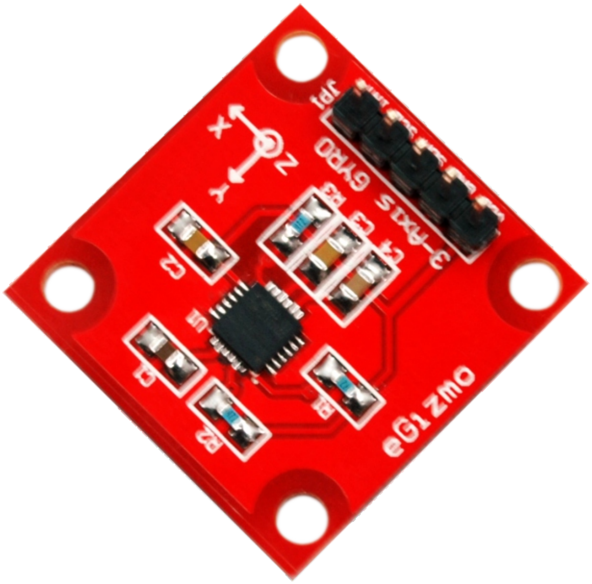
3-Axis GYRO

Hardware Manual Rev 1r0



Features and Specifications

The 3-Axis GYRO makes use of the ITG3205 which functions as a 3-axis MEMS gyro IC. It is frequently optimized for gaming, 3D mice, and 3D remote control applications. It has three 16-bit analog-to-digital converters for digitizing gyro outputs. Works in +3.3V logic.

Chip: ITG-3205
Power input: 2.1V to 3.6V
Size: 4x4x0.9mm

Pin Assignments:

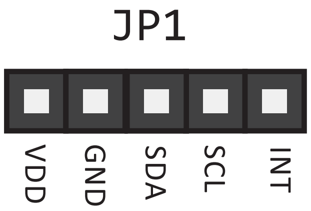


Figure 1. *JP1 Pin I.D. Illustration*

Table 1. *JP1 Pin I.D. & Descriptions*

Pin I.D.	Descriptions
VDD	+3.3V Supply Voltage
GND	Ground
SDA	I^2C serial data
SCL	I^2C serial clock
INT	Interrupt Digital Output

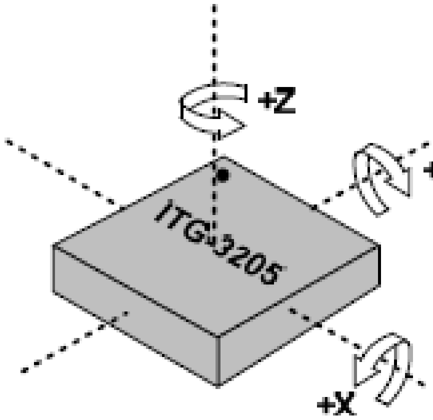


Figure 2. *Orientation of Axes and polarity of rotation*

Features & specifications

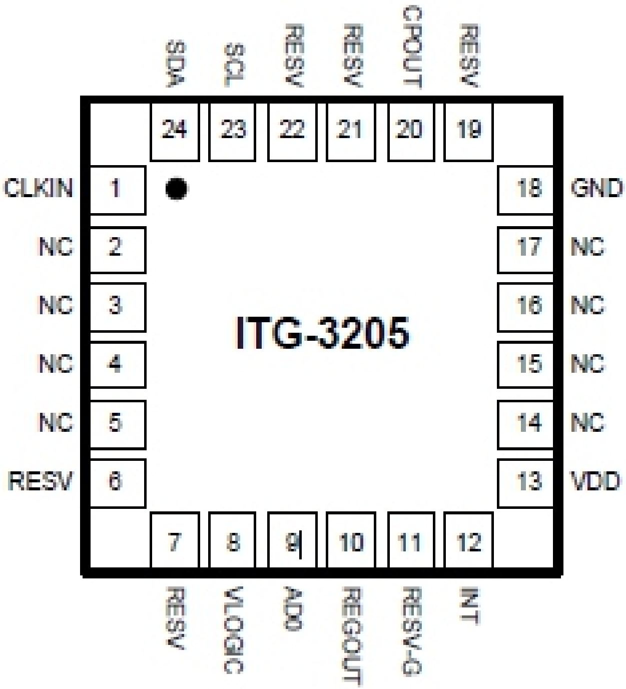


Figure 3. *ITG-3205 Pin I.D. Illustration*

Features

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyros) on one integrated circuit
- Three integrated 16-bit ADCs
- Fast Mode I2C (400kHz) serial interface
- Digitally-programmable low-pass filter

