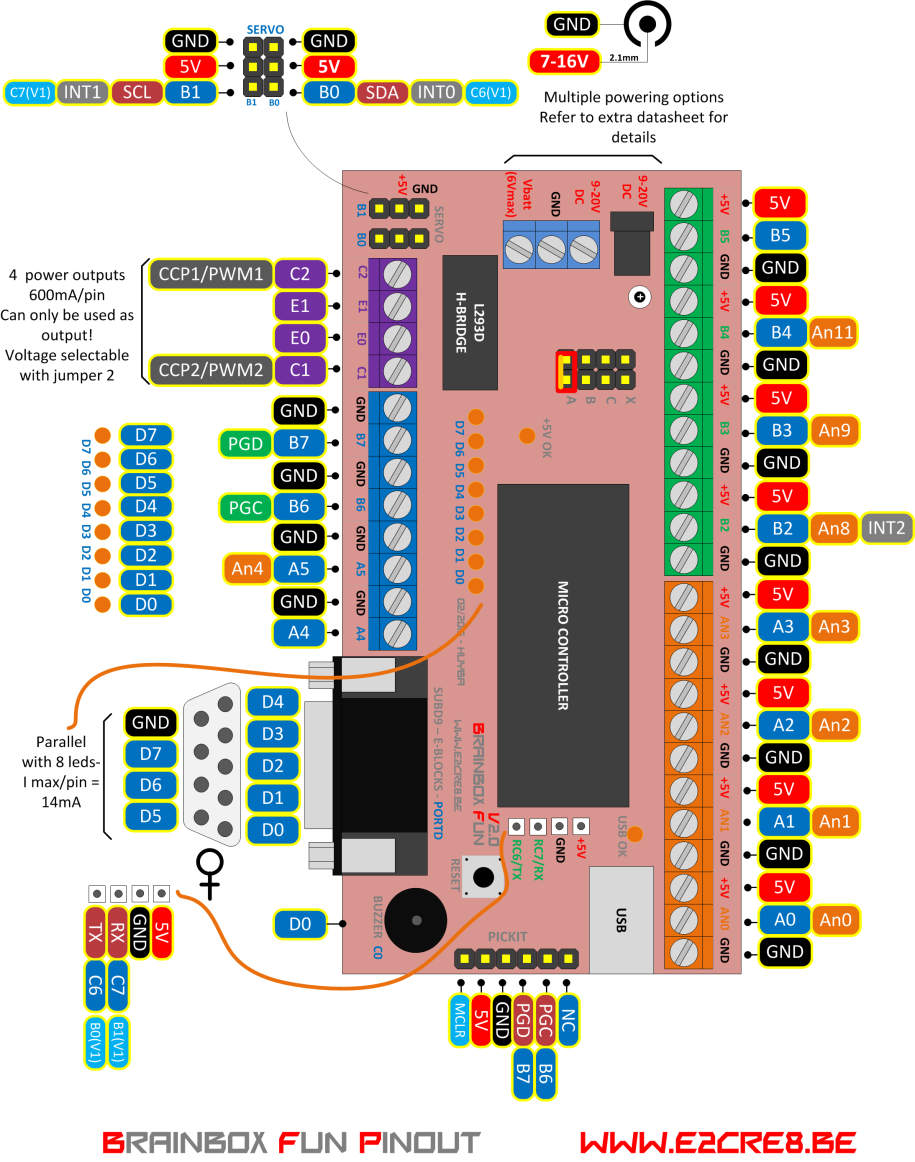
# IO pins microcontrollers

## PIC



TRIS registers bepalen input of output (1 = input, 0 is output)

PORT registers bepalen de toestand van een output pin, of geven de toestand weer van een input pin



# Programma 1 : 8 leds – PORTD – 4MSB en 4LSB flashen alternerend

|  |
| --- |
| TRISD – PORTD  Digital output  While(1)  void main(void)  // comment  #include  #define  !!Functie delay 100msec |

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

// The delay routine of the XC8 compiler is limited.

// \_\_delay\_ms(100); would be too large

// Solution: this function that repeats the 10msec delay 10 times

void delay\_10ms**(**unsigned char n**)**

**{**

**while** **(**n**--** **!=** 0**)**

**{**

\_\_delay\_ms**(**10**);**

**}**

**}**

void main**(**void**)** // main program

**{**

TRISD **=** 0x00**;** // make all pins of PORTD output pins

**while(**1**)** // never ending loop

**{**

PORTD **=** 0xF0**;** // make 4 MSB's of PORTD high

delay\_10ms**(**10**);** // wait for 100msec (10x10msec))

PORTD **=** 0x0F**;** // make 4 LSB's of PORTD high

delay\_10ms**(**10**);** // wait for 100msec (10x10msec))

**}**

**}**

# Programma 2: Binaire teller 0-255 op leds aan PORTD

|  |
| --- |
| for loop  variables  x=x+1 x++  #define leds PORTD |

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define leds PORTD // replace leds by PORTD

char x **=** 0**;** // declaration of variables

void main**(**void**)** // main program

**{**

TRISD **=** 0x00**;** // make all pins of PORTD output pins

**while(**1**)**

**{**

**for** **(**x **=** 0**;** x **<** 255 **;** x**=** x **+** 1**)** // x is 0 to start; as long as x is smaller than 255; increment x

