

# De joystickpoorten van de MSX deel 2

## Deel 2: Analoge input

Peter van Overbeek

PTC-Print, februari 1991 - Nummer 44

*Scanned, ocr'ed and converted to PDF by HansO, 2001*

Voor veel toepassingen kunnen we goed uit de voeten met binaire keuzen, dat wil zeggen een keuze uit twee mogelijkheden: Aan of Uit, Open of Dicht, Licht of Donker, ja of Nee, Koud of Warm enzovoort. Voor andere toepassingen wensen we juist wat meer gradaties. In die gevallen zouden we bijvoorbeeld willen weten hoeveel licht er is, of wat de temperatuur is. Ookdaarin blijkt ons de MSX-computer te hulp tekomen. De mogelijkheden, maar ook de beperkingen van analoge input via de joystickpoorten zullen we hier eens wat nader bekijken. Daarbij besteden we vooral aandacht aan een simpel maar zeer nuttig stukje hardware: de zogeheten Paddle, die ik maar zo blijf noemen omdat ik erzogauwgeen goed Nederlands woord voor weet.

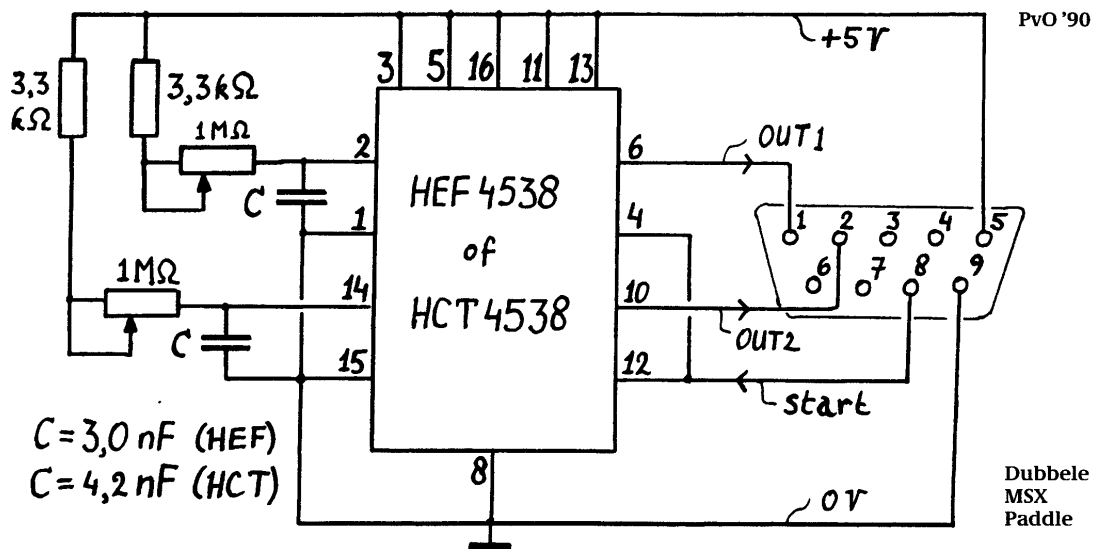
### Een Paddle?

Het MSX Technical Data Book - de MSX Bijbel dus - geeft in het hoofdstuk: Hardware Specificaties aan, dat Paddles aangesloten kunnen worden op de joystickpoorten. Het geeft ook een principeschema van zo'n ding. Een paddle heeft een regelknop, vergelijkbaar met de knop waarmee je de radio hard of zacht kan zetten. De stand van die knop kan via de BASIC-instructie PDL door de MSX-computer vastgesteld worden, zodat het resultaat in het programma gebruikt kan worden. Op elke joystickpoort kunnen maximaal zes paddles aangesloten worden. Hier in Nederland heb ik ze nooit te koop gezien, misschien omdat ze door elke enigszins handige knutselaar eenvoudig zelf te maken zijn. Dit in tegenstelling tot een muis, waar heel wat meer voor komt kijken, maar die gelukkig dan ook wel kant en klaar te koop is. Als een muis bewogen wordt over een plat vlak, kan de computer daaruit twee getallen afleiden voor de verplaatsing in twee loodrechte richtingen. Een paddle levert per knop slechts een getal dat afhankelijk is van de hoek waarover deze knop is gedraaid vanaf zijn begin-stand. De muis lijkt in het voordeel te zijn, vergeet echter niet dat per joystickpoort slechts een muis maar zes draaiknoppen aangesloten kunnen worden. Voor sommige toepassingen is de muis handiger, voor andere is een paddle te preferen. Een heel belangrijk voordeel van het principe van de paddle is, dat dit ook te gebruiken is voor allerlei andere toepassingen waar we met analoge input te maken hebben.

