



Síguenos en tu Red Social favorita



Cursos Online

Tweets por @project_arduino

**MSRobotics**

@project_arduino

Scratch 2.0 + Arduino Snap! + Arduino. Un nuevo manual para el fin de semana.
msrobotics.net/index.php/manu...



Insertar

Ver en Twitter

Practica con Arduino y brújula hmc5883L

- [Imprimir](#)
- [Correo electrónico](#)

Categoría: [Laboratorio ARD](#)

Publicado: 06 Octubre 2015

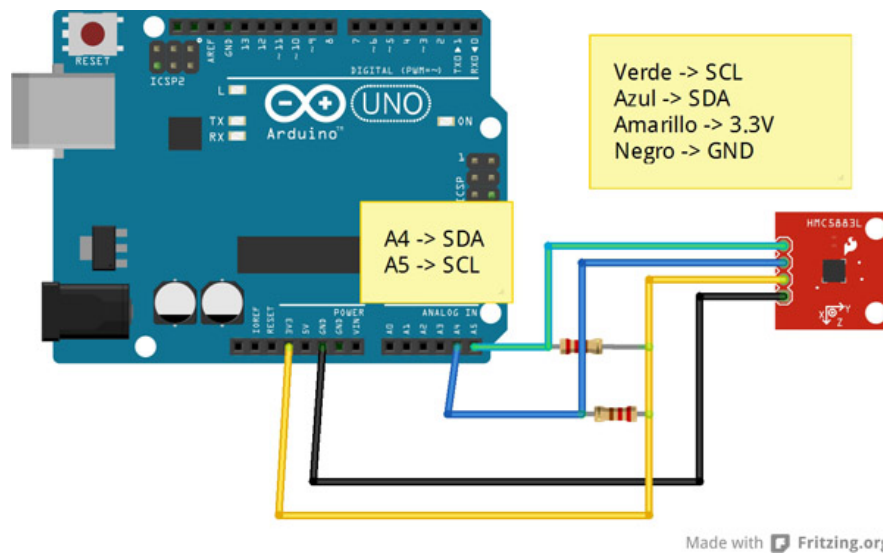
Visto: 849

Hoy os traemos este tutorial para aprender a detectar los puntos cardinales usando una brújula digital hmc5883L con nuestro Arduino. Esta brújula es capaz de detectar el campo magnético en los ejes x,y,z. Tendremos que tener en cuenta que la brújula es sensible a los campos magnéticos que la rodea, por lo tanto si tenemos metales o dispositivos cerca puede influir en el comportamiento del chip. Usaremos la brújula para detectar el campo magnético terrestre y

así saber donde se encuentra el Sur magnético y por consiguiente saber donde esta el Norte el Este y el Oeste.

nota: tener en cuenta que el Sur magnético no es lo mismo que el Sur geográfico. Para más info miraros Wikipedia xD. También hay que tener en cuenta que el sensor detecta los campos magnéticos en el eje z, por lo tanto lo tenemos que colocar lo más paralelo posible al suelo para obtener una mayor precisión.

Después del apartado teórico vamos a la práctica. Usaremos dos resistencias de 2.2k tal y como recomienda el datasheet del sensor entre los pines Vcc-SDA y Vcc-SCL. El pin Vcc deberá ir a la entrada de 3.3V de nuestro Arduino ya que el sensor necesita un voltaje entre 2.16 a 3.6. Ya que el sensor funciona mediante I2C, estos es, línea de datos SDA y línea de reloj SCL tenemos que usar los Pines A4 y A5 de nuestro Arduino respectivamente. Aquí tenemos como vamos a conectar los pines del sensor a nuestro Arduino.



Antes de programar vamos a necesitar dos librerías.

1- Wire.h que nos permite comunicarnos con el sensor, encontraremos esta librería ya incluida en el IDE de Arduino. Mas información sobre la librería [aquí](#).

2- hmc5883l.h la librería para controlar el chip en Arduino [aquí](#).

Ahora vamos a programar nuestro Arduino para saber a cuantos grados nos encontramos del Sur magnético, para ello escribiremos este pequeño ejemplo:

```
#include <Wire.h>
#include <HMC5883L.h>
HMC5883L brujula; //declaramos la variable
int error = 0; // para detectar algun error en la brujula

void setup(){
  // Initialize the serial port.
  Serial.begin(9600);
  Wire.begin(); // iniciamos el interfaz I2C
  brujula = HMC5883L(); // iniciamos la instancia de la brujula
  error = brujula.SetScale(1.3); // indicamos la escala que vamos a usar
  if(error != 0){
    Serial.println(brujula.GetErrorText(error));
  }
  error = brujula.SetMeasurementMode(Measurement_Continuous);
  if(error != 0){
    Serial.println(brujula.GetErrorText(error));
  }
}

void loop(){
  MagnetometerRaw raw = brujula.ReadRawAxis(); // recibimos el valor del RAW sin escalar
  MagnetometerScaled scaled = brujula.ReadScaledAxis(); // recibimos el valor escalado con la configuracion eleg
  float angulo = atan2(scaled.YAxis, scaled.XAxis) * (180 / 3.14159265) + 180; // arcotangete x y. sacamos el angulo en grado
  if((angulo < 22.5) || (angulo > 337.5 )){
    Serial.print("Sur");
  }
  if((angulo > 22.5) && (angulo < 67.5 )){
    Serial.print("Suroeste");
  }
  if((angulo > 67.5) && (angulo < 112.5 )){
    Serial.print("Oeste");
  }
  if((angulo > 112.5) && (angulo < 157.5 )){
    Serial.print("Noroeste");
  }
  if((angulo > 157.5) && (angulo < 202.5 )){
    Serial.print("Norte");
  }
  if((angulo > 202.5) && (angulo < 247.5 )){
    Serial.print("Nordeste");
  }
}
```

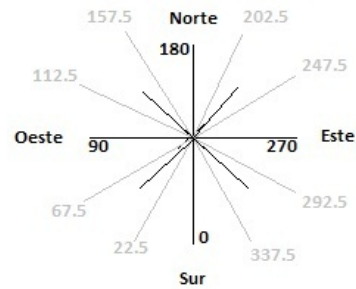


```

    Serial.print("Noreste");
}
if((angulo > 247.5) && (angulo < 292.5 )){
    Serial.print("Este");
}
if((angulo > 292.5) && (angulo < 337.5 )){
    Serial.print("Sureste");
}
delay(100);
}

```

Este es el código que nos ayudará a detectar los puntos cardinales, recordad que estamos sobre los campos magnéticos de la tierra y por lo tanto no coinciden del todo con las coordenadas geográficas. En el código vemos como calculamos el ángulo y según éste decidimos a que coordenada corresponde, el significado de los ifs plasmado en una imagen sería:



- < Anterior
- Siguiente >