

Besturingssystemen: WPO  
PIC18F2455

24 februari 2015

## Doel van de labo's

Aan de hand van 6 labolessen zullen we de basisbegrippen van een ingebed systeem verkennen. Hiernaast zullen we ook de basisconcepten van een ingebed operating system (OS) bekijken. Hieronder vallen preemptive scheduling, takenafhandeling, prioriteiten en interrupts. De labo's zullen gestructureerd opgebouwd worden met als einddoel een primitief besturingssysteem te laten functioneren op een ingebed systeem.

De labo's worden als volgt opgedeeld:

1. In het eerste labo zal men leren werken met de PIC18F2455. Zo zal er aangeleerd worden hoe men een datasheet doorgrondt en hoe de PIC initieel wordt ingesteld. Na dit eerste labo zullen de studenten hebben aangeleerd hoe een "hello-world" programma geschreven wordt. Aan het einde van dit labo zullen we in staat zijn om een hardware-programma te schrijven.
2. Het tweede labo legt de begrippen UART en interrupts uit. Hiermee wordt het mogelijk data te communiceren met een host-computer.
3. In het derde labo zal een schakeling ontworpen worden om een sensor via de ADC in te lezen. Via UART is het dan mogelijk deze waarde te verzenden.
4. Het vierde labo heeft als doel via de timer functies van de microcontroller een real-time klok (RTC) te maken. Hiermee dient dan in het vijfde labo een scheduler gemaakt te worden dat de basis vormt van een besturingssysteem.
5. Zoals reeds aangehaald zal in het vijfde labo een scheduler worden gemaakt. Deze scheduler kan individuele taken zoals het inlezen van een sensor, versturen van informatie apart en periodiek plannen.
6. In het laatste labo wordt dit samengebundeld met een wireless systeem dat werkt via UART. Zo kan de waarde van de ADC draadloos en periodiek verstuurd worden. Het ingebed draadloos ontwikkeld systeem verstuurt zijn informatie volgens een bepaald vooropgesteld protocol. Door de aangeboden conventie te hanteren kan er gestructureerd gecommuniceerd worden met de Beaglebone van de vorige labo's.

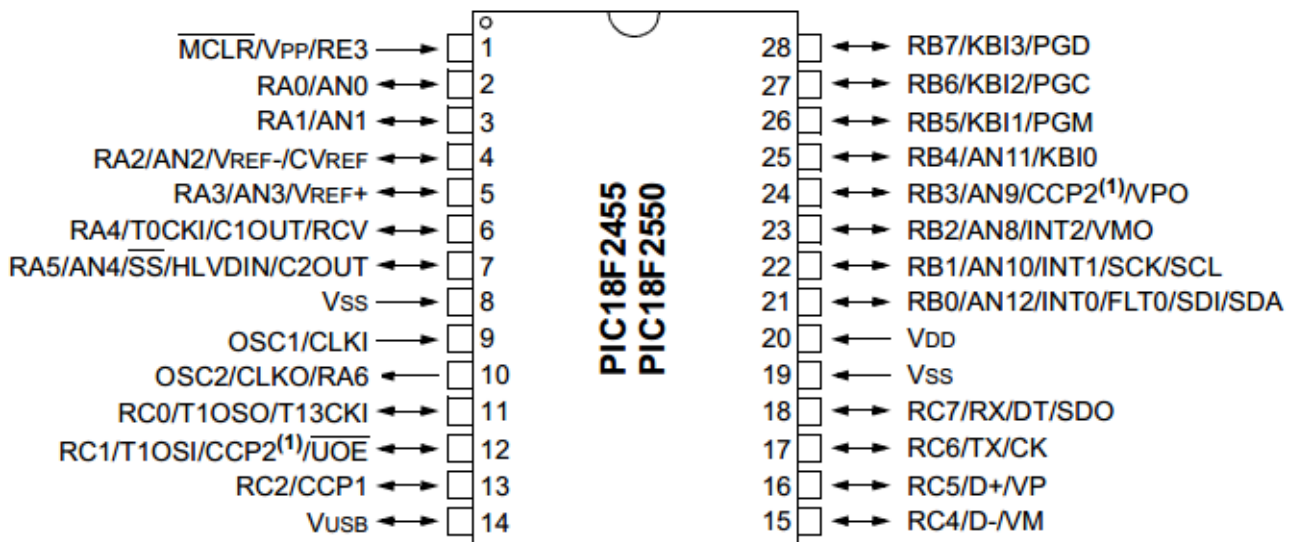
## PIC18F2455

Tijdens de labo's wordt een PIC18F2455 demo-bordje voorzien. Het bordje is uitbreidbaar voor diverse toepassingen. Dit wordt aan de aandachtige lezer overgelaten. Hier dienen de studenten het bord van te ontwikkelen voor de verdere labo's. De PIC18F2455 is een 8-bit controller met 24 KB programmeer geheugen. De communicatie met randapparatuur kan gebeuren via UART, A/E/USART, SPI, I2C of USB. Daarnaast heeft hij eveneens 16-bit timers en 10x10-bit ADC's. Deze dienen aangesproken te worden via registers.