### Werking

Een draaiknop kan tussen zijn begin-en eindstand een oneindig aantal verschillende tussenstanden innemen. De draaihoek is dan ook een analoge waarde. De Paddle geeft, na

een start-sigitaal van de computer, een impuls af waarvan de lengte afhankelijk is van deze draaihoek. Deze impuls lengte is dus ook nog een analoog signaal. De computer vertaalt nu de impuls lengte naar een digitaal (letterlijk betekent dat: op de vingers aftelbaar) resultaat dat van 0 tot 255 kan lopen: in totaal 256 verschillende waarden. We hebben hier dan ook te maken met een vorm van Analooq naar Digitaal Conversie (omzetting), meestal kortweg met ADC aangeduid. Dit gaat als volgt in z'n werk. Wanneer BASIC de instructie PDL(N) tegenkomt (N=1..12) geeft de computer een korte startsignaal op pen 8 van de bijbehorende joystickpoort. Dat is op poort 1 als N oneven is en op poort 2 als N even is. De computer zet nu een teller op nul en onderzoekt de toestand van de met N aangeduide ingang (zie in de vorige aflevering welke dat is). Als deze ingang "hoog" is, wordt de teller opgehoogd en de ingang opnieuw onderzocht. Dit gaat zo door tot de ingang "laag" is geworden of totdat de teller op 255 staat. De bereikte tellerstand wordt aan BASIC te-ruggeleverd. Elke controle van de ingang plus telling kost bijna 12 microseconde (41 T-states voor de insiders), de maximale teltijd is ongeveer 3 milliseconde. De computer meet zo de tijdsduur van de impuls op de ingang in eenheden van 12 microseconde, te beginnen vanaf het startsignaal. Is het resultaat bijv. gelijk aan 10, dan duurde de impuls 120 microseconde of langer, maar minder dan 132 microseconde. Lieten we de ingang open ("hoog") dan telt de computer tot 255 en stopt. Was de ingang echter al kortgesloten met pen 9 toen het startsignaal kwam, dan stopt het tellen direct en is het resultaat 0.



### Zelfbouw van een Paddle

Willen we zelf een Paddle maken, dan hebben we een schakeling nodig die een regelbare impuls maakt: korter dan 12 microseconde voor resultaat 0 tot aan 3 milliseconde voor het maximum 255. Deze impuls moet beginnen op het moment dat de computer een startsignaal geeft op pen 8 van de joystickpoort. De moderne techniek komt ons hierbij te hulp, want dergelijke schakelingen zijn kant en klaar als IC te koop onder de naam Timer, Monosta-biele Multivibrator of One-Shot. We hoeven alleen nog maar een condensator en een weerstand toe te voegen, die samen de impuls tijd bepalen. Een regelbare impuls lengte krijgen we door het gebruik van een regelbare weerstand, in vaktaal een potmeter genaamd.

