

Opgave Tussentijdse Oefeningen Jaarproject I

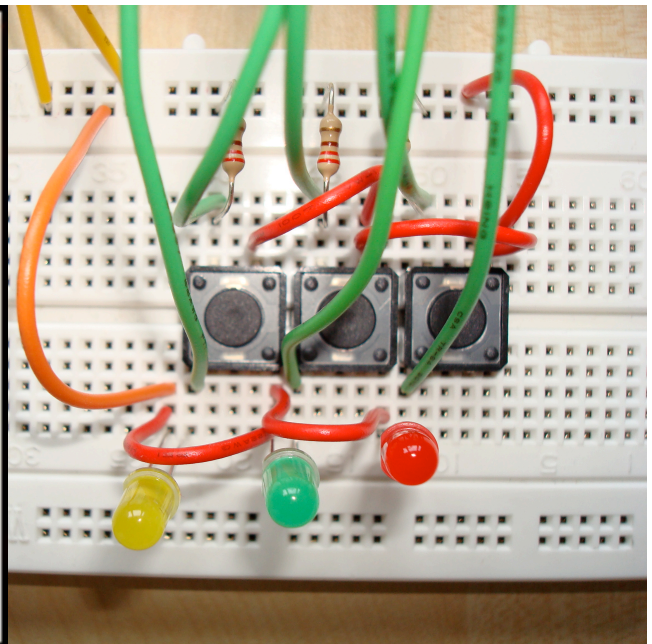
Reeks 1: Input/Output

1 Introductie

Deze oefening bestaat eruit om een kleine *simon* [1,2] te maken door middel van drie ledjes en drie drukknopjes. Het spel verloopt als volgt: eerst laat de computer één ledje aan en uit gaan. Vervolgens moet de speler het knopje indrukken dat overeenkomt met dat ledje. Als de speler het juiste knopje indrukt, laat de computer opnieuw hetzelfde ledje aan en uit gaan.

Achtereenvolgens laat de computer opnieuw een ledje aan en uit gaan. Deze keer moet de speler de overeenkomstige knopjes (zowel voor het knipperen van het eerste als het tweede ledje) indrukken in de juiste volgorde. Als bijvoorbeeld eerst het gele ledje knippert gevolgd door het rode ledje, dan moet de speler eerst de meest linkse knop indrukken en nadien de meest rechtse knop.

Hoe we een ledje kunnen laten knipperen hebben we al getoond en de code hiervoor staat online. In de rest van de opgave tonen we nog hoe je een drukknop kan aansluiten en uitlezen.



2 Herhaling

Armpit scheme is een beperkte implementatie van scheme zoals je deze geleerd hebt in het vak Structuur van Computer Programma's I. Hierdoor kan je dus scheme programma's in armpit scheme programmeren zoals je dat altijd al gedaan hebt. Echter, als we de microcontroller bepaalde functies laten uitvoeren, moeten we gebruik maken van *registers*. Door bepaalde waarden naar deze registers te schrijven, kunnen we bijvoorbeeld een ledje laten knipperen, tijd uitlezen, en zelfs communiceren met een computer. In deze opgave zullen we het **GPIO** (general purpose input output) register onder de loep nemen.

Schrijven naar een register kunnen we doen met de functie *write*, lezen doen we met de functie *read*.

[1] <http://www.bigredtoybox.com/articles/simonindex.shtml>

[2] <http://www.neave.com/games/simon/game.php>

3 Drukknop lezen

Om een drukknop in te lezen moeten we deze verbinden met een *input*-pin van de microcontroller. De microcontroller die we voor dit project gebruiken heeft speciale pinnen die we zowel kunnen gebruiken voor input als voor output. We moeten deze echter wel instellen zodanig dat de microcontroller weet of we een bepaalde pin gebruiken om een ledje te laten branden (output) of om een knopje in te lezen (input).

Pin Name	Type	Description
P0.0 - P0.31 P1.16 - P1.31	Input/ Output	General purpose input/output. The number of GPIOs actually available depends on the use of alternate functions.

In de datasheet kunnen we zien dat P0.0 (poort nul, pin 0) tot P0.31 (poort nul, pin 31) beschikbaar zijn voor zowel input als output. Enkele pinnen hebben echter alternatieve functies; deze pinnen worden bijvoorbeeld gebruikt om te communiceren met je computer. Omdat deze pinnen al gebruikt worden door de scheme interpreter, kan je deze dus niet in je project gebruiken om bijvoorbeeld een ledje te laten branden.

Om in te stellen of we een pin gaan gebruiken als input of output moeten we een bepaalde waarde schrijven naar het **IODIR** register. Elke bit in dit register komt overeen met een pin van de microcontroller. Wanneer we een *één* schrijven naar een bit uit dit register komt dit overeen met het instellen van de overeenkomstige pin als een *output*-pin. De tabel uit de datasheet die aangeeft hoe we dit register kunnen gebruiken is hieronder afgebeeld.

