**Projectdoel**

Het hoofddoel van dit project is de ontwikkeling van een IoT-toepassing: een slimme auto die via meerdere methoden bestuurd kan worden. De auto kan:

* Handmatig worden bestuurd via een **web interface** of een **PlayStation-controller**.
* Autonoom rijden met botsingsvermijding, gebaseerd op sensorgegevens.

De auto verzamelt en verwerkt gegevens zoals versnelling, wiel-RPM’s en afstanden tot obstakels.

**1. Sensoren (Inputs)**

De volgende sensoren zijn gebruikt voor het verzamelen van gegevens:

* **Afstandssensoren**: SR-04 ultrasone sensoren, met logische niveauomzetter om signalen geschikt te maken voor de PSoC6.
* **RPM Teller**: Pulsmodules die het aantal omwentelingen per minuut meten.
* **Accelerometer**: MPU6050 (geïntegreerd via I2C), voor het detecteren van versnelling in de X- en Z-richtingen.
* **Thermistor**: Geïntegreerd op de PSoC zelf, voor het monitoren van de temperatuur.

**2. Actuatoren (Outputs)**

De actuatoren sturen de auto en voorzien deze van verschillende outputopties:

* **Motor Drivers**: Twee drivers die elk twee wielen aansturen via PWM-signalen, waarmee de draaisnelheid en richting worden geregeld.
* **DC-motoren**: Vier motoren, gekoppeld aan mecanum-wielen om omnidirectionele beweging mogelijk te maken.
* **Voedingsbronnen**: Twee 9V-batterijen voor de voeding van de motoren.

Elke commandoactie wordt vertaald naar 8-bit signalen (2 bits per wiel). Voor elk wiel geld dan:

* Voorwaarts: 10
* Achterwaarts: 01
* Stilstaan: 00

**3. Gebruikersinterface**

De auto kan worden bestuurd via twee methoden:

* + Een op python-gebaseerde webserver, waar gebruikers de auto handmatig kunnen besturen en sensorgegevens in real-time kunnen bekijken, de controller is ook verbonden met de python server.
  + Bidirectionele communicatie via het MQTT-protocol wordt gebruikt voor opdrachten en gegevensoverdracht.

**4. Autonoom rijden**

De auto kan autonoom rijden met behulp van:

* **Botsingsvermijding**: Berekeningen op basis van gegevens van ultrasone sensoren zorgen ervoor dat obstakels worden vermeden.

**5. FreeRTOS Code Structuur**

Het FreeRTOS-besturingssysteem wordt gebruikt voor:

* Gelijktijdige verwerking van sensorgegevens.
* Aansturing van actuatoren via PWM-signalen.
* Verwerking van gebruikerscommando’s.

De code is modulair gestructureerd, waardoor onderhoud en foutopsporing eenvoudiger zijn.

**6. Veiligheid en Low-Power Mode**

* **Veiligheid:** Door gebruik te maken van de poort 8883 die gereserveerd is voor TLS, is de verbinding van de clients met de HiveMQ broker versleuteld. Hierdoor kan deze verbinding niet afgeluisterd worden. Door gebruik te maken van gebruikers met wachtwoorden kunnen de clients zich ook authentiseren, waardoor niet iedereen zomaar kan verbinden met de broker.
* **Energiebeheer:** PWM-signalen worden gebruikt om het stroomverbruik van de motoren te beperken.

**Controle van vereisten**

| **Vereisten** | **Status** |
| --- | --- |
| 2 sensoren | ✅ SR-04, MPU6050, RPM teller en thermistor. |
| 2 actuatoren | ✅ LED, vier wielen |
| Gebruik van I2C | ✅ MPU6050 via I2C |
| Gegevensberekening | ✅ Afstandsberekening, versnelling, RPM |
| Bidirectionele WiFi (MQTT) | ✅ MQTT gebruikt |
| Gebruikersinterface op website | ✅ Controller en dataweergave |
| Meerdere apparaten (2 sensor nodes) | ✅ Gebouwd |
| FreeRTOS en PSoC6 | ✅ Gebruikt |
| Low-power modus | ✅ PWM-signalen |
| Veiligheid | ✅ versleutelde poort en wachtwoorden |