Schema's voor een enkele en dubbele paddle met timer-IC NE555 of NE556 zijn te vinden in MSX Nieuwsbrief nr.13.1 (bij PTC-Print nr.5 van april 1986). Ter afwisseling geef ik hierbij een versie met het IC HEF4538, waarmee een dubbele paddle is te maken (zie fig.1). Ook het IC HCT4538 kan gebruikt worden, maar dan wel met een aangepaste condensator. De impuls tijd T kan berekend worden uit:  $T=R \cdot C$  voor een HEF4538 en uit:  $T=0.7 \cdot R \cdot C$  voor een HCT4538. Mijn ervaring is overigens dat de werkelijke impuls-lengte meestal wat langer is dan de berekende. De maximale impuls tijd moet 3 msec zijn voor resultaat 255. Dit kan worden bereikt met een potmeter van 1 MOhm en een condensator van 3,0 nF voor HEF of 4,2 nF voor HCT. Nu zijn condensatoren van 3,0 nF en 4,2 nF niet overal te koop, dus zetten we eenvoudig twee wel verkrijgbare condensatoren parallel: bijv. 2,7 nF en 270 pF voor HEF of 3,9 nF en 270 pF voor HCT. Het is natuurlijk ook mogelijk een andere potmeter te gebruiken, maar dan moet ook een andere condensator gekozen worden zodat de maximale impuls lengte toch weer op 3 msec uitkomt. Dus voor een potmeter met de helft van de weerstand (500 kOhm) een condensator van 6 nF resp. 8,4 nF. Kies de weerstand overigens niet te klein, want anders is resultaat 0 niet meer te bereiken. De serieweerstand van 3,3 kOhm dient ter bescherming van het IC en mag beslist niet kleiner gekozen worden! Door het schema drie keer na te bouwen, kan een zesvoudige paddle worden gebouwd. Gebruik in dat geval behalve ingangen 1 en 2 ook 3,4,6 en 7 van de joystickpoort. Bouw het geheel in een blikken doosje dat met massa (pen 9) verbonden wordt, dit om de stoorgevoeligheid te beperken.

### Gebruik van de Paddle

Om te beginnen gaan we natuurlijk uitproberen of de paddles wel goed werken. Dat gaat met:

```
10 PRINT PDL(1),PDL(3)
20 GOTO 10
```

en nu maar aan de knoppen draaien! Haalt het resultaat niet helemaal 255, dan is de capaciteit toch nog iets te klein, maak dan de extra condensator wat groter, bijv. 330 of 390 pF. Als de uitkomst al voor het einde van de slag 255 is, kies dan de extra condensator wat kleiner: 220 of 180 pF. De paddle kan gebruikt worden net als een muis om iets op het scherm te positioneren of te verplaatsen, maar vaak zal juist daarvoor de muis handiger zijn. Een voordeel van de paddle ten opzichte van de muis is dat alle knoppen onafhankelijk van elkaar in te stellen zijn zonder de andere te beïnvloeden. Een muis is moeilijk in een richting te bewegen zonder daarbij ook de andere richting te beïnvloeden. Een voorbeeld van een toepassing:

```
10 CLS
20 P = PDL(1)+1
30 LOCATE 10,10
40 Q = PDL(3)+1
50 R=1/(1/P+1/Q)
60 PRINT USING "### kOhm parallel aan ### kOhm = ###.# kOhm";P;Q;R
```

70 GOTO 20

Zo kan men de Paddle gebruiken om snel het resultaat van een berekening te onderzoeken met verschillende getallen. De instructie PDL komt hier dus in de plaats van de instructie INPUT. Bij het programmeren in BASIC vervang ik vaak tijdelijk de INPUT-instructies door PDL-instructies om zo snel te kunnen onderzoeken of het programma voor verschillende combinaties van getallen goed werkt.

Een andere toepassing is het uitproberen van geluiden met:

```
10 SOUND 0,PDL(1)
20 SOUND 7,62
30 SOUND 8,PDL(3)16
40 GOTO 10
```

Zelf heb ik een zesvoudige paddle, netjes ingebouwd in een blikken sigarendoosje, waarmee ik zo in korte tijd duizenden geluidscombinaties kan uitproberen. Ik kan U verzekeren dat dit heel wat sneller gaat dan telkens nieuwe getallen intikken op het toetsenbord!

