Raspberry Pi Kompass selber bauen (HMC5883L)

Neben der Nutzung als GPS Ortungssystem kann man mit dem HMC5883L Modul auch einen Raspberry Pi Kompass bauen. Dieser gibt jeweils den Winkel, in dem er sich gerade befindet an. Das HMC5883L Kompass-Modul kann per I2C angesprochen werden, wie viele andere Sensoren auch. Zusammen mit dem Rotationssensor lassen sich komplette Lage- und Positionssysteme bauen.

In diesem Tutorial wollen wir also unseren eigenen elektronischen Raspberry Pi Kompass konfigurieren, sodass er immer die Abweichung (in Grad) zum Norden hin anzeigt.

Kompass Bauteile

In diesem Tutorial habe ich folgende Bauteile benutzt:

- Raspberry Pi 3*
- HMC5883L* / GY-271
- Female-Female Jumper Kabel*

Raspberry Pi Kompass Modul HMC5883L

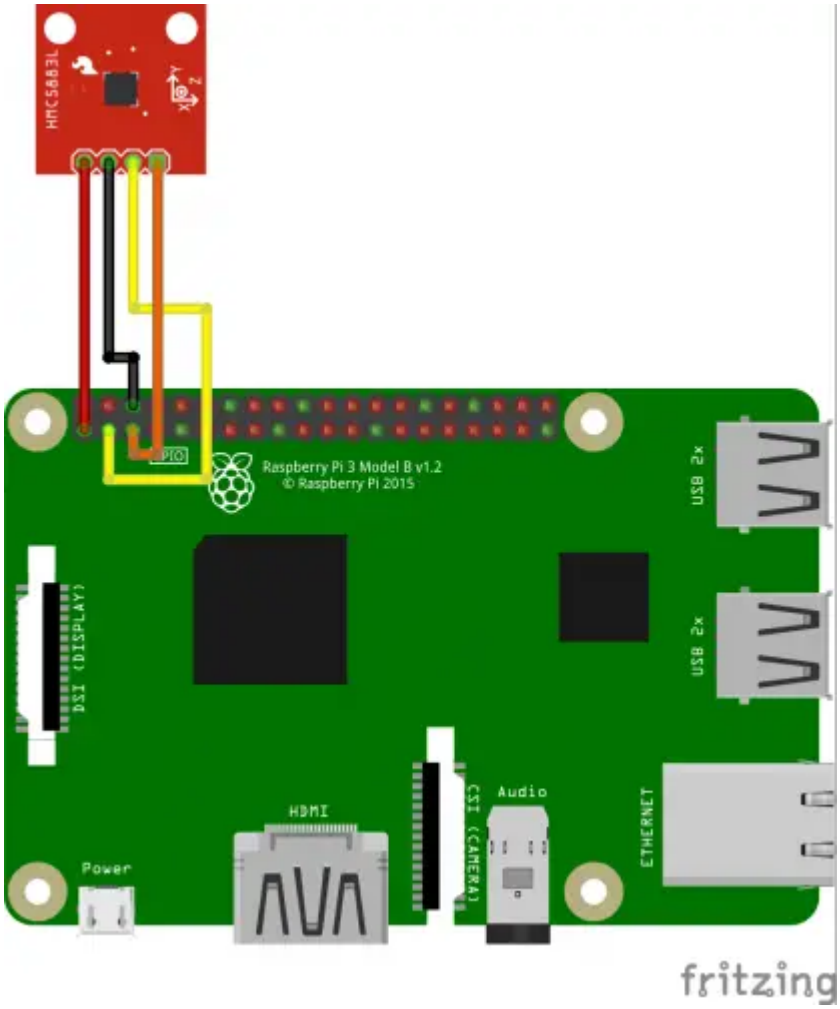
Zur Visualisierung kann z.B. ein Servo Motor genutzt werden, der in die entsprechende Richtung zeigt und so die Kompassnadel nachstellt. Ich habe so etwas ähnliches bereits einmal mit einem Xbox 360 Controller gebaut, was als Inspiration dienen könnte.

