

# Labo datacommunication les 1: digitale in- en uitgangen

## Inleiding

De module datacommunicatie bestaat uit 4 cu labo en 2 cu theorie.

De eerste 5 weken gaan we de USB data-acquisitiekaart verkennen. We leren werken met de digitale in- en uitgangen, analoge in- en uitgangen. Hierbij moeten we gebruik maken van bitoperaties. Daarna zullen we leren werken met een protocol, o.a. I<sup>2</sup>C en het aansturen van het LCD display.

Die taken worden wekelijks op het einde van het labo ingediend. Er worden een drietal van die taken gequoteerd. Die punten worden niet meegedeeld. Wie feedback wil over een project moet die tijdens het labo aan de docent vragen.

De tweede helft van het semester verkennen we o.a. de volgende technologieën:

Arduino

Raspberry Pi

RFID

Smartcard

WII controller

Adam module

Color dome camera,...

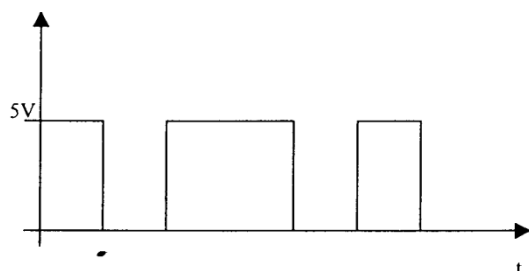
Er wordt per team (2 studenten) één project uitgekozen. Eén project bestaat uit verschillende technologieën.

De totale quotatie van het labo bestaat uit de permanente evaluatie van de taken, de evaluatie van het project en een labo examen. Wie een afwezig heeft voor zijn permanente evaluatie of voor de evaluatie van zijn project krijgt een afwezig voor gans de module.

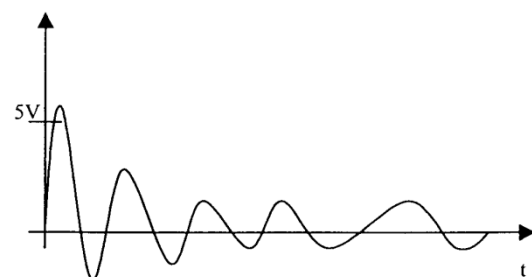
## Digitale signalen

Digitale elektronica maakt gebruik van twee spanningen om een 0 of een 1 voor te stellen. Als bijvoorbeeld de spanning 5V bedraagt, wordt dit aanzien als een digitale 1, als de spanning 0V bedraagt, wordt dit aanzien als een digitale 0.

Figuur 1 en 2 tonen respectievelijk een digitaal en een analoog signaal.



figuur 1: digitaal signaal



figuur 2 : analoog signaal

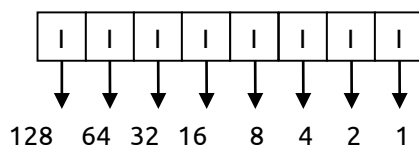
## USB poort

Een digitale 0 = 0V

Een digitale 1 = 5V.

## Bitoperaties

Een byte bestaat uit acht bits. Een byte kan 256 verschillende waarden aannemen van 0 tot 255



Soms moeten we 1 enkele bit van een byte inlezen of sturen zonder de andere 7 bits te beïnvloeden.

1 bit op 0 plaatsen

```

      1100 1010
AND  1111 1101
      1100 1000
    
```

1 bit inlezen

```

      1100 1010
AND  0000 0010
      0000 0010
    
```

1 bit op 1 plaatsen

```

      1100 1000
OR   0000 0010
      1100 1010
    
```

2 bits inlezen

```

      1100 1010
AND  0000 1010
      0000 1010
    
```

## Enkele voorbeelden

### Hoe zet je een bit hoog, zonder de rest te beïnvloeden?

Lees de byte in, en steek die in een variabele x  
 Maak een variabele y, met als waarde, de bit die we hoog willen maken.  
 Voer een or operatie uit met x en y, het resultaat is z

Voorbeeld:

x=8, binair is dit 0000 1000

We willen de meest linkse bit hoog zetten: y = 1000 0000 = 128

x or y:

	0000 1000 (ingelesen waarde)	
or	1000 0000	
	-----	
	1000 1000	= 136

### Hoe zet je een bit laag, zonder de rest te beïnvloeden?

Lees de byte in, steek die in een variabele x  
 Maak een variabele y, met als waarde, de bit die we laag willen maken.  
 Voer een xor operatie uit op y.  
 Voer een and operatie uit met x

Voorbeeld:

We lezen een byte in, de waarde is 254 (1111 1110)

Nu willen we de bit met waarde 8 (0000 1000) laag zetten.

