

Sensors aansluiten op de Raspberry Pi.

Digitale temperatuur sensor + weerstand



DS18B20+Temp.sensor



DS18B20 Temp.sensor WD

stand (LDR) - 200K



LDR



IR sensor set



PIR sensor

Sensors aansluiten op de Raspberry Pi.

De mensheid probeert al eeuwenlang de natuurkundige verschijnselen om hem heen te meten.

Dat gebeurde heel lang met analoge meetinstrumenten. Denk daarbij aan bijvoorbeeld de thermometer en barometer.

In het digitale tijdperk meten we met sensoren die het resultaat in digitale vorm presenteren.

Voor de Raspberry Pi zijn er veel verschillende typen sensoren beschikbaar.

Een greep uit het assortiment: Infrarood, temperatuur, spanning stroom, druk, magnetische veldsterkte, rotatie enz.

Sensors aansluiten op de Raspberry Pi.

We kunnen de sensors die we bij de Raspberry Pi gebruiken verdelen in twee categorieën.

Sensors die een min of meer nauwkeurige equivalente waarde in digitale vorm van het analoge verschijnsel geven en sensors die een aan of uit toestand weergeven.

In het eerste geval kan dat bijvoorbeeld een temperatuur sensor zijn, in het tweede geval een beweging sensor.

Sensoren die een aan of uit toestand weergeven moeten een ingebouwde hysteresis hebben zodat het schakelen bedrijfszeker is.

Sensors aansluiten op de Raspberry Pi.

Temperatuur sensors komen we in twee uitvoeringen tegen.
Met een I2C interface en een One Wire interface

De MCP9808 is precisie temperatuur sensor met een I2C interface.

De DS18B20 is een temperatuur sensor met een one wire interface.

In deze workshop wordt gebruik gemaakt van de DS18B20.

Sensors aansluiten op de Raspberry Pi.

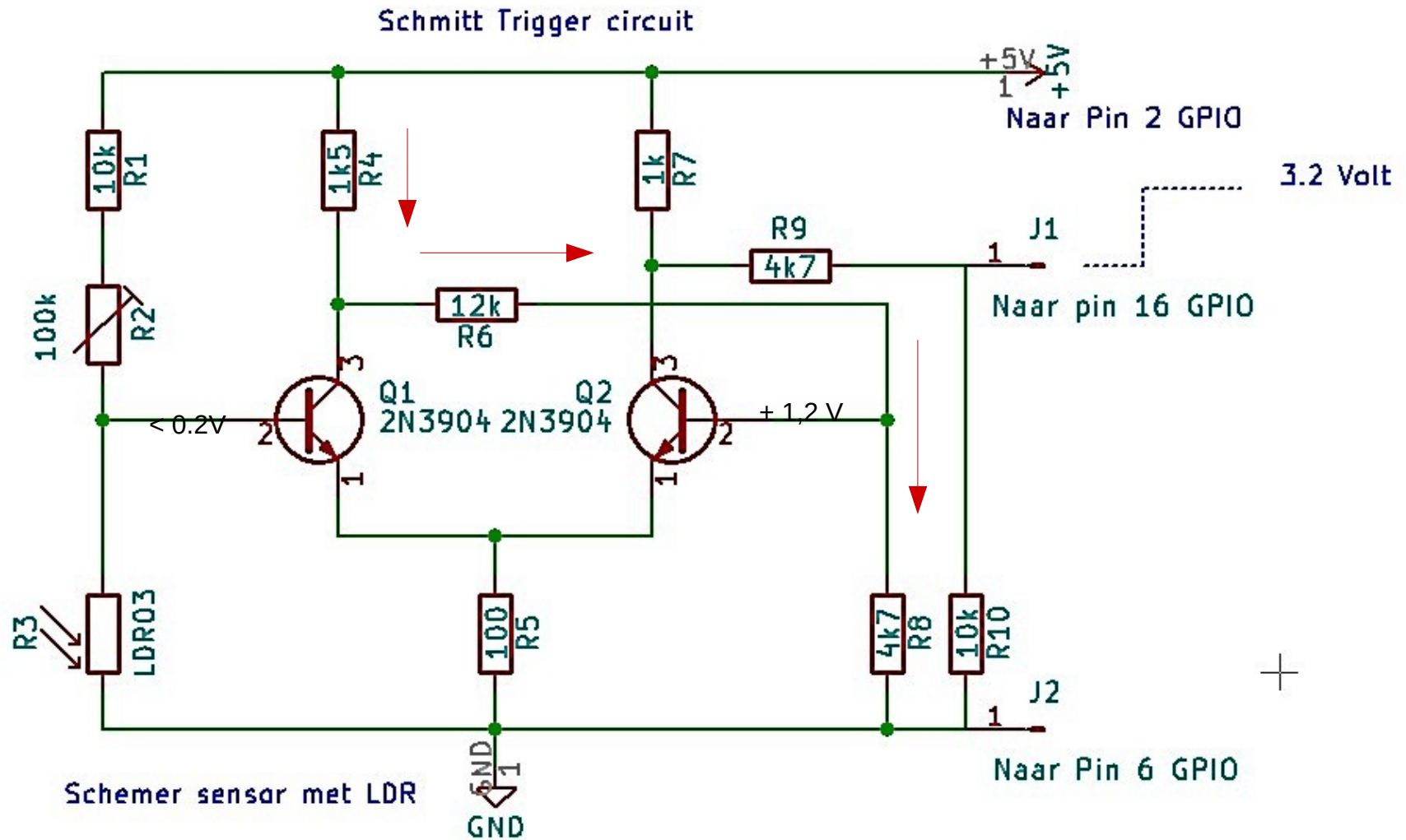
Vertegenwoordiger van de tweede categorie de aan/uit sensors is de PIR sensor, **P**hotovoltai**c I**nfr**a R**ed.

