

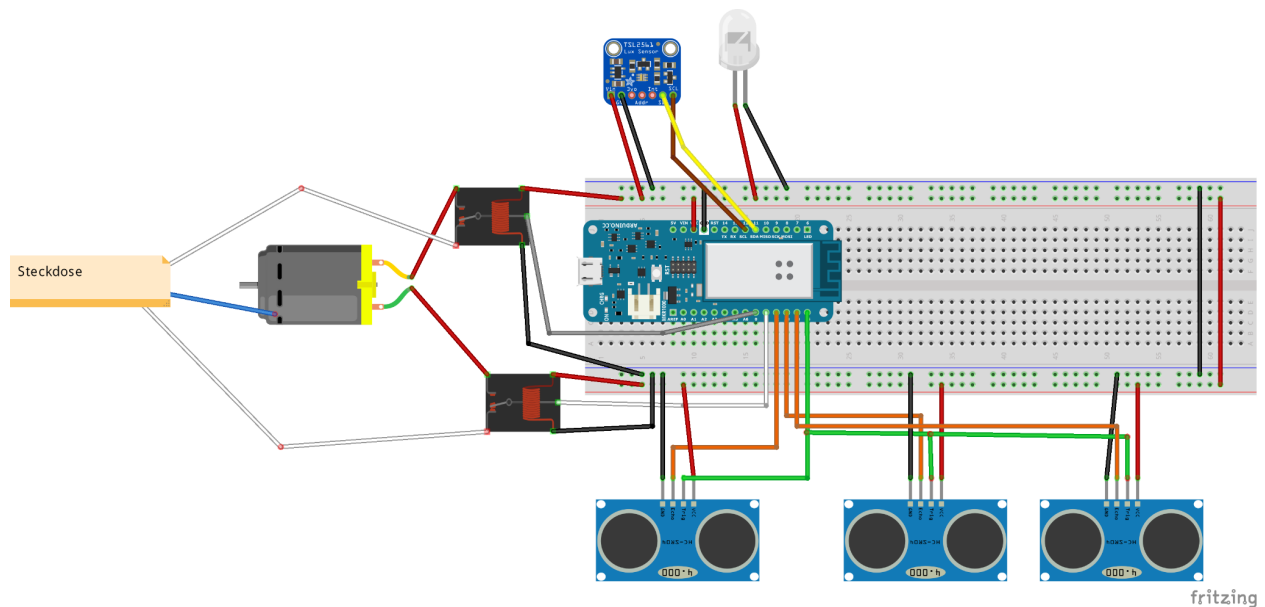
Projektbeschreibung SmartGarage

Projektidee

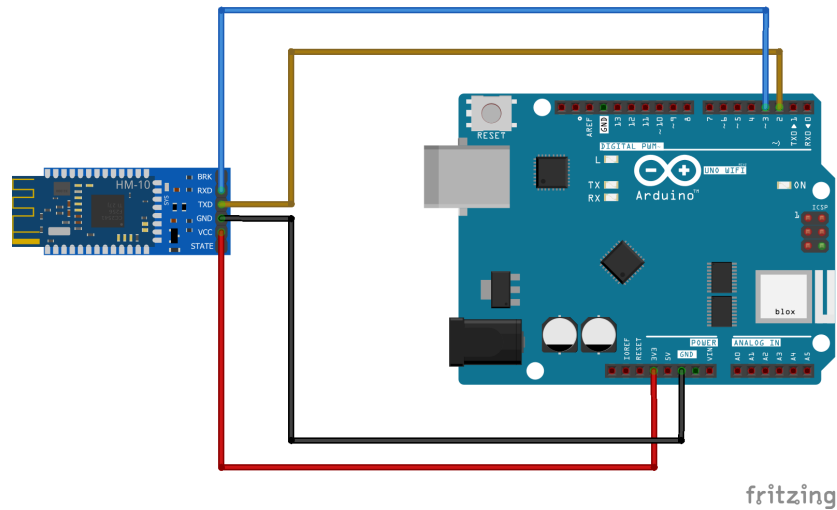
Bei Annäherung eines Autos an eine Garage soll sich das Tor automatisch öffnen. Die Steuerung des Tores wird dabei von einem Arduino übernommen. Zur Annäherungserkennung und Authentifizierung wird eine Beacon-Simulator App verwendet. Zur Übertragung der Öffnungsbefehle wird Mqtt verwendet. Der Broker sitzt am Tor in einem Raspberry Pi. Außerdem kann per Blynk-App und Node-Red Flow das Tor zusätzlich geöffnet und geschlossen werden. Ein Lichtschranken sorgt zudem dafür, dass, falls ein Auto im Torbereich steht, das Tor nicht geschlossen werden kann. Ultraschallsensoren im Inneren der Garage erkennen, ob ein Auto in der Garage steht oder nicht. Die Umsetzung wird mit einem ferngesteuerten Auto und einer Modellgarage aus Holz geplant. Als Tor dient ein Fensterrollo.

