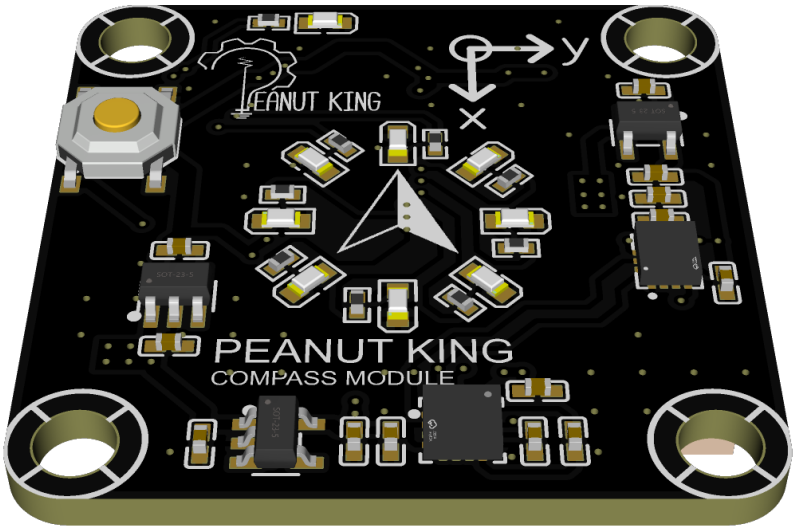
Compass module V2.8



Introduction

Compass module is a yaw rate sensor. Employing a 3-Axis Digital Compass IC HMC5883L and 6-axis MEMS MotionTracking device ICM-42670-P that combines a 3-axis gyroscope and a 3-axis accelerometer. The Compass module is able to measure the vertical axis in degrees.

Features

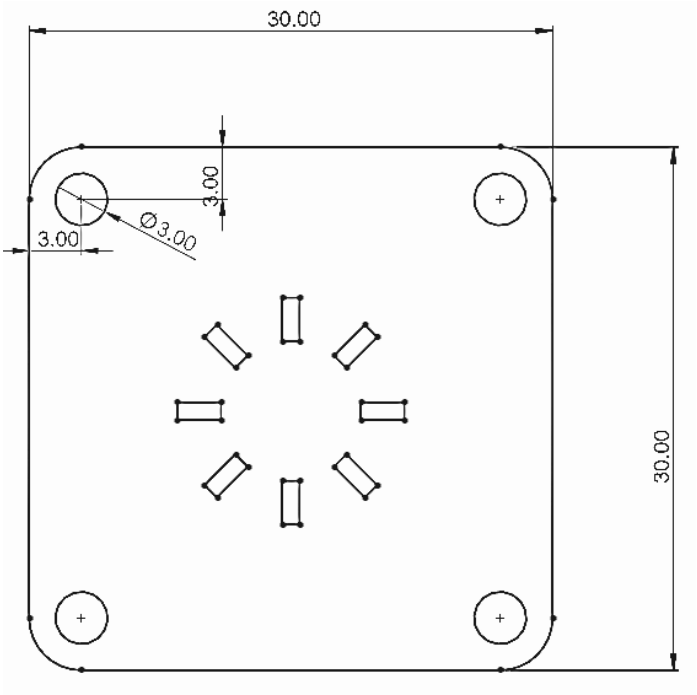
STM32L051 Microcontroller embedded
Reset north point by pressing left side button
Display north point with LED
Cold start time < 0.3s

Specification

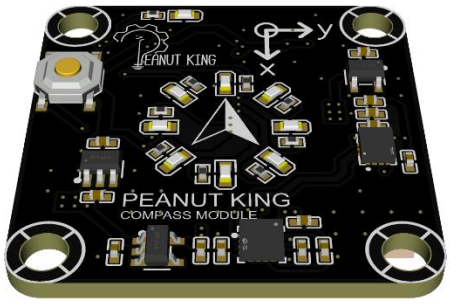
Power supply voltage: 4.5 ~ 5.5V DC
Operating current: ~ 0.7 mA
Operating temperature: -40 ~ 85°C
Operating Humidity: <90%
Size: 30 mm x 30 mm x 10 mm
Weight: 4.2g
Interface: I^2C Standard-mode 7-bits address
7-bits I^2C address: 0x08
Return data: 0-359

