



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Práctica 8: Medición de dos sensores
desde un Convertidor Analógico Digital vía
Bluetooth

Unidad de aprendizaje: Instrumentación

Grupo: 3CM4

Integrantes:

Aguilar Herrera Arianna Itzamina

Nicolás Sayago Abigail

Ramos Diaz Enrique

Profesor(a):

Tellez Barrera Juan Carlos

Fecha de entrega: 2 de diciembre de 2018

Índice

| | |
|---|-----------|
| 1 Objetivos | 3 |
| 2 Listado de materiales | 3 |
| 3 Listado de equipo | 3 |
| 4 Introducción | 4 |
| 4.1 Convertidor ADC0809 | 4 |
| 4.2 Descripción de las líneas del ADC0809 | 4 |
| 4.3 Potenciómetro | 5 |
| 4.4 Amplificador no inversor | 6 |
| 4.5 LM35 | 6 |
| 4.6 Arduino | 6 |
| 4.7 Bluetooth | 7 |
| 5 Desarrollo práctico | 8 |
| 5.1 Recordando la práctica 2: Sensor resistivo - potenciómetro | 8 |
| 5.2 Recordando la práctica 4: Convertidor analógico-digital | 9 |
| 5.3 PE 1 - Conexión de los sensores al CAD | 10 |
| 5.3.1 Esquemas | 10 |
| 5.3.2 Circuito cableado | 11 |
| 5.3.3 Funcionamiento | 11 |
| 5.4 PE 2 - Integración del módulo Bluetooth HC-05 y la placa Arduino al CAD | 12 |
| 5.4.1 Esquemas | 12 |
| 5.4.2 Circuito cableado | 13 |
| 5.4.3 Funcionamiento | 14 |
| 5.5 PE 3 - Programa en Arduino | 15 |
| 5.5.1 Código | 15 |
| 5.5.2 Explicación del código | 16 |
| 5.6 PE 4 - Aplicación Android desarrollada para móviles | 17 |
| 5.6.1 Código e interfaz gráfica | 17 |
| 5.6.2 Explicación del código | 19 |
| 5.6.3 Instalación | 22 |
| 5.7 Funcionamiento completo | 23 |
| 6 Observaciones | 28 |
| 7 Conclusiones | 29 |
| 7.1 Aguilar Herrera Arianna Itzamina | 29 |
| 7.2 Ramos Diaz Enrique | 29 |
| 7.3 Nicolas Sayago Abigail | 30 |
| 8 Anexos | 31 |

| | | |
|-----|--|-----------|
| 8.1 | LM741 | 31 |
| 8.2 | LM7805 | 32 |
| 8.3 | ADC0809 | 32 |
| 8.4 | LM35 Sensor de temperatura integrado | 33 |
| 8.5 | Inversor SN74LS14N | 33 |
| 8.6 | Arduino Mega | 34 |
| 8.7 | HC-05 | 34 |
| 8.8 | Página Web: MIT App Inventor | 35 |
| 8.9 | Arduino IDE | 35 |
| | Referencias | 35 |

1. Objetivos

- ✓ Repasar el funcionamiento y características del convertidor analógico-digital ADC0809.
- ✓ Graficar el ángulo que provoca un potenciómetro y el valor que entrega un sensor de temperatura.
- ✓ Programar una aplicación en Arduino para interpretar las señales de salida de dos sensores.
- ✓ Enviar las mediciones de ambos sensores a un dispositivo móvil, por medio de comunicación vía Bluetooth.
- ✓ Seleccionar cuál de las dos mediciones queremos desplegar en el dispositivo móvil, enviando una señal de control desde éste.

2. Listado de materiales

- 1 Convertidor Analógico Digital ADC0809
- 9 Resistores $2.7K\Omega$
- 11 Leds
- 1 Circuito Integrado 74HC14
- 1 Capacitor 1.5 nF
- 1 Resistor de $2.2K\Omega$
- 1 Diodo IN4001 a 7
- 1 Potenciómetro de $2K\Omega$
- 2 Capacitores $0.1 \mu\text{F}$
- 1 Resistor de $10K\Omega$
- 1 Resistor de $15K\Omega$
- 2 Amplificadores Operacionales LM741
- 12 Resistores de $1K\Omega$
- 1 Circuito integrado LM35
- 1 Dipswitch de 3 entradas
- 1 Regulador 7805
- 2 Presets o potenciómetros de $10K\Omega$

3. Listado de equipo

- 1 Voltímetro digital.
- 1 Fuente de alimentación.
- 1 Osciloscopio digital

4. Introducción

4.1. Convertidor ADC0809

Para nuestra práctica utilizaremos el convertidor **ADC0809**. Este es un conversor de 8 bits (la señal analógica se convierte en una palabra digital de 8 bits), que tiene la posibilidad de leer 8 señales analógicas (8 canales). Posee 28 pines de los cuales 8 corresponden a sus canales analógicos de entrada; éste solo puede leer un canal a la vez y dispone por lo tanto de un selector (multiplexor) de 3 líneas, que permite seleccionar la señal de entrada a convertir, mediante el código binario presente en estas entradas de selección. Posee un tiempo de conversión de 100 microsegundos y una entrada máxima de reloj de 500 KHz.

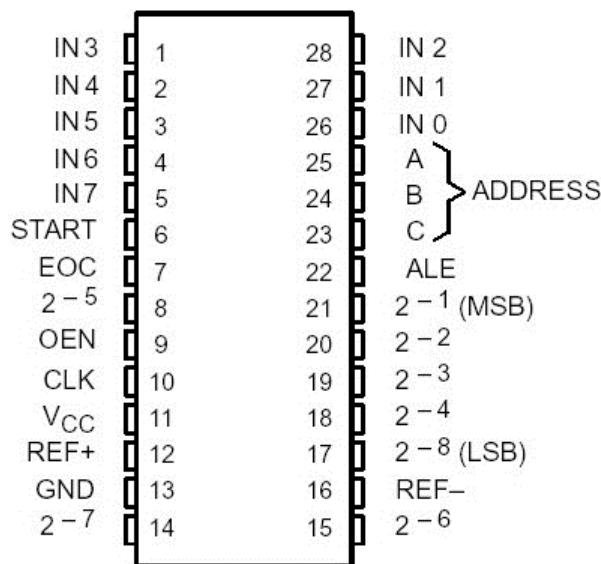


Figura 1: Circuito integrado ADC0809

4.2. Descripción de las líneas del ADC0809

Entradas Analógicas (IN0...IN7): Líneas de entrada de las señales analógicas que se quiere digitalizar.

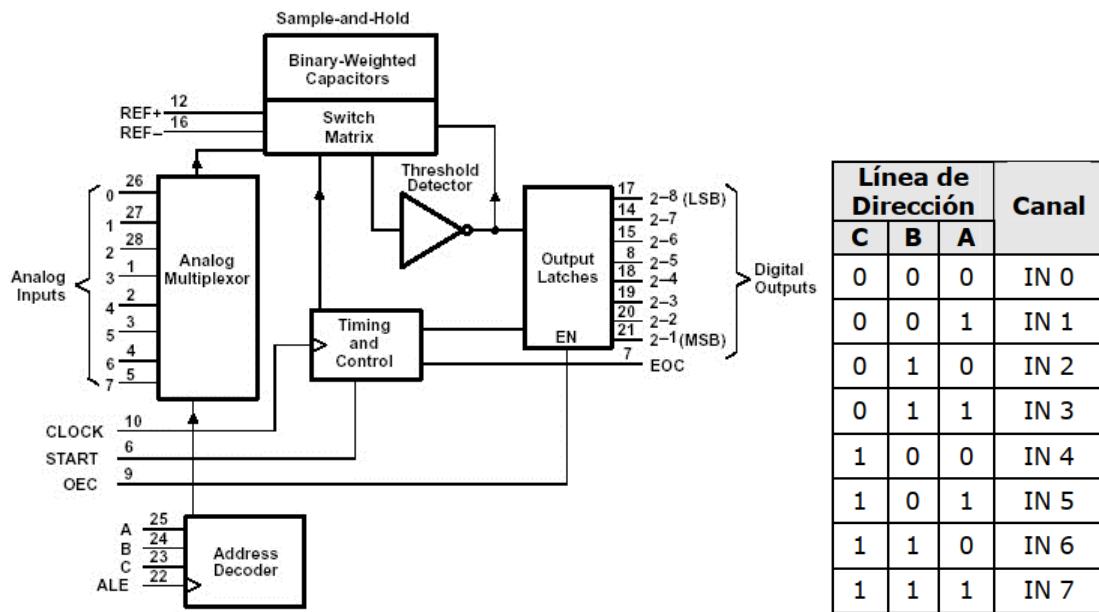
Bus de salida de datos ($2^{-8} \dots 2^{-1}$): Estas líneas de salida entregan la palabra binaria que corresponde al nivel analógico de entrada (poseen tri-state).

OE (Habilitación de salida): Con esta línea se habilita la salida. Cuando OE esta en '0' la salida permanece en tri-state, cuando esta a '1' entrega el código digital de salida.

START: Entrada para indicar al ADC que debe iniciar un nuevo ciclo de conversión.

EOC (Fin de conversión): Cuando el proceso de conversión finaliza, el ADC emite esta señal para indicar que en el bus de datos del mismo hay una palabra digital.

CLK (Clock): Es la entrada correspondiente a la señal de reloj.



4.3. Potenciómetro

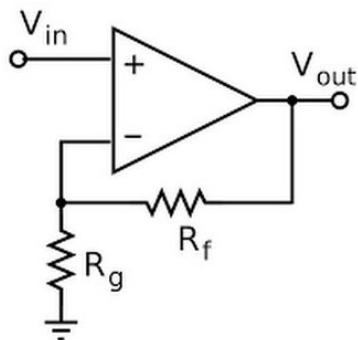
Es un resistor eléctrico con un valor de resistencia variable y generalmente ajustable manualmente. Los potenciómetros tienen tres terminales y se suelen utilizar en circuitos de corriente baja, para circuitos de mayor corriente se utilizan los reóstatos. En muchos dispositivos eléctricos los potenciómetros son los que establecen el nivel de salida. Por ejemplo, en un altavoz, el potenciómetro ajusta el volumen; en un televisor o un monitor de ordenador se puede utilizar para controlar el brillo. El valor de un potenciómetro viene expresado en ohmios (símbolo Ω) como las resistencias, y ese valor corresponde siempre es la resistencia máxima que puede llegar a tener. El mínimo valor lógicamente es cero. Por ejemplo, un potenciómetro de 5 k puede tener una resistencia variable con valores entre 0 y 5000 .



4.4. Amplificador no inversor

En un op-amp ideal, la ganancia del amplificador no inversor está dada por:

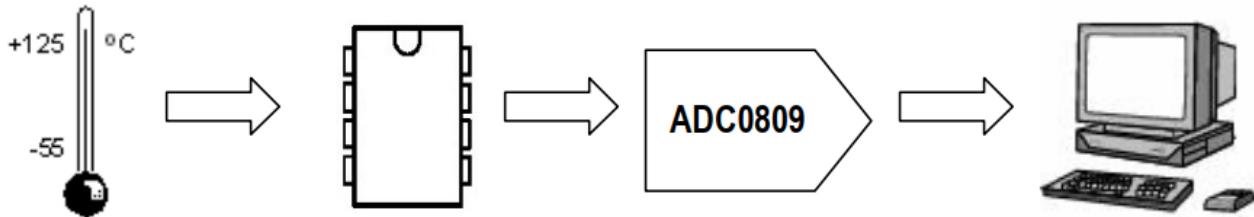
$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_f}{R_1}$$



4.5. LM35

Un tema importante cuando utilizamos un conversor AD es el acondicionamiento de la entrada. Realizaremos el acondicionamiento de una señal proveniente de un sensor de temperatura LM35. Este sensor entrega 10mV/°C, con lo que a 100°C entrega 1V.

Supondremos que mediremos la temperatura ambiente por lo que estableceremos la temperatura máxima en 50 grados centígrados a plena escala (es decir, 500 mV como salida máxima del sensor, y voltaje de entrada al amplificador). A 50 grados centígrados el voltaje a la salida de la etapa de acondicionamiento debemos tener 5V.



4.6. Arduino

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.



El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados.

4.7. Bluetooth

El Bluetooth Special Interest Group (SIG), una asociación comercial formada por líderes en telecomunicación, informática e industrias de red, está conduciendo el desarrollo de la tecnología inalámbrica Bluetooth y llevándola al mercado.

La tecnología inalámbrica Bluetooth es una tecnología de ondas de radio de corto alcance (2.4 gigahertzios de frecuencia) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como ordenadores móviles, teléfonos móviles, otros dispositivos de mano y entre estos dispositivos e Internet. También pretende simplificar la sincronización de datos entre los dispositivos y otros ordenadores.

Permite comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros.