### **Andere toepassingen**

We hebben gezien dat we bij het uitvoeren van de instructie PDL(n) een getal terugkrijgen dat evenredig is met de impulslenkte, die zelf weer evenredig is met de ingestelde weerstand. Bij de maximum weerstand van 1 MOhm het resultaat 255 is. We hebben dus eigenlijk een Ohm-meter gebouwd:

```
10 R = PDL(1)*3.92 + 2
20 PRINT"De weerstand is";R; "kOhm."
30 GOTO 10
```

Laten we nu de potmeter weg en sluiten we een onbekende weerstand aan, dan vertelt de computer ons wel hoeveel kOhm die is. Pas zo nodig de factor 3.92 aan door te ijken met een precisieweer-stand. Het resultaat is overigens ook evenredig met de capaciteit van de condensator, dus even zo gemakkelijk maken we een capaciteitsmeter. Door via een schakelaar uit verschillende weerstanden te kiezen maken we meerdere meetbereiken. Een compleet schema van zo'n condensatormeet-kastje met bijbehorend BASIC-programma stond in MSX Nieuwsbrief nr.54-57 van Augustus 1986. Weer een ander idee is het vervangen van de regelbare weerstand door een temperatuurgevoelige weerstand (NTC). Het resultaat van PDL(n) is dan afhankelijk van de temperatuur en we hebben een digitale thermometer. Een beschrijving met programma stond in MSX Nieuwsbrief nr.47-49 van Juni 1986. Zo bestaan er ook lichtgevoelige weerstanden waarmee een lichtmeter is te maken. Nu gaan we nog even terug naar de regelbare weerstand, maar in plaats van een knop zetten we er een windvaantje op om per computer de windrichting te kunnen bepalen. Of een trekveer, een touwtje eraan met enkele slagen rond het asje van de potmeter en een krachtmeter is geboren. De handige knutselaar zal zo nog veel meer

leuke mogelijkheden weten te bedenken. Het timer-IC NES 5 5 heeft een ingang genaamd "Control" dat we missen bij het One-shot-IC zoals de HEF4538. Op deze ingang staat een spanning van ongeveer 3 Volt, maar als we die veranderen zien we dat de impuls lengte en dus ook het resultaat van PDL(n) wijzigt. Dit kan gebruikt worden om een eenvoudige digitale volt-meter te maken, zoals in de MSX Nieuwsbrief nr.73-77 van October 1986 beschreven is. Bijvoorbeeld te gebruiken voor het meten van de windsnelheid met een windmeter: bestaande uit een door de wind aangedreven gelijkstroommotorje dat als dynamo werkt, met de bekende halve bollen als windvangens. Op de open dag van de PTC dit jaar zag ik iemand die een fotoscanner had gebouwd door een lich-topnemer op de printkop van zijn printer te monteren. De helderheid van elk beeldpunt werd door de computer gemeten via de joystickpoort en de PDL-instructie!

### **Conversiesnelheid**

We hebben gezien dat er heel wat mogelijkheden zijn voor analoge input via de joystickpoorten. Maar er zijn uiteraard ook beperkingen. De belangrijkste is de lage conversiesnelheid. Deze is ook nog afhankelijk van het resultaat. De MSX computer kan zelf laten zien hoeveel tijd voor elke PDL-instructie nodig is. Daarvoor laten we die 1000 maal uitvoeren:

```
10 DEFINT A-Z:TIME = 0
20 FOR I=1 TO 1000
30 P = PDL(1)
40NEXT
50 PRINT TIME/50;"sec."
60 PRINT "P = ";P
```

Het resultaat is op mijn NMS8250:2,42 sec. voor P=0 en 5,68 sec. voor P=255. Zonder regel 30 duurt het tellen tot duizend 0,58 sec. Daaruit volgt dat het uitvoeren van elke PDL-instructie tussen 1,84 en 5,1 milliseconde duurt. Dat betekent dat maximaal een paar honderd conversies per seconde uitgevoerd kunnen worden. Dat kunnen er nog veel minder worden als we ook nog wat met het resultaat willen doen. Daarvoor heeft de computer tenslotte ook tijd nodig.

