MEMORIA PROYECTO ...

Sistemas Empotrados y Distribuidos

Ignacio Gago, Pablo Gordillo y Miguel Isabel

MÁSTER EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID
2015-2016

May 17, 2016

Contents

1 Introducción **1**

2 Componentes Utilizados **3**
2.1 Arduino UNO 3
2.2 Discovery STM32F429 3
2.3 Módulo BH1750 3
2.4 Módulo DTH22 4

3 Protocolo de Comunicación **6**

4 Modelado del Sistema **7**

5 Conexiones **8**
5.1 Modelo humedad-temperatura 8
5.2 Modelo fotovoltaico 8
5.3 Modelo Intercomunicación 8

6 Implementación **9**

7 Referencias **11**

1 Introducción

Hoy en día, cada vez es más común escuchar el término *smart* en nuestra vida cotidiana. Nos estamos dirigiendo a un mundo más automatizado y en continua interacción con la tecnología. Este término se aplica en distintas áreas como por ejemplo la domótica, los sistemas móviles, e incluso el diseño. El objetivo asociado al concepto *smart* es automatizar distintas acciones que antes eran llevadas a cabo por el ser humano.

Las *smart homes* son viviendas automatizadas que aportan servicios de gestión energética, seguridad, bienestar y comunicación. Muchas de estas casas usan sensores de temperatura para regular el funcionamiento de la calefacción y de la caldera, mediante telefonía móvil, Wi-Fi o Ethernet. También controlan los toldos y persianas eléctricas, realizando algunas funciones repetitivas automáticamente o bien para protegerlos del viento.

Gracias a este tipo de tecnologías, también se produce un ahorro energético debido a la racionalización de cargas eléctricas y la gestión de tarifas. Además, se usan sensores de luz y sol para regular el uso de las luces de la casa.

Otro de los conceptos de moda relacionados con este término es el de las *smart cities* o *ciudades inteligentes* con las que se pretende automatizar distintos procesos que permitan mejorar la calidad de vida en las ciudades y avanzar hacia desarrollos más sostenibles apoyados en la innovación y la tecnología. Este concepto tiene también asociado, en muchas ocasiones, una participación ciudadana activa.

Un ejemplo lo encontramos en la ciudad de Barcelona, considerada como una de las principales *smart cities* del mundo. En ella podemos encontrar sensores que miden el ruido del tráfico, analizan la calidad del aire, controlan el aparcamiento e incluso actúan como mecanismos de iluminación inteligente. Además los peatones también son detectados gracias a sensores inteligentes a través de los cuales poder observar la actividad y hábitos de los mismos e identificar las zonas de la ciudad más transitadas.

Dentro de Barcelona, también se puede destacar un proyecto encabezado por el Fab Lab del Instituto de Arquitectura Avanzada de Cataluña, que desarrolla un dispositivo capaz de medir los niveles de contaminación y de ruido del entorno. Este dispositivo tendría que ser portado por cualquier habitante de Barcelona y permite también compartir los datos recogidos en tiempo real.

No obstante, existen otras ciudades que tienen implantados sistemas que

siguen una política de funcionamiento similar: Bilbao tiene instalados medidores de CO_2 en distintas zonas de la ciudad. Estos sensores captan los niveles de CO_2 en el ambiente y posteriormente son enviados a un centro de procesamiento donde se almacenan y se llevan a cabo diversos cálculos. Una vez los datos han sido procesados, estos son utilizados por el Ayuntamiento para restringir el tráfico en diversas zonas de la ciudad o llevar a cabo políticas medioambientales en las zonas más afectadas.

Debido al auge de las *smart cities*, se ha elegido como proyecto de la asignatura la simulación de una pequeña estación en la que dos sistemas independientes (Arduinos Uno) llevarán a cabo mediciones de distintos factores ambientales a través de diversos sensores y posteriormente estos datos serán enviados a un centro de procesamiento de datos (Discovery STM32F249) donde serán analizados y mostrará la información recibida por los Arduinos.

Los sensores seleccionados, por su aplicabilidad tanto en el ámbito de las *smart homes* como en el de las *smart cities* son: un sensor de temperatura y humedad, con el que se puede obtener fácilmente el índice de calor y un sensor de luz. El proyecto elegido es una simplificación de posibles usos que se les puede dar a estos sensores en las *smart cities* de hoy en día como se ha expuesto anteriormente en el caso de Barcelona.