**{**

leds **=** x**;** // visualise x at leds on PORTD

\_\_delay\_ms**(**10**);** // wait for 100msec (10x10msec))

**}**

x **=** 0**;** // set x back to 0

**}**

**}**

# Programma 3 : If then else demo

|  |
| --- |
| if elseif else  Output maskers |

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define leds PORTD // replace leds by PORTD

char x **=** 0**;** // declaration of variables

void delay\_10ms**(**unsigned char n**)**

**{**

**while** **(**n**--** **!=** 0**)** **{**

\_\_delay\_ms**(**10**);**

**}**

**}**

void main**(**void**)** // main program

**{**

TRISD **=** 0x00**;** // make all pins of PORTD output pins

**while(**1**)** // loop zolang x < of gelijk is aan 15

**{**

**while(**x **<=** 15**)** // loop zolang x < of gelijk is aan 15

**{**

leds**=** x**;** // visualise x at leds on PORTD

**if** **(**x **<** 5**)** // als x < 5

**{**

leds **=** leds **|** 0b00010000**;**

// maak RD4 hoog zonder de andere bits te veranderen

**}**

**if** **(**x **<** 10**)**

**{**

leds **=** leds **|** 0b00100000**;**

// maak RD5 hoog zonder de andere bits te veranderen

**}**

**else** **if** **(**x **<** 14**)**

**{**

leds **=** leds **^** 0b11110000**;**

// ^ exor – flip de 4 MSB’s

**}**

**else**

**{**

leds **=** leds **&** 0b00001111**;**

// maak de 4 MSB's laag, zonder iets aan de 4 LSB te veranderen

**}**

delay\_10ms**(**100**);** // wait for 500msec (10x10msec))

x **=** x **+** 1**;**

**}**

}

**}**

# Programma 4 : Case demo

|  |
| --- |
| Case |

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define leds PORTD // replace leds by PORTD

char x **=** 0**;** // declaration of variables

void delay\_10ms**(**unsigned char n**)**

**{**

**while** **(**n**--** **!=** 0**)** **{**

\_\_delay\_ms**(**10**);**

**}**

**}**

void main**(**void**)** // main program

**{**

TRISD **=** 0x00**;** // make all pins of PORTD output pins

**while(**x **<=** 5**)** // loop zolang x < of gelijk is aan 15

**{**

leds**=** x**;** // visualise x at leds on PORTD

**switch(**x**)**

**{**

**case** 1**:** // in het geval x gelijk is aan 1

leds **=** leds **|** 0b00010000**;**

// maak RD4 hoog zonder de andere bits te veranderen

**break;**

**case** 2**:** // in het geval x gelijk is aan 2

leds **=** leds **|** 0b00100000**;**

// maak RD5 hoog zonder de andere bits te veranderen

**break;**

**case** 3**:** // in het geval x gelijk is aan 3

leds **=** leds **|** 0b01000000**;**

// maak RD6 hoog zonder de andere bits te veranderen

**break;**

**default:** // in alle andere gevallen

leds **=** leds **&** 0b00001111**;**

// maak de 4 MSB's laag, zonder iets aan de 4 LSB te veranderen

**break;**

**}**

delay\_10ms**(**100**);** // wait for 500msec (10x10msec))

x **=** x **+** 1**;**

**}**

x **=** 0**;** // set x back to 0

**}**

# Programma 5 : Wanneer input pin B2 hoog is gaan de leds aan PORTD aan

|  |
| --- |
| Digital input  Pic18f4455.h bestand  TRISBbits.RB2  If else  ANSEL (Niet voor 18F) |

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define INPUTB2 PORTBbits.RB2 // replace INPUTB2 by PORTBbits.RB2 during compilation

#define LEDARRAY PORTD // replace LEDARRAY by PORTD during compilation

void main**(**void**)** // main program

**{**

TRISBbits**.**RB2 **=** 1**;** // make pin INPUT pin

TRISD **=** 0x00**;** // make all 8 bits of PORTD output – LEDS

// !!for 16FPIC – Be aware of ANSEL registers

**while(**1**)**

**{** // never ending loop

**if** **(**INPUTB2 **==** 1**)** // if input reads as 'high'

**{**

LEDARRAY **=** 0xFF**;** // All leds ON

**}**

**else** // if input reads as 'low'

**{**

LEDARRAY **=** 0x00**;** // All leds OFF

**}**

**}**

**}**

# Programma 6 : Input maskers

|  |
| --- |
| Input maskers  If (TRUE / FALSE)  Bitwise & - Bytewise && |

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define LEDARRAY PORTD // replace LEDARRAY by PORTD during compilation