De effectieve werkwijze en opbouw wordt gevonden in de datasheet:

<http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>

De IC die we hanteren is de 28-pin PDIP versie. Het pin diagram wordt weergegeven in figuur 1.



Figuur 1: Pin diagram van PIC18F2455

Aan de hand van de datasheet wordt het mogelijk de microcontroller te configureren en te gebruiken. Deze datasheet is cruciaal als handleiding voor de microcontroller.

## MPLAB X IDE

**MPLAB X IDE** De PIC microcontrollers van Microchip worden geprogrammeerd aan de hand van MPLAB X. Dit is een ontwikkelomgeving met als doel het programmeren van 8, 16 en 32 bit PIC microcontrollers. MPLAB X IDE v2.30 is te verkrijgen via volgende URL:

<http://www.microchip.com/pagehandler/en-us/family/mplabx/>

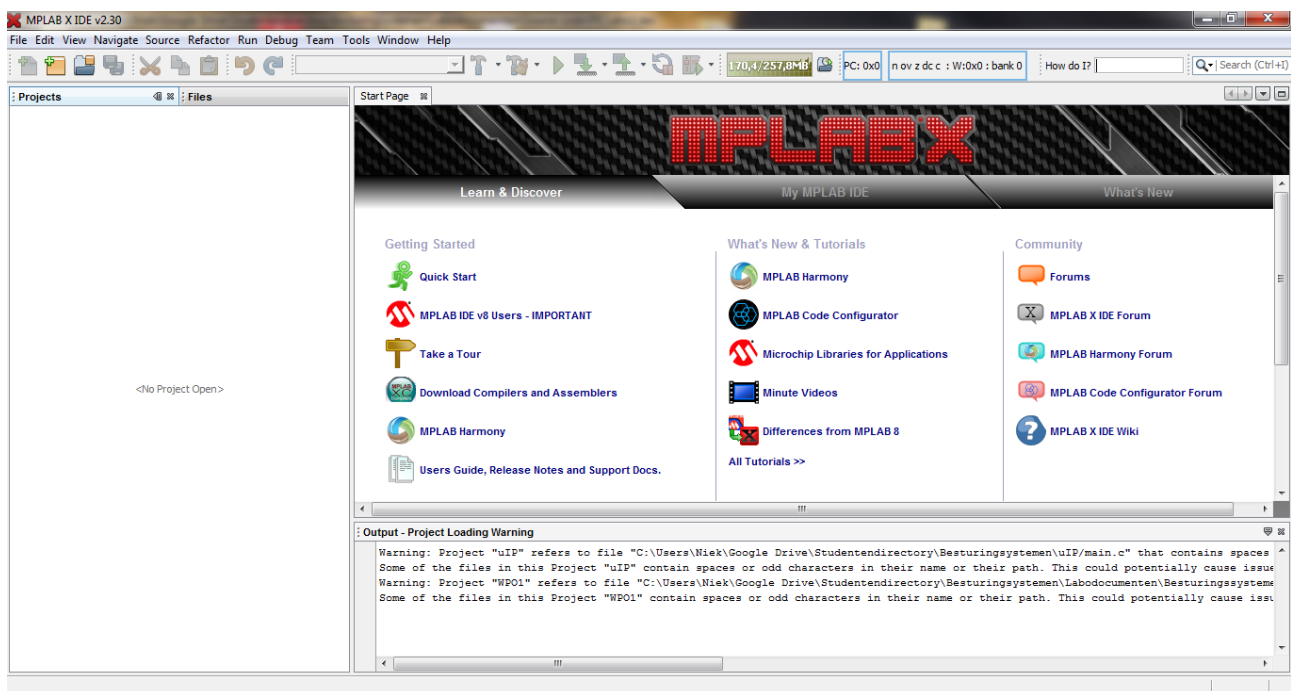
**XC-8 Compiler** Daarnaast dient eveneens een compiler voor de PIC specifieke c-code te worden genstalleerd. Voor 8-bit microcontrollers zijn er twee beschikbare varianten: C18-compiler en XC8-compiler. Deze laatste is de nieuwste variant en zal worden gebruikt in de labo-lessen. De gratis verkrijgbaar licentie is voldoende voor de doeleinden van het labo. Een activatie-code dient niet ingevuld te worden. De XC-8 compiler is te verkrijgen via de volgende URL:

[http://www.microchip.com/pagehandler/en\\_us/devtools/mplabxc/](http://www.microchip.com/pagehandler/en_us/devtools/mplabxc/)

**Putty Seriële monitor** Daarnaast zal nog een seriële monitor nodig zijn om het verzenden en ontvangen van seriële informatie mogelijk te maken. Een voorbeeld van zo'n monitor is Putty. Via een COM-poort op de computer kan seriële informatie verzonden en ontvangen worden. Putty is te verkrijgen op de volgende URL:  
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