Ze worden gebruikt als bewegingssensor of als lichtsluis.

Verder hebben we sensors die een analoge waarde als uitgang hebben zoals een LDR, **L**ight **D**ependent **R**esistor. Bij een LDR neemt de weerstand af bij een toename van licht. Met een Schmitt Trigger circuit maken we er een aan/uit sensor van. Het is dan een niveau schakelaar.

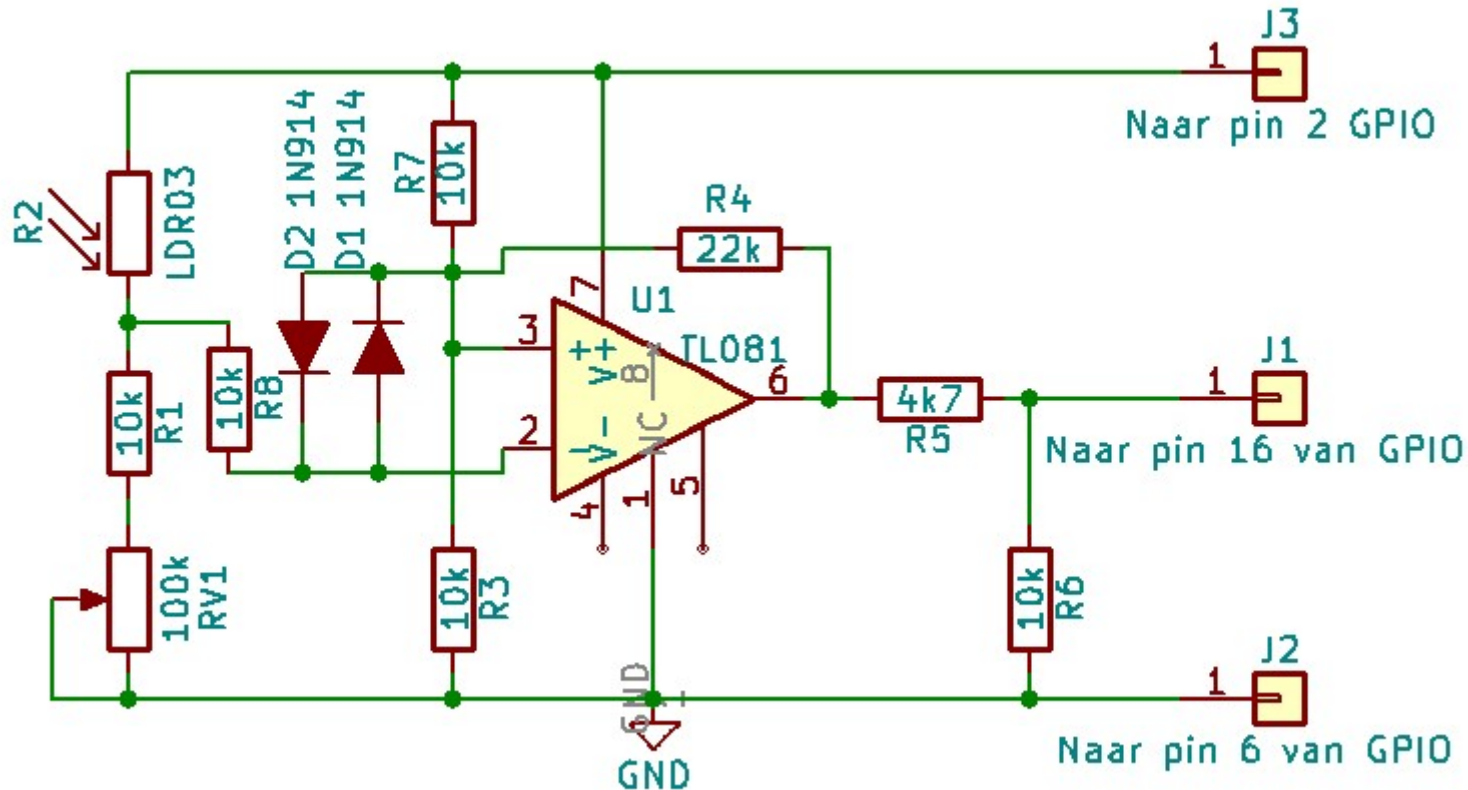
Een Schmitt Trigger is een elektronische schakeling waarvan het uitgangssignaal omschakelt bij een instelbare analoge spanning op de ingang.

Sensors aansluiten op de Raspberry Pi.



Schmitt Trigger circuit met LDR en instelbaar schemer niveau.

Sensors aansluiten op de Raspberry Pi.



Een Schmitt Trigger circuit uitgevoerd met een opamp TL 081

Sensors aansluiten op de Raspberry Pi.

