

Labo datacommunication les 2 : I²C

Doelstelling

Aansluiten van een pull-up of een pull-down ingang.

Leren omgaan met een protocol.

De werking van het I²C protocol kunnen uitleggen en toepassen.

Leren werken met digitale in- en uitgangen.

Digitale ingang

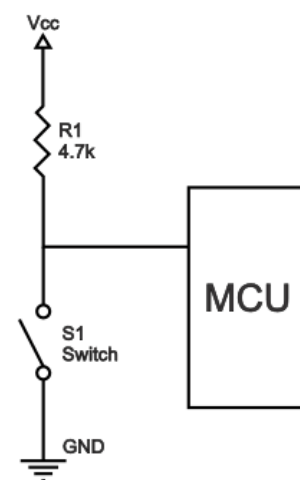
Een digitale ingang vraagt maar een zeer kleine stroom. Vandaar dat we een knop rechtstreeks met een ingang kunnen verbinden zonder gebruik te maken van een extra weerstand. De ingangen zijn ook zeer gevoelig, een kleine stroom kan reeds gedetecteerd worden.

Als we niks aansluiten aan zo'n ingang is die zeer gevoelig voor storingen. Vandaar dat het interessant kan zijn om die ingang via een pulldown resistor met de massa te verbinden ofwel via een pullup resistor naar de +5V.

Pull-up

Er zijn echter 4K7 pullup resistors ingebouwd op het USB DAQ board.

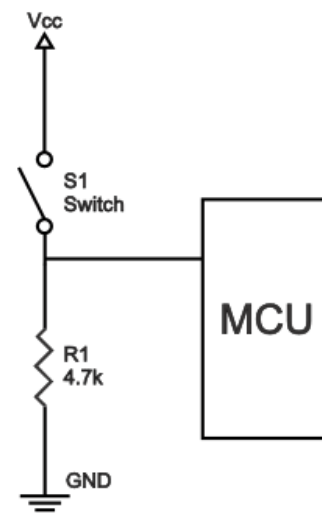
B.v. als we een knop aansluiten aan een pull-up ingang, moeten we het andere uiteinde van de knop verbinden met de massa. Als de knop open is, dan zullen we een HIGH inlezen aan de ingang. Drukken we op de knop, dan hebben we een stroomkring van onze pullup ingang(+5V) naar de massa en zullen we een LOW inlezen.



Pull-up resistor circuit

Pull-down

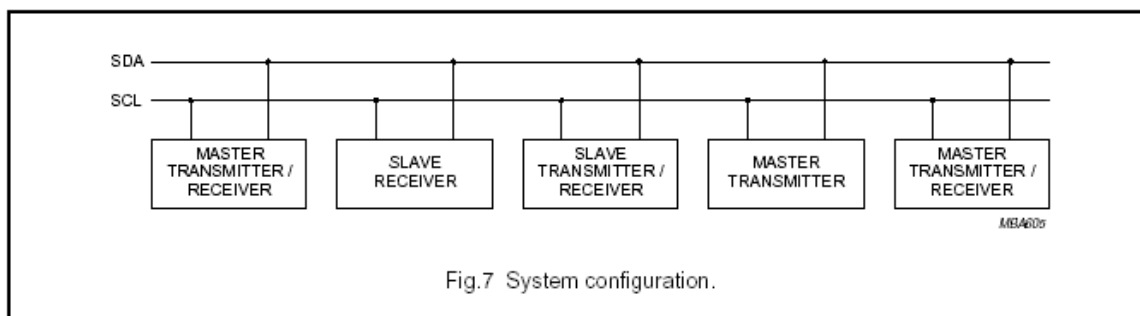
B.v. als we een knop aansluiten aan een pull-down ingang, moeten we het andere uiteinde van de knop verbinden met de spanning. In ons geval de logic 1 of de 5V DC. Als de knop open is, dan zullen we een LOW inlezen aan de ingang. Drukken we op de knop, dan hebben we een stroomkring van de +5V naar de massa en zullen we een HIGH inlezen.



