|  |
| --- |
| **UNSAM, ECYT** |
| Electrónica Digital II: Trabajo práctico Final |
| Implementación de sensor de pequeños campos magnéticos con módulo “Blue Pill” |
|  |
| ***Alumno: Carlos Martín Penizzotto*** |
| ***18/07/2023*** |

|  |
| --- |
|  |

**Profesores:**

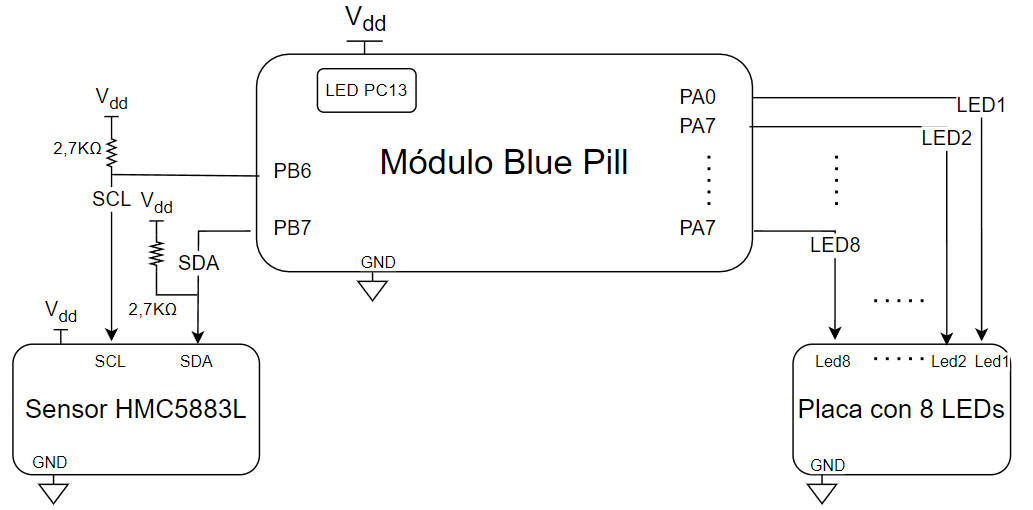
* *Miguel Ángel Sagreras*
* *Nicolás Álvarez*

1. **Objetivo**

El objetivo de este trabajo práctico es realizar la implementación de un sensor de pequeños campos magnéticos en 3 ejes con un microcontrolador STM32F103C8T6 a través del protocolo de comunicación I2C. El resultado de la medición se muestra mediante el encendido de un LED entre 8 LEDs dispuestos en forma de circunferencia, lo que da una idea de la dirección y sentido del campo magnético resultante sobre el plano en donde el sensor está apoyado.  
La componente Z del campo respecto a ese plano es medida y almacenada, y sólo se la utiliza como indicador (a través de otro LED) de si la medición indicada por los LEDs es confiable o no, ya que si gran parte del campo está contenido en ese eje Z, entonces la medición sobre el plano XY no es confiable.  
Al medir campos magnéticos pequeños, se podrá medir en todo momento el campo magnético de la tierra, de manera que el proyecto puede funcionar como brújula.  
Toda la programación del firmware se realiza en formato “bare metal”.

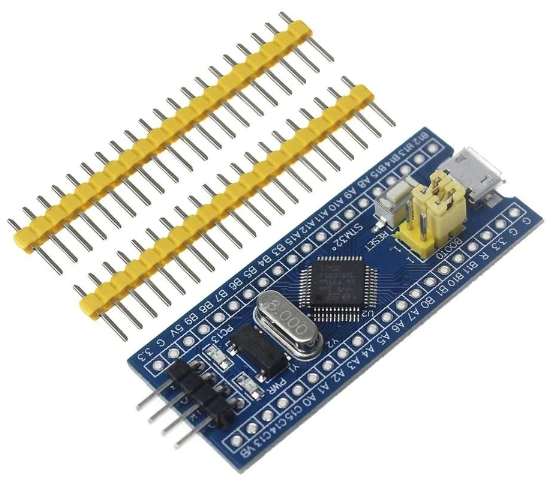
1. **Introducción**
   1. **Diagrama en bloques**

El diagrama en bloques del proyecto es el siguiente:



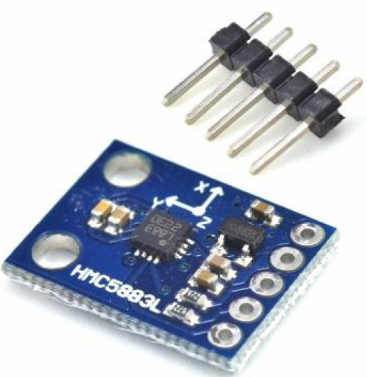
*Figura 1: Diagrama en bloques del proyecto.*

El microcontrolador a utilizar es el STM32F103C8T6 de la empresa STMicroelectronics. El mismo está incorporado en la placa de desarrollo conocida como “Blue Pill”, y en este proyecto trabaja a una frecuencia de 8MHz gracias a un oscilador RC interno.