Raspberry Pi Kompass Anschluss

Das HMC5883L Modul hat i.d.R. fünf Anschlüsse, wovon wir allerdings nur vier benötigen. Der „DRDY“ bzw. „RDY“ Abschluss bleibt frei.

Die restlichen vier Pins des Sensors werden wie folgt an den Pi angeschlossen: VCC -> 3.3V (Pin 1), GND -> GND (Pin 6), SCL -> GPIO3 (Pin 5), SDA -> GPIO2 (Pin 3). Manche Sensoren haben sechs Pins. Falls du einen solchen (mit 3vo und VIN) hast, werden beide an den 3.3V Pin des Pi’s angeschlossen.

Eine Übersicht der Pin Belegung am Raspberry Pi findest du hier. Schematisch sieht die Verbindung zum Raspberry Pi folgendermaßen aus:



Achte auf die Beschriftung der Sensorpins!

Hinweis: Da die Frage des Öfteren vorkam, möchte ich darauf hinweisen, dass man auch mehrere I²C Geräte gleichzeitig am Raspberry Pi betreiben kann. Dazu werden die Anschlüsse (SDA, SCL) parallel angeschlossen. Die Erkennung des Moduls erfolgt über die feste Hardware Adresse (mehr dazu unten). Es können nur nicht zwei Geräte mit identischer I2C Adresse betrieben werden. Bei vielen Sensoren kann man aber durch Jumper Kabel auf der Platine die Adresse ändern. Ist so etwas gewünscht, empfehle ich einen Blick ins Datenblatt.

Raspberry Pi Kompass – Vorbereitung

Bevor wir den Kompass nutzen können, werden einige Softwarepakete benötigt. Unter anderem muss der git Client, die I2C Tools sowie Python3 intalliert sein. Dazu öffnen wir das SSH Terminal und geben folgendes ein:

```
sudo apt-get install git i2c-tools python-smbus python3 python-pip python-virtua
lenv python3-setuptools
```

Nach der erfolgreichen Installation muss I2C noch freigeschaltet werden, falls nicht bereits geschehen. Gib dazu in der Kommandozeile ein:

```
sudo raspi-config
```

Unter „Advanced Options“ > „I2C“ aktivierst du es.

Sofern das Kompassmodul bereits angeschlossen ist, können wir bereits schauen, ob er erkannt wird. Gib dazu sudo i2cdetect -y 1 ein. Die Ausgabe sollte wie folgt aussehen:

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Falls nichts erkannt wurde, solltest du die Verbindung des Sensors überprüfen.

Normalerweise müssen Root-Rechte zum Auslesen des I^2C Bus vorhanden sein. Da die später genutzte Bibliothek diese aber nicht hat, müssen wir dies erst freischalten. Dazu erstellen wir eine Datei

```
sudo nano /etc/udev/rules.d/99-i2c.rules
```

mit folgendem Inhalt:

```
SUBSYSTEM=="i2c-dev", MODE="0666"
```

Mit STRG+O speichern wir diese und mit STRG+X beenden wir den Editor und kehren zurück ins Terminal.

Raspberry Pi Kompass Bibliothek (HMC5883L) einrichten

Um das Kompass Modul nutzen zu können, benötigen wir zwei Python 3 Bibliotheken, die wir nun installieren:

```
git clone https://github.com/quick2wire/quick2wire-python-api
cd quick2wire-python-api
```

Hierin befinden sich einige Dateien, die wir der Python-Pfad Variablen hinzufügen müssen. Solltest du den Ordner umbenannt oder in einem anderen Verzeichnis haben, kannst du dir mit `pwd` anzeigen lassen, in welchem Ordner du dich befindest. Um den Python-Pfad zu aktualisieren, bearbeiten wir die Profiles-Datei:

```
sudo nano /etc/profile
```

Ganz oben fügen diese beiden Zeilen hinzu (Pfad ggf. anpassen):

```
export QUICK2WIRE_API_HOME=/home/pi/quick2wire-python-api
export PYTHONPATH=$PYTHONPATH:$QUICK2WIRE_API_HOME
```

Nach dem Speichern der Datei starten wir den Pi neu.

```
sudo reboot
```

Nachdem wir uns wieder per SSH verbinden haben, überprüfen wir noch schnell, ob der Pfad nun gesetzt wurde:

```
echo $PYTHONPATH
```

Dieser sollte nun unseren angegebenen Pfad enthalten. Außerdem sollte ein I2C Gerät erkannt werden (Kompassmodul muss angeschlossen sein):

```
ls /dev/i2c-*
```

Die angezeigte Nummer musst du dir merken, da sie gleich gebraucht wird (bei mir ist es 1).

