



Hochschule  
Kaiserslautern  
University of  
Applied Sciences

## Ergänzung zu Bedienungsanleitung Fridolin V 1.1



me.Jürgen Edel  
Hochschule Kaiserslautern  
Campus Pirmasens  
Carl-Schurz-Straße 10-16  
66953 Pirmasens

email: [juergen.edel@hs-kl.de](mailto:juergen.edel@hs-kl.de)

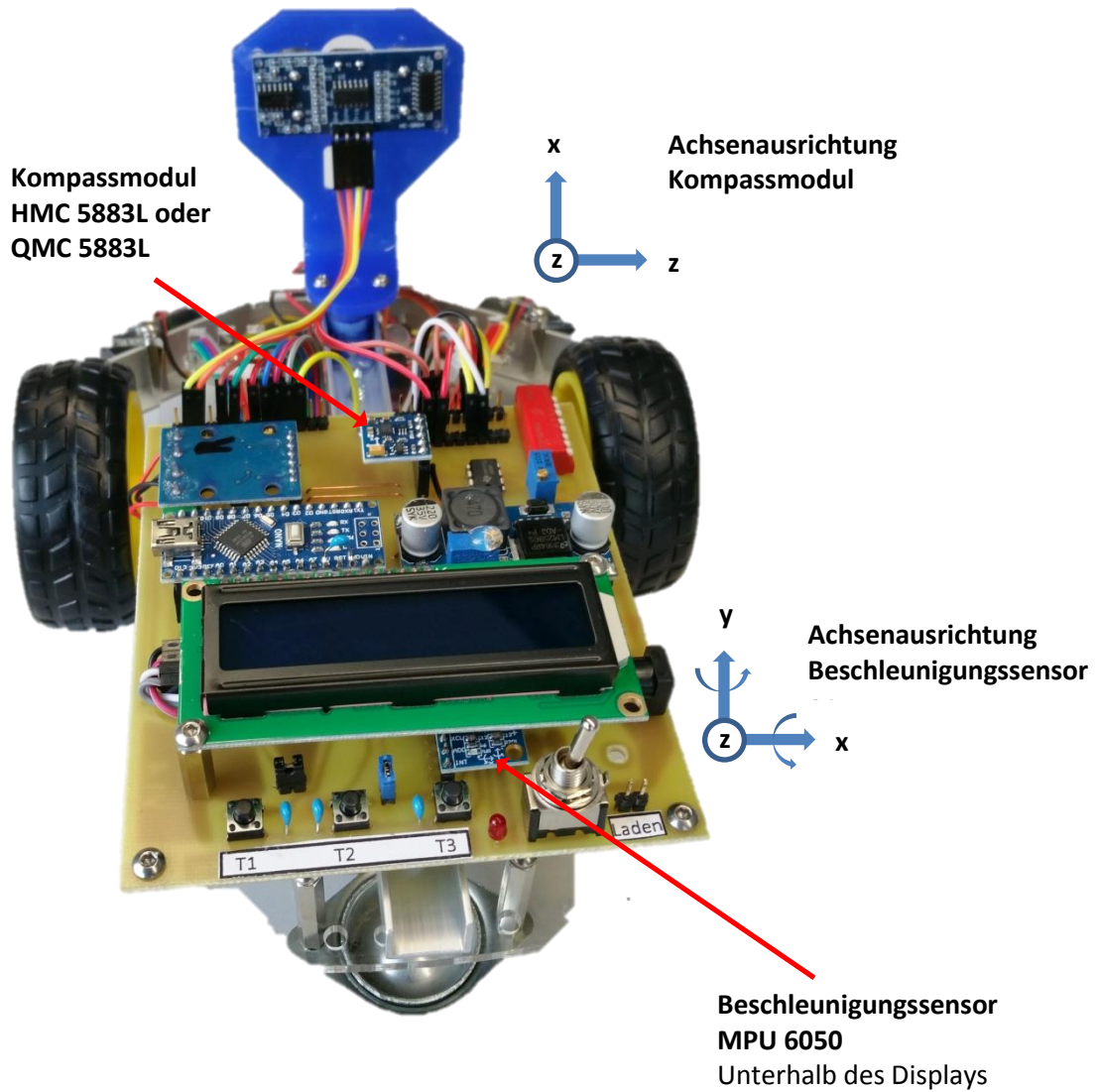
Idee und Umsetzung Jürgen Edel – Autor Jürgen Edel