void main**(**void**)** // main program

**{**

TRISB **=** TRISB **|** 0b00010100**;**

// B4 & B2 input pin without changing other pins

TRISD **=** 0x00**;** // make all 8 bits of PORTD output LEDS

// !!for 16FPIC Be aware of ANSEL registers

**while(**1**)**

**{** // never ending loop

**if** **((**PORTB **&** 0b00000100**)** **&&** **!(**PORTB **&** 0b00010000**))**

// if B4 is 1 and B2 is 0

**{**

LEDARRAY **=** 0x0F**;**

**}**

**else** **if** **(!(**PORTB **&** 0b00000100**)** **&&** **(**PORTB **&** 0b00010000**))**

// if B4 is 0 and B2 is 1

**{**

LEDARRAY **=** 0xF0**;**

**}**

**else** **if** **((**PORTB **&** 0b00000100**)** **&&** **(**PORTB **&** 0b00010000**))**

// if B4 is 0 and B2 is 1

**{**

LEDARRAY **=** 0xFF**;**

**}**

**else**

**{**

LEDARRAY **=** 0b10000001**;**

**}**

**}**

**}**

# Programma 7 : Functies

|  |
| --- |
| Functies types:   * void functie (void) – krijgt geen data mee en geeft geen data terug * void functie (var) – krijgt data mee, maar geeft geen data terug * var functie (var) – krijgt data mee en geeft data terug |

#include <xc.h> // general XC8 library

#define LEDARRAY PORTD // replace LEDARRAY by PORTD during compilation

char x**;** // global variable x (can be used all over the program and in every function)

// function prototypes - maak aan de compiler duidelijk dat deze 3 functies bestaan

// deze functies mogen nu na de main loop uitgescheven worden

void increment\_x**(**void**);**

void delay**(**unsigned short long w**);**

int macht**(**int x**,** int y**);**

void main**(**void**)** // main program

**{**

TRISB **=** TRISB **|** 0b00000100**;** // B2 input pin without changing other pins

TRISD **=** 0x00**;** // make all 8 bits of PORTD output LEDS

**while(**1**)**

**{** // never ending loop

**if** **(**PORTB **&** 0b00000100**)** // als pin RB2 hoog is

**{**

x **=** 2**;** // x = 2

LEDARRAY **=** x**;** // visualiseer x op leds = 2

delay **(**480000**);** // functie wacht 480000

increment\_x**();** // functie x = x + 1 -> x is nu dus 3

LEDARRAY **=** x**;** // visualiseer x op leds = 3

delay **(**480000**);** // functie wacht 480000

x**=** macht**(**x**,**3**);** // x was 3 - tot 3e macht maakt 27 - x is

//nu dus 27

LEDARRAY **=** x**;** // visualiseer x op leds = 27 = 0b00011011

delay **(**480000**);** // functie wacht 480000

**}**

LEDARRAY **=** 0**;**

**}**

**}**

//\*\*\*\*\* Functies \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void increment\_x**(**void**)**

**{**

x **=** x **+** 1**;** // x = x + 1

**}**

void delay**(**unsigned short long w**)**

**{**

unsigned short long i **=** 0**;** // local variable i - can only be used in this function

**for** **(**i **=** 0**;** i**<**w**;** i**++){}** // for loop die w maal herhaald zal worden

// door de grote variabelen zal dit toch enige tijd duren

**}**

int macht**(**int x**,** int y**)**

**{**

int i**,** m**;** // declaration of 2 local variables

int a **=** x**;** // stel a gelijk aan grondtal x

**for** **(**i**=**1**;** i**<**y**;** i**++)** // herhaal zolang de exponent niet bereikt is

**{**

m**=(**a**\***x**);**

a**=**m**;**

**}**

**return** m**;** // geef de uitkomst terug aan het programma nadat de functie is volbracht

**}**

# Programma 8 : functies in c en h files

|  |
| --- |
| Functies en functie prototypes kunnen ook in afzonderlijke bestanden gezet worden zodat ze door meerdere programma’s kunnen worden gebruikt. |

#include <xc.h> // general XC8 library

#include "E2CRE8.h" // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define LEDARRAY PORTD // replace LEDARRAY by PORTD during compilation

char x**;** // global variable x (can be used all over the program and in every function)

void main**(**void**)** // main program

**{**

TRISB **=** TRISB **|** 0b00000100**;** // B2 input pin without changing other pins

TRISD **=** 0x00**;** // make all 8 bits of PORTD output LEDS

**while(**1**)**

**{** // never ending loop

**if** **(**PORTB **&** 0b00000100**)** // als pin RB2 hoog is

**{**

x **=** 2**;** // x = 2

LEDARRAY **=** x**;** // visualiseer x op leds = 2

delay **(**480000**);** // functie wacht 480000

increment\_x**();** // functie x = x + 1 -> x is nu dus 3

LEDARRAY **=** x**;** // visualiseer x op leds = 3

delay **(**480000**);** // functie wacht 480000

x**=** macht**(**x**,**3**);** // x was 3 - tot 3e macht maakt 27 - x is nu dus 27