We berekenen het inverse van 8 door een XOR operatie uit te voeren met 255:

8 XOR 255	00001000	
xor	11111111	
	-----	
Y=	11110111	= (255-8=247)

Nu voeren we een and operatie uit met het resultaat van de xor operatie:

```
254 and 247      11111110
                  and  11110111
                  -----
                  11110110
```

Samengevat:  $z = x \text{ and } (y \text{ xor } 255)$  waarbij y de bits zijn die we laag willen zetten.

Hoe weten we of een bit hoog staat?

Gegeven: ingelezen waarde x

Vraag: Is bit met waarde 64 hoog?

Oplossing:

byte met bit die 64 voorstelt hoog zetten = 0100 0000

$y = x \text{ and } 64$

Als y verschilt van 0 dan is de bit hoog.

Voorbeeld:

Is de bit met waarde 32 hoog in 42?

```
32 =  0010 0000
42 =  0010 1010
-----
```

AND =0010 0000 --> dit verschilt van 0 , dus is de bit hoog.

In C# kunnen we met de onderstaande instructies de bitoperaties uitvoeren. Let op het verschil tussen een logische AND & en een conditionele AND &&.

Equality	$x == y$ $x != y$
Logical AND	$x \& y$
Logical XOR	$x \wedge y$
Logical OR	$x   y$
Conditional AND	$x \&\& y$
Conditional OR	$x    y$

## Aansturen USB data-acquisitiekaart vanuit vb.net

### Hardware

- maximaal 25 mA per uitgang, als er meer stroom nodig is kan een externe voeding gebruikt worden.
- Op de ingangen mogen slechts spanningen tussen 0 V en +5 V worden gezet.
- Uitgangen mogen niet worden kortgesloten of met andere uitgangen worden verbonden.
- Op een uitgang mag geen (externe) spanning worden gezet.

De USB poort kan slechts een beperkte hoeveelheid stroom leveren. Eis je teveel stroom van je poort dan kan je die beschadigen. Let dus op bij het verplaatsen van jumpers (schakelaars) of het veroorzaken van kortsluitingen.

Deze data-acquisitiekaart met USB-aansluiting heeft 11 digitale uitgangen, acht digitale ingangen, twee 10-bits analoge uitgangen en vijf 10-bits analoge ingangen met eeningangsspanning van 0 tot 5 V. Het hart van het systeem is een Microchip USB-microcontroller type PIC18F4550, geprogrammeerd in C. Die microcontroller is de USB-interface tussen de PC en de elektrische schakeling. De controller is uitgerust met speciale hardware en software om het USB protocol op een gemakkelijke manier te implementeren.

### De USB data-acquisitiekaart

### Algemeen

Het schema van de kaart is te zien in de map "USB daq board". Op onderstaande foto kunnen we verschillende zaken lokaliseren.



Linksonder vinden we de aansluiting van de USB fiche terug. Na het aansluiten van de kabel moet in het midden van het board de groene power LED oplichten.

Rechts van de USB fiche kunnen we optioneel een externe AC-voeding aansluiten.

Via de jumper kunnen we dan een keuze maken tussen de twee bronnen.

Midden onder de microcontroller vind je twee rode status leds terug. Eén van die twee leds knippert. Als het stuurprogramma is geïnstalleerd, gaan de status leds om de beurt knipperen. Het printje is dan klaar voor gebruik.

