Mini Huis Documentatie

Guillaume de Oliveira Andrezo

Sensoren

PIR Sensor

Omschrijving

De PIR (passive infrared motion) sensor is een sensor ontworpen die je in staat stelt om beweging te detecteren. Deze wordt daarom vaak gebruikt om te bepalen of een persoon zich binnen of buiten het bereik van de sensor begeeft. Ze zijn klein, goedkoop, energiezuinig, gemakkelijk te gebruiken en slijten niet.

Specificaties

- Spanning: 3.3 ~ 5V, 6V Maximum
- Stroom: 15uA
- Detectiehoek: 100 °
- Detectieafstand: 7 meter

Schakeling

De PIR-sensor wordt aangesloten 5V, ground en een digitale pin op de arduino en wordt vervolgens ingesteld als een INPUT pin. Deze werkt eigenlijk net hetzelfde als een drukknop.

```
void ReadMotion()
{
  motionVal = digitalRead(MOTIONSENSOR); // Read motionsensor
  if (motionVal == HIGH)
  { // check if the sensor is HIGH and state is LOW
    if (motionState == LOW)
    {
      motionState = HIGH; // Door is open
      Door.write(30);
    }
  }
  else
  { // If sensor is LOW and state is HIGH
    if (motionState == HIGH)
    {
      motionState = LOW; // Door is closed
      Door.write(170);
    }
  }
}
```

DHT Sensor

Omschrijving

De DHT Sensor bevat een temperatuur en vochtigheid sensor waarmee we op elk moment deze 2 grootheden mee kunnen opmeten. De sensor heeft een snelle reactiesnelheid, is accuraat en heeft een goede prijs/kwaliteit verhouding.

Specificaties

- Spanning: 5V
- Temperatuur bereik:-40 - 80°C / error < ±0.5°C
- Vochtigheid bereik: 0 - 100% RH / error ±2% RH

Schakeling

De DHT-sensor wordt, net als de PIR-sensor, aangesloten op 5V, ground en een digitale pin op de arduino. Vervolgens wordt er gebruik gemaakt van 2 libraries om de DHT aan te spreken.

```
#include "DHT.h"
#include <Adafruit_Sensor.h>

#define DHTTYPE DHT22
int DHTPin = 2;
DHT dht(DHTPin, DHTTYPE);

void setup()
{
    pinMode(DHTPin, INPUT);
    dht.begin();
    Humidity = dht.readHumidity();
    Temp = dht.readTemperature();
}
```

We gebruiken de library om een “DHT” object aan te maken die we later aanspreken om de temperatuur en vochtigheid op te vragen.

Actuatoren

Drukknoppen

Omschrijving

De drukknoppen die gebruikt werden in dit project zijn heel simpel. Namelijk de 2 voorste pinnen zijn los van de 2 achterste pinnen. Deze maken contact wanneer er op de knop gedrukt wordt, wat maakt dat het circuit sluit en er stroom vloeit.

Schakeling

Om het mezelf makkelijk te maken, gebruikte ik bij het maken van dit project de “ezButton” library. Hier zitten standaard al debounce technieken in zodat ik deze niet nog eens zelf moet implementeren. Een kant van de button gaat naar ground en de andere naar een pin op de arduino.

```
#include <ezButton.h>

ezButton button1(BUTTON_PIN);

void loop()
{
  button1.loop();
  if (button1.isPressed())
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
}
```

We beginnen eerst met het definiëren van een “ezButton” object. Hier zitten dankzij de library een aantal handige functies in die we kunnen aanroepen in ons programma. Hier links een simpel voorbeeld.

OLED Display

Omschrijving

De OLED-display in het project is een tweekleurig display met een resolutie van 128x64 pixels. Het biedt vele voordelen ten op zichten van een lcd-display, namelijk: een hoog contrast, een slank ontwerp, brede kijkhoeken en een laag stroomverbruik.

Specificaties

- Invoer spanning: 3.3V/5V
- Aanpasbaar I2C adres
- Laag stroomverbruik
- Geel en blauwe 128×64 pixels
- Hoog contrast, hoge helderheid
- Brede werktemperatuur: -40℃ ~ +85 ℃

Schakeling

Het display was een van de lastigste componenten om aan te sluiten. Het display maakt gebruik van I2C om te communiceren met de arduino. Hiervoor sluiten we de SDA-pin van het display aan op de SCL-pin van de arduino en de SCL-pin van de display op de SDA-pin van de arduino.

```
#include <Arduino.h>
#include <U8g2lib.h>
#include <SPI.h>

U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ 19, /* data=*/ 18, /* reset=*/ U8X8_PIN_NONE);   //Software I2C

void setup() {
  u8g2.begin();
  u8g2.setFont(u8g2_font_luBIS08_tf);   // choose a suitable font
}

void loop() {
  u8g2.clearBuffer();                   // clear the internal memory
  u8g2.drawStr(0,10,"Hello World!");    // write something to the internal memory
  u8g2.sendBuffer();                    // transfer internal memory to the display
  delay(100);
}
```

Om hierna gebruik te maken van het display, moeten we eerst een “u8g2” object aanmaken door gebruik te maken van de “u8g2lib” library. Vervolgens kunnen we hier een aantal functies op aanroepen nadat we het display geïnitialiseerd hebben met “begin()” en een font hebben gegeven met “setfont()”.

