

Een schakelaar aansluiten

In dit artikel gaan we verder experimenteren met de EPP0-programmer en het bijbehorende demoboord dat op de radiomarkt in Rosmalen geïntroduceerd is op de stand van museum Jan Corver. De lezers die dit demoboord niet hebben, kunnen wellicht door het lezen van dit artikel en het bekijken van de software van de site hun kennis op PIC-gebied uitbreiden. Ik ga ervan uit dat het maken van een project binnen MPLAB, laden van de ASM-file, genereren van de HEX-file en het programmeren van de PIC-processor met de EPP0-programmer bekend is na het lezen van het vorige artikel. Hier ga ik verder niet op in.

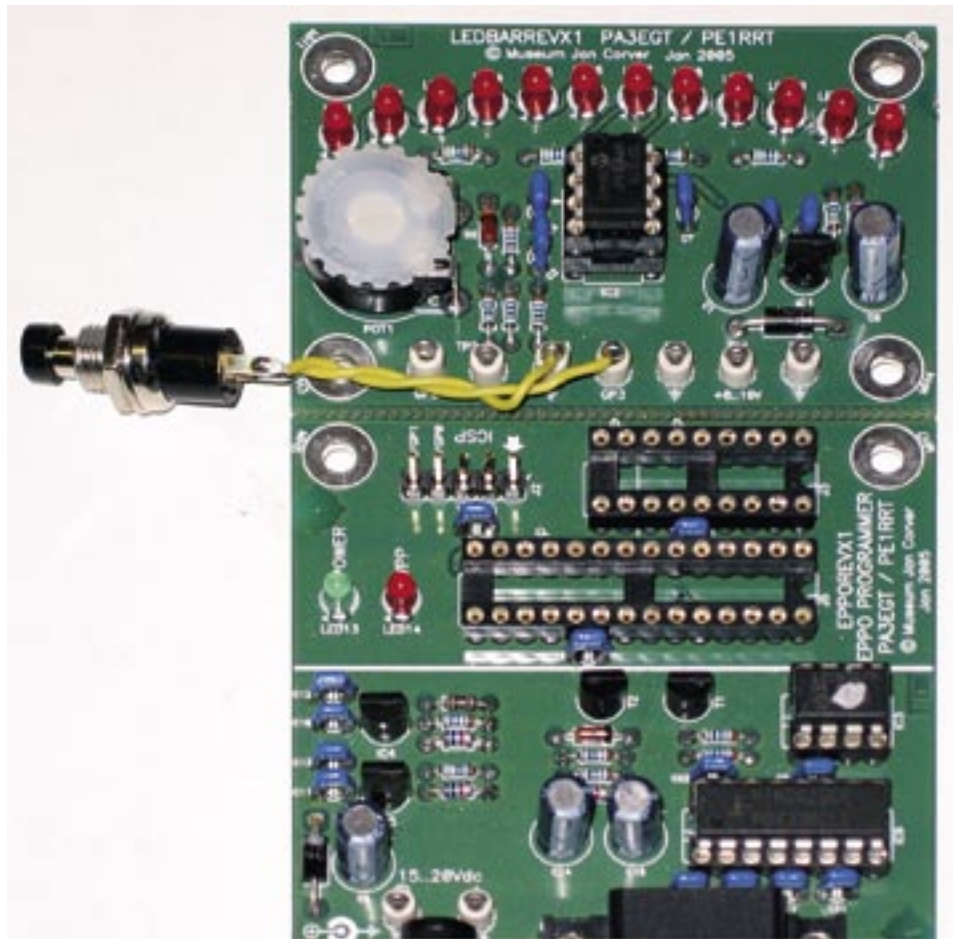
Nu iedereen uitgespeeld is met het aansturen van de LEDs zullen we een poging wagen een schakelaar op het demoboord aan te sluiten. Aansluiten zal het probleem niet zijn maar we willen deze schakelaar uit kunnen lezen in onze software. Ik heb bij dit voorbeeld een drukschakelaar gepakt met een maakcontact. Deze lag toevallig vooraan in de junkbox. Later zullen we zien dat het gebruik van een maak- of verbreekcontact simpel binnen de software te regelen is. In dit geval is de schakelaar aangesloten op TP3 en TP4 op het demoboord, zijnde massa en IO-pin GP3.