## **Digitale uitgangen**

Rechtsboven vinden we 8 digitale uitgangen van poort D terug. Poort D is bi-directioneel, we kunnen zowel schrijven naar poort D en we kunnen ook de status van poort D opvragen. Hun status wordt weergegeven met een rode Led. Parallel op de leds worden die uitgangen naar buiten gebracht via de witte connector J4 (boven het LCD display). Die 8 uitgangen worden ook gebruikt om het LCD display te bedienen. Twee ervan worden gebruikt voor het aansturen van het 7-segment display. (rechtsonder). Drie uitgangen van poort D worden gebruikt om het LCD display aan te sturen.

## **Digitale ingangen**

De digitale ingangen zijn voorzien van interne pullup-weerstanden. De digitale ingangen zijn aangesloten op poort B. De ingangen hebben dus standaard 5 Volt spanning of m.a.w. de ingangen hebben als standaard waarde een digitale 1.

Aan de rechterkant van de kaart kunnen we 4 digitale ingangen aansluiten op de connector. Parallel op twee van die ingangen is er een drukknop aangesloten.

De overige digitale ingangen worden gebruikt door I<sup>2</sup>C en de draaischakelaar.

## **Analoge uitgangen**

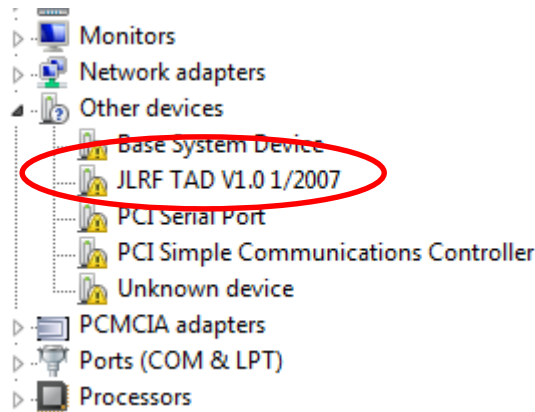
Linksonder vind je twee groene leds terug. De felheid waarmee die leds branden geven de status van de analoge uitgangen weer. De twee 10 bits analoge uitgangen zijn ook verbonden met de connector linksonder.

## **Analoge ingangen**

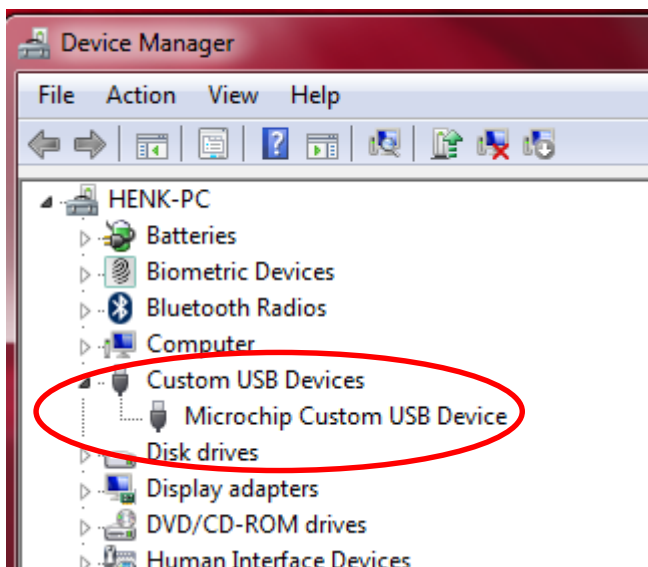
Op twee analoge ingangen zijn er twee potentiometers aangesloten. Twee andere analoge ingangen kunnen op de connector worden aangesloten.

## Installatie

1. Sluit de USB data-acquisitiekaart aan op de PC. De PC zoekt de device driver JLRF TAD V1.0 1/2007. De groene power led in het midden van het board moet branden. 1 rode led in het midden van het board knippert snel. Als de driver correct geïnstalleerd is moeten de leds beurtelings knipperen.  
De correcte driver wordt niet gevonden en onder device manager krijgen we onderstaande foutmelding.

