CMPS10 Library Manual

Written by Neil Dhar
Adapted from http://www.robot-electronics.co.uk/htm/cmps10doc.htm

Compass Sensor I2C (CMPS10_I2C)

Constructor

```
CMPS10_I2C::CMPS10_I2C(TwoWire * _bus, byte Addr)
```

```
CMPS10_I2C::CMPS10_I2C(TwoWire * _bus, byte Addr, int _xOffset, int
_xScale, int _yOffset, int _yScale)
```

The constructor function takes a pointer to the I2C bus that the sensor is on as well as the sensor's I2C address.

The alternative constructor takes a series of calibration values as well. To calibrate the compass, first find the minimum and maximum value for each axis. Do this by reading all the outputs of an axis when moving the sensor in a complete circle in the environment you will be using it in. Subsequently, obtain the Offset value for each axis by taking -(max+min)/2. Then get the Scale value for each axis using (max-min)/2. This allows for normalization of the values on each axis and restricts them to between [-1,1]. This then allows the atan function to effectively compare the values.

A sample instantiation of the class as an object `myCMPS` with I2C address `0xEE` is as follows:

```
CMPS10_I2C myCMPS(&Wire, 0xEE);
```

Read Tilt Compensated Bearing

```
int CMPS10_I2C::read()
```

The function returns the value from the compass. For example, to read from the object `myCMPS`.:

```
myCMPS.read();
```

Read Magnetometer Bearing

```
int SRF10::magRead()
```

This provides readings much faster. It requires the offsets and scales to be provided in the constructor. For example, to read the value from `myCMPS`:

```
myCMPS.magRead();
```

Calibrate

```
void CMPS10_I2C::calibrate()
```

Plug in to computer, open Serial monitor and run the function. Follow onscreen prompts.

Factory Reset

```
void CMPS10_I2C::factoryReset()
```

Plug into computer and run the function. Send a single letter ‘a’ when asked for input.

Read Raw Magnetometer Value

```
int CMPS10_I2C::readMagAxis(char axis)
```

Call with ‘X’ or ‘Y’ to read the respective magnetometer values.

Compass Sensor Serial (CMPS10_Serial)

Constructor

```
CMPS10_Serial::CMPS10_Serial(HardwareSerial * _bus)
```

```
CMPS10_Serial::CMPS10_Serial(HardwareSerial * _bus, int _xOffset, int
_xScale, int _yOffset, int _yScale)
```

The constructor function takes a pointer to the Serial bus that the sensor is on.

The alternative constructor takes a series of calibration values as well. To calibrate the compass, first find the minimum and maximum value for each axis. Do this by reading all the outputs of an axis when moving the sensor in a complete circle in the environment you will

be using it in. Subsequently, obtain the Offset value for each axis by taking -(max+min)/2. Then get the Scale value for each axis using (max-min)/2. This allows for normalization of the values on each axis and restricts them to between [-1,1]. This then allows the atan function to effectively compare the values.

A sample instantiation of the class as an object `myCMPS` with on bus `Serial1` is as follows:

```
CMPS10_Serial myCMPS(&Serial1, 0xEE);
```

Set Baud Rate

```
void CMPS10_Serial::setBaud(int baudRate)
```

This function can change the baud rate from the default 9600. It will also switch the baud rate on the appropriate serial bus. For example, to set the baud rate for the object `myCMPS` to 19200:

```
myCMPS.setBaud(19200);
```

Read Tilt Compensated Bearing

```
int CMPS10_Serial::read()
```

The function returns the value from the compass. For example, to read from the object `myCMPS`:

```
myCMPS.read();
```

Read Magnetometer Bearing

```
int CMPS10_Serial::magRead()
```

This provides readings much faster. It requires the offsets and scales to be provided in the constructor. For example, to read the value from `myCMPS`:

```
myCMPS.magRead();
```

Calibrate

```
void CMPS10_Serial::calibrate()
```

Plug in to computer, open Serial monitor and run the function. Follow onscreen prompts.

Factory Reset

```
void CMPS10_Serial::factoryReset()
```

Plug into computer and run the function. Send a single letter ‘a’ when asked for input.

Read Raw Magnetometer Value

```
int CMPS10_Serial::readMagAxis(char axis)
```

Call with ‘X’ or ‘Y’ to read the respective magnetometer values.