We kunnen een aan/uit sensor op drie manieren gebruiken
In ons programma:

- Wachten tot er wat gebeurt, het programma is dan verder inactief. Nadeel de RPi kan verder geen taken uitvoeren.
- Polling, om de zoveel tijd kijken of de sensor actief is geworden. Nadeel kans op missen van het cruciale moment.
- Op interrupt basis, het lopende programma onmiddellijk na het ontvangen het sensor signaal onderbreken en de taak uitvoeren verbonden aan de interrupt.
- Het volgende programma werkt met interrupts.

Sensors aansluiten op de Raspberry Pi.

```
#Bewegingssensor
```

```
import RPi.GPIO as GPIO
GPIO.setmode (GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(12, GPIO.IN, pull_up_down =GPIO.PUD_DOWN)
GPIO.setup(16, GPIO.OUT)
GPIO.setup(18, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

def pir_callback(channel): # callback functie die de LED aan zet
    GPIO.output(16,1)

def pir_callback1(channel):
    GPIO.output(16,0)

GPIO.add_event_detect(12, GPIO.RISING, callback = pir_callback)

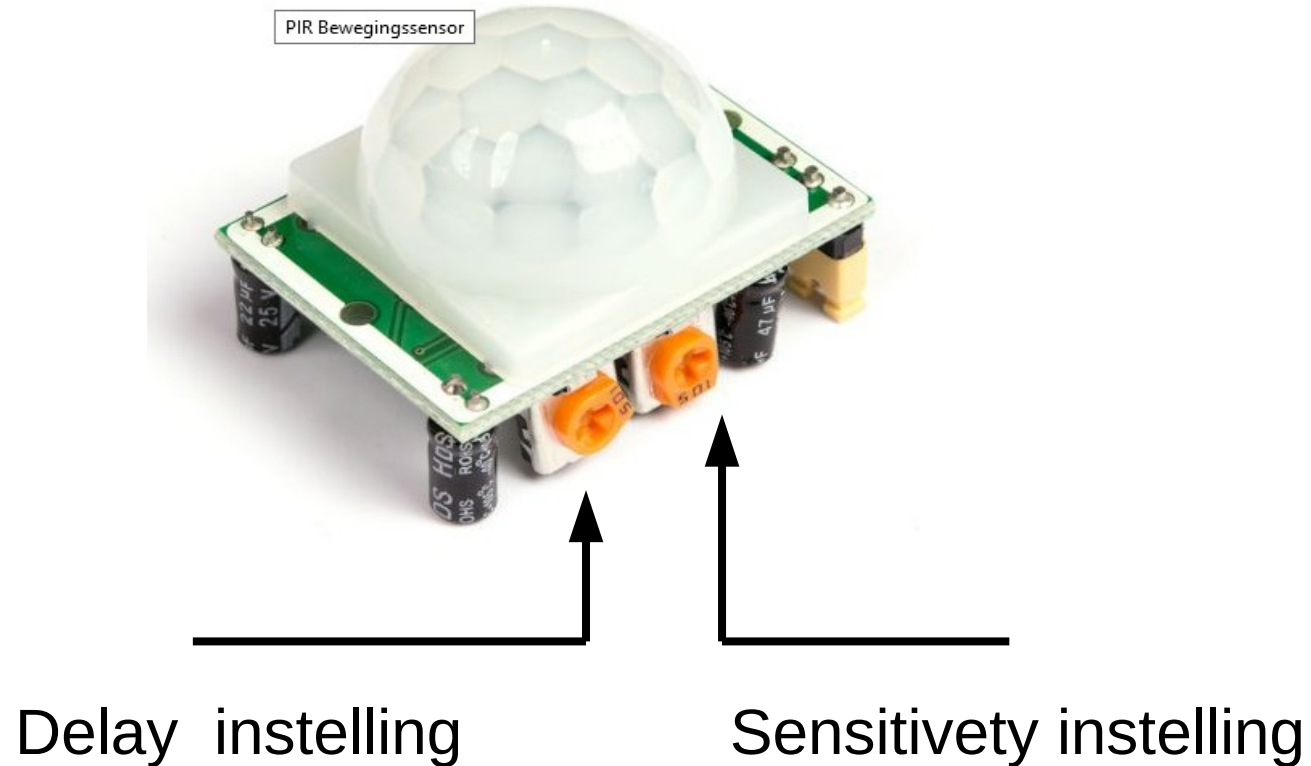
GPIO.add_event_detect(18, GPIO.FALLING, callback = pir_callback1)
```

Sensors aansluiten op de Raspberry Pi.