Servo Motor

Omschrijving

Een servo motor is een soort elektrische motor die wordt gebruikt voor het regelen van de positie van een draaiarm met hoge precisie. Deze maakt gebruik van een PWM-signaal (Pulse Width Modulation) om de Servo te sturen, we connecteren dus de data pin van de servo met een van de PWM pinnen op de arduino.

Het kan draaiende of lineaire bewegingen produceren en wordt vaak gebruikt in toepassingen waar nauwkeurige positionering en controle van de rotatiehoek belangrijk zijn.

Specificaties

- Invoer spanning: 4.8V-6V
- Invoer stroom: < 500mA
- Torque: 1.6 kg/cm
- Werk temperatuur: -30℃ ~ +60℃
- Gewicht: 9 gram

Schakeling

We maken bij de servo nogmaals gebruik van een library en een object waar we functies op aan kunnen roepen. In dit geval de “Servo” library en object. Na het aanmaken van het object koppelen we deze aan de pin waar de servo op geconnecteerd is. Vanaf dan kunnen we simpelweg een hoek schrijven naar de servo.

```
#include <Arduino.h>
#include <Servo.h>

Servo myservo; // create servo object to control a servo

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  myservo.write(0);
  delay(2000);
  myservo.write(180);
  delay(2000);
}
```

Buzzer

Omschrijving

Een piëzo buzzer is een elektronisch apparaat dat geluid produceert door het aansturen van een piëzo-elektrisch element. Wanneer elektrische spanning wordt aangelegd, veroorzaakt dit een mechanische vervorming in het keramische materiaal. Door de trillingen van het piëzo-element wordt geluid gegenereerd. De frequentie en de spanning van de aangelegde stroom bepalen de toonhoogte en het volume van het geproduceerde geluid.

Specificaties

- Invoer spanning: 5V
- Impedantie: 80 Ω
- Geluidsontwikkeling: 85 dB
- Resonantiefrequentie: 2400 Hz

Schakeling

```arduino
#include <Arduino.h>

#define BUZZER 8

void setup()
{
    pinMode(BUZZER, OUTPUT);
}

void loop()
{
    tone(BUZZER, 440, 500);
    delay(1000);
}
```

Om de buzzer in ons project te implementeren verbinden we eerst de negatieve kant van de buzzer met de ground van de arduino. Vervolgens verbinden we de positieve kant van de buzzer met een van de digitale pinnen op de arduino. Verder kunnen we gewoonweg deze pin als output zetten en met de built-in tone() functie een bepaalde frequentie voor een bepaalde lengte laten horen.

RGB LED

Omschrijving

Een RGB LED is een light emitting diode met de mogelijkheid om meerdere kleuren licht uit te zenden door het combineren van de primaire kleuren rood (R), groen (G) en blauw (B). Het is samengesteld uit 3 afzonderlijke LED-elementen (één voor elke kleur) in één behuizing.

Specificaties

- Doorlaatspanning :1,9V rood, 3,3V groen, 3,3V blauw

Schakeling

```
#include <Arduino.h>

#define RED 3
#define GREEN 5
#define BLUE 6

void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
}

void loop()
{
  analogWrite(RED, 255);
  analogWrite(GREEN, 255);
  analogWrite(BLUE, 255);
  delay(2000);
  analogWrite(RED, 0);
  analogWrite(GREEN, 0);
  analogWrite(BLUE, 0);
  delay(2000);
}
```

Aangezien een RGB LED 3 LED's in een is, moeten we alle 3 de beentjes verbinden met een digitale pin op de arduino. Het 4de en langste beentje wordt verbonden met ground bij common kathode en 5V bij common anode. De LED die gebruikt werd in dit project is common anode.

Omdat we bij een RGB LED niet alleen maar rood, groen en blauw willen laten zien, maar juist een breed scala aan kleuren. Moeten we de RGB-beentjes verbinden met PWM-pinnen op de arduino. Deze zorgen ervoor dat we de helderheid van de diodes kunnen aanpassen.

Na het instellen van de pinmodes als output kunnen we elke aan kleur individueel een waarde toekennen om ze tot de juiste kleur te bekomen.

Reflectie

Wat ging er goed en minder goed bij het programmeren van je arduino?

Alles verliep vrijwel vlekkeloos dankzij de voorkennis die ik heb van mijn vorige richting. Heb enkel even gesukkeld met het display in combinatie met al de rest. Hierover meer bij de volgende vraag.

Welke aanpassingen heb je gedaan om de sensor/actuatoren optimaal te laten werken?

Ik merkte dat als ik het display elke loop refreshte, de buttons niet meer responsive waren. De arduino is dan zodanig veel bezig met het communiceren met het display dat de buttonpresses niet elke keer herkend werden.

Om dit op te lossen heb ik gebruik gemaakt van millis(). Ik heb er namelijk voor gezorgd dat het display enkel wordt geüpdatet wanneer er een bepaalde tijd is verstreken. Deze delay kan ik zelf kiezen en staat nu op 2 seconden. Wat maakt dat de buttons werken zoals verwacht.