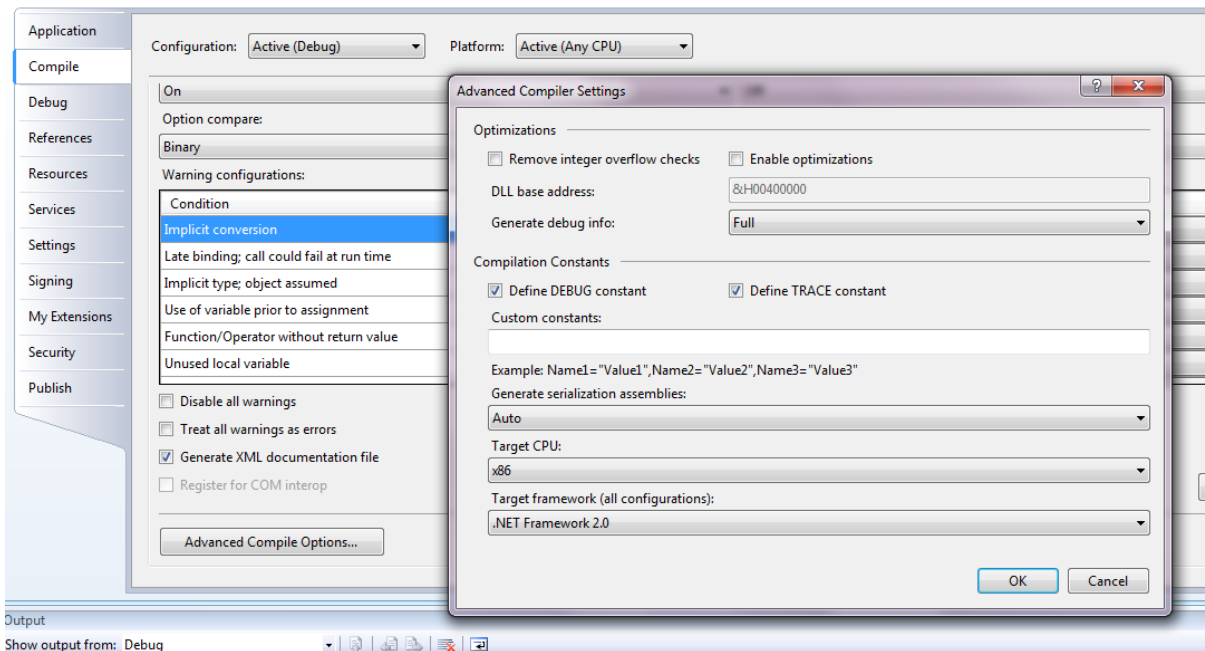


2. De nodige stuurprogramma's (mchpusb.inf) kan je in de map driver terugvinden. Pas het stuurprogramma aan en browse naar de map "driver".  
Als alles goed verlopen is, is er een microchip custom usb device toegevoegd in de device manager. De twee rode leds op het board moeten nu beurtelings knipperen.



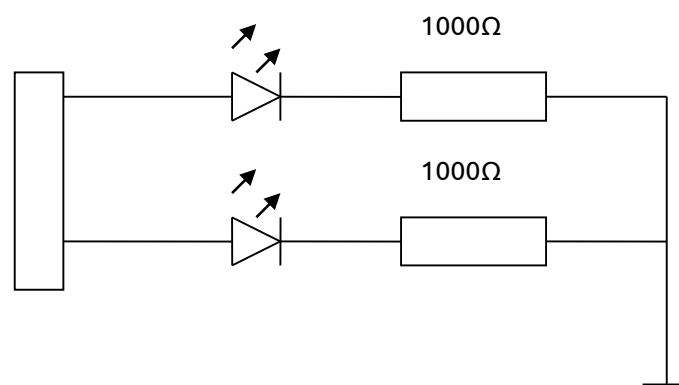
1. Controleer of de file mpusbapi.dll zich in de c:\Windows\system32 directory of in de c:\windows\sysWow64 map voor de 64 bit systemen bevindt. Indien niet moet je de file manueel kopiëren ofwel voeg je de file toe aan je visual studio project.

- Schakel het gebruikersaccount beheer uit. (UAC) ofwel moet je Visual studio telkens runnen als administrator. Wie met 64 bits systeem werkt moet onder de advanced compile options de de target CPU op x86 plaatsen.