Table 2. *ITG-3205 I.D. & Descriptions*

Number	Pin	Pin Description
1	CLKIN	Optional external reference clock input. Connect to GND if unused.
8	VLOGIC	Digital IO supply voltage. VLOGIC must be ≤ VDD at all times.
9	AD0	I^{2}C Slave Address LSB
10	REGOUT	Regulator filter capacitor connection
12	INT	Interrupt digital output (totem pole or open-drain)
13	VDD	Power supply voltage
18	GND	Power supply ground
11	RESV-G	Reserved - Connect to ground.
6, 7, 19, 21, 22	RESV	Reserved. Do not connect.
20	CPOUT	Charge pump capacitor connection
23	SCL	I^{2}C serial clock
24	SDA	I^{2}C serial data
2, 3, 4, 5, 14, 15, 16, 17	NC	Not internally connected. May be used for PCB trace routing.

PCB BOARD PRESENTATION

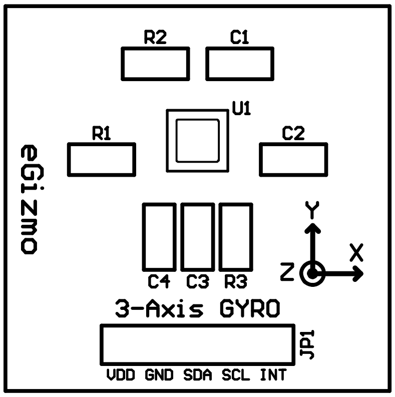


Figure 4. *3-Axis GYRO (silkscreen layout)*

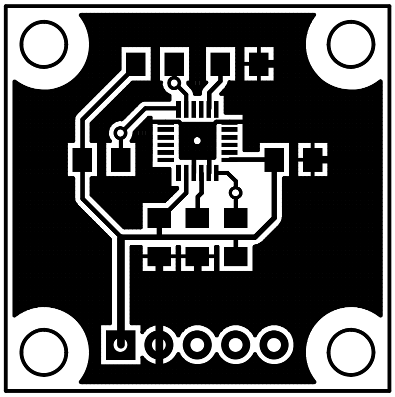


Figure 5. *3-Axis GYRO PCB Copper Pattern (Top Layer)*

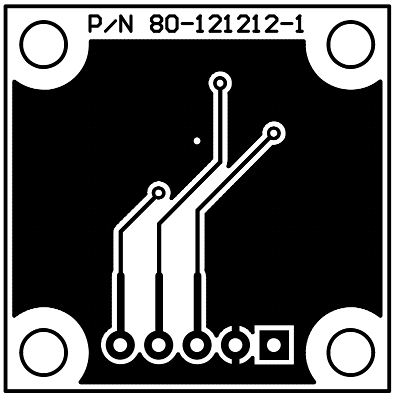


Figure 6. *3-Axis GYRO PCB Copper Pattern (Bottom Layer)*

Schematic diagram

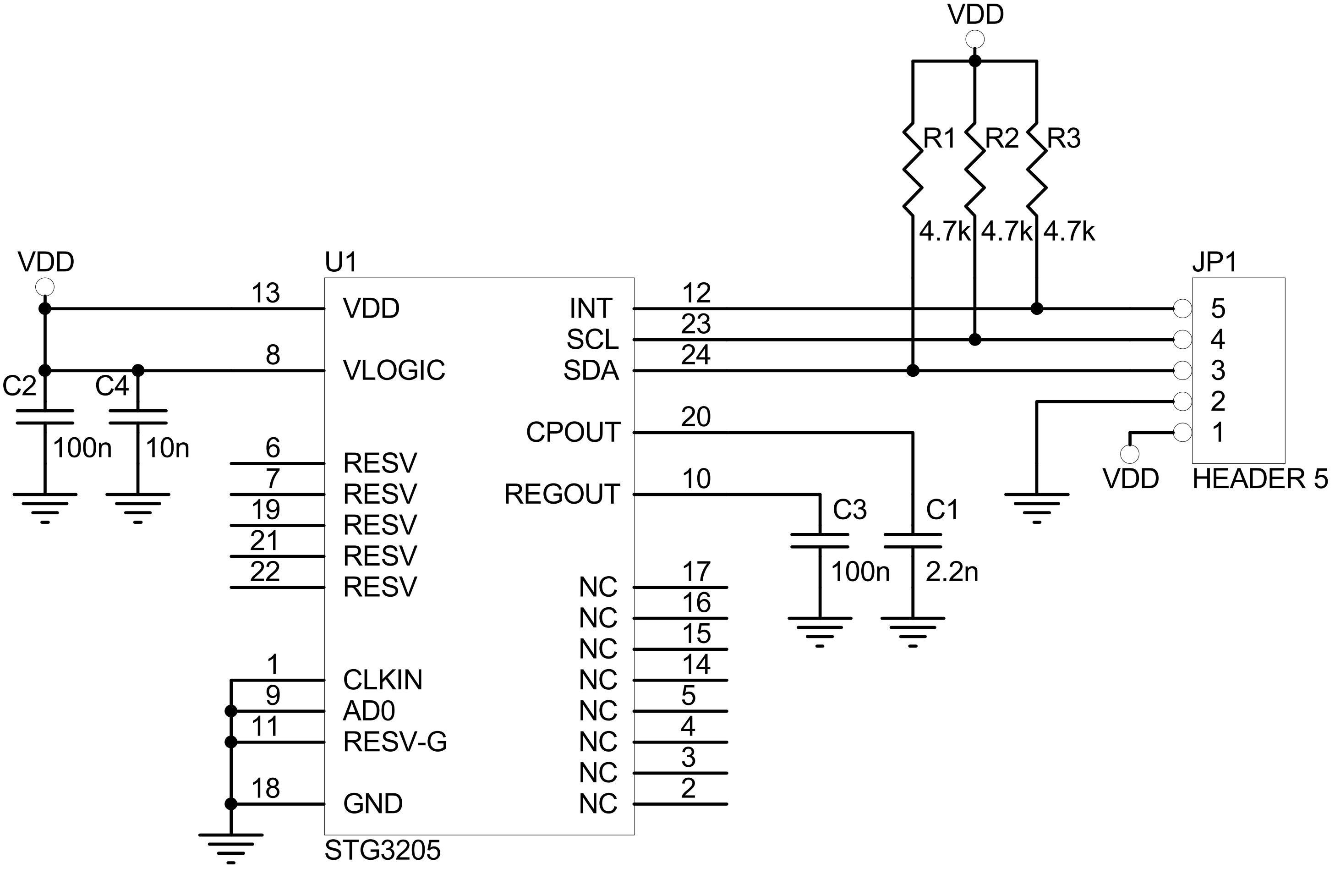


Figure 7. *Schematic Diagram of 3-Axis GYRO*