Zurück im Verzeichnis der Bibliothek können wir nun die Installation starten:

```
cd quick2wire-python-api
sudo python3 setup.py install
```

Nun können wir auch endlich die eigentliche Bibliothek installieren, welche den HMC5883L Kompass auslesen kann.

```
git clone https://bitbucket.org/thinkbowl/i2clibraries.git
```

Hier sind auch weitere I2C Bibliotheken wie u.a. für den ITG-3205, ADXL345 und LCD Displays vorhanden.

Falls du möchtest, kannst du den Pfad der Bibliotheken auch der Pythonpfad-Variable hinzufügen (analog zum oben gezeigtem).

Raspberry Pi Kompass testen

Nun wird es Zeit, dass wir den elektrischen Kompass testen. Dazu rufen wir die Python3 Konsole auf:

```
sudo python3
```

Im folgenden sind einige Beispielcodes, die die Funktionen beschreiben. Sollte dein I2C Gerät nicht die Nummer 1 haben (wie bei mir), Musst du dies in Zeile 3 anpassen.

```
1 from i2clibraries import i2c_hmc5883l
2
3 hmc5883l = i2c_hmc5883l.i2c_hmc5883l(1)
4
5 hmc5883l.setContinuousMode()
6 hmc5883l.setDeclination(2, 15)
7
8 print(hmc5883l)
```

Die Deklination solltest du für höchstmögliche Genauigkeit setzen (Zeile 6). Bei mir beträgt diese 2° und 15′ (Bogeminuten). Die „Magnetic Declination" ist an jedem Ort unterschiedlich und kann z.B. über diese Seite herausgefunden werden.

In dem Beispiel interessiert uns der letzte Wert, welcher auch über hmc5883l.getHeadingString() ausgelesen werden kann. Ein Wert von 0 bedeutet, dass der Sensor gerade in Richtung Norden zeigt.

Axen auslesen

Neben dem Winkel ist es auch möglich die Rotation der Axen auszulesen:

```
1 from i2clibraries import i2c_hmc5883l
2
3 hmc5883l = i2c_hmc5883l.i2c_hmc5883l(1)
4
5 hmc5883l.setContinuousMode()
6
7 # To scaled axes
8 (x, y, z) = hmc5883l.getAxes()
```

Leider ist die Website mit der kompletten Dokumentation inzwischen offline, aber kann glücklicherweise noch über die Wayback Machine aufgerufen werden.

GPS Kompass Navigation

* Einige Links können Affiliate-Links sein. Wir erhalten als Amazon-Partner u.U. eine Provision, wenn Sie etwas kaufen, nachdem Sie auf einen dieser Links auf unserer Website geklickt haben