2 Componentes Utilizados

Como se ha expuesto en la Sección 1, el objetivo del proyecto de la asignatura es realizar un sistema de medición de temperatura, humedad y luz a través de dos placas Arduino Uno. Los datos recogidos por cada una de las placas serán enviados al tercer componente del proyecto, una placa Discovery STM32F429, la cual se encargará de procesarlos y mostrar los datos recibidos por su pantalla.

El sensor de humedad y temperatura utilizado es el modelo DHT22 y estará conectado al primer Arduino a través de una placa de prototipado. Este Arduino se encargará de recibir tanto los datos de la humedad y la temperatura, procesarlos como se explica en la Sección 2.4 y enviarselos a la Discovery utilizando un protocolo de comunicación(Sección 3) desarrollado por el grupo.

El sensor fotovoltaico utilizado es el modelo BH1750 y estará conectado al segundo Arduino a través de otra placa de prototipado. Este se encargará de mendir la luz existente en el ambiente y se la enviará a la Discovery.

Finalmente la placa Discovery recibirá los datos enviados por los Arduinos a través de dos de sus UART y utilizando el protocolo de comunicación desarrollado, los procesará y mostrará el resultado a través de su pantalla LCD.

FiXme Fatal: Pablo: Introducir aqui la maquina de estados o la red de Petri...

La máquina de estados que sigue el sistema se muestra a continuación. En ella se describe el comportamiento explicado anteriormente...

Todo el código utilizado para el desarrollo del proyecto puede encontrarse en `https://github.com/tutugordillo/SED`.

2.1 Arduino UNO

2.2 Discovery STM32F429

2.3 Módulo BH1750

El componente BH1750 GY-302 es un sensor fotovoltaico que permite llevar a cabo mediciones digitales de la intensidad de luz ambiental. Se trata de una versión mejorada de los sensores analógicos de luz formados por fotoresistores LDR, pequeñas resistencias que varían en función de la luz que incide sobre las mismas.

El sensor devuelve un valor medido en Lux. Un Lux es la unidad derivada

del Sistema Internacional de Unidades para el nivel de iluminación y equivale a un $lumen/m$. Las mediciones se pueden llevar a cabo en dos modos: *modo medición una vez* y *modo medición continua*. Además existen distintas resoluciones (1lx, 0.5lx, 4lx) con las que llevar a cabo las mediciones y de esta manera ajustar la precisión con la que se lleva a cabo la medición. Así el módulo es capaz de detectar un mínimo de 0.11 lx y un máximo de 100000 lx.

De la combinación del modo de medición y la resolución se obtienen 6 modos de ejecución:

- Continuosly H-Resolution Mode
- Continuosly H-Resolution Mode2
- Continuosly L-Resolution Mode
- One Time H-Resolution Mode
- One Time H-Resolution Mode2
- One Time L-Resolution Mode

Los modos High(H) trabajan con resoluciones de 1 lx y 0.5 lx

La comunicación se lleva a cabo a travÃĺs de I2C. En función de como se encuentre conectado el pin ADDR del módulo se puede trabajar con dos direcciones:

- Si se encuentra conectado al pin A3 o a alimentación se trabajará con la dirección 0x5C.
- Si se deja sin conectar o conectado a GND la comunicación I2C se llevará a cabo mediante la dirección 0x23.

2.4 Módulo DTH22

El módulo DTH22 es un sensor de humedad y temperatura digital. Normalmente el sensor cuenta con 4 pines, uno para la alimentación, otro para tierra, uno de datos y un cuarto que no se utiliza. En este caso se ha utilizado el modelo AOSONG AM2302 con la ventaja de que únicamente posee los tres pines hábiles y posee una resistencia interna permitiendo su funcionamiento sin necesitar ningún otro componente.

Las mediciones se llevan a cabo en conjuntos de 40 bits donde los 16 primeros hacen referencia a la humedad, los 16 siguientes a la temperatura y los 8 últimos como *checksum*. Todas las mediciones se realizan de manera digital. La comunicación comienza con una señal del microcontrolador al sensor indicando que comienza la transmisión. Mientras no se reciba esta señal el sensor se encuentra en estado de *stand-by*. Tras recibir la señal de comienzo el sensor responderá enviando los 40 bits de datos tras los cuales volverá al estado de *stand-by* hasta que reciba una nueva señal de comienzo de la transmisión. El proceso completo de comunicación se lleva a cabo en unos dos segundos. La transmisión de todos los bits por parte del sensor al microcontrolador comienza con una señal baja (LOW) de 50 μs y posteriormente una señal alta (HIGH) en función de cuya longitud el bit transmitido será un 1 o un 0:

- Si la señal se transmite durante un tiempo inferior a 30 μs el bit transmitido será un 0.
- Si la señal HIGH se transmite durante 70 μs el bit transmitido será un 1.

