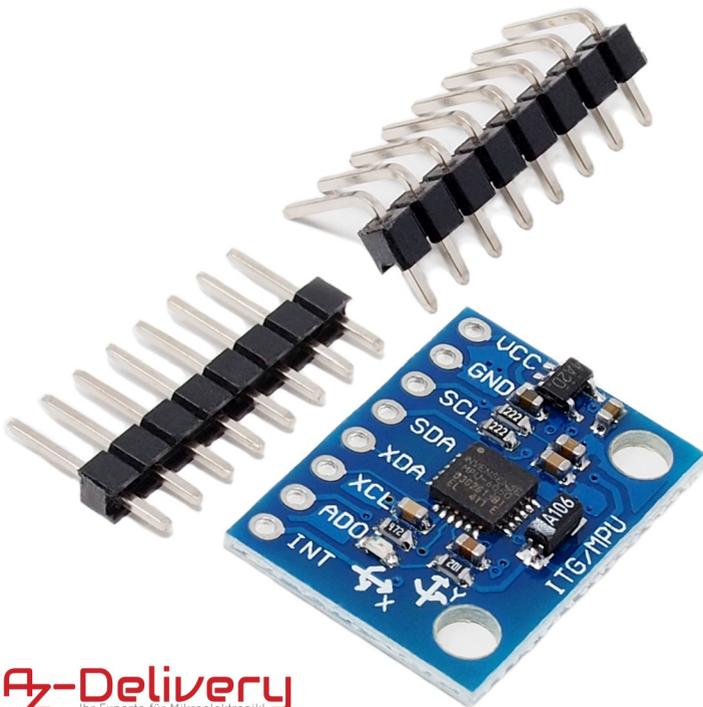


Willkommen!

Und herzlichen Dank für den Kauf unseres MPU6050 6-DOF-Sensors! Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zur Ausgabe der Werte.

Viel Spaß!



Das GY-521 Sensor-Modul mit MPU6050 Chip ist ein Lage- und Beschleunigungssensor für drei Achsen. Neben diesen sechs Werten gibt der über das I2C-Protokoll angesprochene Sensor auch die Temperatur aus. Das Auslesen der Daten ist nicht sehr anspruchsvoll, die Auswertung jedoch schon. Da es verschiedene Filter und Kalibrierungsmöglichkeiten gibt zeigen wir Ihnen heute das Auswerten der Daten über den integrierten DMP.

Das Modul bietet neben dem Anschluss als I2C-Slave auch die Möglichkeit als I2C-Host und I2C-Slave gleichzeitig zu fungieren. So besteht die Option, weitere Sensoren, beispielsweise ein Kompass-Modul oder einen Luftdrucksensor an die Pins XDA und XCL anzuschließen und die Werte über den DMP verarbeiten zu lassen.

Eine genaue Verwendung des 1024 Byte großen FIFO-Speicher sowie der internen Register entnehmen Sie bitte der weiter unten im Text verlinkten Register-Map.

Das Modul kann dank einem Festspannungsregler auf der Platine zwischen 5 und 3.3V betrieben werden.

Bevor der Sensor brauchbare Daten liefert lassen wir diesen ein paar Minuten im Betrieb um sich zu akklimatisieren und bestimmen im Anschluss die Offset Werte, welche wir dann im Sketch anpassen.

Vor der Inbetriebnahme neuer I2C-Geräte ist man generell gut beraten die Adresse des Chips mithilfe eines I2C-Scanners auszulesen, da diese variieren kann und wir den Scanner benutzen können um eine gewissen Grundfunktionalität zu prüfen.

Die wichtigsten Informationen in Kürze

» **Abmessungen:** 21mm x 16mm x 5mm

» **Verbindung:**

VCC	3.3V / 5V
GND	Masse
SCL	I2C-Clock
SDA	I2C-Data
XDA	I2C-Master-Data
XCL	I2C-Master-Clock
ADO	high / low (Adress-Select)
INT	Interrupt

» **Temperaturbereich:** -10 - 45 °C

» **Maximale Leistungsaufnahme:** 25mW

» **Programmierung über I2C-Bus**

Auf den nächsten Seiten findest du Informationen zur

» ***Einrichtung der Hardware***

und eine Anleitung für

» ***das Auslesen der Sensordaten.***

Diese Anleitung setzt voraus, dass du weißt, wie du Sketche auf einen Arduino hochlädst und den Serial Monitor verwendest!

