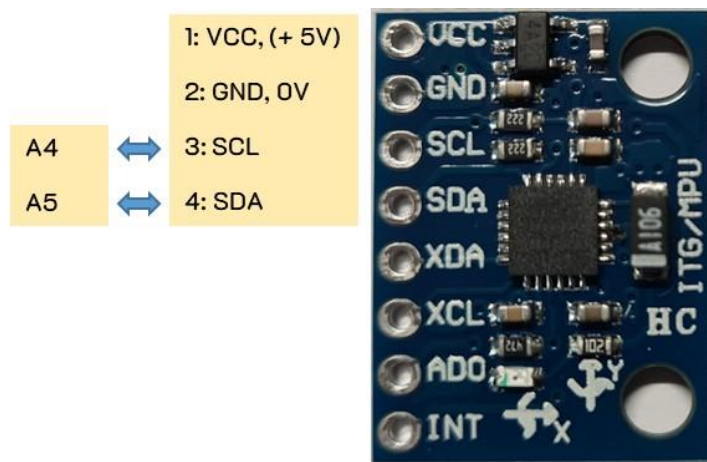




Der Arduino und Gyrosensoren

Gyrosensoren bieten die Möglichkeit, die Bewegung des Roboters im Raum zu überwachen und dem Programm damit zusätzlich die Möglichkeit zu bieten, Bewegungen aufzuzeichnen. Das folgende Beispiel orientiert sich am GY-521 Modul mit dem MPU6050. Es handelt sich um ein 3-Achsen-Gyroskop, ein 3-Achsen-Beschleunigungsmesser, einen digitalen Bewegungsprozessor und einen Temperatursensor. Mit dem integrierten Prozessor können die Rohdaten aufbereitet werden, sofern man sie nicht selbst aufbereiten will.



Das Modul wird über den I2C-Bus angeschlossen, es benötigt daher zwingend die beiden Anschlüsse A4 und A5 (Arduino Uno).

Dazu bietet der Arduino die Bibliothek „Wire“, deren Headerdatei wir mit

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

einbinden.

Dann nutzen wir noch die Bibliothek „MPU6050_light“, die uns die Möglichkeiten der MPU erschließt, ohne dass wir uns mit der dahinter liegenden Theorie befassen müssten (die Theorie hinter der Erfassung der Rohdaten, ihrer Kalibrierung und der Berechnung der Roll-Nick-Gier-Bewegung würde den Rahmen hier sprengen).

Installiert die oben genannte Bibliothek (über die Arduino-Bibliotheksverwaltung) und bindet sie in das Programm ein:

```
#include <MPU6050_light.h>
```

Wir deklarieren und eine Variable und initialisieren sie durch Übergabe des Wire-Objekts. Gleich danach erstellen uns wieder eine Initialisierungsfunktion für den Gyro:



```
MPU6050 Mpu(Wire);  // Initialisierung des Mpu-Objekts

void initGyro() {
  Wire.begin();
  Mpu.begin();
  // Achtung: 1 Sekunde nicht bewegen zur Kalibration des Offsets
  delay(1000);
  Mpu.calcOffsets();
}
```

Wie man dem Code entnehmen kann, benötigt die MPU etwa eine Sekunde, um sich zu kalibrieren. Während dieser Zeit darf das Fahrzeug nicht bewegt werden.

Jetzt können wir mit dem Objekt Mpu arbeiten.

Nun können wir die Daten abfragen.

```
Mpu.update();

Serial.print("X: " + String(Mpu.getAngleX()) + "\t");
Serial.print("Y: " + String(Mpu.getAngleY()) + "\t");
Serial.println("Z: " + String(Mpu.getAngleZ()));

delay(50);
```

Die Abfrage der Rohdaten und die Umrechnung erledigen wir durch den Aufruf Mpu.update(). Anschließend können wir beispielsweise die Winkel abrufen (das reicht für unsere Aufgabenstellungen).

Der Z-Winkel beschreibt die Rotation um die Hochachse, also genau die Bewegung, die das Fahrzeug macht, wenn wir drehen.

Diesen Winkel wollen wir verwenden, um unsere Fahrzeugsteuerung besser zu kontrollieren. Der X-Winkel würde eine Drehung des Fahrzeugs um die Längsachse beschreiben (z.B., wenn das Fahrzeug schief fahren würde, etwa mit 2 Rädern über eine Rampe) und der Y-Winkel beschreibt uns die Schiefe, die hinauf- oder herunterfahren würden (die Steigung).

Achtung: leider benötigt die Wire-Bibliothek den Zugriff auf die Interrupts. Die Abfrage der MPU in der ISR ist daher nicht möglich. Statt dessen nutzen wir die MPU mit der millis()-Funktion. Grundsätzlich sind auch andere Installationen möglich. Thematisch stehen hier jedoch die Robot-Funktionen im Vordergrund, weswegen die vorhandenen Bibliotheken Verwendung finden sollen.