Para conocer el número de ciclos transmitidos se hace de forma relativa, es decir, se mide el número de ciclos propios de LOW y se compara este valor con el número de ciclos de HIGH, de esta forma se puede saber si se ha trasmitido un 1 o un 0. Además, durante este cálculo se desactivan todas las interrupciones para que no varíen el valor final con ciclos que interfieren en la medición.

3 Protocolo de Comunicación

El protocolo de comunicación utilizado tanto por la Discovery como por los Arduinos para llevar a cabo la transimisón de los datos ha sido desarrollado por los integrantes del grupo.

La comunicación se lleva a cabo a través de las UART de cada uno de los tres dispositivos y las conexiones pueden ser encontradas en la sección 5.3.

Para facilitar el proceso de comunicación se ha definido una estructura de datos que contiene el tipo de datos que se envía (luz, temperatatura o humedad) y su valor numérico asociado. De esta forma, el primer byte que se envía indica el tipo de dato, y a continuación viene el valor numérico que es un float puesto que es el tipo más pequeño que contiene cualquiera de los tres tipos de datos. Además, también han sido implementadas las funciones de envío y recepción de datos a través de un puerto serie implementado en la librería SoftwareSerial.

4 Modelado del Sistema

En la Figura 4, se muestra el modelo del sistema que se va a desarrollar como una máquina de estados con cinco componentes diferencidos:

- Las componentes BH1750 y DTH22 están compuestas por dos estados: En *S0* el sensor está midiendo la magnitud física correspondiente y en *S1* el sensor envía los datos medidos al Arduino Uno al que está conectado, volviéndose al estado *S0* para comenzar una nueva medición.
- Los Arduinos Uno están compuestos por tres estados: En *S2* se reciben los datos del sensor, en *S3* se procesan para mandarlos de manera unificada usando la estructura de datos Data (definida en Comm.h) y, finalmente, en *S4* se envían los datos a la Discovery, volviendo a esperar nuevos datos en *S2*.
- La Discovery SMT32 está formada por tres estados: *S5* espera a recibir datos de uno de los Arduinos. Una vez que los recibe, los procesa en el estado *S6* y, finalmente, se produce la visualización de los mismos en *S7*.

Este proyecto es un sistema completamente automatizado y, por lo tanto, la interacción con el usuario es inexistente. Esto lleva a que no haya diagramas de casos de uso para el usuario. Por la misma razón, el diagrama de secuencia es completamente similar a flujo de eventos esperados por la máquina de estados que define el sistema. No obstante, durante la demo que se realice en clase, sí que habrá interacción, ya que se provocará la variación de las magnitudes físicas medidas por los sensores involucrados en el proyecto.

5 Conexiones

En esta sección se muestra la interconexión física entre los distintos componentes utilizada en el proyecto.

5.1 Modelo humedad-temperatura

En la Figura 5.1 se muestra las conexión física entre el ArduinoUno y el sensor de humedad y temperatura DHT22. Las mediciones llevadas a cabo por el sensor son digitales, razón por la que se ha utilizado el pin 2 de la placa Arduino, que es conectado al pin de datos del sensor a través del cable amarillo.

Los cables azul y verde unen el sensor a tierra y a alimentación respectivamente. El pin 3 del sensor (comenzando por la izquierda) no se utiliza como se explicó en la Sección 2.4.

5.2 Modelo fotovoltaico

La conexión física entre el modelo foltovoltaico y el Arduino Uno puede ser observado en la figura 5.2. Los cables azul y negro unen el sensor a tierra y a alimentación, respectivamente. Al igual que en el modelo anterior, las mediciones de luminosidad son realizadas de manera digital, por tanto, SDA y SCL están unidos a los puertos A4 y A5, respectivamente.

5.3 Modelo Intercomunicación

6 Implementación

En esta sección se presentan los detalles de la implementación llevada a cabo así como determinadas decisiones que se han tomado exponiendo la razón de las mismas.