- Voeg de klasse MPUSB.cs toe aan jullie project, deze klasse bevat verschillende methoden om met het board te kunnen communiceren.

## Voorbeeld aansturen LED's



alle Leds aan `MPUSB.WriteDigitalOutPortD(255);`

alle Leds uit `MPUSB.WriteDigitalOutPortD(0);`



## Opdracht week 1a: digitale in- en uitgangen

### Inleiding

We beperken ons deze week tot de digitale in- en uitgangen. We maken een klein programma waarmee we de 8 leds op het interfacebordje kunnen bedienen. We zorgen ervoor dat de gui van het programma steeds de status van de 8 leds kan weergeven en bedienen. We kunnen ook de status van de twee drukknoppen visualiseren.

### Doelstelling

Leren werken met bitoperatoren.

Leren werken met digitale in- en uitgangen.

### Stappenplan

#### Windowsproject aanmaken

Maak een nieuw WPF project aan.

- Geef het project de volgende naam:  
Groepsnr-loginnaam1--digitaalInOut-01
- Voeg de klasse MPUSB.cs toe aan het project.
- Importeer de file "mpusbapi.dll".

#### Window ontwerpen

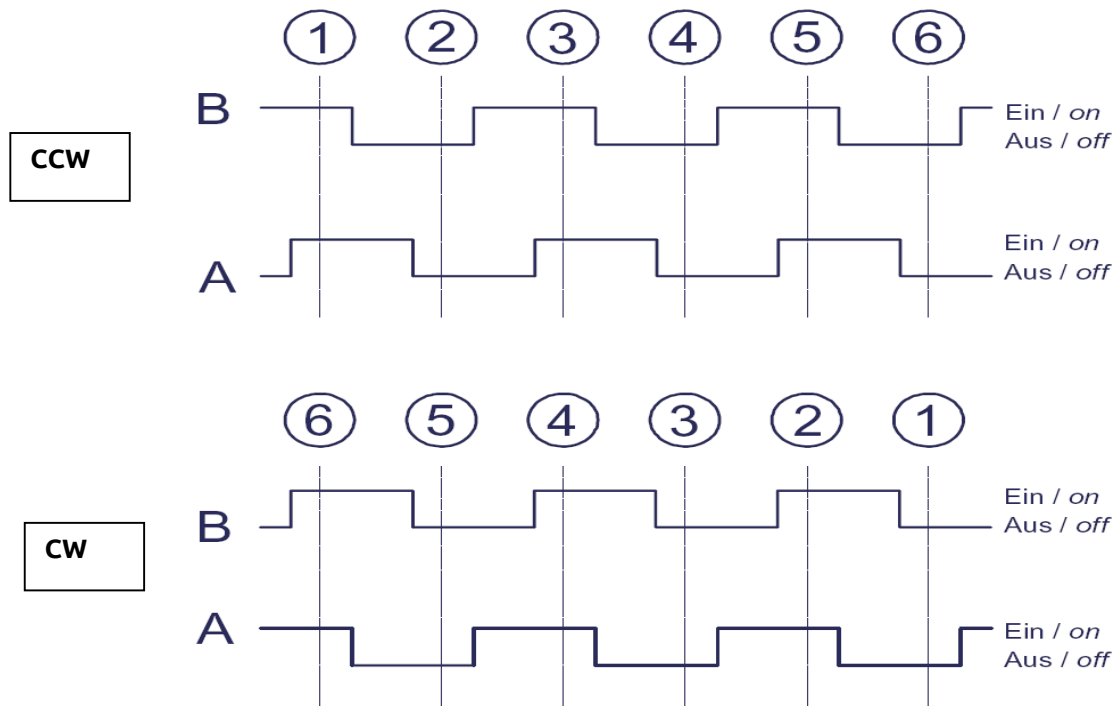
- Plaats 8 buttons op de GUI.
  - Open (btnOpen)
  - Close (btnClose)
  - Versie (btnVersie)
  - Leds aan (btnLedsAan)
  - Leds uit (btnLedsUit)
  - Omhoog (btnOmhoog)
  - Omlaag (btnOmlaag)
  - Teller (btnTeller)
- Plaats een tekstvak (txtVersie)
- Voorzie 8 buttons of checkboxes om de status van de leds weer te geven en om de leds afzonderlijk te kunnen bedienen. Plaats die 8 controls in een groupbox.
- Voorzie 2 buttons of checkboxes om de status van de drukknoppen van het acquisitiekaartje te kunnen weergeven.
- Voeg een timerobject toe.