Voor de hierboven genoemde toepassingen is dat allemaal ruimschoots snel genoeg, voor andere kan het te langzaam zijn. De PDL-instructie voert het tellen en testen van de impuls op de ingang al in een zo kort mogelijk stukje machinetaal uit, dit kan dan ook niet versneld worden. De enige uitweg naar hogere snelheden is door van een hardware ADC gebruik te maken, zoals die van UNIFACE of die in de Muziek-module.

### **Nauwkeurigheid**

De Paddle hardware zet de (analoge) weerstand om in een eveneens analoge impuls lengte. De computer meet deze impuls lengte door elke 12 microseconde te onderzoeken of de impuls nog aanwezig is. Hier vindt dus de feitelijke digitalisering plaats. De werkelijke impuls lengte kan daarom ook tussen 0 en 12 microseconde langer

zijn van de gemeten lengte. We weten alleen dat de impuls opgehouden is voordat de volgende aanwezigheidstest plaatsvond. Om te zien wat dat betekent gaan we even terug naar de paddle als Ohm-meter en bekijken de invloed op het resultaat. De weerstand van max. 1 MOhm werd gemeten in maximum 255 stapjes van bijna 4 kOhm elk. De echte weerstand kan tot 4 kOhm groter zijn als de gemeten weerstand. Om een symmetrische foutenverdeling te krijgen tellen we gewoon de gemiddelde fout (2 kOhm) bij het resultaat op. De nauwkeurigheid wordt dan: +/- 2 kOhm. Aan het einde van de schaal, bij 1 MOhm is dit maar +/- 0,2%. Aan het begin van het bereik, bijv. bij 10 kOhm, is de absolute nauwkeurigheid nog steeds +/- 2 kOhm, maar procentueel is dat nu al +/- 20%.

Voor een goede nauwkeurigheid moeten we dan ook zoveel mogelijk boven in het bereik werken. Daarom is het ook zinvol om met meerdere bereiken te werken, zoals in het al eerder genoemde condensatormeetkastje. Maak elk bereik niet groter dan 10 keer. Zo kan 100 kOhm tot 1 MOhm gemeten worden met +/- 2 kOhm nauwkeurigheid. Door dan een 10 maal grotere condensator aan te sluiten kan van 10 tot 100 kOhm gemeten worden met een nauwkeurigheid van +/- 200 Ohm. Willen we een grotere nauwkeurigheid, dan laat de PDL-instructie ons in de steek.

## **Machinetaal**

We zagen al dat we met een eigen stukje machinetaal de Analoog naar Digitaal conversie niet sneller kunnen maken, maar de nauwkeurigheid vergroten gaat wel. De nauwkeurigheid die we met de instructie PDL kunnen halen hangt immers samen met het gebruik van de 8-bits teller (0..255) door de computer. Laten we maar eens kijken hoe BASIC dit uitvoert. Register C is de teller en register B geeft het te controleren bit en daarmee de te testen ingang aan. De interrupt moet uiteraard uitgeschakeld.

```

DI
LD C,00
LUS IN A,A2
AND B
JRZ,EXIT
INC C
JP NZ,LUS
DEC C
EXIT EI
RET

```

Als we in plaats daarvan registerpaar DE als teller gebruiken kan de nauwkeurigheid sterk opgevoerd worden.

```

DI
LD DE,0000
LUS INA,A2
AND B
JR,.EXIT
INC DE

```

```
LD A,D
OR C
JP NZ,LUS
DEC DE
EXIT EI
RET
```

Deze lus heeft echter 53 T-states oftewel bijna 15 microseconde nodig. Het uitvoeren met maximaal 65536 keer kost bijna een seconde. De extra nauwkeurigheid gaat dan ook wel ten koste van de snelheid. Als tussenoplossing kan men bijv. tot 1000 of tot 5000 laten tellen. De eenvoudigste manier is om registerpaar DE een beginwaarde te geven, die later weer van het resultaat wordt afgetrokken.