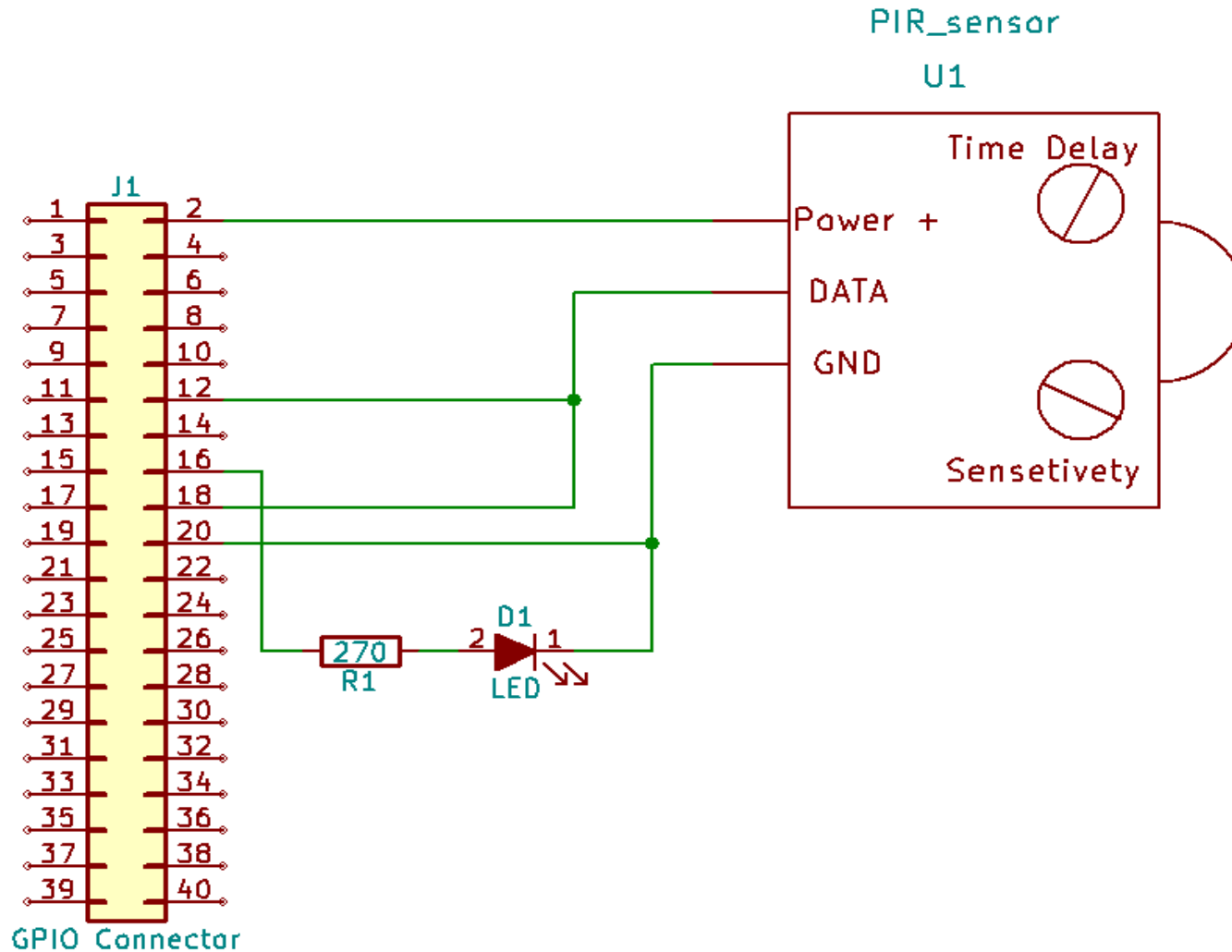
De PIR bewegingssensor produceert een stijgende flank als een IR stralend bewegend object wordt gedetecteerd.

Sensors aansluiten op de Raspberry Pi.



De sensitivity instelling bepaalt de detectie range 3 tot 7 meter.
De delay instelling bepaalt hoe lang na detectie de uitgang hoog blijft, minimaal 3 tot maximaal 300 seconden.
De sensor heeft een ingebouwde Schmitt trigger schakeling.

Sensors aansluiten op de Raspberry Pi.



Sensors aansluiten op de Raspberry Pi.

```
#lichtsensor
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(12,GPIO.IN) # pin 12 als input voor de Schmitt trigger
GPIO.setup(16,GPIO.OUT) #pin 16 als output voor de actie uitvoering
GPIO.setup (18,GPIO.IN) # pin 18 als input voor de Schmitt trigger

def ldr_callback1(channel): # functie wordt uitgevoerd als een positief gaand
    GPIO.output(16,1)      # interrupt signaal wordt
#ontvangen
    print("LDR rising edge detected")

def ldr_callback2(channel): #functie wordt uitgevoerd als een negatief gaand
    GPIO.output(16, 0)     # interrupt signaal wordt ontvangen
    print ("LDR falling edge detected")

GPIO.add_event_detect(12, GPIO.RISING, callback = ldr_callback1)
GPIO.add_event_detect(18, GPIO.FALLING, callback = ldr_callback2)
```

Sensors aansluiten op de Raspberry Pi.