  
*Figura 2: Placa de desarrollo “Blue Pill”.*

El sensor de pequeños campos magnéticos elegido fue el HMC5883L, de la empresa Honeywell. El mismo puede medir campos magnéticos de hasta aproximadamente 8 Gauss en los 3 ejes del espacio. Como referencia y respecto al uso como brújula, el campo magnético de la Tierra tiene una intensidad aproximada de entre 0,25 y 0,65 Gauss.

El sensor se puede adquirir ya montado en una pequeña placa de desarrollo, listo para su uso, que evita al usuario tener que realizar soldaduras de montaje superficial.

  
*Figura 3: Placa de desarrollo del sensor HMC5883L*

En cuanto a la placa de LEDs, se diseñó una placa con 8 LEDs color verde en forma de circunferencia, con una resistencia de 150 Ohms en serie a cada uno. La corriente que circula por ellos al encenderse es de aproximadamente 8 mA, lo cual es soportado correctamente por los pines de GPIOs del microcontrolador. La placa cuenta con 8 pines Dupont para los 8 LEDs, más un pin para masa, los cuales no se ven en la figura 4 porque están del lado de abajo.



*Figura 4: Placa de LEDs*

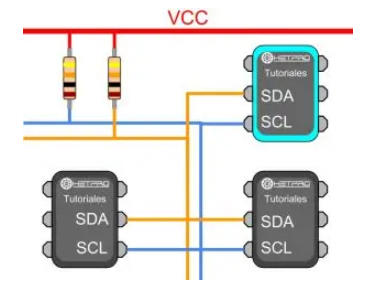
* 1. **El protocolo I2C**

“Inter-Integrated Circuit” es un bus de comunicación serial sincrónica desarrollado por Phillips Semiconductor en 1982, con motivo de poder conectar varios circuitos integrados en una misma placa.

Consta de apenas dos líneas, una de datos llamada SDA (Serial Data) y una de clock llamada SCL (Serial Clock). En este protocolo se pueden diferenciar dos roles: el maestro y el esclavo. El maestro se encarga de suministrar el clock, y también de iniciar y parar la comunicación. Sólo puede haber un maestro en un mismo instante de tiempo. En una típica comunicación, un maestro suministra el clock y manda un bit de start para iniciar la comunicación. Continua poniendo en el bus SDA la dirección del periférico con quien se quiere comunicar, y espera su bit de ACK. Luego, se envía la dirección de memoria que se quiere leer o escribir del esclavo, y a continuación se envía el dato a escribir, o se recibe el dato a leer.

Básicamente tiene dos modos de operación: maestro transmisor con esclavo receptor, o maestro receptor con esclavo transmisor. Las tramas exactas serán analizadas en el apartado de desarrollo del firmware.