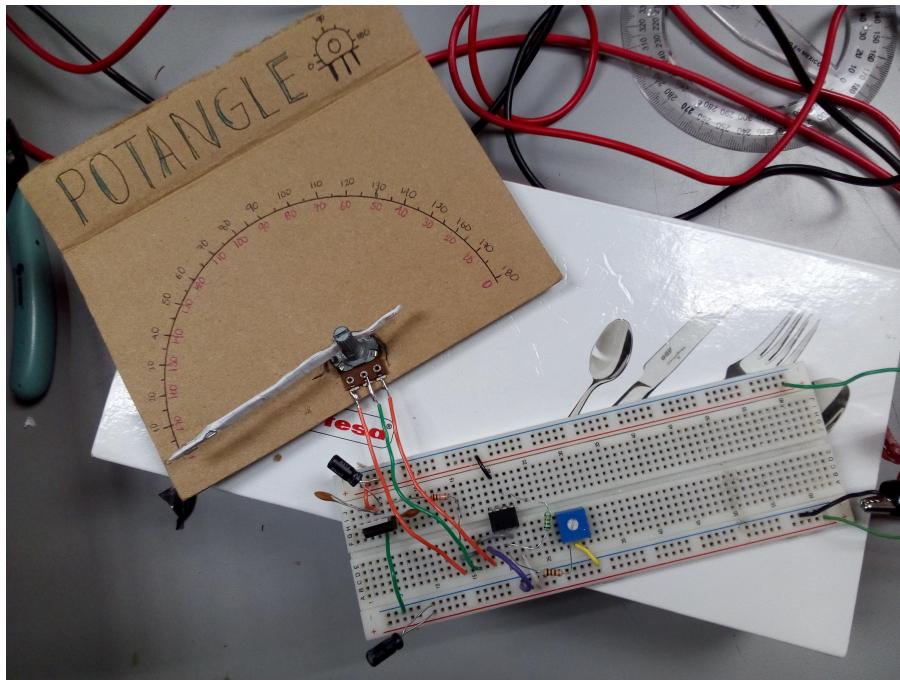
5. Desarrollo práctico

5.1. Recordando la práctica 2: Sensor resistivo - potenciómetro

En la práctica número 2 utilizamos un potenciómetro como sensor resistivo. Éste fue diseñado a partir de uno con valor de $2K\Omega$, una fuente de 12V y una corriente de 1 mA.

Posteriormente, se acondicionó la señal del potenciómetro para obtener una salida constante de 5V, por medio de un amplificador operacional o un preset $10K\Omega$ para ajustar el valor.

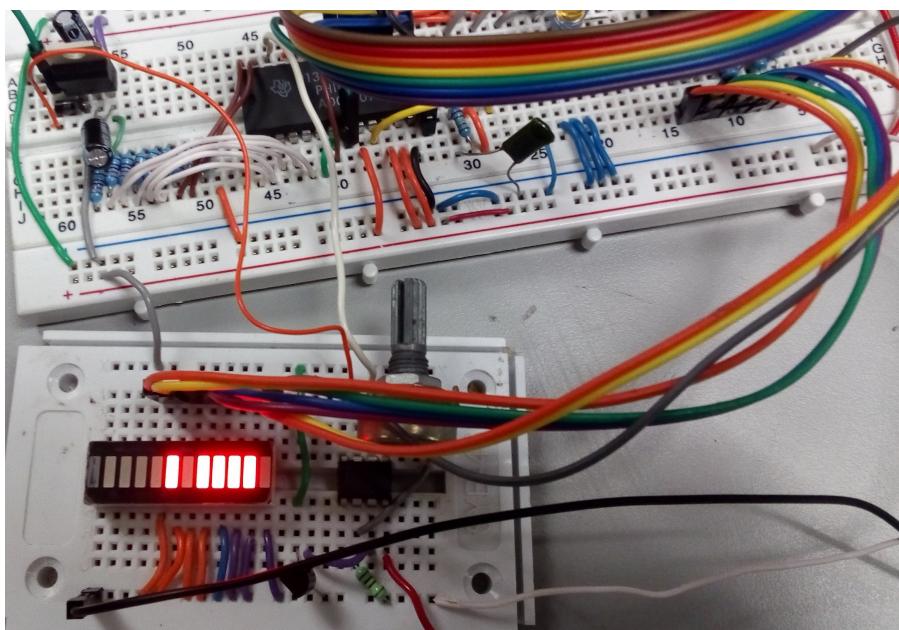
Este sensor mide el ángulo que produce la inclinación del cursor del potenciómetro:



5.2. Recordando la práctica 4: Convertidor analógico-digital

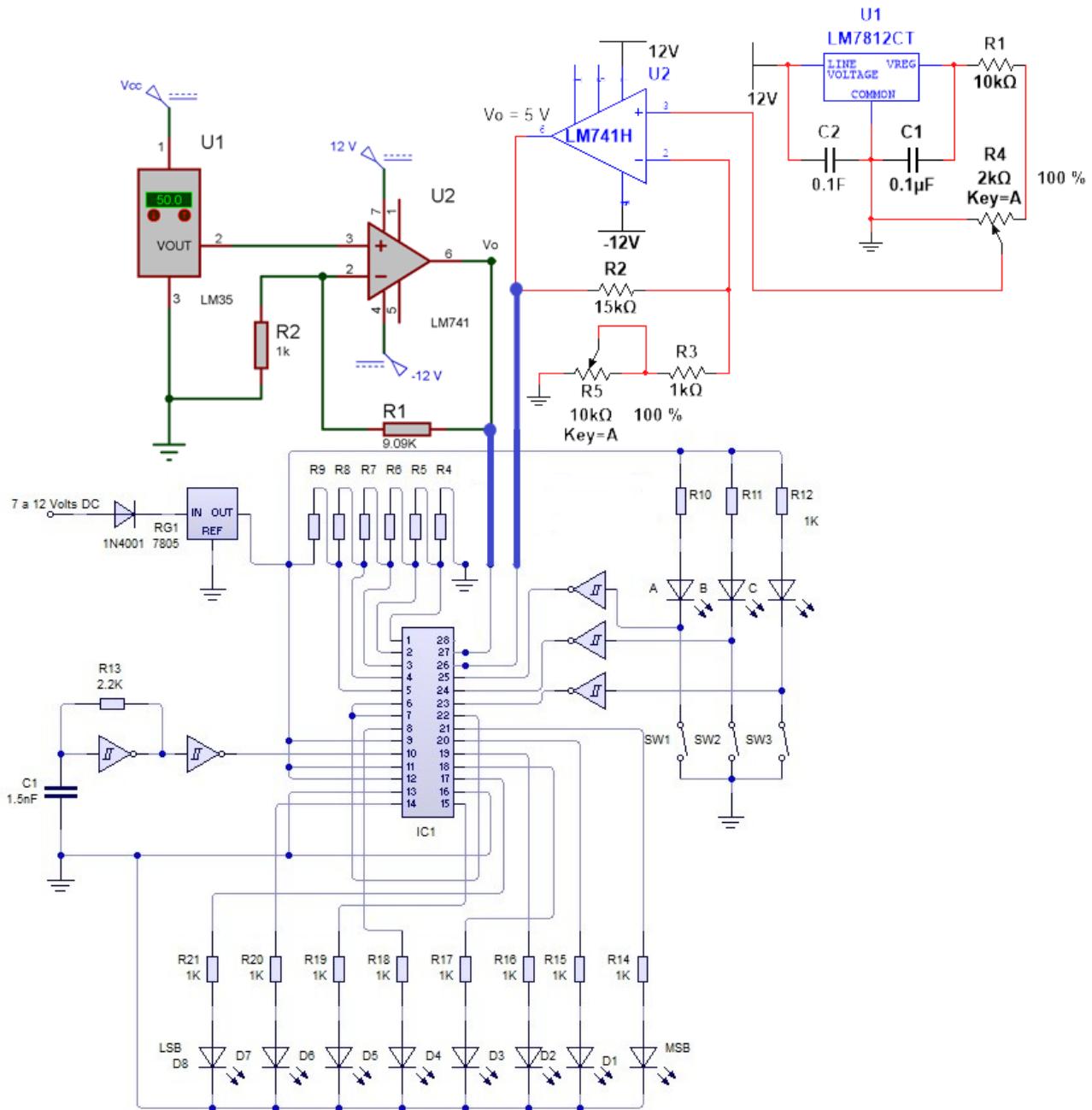
En la práctica número 4 aprendimos el uso y las características de un convertidor analógico - digital ADC0809. Se colocaron resistencias en cada uno de los 8 canales del CAD y se midió el voltaje de entrada, y el valor digital de salida en bits, en binario y en decimal.

Posteriormente se acondicionó un sensor de temperatura LM35, de modo que a 50°C ambientales, el sensor entrega un valor de 5V. Éste sensor se coloco en el canal 0 del CAD, y se obtuvieron las mediciones del voltaje de entrada y su valor digital de salida en bits.

