Freestylen met sensoren en servo

Jean-Yves Richard 2019-2020

Ten eerste zal ik beginnen van, waarom heb ik deze project gekozen, en voor deze vraag heb ik geen antwoord. Wat ik wel ga beginnen te zeggen is dat in het begin het vele anders ging zijn. Ik had eerder gedacht aan een servo die naar aanvraag een spoel ging laten wikkelen van het aantal wikkelingen dat je wou. Kleine probleem, onze servo is niet bedoelt of 360° te draaien en kan dat dus niet. Deze ging nog een temperatuur- en vochtigheid sensor hebben om op een bepaalde limiet te stoppen. Doordat deze project niet kon gemaakt worden zocht ik een ander iets maar wou nog steeds een servo gebruiken om deze te leren. Toen dacht ik weer dat deze project moest laten tonen wat je geleerd heb in deze jaar en dus heb ik wat sensoren en ander genomen en met elkaar verbonden tot een iets. Mijn eerste idee was niet zo verschillend als hoe het nu uitziet, maar het had wel een led-lampje erbij. Wat deze ging doen was aan gaan wanneer het licht onder de “limiet” ging die je kon veranderen. Wat dus leuk was met deze is dat hij dus aan ging, en wanneer hij zijn cyclus deed en de lamp (vastgeplakt aan de onderkant van de servo) boven de lichtsensor ging, ging de sensor het detecteren en lamp uit doen. Dus het ging de hele tijd zijn cyclus beginnen met een lamp aan, en zijn cyclus eindigen met lamp uit enz.

We zullen eerst beginnen elke programma voor elke project bekijken. Deze is dus de lichtsensor.

```
const int sensorPin = 34;
float lightVal;
float lightVal1;
int lightVal2;


void setup() {
  lightVal = analogRead(sensorPin);
  Serial.begin(9600);
}

void loop() {
  lightVal = analogRead(sensorPin);
  Serial.print("sensor value = ");
  lightVal1 = lightVal / 4095;
  lightVal2 = lightVal1 * 100;
  Serial.print(lightVal2);
  Serial.println(" %");
  delay(200);
}
```

Wat we eerst zullen doen is dus de sensorpin initialiseren op GPIO 34, en daaronder maken we 3 variabels aan.

We zullen dus eerst een variabel zeggen wat deze doet en dus zal analoog een pin lezen. Dan starten we nog de serie poorten.

Nu beginnen we de loop en dus zeggen we dat het variabel lightVal wordt het gegeven van het analoge waarde van gpio 34, hieronder is deze niet belangrijk maar dient om te volgen op een serie monitor. Dan wat de 2 volgende lijnen zijn is enkel om van een waarde die tussen 0 en 4095 een percentage van te maken. Dat is dus alles wat in je programma hoeft. Je kan altijd wat hieronder is ook schrijven maar is enkel om op je serie monitor te lezen.

Voor het aansluiten gaat het dus van de bron naar lichtsensor, lichtsensor gaat naar potentiometer, dan van het middelste pin van potentiometer gaat deze naar pin D34 en het laatste pin van de potentiometer gaat naar de ground.

Hieronder is nu het programma van de servo.

```
#include "esp32-hal-ledc.h"

void setup() {
  ledcSetup(1, 50, 16); // channel 1, 50 Hz, 16-bit width
  ledcAttachPin(2, 1);   // GPIO 22 assigned to channel 1
}

void loop() {
  for (int i = 100 ; i < 8000 ; i = i + 100) {
    delay(25);
    ledcWrite(1, i);
  }

  for (int i = 8000 ; i > 1000 ; i = i - 100) {
    delay(25);
    ledcWrite(1, i);
  }
}
```

Eerst gaan we een library toevoegen en dan meteen naar de setup. In de setup moet deze twee lijnen gezet worden. De loop is heel simpel, een for loop die van 100 naar 8000 zal tellen en elke stap dat hij zal zetten, zal de potentiometer ook doen. De tweede for loop zal exact hetzelfde doen maar van 8000 naar 1000. Deze library is niet het beste library voor potentiometers maar deze werkte met de ESP82 dev board dat ik gebruikte. Ik heb de cijfers ook verandert van de 100 en 1000 die normaalgezien 0 moest zijn maar heb deze verandert doordat hij te ver ging en dus de servo forceerde.

Dan heb je de OLED die een niet zo moeilijke programma is maar wel lang.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
  delay(2000);
}
```

Zoals altijd heb je du library's en hier heb je er 3 die je dus zal moeten downloaden. Dan zal je de OLED moeten definiëren van hoe groot hij is (dus 128 x 64) en juist daaronder heb je nog een stuk code die ook belangrijk is. Voor de setup heb ik weer en serial monitor gestart, en dan heb je de rest die ook erbij komt.