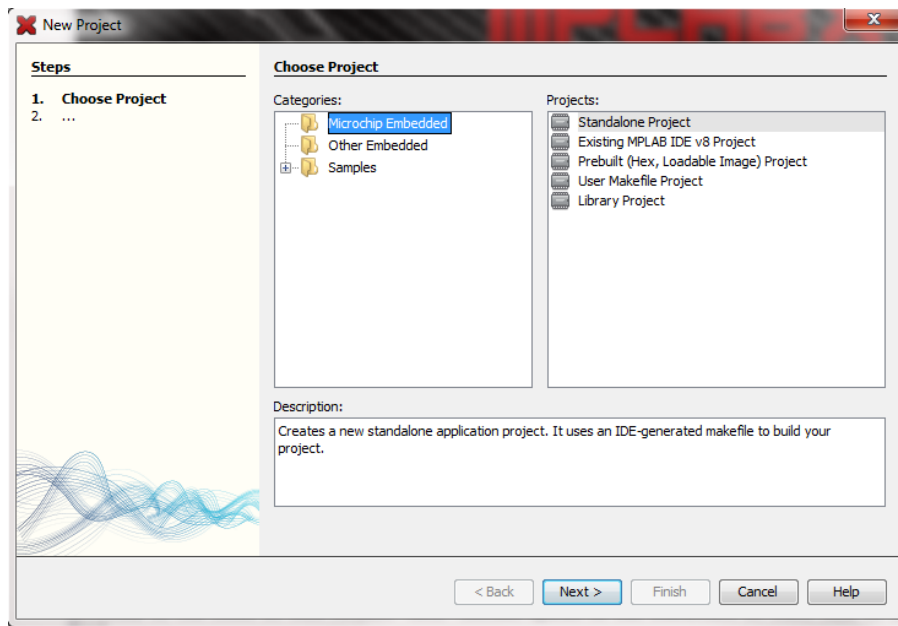
## Aanmaak eerste project

Nadat MPLAB X IDE en de XC-8 compiler zijn geïnstalleerd kan men overgaan tot de aanmaak van het eerste project. Het hoofdvenster dat men dient te zien na het openen van MPLAB X IDE wordt weergegeven in figuur 2. De aanmaak van een project gebeurt door in MPLAB X IDE hoofdvenster links boven **File** → **New project** te selecteren.



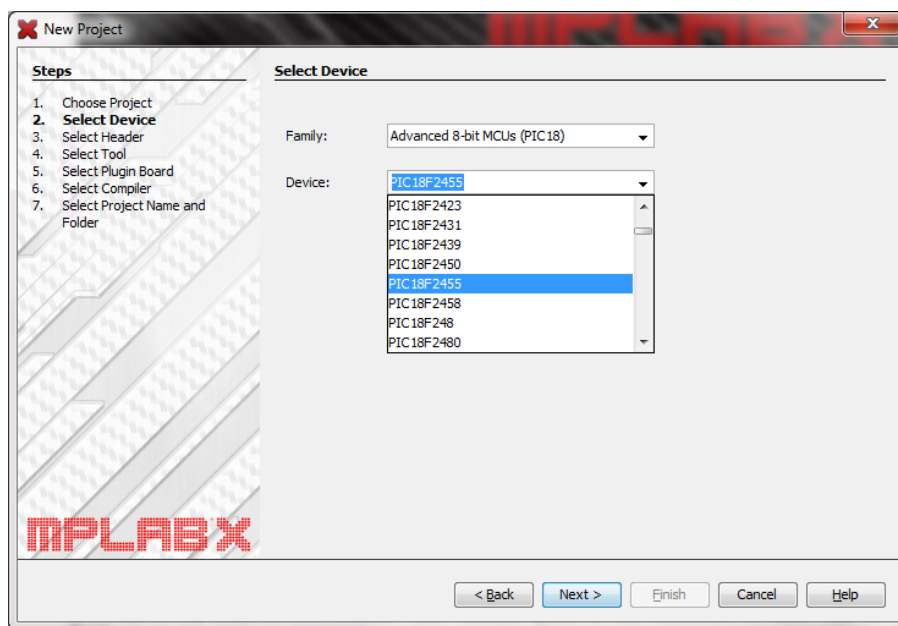
Figuur 2: Hoofdvenster van MPLAB X IDE

Een vragenbox opent zoals weergegeven in figuur 3. Hier selecteert men “**Microchip embedded** en “**Standalone project**. Telkens wanneer men op de drukknop “**Next** duwt, wordt een nieuwe vragenbox geopend.



