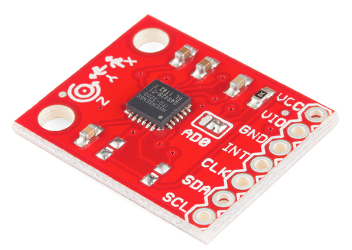
# Arduino Gyrosensor

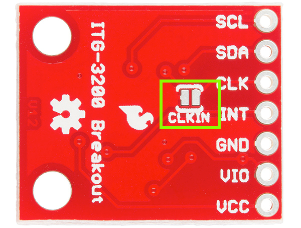
(written by Nam Pham, 2017)

Ein Gyrosensor ist ein Beschleunigungs- oder Lagesensor, der auf kleinste Beschleunigungen, Drehbewegungen oder Lageänderungen reagiert.

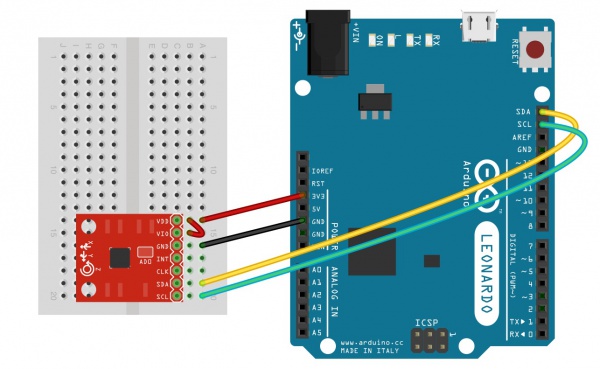
Der Gyrosensor ITG-3200 braucht Strom zwischen **2.1 and 3.6V**.



close this jumper with solder if you’re NOT using an external clock source.



Anschliessen des Gyrosensors an Arduino



//The Wire library is used for I2C communication

#include <Wire.h>

//This is a list of registers in the ITG-3200. Registers are parameters that determine how the sensor will behave, or they can hold data that represent the

//sensors current status.

//To learn more about the registers on the ITG-3200, download and read the datasheet.

char WHO\_AM\_I = 0x00;

char SMPLRT\_DIV= 0x15;

char DLPF\_FS = 0x16;

char GYRO\_XOUT\_H = 0x1D;

char GYRO\_XOUT\_L = 0x1E;

char GYRO\_YOUT\_H = 0x1F;

char GYRO\_YOUT\_L = 0x20;

char GYRO\_ZOUT\_H = 0x21;

char GYRO\_ZOUT\_L = 0x22;

//This is a list of settings that can be loaded into the registers.

//DLPF, Full Scale Register Bits

//FS\_SEL must be set to 3 for proper operation

//Set DLPF\_CFG to 3 for 1kHz Fint and 42 Hz Low Pass Filter

char DLPF\_CFG\_0 = 1<<0;

char DLPF\_CFG\_1 = 1<<1;

char DLPF\_CFG\_2 = 1<<2;

char DLPF\_FS\_SEL\_0 = 1<<3;

char DLPF\_FS\_SEL\_1 = 1<<4;

//I2C devices each have an address. The address is defined in the datasheet for the device. The ITG-3200 breakout board can have different address depending on how

//the jumper on top of the board is configured. By default, the jumper is connected to the VDD pin. When the jumper is connected to the VDD pin the I2C address

//is 0x69.

char itgAddress = 0x69;

//In the setup section of the sketch the serial port will be configured, the i2c communication will be initialized, and the itg-3200 will be configured.

void setup()

{

//Create a serial connection using a 9600bps baud rate.

Serial.begin(9600);

//Initialize the I2C communication. This will set the Arduino up as the 'Master' device.

Wire.begin();

//Read the WHO\_AM\_I register and print the result

char id=0;

id = itgRead(itgAddress, 0x00);

Serial.print("ID: ");

Serial.println(id, HEX);

//Configure the gyroscope

//Set the gyroscope scale for the outputs to +/-2000 degrees per second

itgWrite(itgAddress, DLPF\_FS, (DLPF\_FS\_SEL\_0|DLPF\_FS\_SEL\_1|DLPF\_CFG\_0));

//Set the sample rate to 100 hz

itgWrite(itgAddress, SMPLRT\_DIV, 9);

}

//The loop section of the sketch will read the X,Y and Z output rates from the gyroscope and output them in the Serial Terminal

void loop()

{

//Create variables to hold the output rates.

int xRate, yRate, zRate;

//Read the x,y and z output rates from the gyroscope.

xRate = readX();

yRate = readY();

zRate = readZ();

//Print the output rates to the terminal, seperated by a TAB character.

Serial.print(xRate);

Serial.print('\t');

Serial.print(yRate);

Serial.print('\t');

Serial.println(zRate);

//Wait 10ms before reading the values again. (Remember, the output rate was set to 100hz and 1reading per 10ms = 100hz.)

delay(10);

}

//This function will write a value to a register on the itg-3200.

//Parameters:

// char address: The I2C address of the sensor. For the ITG-3200 breakout the address is 0x69.

// char registerAddress: The address of the register on the sensor that should be written to.

// char data: The value to be written to the specified register.

void itgWrite(char address, char registerAddress, char data)

{

//Initiate a communication sequence with the desired i2c device

Wire.beginTransmission(address);

//Tell the I2C address which register we are writing to

Wire.write(registerAddress);

//Send the value to write to the specified register

Wire.write(data);

//End the communication sequence

Wire.endTransmission();

}

//This function will read the data from a specified register on the ITG-3200 and return the value.

//Parameters:

// char address: The I2C address of the sensor. For the ITG-3200 breakout the address is 0x69.

// char registerAddress: The address of the register on the sensor that should be read

//Return:

// unsigned char: The value currently residing in the specified register

unsigned char itgRead(char address, char registerAddress)

{

//This variable will hold the contents read from the i2c device.

unsigned char data=0;

//Send the register address to be read.

Wire.beginTransmission(address);

//Send the Register Address

Wire.write(registerAddress);

//End the communication sequence.

Wire.endTransmission();

//Ask the I2C device for data

Wire.beginTransmission(address);

Wire.requestFrom(address, 1);

//Wait for a response from the I2C device

if(Wire.available()){

//Save the data sent from the I2C device

data = Wire.read();

}

//End the communication sequence.

Wire.endTransmission();

//Return the data read during the operation

return data;

}

//This function is used to read the X-Axis rate of the gyroscope. The function returns the ADC value from the Gyroscope

//NOTE: This value is NOT in degrees per second.

//Usage: int xRate = readX();

int readX(void)

{

int data=0;

data = itgRead(itgAddress, GYRO\_XOUT\_H)<<8;

data |= itgRead(itgAddress, GYRO\_XOUT\_L);

return data;

}

//This function is used to read the Y-Axis rate of the gyroscope. The function returns the ADC value from the Gyroscope

//NOTE: This value is NOT in degrees per second.

//Usage: int yRate = readY();

int readY(void)

{

int data=0;

data = itgRead(itgAddress, GYRO\_YOUT\_H)<<8;

data |= itgRead(itgAddress, GYRO\_YOUT\_L);

return data;

}

//This function is used to read the Z-Axis rate of the gyroscope. The function returns the ADC value from the Gyroscope

//NOTE: This value is NOT in degrees per second.

//Usage: int zRate = readZ();

int readZ(void)

{

int data=0;

data = itgRead(itgAddress, GYRO\_ZOUT\_H)<<8;

data |= itgRead(itgAddress, GYRO\_ZOUT\_L);

return data;

}