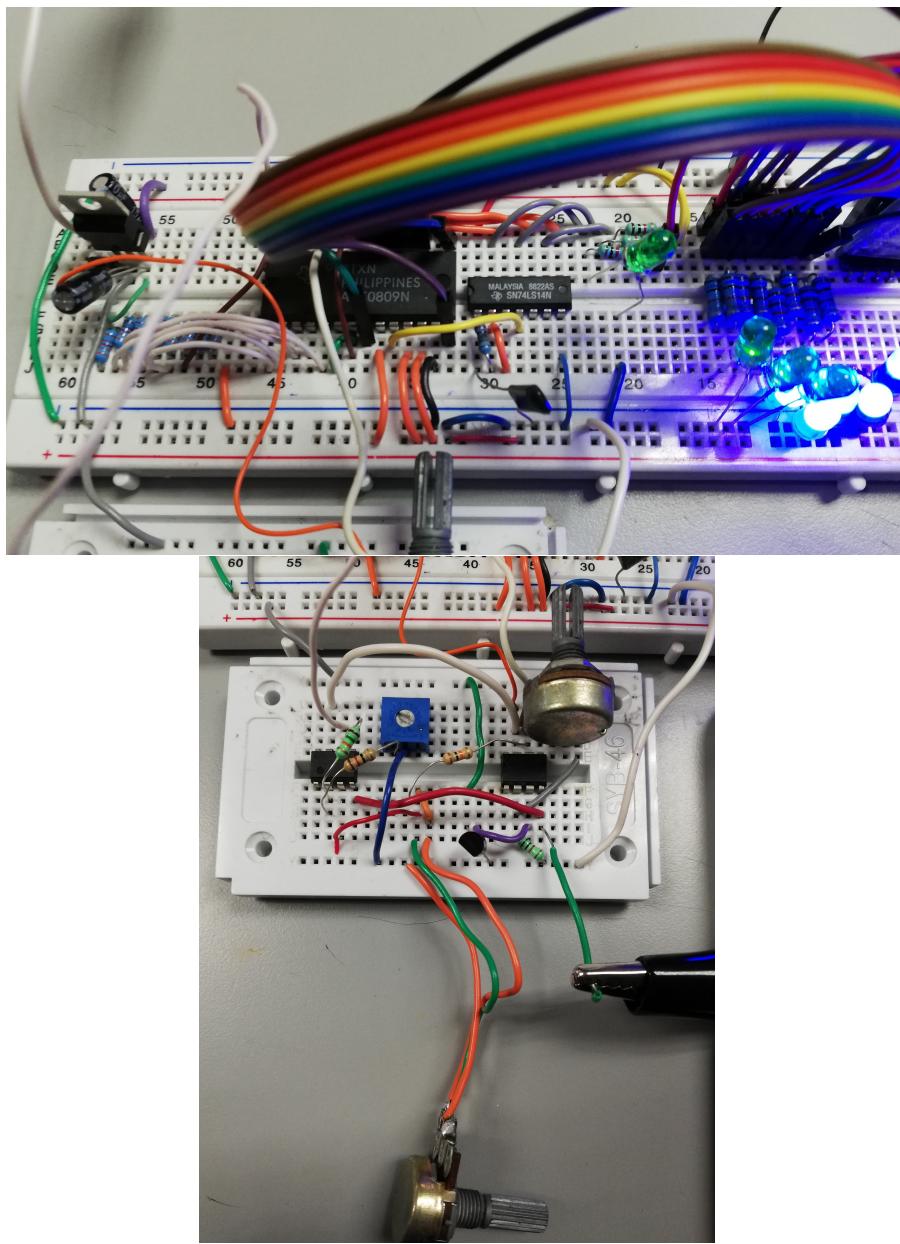


5.3. PE 1 - Conexión de los sensores al CAD

5.3.1. Esquemas



5.3.2. Circuito cableado



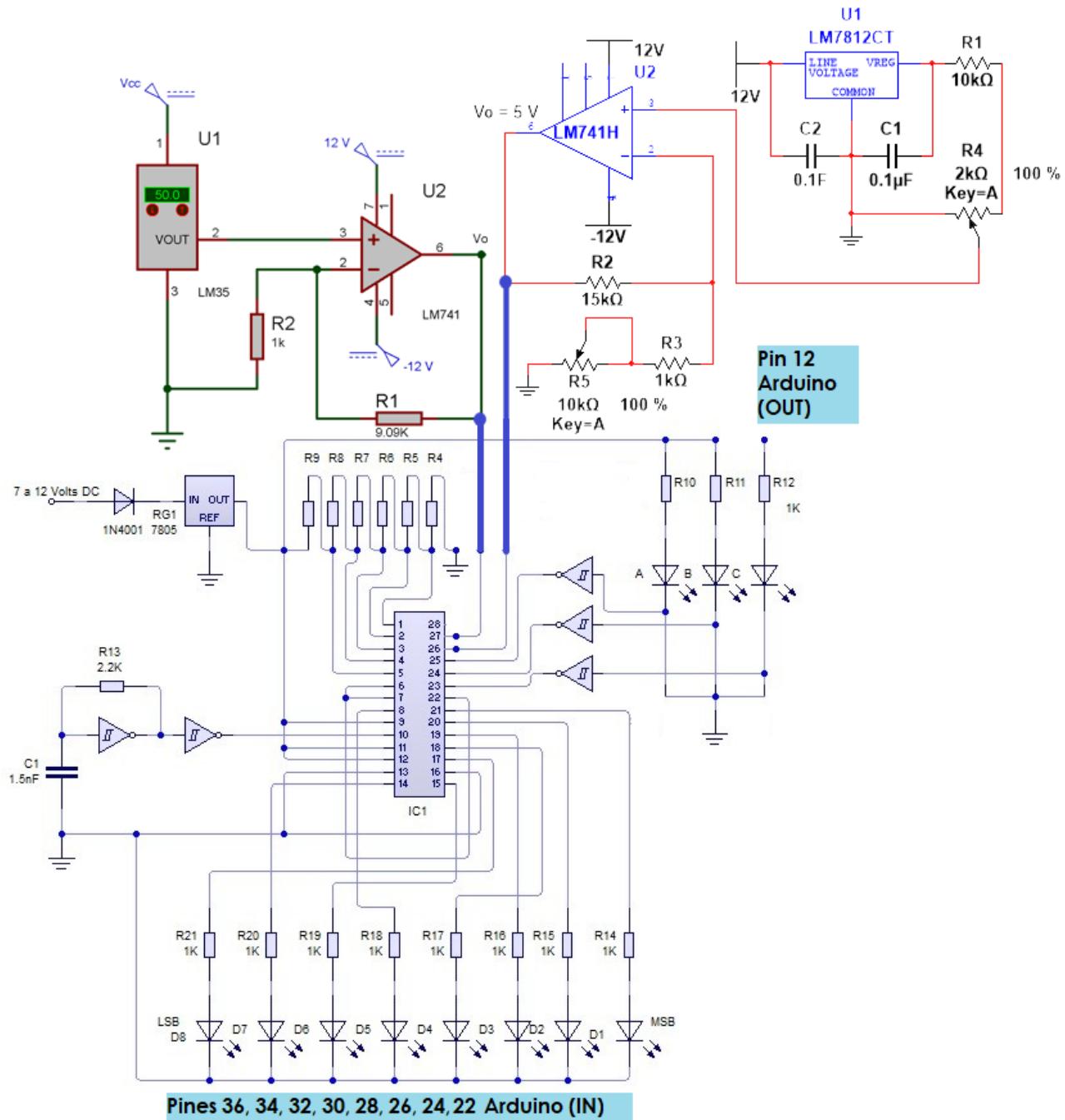
5.3.3. Funcionamiento

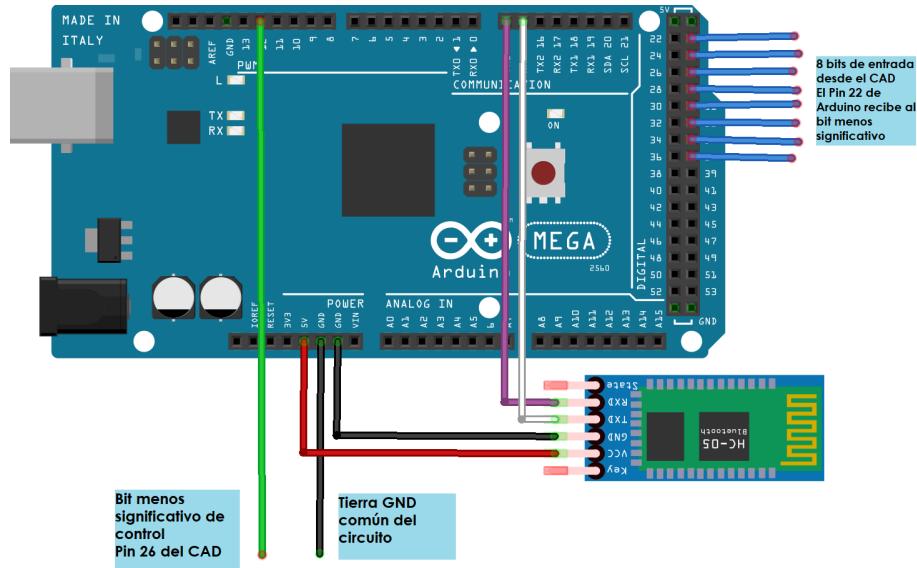
La primera parte de la práctica es sencilla, conectaremos el sensor de temperatura ambiental LM35 de la práctica 4 y el sensor resistivo: potenciométrico, que mide el ángulo de inclinación del cursor del potenciómetro, de la práctica 2 en el Convertidor Analógico - Digital, en el canal 0 y en el canal 1 respectivamente.

Con ayuda de los 3 bits de control, vamos seleccionando el canal que deseamos visualizar en el arreglo de los 8 LEDs a donde llegan los 8 bits digitales de salida del CAD.

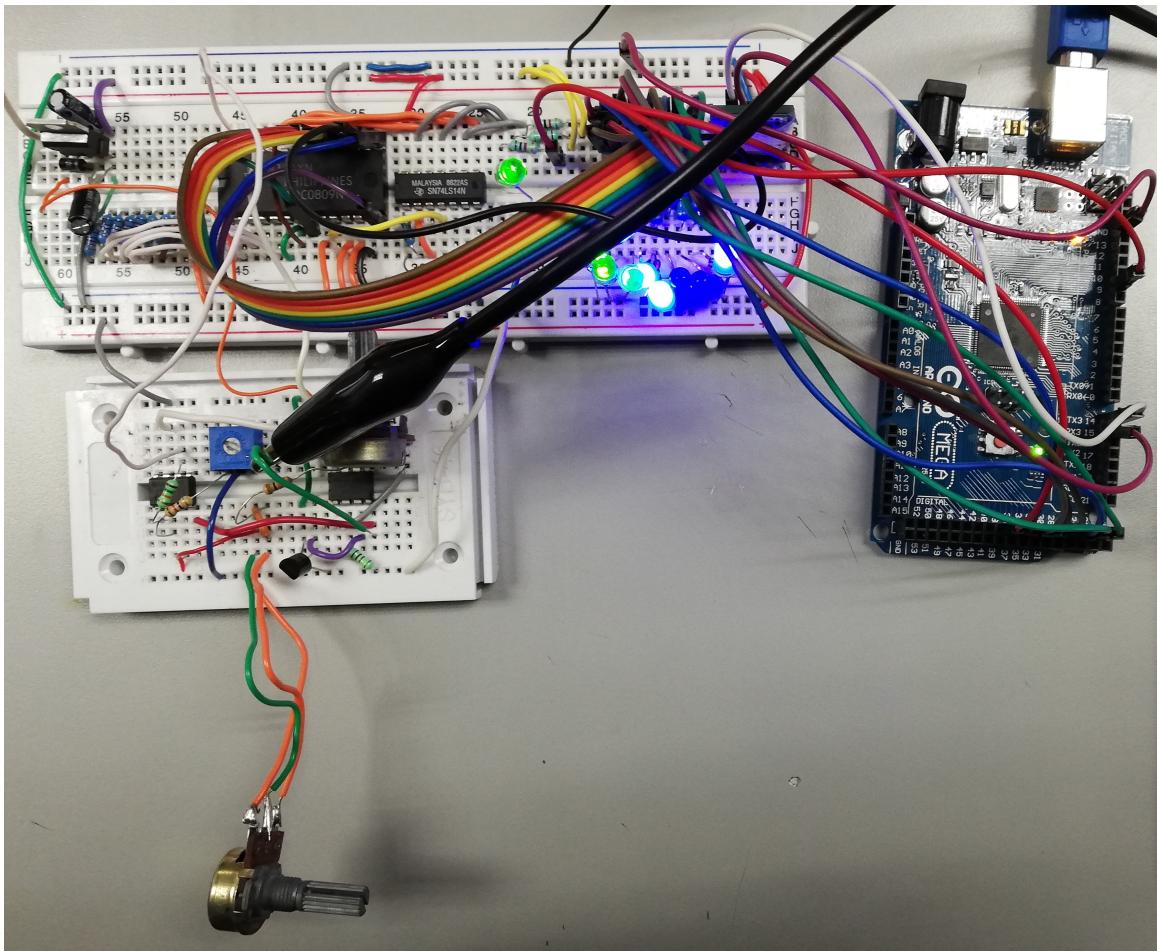
5.4. PE 2 - Integración del módulo Bluetooth HC-05 y la placa Arduino al CAD

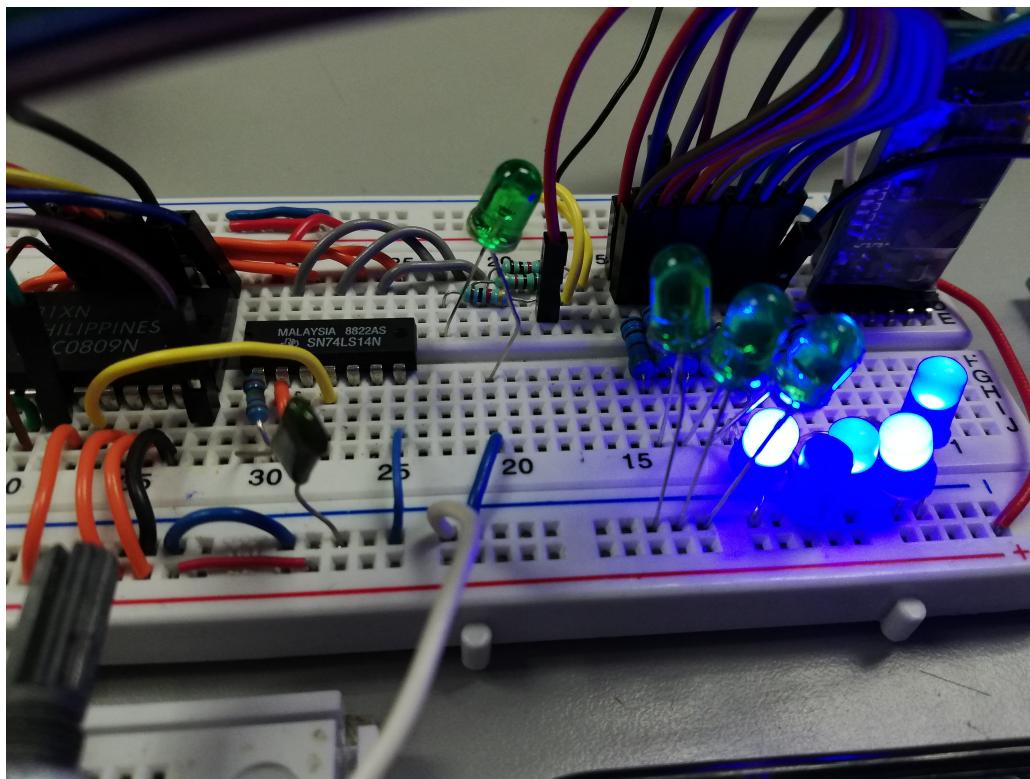
5.4.1. Esquemas





5.4.2. Circuito cableado





5.4.3. Funcionamiento

Una vez funcionando ambos sensores y el CAD, solo queda conectar el módulo Bluetooth HC-05 y enviar la señal digital de 8 bits a la placa Arduino.

Enviamos el pin que entrega 5V al pin VCC del módulo Bluetooth HC-05, quien va a ser el encargado de transmitir la señal digital del CAD recibida y codificada desde la placa de Arduino hacia nuestra aplicación Android en nuestro dispositivo móvil, y recibir los comandos desde la aplicación Android del dispositivo móvil hacia el Arduino, y de éste al CAD.

Es de suma importancia cablear la salida del pin RX3 del Arduino hacia el pin TxD del HC-05, y de forma similar cablear la salida del pin TX3 del Arduino hacia el pin RxD del HC-05.

Se colocan de forma cruzada, pues el pin RX3 de Arduino va a recibir datos por medio del pin TxD del HC-05 y viceversa, el pin RxD del HC-05 va a recibir datos por medio del pin TX3 de Arduino. A diferencia de la práctica 6, donde visualizamos los pulsos cardíacos por segundo, aquí sí nos interesan ambos canales de comunicación serial.

Desde la placa Arduino enviaremos una señal digital, que será el bit menos significativo de control para ir cambiando entre los canales 0 y 1 del CAD desde nuestra aplicación móvil Android. Esto se explicará con más detalle más adelante.

Los otros 2 bits de control restantes los mandamos a 5V - V_{CC} .

No debemos olvidar colocar las tierras de ambos componentes en la tierra común del

circuito del CAD junto con los sensores.