El primer arduino se encargará de medir tanto la temperatura como la humedad. Para la comunicación se ha utilizado la librería SerialSystem.

El código que implementa la lectura de datos recibidos tanto del sensor de temperatura como del de luz es inmediato según ha sido explicado en la sección 2. En ambos casos, se iniciliza el Serial que se utiliza para transmitir los datos en la función *setup* y la lectura de los mismos en la función *loop*.

En el sensor de temperatura, la lectura de los datos se hace de manera directa, sobre las direcciones de memoria en las que escribe el sensor. Mientras que enla lectura de la luz, se utiliza la librería Wire para la comunicación entre el sensor y el Arduino.

Finalmente los detalles de implementación sobre la placa STM32F429I-Discovery están relacionados con la comunicación de la misma con los dos Arduinos utilizados y la utilización de su pantalla LCD. En cuanto a la comunicación con los Arduinos se ha utilizado dos de las UARTS disponibles por la placa. La STM32F429I dispone de varias UARTS y USARTS. La diferencia entre las mismas es que las USARTS amplían la funcionalidad de las UARTS permitiendo la comunicación síncrona entre la Discovery y el otro dispositivo involucrado. En nuestro caso ambas nos permiten llevar a cabo los objetivos marcados. Para implementar la comunicación se ha utilizado el protocolo descrito en la Sección 3. Para el funcionamiento de las dos UARTS escogidas (una por Arduino) se ha utilizado la librería de UARTS de sSTM. En primer lugar se lleva a cabo la inicialización de las mismas en las que se especifican los atributos de la comunicación como son el ancho de banda (9600 baudios en nuestro caso), la paridad o el número de bits de finalización. Posteriormente comienza un bucle de espera activa para identificar la recepción de los datos por parte de cada uno de los arduinos, tras lo cual se procesan los datos recibidos acorde al protocolo descrito y se procesan los mismos.

En lo referente a la LCD se ha utilizado BSP (Board Support Package), un middleware de STM que nos permite controlar distintos periféricos de la placa. Lo que hace BSP es mejorar la legibilidad y mejorar la interacción entre el programador y las librerias que manejan los drivers de los distintos periféricos. Si inspeccionamos el código existente en BSP referente a la

pantalla LCD (...) se puede observar que utiliza directivas existentes en la librería de STM para la pantalla LCD e implementa nuevas directivas que facilitan la programación de la pantalla LCD permitiendo por ejemplo mostrar un String llamando directamente a una directiva a la que hay que especificarle el número de línea en la que se quiere que se pinte y el String. Esta directiva internamente lo que hace es enviar a la LCD caracter por caracter el String especificado utilizando directivas de la librería $stm32f4xx\_hal\_ltdc$ calculando de manera automática la posición en la que se dibuja cada caracter.

El controlador de la pantalla LCD soporta 2 contenedores como máximo en los que dibujar. Estos contenedores se encuentran almacenados en la memoria SDRAM externa de la placa. El contenido se guarda de forma matricial de tal forma que se trabaja con una matriz 320x239 (o su equivalente vectorial), la resolución de la pantalla LCD . Cada una de las posiciones de esta matriz representa un bit y lo que hacen las funciones de BSP es escribir cada pixel del color indicado con la finalidad de acabar formando los caracteres específicos.

La implementación y código del proyecto se encuentra disponible en `https://github.com/tutugordillo/SED` dentro del directorio *Proyecto Final*.

7 Referencias

- Datasheet del sensor DHT22. [https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf]
- Datasheet del sensor BH1750. [http://www.alldatasheet.com/datasheet-pdf/pdf/338083/ROHM/BH1750FVI.html]
- Información de la librería SoftwareSerial.[https://www.arduino.cc/en/Reference/SoftwareSerial]
- Getting started with the STM32F429 Discovery kit. [http://www.st.com/resource/en/user_manual/dm00107720.pdf]
- Description of STM32F4xx HAL drivers. [http://www.st.com/resource/en/user_manual/dm00105879.pdf]
- Librería de USART para Discovery. [http://stm32f4-discovery.com/2014/04/library-04-connect-stm32f429-discovery-to-computer-with-usart/]
- Librería para el display de la Discovery. [http://www.pierpaolobagnasco.com/2014/07/13/stm32f429-discovery-display/]

