```
/*
 Arduino    Magnetometer
 A5         SCL
 A4         SDA
 3.3V       VCC
 GND        GND
 */

#include <Wire.h> //I2C Library
#include <HMC5883L.h> // HMC5883L Magnetometer Library

HMC5883L compass; // Store our compass as a variable
int error = 0; // Record any errors that may occur in the compass

const int pinB = 9;
const int pinG = 10;
const int pinR = 11;

int r = 0;
int g = 0;
int b = 0;

int maximo = 100;


void setup() { // configuring the microcontroller and compass
  Serial.begin(9600); // Initialize the serial port
  pinMode(pinB, OUTPUT);
  pinMode(pinG, OUTPUT);
  pinMode(pinR, OUTPUT);

  Wire.begin(); // Start the I2C interface.
  compass = HMC5883L(); // Construct a new HMC5883 compass
  error = compass.SetScale(1.3); // Set the scale to +/- 1.3 Ga of the compass
  if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));

  Serial.println("Setting measurement mode to continous.");
  error = compass.SetMeasurementMode(Measurement_Continuous); // Set the measurement mode to Continuous
  if(error != 0) // If there is an error, print it out.
    Serial.println(compass.GetErrorText(error));
}

// Our main program loop.
void loop()
{
  // Retrive the raw values from the compass (not scaled).
  MagnetometerRaw raw = compass.ReadRawAxis();
  // Retrived the scaled values from the compass (scaled to the configured scale).
  MagnetometerScaled scaled = compass.ReadScaledAxis();

  // Values are accessed like so:
  int MilliGauss_OnThe_XAxis = scaled.XAxis;// (or YAxis, or ZAxis)

  // Calculate heading when the magnetometer is level, then correct for signs of axis.
  float heading = atan2(scaled.YAxis, scaled.XAxis);

  float declinationAngle = -128.3/1000; //declinacion calculada de buenos aires en www.magnetic-declination.com y
conversion a radianes en http://www.wolframalpha.com/
```

```
  heading += declinationAngle;

  // Correct for when signs are reversed.
  if(heading < 0)heading += 2*PI;


  if(heading > 2*PI) heading -= 2*PI;

  heading = heading * 180/M_PI;

  //Output(raw, scaled, heading, headingDegrees);

  if((heading >= 0)&&(heading<90)){
    r = int(map(heading,0,90,maximo,0));
    g = int(map(heading,0,90,0,maximo));
    b = 0;
  }else if((heading >= 90)&&(heading<135)){
    r = 0;
    g = maximo;
    b = int(map(heading,90,135,0,maximo));
  }else if((heading >= 135)&&(heading<180)){
    r = int(map(heading,135,180,0,maximo));
    g = maximo;
    b = maximo;
  }else if((heading >= 180)&&(heading<225)){
    r = int(map(heading,180,225,maximo,0));
    g = int(map(heading,180,225,maximo,0));
    b = maximo;
  }else if((heading >= 225)&&(heading<=360)){
    r = int(map(heading,225,360,0,maximo));
    g = 0;
    b = int(map(heading,225,360,maximo,0));
  }

  analogWrite(pinR,r);
  analogWrite(pinG,g);
  analogWrite(pinB,b);
  Serial.print("tHeading: ");
  Serial.println(heading);
}

// Output the data down the serial port.
void Output(MagnetometerRaw raw, MagnetometerScaled scaled, float heading, float headingDegrees)
{
  Serial.print("Raw:\t");
  Serial.print(raw.XAxis);
  Serial.print("   ");
  Serial.print(raw.YAxis);
  Serial.print("   ");
  Serial.print(raw.ZAxis);
  Serial.print("   \tScaled:\t");

  Serial.print(scaled.XAxis);
  Serial.print("   ");
  Serial.print(scaled.YAxis);
  Serial.print("   ");
  Serial.print(scaled.ZAxis);

  Serial.print("   \tHeading:\t");
```

```
  Serial.print(heading);
  Serial.print(" Radians   \t");
  Serial.print(headingDegrees);
  Serial.println(" Degrees   \t");
}
```