Pull-down resistor

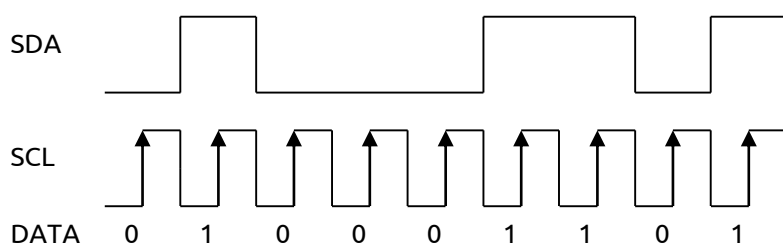
Inleiding

De I²C-bus (inter-IC-bus) is een tweedraads dataverbinding tussen één en meerdere processoren (masters) en speciale periferiecomponenten (slaves). Al deze componenten zijn op dezelfde bus aangesloten en hebben elk hun eigen unieke adres. Adressen en data worden via dezelfde lijnen verzonden. De I²C- bus maakt een uiterst eenvoudige verbinding mogelijk tussen vele IC's waarbij uitbreidingen achteraf probleemloos kunnen worden gerealiseerd.



De I²C-bus maakt gebruik van de seriële datalijn SDA en het kloksignaal SCL. Data en adressen worden net als bij schuifregisters, samen met een kloksignaal verzonden. Beide lijnen kunnen in beide richtingen worden gebruikt.

Telkens als de kloklijn wijzigt van laag naar hoog wordt er een bit doorgestuurd.



Wij gaan een 7 segment display aansturen via I²C. Een 7 segment display bestaat in feite uit 8 LED's. Zo'n display zouden we dus ook rechtstreeks kunnen aansluiten aan poort D van onze microchip m.a.w. parallel op onze 8 LED's.

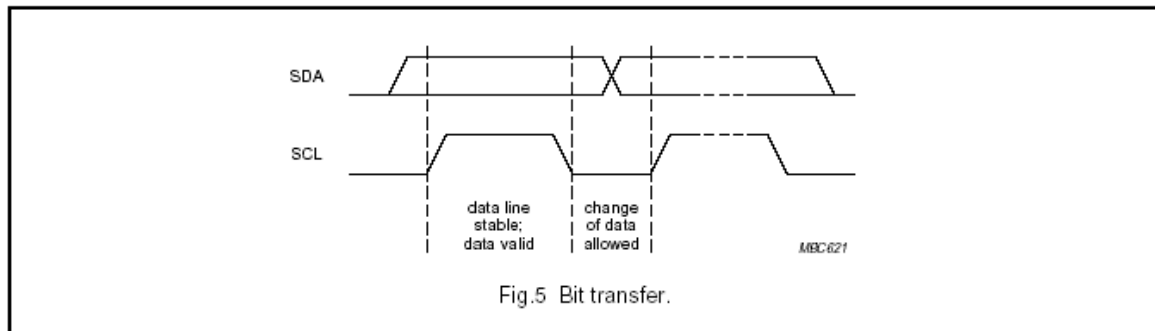
Op de usb data-acquisitiekaart hebben we hardware matig de volgende verbindingen gemaakt :

SDA	poort D databit 0 (waarde 0 of 1)
SCL	poort D databit 1 (waarde 0 of 2)

Als we bijvoorbeeld een databit 1 willen sturen hebben we de volgende code nodig :

```
SDA=>I      SCL=>0      writeDigitalOutPortD(1)
SDA=>I      SCL=>I      writeDigitalOutPortD(3)
                        Wait(1)
SDA=>I      SCL=>0      writeDigitalOutPortD(1)
```

