



Perfect parkeren met LiDAR

Rob Reynolds (USA)

Afstandssensoren zijn geweldig! Meestal zijn ze de eerste stap van een maker bij het ontwikkelen van een autonome robot, maar wist u dat er nog veel meer toepassingen zijn? U kunt het zo gek niet bedenken. Van Halloween-rekwisieten tot irritante muziekinstrumenten en begeleidingssystemen voor parkeergarages, alles is mogelijk met afstandssensoren. Dus, laten we ze uit onze stoffige werkplaats halen, er de wijde wereld mee intrekken en er iets mee doen!

Eerst bedenken wat u wilt

Ik heb nog nooit een tennisbal of iets dergelijks aan het plafond van mijn garage gehangen om me precies te laten weten wanneer ik moet stoppen als ik mijn auto de garage binnenreed. Maar nu mijn kinderen de leeftijd hebben om te gaan rijden, wordt het daar misschien tijd voor. Nou speel ik geen tennis, maar ik speel wel met elektronica, dus ik ga gebruiken wat ik heb liggen. Nee, ik ga geen Raspberry Pi aan het plafond van mijn garage hangen. Mijn plan is om een afstandssensor te gebruiken en een klein stoplicht te maken. Als de auto de garage binnenkomt, gaat er een groene LED branden. Als de auto dichterbij komt, gaat er een gele LED branden en als de auto ver genoeg naar binnen is gekomen, ziet de bestuurder een rode LED. Als behuizing wil ik een eenvoudig stoplicht ontwerpen en 3D-printen, zodat het er ook nog goed uitziet!

Welke sensor is geschikt voor u?

Met zoveel afstandssensoren om uit te kiezen (**figuur 1**) kan het lastig zijn om de beste keuze te maken voor uw project. Er zijn verschillende overwegingen, zoals bereik, resolutie, interface, updatefrequentie en

kosten. Soms moet u beslissen welke eigenschappen minder belangrijk zijn (moet er echt 635 keer per seconde een update komen?), en welke randvoorwaarden er gelden ("de leraar zei dat we I²C-componenten moesten gebruiken"). Mijn garage is ruim acht meter lang van de achterwand tot aan de garagedeur. Misschien zou het net lukken met een van onze XL-MaxSonar ultrasoon-units, die tot ongeveer 8 meter werken, maar ik ga bij dit project voor de TFMini – Micro LiDAR Module [1].

Ik gebruik die met het SparkFun Qwiic-pakket, want daar zit een boost-board bij. De TFMini werkt op 5 V, maar communiceert op 3,3 V. Maar het boost-board dat bij het Qwiic-pakket wordt geleverd, maakt dat er geen level shifter nodig is bij gebruik van een 5V-board, en ook geen dubbele voeding als ik zou kiezen voor een 3,3V-board. Door dit te combineren met het Sparkfun Redboard, dat een ingebouwde Qwiic-connector heeft, wordt het sensorgedeelte van dit project een gewone plug&play-oplossing (**figuur 2**). Voeg nog wat heldere LED's, weerstanden en een voeding toe, en we kunnen beginnen!



Qwiic-systeem = snel en eenvoudig gebruik van I²C-componenten

Het Qwiic-systeem maakt de opbouw heel eenvoudig. De TFMMini Qwiic-versie wordt geleverd met het boost-board en een paar kabels. De ene kabel gaat van de module naar het boost-board, en de andere van het boost-board naar de Qwiic-connector op het RedBoard. De groene, gele en rode LED zijn verbonden met pinnen 8, 9 en 10. Voor de code heb ik gewoon de sketch *LidarTest.ino* (die te vinden is op de *TFMini Qwiic's Hookup Guide* webpagina [2]) in geringe mate veranderd. **Listing 1** toont het aangepaste programma. Ik heb ook een 3D-ontwerp van een stoplicht gemaakt, en die kan, net als de Arduino-sketch, worden gedownload van mijn GitHub-repository [3].

WAARSCHUWING! Omdat de TFMMini nogal wat stroom trekt en het SparkFun RedBoard een stroombegrenzing heeft, zal het niet lukken om het complete project te voeden via de voedingsconnector op het RedBoard. Dan blijft de groene LED branden en de chauffeur, die helemaal vertrouwt op dit mooie nieuwe project, knalt keihard tegen de muur. Voeden via de microUSB-aansluiting werkt wel zonder problemen. Dus ook al lonkt die voedingsconnector op het board, laat u daar niet door verleiden! Dat bespaart u een hoop ellende.