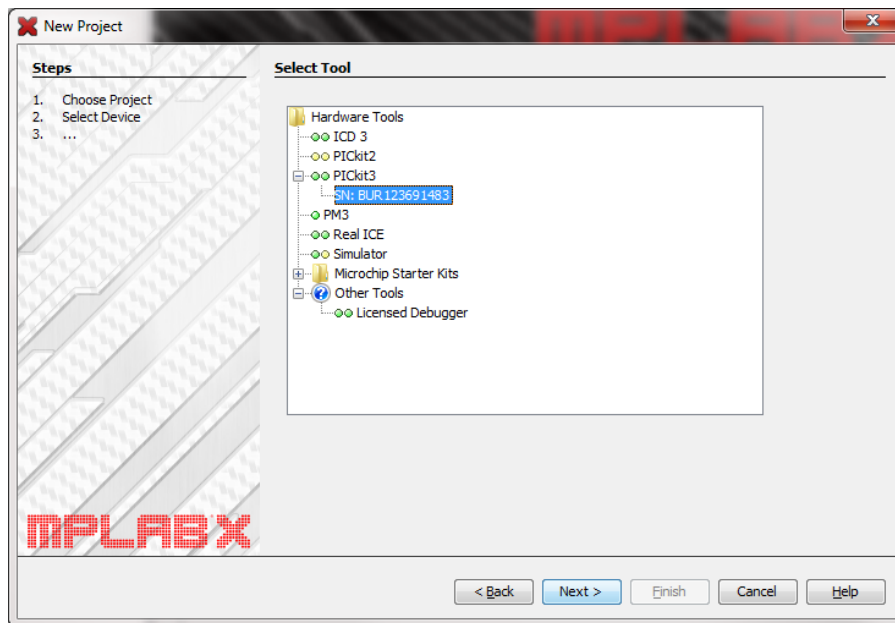
Figuur 3: Vragenbox aanmaak nieuw project

Een nieuwe vragenbox opent zich. In deze vragenbox selecteert men **PIC18F2455** onder **Advanced 8-bit MCUs (PIC18)**. Dit wordt weergegeven in figuur 4. Indien in latere projecten een andere PIC wordt gehanteerd zal een andere keuze uit deze lijst moeten gemaakt worden.



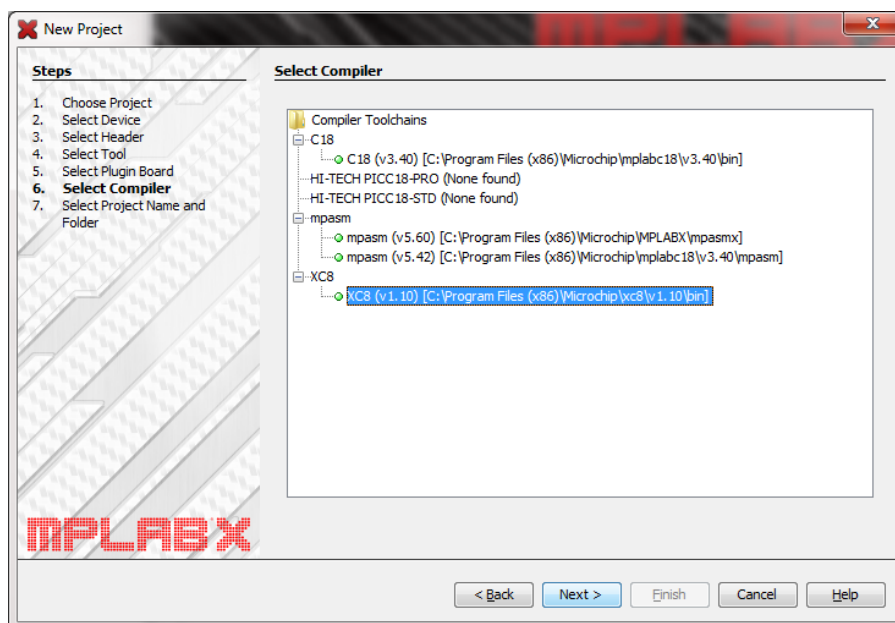
Figuur 4: Vragenbox keuze microcontroller

Na de keuze van de pic, dient de programmer gekozen te worden. De PICKit dient nu aangesloten te worden, indien men nog niet over een programmer zoals de PICKit beschikt, dan wordt gewoon op **next** geduwd. In de vragenbox zoals afgebeeld in figuur 5 selecteert men dan uiteindelijk de aangesloten **PICKit3**. Indien er een PICKit 2 wordt gebruikt kiest men deze. De PICKit is gedetecteerd wanneer zijn serial number (S/N) wordt weergegeven.