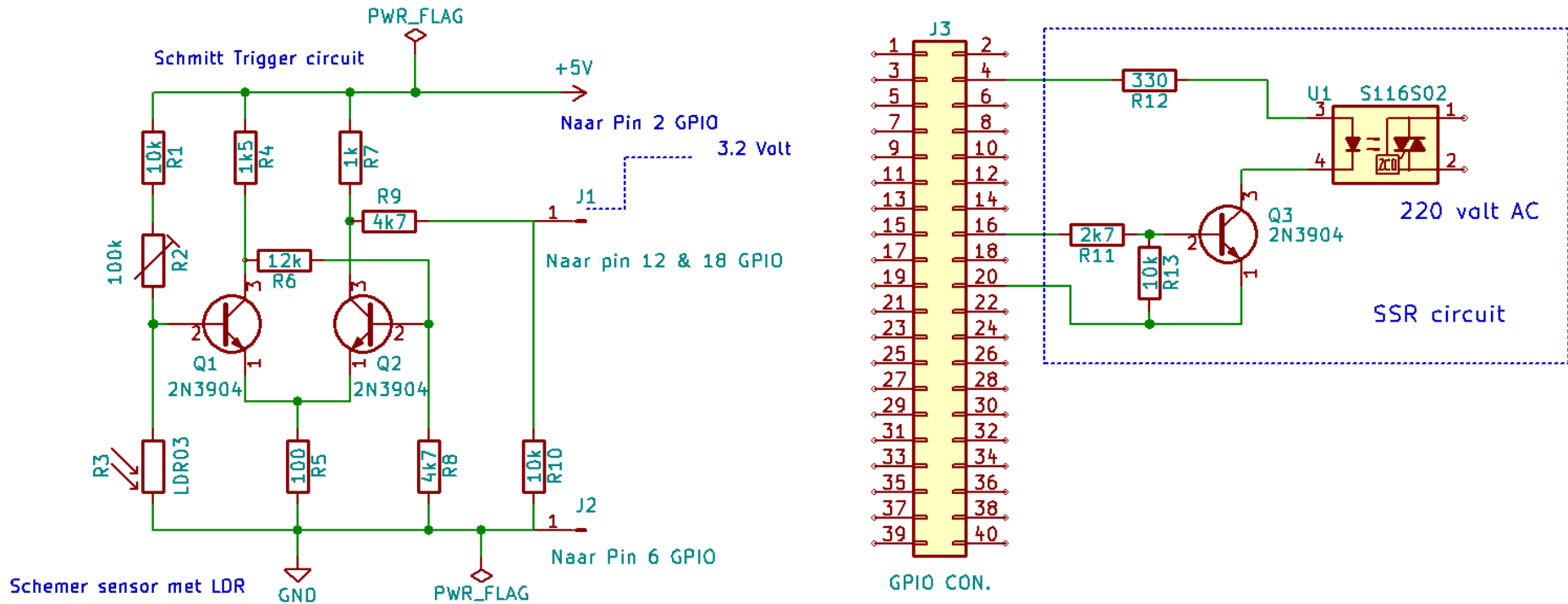
In het programma lichtsensoren configureren we de pinnen 12 en 18 als ingangen en pin 16 van de GPIO als uitgang.

Het uitgangssignaal van de Schmitt trigger wordt op de pinnen 12 en 18 aangesloten.

Pin 12 reageert op een stijgende flank en pin 18 op de neergaande flank van het uitgangssignaal .

Pin 16 is de uitgang en stuurt bijvoorbeeld een SSR aan.

Sensors aansluiten op de Raspberry Pi.



Schema van het schemer sensor circuit.

Sensors aansluiten op de Raspberry Pi.

Temperatuur meting met de DS18B20 sensor.

Om met de DS18B20 metingen te kunnen doen moeten we vooraf een aantal instellingen doen.

Open de terminal en type *sudo raspi-config*

Het configuratie venster opent. Selecteer optie 5.

Selecteer vervolgens optie P7, 1-Wire. Activeer het 1- wire interface.

Klik vervolgens op OK en sluit de terminal.

Sensors aansluiten op de Raspberry Pi.

Temp drivers installeren.

Open de terminal en type: *sudo nano /etc/modules*
Voeg onderaan het nu geopende bestand de volgende regels toe:

```
w1-gpio pullup=1  
w1-therm
```

Save het bestand, sluit af en herstart de RPi.

Sensors aansluiten op de Raspberry Pi.

Test de aanwezigheid van de DS18B20 sensor.

Sluit pin 1 van de sensor aan op pin 6 van de GPIO.

Sluit pin 2 van de sensor aan op pin 7 van de GPIO.

Sluit pin 3 van de sensor aan op pin 1 van de GPIO.

Plaats een weerstand van 4,7 kOhm tussen pin 2 en 3 van de sensor.

Open de terminal en type:

```
cd /sys/bus/w1/devices/w1_bus_master1/
```

Type: *Enter* en vervolgens *ls*

Je krijgt nu een overzicht met o.a. de unieke code van de sensor. Deze code ben je in je programma nodig.

Sensors aansluiten op de Raspberry Pi.

Inhoud van de map: `/w1_bus_master1/`

```
pi@barnhouse:~ $ cd /sys/bus/w1/devices/w1_bus_master1/
pi@barnhouse:/sys/bus/w1/devices/w1_bus_master1 $
ls
28-0317243f25ff w1_master_attempts      w1_master_search
driver          w1_master_max_slave_count  w1_master_slave_count
power          w1_master_name             w1_master_slaves
subsystem      w1_master_pointer          w1_master_timeout
uevent         w1_master_pullup           w1_master_timeout_us
w1_master_add  w1_master_remove
pi@barnhouse:/sys/bus/w1/devices/w1_bus_master1 $
```

`28-0317243f25ff` is de unieke code van de gebruikte sensor, noteer deze voor later gebruik.

Sensors aansluiten op de Raspberry Pi.

```
# Temperatuur sensor DS18B20 uitlezen.

tempfile = open("/sys/bus/w1/devices/28-0317243f25ff/w1_slave")
# de inhoud van de map w1_slave als tekstbestand toegekend
# de variabele tempfile
text = tempfile.read() #de inhoud van tempfile toegekend aan text
tempfile.close() # de tempfile wordt gesloten
secondline = text.split("\n") [1] # de twee regels met data worden
#gesplitst op het newline karakter en de 2e regel wordt in de variable
#secondline geplaatst.
temperatuurdata =secondline.split(" ") [9] #de data in de tweede
#lijn wordt gesplitst op spaties en het tiende woord in de rij wordt
# toegekend aan de variabele temperatuurdata.
temperatuur = float(temperatuurdata [2:])
temperatuur = temperatuur/1000
print ( "%.1f" % temperatuur)
```

Sensors aansluiten op de Raspberry Pi.