Nu is het uw beurt!

Dit project is heel gemakkelijk te bouwen, maar u kunt ook verder gaan! U hebt de code, u hebt de .STL-bestanden [3] en gebruik die als uitgangspunt. Er zijn allerlei mogelijkheden voor uitbreiding:

- voeg wat knoppen toe om de optimale remafstand gemakkelijker in te stellen;
- ontwerp en print een betere behuizing;
- gebruik meer LED's om het stoplicht beter zichtbaar te maken.

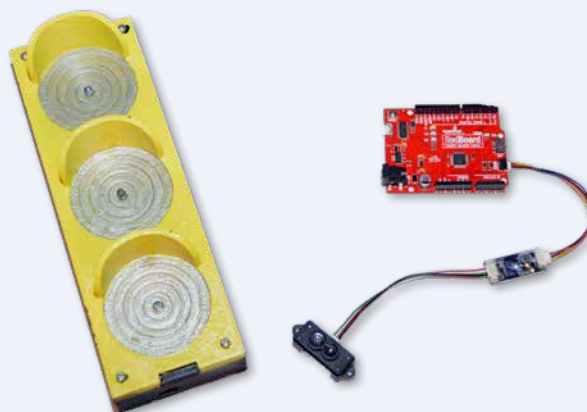
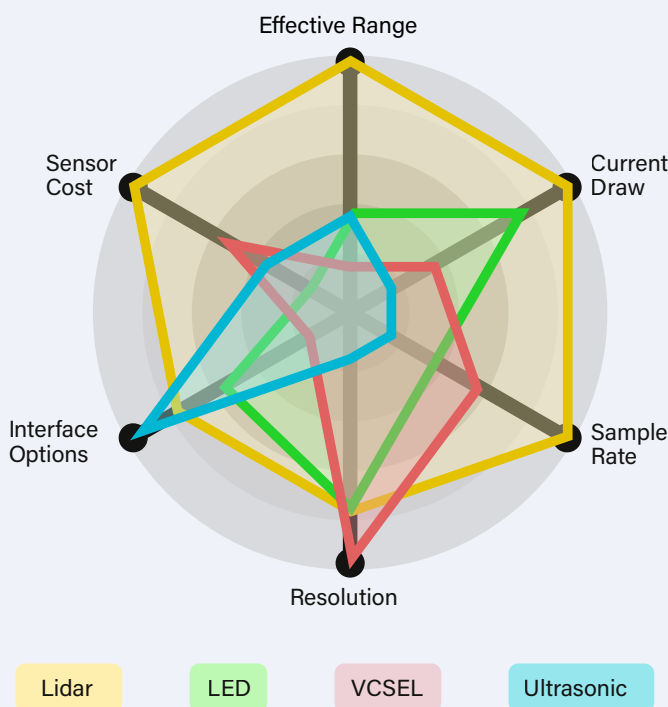
Misschien kunt u zelfs het RedBoard omruilen voor een Raspberry Pi en dit project in Python programmeren, waarbij via een display meer visuele feedback mogelijk is.

Biedt het gebruik van een enkelkaartcomputer zoals de Raspberry Pi bij dit project enig voordeel ten opzichte van een microcontroller? Absoluut niet. Maar waarom zou u een project simpel houden als u het ook kunt overdrijven? Happy hacking, mensen!

200698-04

[vervolg op volgende pagina]

Distance Sensor Comparison



Figuur 2. De elektronica en de 3D-geprinte behuizing voor het project 'Perfect Parkeren met Lidar'.

Figuur 1. Om de juiste afstandssensor te kiezen voor uw toepassing, moet u een afweging maken tussen de beschikbare technologieën en uw ontwerpdoel(en).