SAMPLE CIRCUIT DIAGRAM WITH SAMPLE CODES

```
//From http://www.varesano.net//
#include <Wire.h> // I2C library, gyroscope
// Gyroscope ITG3200
#define GYRO 0x68 //  when AD0 is connected to GND
,gyro address is 0x68.
//#define GYRO 0x69   when AD0 is connected to VCC
,gyro address is 0x69
#define G_SMPLRT_DIV 0x15
#define G_DLPF_FS 0x16
#define G_INT_CFG 0x17
#define G_PWR_MGM 0x3E
#define G_TO_READ 8 // 2 bytes for each axis x, y, z
// offsets are chip specific.
int g_offx = 120;
int g_offy = 20;
int g_offz = 93;
int hx, hy, hz, turetemp;
//initializes the gyroscope
void initGyro()
{
 /*****************************************
 * ITG 3200
 * power management set to:
 * clock select = internal oscillator
 * no reset, no sleep mode
 * no standby mode
 * sample rate to = 125Hz
 * parameter to +/- 2000 degrees/sec
 * low pass filter = 5Hz
 * no interrupt
 ******************************************/
 writeTo(GYRO, G_PWR_MGM, 0x00);
 writeTo(GYRO, G_SMPLRT_DIV, 0x07); // EB, 50, 80,
7F, DE, 23, 20, FF
 writeTo(GYRO, G_DLPF_FS, 0x1E); // +/- 2000
dgrs/sec, 1KHz, 1E, 19
 writeTo(GYRO, G_INT_CFG, 0x00);
}
```