## Inhalt

Erweiterung Fridolin .....	2
Modul-Beschreibung .....	2
HMC5883L Programmcode .....	2
QMC5883L Programmcode .....	2
MPU6050 Programmcode .....	2

## Erweiterung Fridolin

### Erweiterung Fridolin V1.3 bis V1.4



## Modul-Beschreibung

Kompassmodul GY 271

HMC 5883L

QMC 5883L



DRDY ( nicht belegt)  
SDA (Datenleitung)  
SCL (Taktsignal)  
Masse (GND)  
VCC 5 Volt

Bei dem Kompassmodul handelt es sich um einen HMC 5883L (GY 271) oder OMC 5883L (GY 271)

Der Sensor kann die magnetische Flussdichte von Magnetfeldern in 3Achsen messen. Der Anschluss an den Arduino erfolgt über den I2C Bus.

Bei welchem Fridolin welches Modul verbaut wurde ist in der Übersichtstabelle ersichtlich.  
**Die Übersichtstabelle sowie die Datenblätter der Module finden Sie auf dem BSCW Server in dem Ordner Fridolin-Dokumente**

Beschleunigungssensor GY 521

MPU 6050

VCC 5 Volt  
Masse /GND)  
SCL (Taktsignal)  
SDA (Datenleitung)  
Nicht belegt  
Nicht belegt  
Nicht belegt  
Nicht belegt



Der Beschleunigungssensor ist ein 6 -Achsiges I2C Prozessor der über ein 3-Achsen-Gyroskop und ein 3-Achsen Beschleunigungsmesser verfügt. In allen Fridolin ist das gleiche Modul verbaut. MPU 6050 (GY-521)

Testprogramme zum Auslesen der Rohdaten sind beigefügt.  
**Die gleichen Programme stehen auch als Download im BSCW Server bereit unter Fridolin Programme\_Bibliotheken\_Treiber**

MPU6050\_Rohdaten.ino  
HMC5883L\_Rohdaten.ino  
QMC5883L\_Rohdaten.ino

## HMC5883L Programmcode

```
/*
 * Programm zum Auslesen der Rohdaten
 * für den HMC 5883L Kompassensor
 * Die Rohwerte der Achsen können über den seriellen Monitor
 * ausgegeben werden
 * Die Baud Rate des seriellen Monitor ist 9600
 */

#include<Wire.h> // Bibliothek TWI Bus einbinden

const byte AdresseHMC = 0x1E; // Busadresse des HMC
const byte HMCRegister = 0x02; // Mode Register
const byte HMCMessung = 0x00; // Dauermessung
const byte datenAusgang = 0x03; // Daten Ausgang MSB Register Adresse

// Variable festlegen
int x=0; // x-Achse
int y=0; // y-Achse
int z=0; // z-Achse
int winkel=0; // Winkelangabe
int gerade=0;
int sub=0;

void setup()
{
  Serial.begin(9600); // Serielle Ausgabe über Monitor starten
  Wire.begin(); // Buskommunikation starten
  Wire.beginTransmission(AdresseHMC); // Kommunikatio mit HMC starten
  Wire.write(HMCRegister); // Modus Register wählen
  Wire.write(HMCMessung); // Dauerhafte Messung starten
  Wire.endTransmission(); // Kommunikation beenden
}
void loop()
{
  // Einlesen der Daten starten
  Wire.beginTransmission(AdresseHMC); // Buskommunikation starten
  Wire.write(datenAusgang); // wähle MSB Register
  Wire.endTransmission(); // Kommunikation beenden

  // Alle Achsen einlesen
  Wire.requestFrom(AdresseHMC,6);
  if(6<=Wire.available())
  {
    x=Wire.read()<<8; // x-msb
    x|=Wire.read(); // x-lsb
    z=Wire.read()<<8; // y-msb
```

```
z|=Wire.read(); // y-lsb
y=Wire.read()<<8; // z-msb
y|=Wire.read(); // z-lsb
}
// Werte der Achsen über Seriellen Monitor ausgeben
Serial.print("xachse: ");
Serial.print(x);
Serial.print(" yachse: ");
Serial.print(y);
Serial.print(" zachse: ");
Serial.println(z);

delay(250);

}
```