VORHERIGER BEITRAG
13 tolle Raspberry Pi Projekte für Kinder und Jugendliche

NÄCHSTER BEITRAG
Expand Raspberry Pi GPIOs with I2C Port Expander

ÄHNLICHE BEITRÄGE

ESP32 TDS Sensor: Wasserqualität im Smart Home messen

ESP32 pH-Sensor – Automatische pH-Messung für Pool und Hydroponik

Home Assistant auf dem Raspberry Pi – Erste Schritte

How-To: Bluetooth Verbindung Zwischen ESP32 und Raspberry Pi's

4 KOMMENTARE

Kurt Mühlemann am 2. November 2020 10:20

Lieber Felix

Ist diese Seite so neu oder findet sie so wenig Interessenten, dass ich den ersten Kommentar schreiben darf? Ich habe das Tutorial nachvollzogen und musste zuerst beim Befehl „pi@raspberrypi:~ $ sudo i2cdetect -y 1“ leer schlucken, denn die gefundene Adresse lautet bei mir Hex 0d statt Hex 1e.
Meine Recherche auf dem Internet zeigte, dass das häufig vorkommt, weil über Ebay und Amazon Magnetsensoren unter dem Namen „HMC5883L / GY-271“ vertrieben werden, die nicht den HMC5883L-Chip sondern den chinesischen QMC5883L-Chip enthalten. Der echte (Honeywell) Chip ist mit L883 bezeichnet, der chinesische (unter Honeywell Lizenz) mit 5883.
Die Unterschiede der beiden Chips sind unter https://belchip.by/sitedocs/10882.pdf zusammengefasst. Für die Software zum Modul mit dem QMC5883L verweise ich auf https://github.com/RigacciOrg/py-qmc5883l
Viel Vergnügen beim Experimentieren mit Eurem elektronischen Kompass!

ANTWORTEN >

Heinz am 25. Januar 2021 11:11

Hallo Felix,
mein Robotorauto würde ich gern mit einem Kompassmodul bestücken. Das Modul kann ich dank Deines tollen Tutorials ansprechen, leider steigt es in Zeile 3 mit folgender Fehlermeldung aus. Ich wäre dankbar wenn Sie mir weiterhelfen würden.
Mit freundlichen Grüßen

pi@raspberrypi:~$ python3 ko.py
Traceback (most recent call last):
File ko.py“, line 3, in
hmc58831 = i2c_hmc58831.i2c_hmc58831 (1)
file „/home/pi/quick2wire-pyhton-api/i2clibraries/i2c_hmc 58831.py“ line 29, in _init_
self.setScale(gaus)
file „/home/pi/quick2wire-pyhton-api/i2clibraries/i2c_hmc 58831.py“ line 76, in setScale
self.setOption(self.ConfigurationRegisterB, self.scale_reg)

file „/home/pi/quick2wire-pyhton-api/i2clibraries/i2c_hmc 58831.py“ line 87, in setOption
self.bus.write_byte(register, options)
file „/home/pi/quick2wire-pyhton-api/i2clibraries/i2c.py“ line 14, in write_byte
writing_bytes(self.addr, *bytes))
file „/home/pi/quick2wire-pyhton-api/i2clibraries/i2c.py“ line 78, in transaction
ioctl(self.fd, I2C_RDWR, ioctl_arg)

eingesetztes Modul: HMC5883L/GY-271

ANTWORTEN >

Heinz am 28. Januar 2021 21:21

Leider bekomme ich nie Rückmeldungen, mache ich etwas verkehrt?

ANTWORTEN >

Andreas am 15. Februar 2021 23:14

Hallo Heinz,

deine Zeile 3:
hmc58831 = i2c_hmc58831.i2c_hmc58831 (1)

Den Fehler habe ich auch gemacht, es ist keine „1“, es ist ein „l“, also ein kleines „L“
hmc5883l = i2c_hmc5883l.i2c_hmc5883l (1)

VGA

ANTWORTEN >

HINTERLASSE EINEN KOMMENTAR

Dein Kommentar

Name *

Email *

Meinen Namen, E-Mail und Website in diesem Browser speichern, bis ich das nächste Mal kommentiere.

Benachrichtige mich über nachfolgende Kommentare via E-Mail.

Benachrichtige mich über neue Beiträge via E-Mail.

KOMMENTAR ABSCHICKEN

Raspberry Pi Samba Server installieren: Dateien im lokalen Netzwerk teilen

Raspberry Pi als AirPlay-Empfänger verwenden

Arduino und Raspberry Pi miteinander kommunizieren lassen

Raspberry Pi + Android/iOS: Funksteckdosen per App schalten

Durchflussmesser / Water Flow Sensor am Raspberry Pi auslesen

Amazon Alexa (Deutsch) auf dem Raspberry Pi installieren

Blog abonnieren

Abboniere Raspberry Pi Tutorials, um kein Tutorial mehr zu verpassen!

E-Mail-Adresse

ABONNIEREN

TUTORIALS FOR RASPBERRY PI

KONTAKT & IMPRESSUM UNTERSTÜTZEN DATENSCHUTZ