LEDARRAY **=** x**;** // visualiseer x op leds = 27 = 0b00011011

delay **(**480000**);** // functie wacht 480000

**}**

LEDARRAY **=** 0**;**

**}**

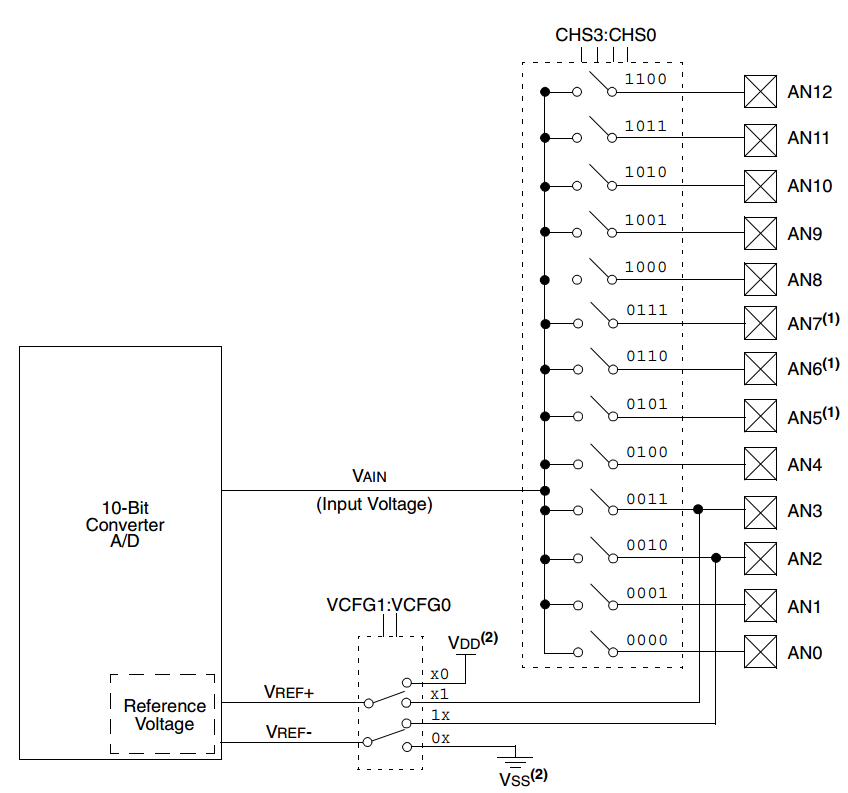
**}**

|  |
| --- |
| E2CRE8.h |
| #include <xc.h> // include processor files - each processor file is guarded.  void increment\_x**(**void**);**  void delay**(**unsigned short long w**);**  int macht**(**int x**,** int y**);** |

|  |
| --- |
| E2CRE8.c |
| #include <xc.h> // general XC8 library  #define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz  char x**;** // global variable x (can be used all over the program and in every function)  //\*\*\*\*\* Functies \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void increment\_x**(**void**)**  **{**  x **=** x **+** 1**;** // x = x + 1  **}**  void delay**(**unsigned short long w**)**  **{**  unsigned short long i **=** 0**;** // local variable i - can only be used in this function  **for** **(**i **=** 0**;** i**<**w**;** i**++){}** // for loop die w maal herhaald zal worden  // door de grote variabelen zal dit toch enige tijd duren  **}**  int macht**(**int x**,** int y**)**  **{**  int i**,** m**;** // declaration of 2 local variables  int a **=** x**;** // stel a gelijk aan grondtal x  **for** **(**i**=**1**;** i**<**y**;** i**++)** // herhaal zolang de exponent niet bereikt is  **{**  m**=(**a**\***x**);**  a**=**m**;**  **}**  **return** m**;** // geef de uitkomst terug aan het programma nadat de functie is volbracht  **}** |

# AD conversie

PIC 18F4455 (Brainbox Fun)



# Programma 9: AD conversie

|  |
| --- |
| AD convertor registers |

/\*This program 'reads' the analog voltage at pin AN0

\* The 8 bit result is displayed at the 8 leds of PORTD

\* Connect a potmeter to AN0 (between 5V and GND)\*/

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define LEDARRAY PORTD // using a define to give pin a better name

void main**(**void**)** // main program

**{**

//alle these settings are well explained in the datasheet of the 18F4455

TRISD **=** 0x00**;** // make all 8 bits of PORTD output - LEDS

TRISAbits**.**RA0 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA1 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA2 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA3 **=** 1**;** // make pin INPUT pin

ADCON2bits**.**ADFM **=** 0**;** // 0 = Left justified -> 8 bit adc

ADCON2bits**.**ACQT2 **=** 1**;**

ADCON2bits**.**ACQT1 **=** 1**;**

ADCON2bits**.**ACQT0 **=** 1**;** // 111 = Aquisition time of 20 TAD - slowest

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;** // FOSC/64 is necessary for 48Mhz clock

ADCON1bits**.**VCFG1 **=** 0**;** // REF- = VSS

ADCON1bits**.**VCFG0 **=** 0**;** // REF+ = VDD

ADCON1bits**.**PCFG3 **=** 1**;**

ADCON1bits**.**PCFG2 **=** 0**;**

ADCON1bits**.**PCFG1 **=** 1**;**

ADCON1bits**.**PCFG0 **=** 1**;** //1011 makes AN0, AN1, AN2, AN3 all AD pins

// all other AD pins are normal digital IO pins

ADCON0bits**.**CHS3 **=** 0**;**

ADCON0bits**.**CHS2 **=** 0**;**

ADCON0bits**.**CHS1 **=** 0**;**

ADCON0bits**.**CHS0 **=** 0**;** // 0000 = Select Channel AN0 for AD conversion

ADCON0bits**.**ADON **=** 1**;** // switch AD convertor on

**while(**1**)**

**{** // never ending loop

ADCON0bits**.**GODONE **=** 1**;** // Start AD conversion

**while** **(**GODONE**){}** // wait for AD conversion to complete

LEDARRAY **=** ADRESH**;** // show 8 bit AD result on LEDS of portD

**}**

**}**

# Programma 10: gebruik van arrays – gemiddelde van metingen

|  |
| --- |
| Arrays zijn een ideale manier om data in verschillende cellen het ram geheugen als groep aan te spreken.  Met dit programma demonstreren we een “moving average window” filter die steeds het gemiddelde van de laatste 10 metingen weergeeft. |