5.5. PE 3 - Programa en Arduino

5.5.1. Código

```
1 double bits[8], decimal;
2 int temp, angulo;
3 const int a = 22, b = 24, c = 26, d = 28, e = 30, f = 32, g = 34, h = 36, salida = 9;
4 const double resolucion = 5.00/255.00;
5 int val = '0', estado = 0;
6
7 void setup(){
8     // put your setup code here, to run once:
9     Serial.begin(9600);
10    Serial3.begin(9600);
11    pinMode(a, INPUT);
12    pinMode(b, INPUT);
13    pinMode(c, INPUT);
14    pinMode(d, INPUT);
15    pinMode(e, INPUT);
16    pinMode(f, INPUT);
17    pinMode(g, INPUT);
18    pinMode(h, INPUT);
19    pinMode(salida, OUTPUT);
20 }
21
22 void loop(){
23     // put your main code here, to run repeatedly:
24     bits[0] = digitalRead(a);
25     bits[1] = digitalRead(b);
26     bits[2] = digitalRead(c);
27     bits[3] = digitalRead(d);
28     bits[4] = digitalRead(e);
29     bits[5] = digitalRead(f);
30     bits[6] = digitalRead(g);
31     bits[7] = digitalRead(h);
32     decimal = bits[0] + bits[1]*2 + bits[2]*4 + bits[3]*8 + bits[4]*16 + bits[5]*32 + bits[6]*64 +
33         → bits[7]*128;
34     temp = resolucion*decimal*10;
35     angulo = resolucion*decimal*57;
36
37     if(Serial3.available()){
38         val = Serial3.read();
39     }
40     if(val == '1'){
41         estado = 1;
42     }
43     else if(val == '0'){
44         estado = 0;
45     }
46     if(estado == 0){
47         digitalWrite(salida, HIGH);
48         Serial.print("Canal 0: ");
49         Serial3.println(temp);
50         Serial.println(temp);
51     }
52     else if(estado == 1){
53         digitalWrite(salida, LOW);
54         Serial.print("Canal 1: ");
55         Serial3.println(angulo);
56         Serial.println(angulo);
```

```

56     }
57     delay(50);
58 }
```

5.5.2. Explicación del código

Primero declaramos un arreglo de 8 bits, una variable que tendrá el valor decimal de estos, las variables para los valores finales de temperatura y ángulo de inclinación, los pines como enteros constantes, la resolución del CAD, que es 0.019607, una variable de lectura del bit de control de la aplicación móvil y una variable del estado o canal actual.

En el bloque *setup()* inicializamos dos canales seriales de comunicación: Serial() para ver los resultados en la computadora, y Serial3(), que será la comunicación vía Bluetooth. También establecemos que queremos 8 pines de entrada y uno de salida.

En el bloque *loop()* leemos digitalmente cada uno de éstos pines (8 bits digitales desde el CAD), y obtenemos su valor en decimal (el recibido es en binario).

Luego, calculamos los valores tanto de temperatura como de ángulo. Este se hace multiplicando el valor obtenido decimal por la resolución del CAD por la proporción entre el valor real y el medido.

Esta proporción se obtiene dividiendo el valor real máximo de los sensores: 50 para el de temperatura y 285 para el ángulo, entre el valor de voltaje máximo entregado por estos, que es 5V en ambos.

Para el sensor de temperatura este valor es $= \frac{50}{5} = 10$.

Para el potenciómetro este valor es $= \frac{285}{5} = 57$.

Posteriormente, leemos desde el canal Serial3() con *read()* el canal a medir para modificar el estado o canal de lectura actual y el bit de control. Si es igual a "0", cambiamos el estado actual de lectura a 0, si es igual a "1", cambiamos el estado actual de lectura a 1. Esto se hace así debido a que, si no manejamos un estado actual de lectura, el Arduino con cambiará entre canales del sensor de forma "permanente", solo lo hará en un lapso de un milisegundo.

NOTA: Al utilizar un circuito inversor en el CAD, utilizamos lógica negativa:

Si el canal deseado a medir es el 0, enviamos un 1 (encender) al bit de control.

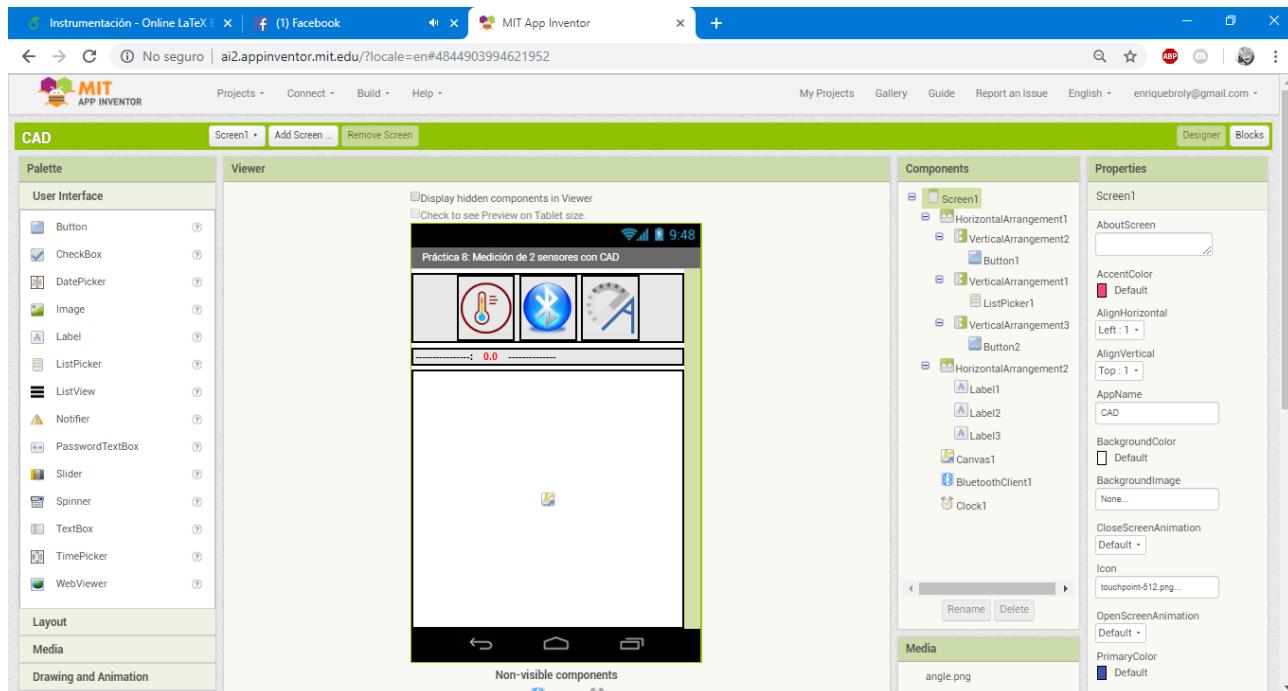
Si el canal deseado a medir es el 1, enviamos un 0 (apagar) al bit de control.

Ya solo nos queda revisar cuál es el estado actual de lectura, y dependiendo si es del canal 0 o 1, enviaremos ya sean los valores de temperatura o ángulo calculados por medio del canal Serial3() con *println* (importante escribir el salto de línea al final).

Añadimos un pequeño *delay()* de 50 milisegundos.

5.6. PE 4 - Aplicación Android desarrollada para móviles

5.6.1. Código e interfaz gráfica

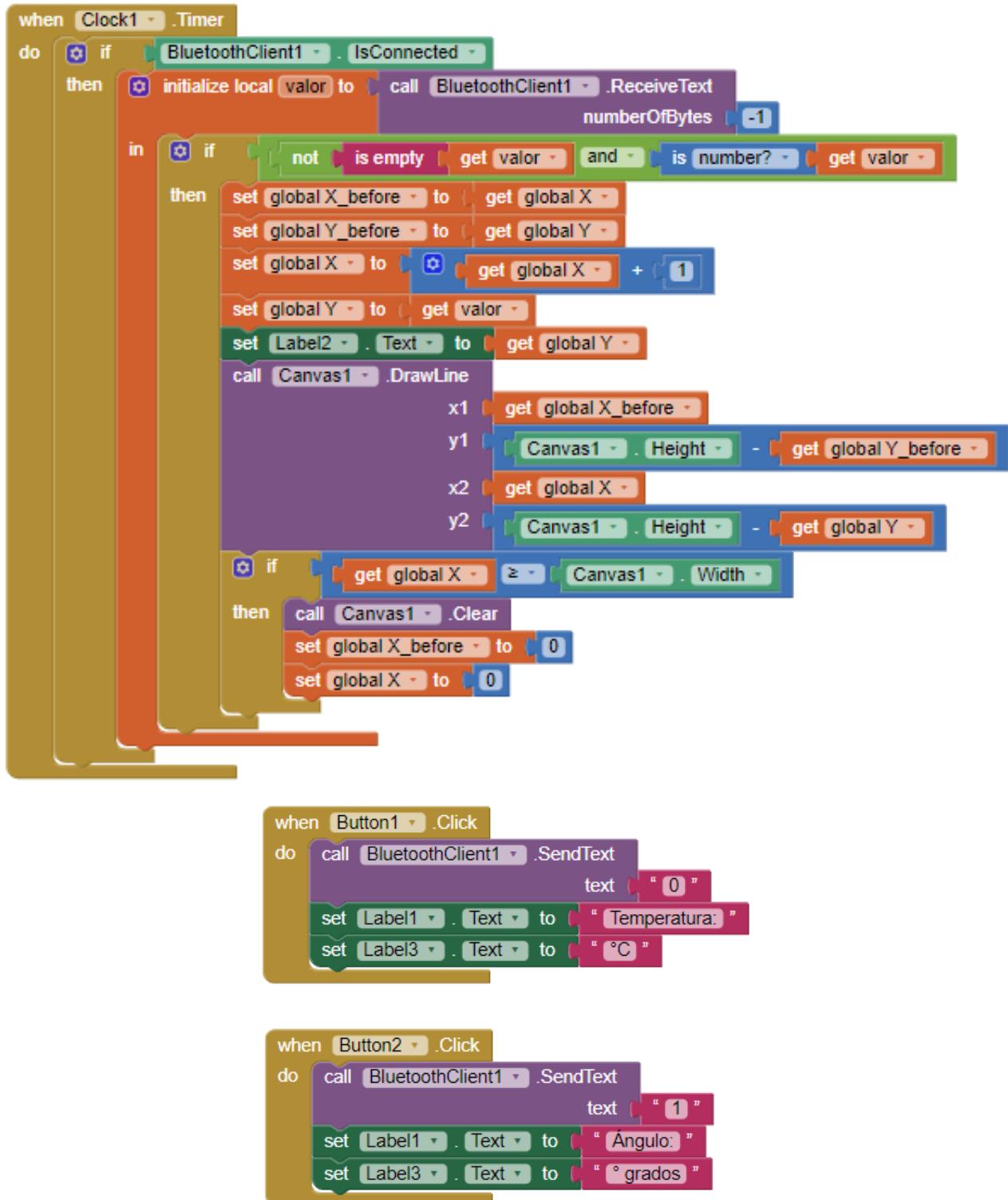


```
initialize global [Y_before] to 0
initialize global [X_before] to 0
initialize global [Y] to 0
initialize global [X] to 0
```

```
when [ListPicker1].BeforePicking
do
  if [BluetoothClient1].Available then
    set [ListPicker1].Elements to [BluetoothClient1].AddressesAndNames
```



```
when [ListPicker1].AfterPicking
do
  evaluate but ignore result
  call [BluetoothClient1].Connect
  address [ListPicker1].Selection
```



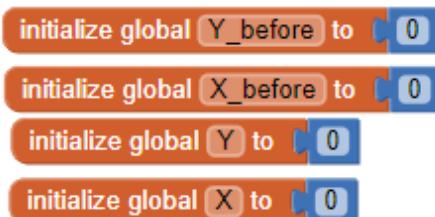
5.6.2. Explicación del código

Para crear esta aplicación, utilizamos la página Web **MIT App Inventor**.

Primero creamos la interfaz gráfica, que consiste en tres botones: uno con una lista que contendrá todos los dispositivos Bluetooth emparejados que el dispositivo móvil posee (en donde se encuentra el ícono del Bluetooth azul), y dos con los que seleccionaremos el canal que deseamos visualizar (en donde se encuentran un ícono según el sensor: un termómetro para el de temperatura y un transportador para el potenciómetro), un cuadro de texto en donde se mostrarán las mediciones de los sensores, y un canvas en donde graficaremos estas mismas señales.

Además, contamos con dos elementos no visibles: un reloj para que nuestra aplicación esté sincronizada con el temporizador del Arduino y los datos se muestren en tiempo real, **este debe ser inicializado en un valor de 1 milisegundo**, y el cliente de conexión Bluetooth. **Tenemos que establecer el byte delimitador a 10, que es el valor ASCII para un salto de línea**.