Alle Links im Überblick

MPU6050:

- » Datenblatt: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- » Register-Mapping: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>
- » Bibliothek : <https://github.com/jrowberg/i2cdevlib>
- » I2C-Scanner : <https://playground.arduino.cc/Main/I2cScanner>

Programmieroberflächen:

- » Arduino IDE: <https://www.arduino.cc/en/Main/Software>
- » Web-Editor: <https://create.arduino.cc/editor>
- » Arduino-Erweiterung für SublimeText: <https://github.com/Robot-Will/Stino>
- » Arduino-Erweiterung "Visual Micro" für Atmel Studio oder Microsoft Visual Studio: <http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx>
- » PlatformIO: <https://platformio.org/>

Arduino Tutorials, Beispiele, Referenz, Community:

- » <https://www.arduino.cc/en/Tutorial/HomePage>
- » <https://www.arduino.cc/en/Reference/HomePage>

Interessantes von AZ-Delivery

» Arduino Zubehör:

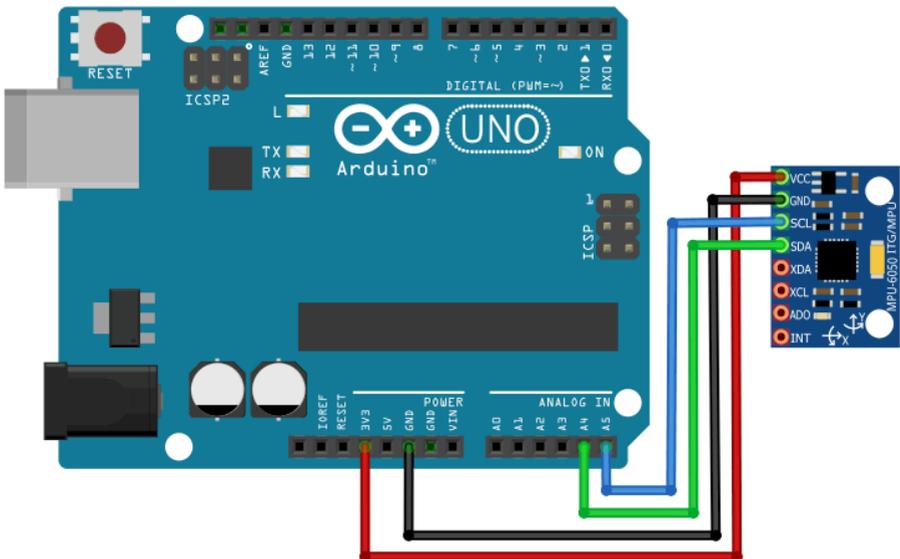
<https://az-delivery.de/collections/arduino-zubehor>

» AZ-Delivery auf Facebook:

<https://www.facebook.com/AZDeliveryShop/>

Einrichtung der Moduls

Wir beginnen mit der Einrichtung der Hardware:

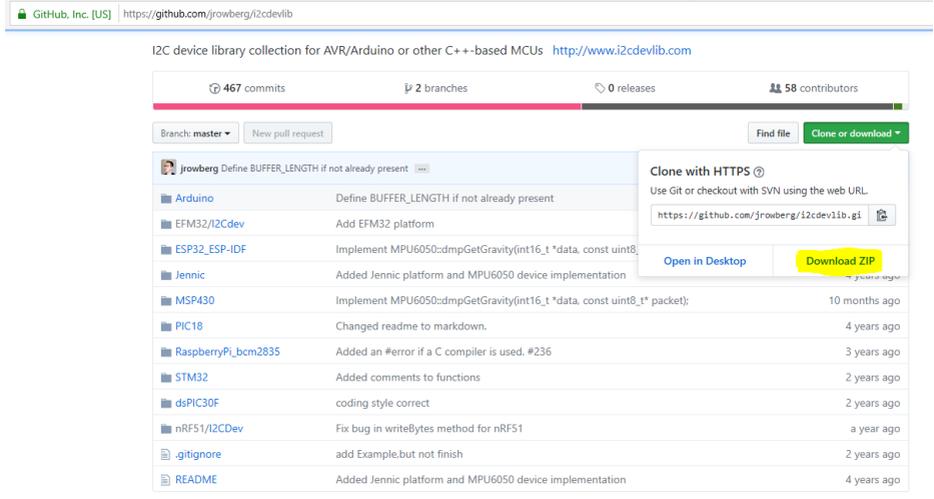


Die oben gezeigte Verdrahtung ist i.d.R. ausreichend. Um jedoch alle Funktionen nutzen zu können ist es empfehlenswert zusätzlich den INT-Pin des Moduls an Pin 2 des Arduinos mit anzuschließen (Digital2, PWM, Interrupt).