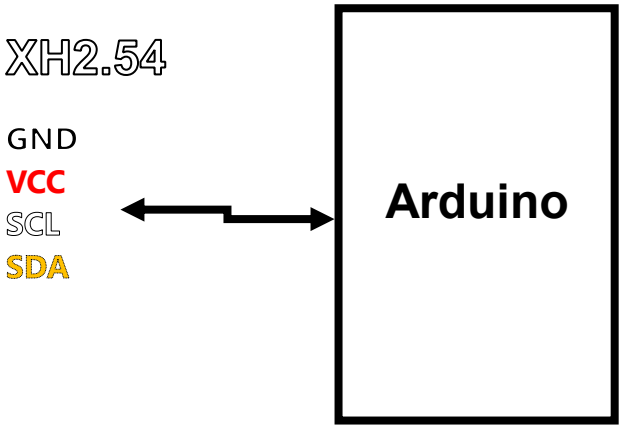
Compound

Front:



Interface





Related Links:

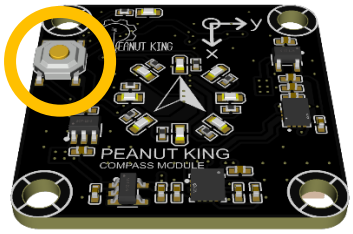
Arduino library Link:
https://github.com/peanut-king-solution/PeanutKing_ArduinoShield

Micro: bit library link:
https://github.com/peanut-king-solution/pxt-pks-shield

Compass Calibration

In some cases, your compass reading may be distorted because of some external interference. Here are the steps to calibrate the compass.

Step1: Press and hold the reset button for 2 seconds until all indicators flash.



Step 2: Rotate the robot in an anticlockwise direction for at least 1 turn, slowly and steadily. Keep the indicator pointing in the same direction while turning the robot.

Rotate until all the indicators flash again to end the calibration process. Check the indicators by rotating the robot and see if the indicators point correctly. If the readings are still distorted, re -do the above steps.

Communication Bus Interface I^2C and Its Address

This device, functioning as a slave, will be connected to the master device via I2C, under master control through a serial interface.

This device is compliant with I^2C-Bus Specification. As an I^2C compatible device, this device has a 7-bit serial address and supports I^2C protocols. This device supports standard mode 100kHz.

The default I^2C address is 0x08
If others I^2C address options are required, please contact factory

Register Map

Addr.	7	6	5	4	3	2	1	0	Access
0x40H	ACC_X_RAW MSB[15:8]								Read only
0x41H	ACC_X_RAW LSB[7:0]								Read only
0x42H	ACC_Y_RAW MSB[15:8]								Read only
0x43H	ACC_Y_RAW LSB[7:0]								Read only
0x44H	ACC_Z_RAW MSB[15:8]								Read only
0x45H	ACC_Z_RAW LSB[7:0]								Read only
0x46H	GYR_X_RAW MSB[15:8]								Read only
0x47H	GYR_X_RAW LSB[8:0]								Read only
0x48H	GYR_Y_RAW MSB[15:8]								Read only
0x49H	GYR_Y_RAW LSB[7:0]								Read only
0x4aH	GYR_Z_RAW MSB[15:8]								Read only
0x4bH	GYR_Z_RAW LSB[7:0]								Read only
0x4cH	MAG_X_RAW LSB[7:0]								Read only
0x4dH	MAG_X_RAW MSB[15:8]								Read only
0x4eH	MAG_Y_RAW LSB[7:0]								Read only
0x4fH	MAG_Y_RAW MSB[15:8]								Read only
0x50H	MAG_Z_RAW LSB[7:0]								Read only
0x51H	MAG_Z_RAW MSB[15:8]								Read only
0x52H	Reserved								/
0x53H	Reserved								/
0x54H	Reserved								/
0x55H	Reserved								/
0x56H	COMPASS_YAW LSB[7:0]								Read only
0x57H	COMPASS_YAW MSB[15:8]								Read only