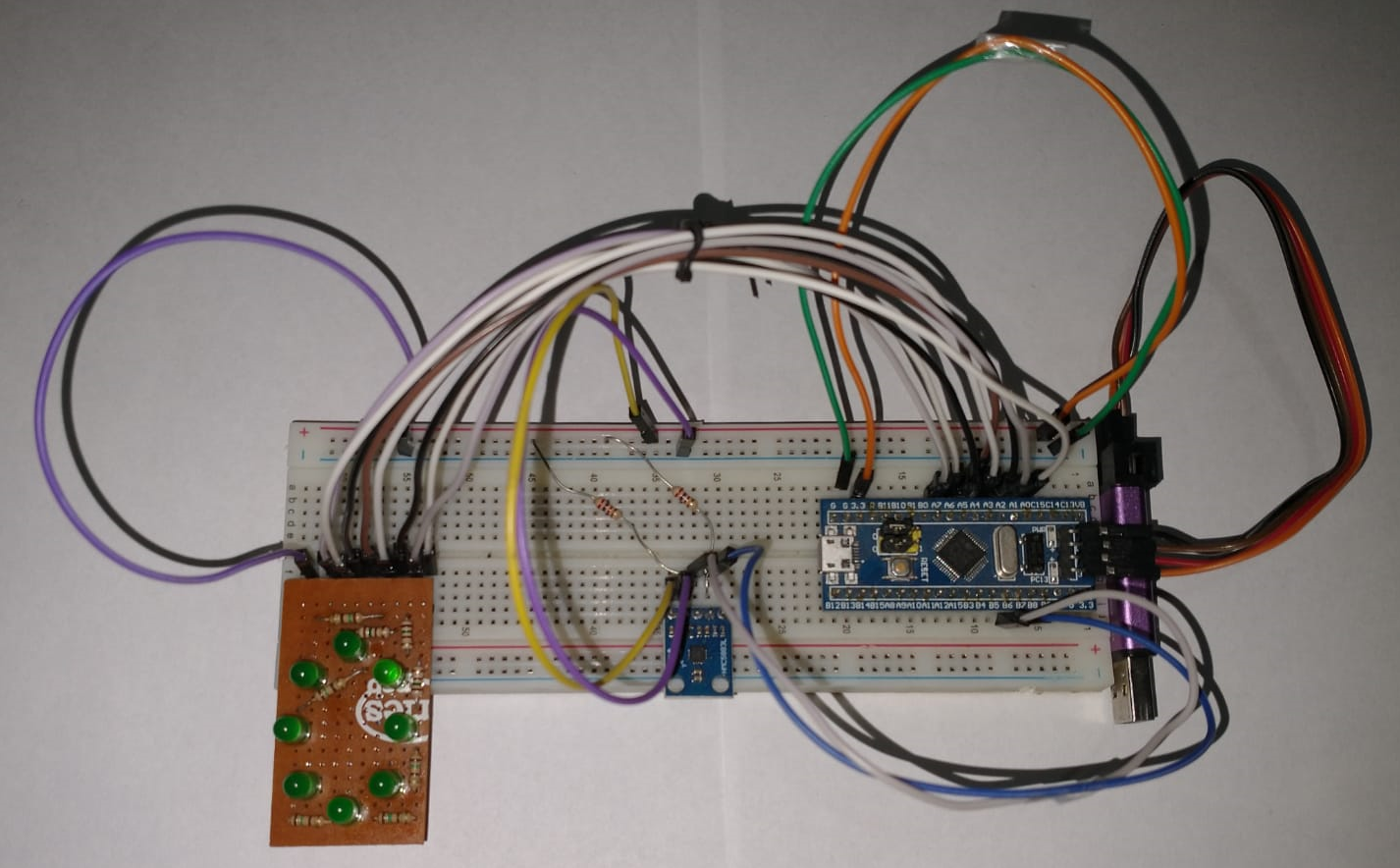
El protocolo exige que ambas líneas, SDA y SCL, dispongan de una resistencia de pull up, de manera que el estado natural de ambas líneas sea alto. Por ello, es también necesario que todas las salidas que operen con I2C sean del tipo open drain, para que puedan aceptar el estado alto natural proveniente del exterior.



*Figura 5: Ejemplo de conexionado I2C (fuente:* *hetpro-store.com)*

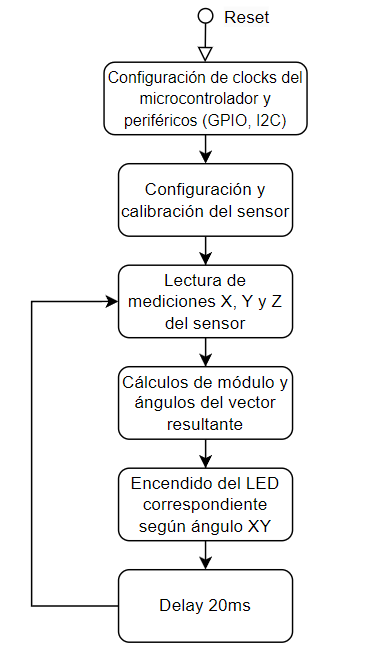
1. **Desarrollo**
   1. **Interconexión del hardware**

La interconexión de los módulos de hardware correspondientes al microcontrolador, sensor y LEDs se realizó mediante una placa de prototipo y cables tipo Dupont. Todas las conexiones siguen el diagrama en bloques mostrado en la figura 1. Se muestra a continuación la implementación final:



*Figura 6: Interconexión del hardware*

* 1. **Desarrollo del firmware**
     1. **Diagrama de flujo**

****

*Figura 7: Diagrama del flujo del código.*

* + 1. **Configuraciones de clocks del microcontrolador y periféricos.**

Dentro del código se utilizaron las siguientes funciones:

* void config\_clocks\_puertos(void)

Función: Deja al clock del sistema por defecto (8MHz con oscilador RC, sin PLL). Activa el clock del puerto C (usado para controlar el LED del PC13), del puerto B (usado para generar las señales I2C) y del puerto A (usado para encender los LEDS indicadores de campo).