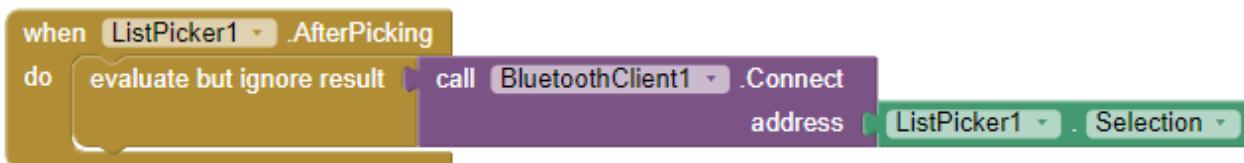
Pasando a los bloques de código, antes que nada inicializamos 4 variables globales, que nos servirán para graficar las señales.



Primero, le decimos a la aplicación que cada vez que se inicie, cargue todos los dispositivos Bluetooth emparejados al dispositivo móvil junto a su MAC.



Luego, decimos que al momento de seleccionar un dispositivo de la lista, la aplicación se conecte a él enviándole su dirección MAC por medio de la función *call BluetoothClient1.Connectaddress()*.

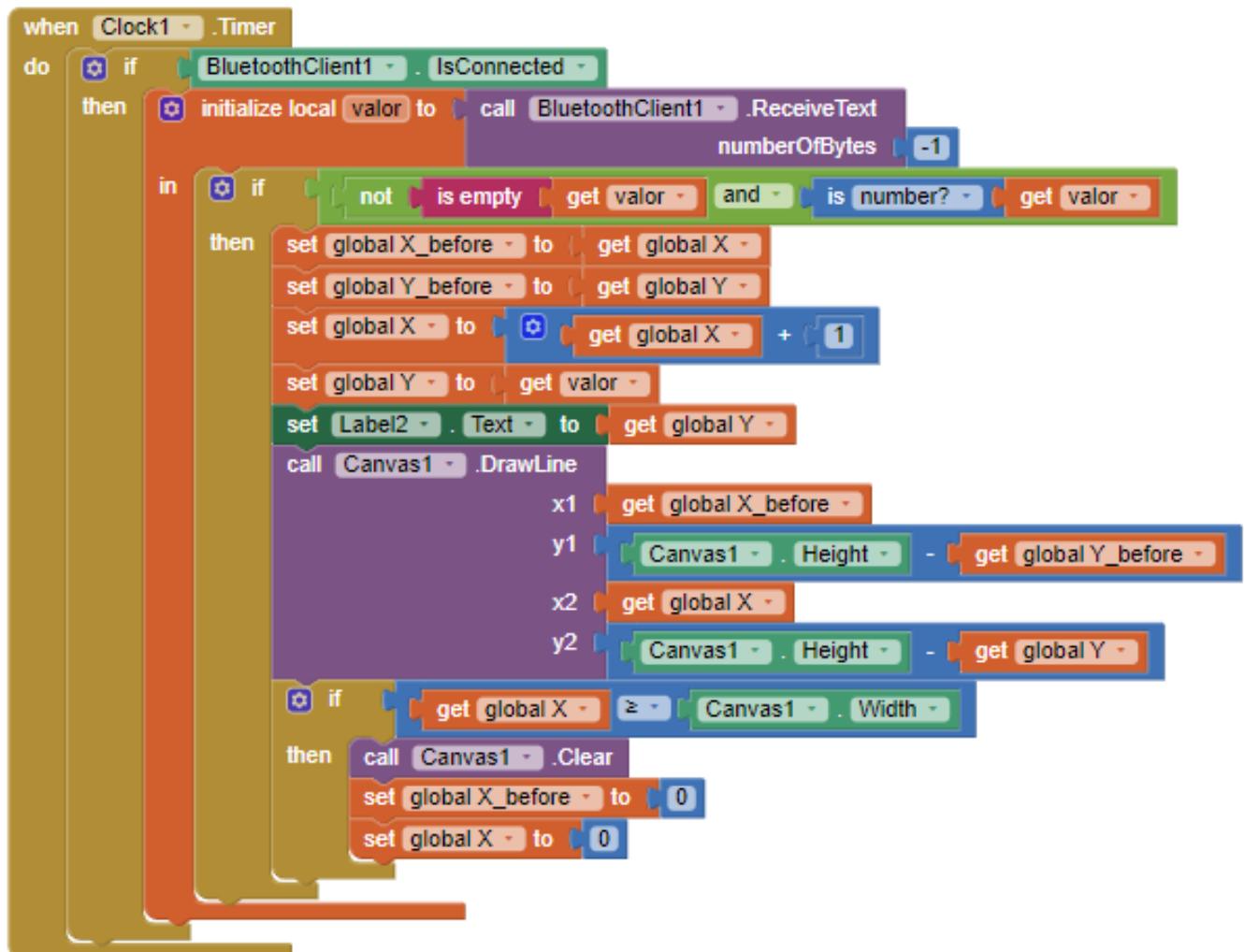


Ahora, cada vez que haya un pulso de reloj en la aplicación de un milisegundo, verificar que el cliente Bluetooth este conectado para así llamar a la función `ReceiveText(numberOfBytes)` dentro de la variable `valor`, y le enviamos un -1, esto significa que la aplicación sólo leerá información hasta encontrar el byte delimitador (salto de línea) cada milisegundo, y así evitar desplegar información basura.

Posteriormente, revisamos si el valor obtenido no es nulo y que sea un número, si sí cumple con ambas condiciones, nos dispondremos a graficar este valor. Esta parte es muy importante.

Con ayuda de las variables globales antes mostradas, iremos aumentando el valor en el eje X del canvas a cada milisegundo, y le asignaremos este valor leído al eje Y del canvas. De igual modo, desplegamos en el cuadro de texto el valor leído desde el Bluetooth.

A continuación, utilizamos la función `DrawLine()`, y le enviamos los valores de los ejes X, Y. De igual modo, tenemos un valor anterior a éstos, y los iremos sumando al actual para que la gráfica se vaya desplazando por la pantalla del celular (el eje X es el tiempo y el eje Y la medición).

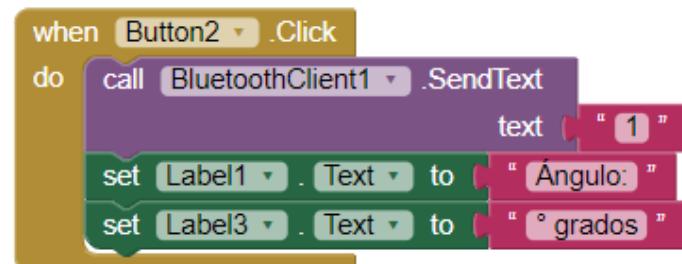
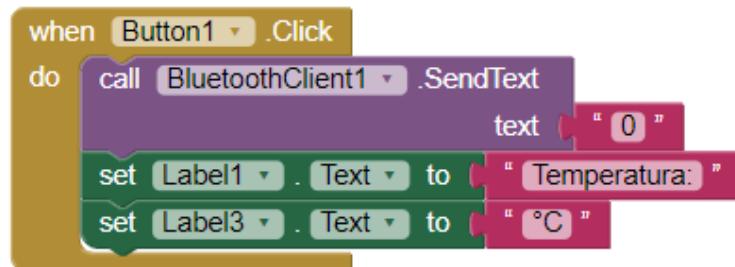


Al final del bloque, lo que hacemos es revisar si la gráfica ha llegado el final de nuestra pantalla (la parte derecha), si lo hace, limpiamos ésta y continuamos desde el inicio (la parte izquierda).

Enviar señal al Arduino

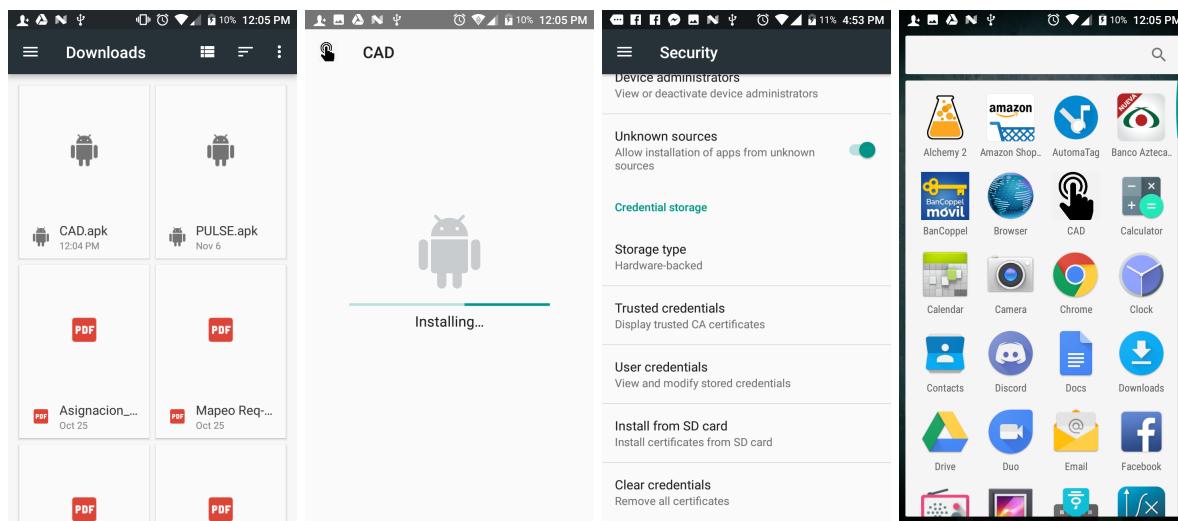
En los siguientes dos bloques lo que hacemos es que, cuando se pulsa el botón, ya sea del sensor de temperatura o del potenciómetro, cada uno de ellos enviará un "0." "1" respectivamente al Arduino, y en la pantalla cambiará el texto que indica qué sensor se está midiendo actualmente.

Tanto la información tanto recibida como enviada serán transmitidas por medio del canal de comunicación Serial3().



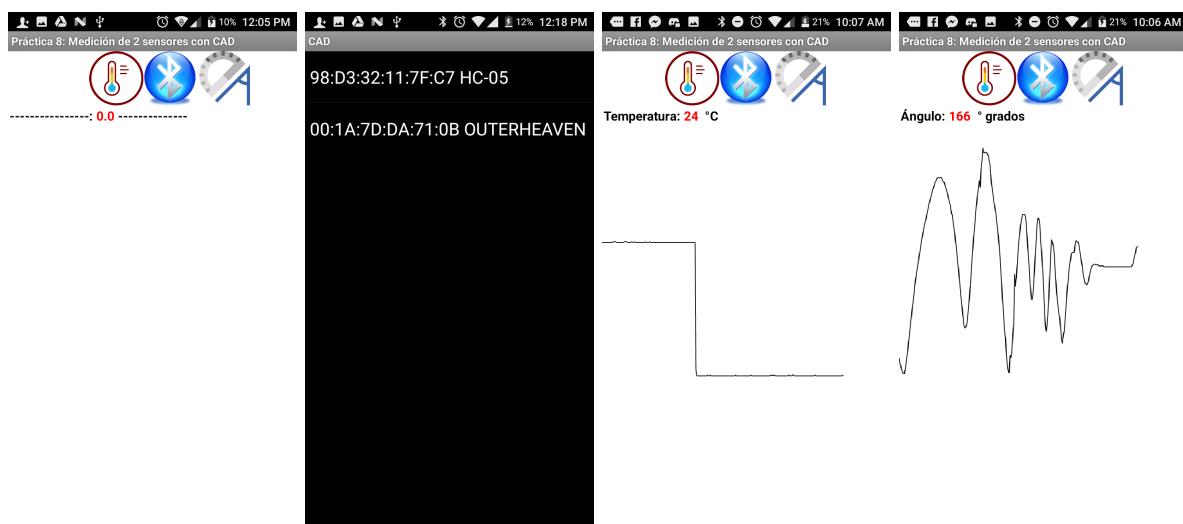
5.6.3. Instalación

1. Descargar e instalar el archivo "CAD.apk" desde el móvil.



Importante: Activar la opción de instalación de aplicaciones de origen desconocido desde Ajustes.