## QMC5883L Programmcode

```
/*
 * Programm zum Auslesen der Rohdaten
 * für den QMC 5883L Kompassensensor
 * Die Rohwerte der Achsen können über den seriellen Monitor
 * ausgegeben werden
 * Die Baud Rate des seriellen Monitor ist 9600
 */

#include <Wire.h> //I2C Arduino Library

#define AdresseQMC 0x0D //I2C Adresse QMC
int x=0; //x achse
int y=0; //y achse
int z=0; //z achse

void setup()
{
  Serial.begin(9600);          // Serielle Ausgabe über Monitor starten
  Wire.begin();               // Buskommunikation starten

  Wire.beginTransmission(AdresseQMC); // Kommunikation mit QMC starten
  Wire.write(0x0B);           // Dauerhafte Messung
  Wire.write(0x01);           // Dauerhafte Messung
  Wire.endTransmission();
  Wire.beginTransmission(AdresseQMC);
  Wire.write(0x09);           // Mode Register wählen
  Wire.write(0x1D);           // Register setzen
  Wire.endTransmission();    // Kommunikation beenden
}

void loop()
{
  // Einlesen der Daten starten
  Wire.beginTransmission(AdresseQMC);
  Wire.write(0x00); //Datenausgabe msb Register
  Wire.endTransmission();

  //Alle Achsen einlesen
  Wire.requestFrom(AdresseQMC, 6);
  if (6 <= Wire.available()) {
    x = Wire.read(); //MSB
    x |= Wire.read() << 8; //LSB
    y = Wire.read(); //MSB
    y |= Wire.read() << 8; //LSB
    z = Wire.read(); //MSB
    z |= Wire.read() << 8; //LSB
  }
}

Jürgen Edel
Hochschule Kaiserslautern
```

```
// Werte der Achsen über Seriellen Monitor ausgeben
Serial.print("xachse: ");
Serial.print(x);
Serial.print(" yachse: ");
Serial.print(y);
Serial.print(" zachse: ");
Serial.println(z);
delay(250);
}
```



## MPU6050 Programmcode

```
/*
 * Programm zum Auslesen der Rohdaten
 * für den MPU6050 Beschleunigungssensor
 * Die Rohwerte der Achsen können über den seriellen Monitor
 * ausgegeben werden
 * Die Baud Rate des seriellen Monitor ist 9600
 */

#include<Wire.h>

//Adresse Beschleunigungssensor (MPU6050) GY 571
const int MPU=0x68;

//Variable
int ax; // beschleunigung x achse
int ay; // beschleunigung y achse
int az; // beschleunigung z achse
int gx; // kreisel x achse
int gy; // kreisel y achse
int gz; // kreisel z achse
int temp; // temperatur

//----- Setup -----
void setup()
{
    Serial.begin(9600); //Serieller Monitor sarten
    Wire.begin(); //I2C Bus sarten
    Wire.beginTransmission(MPU); //MPU Datenübertragung starten
    Wire.write(0x6B); //Register setzen

    //Ini MPU-6050
    Wire.write(0);
    Wire.endTransmission(true);
}

void loop()
{
    Wire.beginTransmission(MPU); // Datenübertragung starten
    Wire.write(0x3B); // Register setzen
    Wire.endTransmission(false); // Datenübertragung stoppen

    Wire.requestFrom(MPU,14,true);
    // Achsen auslesen
    ax=Wire.read()<<8|Wire.read(); //0x3B
    ay=Wire.read()<<8|Wire.read(); //0x3D
}

Jürgen Edel
Hochschule Kaiserslautern
```

```
az=Wire.read()<<8|Wire.read(); //0x3F
temp=Wire.read()<<8|Wire.read(); //0x41
gx=Wire.read()<<8|Wire.read(); //0x43
gy=Wire.read()<<8|Wire.read(); //0x45
gz=Wire.read()<<8|Wire.read(); //0x47

Serial.print("Achse:");
Serial.print("\tX: ");Serial.print(ax);
Serial.print("\tY: ");Serial.print(ay);
Serial.print("\tZ: ");Serial.print(az);

Serial.print("\tKreisel:");
Serial.print("\tX: ");Serial.print(gx);
Serial.print("\tY: ");Serial.print(gy);
Serial.print("\tZ: ");Serial.print(gz);

Serial.print("\tTemperatur: "); Serial.println(temp/340.00+36.53);

delay(250);
}
```