* void configurar\_systick(void)

Función: Configura el systick para operar con 1MHz como input clock. Carga su valor de cuenta con 0x4E20, tal que la demora de tiempo generada sea de 20ms. Activa su interrupción y también la del NVIC.

* void configurar\_i2c(void)

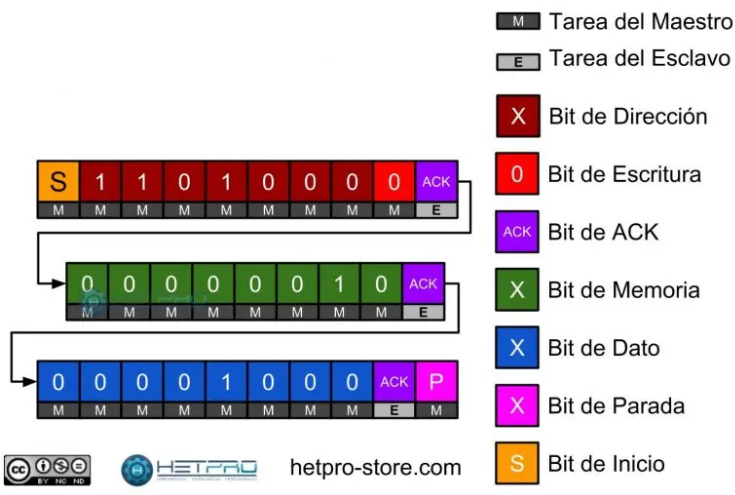
Función: Configura los puertos B6 y B7 para que funcionen como salida, en modo “Alternate function Open Drain”.

También, activa el clock del módulo I2C, y configura al mismo para lograr una frecuencia de 100KHz de clock en SCL, y un rise time máximo de 9 ciclos de clock de sistema (125ns \* 8= 1000ns, que es el máximo rise time permitido para 100KHz por el protocolo I2C. Se suma un número más al 8 por instrucción del fabricante del microcontrolador en su manual). El rise time de SCL puede permitir un valor máximo, para considerar aquellas capacidades parásitas que resulten del bus que puedan llegar a ralentizar la señal cuadrada.  
Por último, la función enciende el periférico I2C.

* + 1. **Configuraciones del sensor**
* void configurar\_sensor(void)

Función: Utiliza las funciones desarrolladas para I2C para configurar en el registro de configuración “A” del sensor un promedio de 8 muestras en cada salida, y una frecuencia de salida de datos de 15 Hz. También, se configura en el registro de configuración “B” una ganancia de (1/330) Gauss/cuenta, y por último, en el registro de configuración de modo, el modo de medición continua (de lo contrario el sensor sólo mide una vez).

* + 1. **Trama I2C de datos para escritura**

****

*Figura 8: Trama de datos para escritura (fuente hetpro-store.com)*

Se muestra a continuación una secuencia de escritura del registro de configuración “A” del sensor, la cual se lleva a cabo con las siguientes funciones:

enviar\_start(); //Setear el start bit

enviar\_direccion(ADDRESS\_HMC5883\_WRITE); //Enviamos direccion de escritura del sensor

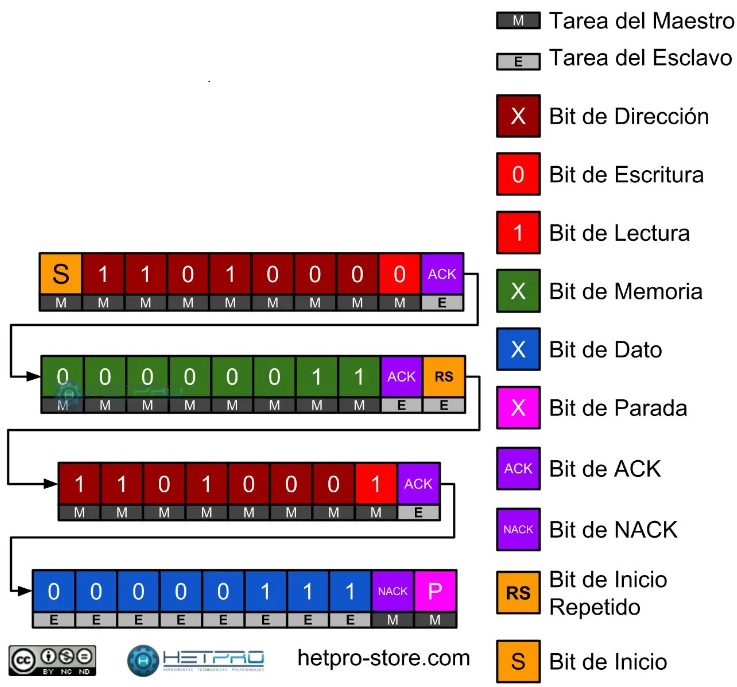
escribir\_dato(00); //Enviamos "byte de direccion de memoria"