De inhoud van het `/w1_slave` bestand ziet er a.h.v. uit:

```
62 01 4b 46 7f ff 0c 10 92 : crc=92 YES  
62 01 4b 46 7f ff 0c 10 92 t=22125
```

De 2^e regel gebruiken we bij het weergeven van de temperatuur. De regel wordt gesplitst op de spaties en het 10^e element wordt in de variabele temperatuur geplaatst.

Sensors aansluiten op de Raspberry Pi.

Python script om temperatuur te meten met een DS18B20 sensor.

```
#ds18b20_temp_02a
while True:

    tempfile = open("/sys/bus/w1/devices/28-0317243f25ff/w1_slave")
    text = tempfile.read()
    tempfile.close()
    secondline = text.split("\n") [1]
    temperatuurdata =secondline.split(" ") [9]
    temperatuur = float(temperatuurdata [2:])
    temperatuur = temperatuur/1000
    print ( "%.1f" % temperatuur)
```

Gebruik de unieke code van de DS18B20
Stop de meting met Enter of Cntrl + C.

Sensors aansluiten op de Raspberry Pi.

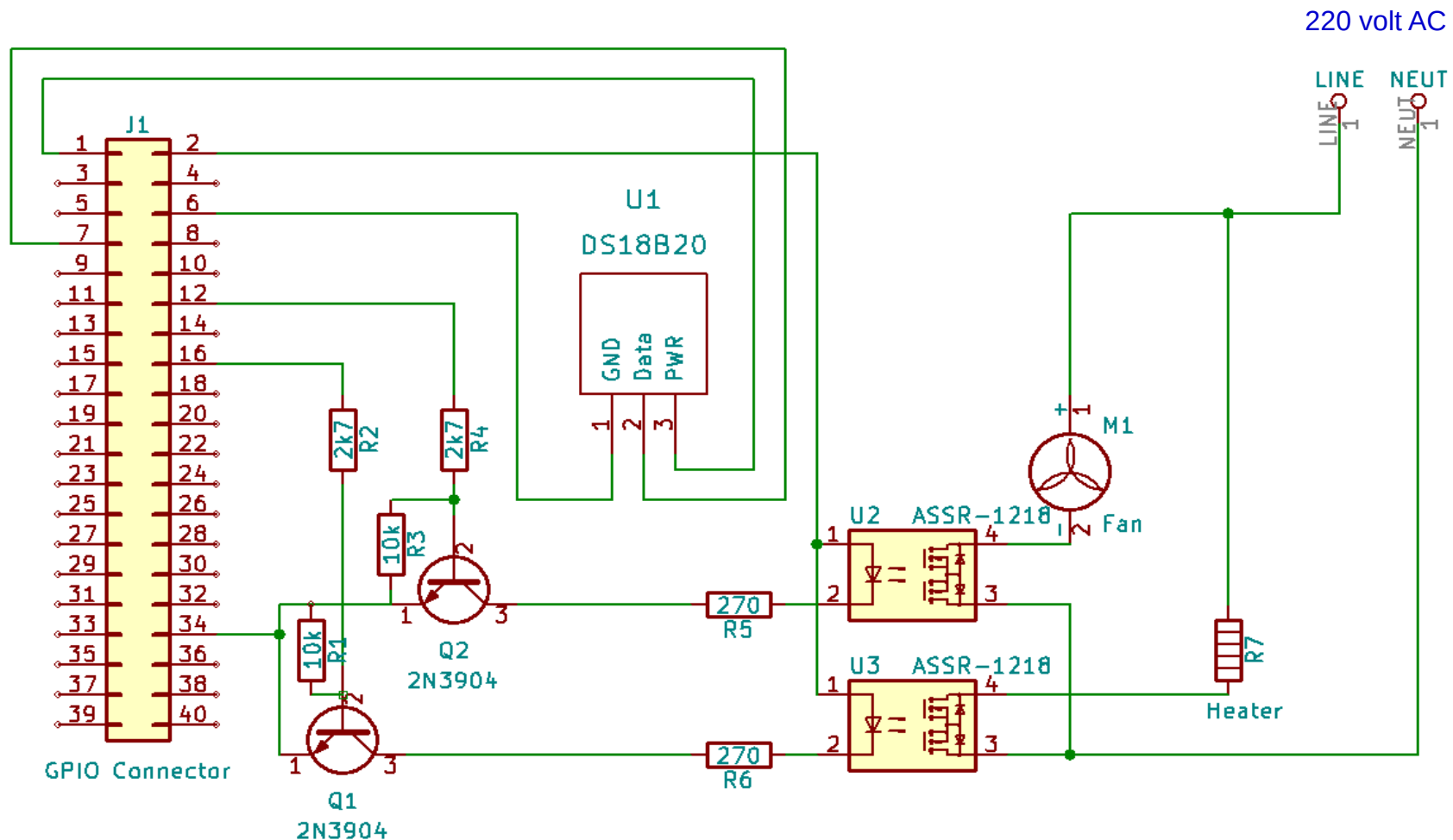
```
#ds18b20_temp_02b
a = input("Geef een streef temperatuur in: ")
a = float(a)
while True:
    try:
        tempfile = open("/sys/bus/w1/devices/28-0317243f25ff/w1_slave")
        text = tempfile.read()
        tempfile.close()
        secondline = text.split("\n") [1]
        temperatuurdata =secondline.split(" ") [9]
        temperatuur = float(temperatuurdata [2:])
        temperatuur = temperatuur/1000
        print ( "%.1f" % temperatuur)
        if temperatuur < a:
            print("Schakel de verwarming in!")

        else:
            print("Schakel de koeling in!")

    except KeyboardInterrupt: #Gebruik Control +C
        print("Meting gestopt")
```