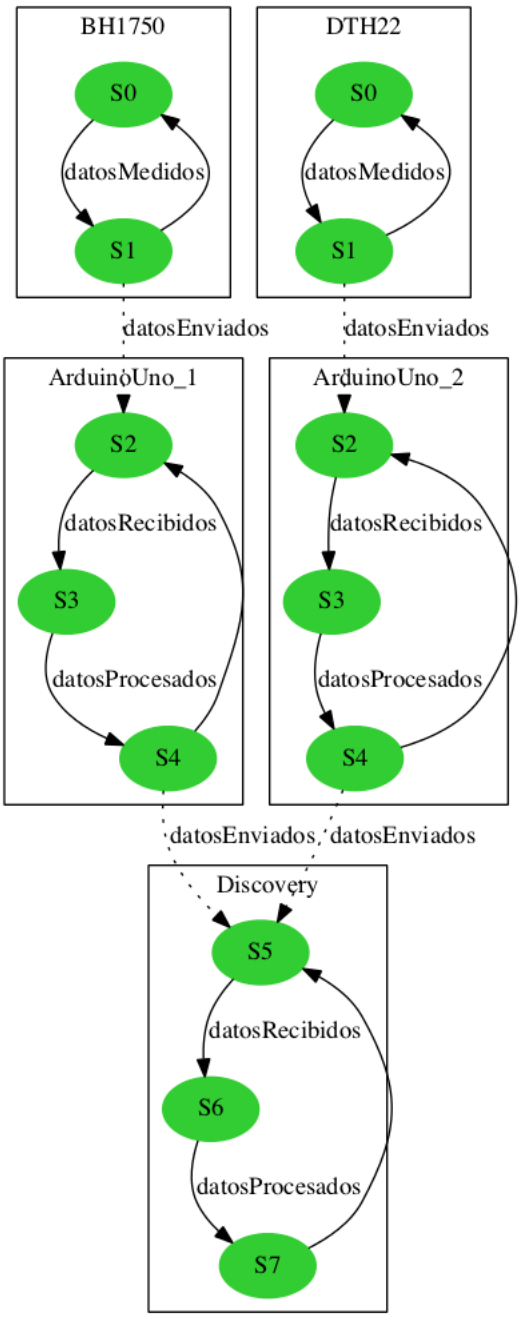

Figure 1: Máquina de estados

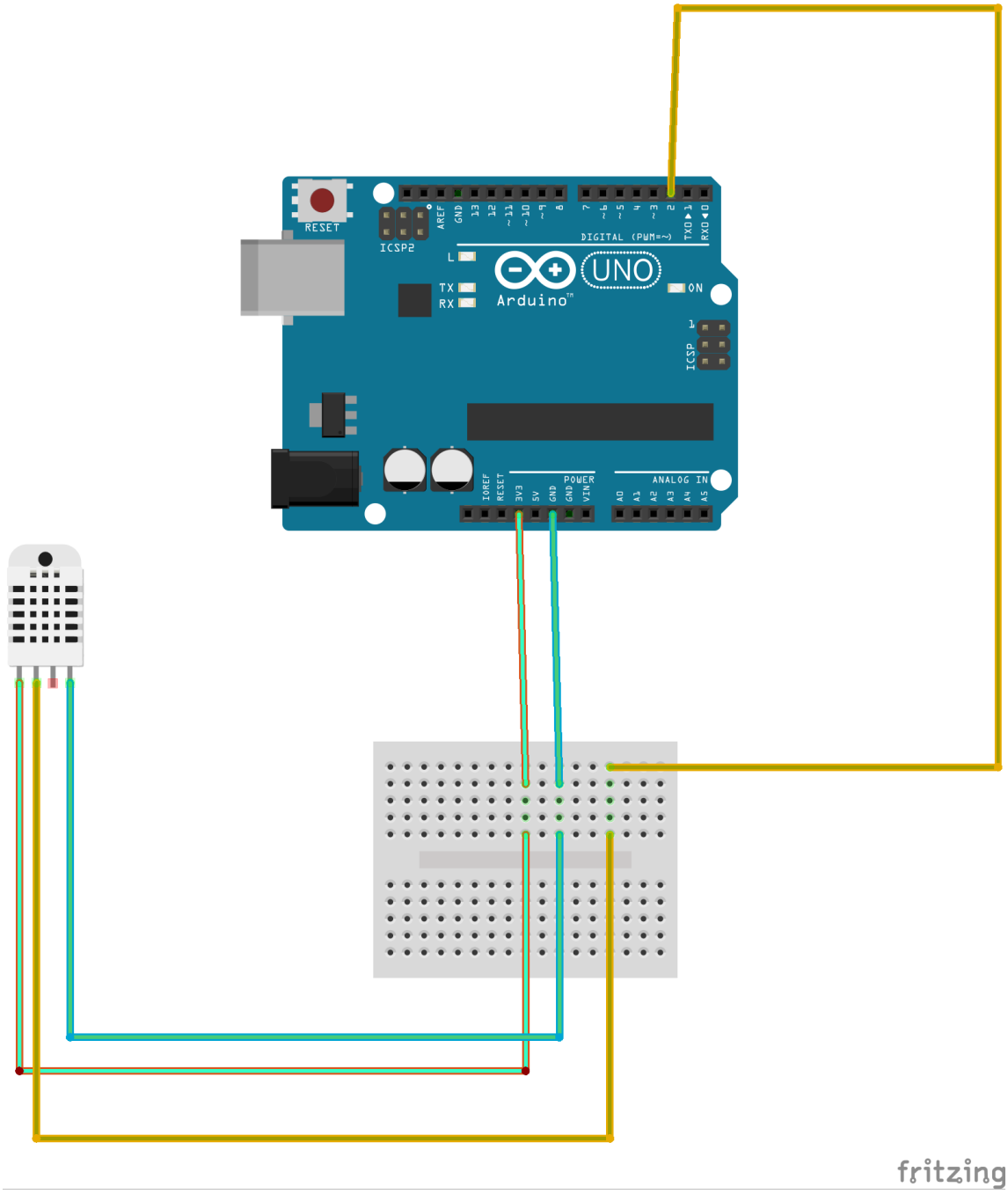

