

Projekt: Brandmelder

Um dieses Projekt meistern zu können solltest du vorher die Dokumentation folgender Sensoren angesehen und verstanden haben

➔ **Flamm-Sensor**

➔ **Gas-Senor**

Der darüber hinaus im Projekt vorkommende Summer hat lediglich einen Plus- (langes Bein) und einen Minus-Pol (kurzes Bein). Er wird über GND und einen PinOutput angesprochen.

Ziel des Projekts

Du sollst den prinzipiellen Aufbau und die Funktion eines Brand-Melders (Rauch-Melder, Feuer-Melder) verstehen und nachvollziehen.

Dabei wird ein Flamm-Sensor verwendet, der auf offene Flammen (Test mit Feuerzeug) reagiert.

Außerdem kommt ein Rauch-Sensor zum Einsatz, der bereits geringe Mengen Rauch in der Luft detektieren kann.

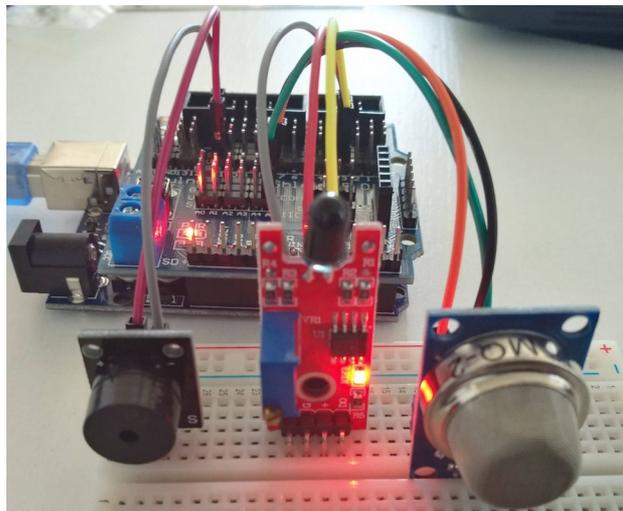
Spricht einer der beiden Sensoren an, so soll ein lauter Warnton erklingen.

Hardware

Du benötigst

- 1 Gas-Sensor** (MQ-2 oder MQ-3 oder MQ-135)
- 1 Flamm-Sensor** (Infrarot-Sensor)
- 1 aktiven Piezo-Summer**
- 1 Arduino** (Uno)
- 1 Steckbrett**
- 1 Sensor-Shield**
- 8 Kabel** (male/female)

Aufbau



Der **Summer** kommt auf einen **Ausgangs-Pin**.

Der **Flamm-Sensor** kommt auf einen **digitalen Eingangs-Pin**.

Der **Gas-Sensor** kommt auf einen **analogen Eingangs-Pin**.

Technische Vorbereitungen

Tipp: Stecke vor jedem einzelnen Test das Arduino ab und wieder an und starte die virtuelle Maschine in BlueJ neu.

Schließe bei seltsamen **Fehler-Meldungen (rot in der Konsole)** BlueJ, stecke das Arduino ab und wieder an und öffne danach BlueJ neu. Damit sparst du dir sehr viel Frust ...

1. **Teste zuerst nur den Summer alleine interaktiv**, d.h. Erstelle in BlueJ durch Rechts-Klick ein Objekt der Klasse PinOutput und prüfe interaktiv, ob dein Summer wirklich funktioniert.

2. **Teste nun nur den Flamm-Sensor, ohne (!!!)** seinen digitalen Ausgang mit dem Arduino zu verbinden. Stecke zunächst nur GND und 5V an.

Der Sensor hat eine Kontroll-LED (im Bild unter dem Poti), die angeht, wenn eine Flamme detektiert wird und wieder ausgeht, wenn keine Flamme detektiert wird. Das Sonnenlicht oder manche Lampen können aber stören! Drehe am Potentiometer, bis die Kontroll-LED aus ist und teste mit einem Feuerzeug in ca. 20cm Entfernung, ob die LED bei einer Flamme angeht. Schließe erst jetzt das Kabel vom digitalen Ausgang des Flamm-Sensors an einen digitalen Eingang des Arduino.

Teste interaktiv die Methoden `isHigh()` und `isLow()` jeweils bei ein- und ausgeschaltetem Feuerzeug.

3. Der **Gas-Sensor** muss ca. **1-2 Minuten** an GND und 5V angeschlossen sein, bis er **aufgewärmt** ist.

Teste nun den Gas-Sensor interaktiv indem du per Rechts-Klick ein Objekt der Klasse PinAnalog erzeugst. Rufe die Methode `readValue()` mehrmals auf und ändere unmittelbar vorher den Zustand der Umgebungs-Luft:

Halte die Luft an und hauche anschließend langsam und andauernd über den Sensor. Der Wert sollte nun etwas steigen.

Lasse etwas Gas des Feuerzeugs zum Sensor strömen. Der Wert sollte nun deutlich steigen. Lasse ein Streichholz etwas brennen, blase es aus und halte das noch rauchende Streichholz direkt unter den Sensor. Der Wert sollte nun ähnlich wie beim Versuch mit Feuerzeug-Gas sein.

Merke dir die Werte für „saubere“ Umgebungs-Luft, Gas bzw. Rauch.

Die Software

4. Erstelle eine **Klasse BrandMelder**, mit je einem Referenz-Attribut der Typen PinInput, PinOutput und PinAnalog.

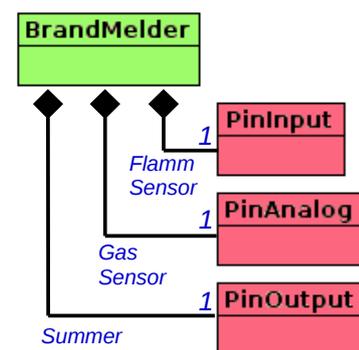
5. Erzeuge im **Konstruktor** die Objekte an den entsprechenden Pins. Setze den Summer auf LOW.

6. Schreibe eine **Methode aktivieren()**:

In einer Endlos-Wiederholung [`while (true)`] erledigst du nun folgendes:

Wenn der Flamm-Sensor eine Flamme erkennt oder der Gas-Sensor einen Wert liefert, der um 150 höher liegt als der für „saubere“ Luft (*variieren diesen Wert gegebenenfalls*), so soll der Summer angehen, ansonsten soll der Summer ausgehen.

Nach der Fallunterscheidung (*aber noch innerhalb der Wiederholung*) gönnst du dem System eine halbe Sekunde Pause :-)



Das Programm kannst du nun nur durch einen Neustart der virtuellen Maschine stoppen!