Listing 1. De sketch voor Perfect Parking with LiDar

```
/*
  TFMiniStopLight.ino
  Rob Reynolds, November 19, 2018

  This code is a small practical demonstration
  application of the TFMini Lidar Module. The 3D
  files can be found in the Github repository, here
  [ https://github.com/ThingsRobMade/TFMini\_Stop\_Light ]
  Based heavily on the previous collaborative work
  done by Nate Seidle and Benewake. The original
  example sketch for the Qwiic Enabled TFMini can be
  found here:
  (https://www.sparkfun.com/products/14786)

  This code is free, but if you find it useful,
  and we meet someday, you can buy me a beer
  (Beerware license).
*/

#include <Wire.h>

uint16_t distance = 0; //distance
uint16_t strength = 0; //signal strength
uint8_t rangeType = 0; //range scale
/*Value range:
  00 (short distance)
  03 (intermediate distance)
  07 (long distance) */

boolean valid_data = false;
//ignore invalid ranging data

const byte sensor1 = 0x10;
//TFMini I2C Address

// Define pins for LEDs
const int greenLED = 8;
const int yellowLED = 9;
const int redLED = 10;
int stopLimit = 160; //Change this number of cm to
adjust stop distance

void setup()
{
  Wire.begin();

  Serial.begin(115200);
  Serial.println("TFMini I2C Test");
  //For testing using Serial Monitor

  // Set LED pins as outputs
  pinMode(redLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
}

void loop()
{
```

*U zou zelfs het
RedBoard kunnen
omruilen voor een
Raspberry Pi en dit
project in Python
programmeren.*

Rob Reynolds

```
if (readDistance(sensor1) == true)
{
  if (valid_data == true) {
    Serial.print("\tstopLimit");
    /* These Serial.print lines remain for testing and
    adjustment purposes */
    Serial.print(stopLimit);
    Serial.print("\tDist");
    Serial.print(distance);
    Serial.println();

    if (distance <= stopLimit) {
      digitalWrite(redLED, HIGH);
      digitalWrite(yellowLED, LOW);
      digitalWrite(greenLED, LOW);
    }
    else if (distance > stopLimit && distance
    < (stopLimit + 200) ) { //change this number to
    increase distance yellow stays lit
      digitalWrite(redLED, LOW);
      digitalWrite(yellowLED, HIGH);
      digitalWrite(greenLED, LOW);
    }
    else if (distance > (stopLimit + 199) ) {
    //change this number to adjust when yellow LED
    illuminates
      digitalWrite(redLED, LOW);
      digitalWrite(yellowLED, LOW);
      digitalWrite(greenLED, HIGH);
    }
  }

  // else {
  //   Serial.println("Read fail");
  // }

  delay(50);
  //Delay small amount between readings
}

//Write two bytes to a spot
boolean readDistance(uint8_t deviceAddress)
{
```



```

Wire.beginTransmission(deviceAddress);
Wire.write(0x01); //MSB
Wire.write(0x02); //LSB
Wire.write(7); //Data length: 7 bytes for
distance data
if (Wire.endTransmission(false) != 0) {
    return (false); //Sensor did not ACK
}
Wire.requestFrom(deviceAddress, (uint8_t)7); //
Ask for 7 bytes

if (Wire.available())
{
    for (uint8_t x = 0 ; x < 7 ; x++)
    {
        uint8_t incoming = Wire.read();

        if (x == 0)
        {
            //Trigger done
            if (incoming == 0x00)
            {
                //Serial.print("Data not valid: ");
                //for debugging
                valid_data = false;
                //return(false);
            }
            else if (incoming == 0x01)
            {
                Serial.print("Data valid:    ");
                valid_data = true;
            }
        }
        else if (x == 2)
            distance = incoming;
        //LSB of the distance value "Dist_L"
    }
}

```

```

        else if (x == 3)
            distance |= incoming << 8; //MSB of the
distance value "Dist_H"
        else if (x == 4)
            strength = incoming; //LSB of signal
strength value
        else if (x == 5)
            strength |= incoming << 8; //MSB of signal
strength value
        else if (x == 6)
            rangeType = incoming; //range scale
        }
    }
    else
    {
        Serial.println("No wire data avail");
        return (false);
    }

    return (true);
}

```



Gerelateerde producten

Bent u op zoek naar de belangrijkste producten die in dit artikel genoemd zijn?
Elektor en SparkFun staan voor u klaar!

> **SparkFun RedBoard - geprogrammeerd met Arduino**
www.elektormagazine.nl/esfe-en-perfectparking1

> **TFMini - Micro LiDAR Module**
www.elektormagazine.nl/esfe-en-perfectparking2



WEBLINKS

[1] TFMini - Micro LiDAR Module: <https://www.sparkfun.com/products/14588>

[2] TFMini Qwiic's Hookup Guide: <https://bit.ly/2Xzun4r>

[3] Perfect parkeren: sketch en STL-bestanden voor de behuizing: https://github.com/ThingsRobMade/TFMini_Stop_Light