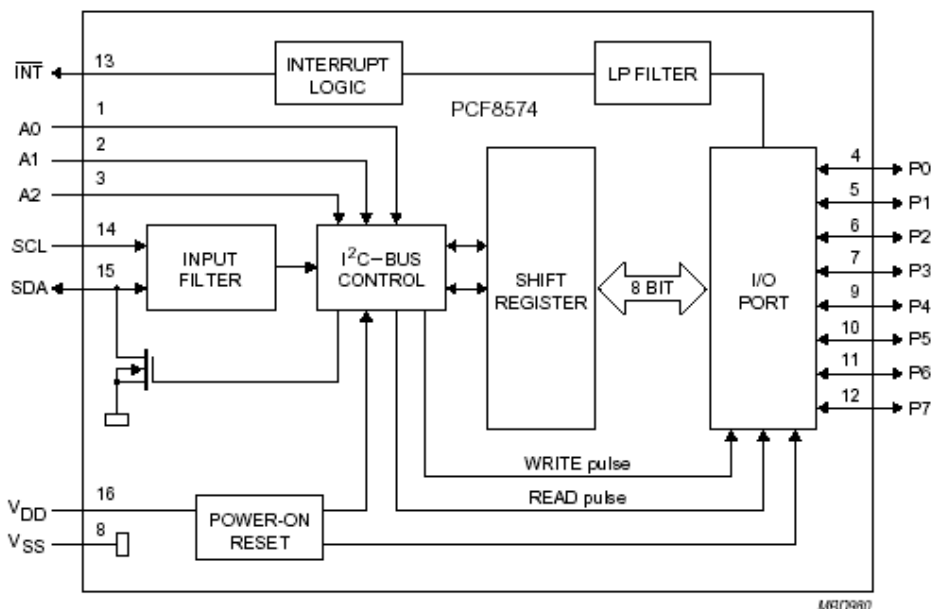
Zie ook onderstaande figuur



Component PCF8574 (I²C IC)

Op onderstaand schema zien we duidelijk de 8 digitale datalijnen, die op het bordje verbonden zijn met het 7 segment display.

We hebben nog de voedingslijnen, de klok en de datalijn. We kunnen ook nog de 3 adreslijnen opmerken (A0,A1 en A3).



Vooraleer we data kunnen sturen naar een bepaalde slave moeten we eerst het adres van een bepaalde slave doorsturen. Als we meerdere 7-segment displays willen aansturen en dus meerdere slave 's willen gebruiken hebben we dus een probleem. Vandaar dat we 3 bits van de PCF8574 zelf kunnen instellen. We kunnen dus 8 identieke PCF8574 ic's gebruiken = > 64 datalijnen.

Het adres bestaat dus uit een vast en een veranderlijk gedeelte.

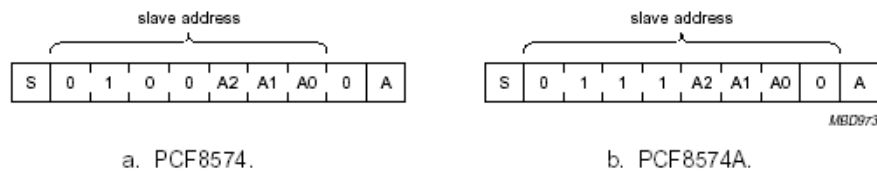


Fig.10 PCF8574 and PCF8574A slave addresses.

Het adres van onze IC kan dus de volgende waarden aannemen :

64,66,68,...,78

De waarden van A0,A1 en A2 kunnen we instellen met de DIP switches.

Als de drie dipswitchen NIET op on staan krijgt onze slave IC een nul voor A0,A1,A2

Het I²C-busprotocol

Het I²C-busprotocol kent een reeks nauwkeurig gedefinieerde toestanden die het mogelijk maken dat elke busdeelnemer begin en einde van een bericht plus de adressering daarvan kan herkennen.

- Rusttoestand : SDA en SCL zijn beide hoog en niet actief.
- Startconditie: SDA wordt door de master laag gemaakt, terwijl SCL hoog blijft.
- Stopconditie: SDA gaat van laag naar hoog, terwijl SCL hoog blijft.

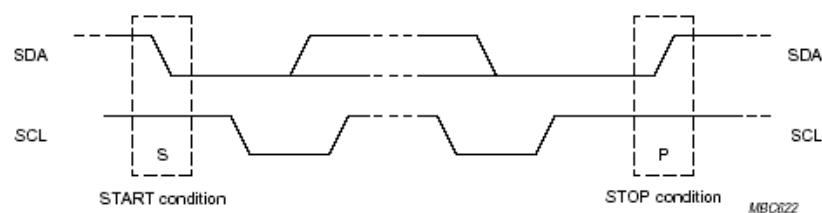
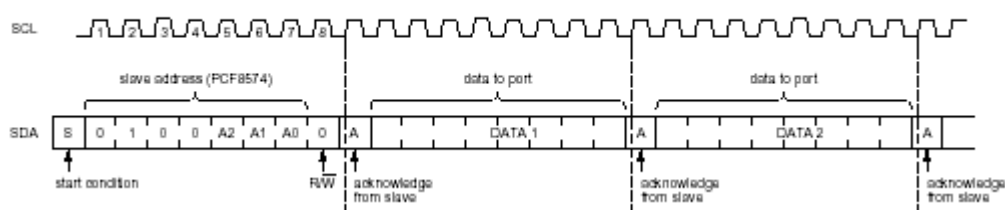
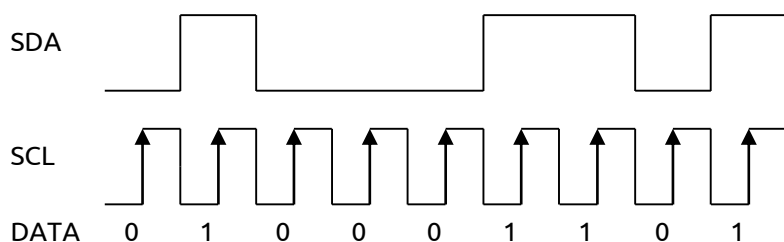


Fig.6 Definition of start and stop conditions.