escribir\_dato(0b01110000); //Configuramos 8 muestras promediadas en cada

output, 15 Hz de frecuencia de output,y modo "normal" de medición.

enviar\_stop(); //Enviamos el bit de stop

* + 1. **Trama I2C de datos para lectura**

****

*Figura 9: Trama de datos para lectura (fuente hetpro-store.com)*

Para realizar un comando de lectura, el primer comando (dado por el bit menos significativo de la dirección del periférico) debe ser de escritura, debido a que se “escribe” en el dispositivo a leer la dirección de su registro cuyo valor se quiere leer. Una vez obtenido el ACK por parte del esclavo, se debe enviar un bit de re-start, o en su defecto, parar la comunicación y volverla a iniciar con un nuevo bit de start. Este último método fue implementado en el código.  
Se muestra a continuación un ejemplo de lectura de datos del registro X del sensor, utilizando la trama mostrada en la figura 9, y las funciones desarrolladas para ese fin:

enviar\_start(); //Setear el start bit

enviar\_direccion(ADDRESS\_HMC5883\_WRITE); //Enviamos direccion de escritura del sensor

escribir\_dato(registro); //Byte de dirección de memoria a leer

enviar\_stop(); //En lugar de enviar bit de re-start, enviamos bit de stop y luego volvemos a iniciar la comunicación.

enviar\_start();

enviar\_direccion(ADDRESS\_HMC5883\_READ); //Enviamos dirección de lectura del sensor.

datos=leer\_dato(); //Leemos el byte y lo dejamos en la variable "datos" para retornarla.

enviar\_stop();

* + 1. **Calibración del sensor**

El sensor HMC5883L cuenta con un modo de auto-testeo y calibración, en donde el mismo puede generar internamente un campo de valor conocido, el cual se superpone al campo externo que haya al momento de calibrar. El sensor es capaz de primero tomar una medición del campo externo, y luego otra con tanto campo externo como de calibración, y luego restar ambos valores. La hoja de datos indica que, configurando una ganancia de 1/660 Gauss/cuenta, se espera tener en los registros de medición X e Y el valor 766, y en el de Z, el valor 713. Una vez obtenidos los resultados, podremos dividir el valor esperado por el resultado para generar un coeficiente de calibración.

Los valores de calibración obtenidos fueron:

xcal=0.876932174, ycal= 0.930176407, zcal= 0.88133496

Cabe remarcar que esta calibración sólo opera con los elementos sensibles internos al sensor, de manera que es independiente del ambiente externo. Por ello, sólo es necesaria correrla una única vez. Es decir, si el ambiente presente tiene ruido magnético o interferencias, la función de calibración no mejorará las condiciones de medición pues no opera con eso. Sólo mejora la exactitud de los elementos sensibles.

Actualmente, esta función está comentada en el firmware.

1. **Conclusiones**

Este trabajo práctico me ha servido como alumno para aprender más sobre el protocolo I2C y su forma de aplicarlo. El usar el formato bare metal obliga al usuario a conocer cada detalle de la comunicación, ya que cualquier pequeño error de implementación puede resultar en que la comunicación falle.  
El código realizado podría tener una gran variedad de aplicaciones en el futuro, ya que, una vez guardados los datos de la medición de campo magnético, se le pueden dar muchos más usos que solamente mostrarlos con LEDs. Además, la estructura del código podría llegar a utilizarse para otro sensor distinto, ya que con pequeños cambios de definiciones y valores numéricos, se podría tener al nuevo sensor funcionando en muy poco tiempo de desarrollo.

El funcionamiento final del proyecto como brújula es bastante estable, aunque, por supuesto, sufre de interferencias magnéticas en ciertos casos. También puede usarse al proyecto como indicador del sentido y dirección de campos magnéticos más fuertes, como el de un imán de neodimio. En ese caso, su funcionamiento es aún más estable.