

Software behorende bij artikel "Naar de knoppen & LCD op I2C" van D. Jansen

Knopje 1.

```
K$regfile = "m328def.dat"      ' instelling type micro controller

$crystal = 16000000           ' kristal frequentie (16 MHz)
'Config Pind.7 = Output      ' luidsprekertje, niet noodzakelijk
DdrD.2 = 0                    ' maak van de uitgang een ingang (bij opstarten is dit de standaard conditie)
                              ' alleen noodzakelijk als de poort eerder als uitgang was gedefinieerd.
Portd.2 = 1                   ' schakel de interne pull-up weerstand in, nodig als er geen externe pull-up is.
Sound Portd.7 , 800 , 800    ' piepje, niet noodzakelijk, wel leuk.

Do                             ' begin programma
  If Pind.2 = 0 Then           ' als de knop is ingedrukt
    Toggle Portb.5            ' schakel het reeds aanwezige ledje in/uit
    Waitms 100' wacht even
  End If
Loop                           ' herhaal
End                             ' einde
```

Knopje 2.

```
$regfile = "m328pdef.dat"     ' micro controller
$crystal = 16000000           ' kristal 16 MC
$baud = 19200                 ' communicatie met PC
Dim A As Byte : A = 1         ' telt het aantal malen inschakelen
Dim B As Byte : B = 2         ' bij elke vierde uitschakelpuls, LEDje aan
Knopje Alias Pind.2          ' het knopje op interrupt poort D.2 heet nu: knopje
Knopje = 1                    ' inschakelen pull-up weerstand
Ledje Alias Portb.5          ' ledje benoemen
Declare Sub Knopje_isr       ' niet noodzakelijk voor interrupt routines
Config Ledje = Output        ' poort als uitgang instellen
Enable Interrupts            ' interrupts inschakelen
Enable Int0                  ' interrupt -0- inschakelen
Config Int0 = Change         ' interrupt bij elke op en neergaande flank
                              ' Rising voor opgaande flanken en Falling voor neergaande flanken
On Int0 Knopje_isr           ' bij interrupt naar deze routine
Sound Portd.7 , 800 , 800    ' piepje
Print "{012}"                ' schoon scherm bij aanvang programma
Do                            ' hoofdroutine
  B = A Mod 4                 ' berekenen 4e indruk
  If B = 1 And A > 3 Then     ' als deling 'rest -1-' oplevert en na 3e indruk
```

```

                                ' om te voorkomen dat het ledje bij aanvang aan is, moet A>3 zijn.
    Ledje = 1                                ' na elke vierde knopindruk ledje aan zetten
Else
    Ledje = 0                                ' in elke andere situatie dooft het ledje
End If
Loop                                        ' herhaal
End

Knopje_isr:                                ' interrupt routine
Disable Int0                                ' tijdens interrupt blokkeer interrupts
Delay                                        ' enige 1µs traagheid tbv ontendersen
If Knopje = 0 Then                            ' als de knop is ingedrukt, is de ingang laag, neergaande flank.
    Print " knopje "; A ; " maal ingedrukt"
End If
If B = 0 Then
    Print " ledje aan"
End If
If Knopje = 1 Then                            ' als het knopje wordt losgelaten, is er een opgaande flank.
    Print " knopje "; A ; " maal losgelaten"
    A = A + 1                                ' teller wordt met 1 opgehoogd
End If
Enable Int0                                ' interrupt worden weer ingeschakeld
Return

```

Knopje 3.

```

$regfile = "m328pdef.dat"                    ' micro controller
$crystal = 16000000                          ' kristal 16 MC
Config Scl = Portc.5                          ' I2C klok
Config Sda = Portc.4                          ' I2C data
I2cinit                                       ' I2C initialisatie
$lib "Lcd_i2c.lib"                            ' de I2C LCD driver uit de bibliotheek
Config I2cdelay = 1                          ' 1 µs vertraging, zonder doet hij het ook
Const Pcf8574_lcd = &H40                     ' schrijfadres van de I2C chip
Dim _lcd_e As Byte                            ' noodzakelijk voor LCD beheer
_lcd_e = 128                                  ' bovenste twee regels van het LCD selecteren
Enable Interrupts                             ' interrupts inschakelen
Enable Int0                                   ' interrupt -0- inschakelen
Config Int0 = Change                          ' interrupt bij elke op en neergaande flank
On Int0 Statusmelding_isr                    ' bij interrupt naar deze routine
Dim A As Byte : A = 1                         ' telt het aantal malen inschakelen
Dim B As Byte : B = 2                         ' bij elke vierde uitschakelpuls, LEDje aan
Sound Portd.7 , 800 , 800                    ' piepje