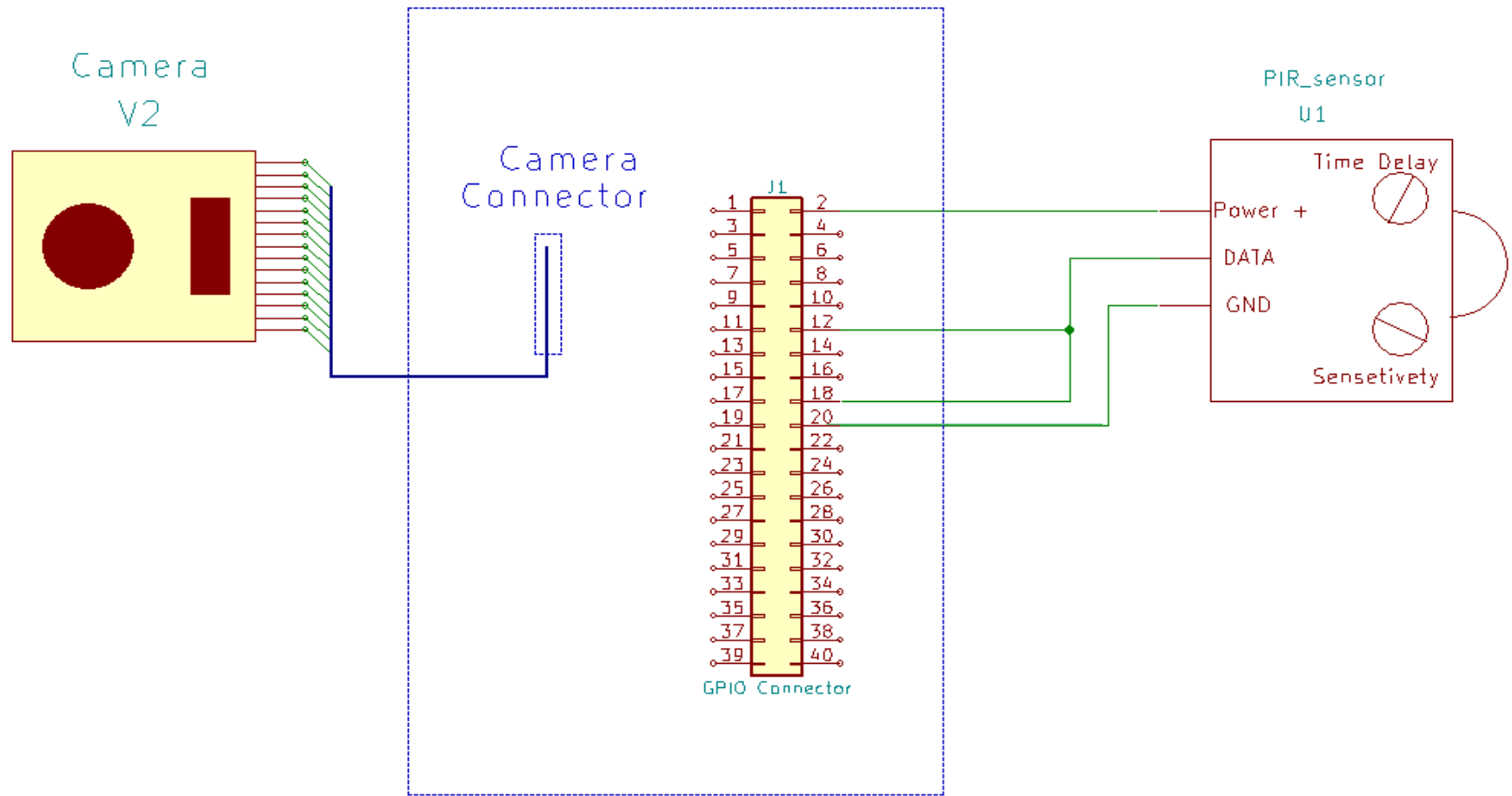
Sensors aansluiten op de Raspberry Pi.

Temperatuur regeling met DS18B20 sensor



Sensors aansluiten op de Raspberry Pi.

f



RPi Camera met bewegingssensor.

Sensors aansluiten op de Raspberry Pi.