- Data-overdracht: de betreffende zender zet 8 databits op datalijn SDA, die door de klokpulsen van de master op kloklijn SCL verder worden geschoven. Overdracht begint steeds met de meest beduidende bit.

- Bevestiging (Acknowledge): de ontvanger bevestigt de ontvangst van een byte door SDA laag te maken tot de master de negende klokpuls op SCL heeft gezet



Merk op dat we met databit 0 van het adres kunnen bepalen of we willen lezen of schrijven naar de IC.

Zolang dat we geen stopconditie gebruiken kunnen we blijven databits sturen zonder dat we opnieuw het adres moeten doorsturen.

Het programma kan er dus als volgt uitzien:

Startconditie

Adres doorsturen (8 bits doorklokken)

Ack

Data doorsturen (8 bits doorklokken)

Ack

Stopconditie

Hoe acknowledge inlezen?

Het PCF8574 kan de SDA lijn naar 0 brengen, als we dus de ack willen inlezen moeten we dus eerst de SDA lijn hoog zetten.

Bevestiging (Acknowledge): de ontvanger bevestigt de ontvangst van een byte door SDA laag te maken van zodra de master de negende klokpuls op SCL heeft gezet

De ack is aangesloten op de zesde bit van poort B. (RB5)

Als de status van bit 5 hoog staat heeft de ontvanger de data goed ontvangen. Het lage signaal dat van de I²C wordt immers geïnverteerd door de inverse smittrigger.

Stappen om de acknowledge controleren

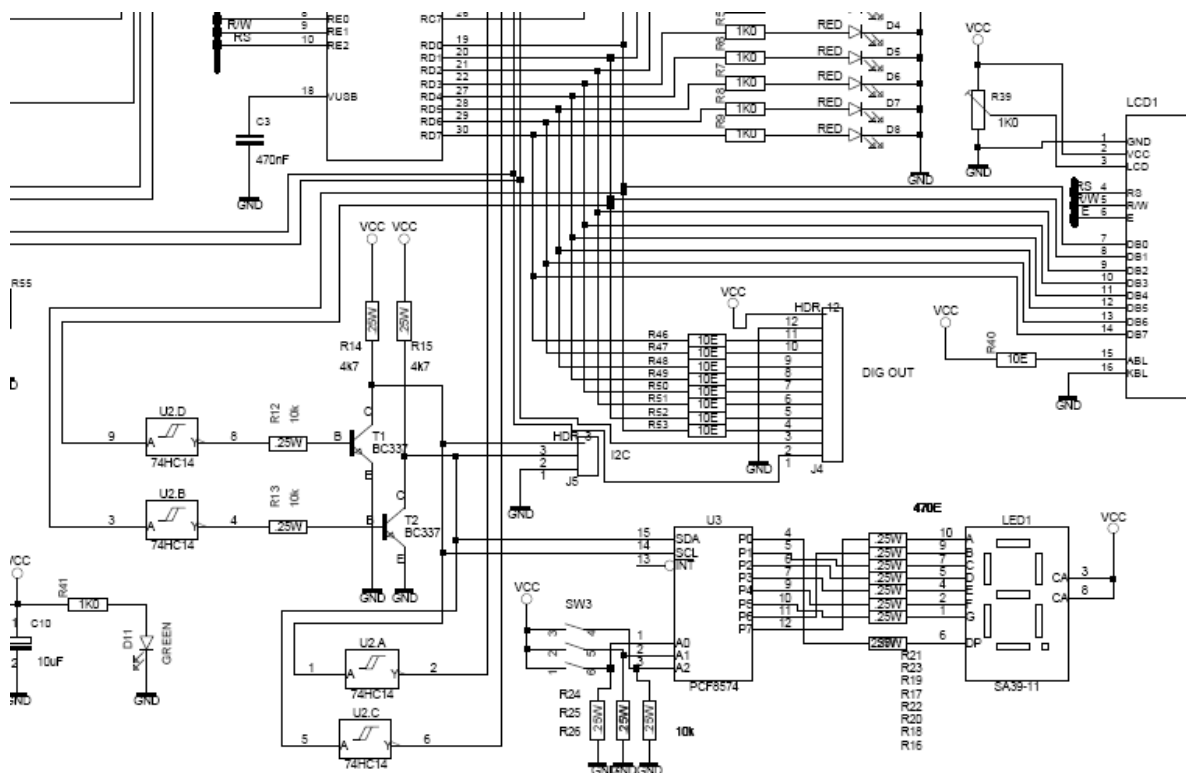
1. SDA lijn hoog brengen
2. klokpuls geven
3. efkes wachten
4. Bit 5 van poort B.
Krijgen we een nul op bit 5 is ack niet ok.
Bij een één op bit 5 is ack ok.
5. Klok terug naar omlaag brengen