Werking van de schakeling

Het doel dat ik me heb gesteld is om door een druk op de knop te kunnen wisselen tussen het laten branden van LED1 en LED7. Ik wil voor ik de andere LED laat branden ook zeker weten dat de schakelaar losgelaten is. Een oplossing voor een probleem als dit is op diverse manieren te benaderen. Ik ben zelf van huis uit geen programmeur en heb ook niet de intentie (of de capaciteiten) om perfecte software te schrijven. Ik probeer in de programma's die ik schrijf zo veel mogelijk zaken te voorzien en ben over het algemeen blij als het werkt en ik ook weet waarom. Het volgende programma zal qua opzet geen schoonheidsprijs winnen maar in deze structuur is wel heel duidelijk stap voor stap de werking te zien. De volgende stappen zijn te onderscheiden:

```
Start
LED1 aan
    Is schakelaar ingedrukt?
    Is schakelaar losgelaten?
LED7 aan
    Is schakelaar ingedrukt?
    Is schakelaar losgelaten?
Ga naar start
```

Het uitlezen van de schakelaar zullen we in een loop doen: we blijven net zo lang kijken naar de status van de schakelaar tot deze



Schakelaar aangesloten op het demoboord van de EPP0-programmer

start			; begin hoofdprogramma
	call	LED1aan	; laat LED1 branden
			; in deze routine wordt ook GP3 meteen als ingang gedefinieerd
push1			; loop om te testen of schakelaar ingedrukt is
	btfdc	pushbutton	; sla als schakelaar is ingedrukt de volgende instructie over
	goto	push1	; ga naar push1, loop zolang schakelaar niet ingedrukt is
	call	delay	; anti-dendervertraging
	btfdc	pushbutton	; sla als schakelaar is ingedrukt de volgende instructie over
	goto	push1	; ga naar push1, loop zolang schakelaar niet ingedrukt is
release1			; loop om te testen of schakelaar losgelaten is
	btfdc	pushbutton	; sla als schakelaar is losgelaten de volgende instructie over
	goto	release1	; ga naar release1, loop zolang schakelaar niet losgelaten is
	call	delay	; anti-dendervertraging
	btfdc	pushbutton	; sla als schakelaar is losgelaten de volgende instructie over
	goto	release1	; ga naar release1, loop zolang schakelaar niet losgelaten is
	call	LED7aan	; laat LED7 branden
push2			; loop om te testen of schakelaar ingedrukt is
	btfdc	pushbutton	; sla als schakelaar is ingedrukt de volgende instructie over
	goto	push2	; ga naar push2, loop zolang schakelaar niet ingedrukt is
	call	delay	; anti-dendervertraging
	btfdc	pushbutton	; sla als schakelaar is ingedrukt de volgende instructie over
	goto	push2	; ga naar push2, loop zolang schakelaar niet ingedrukt is
release2			; loop om te testen of schakelaar losgelaten is
	btfdc	pushbutton	; sla als schakelaar is losgelaten de volgende instructie over
	goto	release2	; ga naar release2, loop zolang schakelaar niet losgelaten is
	call	delay	; anti-dendervertraging
	btfdc	pushbutton	; sla als schakelaar is losgelaten de volgende instructie over
	goto	release2	; ga naar release2, loop zolang schakelaar niet losgelaten is
	goto	start	; ga naar start, LED1 gaat dan automatisch weer aan

