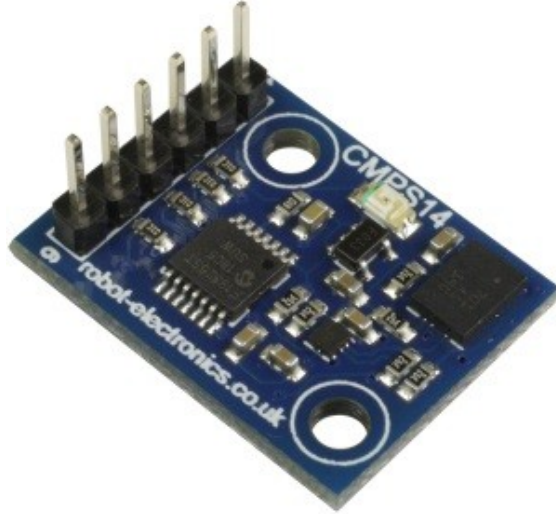


# CMPS14 – documentation



## Introduction

The CMPS14 is our 5th generation tilt compensated compass. Employing a 3-axis magnetometer, a 3-axis gyro and a 3-axis accelerometer. At the core of the module is the superb BNO080 running algorithms to remove the errors caused by tilting of the PCB. The module also allows the calibration to be stopped and instead rely on a static calibration profile.

Power supply requirements are flexible, you can feed between 3.3 - 5v and the module draws a nominal 18mA of current. A choice of serial or I2C interfaces can be used for communication.

## Overview of outputs

**Heading, 16 bit** – 2 outputs, one calculated by Bosch and one by us

**Heading, 8 bit** – 0-255 scaled for simpler requirements

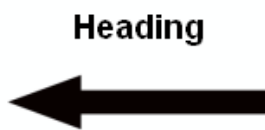
**Pitch** – +/- 0-90° or +/- 0-180°

**Roll** – +/- 0-90°

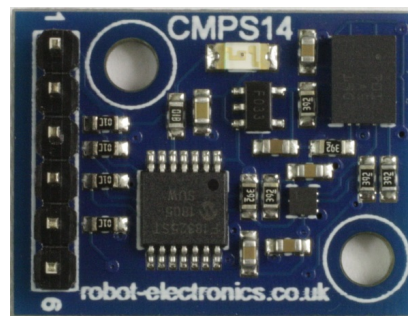
**Raw sensor outputs** – 3 x 16 bit integers for each of the Magnetometer, accelerometer and gyro

## Mode selection

Serial or I2C mode is easily selected with the state of the mode pin. Note the CMPS14 looks at the mode selection pin at power-up only.



3.3v-5v  
SDA/TX  
SCL/RX  
Mode  
Factory use  
0v ground



For **I2C** the mode pin can be left open or pulled to the supply voltage, for **serial mode** the mode pin should be connected to 0v ground.

## I2C Mode

Register	Function
0x00	Command register (write) / Software version (read)
0x01	Compass Bearing 8 bit, i.e. 0-255 for a full circle
0x02, 0x03	Compass Bearing 16 bit, i.e. 0-3599, representing 0-359.9 degrees. register 2 being the high byte. This is calculated by the processor from quaternion outputs of the BNO055
0x04	Pitch angle - signed byte giving angle in degrees from the horizontal plane (+/- 90°)
0x05	Roll angle - signed byte giving angle in degrees from the horizontal plane (+/- 90°)
0x06, 0x07	Magnetometer X axis raw output, 16 bit signed integer (register 0x06 high byte)
0x08, 0x09	Magnetometer Y axis raw output, 16 bit signed integer (register 0x08 high byte)
0x0A, 0x0B	Magnetometer Z axis raw output, 16 bit signed integer (register 0x0A high byte)
0x0C, 0x0D	Accelerometer X axis raw output, 16 bit signed integer (register 0x0C high byte)
0x0E, 0x0F	Accelerometer Y axis raw output, 16 bit signed integer (register 0x0E high byte)
0x10, 0x11	Accelerometer Z axis raw output, 16 bit signed integer (register 0x10 high byte)
0x12, 0x13	Gyro X axis raw output, 16 bit signed integer (register 0x12 high byte)
0x14, 0x15	Gyro Y axis raw output, 16 bit signed integer (register 0x14 high byte)
0x16, 0x17	Gyro Z axis raw output, 16 bit signed integer (register 0x16 high byte)
0x1C, 0x1D	Roll angle 16 bit - signed byte giving angle in tenths of degrees from the horizontal plane (+/- 180°, 0x1C is the high byte)
0x1E	Calibration state, bits 0 and 1 reflect the calibration status (0 un-calibrated, 3 fully calibrated)

Register 0x00 in the event of a read the CMPS14 will reply with the software version, for a write it acts as the command register.