Register Descriptions

1. **Registers 0x40 to 0x45 – Accelerometer Measurements**
   ACC_X_RAW_MSB, ACC_X_RAW_LSB, ACC_Y_RAW_MSB, ACC_Y_RAW_LSB ,ACC_Z_RAW_MSB and ACC_Z_RAW_LSB

   Type: Read Only

   Description:

   These registers store the most recent accelerometer measurements.

   The acceleration in m/s^2 for a given register value may be computed as:

   Acceleration in m/s^2 = (ACC_RAW Register Value as a int16_t)*16.0*9.81[1]/32768

2. **Register 0x46 to 0x4b – Gyroscope Measurements**
   GYR_X_RAW_MSB, GYR_Y_RAW_LSB, GYR_Y_RAW_MSB, GYR_Y_RAW_LSB, GYR_Z_RAW_MSB and GYR_Z_RAW_LSB

   Type: Read Only

   Description:

   These registers store the most recent gyroscope measurements.

   The angular acceleration in rad/s^2 for a given register value may be computed as:

   Angular acceleration in ms^{-2} = (GYR_RAW Register Value as a int16_t)*1000/32768

3. **Registers 0x4c to 0x45 – Magnetometer Measurements**
   MAG_X_RAW_LSB, MAG_X_RAW_MSB, MAG_Y_RAW_LSB, MAG_Y_RAW_MSB, MAG_Z_RAW_LSB and MAG_Z_RAW_MSB

   Type: Read Only

   Description:

   These registers store the most recent magnetometer measurements in each axis.

   Each axis has 16 bit data width in 2's complement, i.e., MSB of 0x4d indicates the sign of Y axis.

   The output data of each channel saturates at -32768 and 32767.

4. **Register 0x56 to 0x57 – Compass Yaw Measurements**
   COMPASS_YAW_LSB, COMPASS_YAW_MSB

   Type: Read Only

   Description:

   These registers store the most recent calculated compass yaw angle.

   The yaw angle in degree for a given register value may be computed as:

   Yaw angle in degree = (COMPASS_YAW Register Value as a uint16_t)/100

   The calculated compass yaw angle bounded from 0 to 359.

[1] 9.81 refers to the acceleration due to gravity on Earth. This value can be replaced with a more precise decimal value for increased accuracy.