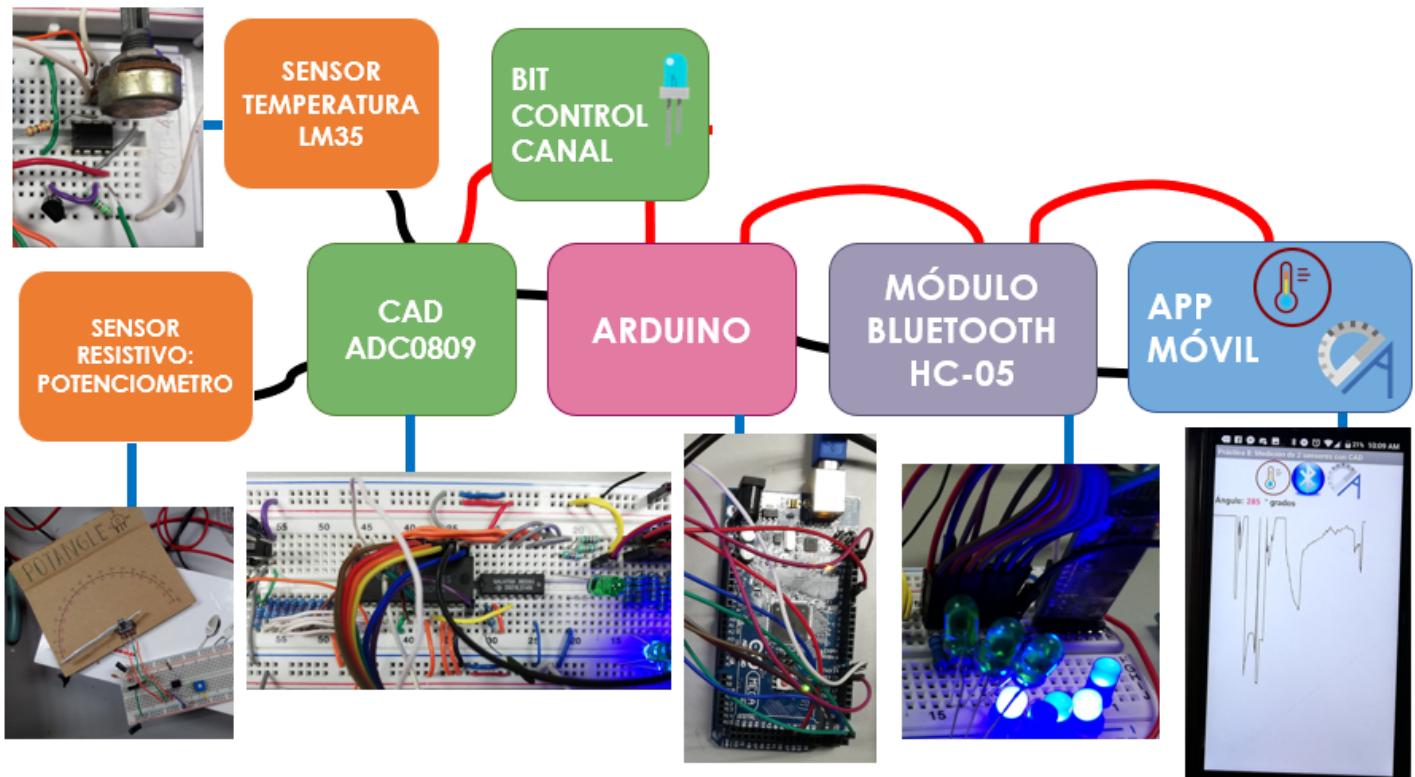
2. Activar el Bluetooth en el celular y emparejar el módulo HC-05 con el código "1234".
3. Entrar a la app.
4. Pulsar el ícono de Bluetooth azul, aparecerá el listado de dispositivos Bluetooth, seleccionar el HC-05
5. Pulsar el ícono del termómetro. La app comenzará a desplegar y graficar la medición del sensor en el canal 0, es decir, la temperatura ambiente en grados Celsius. El rango es de 0°C a 50°C.
6. Para cambiar de canal, pulsar el otro ícono con un transportador. La app comenzará a desplegar y graficar la medición del sensor en el canal 1, es decir, el ángulo del inclinación del cursor del potenciómetro. El rango es de 0° a 285°.
7. Podemos ir cambiando de sensores pulsando en sus íconos las veces que deseemos.



5.7. Funcionamiento completo

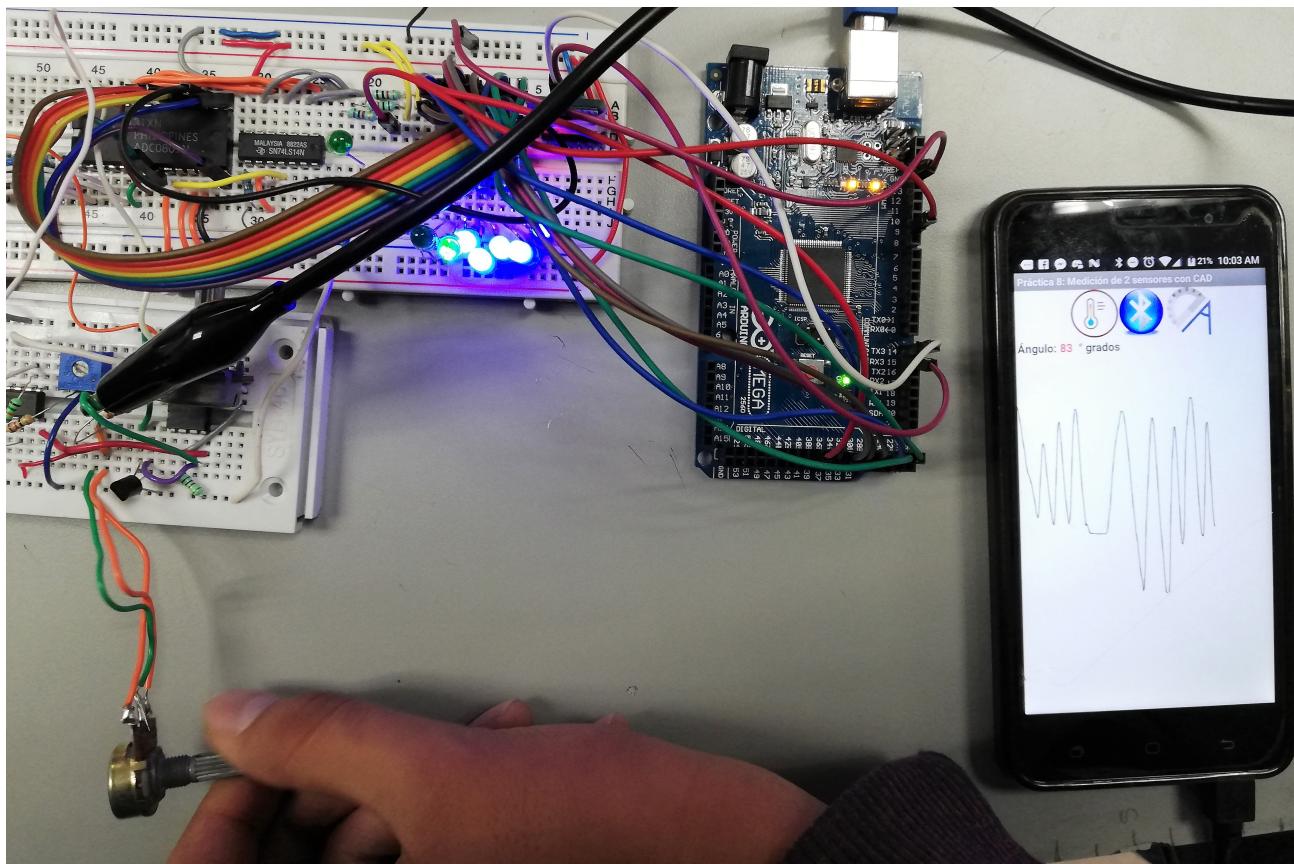
La línea negra es la comunicación de envío desde los sensores hasta la app móvil que los despliega.

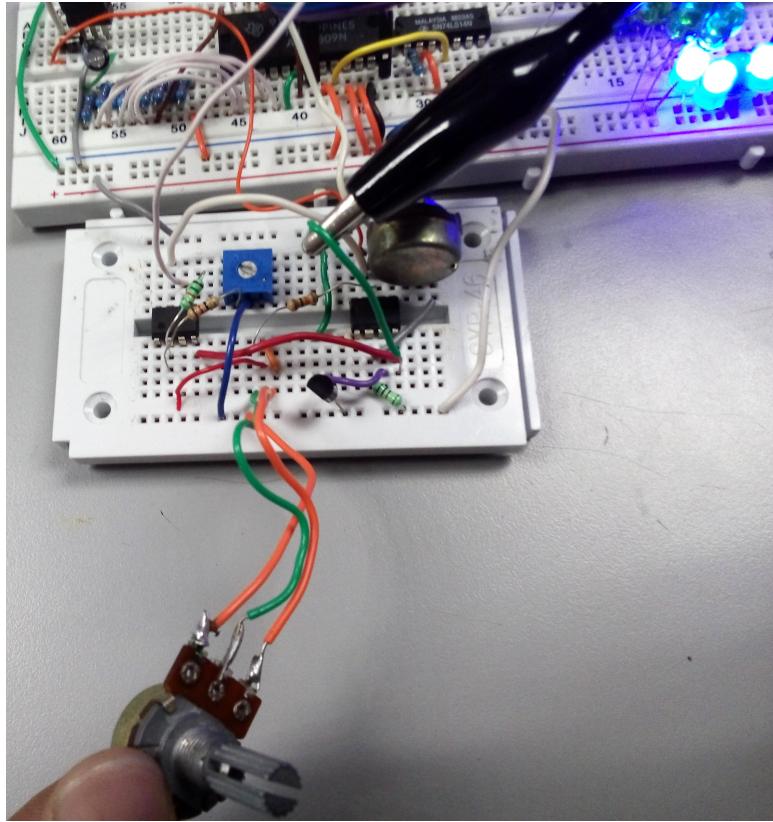
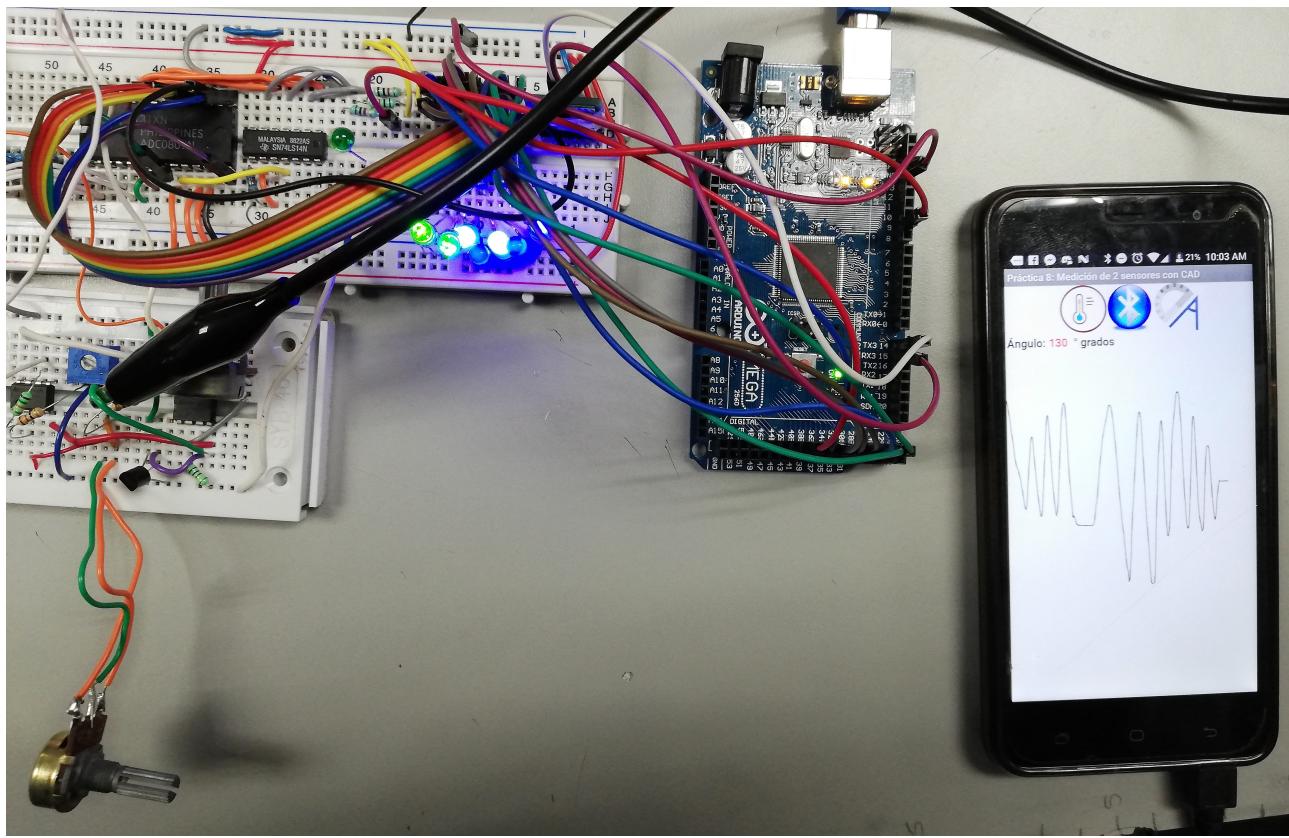
La línea roja es el envío de la señal desde la app móvil para seleccionar y cambiar entre ambos sensores del CAD.