## Reageer op verschillende events

- Open en sluit de connectie met de acquisitiekaart. Vraag de versie-info op en visualiseer ze in het tekstvak. Deze functie geeft een 0 terug als de connectie met het daq-board geslaagd is, bij een -1 is de connectie niet gelukt.
- Laat alle LED's branden en doven. (btnLedsAan en btnLedsUit) Visualiseer op de GUI
- Vraag telkens (met de timer) de status van de LED's op en visualiseer. Gebruik hiervoor het commando readDigitalPortD. (tip : gebruik de tag-property van bv. de checkbox en vul daar de bitwaarde in van de overeenkomstige led.)
- Laat de leds tellen van 0 tot 255.(button Teller) Visualiseer op de GUI
- Zorg ervoor dat elke LED afzonderlijk kan bediend worden en visualiseer.
  - Zet de overeenkomstig LED aan of af met behulp van de booleaanse and,not en or instructies. (maak gebruik van de bitwaarden in de tag property)
- Visualiseer de status van beide drukknoppen. Filter de correcte bits uit de ingelezen byte. Lees met het tick event van de timer de status van de drukknoppen binnen om 100 ms.
- Via de buttons omhoog en omlaag gaan we een ingesteld bitpatroon verschuiven. Maak gebruik van shiftoperator (<< en >>) om de bits te verschuiven en visualiseer.  
Bv.  $Y = X \gg 1$      $X = 0011\ 0000$   
 $Y = 0001\ 1000$
- Zorg er voor dat een brandende led niet verdwijnt. Stel dat LED 1 brandt en de bits worden naar boven geshift dan moet LED 8 gaan branden.
- Verschuif het bitpatroon met behulp van een trackbar.