Figure 2: Conexón entre el sensor DHT22 y ArduinoUNO

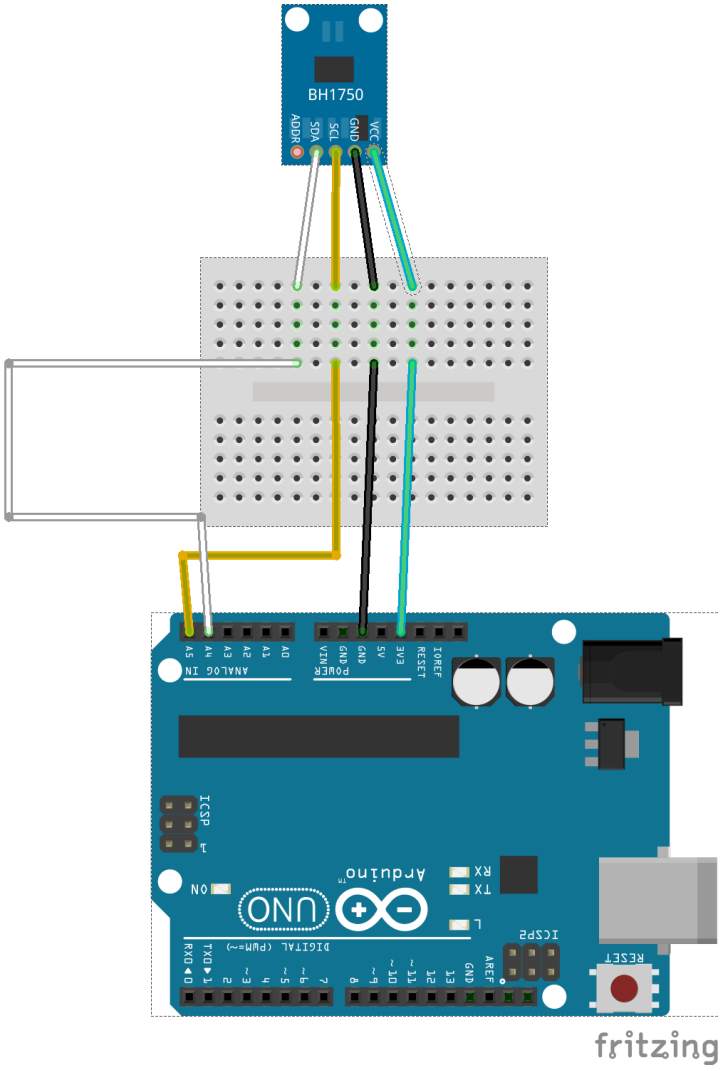

Figure 3: Conexón entre el sensor BH1750 y ArduinoUNO