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define LEDARRAY PORTD // using a define to give pin a better name

#define samples 10

char avg**[**samples**]** **=** **{**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**,**0**};**

int total **=** 0**;**

int average **=** 0**;**

void delay\_10ms**(**unsigned char n**)**

**{**

**while** **(**n**--** **!=** 0**)**

**{**

\_\_delay\_ms**(**10**);**

**}**

**}**

void main**(**void**)** // main program

**{**

//alle these settings are well explained in the datasheet of the 18F4455

TRISD **=** 0x00**;** // make all 8 bits of PORTD output - LEDS

TRISAbits**.**RA0 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA1 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA2 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA3 **=** 1**;** // make pin INPUT pin

ADCON2bits**.**ADFM **=** 0**;** // 0 = Left justified -> 8 bit adc

ADCON2bits**.**ACQT2 **=** 1**;**

ADCON2bits**.**ACQT1 **=** 1**;**

ADCON2bits**.**ACQT0 **=** 1**;** // 111 = Aquisition time of 20 TAD - slowest

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;** // FOSC/64 is necessary for 48Mhz clock

ADCON1bits**.**VCFG1 **=** 0**;** // REF- = VSS

ADCON1bits**.**VCFG0 **=** 0**;** // REF+ = VDD

ADCON1bits**.**PCFG3 **=** 1**;**

ADCON1bits**.**PCFG2 **=** 0**;**

ADCON1bits**.**PCFG1 **=** 1**;**

ADCON1bits**.**PCFG0 **=** 1**;** //1011 makes AN0, AN1, AN2, AN3 all AD pins

// all other AD pins are normal digital IO pins

ADCON0bits**.**CHS3 **=** 0**;**

ADCON0bits**.**CHS2 **=** 0**;**

ADCON0bits**.**CHS1 **=** 0**;**

ADCON0bits**.**CHS0 **=** 0**;** // 0000 = Select Channel AN0 for AD conversion

ADCON0bits**.**ADON **=** 1**;** // switch AD convertor on

**while(**1**)**

**{** // never ending loop

ADCON0bits**.**GODONE **=** 1**;** // Start AD conversion

**while** **(**GODONE**){}** // wait for AD conversion to complete

**for** **(**char x **=** samples**;** x **>** 0**;** x**--)**

**{**

avg**[**x**-**1**]=** avg**[**x**-**2**];** // shift all registers 1 place up

**}**

avg**[**0**]** **=** ADRESH**;** // put last AD value in avg[0]

**for** **(**char x **=** samples**;** x **>** 0**;** x**--)**

**{**

total **=** total **+** avg**[**x**-**1**];** // add all register values

**}**

average **=** total **/** samples**;** // divide by number of values to average

LEDARRAY **=** average**;** // show average on 8 leds

total **=** 0**;** // reset total

delay\_10ms**(**50**);** // wait 500msec for next measurement

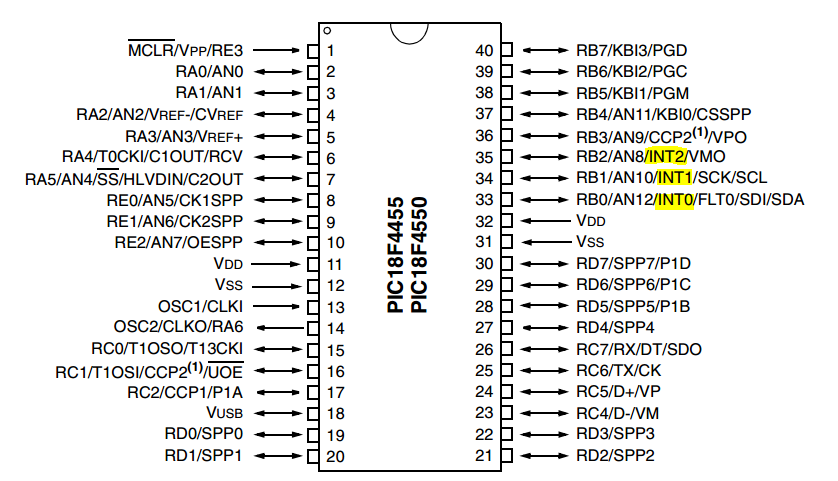
**}**

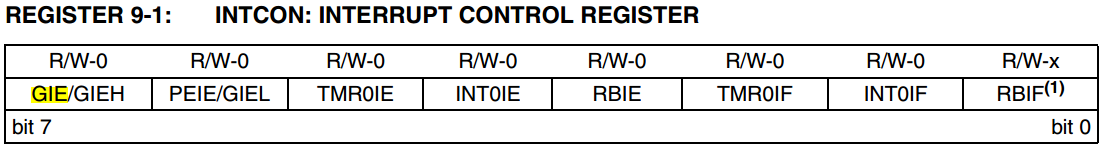
**}**

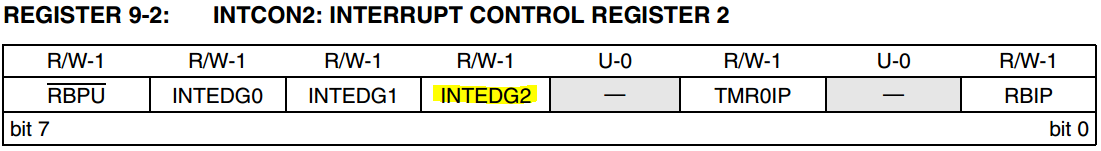
## Array avg[10]