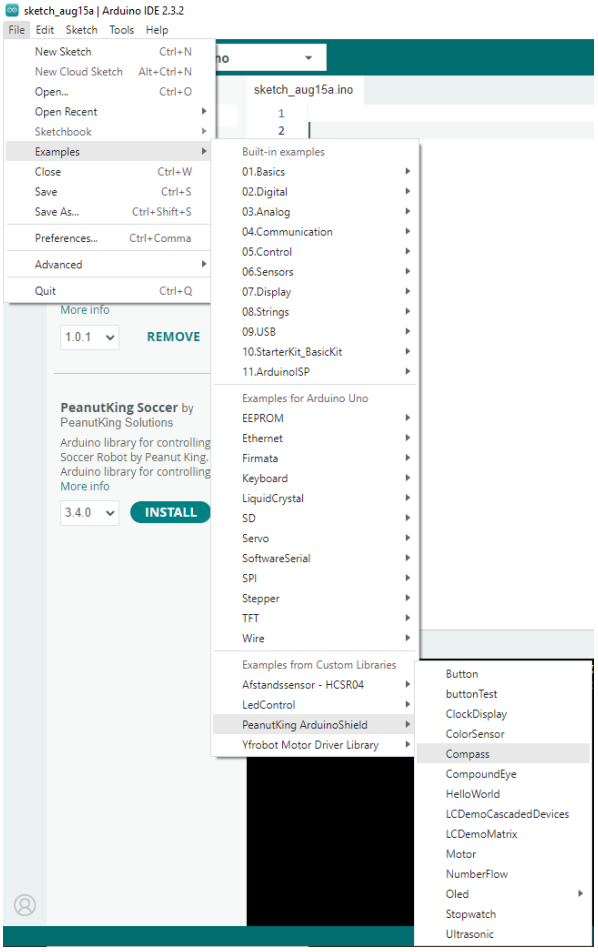
Connection

Peanut King Uno Shield/ micro: bit Robot Shield

Use 4 pin connecting cable to connect with the Shield

Sample Code

Path: File → Examples → PeanutKing ArduinoShield → compass



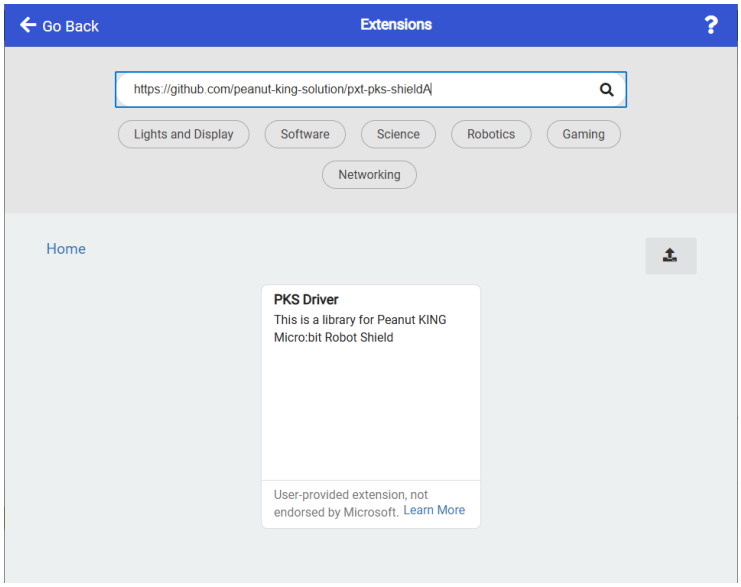

```
#include <PeanutKingArduinoShield.h>
PeanutKingArduinoShield robot;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int compass_value = robot.compassRead();
  Serial.print("compass_value: ");
  Serial.println(compass_value);
}
```

Micro: bit Robot Shield

Use 4 pin connecting cable to connect with Micro: bit Robot shield

Install PeanutKing Micro:bit Robot Shield extension

Path: Extension →enter “https://github.com/peanut-king-solution/pxt-pks-shield”



Sample Code

Revision Information

Change from 1.0 (April ,2026) to current version	Page
2.0	**10**

Important Information and Disclaimer

The information in this statement reflects Peanut King Solution Limited's knowledge and belief as of the date it is provided. Peanut King Solution Limited relies on information from third parties and does not guarantee its accuracy. Efforts are ongoing to better integrate third-party information. Peanut King Solution Limited has taken and continues to take reasonable steps to provide accurate and representative information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. Certain information is considered proprietary by Peanut King Solution Limited and its suppliers, so CAS numbers and other limited information may not be available for release.

NOTICE

Peanut King Solution Limited reserves the right to make changes to the products in this document to improve performance or for any other reason, or to discontinue them without notice. Customers should contact Peanut King Solution Limited for the latest product information before placing orders or designing Peanut King Solution Limited products into systems. Peanut King Solution Limited assumes no responsibility for the use of any products or circuits described in this document or customer product design, does not convey any license under any patent or other right, and does not guarantee that the circuits are free of patent infringement. Peanut King Solution Limited makes no claim regarding the suitability of its products for any particular purpose and assumes no liability arising from the use of any product or circuit, specifically disclaiming any and all liability, including consequential or incidental damages.

PEANUT KING SOLUTION LIMITED PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY OR DEATH. THE USE OF PEANUT KING SOLUTION LIMITED PRODUCTS IN LIFE SUPPORT SYSTEMS IS NOT AUTHORIZED.