```

```

Cls                                     ' LCD schermpje schoonmaken
Lcd "Knop test met interrupt"          ' tekst op LCD

Do                                     ' hoofdroutine
  B = A Mod 4                          ' controle op vierde knopindruk teller delen door 4
  If B = 1 And A > 3 Then               ' als deling 'rest -1-' oplevert en na 3e indruk
    Portb.5 = 1                        ' na elke vierde knopindruk ledje aan zetten
  Else
    Portb.5 = 0                        ' in elke andere situatie dooft het ledje
  End If
Loop                                   ' herhaal
End

Statusmelding_isr:                    ' interrupt routine
Disable Int0                           ' tijdens interrupt blokkeer interrupts
Delay                                   ' enige traagheid tbv ontendersen
If Pind.2 = 0 Then                     ' als de knop is ingedrukt, is de ingang laag
  Cls                                   ' scherm schoon
  Locate 1 , 1                         ' positie 1 op regel 1 (eerst positie, dan regel)
  Lcd " knop " ; A ; " x in"           ' tekst met aantal indrukken
End If
If B = 0 Then
  Locate 3 , 2                          ' regel 2 positie 3
  Lcd "LEDje aan"                      ' tekst
End If
If Pind.2 = 1 Then                     ' het knopje wordt losgelaten, opgaande flank
  If B <> 0 Then Cls                    ' als het LEDje uit moet zijn, tekst weg
  Locate 1 , 1                          ' positie 1 regel 1
  Lcd " knop " ; A ; " x los"          ' tekst
  A = A + 1                             ' teller wordt met 1 opgehoogd
End If
Enable Int0                             ' interrupt 0 inschakelen
Return

```

Knopje 4.

```

$regfile = "m328pdef.dat"              ' de micro controller
$crystal = 16000000                    ' kristalfrequentie
$baud = 19200                          ' communicatie snelheid
Print "{012}"                          ' scherm schoonmaken
Dim Knopje_1 As Integer                 ' knop indruk teller 1
Dim Knopje_2 As Integer                 ' knop indruk teller 2
Enable Interrupts                       ' interrupts inschakelen

```

```

Enable Pcnt1          ' PC interrupts van poort C1...6 in register 1.
    ' Deze twee knopjes zitten op Pc Interrupt Register 1
    ' knopje -1- op C.2 interrupt 10 en knopje -2- op C.3 interrupt 11
Pcmsk1 = 12          ' twee ingangen 10 en 11 openen locatie 8 en 4. (4+8=12)
Portc = &B00001100  ' pull up C 2&3 inschakelen (vergelijkbaar met hierboven)
On Pcnt1 Knopjes_isr ' in deze routine uitzoeken welke knop is ingedrukt.

Do                  ' hoofdroutine bevat kijkt alleen naar knoptellers
    Print "Knopje 1 is "; Knopje_1; " maal ingedrukt "; "Knopje 2 is "; Knopje_2; " maal ingedrukt"
    Wait 1          ' wacht een seconde
Loop               ' herhaal
End

Knopjes_isr:
    If Pinc.2 = 0 Then Incr Knopje_1      ' knopje -1- ingedrukt
    If Pinc.3 = 0 Then Incr Knopje_2      ' knopje -2- ingedrukt
Return