Register 0x01 is the bearing as a 0-255 value, this may be easier for some applications than 0-3599 which requires two bytes.

For those who require a bearing with better resolution registers 0x02 and 0x03 (high byte first) form a 16 bit unsigned integer in the range 0-3599. This represents 0-359.9°.

Registers 0x04 and 0x05 are the pitch and roll angles, giving an angle of 0 when the board is flat and up to +/- 90° at maximum tilt in either direction.

There is then an array of registers (0x06-0x17) providing all the raw sensor data from the 3 sensors.

Registers 0x1C (high byte) and 0x1D (low byte) form a 16 bit roll angle for +/- 180 from the horizontal plane. Value is in tenths of degrees (range of +/- 1800).

Finally we have register 0x1E, this provides feedback on the degree of the calibration that the automatic calibration routines have achieved.

## Calibration of the CMPS14

The CMPS14 is shipped with a static factory calibration, but the module is able to automatically update its calibration with algorithms running in the background if desired. To achieve background calibration the user just needs to turn the functionality on and perform the required simple movements.

### Changing the calibration configuration

To change the configuration write the following to the command register 0x98, 0x95, 0x99 with a 20ms delay after each of the three bytes. You can then pass the setup byte to the command register, it takes the form:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Calibration config	1	x	x	Periodic auto save enable	x	Gyro Cal enable	Accel Cal enable	Mag Cal enable

Bit 7 is always logic high

Bits 6,5 and 3 are undefined

Bit 4 enables a periodic automatic save of calibration data so manual saving is not required

Bit 2 enables the background gyro calibration – recommended to be off for very still applications

Bit 1 enables the accelerometer background calibration

Bit 0 enables the magnetometer background calibration

Gyro – calibrated by the CMPS14 being in a stationary state.

Accelerometer – calibrated by tilting the module to roughly 45 and 90 degrees on one axis.

Magnetometer – a few random movements easily calibrates the CMPS14

The level to which the CMPS14 has been calibrated can be checked by reading register 0x1E, the gyro, accelerometer and magnetometer are allocated 2 bits each in the register. A value of 0 in the two bits reflects an uncalibrated state, when fully calibrated this will become 3 (both bits set). There is also a complete system calibration level. Please note the Gyro feedback does not currently work, this is a bug in the sensor itself that is scheduled to be fixed.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cal status	System calibration		Gyro calibration		Accelerometer calibration		Magnetometer calibration	

### Storing and erasing calibration profiles

After calibrating the compass the profile can be stored so it will be automatically reloaded when the module is ready for use again.

To **store a profile** write the following to the command register 0xF0, 0xF5, 0xF6 with a 20ms delay after each of the three bytes.

If you wish to **erase the stored profile** write the following to the command register 0xE0, 0xE5, 0xE2 with a 20ms delay after each of first two bytes. The final byte will reset the BNO080 so a delay of 300ms is recommended.

### Changing the I2C Bus Address

To change the I2C address of the CMPS14 you must have only one module on the bus. Write the 3 sequence commands in the correct order followed by the address with 20ms between writes. Example; to change the address of a compass currently at 0xC0 (the default shipped address) to 0xC2, write the following to the command register 0 at address 0xC0: (0xA0, 0xAA, 0xA5, 0xC2 ) with a 20ms delay after each of the first three bytes. These commands must be sent in the correct sequence to change the I2C address, additionally, No other command may be issued in the middle of the sequence. The sequence must be sent to the command register at location 0, which means 4 separate write transactions on the I2C bus. When the CMPS14 is re-powered will flash its address out on the LED.

Address		flashes
Decimal	Hex	
192	C0	1
194	C2	2
196	C4	3
198	C6	4
200	C8	5
202	CA	6
204	CC	7
206	CE	8

Take care not to set more than one device to the same address, there will be a bus collision and very unpredictable results.

# Serial mode

## Communication settings

The Serial mode operates over a link with a default baud rate of 9600 bps (no parity, 2 stop bits) and 3.3v-5v signal levels. This is not RS232. Do not connect RS232 to the module, the high RS232 voltages will irreversibly damage the module.