Schaltung

Garage



Fahrzeug



Requirements und verwendete Tools

- I. **Verwenden Sie auf jeden Fall 2 Boards**
 - A. 1x Arduino MKR1000, 1x Arduino Uno WiFi Rev2, 1x Raspberry Pi 3b
- II. **Verwenden Sie zumindest 1 Sensor/Aktuator, der NICHT Teil des IoT Bundles ist**
 - A. z.B. Lichtsensor TSL2561
- III. **Verwenden Sie insgesamt mindestens 3 Sensoren/Aktuatoren**
 - A. Sensoren: 3x Ultraschallsensor HC-SR04, 1x Lichtsensor TSL2561, 1x Bluetooth Modul HM10
 - B. Aktuatoren: 1x AC Motor, 2x Relais, 1x Leuchtdiode
- IV. **Verwenden Sie in Ihrem Projekt auf irgendeine Art und Weise Ihr Smartphone**
 - A. Blynk-App zur Steuerung des Tores
- V. **Verwenden Sie in Ihrem Projekt eine kabellose (WiFi) Verbindung zu einem Computer bzw. der Cloud**
 - A. MQTT mit Node-Red zur Steuerung des Tores

Komponentenbeschreibung

- **Arduino MKR1000:** Sitzt am Dach der Garage, empfängt die Signale der Ultraschallsensoren und des Lichtsensors. Steuert auf Befehl vom Mqtt Topic "/garage_command" das Tor und berücksichtigt dabei die Sensorwerte.
- **Arduino Uno WiFi Rev2 + Bluetooth Modul HM-10:** Das HM-10 Bluetooth Modul ist mit einem CC2541 Chip ausgestattet und funktioniert unter iOS und Android. Dieses

Modul lässt sich mit verschiedensten AT+ Befehlen steuern. Dies erfolgt mittels den Arduino Uno Wifi Rev2. Die Geräte wurden am Ferngesteuerten Fahrzeug montiert und schicken einen Öffnungsbefehl, wenn sie sich nahe genug am Garagentor befinden.

- **Raspberry Pi 3b:** Mikrocontroller, auf dem ein Linux Betriebssystem (Raspberry Pi OS) läuft. Sitzt ebenso am Dach der Garage, jedoch ohne Verbindung zur Schaltung. Auf dem Raspberry läuft der Mqtt Broker (Eclipse Mosquitto), der die Nachrichten von den Subscribern empfängt und an die Publisher verteilt.
- **Ultraschallsensoren HC-SR04:** Eignet sich zur Entfernungsmessung von bis zu 5 Metern mit einer Auflösung von 3mm. Die Sensoren sind in der Garage verbaut und schicken ein Signal (alle 100ms), falls sich ein Auto in der Garage befindet.
- **Lichtsensoren TSL2561 + Leuchtdiode:** Leuchtdiode strahlt Licht auf Sensor. Bei starker Änderung der aufgenommenen Lichtstärke befindet sich ein Hindernis zwischen Sensor und Diode (in Nähe des Tors) und das Schließen des Tores wird verhindert.
- **AC Motor + Relais SRD-5VDC-SLC:** Der AC Motor, ein Rolladenmotor, wird über 2 Relais mit dem Arduino MKR1000 angesteuert. Zusätzlich wird am AC Motor die maximale Höhe des Rollos im offenen, als auch im geschlossenen, Zustand eingestellt.
- **Smartphone mit Beacon-Simulator App:** Zur Annäherungserkennung dient eine Beacon-Simulator App auf einem Smartphone, die in einer festgelegten Frequenz Bluetooth-Signale aussendet, die in der Reichweite begrenzt werden können. Mit dem HM-10 Modul am Fahrzeug wird bei Erkennung des Signals die Öffnung der Garage eingeleitet.
- **Ferngesteuertes Fahrzeug:** Wird zu Demonstrationszwecken verwendet. Normal steuerbar mit der Fernbedienung.