```

Voorbeeld van schrijven naar LCD

```

' LCD aansturen via I2C

$regfile = "m328pdef.dat"      ' gebruikte processor Atmega328p op Arduino nano
$crystal = 16000000           ' kristal op bord is 16 Mhz
$baud = 19200                 ' baud rate communicate. 19200,N,8,1
Config Scl = Portc.5          ' I2C klok
Config Sda = Portc.4          ' I2C data
I2cinit                       ' I2C initialisatie

$lib "Lcd_i2c.lib"            ' de I2C LCD driver uit de bibliotheek
Config I2cdelay = 1           ' 1 µs vertraging, zonder doet hij het ook
Const Pcf8574_lcd = &H40      ' schrijfadres van de I2C chip
Dim N As Byte                 ' tellertje voor het heen en weer schuiven van de tekst
Dim _lcd_e As Byte           ' noodzakelijk voor LCD beheer
_lcd_e = 128                  ' bovenste twee regels van het LCD selecteren
                              ' de driver gaat uit van een 4-regelig LCD
                              ' _lcd_e = 128 select E1,
                              ' 64 select E2,
                              ' 192 select both (voor CLS of DefLCDChar.)
Config Lcd = 16 * 2           ' selecteer twee regelig LCD met 16 karakterplaatsen.
                              ' andere mogelijkheden: 16 * 4 ; 20 * 4 ; 20 * 2 ; 16 * 1a
                              ' 16*1a zou 16*2 moeten zijn, maar dat werkt bij mij niet.
Cls                            ' LCD schermje schoonmaken
_lcd_e = 192                  ' Selecteer beide helften om een eigen karakter te bouwen.
' Je kan maximaal 7 karakters maken hier als voorbeeld het kwadraat 2 teken.
' Een hulpmiddel vind je op: http://lcd-font-maker.software.informer.com/download/
' Beschouw een karakter als een matrix van 7 kolommen en 8 rijen.
' Elke rij wordt binair gevormd en decimaal weergegeven.
Deflcdchar 1, 24, 4, 8, 16, 28, 0, 0, 0      ' eerste karakter het 2 teken
_lcd_e = 128                          ' Terug naar normale stand, twee bovenste regels.
Cls                                    ' Noodzakelijk om het karakter genereren af te sluiten.

Lcd "Bascom I"; Chr(1); "C-LCD."           ' op regel -1- de tekst: Bascom I2C-LCD.
Wait 1                                    ' wacht een seconde
Lowerline                                  ' selecteer regel 2
Lcd "Heen en weer."                       ' na één seconde op regel -2- de tekst: Heen en weer

```

```

Wait 2                                ' wacht 2 seconden

For N = 1 To 10                        ' schuif de boel 10 posities naar rechts
  Shiftlcd Right
  Waitms 200                          ' korte pauze
Next
For N = 1 To 20                        ' schuif nu de boel 20 posities naar links
  Shiftlcd Left
  Waitms 200
Next
For N = 1 To 11                        ' schuif nu weer naar rechts.
  Shiftlcd Right
  Waitms 200
Next
End

```

De listing van: Lcd_i2c.lib

```

copyright = Kent Andersson
www       = none
email    = kent@smmab.se
comment  = I2C LCD driver
libversion = 1.02
date     = 31 march 2002
statement = You are free to use this code any way you like, if you are able to optimize
statement = it better, please send me an update on my e-mail.
history  = No known bugs.

;define a constant named PCF8574_LCD pointing to the i2c address
;dimension _lcd_e as byte to control the E-lines (4 lines LCD:s)
;_lcd_e should have one of the following values
;128 to enable E1, 64 to enable E2, 192 to enable both E1 and E2 (cls, deflcdchar)
;Connect the following pins from PCF8574 to LCD
;P0 - D4
;P1 - D5
;P2 - D6
;P3 - D7
;P4 - RS
;P5 - RW (not used, set to 0 to ground for write)
;P6 - E2 (on 1 or 2 line display nc)
;P7 - E1

[_Init_LCD]
_Init_LCD:
*BASIC: waitms 50
ldi r16,&hc0 ; this is to make the initialization on both halves of a 4-line LCD
Ldi _temp1, &h03 ; at init-time I call all routines before _lcd_e is loaded into r16
Rcall _Send_to_LCD
*BASIC: waitms 4
Rcall _Send_to_LCD
Rcall _Send_to_LCD
Ldi _temp1, &h02
Rcall _Send_to_LCD
Ldi _temp1, &h28
Rcall _Write_lcd_byte ;RS flag should to be 0, so jump directly to write byte
Ldi _temp1, &h08
Rcall _Write_lcd_byte
Ldi _temp1, &h0c
Rcall _Write_lcd_byte
Ldi _temp1, &h01
Rjmp _Write_lcd_byte
[END]

[_Write_lcd]
_Write_lcd:
*lds r16,{_lcd_e}
sbr r16,&h10 ;set RS flag (4 high bits is used as control)

```

```

rjmp _write_lcd_byte
_Lcd_Control:
* lds r16,{_lcd_e}
_write_lcd_byte:
push r24
Swap _temp1
Rcall _Write_lcd_Nibble
pop r24
_Write_lcd_nibble:
Cbr _temp1,&HF0 ; E1, E2, RW, RS low
_Send_to_LCD:
push r26
push r27
ldi r25,3
_send_3_times:
*BASIC: i2csend pcf8574_LCD,_temp1
eor _temp1,R16 ; Xor E1 and/or E2 to set them
; @genus(250)
ldi r26,0
_LCD_delay:
dec R26
brne _LCD_delay
Dec R25
Brne _send_3_times
eor _temp1,R16 ; Xor to restore to initial state (even number of times)
pop r27
pop r26
ret
[END]

```