IODIR	Description	Value after Reset
31:0	Direction control bits (0 = INPUT, 1 = OUTPUT). Bit 0 in IO0DIR controls P0.0 ... Bit 31 in IO0DIR controls P0.31	0

In het blinking led voorbeeld hebben we voor het instellen van een *output*-pin de volgende functie gebruikt.

```
(define (set-dir-output! port pin)
  (write (logior (read port IODIR) pin) port IODIR))
```

Deze functie leest eerst de oude waarde van het **IODIR** register en neemt vervolgens de logische *or* met de meegegeven pin waarde. Tenslotte wordt dit resultaat opnieuw naar het **IODIR** register geschreven. Dit komt overeen met het schrijven van een *één* naar de bit in het register die overeenkomt met de meegegeven pin. Je zou een soortgelijke functie kunnen maken voor het instellen van een input poort. Voor dit project is dat echter niet nodig aangezien de waarde van **IODIR** gelijk is aan *nul* telkens we de controller herstarten (dit is weergegeven in de laatste kolom van de tabel). Dit houdt dus in dat elke pin ingesteld is als input na het herstarten van de microcontroller.

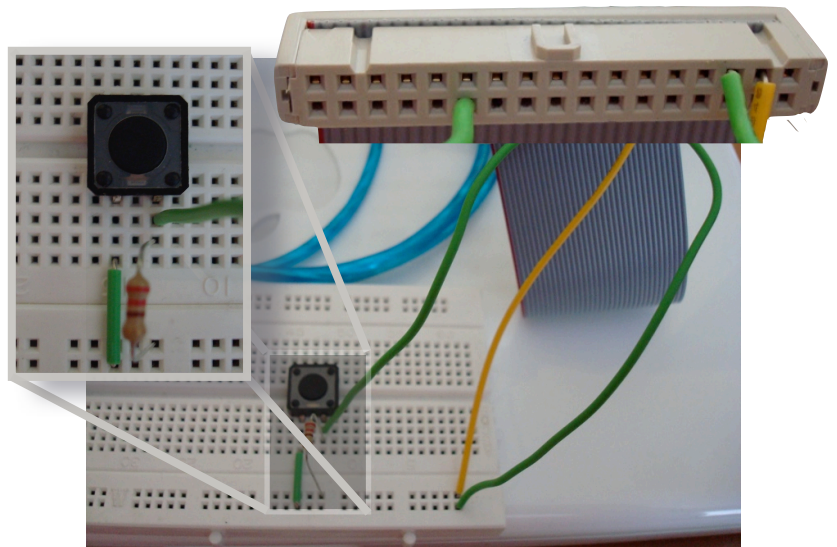
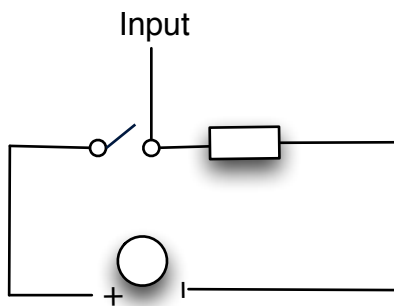
Nu we exact weten hoe we onze microcontroller kunnen instellen voor input/output rest ons nog de vraag hoe we de waarde van een *input*-pin kunnen inlezen. Dit kunnen we doen door het pin-status register in te lezen. We lezen dit register uit en nemen de logische *and* met de pin hierdoor krijgen we ofwel de pin waarde zelf terug ofwel *nul*.

```
(define (read-pin port pin)
  (logand pin (read port IOPIN)))
```

4 Druknop aansluiten

In de vorige sectie hebben we gezien hoe we een input-pin kunnen uitlezen, maar we hebben nog niet gespecificeerd wat er nu juist gelezen wordt. De ingelezen waarde die we terugkrijgen geeft aan of de input-pin verbonden is met de positieve kant of de negatieve kant van de bron. Dit komt erop neer dat wanneer we de input-pin verbinden met positieve kant van de bron we een *één* uitlezen in het andere geval een *nul*. Wanneer we de input-pin niet verbinden noemt dit een *zwevende pin*. Het lezen van een zwevende input pin kan zowel een *één* als een *nul* als resultaat hebben.

We kunnen een kleine schakeling maken die ervoor zorgt dat de input-pin verbonden is met de positieve kant van de bron als we de drukknop indrukken en met de negatieve kant als we de drukknop loslaten. Deze schakeling is hieronder weergegeven, we zien hier de *bron*, een *schakelaar* en een *weerstandje*. De weerstand heeft geen polariteit en de richting waarin je deze aansluit heeft dus geen belang. In het pakket dat we hebben meegegeven zitten verschillende weerstanden, de weerstand die je hier nodig hebt heeft eerst twee rode strepen gevolgd door een bruine en een gouden streep. Naast het schema zie je een hoe je de drukknop uit het pakket kan aansluiten op het *breadbord*.



5 Druknop testen

De volgende code kan je gebruiken om te kijken of je schakeling werkt, in het voorbeeld is input verbonden met pin 5.

```
(define GPIO_0 #xE0028000)
(define IODIR #x08)
(define IOSET #x04)
(define IOCLR #x0C)
(define IOPIN #x00)

(define (read-pin port pin)
  (logand pin (read port IOPIN)))

(define (test port pin)
  (if (= (read-pin port pin) 0)
      (test port pin)
      (display "Gedrukt")))

(test GPIO_0 (ash 1 5))
```