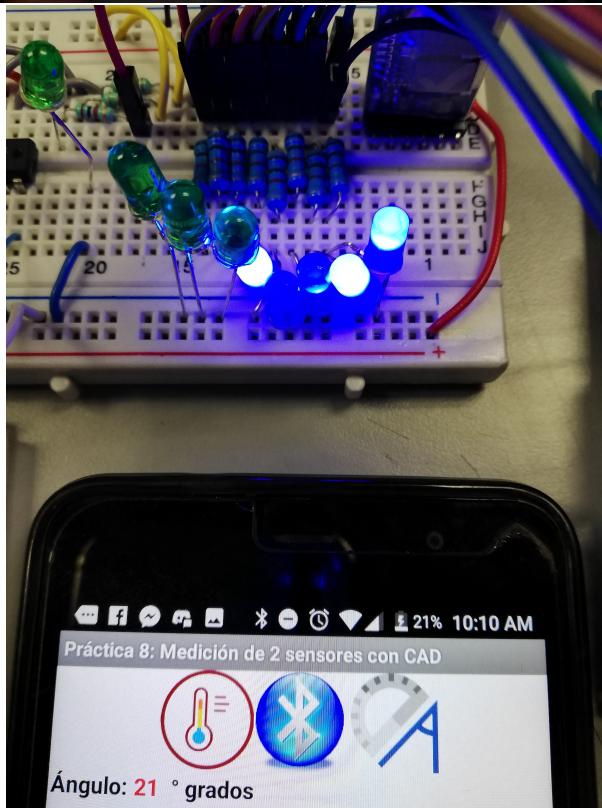
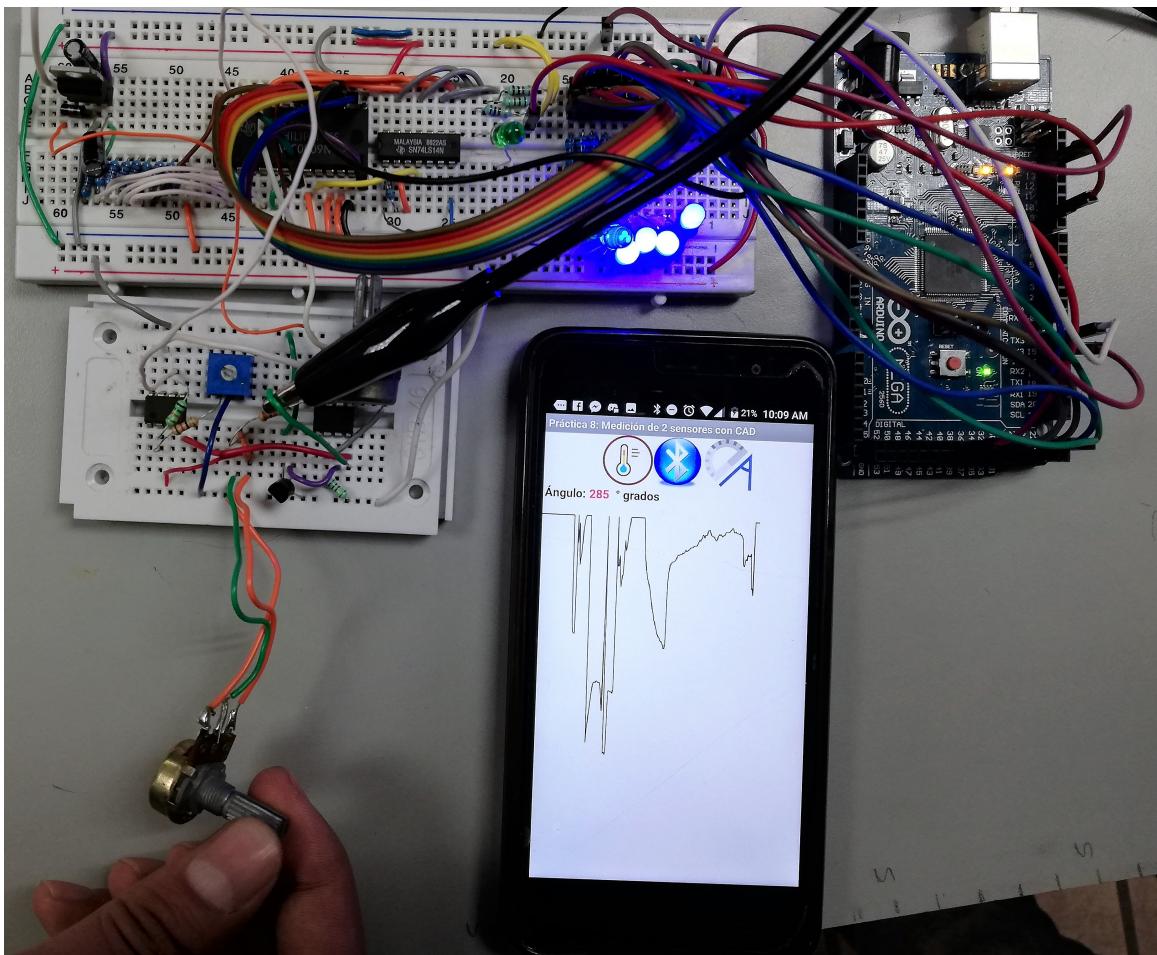
- Sensor Temperatura LM35: Entrega una señal que corresponde a la temperatura ambiental en un rango de 0 °C a 50 °C, de 0V a 5V respectivamente.

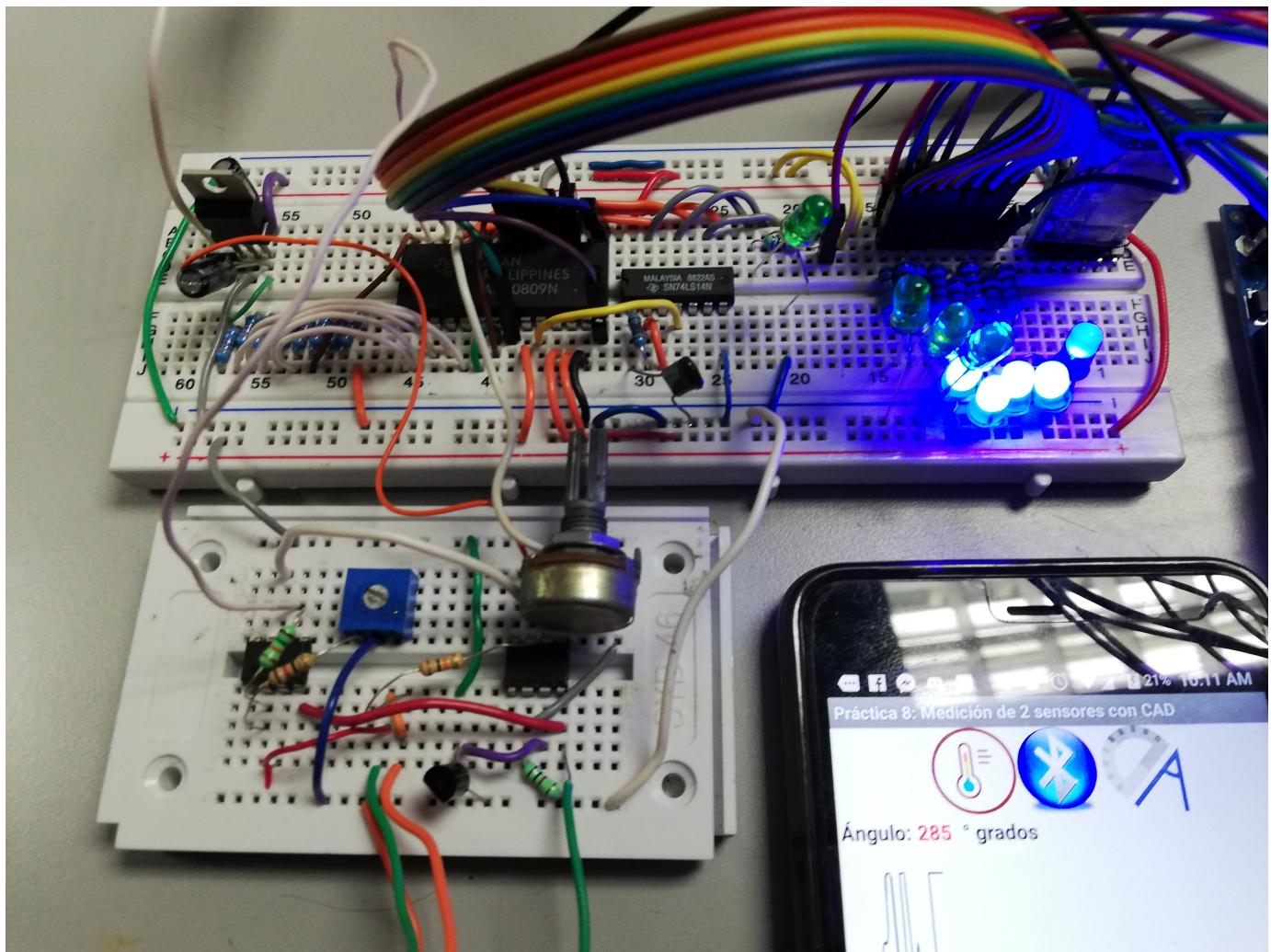
- Sensor Resistivo - Potenciómetro: Entrega una señal que corresponde al ángulo de inclinación del cursor de un potenciómetro en un rango de 0° a 285°, de 0V a 5V respectivamente.
- CAD ADC0809: Convierte las señales analógicas de ambos sensores, conectados al canal 0 y al canal 1, a una digital, por medio de 8 bits. Según el bit de control, intercambia entre canales.
- Arduino: Calcula el valor de temperatura y ángulo real, según el canal seleccionado actual. Recibe la señal del canal a leer desde la app móvil y cambia el bit de control.
- Módulo Bluetooth HC-05: Retransmite la información recibida a un dispositivo móvil, y viceversa. Comunicación bidireccional.
- App Móvil: Se conecta al HC-05 y despliega con gráficas los valores de temperatura y ángulo, según el canal seleccionado. Con ayuda de dos botones, intercambia entre canales para seleccionar el sensor a medir al pulsar en el botón correspondiente, enviando una señal al Arduino.
- Bit de Control del Canal: Recibe el valor del bit de control de canal del CAD desde Arduino, encendiendo o apagando el bit.











6. Observaciones

- ✓ En el vídeo anexado a este reporte se puede visualizar a mayor detalle el comportamiento y funcionamiento del circuito. Link: <https://drive.google.com/drive/folders/1PjzsLk1Y-OyaEWK74fk6T7k4YGvS6spt?usp=sharing>
- ✓ También anexamos los códigos tanto del programa de Arduino como de la aplicación desarrollada en MIT App Inventor y su APK. Link: https://drive.google.com/drive/folders/1SHBoB-2Rbz-IIuS-W_bzBHui_Ayywmvf?usp=sharing
- ✓ Como utilizamos la comunicación serial 3 del Arduino, no importa si hay pines conectados al circuito o al módulo HC-05, a diferencia de la práctica 6.
- ✓ Es preferible que primero se conecte todo el circuito completo, y luego se ejecute la aplicación, para evitar errores.

- ✓ En el bit de control que envía el Arduino hacia en control del CAD no es necesario colocar una resistencia, el voltaje que envía un pin de salida de Arduino es de 5V, y su corriente es muy pequeña, por lo que el circuito no sufrirá de daño alguno. Esto es más por prevención.

7. Conclusiones

7.1. Aguilar Herrera Arianna Itzamina

Esta fue la última práctica, la cual fungió a la par como "proyecto", por lo cual fue una práctica integradora, donde se usaron dos sensores, los cuales fueron el de temperatura y la medición de ángulos, estos se conectaron a un programa hecho en arduino, para poder mostrar los resultados de los instrumentos mencionados en una aplicación en el celular.

Puesto que eran circuitos que ya se habían armado con anterioridad, no fue complicado el retomarlos, ya que se conversaron como en su momento se entregaron, por lo que lo que había que hacer era conectarlos entre sí, para posteriormente enviarlos mediante la señal bluetooth los datos obtenidos y estos se pudieran mostrar en la aplicación antes mencionada.

Otro plus en esta práctica fue el de poder manipular el instrumento desde el celular, es decir, que no fuera de forma manual directo en el circuito.

7.2. Ramos Diaz Enrique

Como última práctica, esta práctica fue una integradora de muchas cosas que se revisaron a lo largo del semestre: acondicionamiento de señales, interpretación de éstas, despliegue, manejo y conversión, mediciones de sensores, etc. Además del complemento de la programación en Arduino y en dispositivos móviles, que como tal no forma parte de la materia de instrumentación, pero se agradece que tengamos una pequeña introducción a estos tópicos.

A nivel de armado del circuito, no hubo mayor problema. Todos los circuitos los conservamos armados todo el semestre, en especial el del Convertidor Analógico - Digital. Y solo era cuestión de conectar los dos sensores a los canales del CAD, e integrar el módulo Bluetooth y la placa Arduino de mismo modo que en la práctica 6 de los pulsos cardíacos por minuto.

El verdadero reto fue lograr una comunicación bidireccional entre el Arduino y la aplicación móvil vía Bluetooth, pues el objetivo era manipular desde nuestro celular el comportamiento del circuito, es decir, sin interferir de forma manual o mecánica (con un dipswitch, por ejemplo).

Para esto, se tuvo que cuidar muy bien las condiciones dentro del Arduino para que éste sepa que señal desplegar, según el botón seleccionado en la aplicación. Por otro lado, para obtener una mejor presentación en la aplicación, se optó por graficar las señales medidas.