## Commands for Serial

Command	Name	Bytes returned	Returned data description
0x11	GET VERSION	1	Software version
0x12	GET BEARING 8 BIT	1	Bearing as a single byte 0-255
0x13	GET BEARING 16 BIT	2	Bearing (16 bit), high byte first 0-3599
0x14	GET PITCH	1	Pitch angle +/- 0-90°
0x15	GET ROLL	1	Roll angle +/- 0-90°
0x19	GET MAG RAW	6	Raw magnetic data, 16 bit signed: X high, X low, Y high, Y low, Z high, Z low
0x20	GET ACCEL RAW	6	Raw accelerometer data, 16 bit signed: X high, X low, Y high, Y low, Z high, Z low
0x21	GET GYRO RAW	6	Raw gyro data, 16 bit signed: X high, X low, Y high, Y low, Z high, Z low
0x23	GET ALL	4	Angle high, angle low (0-3599), pitch (+/- 0-90), roll (+/- 0-90)
0x24	GET CALIBRATION STATE	1	Bits 0 and 1 reflect the calibration status (0 uncalibrated, 3 fully calibrated)
0x26	GET ROLL 180	2	roll angle (16 bit) high bytes first +/- 0-180°
0x98	CHANGE CALIBRATION CONFIG BYTE 1	1	Returns ok (0x55)
0x95	CHANGE CALIBRATION CONFIG BYTE 2	1	Returns ok (0x55)
0x99	CHANGE CALIBRATION CONFIG BYTE 3	1	Returns ok (0x55)
0xF0	STORE CALIBRATION BYTE 1	1	Returns ok (0x55)
0xF5	STORE CALIBRATION BYTE 2	1	Returns ok (0x55)
0xF6	STORE CALIBRATION BYTE 3	1	Returns ok (0x55)
0xE0	DELETE CALIBRATION BYTE 1	1	Returns ok (0x55)
0xE5	DELETE CALIBRATION BYTE 2	1	Returns ok (0x55)
0xE2	DELETE CALIBRATION BYTE 3	1	Returns ok (0x55)
0xA0	BAUD 19200	1	Returns ok (0x55)
0xA1	BAUD 38400	1	Returns ok (0x55)

## Calibration of the CMPS14

The CMPS14 is shipped with a static factory calibration, but the module is able to automatically update its calibration with algorithms running in the background if desired. To achieve background calibration the user just needs to turn the functionality on and perform the required simple movements.

### Changing the calibration configuration

To change the configuration write the following sequence of bytes 0x98, 0x95, 0x99 , remembering to pick up the response byte OK (0x55) after each byte. You can then pass the setup byte to the command register, it takes the form:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Calibration config	1	x	x	Periodic auto save enable	x	Gyro Cal enable	Accel Cal enable	Mag Cal enable

Bit 7 is always logic high

Bits 6,5 and 3 are undefined

Bit 4 enables a periodic automatic save of calibration data so manual saving is not required

Bit 2 enables the background gyro calibration – recommended to be off for very still applications

Bit 1 enables the accelerometer background calibration

Bit 0 enables the magnetometer background calibration

Gyro – calibrated by the CMPS14 being in a stationary state.

Accelerometer – calibrated by tilting the module to roughly 45 and 90 degrees on one axis.

Magnetometer – a few random movements easily calibrates the CMPS14

The level to which the CMPS14 has been calibrated can be checked by using command 0x24, the gyro, accelerometer and magnetometer are allocated 2 bits each in the register. A value of 0 in the two bits reflects an uncalibrated state, when fully calibrated this will become 3 (both bits set). There is also a complete system calibration level.. Please note the Gyro feedback does not currently work, this is a bug in the sensor itself that is scheduled to be fixed.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cal status	System calibration		Gyro calibration		Accelerometer calibration		Magnetometer calibration	

### Storing and erasing calibration profiles

After calibrating the compass the profile can be stored so it will be automatically reloaded when the module is ready for use again. To do this we need to write a sequence of 3 bytes.

To **store a profile** send the sequence 0xF0, 0xF5, 0xF6 to the CMPS14, remembering to pick up the response byte OK (0x55) after each byte.

If you wish to **erase the stored profile** send the sequence 0xE0, 0xE5, 0xE2 again each byte will return an OK (0x55). The final byte will reset the BNO080 so a delay of 300ms is recommended.

### Changing the baud rate

The default serial baud rate of 9600 can be changed. There are two other baud rates that can be used, for 19200 just send 0xA0 or alternatively for 38400 send 0xA1. Please note that the CMPS14 will always default to its 9600 bps rate after power cycling and after setting a new baud rate the ok response (0x55) will be sent at the newly selected speed.

# Board dimensions