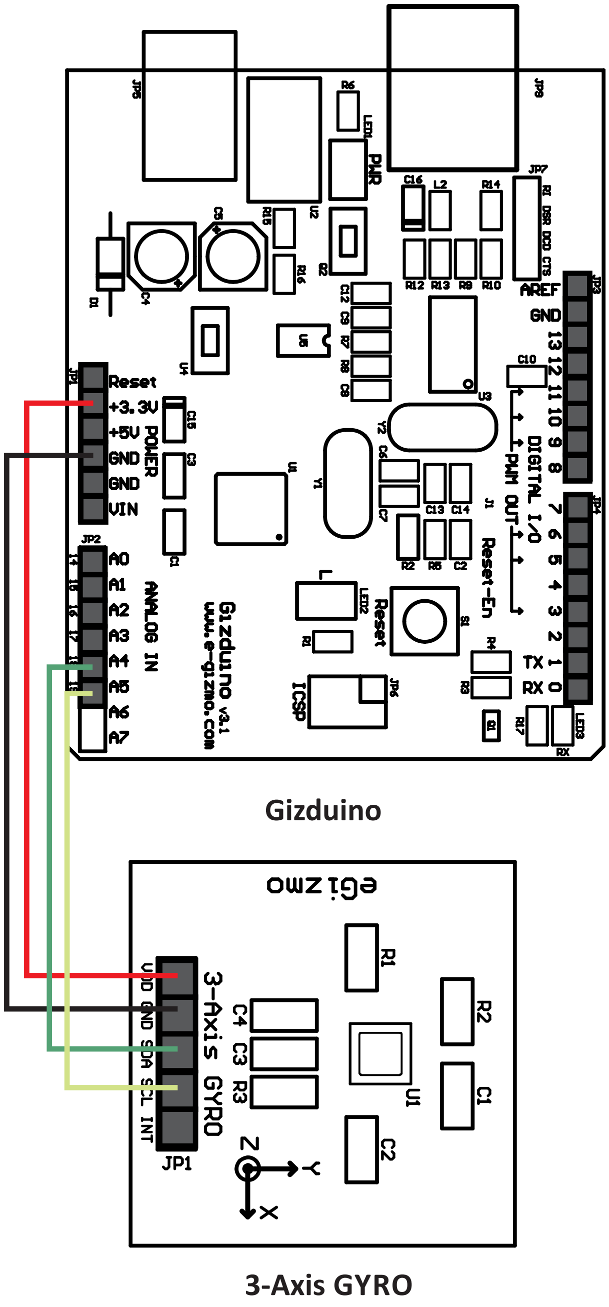


Figure 8. Sample wire connection

SAMPLE CODES

```
void getGyroscopeData(int * result)
{
 /**************************************
 Gyro ITG-3200 I2C
 registers:
 temp MSB = 1B, temp LSB = 1C
 x axis MSB = 1D, x axis LSB = 1E
 y axis MSB = 1F, y axis LSB = 20
 z axis MSB = 21, z axis LSB = 22
 *************************************/
 int regAddress = 0x1B;
 int temp, x, y, z;
 byte buff[G_TO_READ];
 readFrom(GYRO, regAddress, G_TO_READ, buff);
//read the gyro data from the ITG3200
 result[0] = ((buff[2] << 8) | buff[3]) + g_offx;
 result[1] = ((buff[4] << 8) | buff[5]) + g_offy;
 result[2] = ((buff[6] << 8) | buff[7]) + g_offz;
 result[3] = (buff[0] << 8) | buff[1]; // temperature
 }
 //
void setup()
{
 Serial.begin(9600);
 Wire.begin();
 initGyro();
}
//
void loop()
{
 byte addr;
 int gyro[4];
 getGyroscopeData(gyro);
 hx = gyro[0] / 14.375;
 hy = gyro[1] / 14.375;
 hz = gyro[2] / 14.375;
 turetemp = 35+ ((double) (gyro[3] + 13200)) / 280;
// temperature
 Serial.print(" X=");
 Serial.print(hx);
 Serial.print(" Y=");
 Serial.print(hy);
 Serial.print(" Z=");
 Serial.print(hz);
 Serial.print(" F=");
 Serial.print(turetemp);
 Serial.print((char)223);
 Serial.println("C");
 delay(100);
}
//---------------- Functions
//Writes val to address register on ACC
void writeTo(int DEVICE, byte address, byte val) {
  Wire.beginTransmission(DEVICE); //start
transmission to ACC
  Wire.write(address);        // send register address
  Wire.write(val);        // send value to write
  Wire.endTransmission(); //end transmission
}
//reads num bytes starting from address register
on ACC in to buff array
 void readFrom(int DEVICE, byte address, int
num, byte buff[]) {
 Wire.beginTransmission(DEVICE); //start
transmission to ACC
 Wire.write(address);        //sends address to read
from
 Wire.endTransmission(); //end transmission

 Wire.beginTransmission(DEVICE); //start
transmission to ACC
 Wire.requestFrom(DEVICE, num);    // request 6
bytes from ACC

 int i = 0;
 while(Wire.available())    //ACC may send less
than requested (abnormal)
 {
   buff[i] = Wire.read(); // receive a byte
   I
 }
 Wire.endTransmission(); //end transmission
]
```