7.3. Nicolas Sayago Abigail

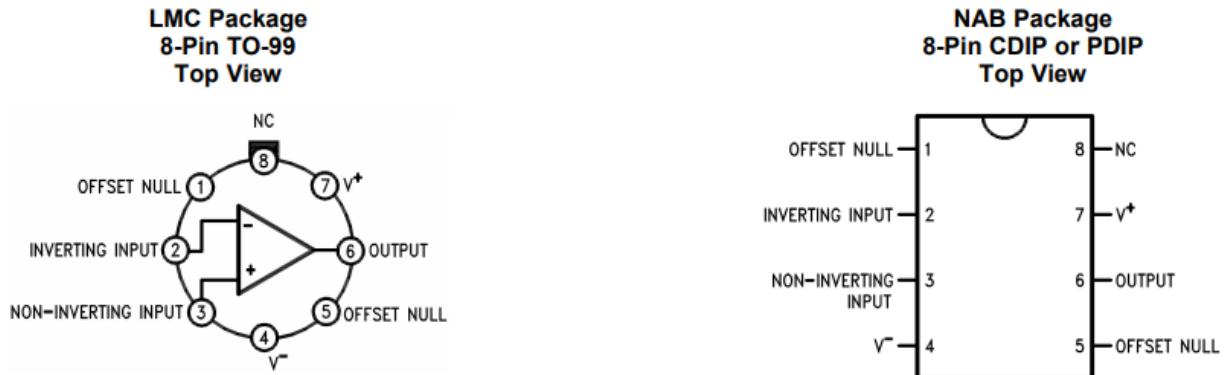
En esta práctica, nosotros elegimos los sensores que queríamos usar, en este caso, el de temperatura y el de medición que en la práctica 2 se llevo a cabo, por lo tanto, armar o condicionar estos sensores y su variable a medir no fue tan difícil puesto que ya lo habíamos hecho anteriormente. La comunicación entre Arduino y la aplicación móvil tampoco era del todo difícil porque ya se había hecho anteriormente.

En esta práctica se puso a prueba el manejar más de una variable a medir y que el convertidor Analógico - digital ADC0809 funcionara de manera correcta al cambiar de sensor, cabe decir que nosotros no habíamos desarmado nuestro circuito anterior.

Puedo destacar que para ser la última práctica, este curso ha superado mis expectativas puesto que en lo personal, los circuitos no son lo mío, sin embargo he aprendido bastante y espero poder integrar mis conocimientos para cosas más grandes, jugando con **Instrumentos**.

8. Anexos

8.1. LM741



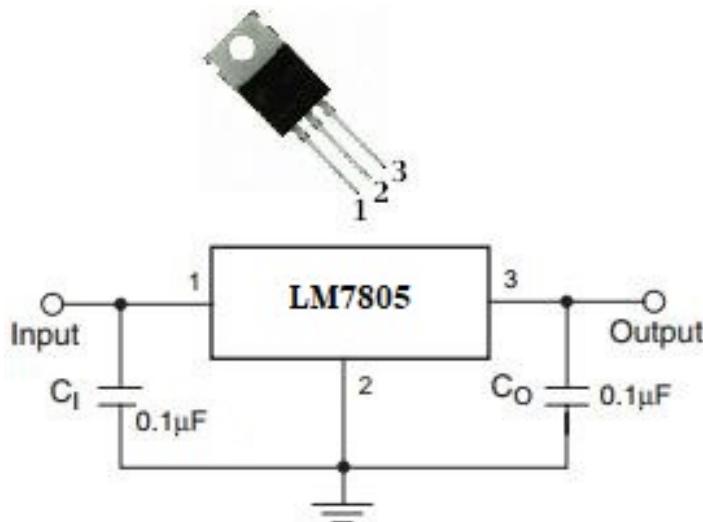
LM741H is available per JM38510/10101

El amplificador operacional LM741 debe estar alimentados por una fuente de +12 V y -12 V en serie. [Consultar hoja de especificaciones del LM741].



8.2. LM7805

En el regulador de voltaje LM7805, deben ir cableados dos capacitores de $0.1\mu F$, como se muestra en el siguiente esquema: [Consultar hoja de especificaciones del LM7805].



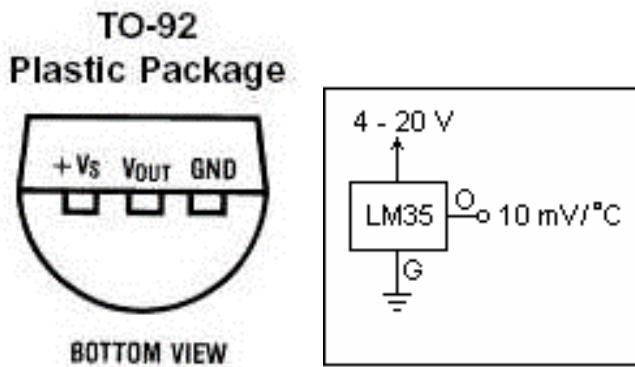
8.3. ADC0809

El ADC0809 es un conversor de 8 bits (la señal análoga se convierte en una palabra de 8 bits), que tiene la posibilidad de leer 8 señales analógicas (8 canales). Posee 28 pines de los cuales 8 corresponden a los ocho canales de entradas analógicos; éste solo puede leer un canal a la vez y dispone por lo tanto de un selector (multiplexor) de 3 líneas digitales, que permite escoger la señal de entrada a convertir, mediante el código binario.

| | | | |
|-----------------|----|----|-------------|
| IN 3 | 1 | 28 | IN 2 |
| IN 4 | 2 | 27 | IN 1 |
| IN 5 | 3 | 26 | IN 0 |
| IN 6 | 4 | 25 | A } |
| IN 7 | 5 | 24 | B } |
| START | 6 | 23 | C } |
| EOC | 7 | 22 | ADDRESS |
| 2 - 5 | 8 | 21 | |
| OEN | 9 | 20 | ALE |
| CLK | 10 | 19 | 2 - 1 (MSB) |
| V _{CC} | 11 | 18 | 2 - 2 |
| REF+ | 12 | 17 | 2 - 3 |
| GND | 13 | 16 | 2 - 4 |
| 2 - 7 | 14 | 15 | 2 - 8 (LSB) |

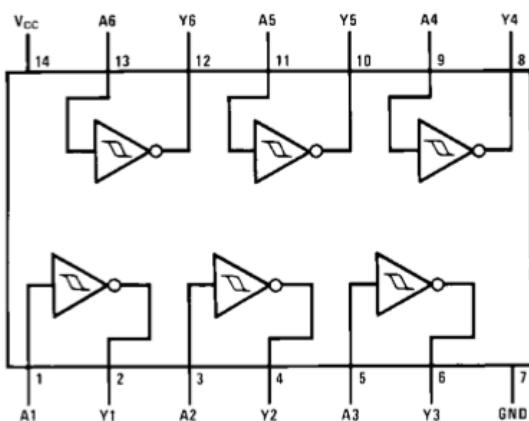
8.4. LM35 Sensor de temperatura integrado

El LM35 es un dispositivo de tres terminales que produce un voltaje de salida $10\text{mV}/^\circ\text{C}$, de modo que el voltaje nominal de salida es 250mV a 25°C y 1V a 100°C . Este sensor puede medir temperaturas debajo de 0°C usando una resistencia de pull-down desde el terminal de salida a una tensión debajo de cero. La precisión del LM35 es de $\pm 1^\circ \text{C}$ desde -55°C a $+150^\circ\text{C}$.



8.5. Inversor SN74LS14N

Connection Diagram



Function Table

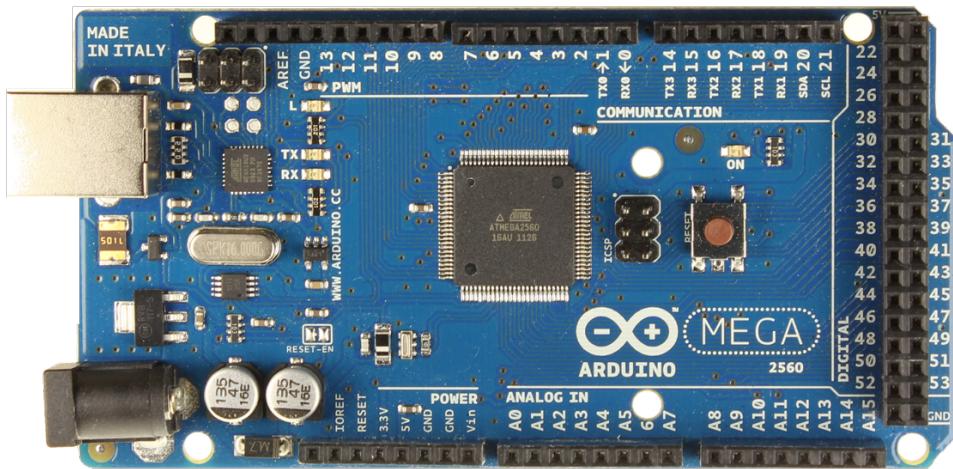
$$Y = \bar{A}$$

| Input | Output |
|-------|--------|
| A | Y |
| L | H |
| H | L |

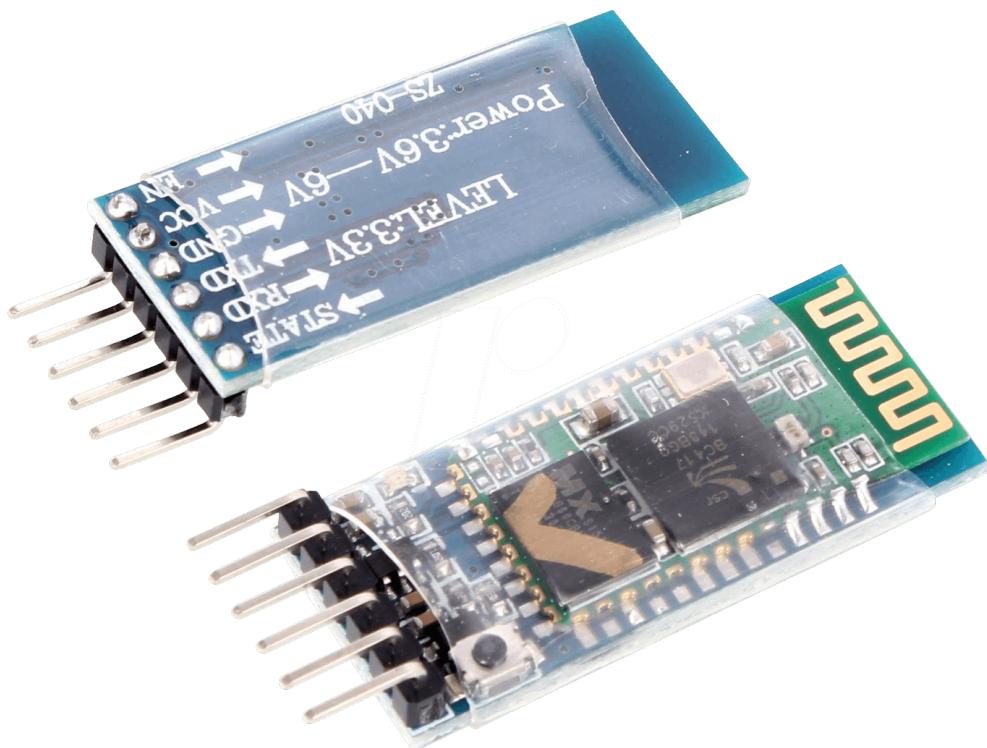
H = HIGH Logic Level

L = LOW Logic Level

8.6. Arduino Mega



8.7. HC-05



8.8. Página Web: MIT App Inventor



Para registrarnos y comenzar a crear aplicaciones, únicamente necesitamos una cuenta de Google.

Link: <http://ai2.appinventor.mit.edu/>

8.9. Arduino IDE

Download the Arduino IDE

The screenshot shows the official Arduino website's software download page. On the left, there is a large teal circular icon containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of the icon, the text "ARDUINO 1.8.7" is displayed in bold. Below this, a paragraph describes the Arduino Software (IDE) as open-source, Java-based, and compatible with Windows, Mac OS X, and Linux. It mentions that the environment is based on Processing and other open-source software. A link to the "Getting Started" page is provided for installation instructions. On the right side of the page, there are download links for different operating systems. For Windows, there are links for "Windows Installer" (for Windows XP and up) and "Windows ZIP file for non admin install". For Mac OS X, there is a link for "Mac OS X 10.8 Mountain Lion or newer". For Linux, there are links for "Linux 32 bits", "Linux 64 bits", and "Linux ARM". At the bottom right, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Link de descarga: <https://www.arduino.cc/en/Main/Software>

Referencias

- [1] J. C. T. Barrera, *Minirobótica y Electrónica en la ESCOM*. [Online]. Available: <http://miniroboticaeducativa.blogspot.com/>

- [2] M. LATAM, *Potenciómetro - ¿Qué es y cómo funciona?* [Online]. Available: <https://www.mecatronicalatam.com/resistencia/potenciometro>
- [3] Y. FM, *Qué es Arduino, cómo funciona y qué puedes hacer con uno.* [Online]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [4] masadelante, *¿Qué significa Bluetooth? - Definición de Bluetooth.* [Online]. Available: <http://www.masadelante.com/faqs/que-es-bluetooth>
- [5] M. Capacitación, *Señales Tx y Rx.* [Online]. Available: <http://cursos.mcilelectronics.cl/señales-tx-y-rx-1/>