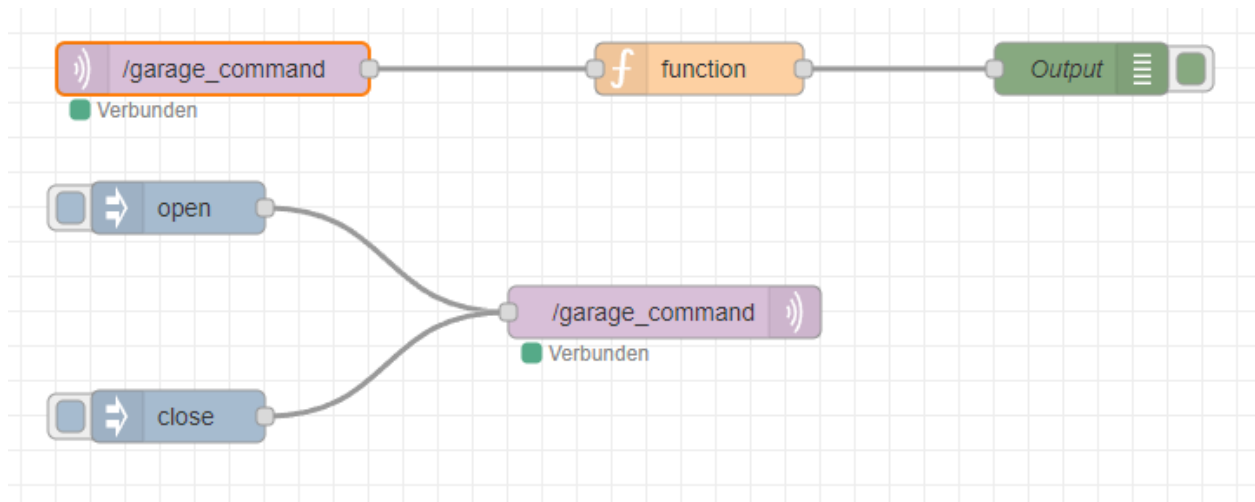
Beschreibung der Software

- **Mqtt Broker:** Wird durch `sudo apt install mosquitto` am Raspberry installiert und ist sofort empfangs- und sendebereit. Die lokale IP-Adresse des Raspberry wird statisch auf 192.168.0.169 konfiguriert, um bei Neustart des Systems die Adresse des Brokers gleich zu lassen.
- **Mqtt Client:** Am Arduino Uno WiFi Rev2, Arduino MKR1000 und in Node-Red werden Mqtt-Clients aufgesetzt. Ersterer dient als Publisher zum Öffnen des Tores, der MKR1000 agiert als Subscriber und Node-Red erfüllt beide Funktionen. Auf den Arduinos wird die Library ArduinoMqttClient verwendet.
- **Ultraschallsensoren:** Verwendet die eigen geschriebene Library Ultrasonic-master.zip vom Professor Plattner. Bei erkannter Veränderung der Distanz in der Garage wird mitgeteilt, dass sich ein Auto in der Garage befindet.
- **Lichtschranke:** Verwendet die Adafruit_TSL2561 Library zur Auswertung der Lichtwerte. Die Diode sendet einen roten Lichtstrahl durchgehend auf den Sensor wodurch eine hohe Lichtleistung in Lux(lumen/m²) erreicht wird. Bei starker Änderung des Werts / Unterbrechung des Lichtempfangs werden Schließ-Befehle des Tors blockiert.

- **Motoransteuerung:** Mit Hilfe des Kommandos `digitalWrite([Port],HIGH);` lässt das Relai Strom durchfließen und mit Hilfe des Kommandos `digitalWrite([Port],LOW);` wird die Stromzufuhr an dem Port unterbrochen. Der Port gibt in diesem Kommando an, welches Relai angesteuert werden soll (Der Port an dem das Signalkabel am Arduino angeschlossen ist).
- **Blynk:** Mithilfe zweier Buttons “open” und “close” werden Funktionen am Arduino MKR1000 aufgerufen, welche zum Schließen oder Öffnen des Tores dienen

Screenshots

- Node-RED



- Blynk-App