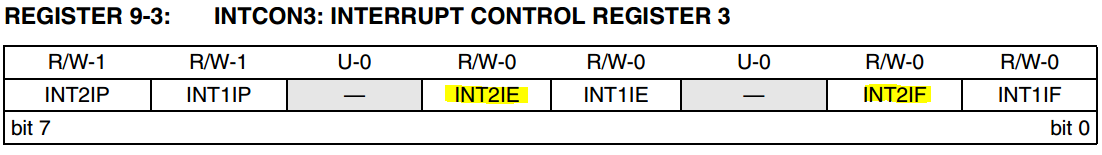


# Externe interrupt









#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

void main**(**void**)** // main program

**{**

//alle these settings are well explained in the datasheet of the 18F4455

TRISD **=** 0x00**;** // make all 8 bits of PORTD output - LEDS

PORTD **=** 0x0F**;** // set 4 LSB's

// Settings to generate an interrupt at INT2(RB2) on rising edge

TRISBbits**.**RB2 **=** 1**;** // set RB2 as input

INTCON2bits**.**INTEDG2 **=** 1**;** // 1 = Interrupt on rising edge

INTCON3bits**.**INT2IE **=** 1**;** // 1 = Enables the INT2 external interrupt

INTCONbits**.**GIE **=** 1**;** // 1 = Enables all unmasked interrupts

**while(**1**)**

**{** // never ending loop

// do nothing at all

**}**

**}**

void interrupt tc\_int **(**void**)** // interrupt routine

**{**

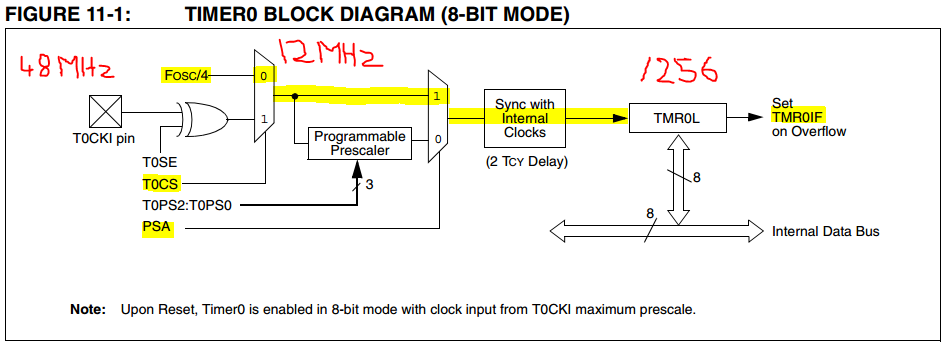
INTCON3bits**.**INT2IF **=** 0**;** // 1 = The INT2 external interrupt occurred

//(must be cleared in software) - by making it 0

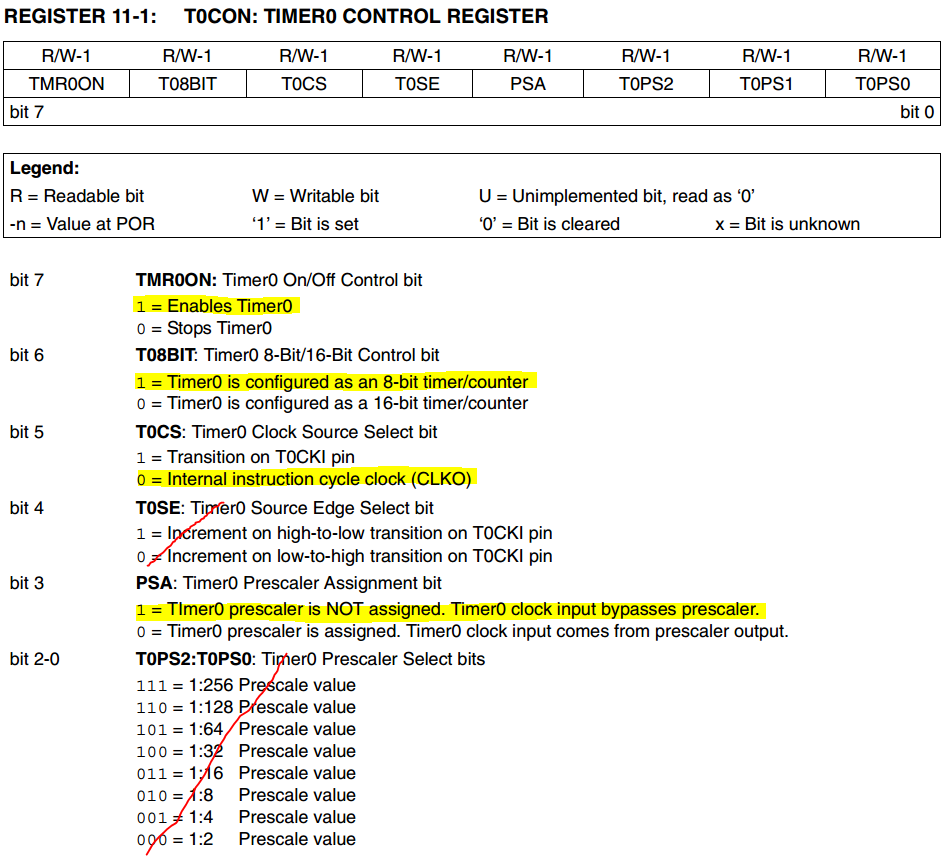
PORTD **=** PORTD**^**0b11111111**;** // flip all bits of PORTD (EXOR)

