; placed in SSPBUF. Byteà SSPBUF
Wait_SPI                               ; Wait until transfer is finished.
        btfss PIR1,SSPIF               ; Testaa lippua
        bra   Wait_SPI                 ; Ei asettunut, hyppää Wait_SPI
        movff  SSPBUF,SPIRECEIVE       ; If result is desired, it is
                                       ; stored in SPIRECEIVE.
        return
```

D/A -muunnin

- Käyttö:

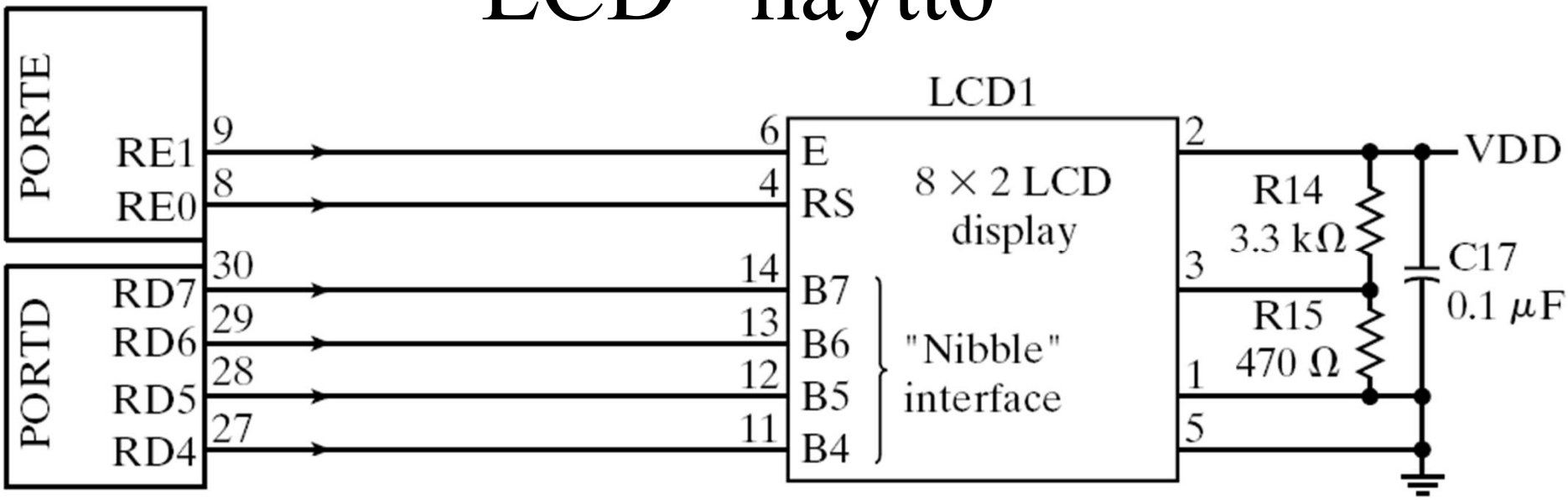
```
;;;;;;; SPItransfer usage example
        bcf    PORTC,RC0                ; Clear PORTC,RC0 to select DAC
        movlf  0x21, BYTE               ; Move 0x21 to BYTE (select DAC-A)
                                        ; (0x22 selects DAC-B)
        rcall  SPItransfer              ; Call SPItransfer subroutine
        movff  Data, BYTE               ; The value Data is copied to BYTE
        rcall  SPItransfer              ; Call SPItransfer subroutine
        bsf    PORTC,RC0                ; Set RC0 to release DAC