verandert. Voor een betere leesbaarheid is dit hierboven niet opgenomen. In het uiteindelijke programma is het echter goed te herkennen. Het op deze manier geschreven programma is natuurlijk ontzettend inefficiënt omdat we een groot gedeelte van onze tijd aan het wachten zijn totdat toevallig een keer de schakelaar ingedrukt of losgelaten wordt. In complexere programma's kunnen we er bijvoorbeeld voor kiezen om het inlezen van een schakelaar in een subroutine op te nemen die we aanroepen als dat nodig is. We houden zo processortijd over om andere nuttige zaken te doen. Wellicht hebben we het daar een volgende keer over. De vertaling van deze stappen naar het programma binnen MPLAB staat in de programmalisting. Om ruimte te sparen heb ik alleen de essentiële en nieuwe zaken opgenomen. Voor uitleg betreffende de initialisatie en het aansturen van de LEDs verwijs ik graag naar het vorige artikel. De complete en werkende file is te downloaden vanaf de site van het museum. Alles achter puntkomma's is commentaar en geen onderdeel van het programma.

We zien dat er voor het uitlezen van de schakelaar gebruik is gemaakt van BTFSC en BTFSS, Dit zijn twee handige instructies om de status van een bit uit te lezen. BTFSS

staat voor Bit Test F, Skip if Set. In het Nederlands wil dat zeggen dat er gericht naar een bepaald bit van een bepaald register (of geheugenadres) wordt gekeken. Als dit bit hoog is, wordt de volgende instructie overgeslagen. Om het inzichtelijker te maken, schrijven we een programmaregel waarin we deze instructie gebruiken voluit. We gebruiken dan het volgende: BTFSS GPIO,3. Concreet gaan we dan testen of bit 3 van GPIO hoog is. Als dit het geval is, slaan we de volgende instructie over. Bit 3 staat voor GP3 in het GPIO-register. Dit is te zien op pagina 19 van de datasheet. We zien hier ook dat we altijd met bit 0 beginnen te tellen. In het programma staat echter 'BTFSS pushbutton'. In het begin van het complete programma (niet opgenomen in de hier getoonde listing) heb ik gedefinieerd dat de naam 'pushbutton' gelijk is aan 'ingang GP3'. Door namen aan bijvoorbeeld ingangen te koppelen is het mogelijk complexere programma's toch overzichtelijk te houden. De instructie die ik hiervoor heb gebruikt is: #define pushbutton GPIO,3.

We bekeken de BTFSS-instructie. Het lijkt haast overbodig om te zeggen dat BTFSC staat voor Bi Test F, Skip if Clear. Als een bit laag is, wordt de volgende instructie overgeslagen. Door BTFSS en BTFSC van plaats te verwisselen kunnen we een verbreekscha-

kelaar gebruiken. Als we naar het schema kijken, zien we dat de ingang voorzien is van een pull-upweerstand. Als de schakelaar ingedrukt wordt, wordt de ingang dus laag. Het is belangrijk om daar rekening mee te houden.

Eén ding hebben we nog niet behandeld en dat is de vertraging die tussendoor wordt aangeroepen. Het drukschakelaartje dat ik heb gebruikt is een zeer goedkoop prul dat last had van denderverschijnselen. Het gevolg hiervan is dat het voor kan komen dat de LEDs een aantal keer geschakeld worden door één toetsdruk. Dit is opgelost door na detectie van het indrukken 100 ms te wachten en dan nog eens te controleren of de toets ingedrukt is. Hetzelfde doen we bij 'loslaten'. We hebben nu een betrouwbare manier gevonden om een schakelaar in onze programma's te gebruiken. Meestal wacht ik zo'n 10 ms, dat is afdoende. Aangezien we al een universele vertraging in ons eerdere programma hadden geprogrammeerd, heb ik (gemakkelijk uitgevallen als ik ben) simpelweg het aantal herhalingen op één gezet en zodoende een vertraging van 100 ms gecreëerd. Het complete programma is hier te vinden: <http://www.jancorver.org/bouw/eppo/>