**}**

# TIMER0 interrupt – 1 seconde



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | / | is | / | is | / | is |
| 48000000 | 4 | 12000000 | 1 | 12000000 | 256 | 46875 |
| 48000000 | 4 | 12000000 | 2 | 6000000 | 256 | 23437,5 |
| 48000000 | 4 | 12000000 | 4 | 3000000 | 256 | 11718,75 |
| 48000000 | 4 | 12000000 | 8 | 1500000 | 256 | 5859,375 |
| 48000000 | 4 | 12000000 | 16 | 750000 | 256 | 2929,6875 |
| 48000000 | 4 | 12000000 | 32 | 375000 | 256 | 1464,84375 |
| 48000000 | 4 | 12000000 | 64 | 187500 | 256 | 732,421875 |
| 48000000 | 4 | 12000000 | 128 | 93750 | 256 | 366,2109375 |
| 48000000 | 4 | 12000000 | 256 | 46875 | 256 | 183,1054688 |



|  |
| --- |
| Dit programma gebruikt TIMER0 in 8 bit mode om exact elke seconde de toestand van de leds aan PORTD te veranderen. TMR0 staat zo ingesteld dat deze 46875 interrupt per seconde veroorzaakt. |

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

unsigned int secteller **=** 0**;** // 0-65535 - moet tot 46875 kunnen tellen

void main**(**void**)** // main program

**{**

//alle these settings are well explained in the datasheet of the 18F4455

TRISD **=** 0x00**;** // make all 8 bits of PORTD output - LEDS

PORTD **=** 0x0F**;** // set 4 LSB's

// Settings to generate an interrupt after 1sec using TMR0

T0CONbits**.**TMR0ON **=** 1**;** //1 = Enables Timer0

T0CONbits**.**T08BIT **=** 1**;** // 1 = Timer0 is configured as an 8-bit timer/counter

T0CONbits**.**T0CS **=** 0**;** // 0 = Internal instruction cycle clock (CLKO)

T0CONbits**.**PSA **=** 1**;** // 1 = Timer0 clock input bypasses prescaler

INTCONbits**.**T0IE **=** 1**;** // Enable interrupt on Timer0 overflow

INTCONbits**.**GIE **=** 1**;** // 1 = Enables all unmasked interrupts

**while(**1**)**

**{** // never ending loop

// do nothing at all

**}**

**}**

void interrupt tc\_int **(**void**)** // interrupt routine

**{**

INTCONbits**.**T0IF **=** 0**;** // int flag must be cleared in software

**if** **(**secteller **>=** 46875**)** // this is true - every second

**{** // after 46875 ints - 1 sec is passed

PORTD **=** PORTD**^**0b11111111**;** // flip all bits of PORTD (EXOR)

secteller **=** 0**;** // reset secteller

**}**

**else**

**{**

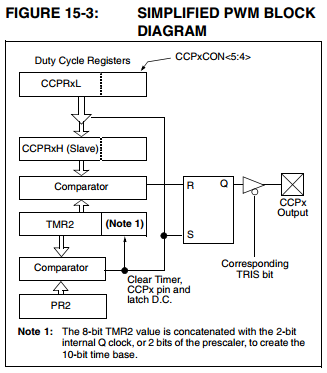
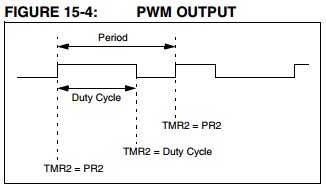
secteller **++;** // increment secteller