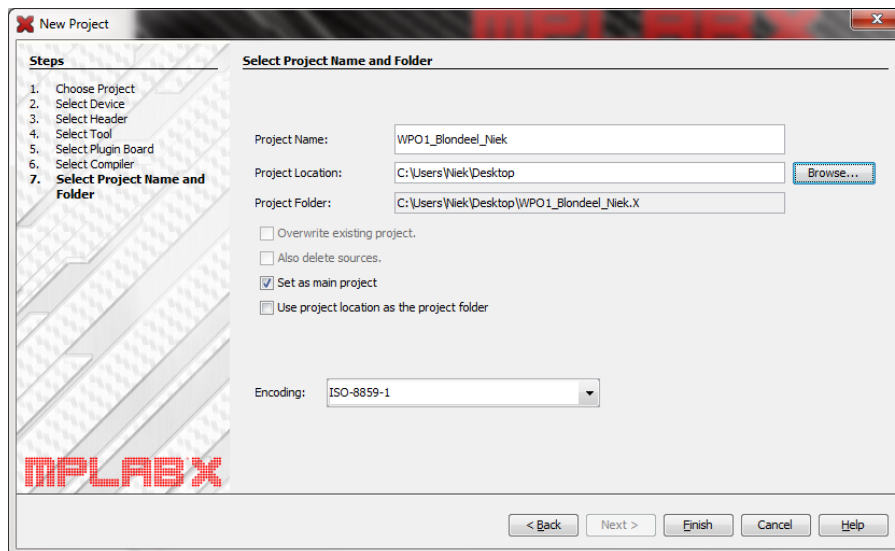
Figuur 5: Vragenbox keuze programmer

In de voorlaatste vragenbox dient de compiler gekozen te worden. Zoals reeds vermeld zullen wij de **XC8 compiler** hanteren. In de vragenbox zoals afgebeeld in 6 selecteert men dan ook de XC8 compiler.



Figuur 6: Vragenbox keuze compiler

Tot slot is er de laatste vragen box die de bestandslocatie van het project bepaald, dit is afgebeeld in 7. Kies een zinnige "**Project Name:** (zoals "WPO1\_<Naam\_van\_student>"). Kies een "**Project Location:** en klik "**Finish.**"

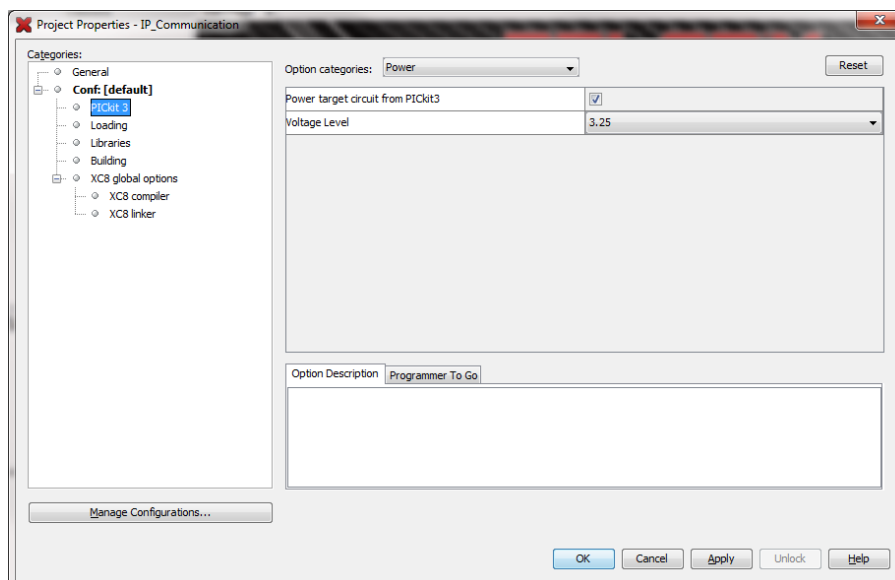


Figuur 7: Vragenbox keuze project opslaan

## Ontwikkelford voeden door PICKit3 of power supply unit (PSU)

Indien nodig kan het bord eventueel gevoed worden door de PICKit3 via de USB poort. De PIC18F2455 ondersteunt spanningen van 2.0 V tot 5.5 V. Aangezien de wireless RF chip van de laatste labo's op 3.3 V wordt gevoed, kiezen we als voedingsspanning 3.3 V.

**Voeden door PICKit3** Om het ontwikkelbord te voeden via de PICKit3 moeten enkele instellingen worden aangepast. In de **Project Properties** (Links bovenaan: Projects tab → rechterklik op Properties) bij “Categories links boven wordt gekozen voor PICKit 3. Er verschijnt een optie tabel. In het dropdown menu bij “**Option categories:** kiest men “**Power**. Figuur 9 toont het uiteindelijk menu. In dit menu dient “**Power target circuit from PICKit 3** aangevinkt te worden met een “**Voltage Level** van 3.25 V. Voor de PICKit 2 wordt een gelijkaardige werkwijze gehanteerd.

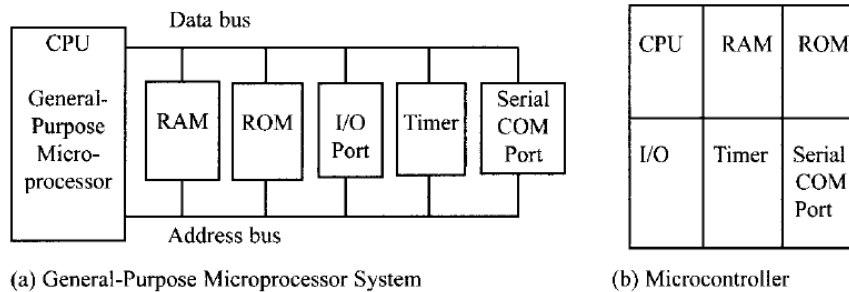


Figuur 8: Vragenbox keuze PICKit power 3.3 V

**Voeden door externe spanningsbron (PSU)** Daarnaast kan op de voorziene pinnen op het ontwikkelbord eventueel 3.3 V extern worden aangebracht via een spanningsbron (PSU).