Beweging geactiveerde camerabeelden.

```
#camera_03
import RPi.GPIO as GPIO
import time
import picamera
camera = picamera.PiCamera()
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(18, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

def pir_callback1(channel):
    camera.start_preview()
    #time.sleep(10)
    #camera.stop_preview()

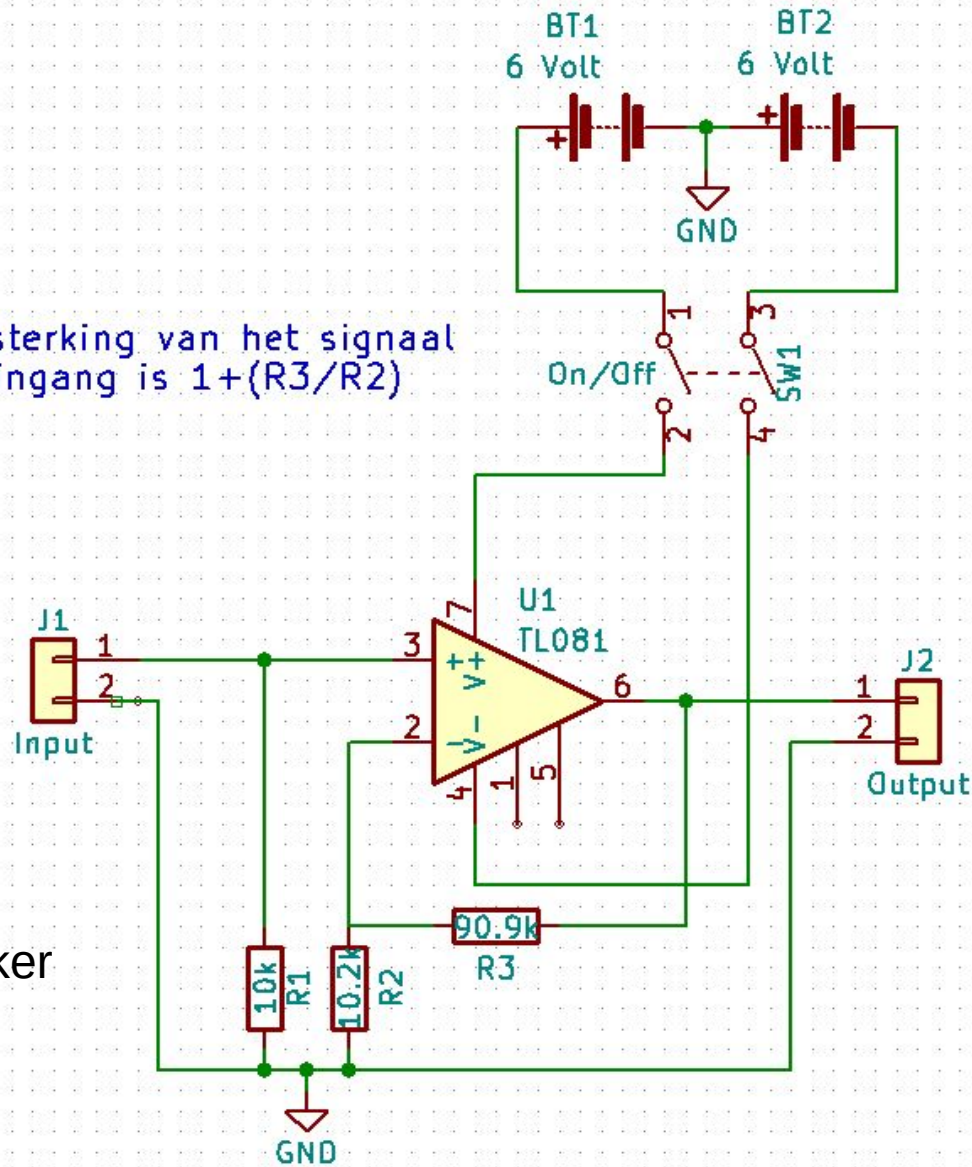
def pir_callback2(channel):
    camera.stop_preview()

GPIO.add_event_detect(12,GPIO.RISING, callback= pir_callback1)
GPIO.add_event_detect(18, GPIO.FALLING, callback=pir_callback2)
```

Sensors aansluiten op de Raspberry Pi.

De versterking van het signaal op de ingang is $1 + (R3/R2)$

Instrumentatieversterker



Sensors aansluiten op de Raspberry Pi.

PRELIMINARY



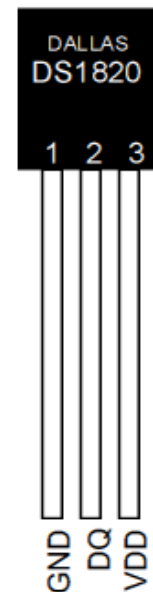
DS18B20 Programmable Resolution 1-Wire[®] Digital Thermometer

www.dalsemi.com

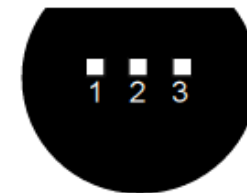
FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$. Fahrenheit equivalent is -67°F to $+257^{\circ}\text{F}$
- $\pm 0.5^{\circ}\text{C}$ accuracy from -10°C to $+85^{\circ}\text{C}$
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and

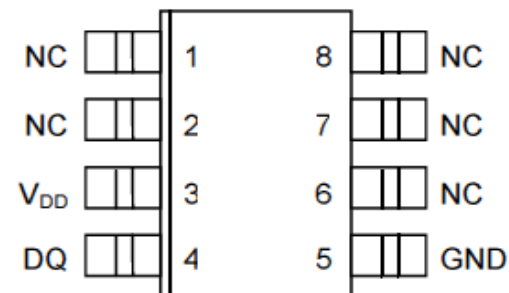
PIN ASSIGNMENT



BOTTOM VIEW



DS18B20 To-92
Package



DS18B20Z
8-Pin SOIC (150 mil)