Hardware

De SDA (datalijn) en de eerste LED zijn parallel aangesloten op bit 0 van poort D. Als we dus bit 0 van poort D op 1 plaatsen, krijgt de SDA lijn een hoog niveau en gaat de eerste LED branden.

De SCL (clocklijn) en de tweede LED zijn parallel aangesloten op bit 1 van poort D. Bij een hoog niveau van de kloklijn zal dus de tweede LED branden.

Het 7 segment display is aangesloten op een spanning van +5V. Als we een segment willen laten oplichten moeten we dus een 0 sturen naar de uitgangen van onze I²C ic. Op die manier krijgen we een spanningsverschil en kan het segment oplichten. Sturen we een 1 dan verschijnt er +5V op onze uitgang. We hebben dus geen spanningsverschil en dus zal het segment ook niet oplichten. Dus als we 8 nullen doorklokken naar onze I²C ic zullen alle segmenten oplichten. Zie onderstaand schema.



opdracht I²C (lesweek 2)

Inleiding

Deze week wordt er nieuw project opgestart. Geef het project de volgende naam : Groepsnr-
loginnaam1-I2C-01.

Stappenplan

- Voeg twee buttons en tekstvak toe aan de window De startconditie en het doorsturen van het adres wordt aan de "init" knop gekoppeld. De data doorsturen koppelen we aan de button "zend". We versturen de data van het tekstvak.
- Programmeer de startconditie
- Schrijf de routines zendEen en zendNul
- We plaatsen de drie dipswitches op off. Het adres van de ic heeft de waarde 64. Die waarde moeten we doorsturen naar de IC. We moeten dus een 0100 0000 sturen. Die 8 bits gaan we dus na elkaar gaan doorklokken. Schrijf gewoon de verschillende stappen onder elkaar. Straks gaan we de code optimaliseren.
- Simuleer de ack door gewoon een 0 of een 1 door te klokken.
- De IC is klaar om 8 bits data te ontvangen. Je kan b.v. 8 nullen sturen zodat alle segmenten van het display zullen oplichten. Schrijf gewoon alle code onder elkaar.
- Programmeer de stopconditie.
- Zoek uit welke waarden je moet sturen om alle cijfers van 0=> 9 aan de praat te krijgen.
- Schrijf een routine om de ack effectief in te lezen. Als we b.v. een foutief adres sturen moeten we een foutboodschap krijgen.
- 1 routine maken die het adres en verschillende data kan doorsturen. Die routine heeft als argument 1 byte die geanalyseerd wordt.

```
byte bitwaarde = 128;
```

```
For n = 1 To 8 (int i = 1; i <= 8; i++){
```

```
1) Vergelijk de byteWaarde met de bitwaarde
   Afhankelijk van het resultaat roep je de routine
   stuurEen of stuurNul aan.
```

```
2) pas de bitwaarde aan
```

```
}
```

- Sluit de twee extern knoppen aan, sluit de connector aan het USB Daq board. Kijk op welke bits de knoppen zijn aangesloten. Zijn het pull-up of pull down ingangen?
- Via een druk op de knop kunnen we een automatische teller laten lopen.
- We hebben ook een tekstvak op de GUI waar we een getal van 0 tem 9 kunnen ingeven
- Door gebruik te maken van de ACK kunnen we het programma het ingestelde adres van de IC laten zoeken.
- Maak gebruik van de zotte knop om de cijfers op het display naar onder of naar boven aan te passen.
- Door het drukken op de zotte knop kan je de digit (puntje rechtsonder) aan of afzetten.

Opmerking : er moet gewerkt worden met bitoperaties om de drukknop en de ack in te lezen.