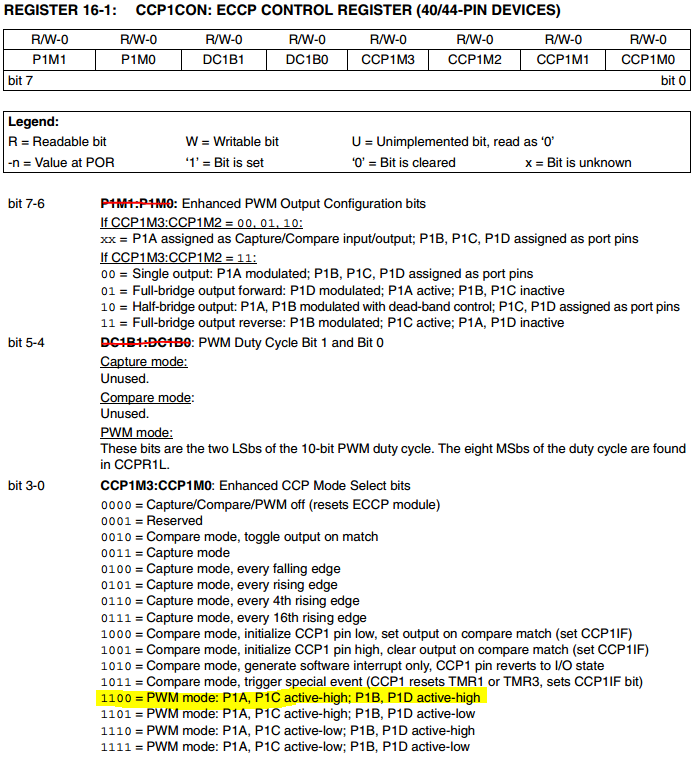
**}**

**}**

# PWM – PULSE WIDTH MODULATION

|  |
| --- |
|  |



/\*

\* File: O-PWM

\* Author: Bart Huyskens - www.e2cre8.be 04/2016

\* HARDWARE: BRAINBOX FUN - PIC18F445 + Bootloader running at 48Mhz

\*

\* PWM signals are modulated at pins C1 and C2

\* This program 'reads' the analog voltage at pin AN0

\* The 8 bit result is displayed at the 8 leds of PORTD

\* The same result is used to determine the Duty cycle at pin C2 and C1

\*

\*/

// PIC18F4455 Configuration Bit Settings -

// all config settings are made in the bootloader

#include <xc.h> // general XC8 library

#define \_XTAL\_FREQ 48000000 // Brainbox Fun running at 48Mhz

#define LEDARRAY PORTD // using a define to give pin a better name

void main**(**void**)** // main program

**{**

//alle these settings are well explained in the datasheet of the 18F4455

TRISD **=** 0x00**;** // make all 8 bits of PORTD output - LEDS

// PWM configuration

TRISCbits**.**RC2 **=** 0**;** // make pin output

TRISCbits**.**RC1 **=** 0**;** // make pin output

//config of CCP1 PWM pin (RC2)

CCP1CONbits**.**CCP1M3 **=** 1**;**

CCP1CONbits**.**CCP1M2 **=** 1**;**

CCP1CONbits**.**CCP1M1 **=** 0**;**

CCP1CONbits**.**CCP1M0 **=** 0**;** //11xx = PWM mode

//config of CCP2 PWM pin (RC1)

CCP2CONbits**.**CCP2M3 **=** 1**;**

CCP2CONbits**.**CCP2M2 **=** 1**;**

CCP2CONbits**.**CCP2M1 **=** 0**;**

CCP2CONbits**.**CCP2M0 **=** 0**;** //11xx = PWM mode

PR2 **=** 0xFF**;** // Full 10 bit resolution is possible

T2CONbits**.**T2CKPS1 **=** 1**;**

T2CONbits**.**T2CKPS0 **=** 0**;** // 10 = prescaler 16

T2CONbits**.**TMR2ON **=** 1**;** // activate TIMER 2 - TMR2 is used for PWM

// AD configuration

TRISAbits**.**RA0 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA1 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA2 **=** 1**;** // make pin INPUT pin

TRISAbits**.**RA3 **=** 1**;** // make pin INPUT pin

ADCON2bits**.**ADFM **=** 0**;** // 0 = Left justified -> 8 bit adc

ADCON2bits**.**ACQT2 **=** 1**;**

ADCON2bits**.**ACQT1 **=** 1**;**

ADCON2bits**.**ACQT0 **=** 1**;** // 111 = Aquisition time of 20 TAD - slowest

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;**

ADCON2bits**.**ADCS2 **=** 1**;** // FOSC/64 is necessary for our fast 48Mhz clock

ADCON1bits**.**VCFG1 **=** 0**;** // REF- = VSS

ADCON1bits**.**VCFG0 **=** 0**;** // REF+ = VDD

ADCON1bits**.**PCFG3 **=** 1**;**

ADCON1bits**.**PCFG2 **=** 0**;**

ADCON1bits**.**PCFG1 **=** 1**;**

ADCON1bits**.**PCFG0 **=** 1**;** //1011 makes pins AN0, AN1, AN2, AN3 all analog input pins

// all other AD pins are normal digital IO pins

ADCON0bits**.**CHS3 **=** 0**;**

ADCON0bits**.**CHS2 **=** 0**;**

ADCON0bits**.**CHS1 **=** 0**;**

ADCON0bits**.**CHS0 **=** 0**;** // 0000 = Select Channel AN0 for AD conversion

ADCON0bits**.**ADON **=** 1**;** // switch AD convertor on

**while(**1**)**

**{** // never ending loop

ADCON0bits**.**GODONE **=** 1**;** // Start AD conversion

**while** **(**GODONE**){}** // wait for AD conversion to complete

LEDARRAY **=** ADRESH**;** // show 8 bit AD result on LEDS of portD

CCPR1L **=** ADRESH**;** // AD result determines PWM duty cycle of RC2

CCPR2L **=** 255 **-** ADRESH**;** // Inverse PWM duty cycle is sent to RC1

**}**

**}**