# 1 Besturingssystemen WPO1: “Hello world” op de PIC18F2455

Het eerste programma dat men zal schrijven op de PIC microcontroller is “hello world”. De eerste vraag die men kan stellen is: “Wat is een microcontroller en waarvoor gebruikt men het?”. Hier wordt het onderscheid gemaakt tussen microprocessors en microcontrollers. Dit onderscheid wordt weergegeven in figuur 6. Deze figuur toont de microcontroller die een compact systeem vormt waar reeds elke component in de IC terug te vinden is. De microprocessor zoals een de x86/x64 series van Intel of AMD zijn complexere gehelen waar individuele systemen worden gekoppeld.



Figuur 9: Microprocessors(a) t.o.v. microcontrollers (b)

De microcontroller is door zijn compactheid perfect voor een ingebed systeem. Ingebedde systemen hebben als doel een specifiek programma keer op keer te herhalen. Een voorbeeld is een printer, die herhaaldelijk zijn data zal ophalen en afprinten. Een voorbeeld van een microprocessor is een Core i5 processor of ARM cortex A8 (Beaglebone) die verschillende programma's zal opstarten, afhankelijk van de input van de gebruiker. Hiervoor is RAM voorzien in deze architectuur, de gebruiker laadt zijn programma in het RAM in en kan zo verschillende applicaties draaien. Een besturingssysteem handelt de overgang tussen deze applicaties af.

Een ingebed systeem met microcontroller heeft vaak enkel ROM, en zal dus enkel een vooraf bepaald programma uitvoeren. Op een microcontroller kan ook gewisseld worden tussen bepaalde taken binnen een programma, hiervoor dient een ingebed besturingssysteem. Afhankelijk van bepaalde gebeurtenissen zoals de tijd (trigger) of een datalijn dat hoog wordt getrokken, kunnen bepaalde taken worden uitgevoerd. Een scheduler zal taken starten bij deze bepaalde gebeurtenissen. Als we kijken naar een besturingssysteem zoals een real-time operating system (RTOS), dan zien we dat de scheduler aan stricte criteria moet voldoen. Het doel bij een RTOS is een zo kleine mogelijke delay op wisselen van taken in te voeren. Zo zal het afhandelen van een taak zeer responsief zijn, het systeem reageert onmiddellijk op externe en interne gebeurtenissen, het voelt real-time aan. Kijken we naar een praktisch voorbeeld zoals een weefgetouwen machine, dan zal voor een perfect gewoven tapijt bijvoorbeeld een besturingssysteem real-time, verschillende motoren moeten aansturen zodat deze gelijktijdig parallel een bepaald motief weven. Enige delay in het stuurmechanisme zorgt voor een andere, verkeerd motief. Het doel van deze lessen zal dan ook zijn enkele basisprincipes aan te leren zodat men reeds over een basis beschikt waarmee men verdere complexere (RT)OS systemen kan doorgronden.

## 1.1 Instructie cyclussen en klok frequentie van de PIC18F2455

Nu we weten waarvoor de microcontroller gebruikt wordt zal het interessant zijn te kennen hoe snel deze microcontroller werkt. De PIC microcontroller is gepipelined waardoor één instructie cyclus welgeteld vier oscillator periodes duurt. De meeste instructies in de PIC zijn 8-bit of 16-bit en duren één instructie cycle. Hoe lang zal de periode van één instructie cycle zijn als men de volgende kristallen gebruikt:

1. (a) 4 MHz
2. (b) 16 MHz
3. (c) 20 MHz

Het antwoord op deze vraag moet later in commentaar bijgevoegd worden in de source code.

## 1.2 Configuratie van de PIC18F2455

Elke PIC microcontroller dient initieel geconfigureerd te worden. Hierdoor worden basisparameters zoals de frequentie van het klokkristal, brown out timers, watchdog timers en eventueel een masterclear ingesteld. Dit zijn de voornaamste parameters, voor een volledige lijst kan men het "settings addendum" van PIC18 raadplegen op volgende url:

<http://ww1.microchip.com/downloads/en/DeviceDoc/51537a.pdf>

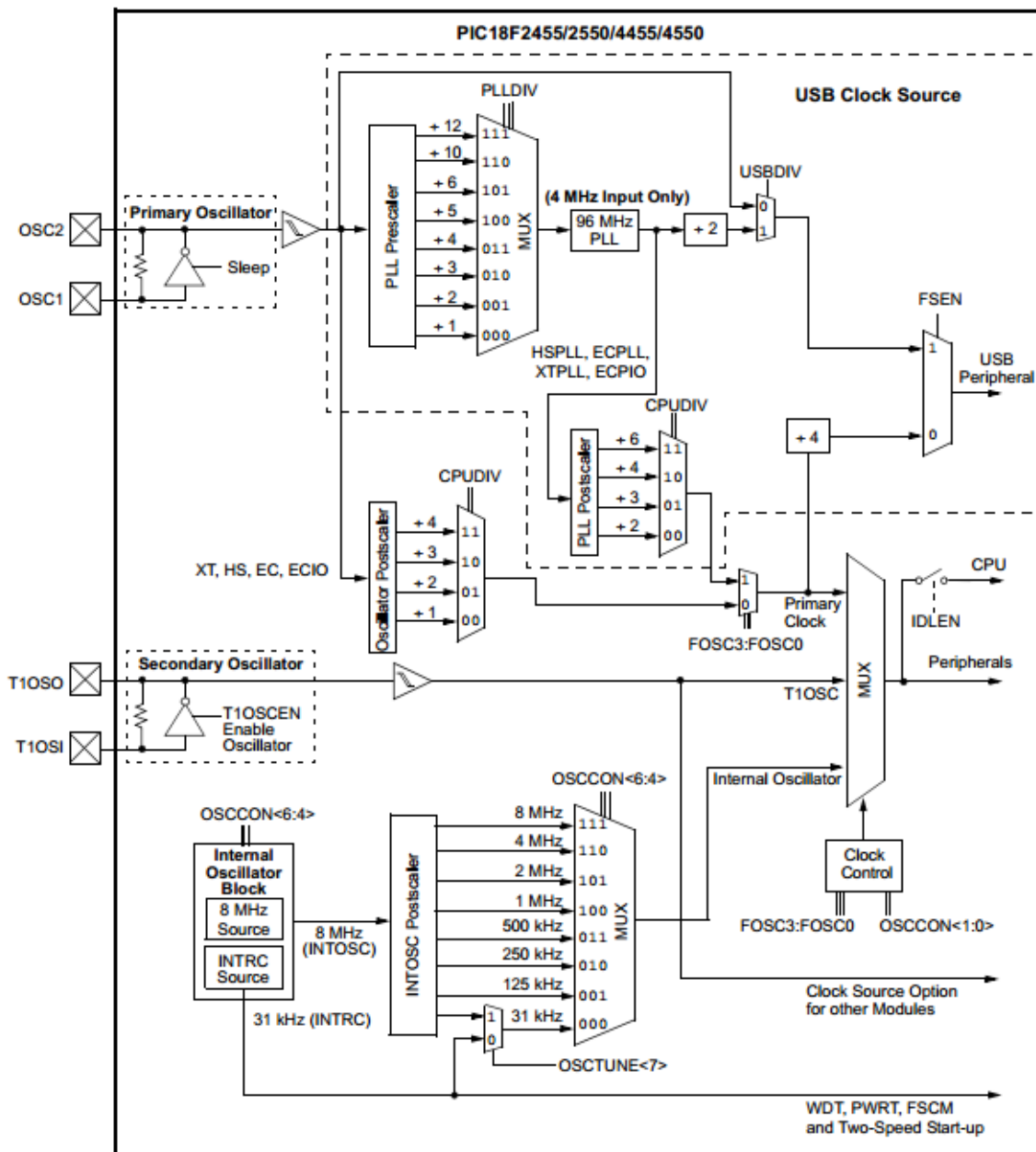
Dit document, tezamen met de datasheet vormen de basis van het PIC ingebed programmeren. Hieronder is een voorbeeld gegeven voor de PIC18F4550, de grotere broer van de PIC18F2455.

```
#include <p18f4550.h>

#pragma config FOSC = INTOSCIO_EC //Internal oscillator, port function on RA6, EC used by USB
#pragma config WDT = OFF //Disable watchdog timer
```

Figuur 10: Configuratie code voor PIC18F2455

de configuratie code voor PIC18F2455 kan hierop gebaseerd worden. Uit het vorige document, het addendum, dient de correcte informatie voor de PIC18F2455 gehaald te worden. Er wordt verwacht dat de config code in commentaar gedocumenteerd is, waardoor het duidelijk is waarom de config effectief zo is ingesteld. De oscillator configuratie kan worden gevonden in hoofdstuk 2 van de datasheet van de PIC18F2455. Het schema dat men in de datasheet vindt is hieronder afgebeeld in figuur 11.



