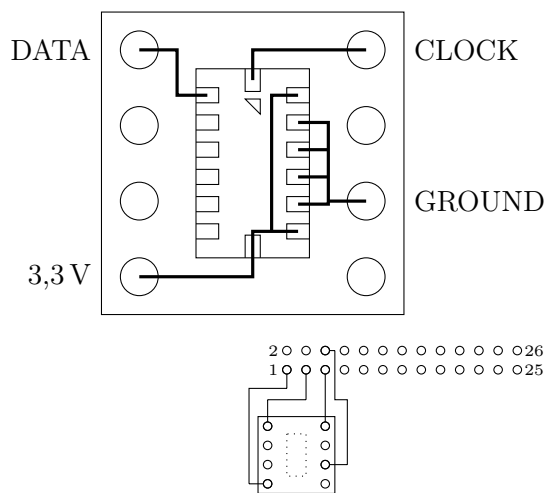


Beschleunigungssensor am GPIO

Überall auf der Erde wird man durch die Anziehungskraft zum Erdmittelpunkt hin beschleunigt. Diese Kraft lässt sich messen. Wenn man dieses in alle drei Raumrichtungen gleichzeitig macht, kann man damit auch die Lage eines Elementes zur Erdoberfläche bestimmen. Um solche Messungen mit dem Raspberry Pi durchführen zu können eignet sich der Chip MMA7455 sehr gut. Es ist aber zu beachten, dass dieser Chip mit den Abmessungen 5 x 3 x 1 mm sehr klein ist. Daher bietet es sich an, ihn auf eine 4 x 4 Lochrasterplatine zu kleben und die notwendigen Anschlüsse mit einer Stiftleiste zu verbinden, so dass diese Kombination in Verbindung mit einem Steckbrett genutzt werden kann. Die Anschlüsse am Chip und die Verbindung mit dem GPIO sind den beiden nachfolgenden Zeichnungen zu entnehmen. Mit etwas Warmklebe kann dann zusätzlich verhindert werden, dass es zu Kontakten zwischen den dünnen Drähten kommt.



Treiber laden

Der MMA7455 wird über das I^2C -Protokoll angesprochen. Der Raspberry Pi unterstützt dieses Protokoll, nachdem die entsprechenden Treiber als Kernelmodule geladen sind. Damit dieses dauerhaft der Fall ist muss in der Datei `/etc/modprobe.d/raspi-blacklist.conf` die Zeile mit `i2c-bcm2708` durch eine Raute auskommentiert werden. Außerdem muss in die Datei `/etc/modules` eine Zeile mit `i2c-dev` hinzugefügt werden. Nach einem Neustart des Raspberry ist dann alles notwendige an Treibern vorhanden. Alternativ können beide Module als Superuser mit dem Befehl `modprobe` direkt geladen werden.

Auslesen mit Python

Damit Python alle nötigen Befehle des I^2C -Protokolls versteht müssen erst noch zwei Pakete installiert werden. Dieses geht am einfachsten mit `sudo apt-get install i2c-tools python-smbus`.



Ein Python-Programm, das die Werte aller drei Achsen ausliest und darstellt wurde in der vierten Ausgabe von The MagPi vorgestellt, die unter <http://www.themagpi.com/en/issue/4> erhältlich ist.



Der Quelltext dieses Programms ist hier noch einmal abgedruckt:

```
1 #!/usr/bin/python
2 import smbus
3 import time
4 import os
5 # import graphics
6 # Define a class called Accel
7 class Accel():
8     b = smbus.SMBus(1)
9     def setUp(self):
10         # Setup the Mode
11         self.b.write_byte_data(0x1D,0x16,0x55)
12         # Setup Calibration
13         self.b.write_byte_data(0x1D,0x10,0)
14         self.b.write_byte_data(0x1D,0x11,0)
15         self.b.write_byte_data(0x1D,0x12,0)
16         self.b.write_byte_data(0x1D,0x13,0)
17         self.b.write_byte_data(0x1D,0x14,0)
18         self.b.write_byte_data(0x1D,0x15,0)
19
20     def getValueX(self):
21         return self.b.read_byte_data(0x1D,0x06)
22
23     def getValueY(self):
24         return self.b.read_byte_data(0x1D,0x07)
25
26     def getValueZ(self):
27         return self.b.read_byte_data(0x1D,0x08)
28
29 MMA7455 = Accel()
30 MMA7455.setUp()
31
32 for a in range(10000):
33     x = MMA7455.getValueX()
34     y = MMA7455.getValueY()
35     z = MMA7455.getValueZ()
36
37     print "x=", x , "\t[" , "=" * (80 * x / 256) , "]" * (80 - (80 * x
38         / 256)) , "]" , (80 * x / 256) , "%"
39     print "y=", y , "\t[" , "=" * (80 * y / 256) , "]" * (80 - (80 * y
40         / 256)) , "]" , (80 * y / 256) , "%"
41     print "z=", z , "\t[" , "=" * (80 * z / 256) , "]" * (80 - (80 * z
42         / 256)) , "]" , (80 * z / 256) , "%"
43
44     time.sleep(0.1)
45     os.system('clear')
```