```
void loop() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 5);
  display.print("Lichthelderheid: ");
  display.print("100");
  display.println("%");
```

Hier is een deel van het project die ook wel het belangrijkste delen heeft. Eerst zal je de display cleanen en dan wat gegevens geven waarvan tekst grote, en tekstkleur. Belangrijk iets! Op mijn OLED heeft hij maar 1 kleur dat hij kan geven (wit) dus had ik dat gegeven weg gewist en dan werkte deze niet meer. Zou je deze programma op een arduino UNO doen, dan mag dat deel weg.
Hoe je dan begint te schrijven is heel simpel, eerst wijs je waar je zou willen schrijven dankzij de setCursor() en dan print je wat je wil printen.

Wat ook heel belangrijk is aan het einde van je programma is dat eindigt met het code display.dispay()

```
display.display();
```

Voor het aansluiten is het wat simpel. VCC moet naar de 3.3V, gnd naar ground, SDA naar SDA en SCL naar SCL. Nu het probleem dat ik had op mijn ESP32 is dat ik niet wist waar de SDA en SCL en daarvoor is er een leuk iets dat je kan doen.

```
Serial.println(SDA);
Serial.println(SCL);
```

Door deze in een apart programma te plaatsen en starten, zal je op de seriële monitor de cijfers 21 en 22 tevoorschijn komen.

Dus zal je je SDA naar D21 zetten en SCL naar D22.

Als laatste heb je het programma nodig die zich met de blynk programma verbind. Daarvoor wijs ik aan om deze te gebruiken van de site zelf die je hier kan vinden https://examples.blynk.cc/?board=ESP32&shield=ESP32%20WiFi&example=GettingStarted%2FBlynkBlink

Daarbij met je enkel zeggen welke bord je gebruikt en hij zal alles voor jou doen.

Wanneer je al deze programmas hebt moet je ze enkel nog met elkaar laten communiceren en daar zal ik het niet meer uitleggen. Maar zal het wel hieronder copy pasten.

```
//blynk
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

//Servo
#include "esp32-hal-ledc.h"

//OLED
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);


char auth[] = "PrWk_OdGEUPnAvgScTRLXzkk-wPJiw9j";
```

```
char ssid[] = "abcdefgh";
char pass[] = "12345678";

//------------ licht sensor ------------
float lightValA;
float lightValB;
int lightValC;
int AanUit = 0;

BLYNK_READ(V1) {
  lightValA = analogRead(34);
  lightValB = lightValA / 4095;
  lightValC = lightValB * 100;
  Blynk.virtualWrite(V1, lightValC);
  Blynk.syncVirtual(V1);
  delay(10);
}

//----------- licht voorwaarde -----------
int LichtVw;

BLYNK_WRITE(V2) {
  LichtVw = param.asInt();
  Serial.println(LichtVw);
}

//---------- cyclussen te maken ----------
int Cyclus;
int CyclusTD;
BLYNK_WRITE(V3) {
```

```
  Cyclus = param.asInt();
  CyclusTD = Cyclus;
  Serial.println(Cyclus);
}

void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);

  Blynk.begin(auth, ssid, pass, "server.wyns.it", 8081);

  ledcSetup(1, 50, 16);
  ledcAttachPin(2, 1);



  Serial.begin(115200);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }

  delay(2000);

}



void loop() {
  Blynk.run();
```

```
OLED();

if (CyclusTD != 0) {
  for (int i = 100 ; i < 8000 ; i = i + 100) {
    delay(25);
    ledcWrite(1, i);
  }
  lightValA = analogRead(34);
  lightValB = lightValA / 4095;
  lightValC = lightValB * 100;
  Blynk.virtualWrite(V1, lightValC);
  OLED();
  delay(300);
  if (LichtVw > lightValC) {
  AanUit = 1;
}

else {
  AanUit = 0;
}

  for (int i = 8000 ; i > 1000 ; i = i - 100) {
    delay(25);
    ledcWrite(1, i);
  }
  CyclusTD = CyclusTD - 1;
}

OLED();
delay(300);
```

```
if (LichtVw > lightValC) {
    AanUit = 1;
  }

  else {
    AanUit = 0;
  }

}


void OLED() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 5);
  display.print("Lichthelderheid: ");
  display.print(lightValC);
  display.println("%");

  display.setCursor(0, 20);
  display.print("Voorwaarde: ");
  display.print(LichtVw);
  display.println("%");

  display.setCursor(0, 35);
  display.print("Lichtstand: ");
  if (AanUit == 1) {
    display.print("Donker");
  }
```

```
  else {
    display.print("Helder");
  }

  display.setCursor(0, 50);
  display.print("Cyclus in wacht: ");
  display.print(CyclusTD);

  display.display();
}
```