;;;;;;; Settings (in Initial)
        MOVLF B'11010010', TRISC        ; Set I/0 for PORTC.
        MOVLF B'00100000', SSPCON1      ; Sets up the SPI interface for the
        MOVLF B'11000000', SSPSTAT      ; DAC. (Use with 10/2.5 MHz clock)
```

LCD –näyttö

- Näytönohjain yhteensopiva Hitachi HD44780:n kanssa
  - Käytetään lähes kaikissa LCD –näytöissä
  - Dataväylä valittavissa 4 tai 8 –bittiseksi, QwikFlash –levyllä se on 4 bittinen
- Näytön koko 8 merkkiä per rivi, 2 riviä
- Osaa näyttää vakio ASCII –merkit
  - Poislukien: ‘\’, ‘~’ ja “delete”
  - Lisäksi 8 käyttäjän määritettävissä oleva merkkiä ja 96 japanilaista symbolia

LCD –näyttö



- 4 bittinen dataväylä, portin D 4 ylintä bittiä
- 2 bittinen kontrolliväylä, portin E 2 alinta bittiä
- Vastukset R14 ja R15 kytkeytyvät kontrastin säätöön
  - Niiden asemesta voitaisiin käyttää säätövastusta, jos halutaan muuttaa kontrastia olosuhteiden mukaan
- Näytön pinnit <7:10> ei kytketty prosessoriin
  - Näissä olisi loput 4 bittiä dataa, jos 8 bittinen dataväylä

LCD –näyttö

- Initialisointi:

```
;;;;;;; InitLCD subroutine;
; Initialize the Optrex 8x2 character LCD.
; First wait for 0.1 second, to get past display's power-on reset time.
; Requires POINT –macro and LoopTime –subroutine to be defined

InitLCD
        MOVLF  10,COUNT                 ; Wait 0.1 second.
L2
        rcall  LoopTime                 ; Call LoopTime 10 times.
        decf   COUNT,F
        bnz    L2
RL2
        bcf    PORTE,0                  ; RS=0 for command.
        POINT  LCDstr                   ; Set up table pointer to
                                        ; initialization string.
        tblrd*                          ; Get first byte from string into TABLAT.
L3
          bsf    PORTE,1                ; Drive E high.
          movff  TABLAT,PORTD           ; Send upper nibble.
          bcf    PORTE,1                ; Drive E low so LCD will process input.
          rcall  LoopTime               ; Wait ten milliseconds.
          bsf    PORTE,1                ; Drive E high.
          swapf  TABLAT,W               ; Swap nibbles.
          movwf  PORTD                  ; Send lower nibble.
          bcf    PORTE,1                ; Drive E low so LCD will process input.
          rcall  LoopTime               ; Wait ten milliseconds.
          tblrd+*                       ; Increment pointer and get next byte.
          movf   TABLAT,F               ; Is it zero?
          bnz    L3
RL3
        return
```

LCD –näyttö

- Initialisointi jatkuu:

```
;;;;;;; POINT macro definition;
POINT   macro  stringname
        MOVLF  high stringname, TBLPTRH
        MOVLF  low stringname,  TBLPTRL
        endm

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;; Constant strings (at the end of user code)
LCDstr  db  0x33,0x32,0x28,0x01,0x0c,0x06,0x00  ; Initialization string
                                                ; for LCD.0x00=loppu

; 0x33, 0x32: setup 4-bit interface
; 0x28:       setup 2 line displaymode
; 0x01:       clear the display
; 0x0c:       turn off cursor, turn on display
; 0x06:       increments cursor automatically
; 0x00:       terminating symbol for the string (null)
```

LCD –näyttö

- Käyttö:

```
;;;;;;;;DisplayC subroutine;;;;;;;;;;;;;;
;
; This subroutine is called with TBLPTR containing the address of a constant
; display string.  It sends the bytes of the string to the LCD.  The first
; byte sets the cursor position.  The remaining bytes are displayed, beginning
; at that position until null is encountered (0x00)
; This subroutine expects a normal one-byte cursor-positioning code, 0xhh, or
; an occasionally used two-byte cursor-positioning code of the form 0x00hh.