## Impulsmomentgever. (zotte knop)

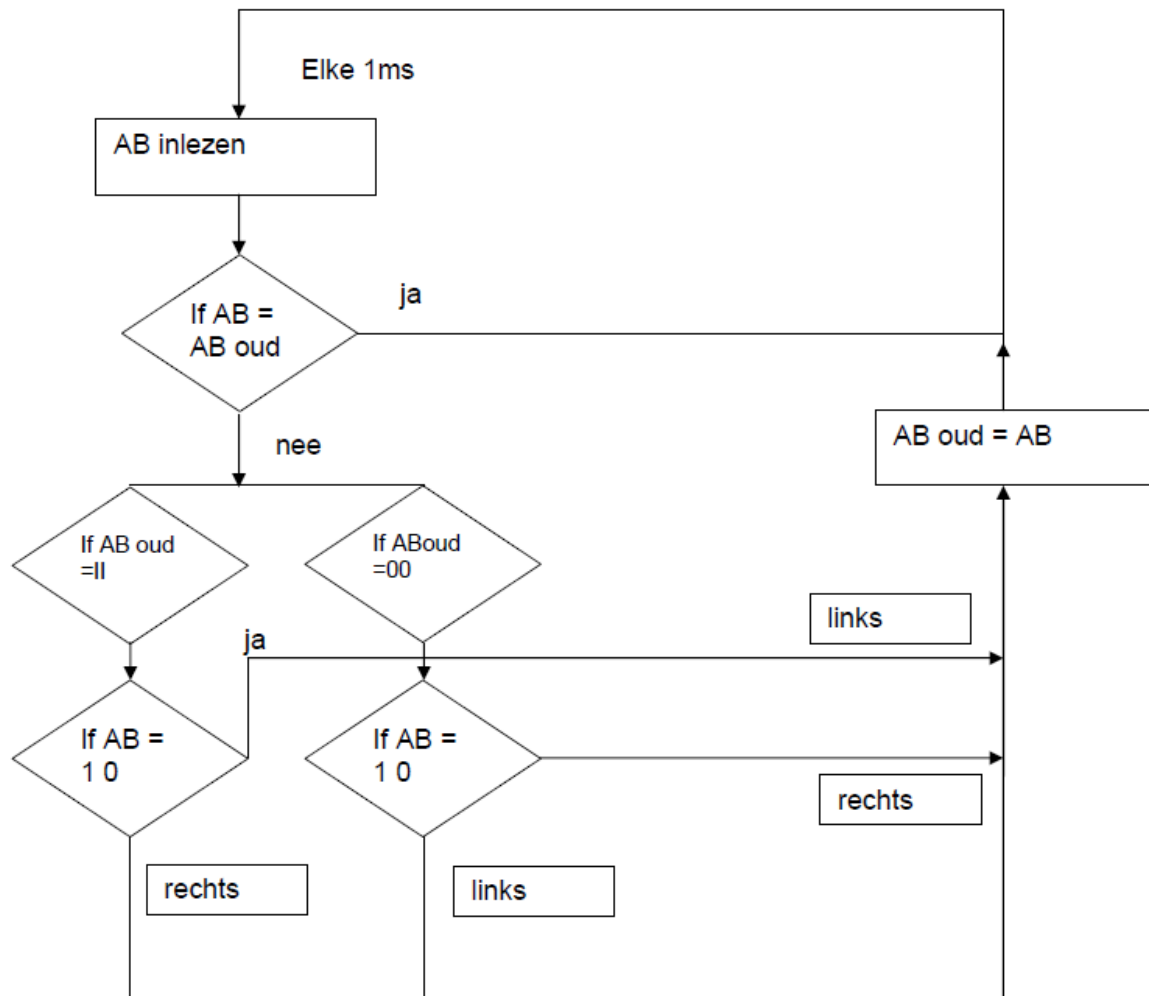
De impulsmomentgever is een knop zonder begin – en eindstand. De stand van de knop kan dan ook niet worden opgevraagd. Er kan enkel gedetecteerd worden of er naar links of naar rechts wordt gedraaid aan de knop. Er kan ook op de knop worden gedrukt. De impulsmomentgever is aangesloten op drie digitale ingangen van poort B. Eén ingang detecteert of er al dan niet op de knop wordt gedrukt. Twee andere ingangen (A&B) hebben we nodig om te bepalen of er naar links of naar rechts wordt gedraaid.



In ruststand (er wordt niet aan de knop gedraaid) zijn de twee ingangen van de zotte knop (A & B) ofwel beide hoog ofwel beide laag. Enkel tijdens het draaien zelf kan de status van beide ingangen (A & B) verschillend zijn.

Om de draairichting van de knop te bepalen zullen we moeten kunnen bepalen welke van de twee ingangen er eerst van toestand veranderd, daarom zullen we voldoende snel (elke milliseconde) de status van de ingangen moeten opvragen.

B.v. als er naar rechts wordt gedraaid (CW) verandert ingang A altijd als eerste van status. Als er naar links wordt gedraaid (CCW) verandert ingang B altijd als eerste van status.



Tip : om foutieve data te vermijden wordt best de else voorwaarde ook gecontroleerd.

B.v. If AB = 10 .....else if AB = 01 .....

## Multithreading, de backgroundworker

In deze oefening gaan we werken met een extra thread om de status van de ingangen op te vragen. Belangrijk is hierbij dat threads niet zomaar naar gui objecten mogen schrijven. Enkel de main thread mag dat doen. Indien een thread dat wel doet, zijn de gevolgen onvoorspelbaar. Soms zal het werken, dan weer niet. Ga dus nooit vanuit een extra thread, bijvoorbeeld een label.text gaan invullen. Vanaf visual studio 2005 is deze manier van werken niet meer toegelaten.

Het aanmaken van een thread neemt wat tijd in beslag en is niet zo eenvoudig. Indien een programma om de haverklap een thread maakt, die dan weer stopt, is dit niet zo efficiënt.