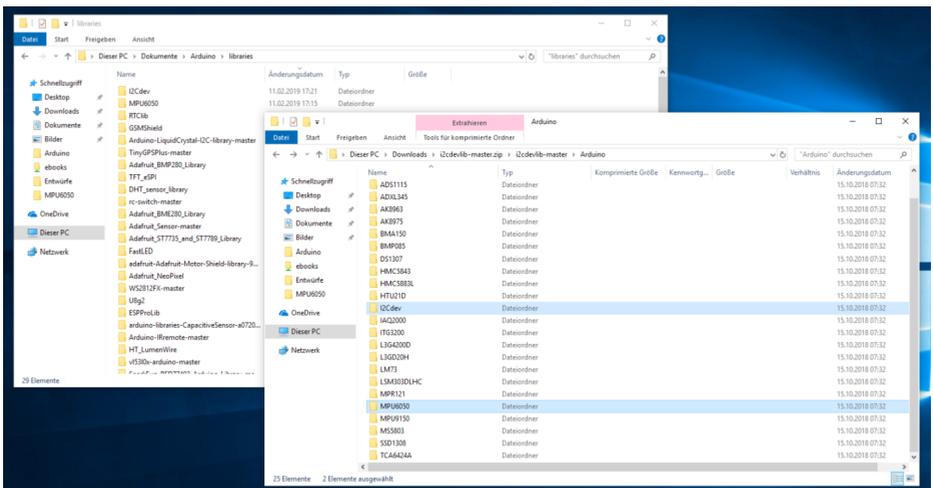
AD0 bietet die Option die I2C-Adresse über die Software zu ändern.

Die Installation der Libraries:

Libraries können auf verschiedene Arten in die ArduinoIDE importiert werden. Neben der Option die Libraries über den Boardverwalter oder einem Import über die Zip-Datei können wir diese auch direkt in das Arduino-Verzeichniss kopieren.



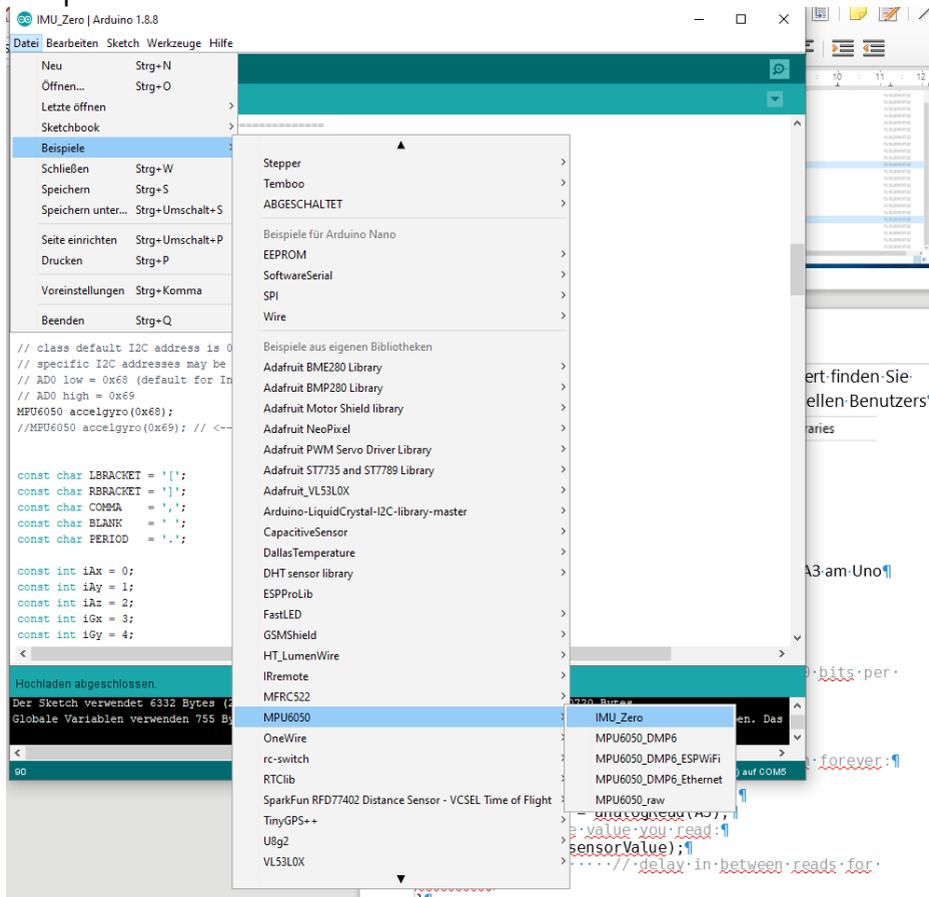
Nach dem [Download des GitHub-Verzeichnisses](#) entpacken wir die Zip-Datei und kopieren die darin enthaltenen Ordner „I2Cdev“ und „MPU6050“ in das Arduino-Library Verzeichniss, wie im Screenshot zu sehen:



Wurde die Arduino IDE über die Setup.exe installiert finden Sie den Ordner im Verzeichnis „Dokumente“ des aktuellen Benutzers

↑ > Dieser PC > Dokumente > Arduino > libraries

War das Kopieren der Dateien erfolgreich finden Sie nach einem Neustart der ArduinoIDE die Bibliothek MPU6050 in den Beispielen.



Das Auslesen der Sensordaten:

Zuerst ermitteln wir die Offset-Werte des Sensors. Dazu öffnen wir das Beispiel „IMU_Zero“ und passen Zeile 91 an, indem wir die I2C-Adresse hinzufügen:

```
// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
MPU6050 accelgyro(0x68);
//MPU6050 accelgyro(0x69); // <-- use for AD0 high
```

Im Anschluss platzieren wir den Sensor auf einem geraden, ruhigen und stabilen Untergrund.

Nach dem kompilieren und übertragen des Sketches auf den Controller öffnen wir den seriellen Monitor mit 9600 Baud:

 COM5

```
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful
averaging 1000 readings each time
expanding:
..
```

Nach der Initialisierung des Sensors dauert es etwas, bis der Sketch komplett durchgelaufen ist. Dieser Vorgang kann mehrere Minuten dauern.

Ist der Sketch durchgelaufen steht am Ende der Ausgabe:

----- done -----

Wir notieren uns die letzten vier Werte in der letzten Zeile:

```
..... XAccel          YAccel          ZAccel          XGyro          YGyro          ZGyro
[-1539,-1538] --> [-10,13] [57,58] --> [-34,9] [945,946] --> [16373,16398] [-6,-5] --> [0,3] [64,65] --> [0,5] [45,46] --> [-2,3]
..... [-1539,-1538] --> [-12,13] [57,58] --> [-45,9] [945,946] --> [16369,16398] [-6,-5] --> [0,3] [64,65] --> [0,5] [45,46] --> [-2,3]
----- done -----
```

```
          ZAccel          XGyro          YGyro          ZGyro
[16373,16398] [-6,-5] --> [0,3] [64,65] --> [0,5] [45,46] --> [-2,3]
[945,946] --> [16369,16398] [-6,-5] --> [0,3] [64,65] --> [0,5] [45,46] --> [-2,3]
```

Jetzt können wir die Offsets in unseren Sketch eintragen, um ein stabiles Messergebnis zu erhalten. Dazu öffnen wir den mitgelieferten Beispielsketch „MPU6050_DMP6“ passen in Zeile 62 wieder mit unserer I2C-Adresse an und springen nach Zeile 203:

```
// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(-6);
mpu.setYGyroOffset(64);
mpu.setZGyroOffset(45);
mpu.setZAccelOffset(945); // 1688 factory default for my test chip

// make sure it worked (returns 0 if so)
if (devStatus == 0) {
  // turn on the DMP, now that it's ready
  Serial.println(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);
```

Nach dem kompilieren können wir den Sketch übertragen. Im Gegensatz zum „IMU_Zero“ ist die Baudrate in diesem Sketch auf 11500 festgelegt.

Du hast es geschafft! Herzlichen Glückwunsch!

Ab jetzt heißt es lernen und ausprobieren. Du weißt nun wie ein Mikrocontroller Bewegungen erfassen kann. Jetzt kannst du versuchen die Werte praktisch einzusetzen. Wie wäre es mit einem selbst balancierenden Roboter oder einem 3D-Eingabe-Gerät?.

Diesen Sensor und noch mehr Hardware findest du natürlich in deinem Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>