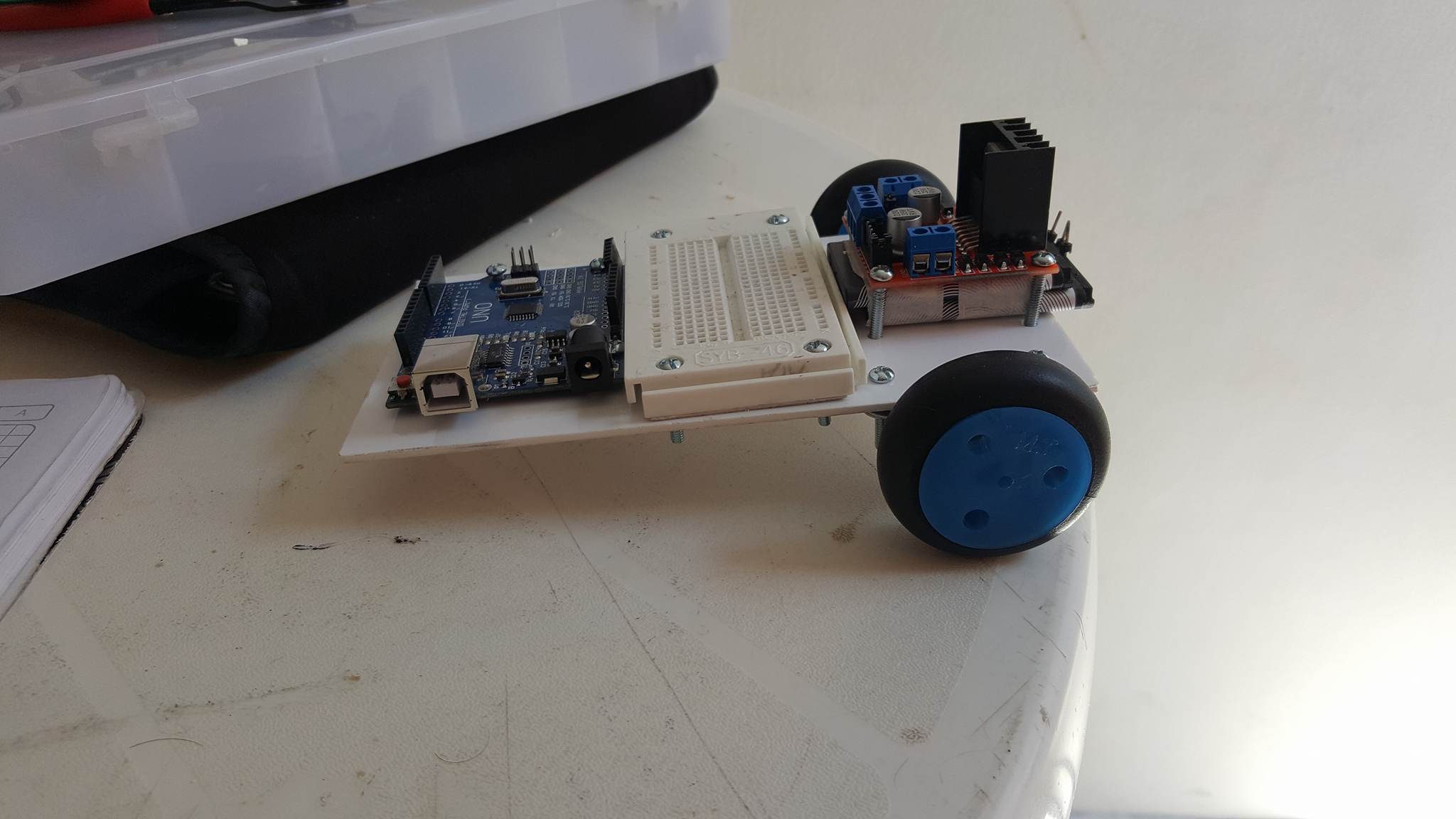
**Documentación 2 Avance Proyecto 2: Seguidor de Línea**