DisplayC
        bcf  PORTE,0                    ; Drive RS pin low for cursor-positioning.
        tblrd*                          ; Get byte from string into TABLAT.
        movf TABLAT,F                   ; Check for leading zero byte.
        bnz L5
        tblrd+*                         ; If zero, get next byte.
L5
        bsf   PORTE,1                   ; Drive E pin high.
        movff TABLAT,PORTD              ; Send upper nibble.(Only upper 4 are outputs)
        bcf   PORTE,1                   ; Drive E pin low so LCD will accept nibble.
        bsf   PORTE,1                   ; Drive E pin high again.
        swapf TABLAT,W                  ; Swap nibbles.
        movwf PORTD                     ; Write lower nibble.
        bcf   PORTE,1                   ; Drive E pin low so LCD will process byte.
        rcall T40                       ; Wait 40 usec. (T40 not defined here)
        bsf   PORTE,0                   ; Drive RS pin high for displayable
                                       ; characters.
        tblrd+*                         ; Increment pointer, then get next byte.
        movf  TABLAT,F                  ; Is it zero?
        bnz L5
        return
```

LCD –näyttö

- Kursorin asemointi näytöllä
  - RS –pinni (RE0) pitää ajaa alas ennen komennon lähettämistä, muuten se tulkitaan näytettäväksi merkiksi
  - Ylärivin merkit osoitetaan numeroilla 0x80-0x87
    - 0x80 on vasen yläkulma
  - Alarivin merkit osoitetaan numeroilla 0xc0-0xc7
    - 0xc0 on vasen alakulma

Lisää tietoa merkki LCD:stä
http://www.ekenrooi.net/lcd/lcd0.shtml

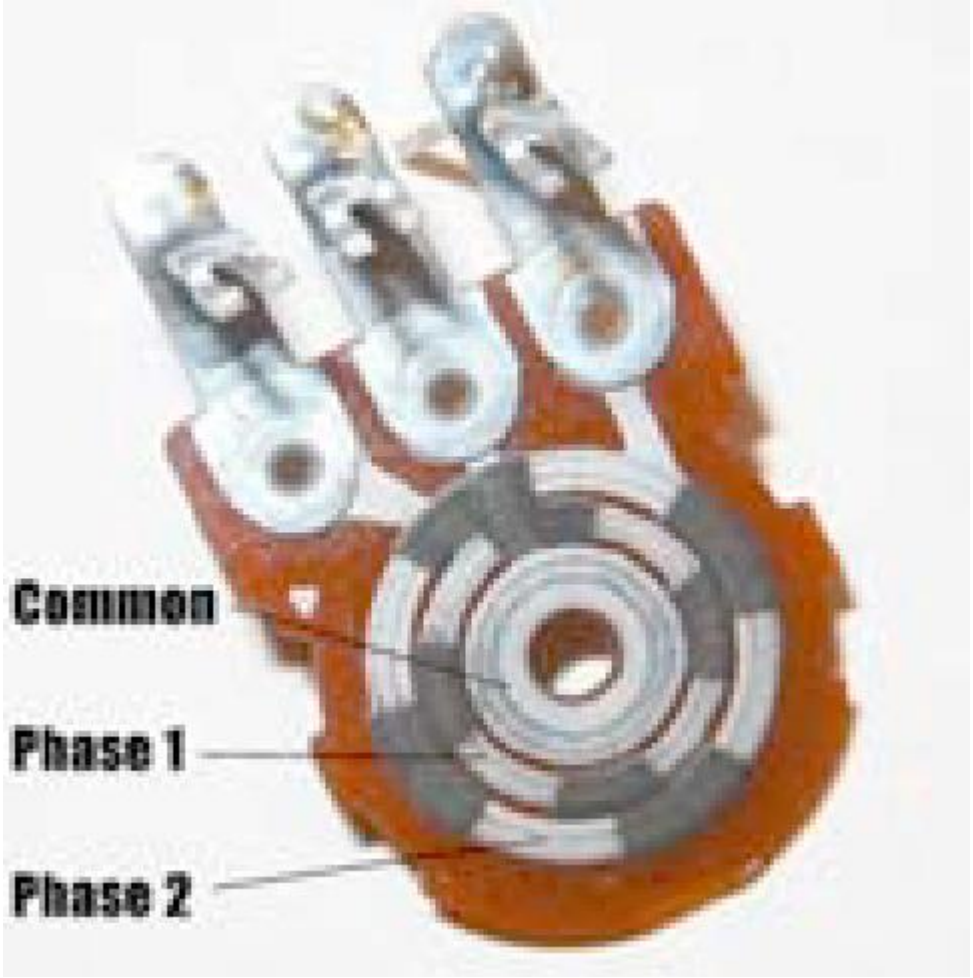
LCD –näyttö

- Omat merkit luodaan kirjoittamalla merkkiä vastaava bittikuvio näytönohjaimelle
  - 5 pikseliä leveä ja 8 pikseliä korkea
  - Tarkempia tietoja kirjasta tai näytön datalehdistä
- Usein käytettyjä symboleja:
  - ° = 0xdf, α = 0xe0, β = 0xe2, π = 0xf7, Σ = 0xf6, Ω = 0xf4
- Merkkijonoja voidaan kirjoittaa ohjelmamuistiin, esim:

```
TITLE1  db  0x80,'E','T','T','_','1',' ',0x00  ; Declaration of name strings on LCD
TITLE2  db  0x80,'B','y',':','_',' ',' ',0x00  ; All strings must be of same lenght,
TITLE3  db  0x80,'t','e','a','n','s','a',0x00  ; else the last letters of previous
TITLE4  db  0x80,'J','a','n','0','5',' ',0x00  ; line will remain on the screen...
```

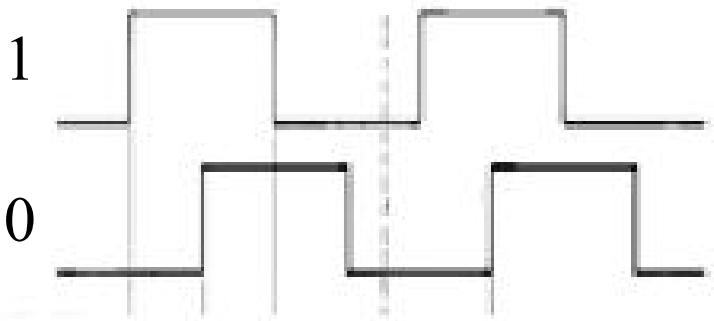
RPG

- 3 pinniä
  - ”Common” kytketään maahan
  - Kahteen muuhun heikko ylösveto
  - Kontaktiliuskat kiertävät kuvassa näkyviä renkaita pitkin, kun nuppia käännetään
  - Kuvassa kirkkaat alueet yhdistävä ja tummat eristävät
  - Tarvitaan vain kaksi signaalijohdinta
  - Ei hyppää askelten yli, vaikka esiintyisi ”bounce”



RPG

- Bitit PORTD<1:0> muuttuvat:
  - Myötäpäivään käännettäessä
    - $00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$
  - Vastapäivään käännettäessä
    - $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$
  - 2 bittinen Grayn koodi
    - Vähentää virheen mahdollisuutta rajapinnoilla (esim. $11 \rightarrow 00$ mahdoton)
  - Oheisessa kuvassa käännetään vastapäivään



RPG

- Käytetään usein:
  - Parametrien syöttämiseen
  - Äänenvoimakkuuden säätämiseen
  - Myös mittauksiin
    - Etäisyys
    - Akselin kääntymä / kierrosnopeus
- Joskus otetaan huomioon myös nopeus
  - Esim. syötettäessä lukuarvoa, nopea pyöritys hyppää 5 askelta kerralla ja hidas etenee yksi kerrallaan

Painokytkin

- Portissa D (RD<3>)
- Heikko ylösvetovastus, painettuna kytkee maahan
- Voidaan käyttää ”kaksitoimisena”
  - Lyhyt näpäytys (<300 ms) tai pitkä painallus (>300 ms)
- Toiminnan ohjaamiseen
  - Start / stop
  - RPG:llä valitun arvon hyväksyminen
  - Etc.
- Debounce yleensä pakollinen (mekaaninen kytkin “hyppii”)

Potentiometri

- Säätövastus (5 kΩ)
  - Reunat kytketty maahan ja VDD:hen
  - Keskimmäinen kontakti kytketty AN4 / RA5:een
- Luetaan PIC:in sisäisellä A/D –muuntimella
- Käytetään:
  - Nopeuden / äänenvoimakkuuden säätöön
  - Arvojen syöttämiseen
  - Kalibrointiin

Lämpötila-anturi

- LM34DZ
  - Analoginen ulostulo
- Kytketty AN0 / RA0:aan
- Luetaan PIC:in sisäisellä A/D –muuntimella
- Kalibrointi (vain karkeasti oikea...)
  - Huoneen lämmössä noin 21 astetta
  - Sormella lämmitettäessä noin 34 astetta
  - Kalibroinniksi saadaan:
    - T(C) = lämpötila Celsiuksina
    - T(m) = mitattu lämpötila
    - Jokainen anturi tarvitsee oman kalibroinnin, jos halutaan oikeasti tarkkoja mittauksia

$$T_{(C)} = \frac{(T_{(m)} - 92) * 5}{12}$$

QwikBug

- Taustalla oleva ohjelma, joka:
  - Toimii BootLoader:ina
    - Ottaa käyttäjän ohjelman vastaan
    - Käytetään demossa
  - Mahdollistaa debuggauksen
    - Voidaan asettaa BreakPoint:eja
    - Voidaan tarkastella muistin sisältöä suorituksen aikana
    - Voidaan muokata muistin sisältöä suorituksen aikana
    - Käytetään demossa
  - Liittää PIC:in sisäisen debuggausmodulin sarjakaapelilla PC:hen
    - Luokan koneissa TeraTerm:issä valmiina oikeat tietoliikenneasetukset (19200, 8, no parity, 1 stop, no flow control, joissain koneissa tarvitaan vielä ”transmit delay” johon sopiva arvo on 40 ms/line)

QwikBug

- Ohjelman lataus: (TeraTerm[Pro]:iä käyttäen)
  - F7: pysäytetään prosessori
  - F3: tyhjätään ohjelmamuisti ja valmistellaan tiedonsiirto
  - Download: raahataan *.hex terminaali-ikkunaan
  - F2: resetöi prosessorin
  - F7: aloittaa ohjelman suorituksen
- Ohjelmaa ladattaessa konfiguraatiobitit ignoroidaan, koska konfiguraatio on asetettu jo samalla kun QwikBug on syötetty laudalle (ei voida muuttaa!)

QwikBug

- Debuggaus:
  - F4: näyttää PC, W, Z, C, N ja FSR:t, lisäksi voidaan määrittää omia muistiosoitteita, jotka näytetään
  - F5: breakpointin määrittely / poisto / tarkistus osoitteen mukaan, lähdekoodin kohtaa vastaava osoite löytyy helpoiten *.lst –tiedostosta
  - F6: lisää muistiosoitteen listaan, jotka näytetään F4:llä ja aina suorituksen keskeytyessä (F7 tai breakpoint)
  - F8: suorittaa ohjelmaa yksi käsky kerrallaan (step)
  - F9: muistin sisällön muokkaus
    - F4, F6 ja F9 operoivat datamuistissa
  - F1: apua, antaa pienen käyttöohjeen

QwikBug

- Kuluttaa joitain resursseja:
  - Ohjelmamuistin yläpäästä menetetään hieman, mutta koska sitä on käytössä 32k, ei yleensä haittaa
  - Paluuosoitepinoon pitää varata muutama paikka QwikBug:ia varten, on 31 paikkaa, joten tuskin tulee ongelmia
  - QwikBug varaa pankin 5 datamuistista kokonaan omaan käyttöönsä, käyttäjälle jää vielä 1280 tavua
  - QwikBug käyttää UART:ia, joten se ja I/O pinnit RC<7:6> menetetään
  - Myös I/O pinnit RB<7:6> menetetään debuggaus modulin takia