Figuur 11: Oscillator schema van PIC18F2455



Aangezien we enkel een primair kristal gebruiken zal het bovenste deel uit dit schema het belangrijkste zijn. We zien dat naast de initiële config, eventueel andere registers zoals FOSC/OSCCON zullen moeten gezet worden. Dit wordt de eerste taak in de main functie, hier komen we later op terug. De XC8 compiler laat twee schrijfwijzes toe voor het definiëren van registers:

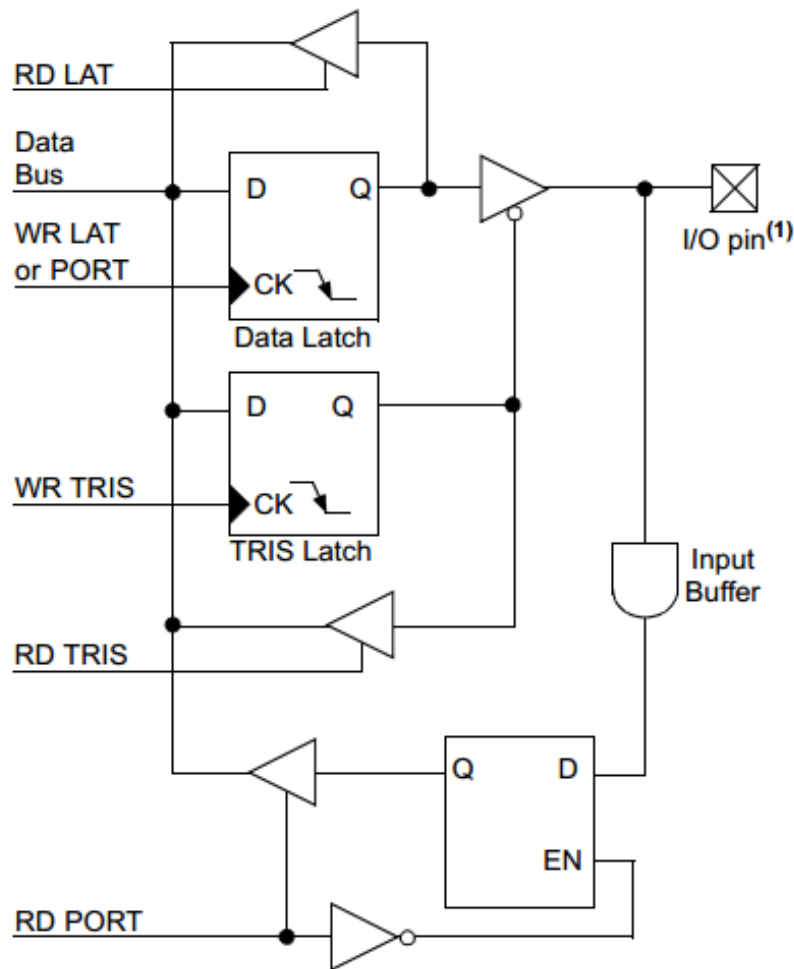
Toewijzing van byte: `OSCCON = 0x00 (tot) 0xFF`

Toewijzing van bit: `OSCCONbits.SCS0 = 0 (of) 1`

De naam van het individuele register vindt men eveneens in de datasheet. Voor de beginnende student kan het handig zijn om de registers bitsgewijs te definiëren (via de gedeclareerde unions). Zo is men zeker welke waarde bij welke register overeenkomt.

### 1.3 I/O pinnen

Het gebruik van I/O pinnen is gelijkaardig aan Arduino code. Zo dient er gedefinieerd te worden welke pin er wordt aangesproken. Daarnaast dient bij de PIC18 reeks ook gedefinieerd te worden of deze een ingang of uitgang is. Dit gebeurt via de LAT, PORT en TRIS registers.



**Note 1:** I/O pins have diode protection to VDD and VSS.

Figuur 12: Intern schema van IO pin in de PIC18F2455(zet een '0' op een IO pin als uitgang)

In figuur 12 wordt de interne structuur van de PIC18 I/O-pin weergegeven. De data bus zal de effectieve waarde zijn die wordt geschreven of gelezen van de I/O pin. Het TRIS register configureert de poort als Input of output ('1' of '0'). Met het LAT register wordt een waarde geschreven (output), met het PORT register wordt een waarde gelezen (input). Voor een diepgaandere uitleg wordt er verwezen naar hoofdstuk 10 in de datasheet.

## 1.4 Opdracht WPO4: “Hello world”

De student zal nu de opgedane kennis uit vorige paragrafen omzetten in een praktisch voorbeeld. Er wordt een “hello world” programma geschreven dat met een periode van 250ms en een dutycycle van 50% een led op pin C0 van het ontwikkelbord laat blinken, dit wordt via een oscilloscoop bevestigd. Als hint wordt de delay functie: “\_delay(C)” meegegeven (# include <xc.h>). Deze zorgt voor een delay van C cycles (let op: limiet voor C is max 198000). Met de reeds uitgerekende waarde kan men een bepaalde tijdsdelay bekomen. Dit programma wordt aan het einde van de het labo ingediend. De lezer moet uit de commentaar van de code verstaan waarom de registers zo zijn gezet.

## Referenties

1. Datasheet: PIC18F2455, <http://ww1.microchip.com/downloads/en/DeviceDoc/39632D.pdf>
2. PIC18 Configuration settings addendum, <http://ww1.microchip.com/downloads/en/DeviceDoc/51537a.pdf>
3. Mazidi, Muhammad Ali, Rolin D. McKinlay, and Danny Causey. PIC microcontroller and embedded systems: using Assembly and C for PIC18. Pearson Prentice Hall, 2008.