Vanaf visual studio 2005 kunnen we gebruik van de Backgroundworker klasse. Die klasse neemt ons een hoop werk uit handen en maakt het ons een stuk makkelijker om asynchrone acties in Windows-applicaties uit te voeren.

De backgroundWorker werkt met events. De belangrijkste events van het BackgroundWorker-object zijn :

- DoWork : Dit event markeert de start van de methode die op de achtergrond moet worden uitgevoerd. (voor ons wordt dit het pollen van de ingangen)
- ProgressChanged : Zodra de BackgroundWorker actief is, kan het de status van de uitgevoerde methode doorgeven via het ProgressChanged-event. Via een ReportProgress wordt dit event aangeroepen.
- RunWorkerCompleted : Dit event verschijnt zodra de methode klaar is. Het kan zijn dat de methode is afgebroken of de methode kan daadwerkelijk klaar zijn met het uitvoeren van het programma.

Hier kunnen we gebruik maken van de methodes RunWorkerAsync(),CancelAsync(),...

Via de methode RunWorkerAsync() wordt het DoWork event op geroepen, in het DoWork event kunnen we een loopke schrijven die voortdurend de status van poortB controleert, als de status van de ingangen wijzigt kunnen we de loop verlaten. Hierdoor wordt het de RunWorkerCompleted event aangeroepen. Nu is het toegestaan om iets in de GUI aan te passen, de extra thread is immers stopgezet. Via e.result kunnen we objecten op een veilige manier terug naar de main thread sturen. Nu moeten we natuurlijk via de methode RunWorkerAsync() terug het DoWork event aangeroepen, zodat we de status van de ingangen opnieuw kunnen opvragen.

Een andere manier van werken is om via het ProgressChanged Event data uit te wisselen tussen de verschillende threads. De DoWork routine controleert voortdurend de status van poortB. Als de status van poortB wijzigt, roepen we het ProgressChanged event aan. Via de argumenten (integer, object) kunnen we op een veilige manier data tussen de verschillende threads doorgeven.

Via de methode CancelAsync() kan de backgroundWorker gestopt worden. In het DoWork event kan je via de CancellationPending property de cancel property van het cancelEventArgs op true plaatsen. Let op de property CancellationPending werkt enkel in het DoWork event.

```
If bgwIngangen.CancellationPending Then
    e.Cancel = bgwIngangen.CancellationPending
End If
```

## Opdracht week 1b : impulsmomentgever

### Inleiding

Je kan verder werken aan de oefening van daarnet.

### Window ontwerpen

- Plaats een tekstvak of een label op de GUI om de bytewaarde van poort B weer te geven.
- Voorzie een label om weer te geven of er naar links of rechts gedraaid wordt aan de knop.
- Voeg een backgroundworker object toe aan het project.

### Reageer op verschillende events

- In plaats van de dll mpusbapi.dll te kopiëren naar de system map. Kan je die file ook toevoegen aan het project. Vergeet wel niet de property "copy to output directory" op "copy always" te plaatsen.
- Laat de waarde van poort B zien op de GUI.
- Bepaal op welke 3 bits de ingangen van de zotte knop zijn aangesloten.
- Laat op de GUI zien of er naar links of naar rechts wordt gedraaid met de knop. Probeer dit eerst uit met een timerobject. Plaats het tick interval op 1ms.
- Wordt er op de eerste drukknop van het bordje gedrukt wordt alles leds uitgezet.
- Zolang we de 2<sup>e</sup> knop indrukken moeten alle Leds branden, bij het loslaten van de knop worden de LED's terug in de vorige toestand gebracht.
- Als er één keer op de zotte knop wordt gedrukt kunnen we het bitpatroon naar links of naar rechts laten bewegen door aan de knop te draaien. Drukken we nogmaals op de knop, dan kan het bitpatroon niet meer verplaats worden met de zotte knop.
- Vervang de timers door een backgroundworker object. Voor extra uitleg zie : [http://msdn.microsoft.com/en-us/library/cc221403\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc221403(v=vs.95).aspx)