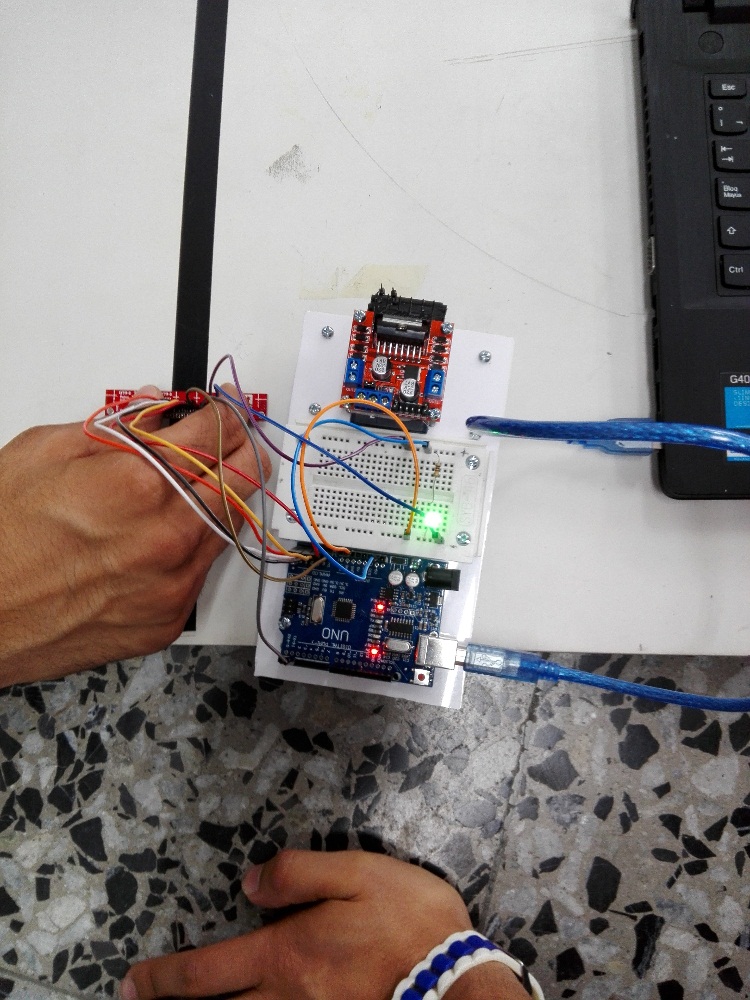
**Presentado por: Andrés Felipe Ardila- Diego Leonardo Cárdenas- Maycol Sabogal**

Para el segundo avance del proyecto 2 de microprocesadores se ha realizado el montaje físico del seguidor de línea para el cual se realizó la base del seguidor se conectaron 2 baterías de celular en serie de 1.02 Amperios por hora y 3.5 Voltios dicho montaje se puede ver apreciado en la figura 1, además de un puente H con el cual alimentaremos el arduino 1 utilizando un capacitor y una resistencia para eliminar el ruido de la salida y así estar lo más pura posible.



**Figura.1 montaje seguidor de línea**

Para continuar también se usaron los sensores por medio del ejemplo de la librería qtrsensor para la plataforma arduino la toma un numero deseado que es el número de sensores que se desean calibrar luego por medio de un for este va imprimiendo en pantalla el valor de cada sensor en la pantalla del arduino como se aprecia en la figura 2 y 3



**Figura.2 calibración del sensor**



**Figura.3 valores en pantalla de la calibración del sensor**

Se ha estudiado las funciones de la librería qtrsensor.h y sus funcionamientos son los siguientes:

void read(unsigned int \*sensorValues, unsigned char readMode = QTR\_EMITTERS\_ON)

Esta función lee el valor que le llega al sensor el cual está en un intervalo de cero a 1023 con el readMode se escoge si para la lectura el emisor debe estar encendido con QTR\_EMITTERS\_ON o apagado con QTR\_EMITTERS\_OFF

void emittersOff()

Esta función apaga el led emisor del sensor

void calibrate(unsigned char readMode = QTR\_EMITTERS\_ON)

esta función calibra los sensores por lo cual el valor del sensor no lo importa internamente en el código y con los valores máximos y minimos que arroja los importa en un arreglo al final de calibración

void readCalibrated(unsigned int \*sensorValues, unsigned char readMode = QTR\_EMITTERS\_ON)

Esta función toma el valor que arroja el sensor y lo lee con un intervalo entre 0 y 1000 donde 0 representa el valor mínimo del sensor y 1000 el valor máximo del sensor

void QTRSensorsRC::init(unsigned char\* digitalPins, unsigned char numSensors, unsigned int timeout = 2000, unsigned char emitterPin = QTR\_NO\_EMITTER\_PIN)

Esta function toma los valores de los pines digitales del arduino que usaran los sensores y el tiempo en el cual se decidirá el tiempo en el cual se dejara en blanco o negro para que se detecte bien la línea por la cual el seguidor de línea usara por lo cual se retornan valores de 0 y 1

void init(unsigned char\* analogPins, unsigned char numSensors, unsigned char numSamplesPerSensor = 4, unsigned char emitterPin = QTR\_NO\_EMITTER\_PIN)

Esta función toma el pin análogo al cual va a llegar la información del sensor además del número de muestras que se tomaran de cada sensor y el pin del sensor que va a emitir señal y por último el número de sensores que se usaran