



Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

Sistemas Operativos Avanzados

Sistema Embebido Arduino

Segundo Cuatrimestre 2017

INFORME FINAL

Proyecto “AutoMAC”

Integrantes			
Apellido	Nombre	DNI	E-mail
Allegretti	Juan Martín	37.607.521	juan.elca@gmail.com
Longo	Lautaro	33.745.357	lautaro.ariel.longo@gmail.com
Ortiz	Gabriela	37.661.538	gabrielaortizacosta@gmail.com
Turko Soto	Carolina	38.010.118	turkosotocarolina@gmail.com
Villafañe	Facundo	38.707.272	Facundo.villafañe.59@gmail.com

VERSIÓN: 1

Contenido

Misión	3
Visión	3
Descripción General	3
Hardware utilizado	3
Componentes	3
Sensores	3
Actuadores	3
Otros componentes	3
Descripción Componentes	4
Sensor HC-SR04	4
Sensor LDR (Light Dependent Resistor)	4
Sensor FC-03 (Tacómetro)	5
Led (Diodo emisor de luz)	5
Motor DC con caja Reductora + Chasis	6
Puente H Integrado	6
Arduino UNO	7
Módulo Bluetooth HC-05	7
Diagrama de Conexión	8
Arduino	8
Diagramas	8
Diagrama de la Solución	8
Sistema Embebido	9
Desplazamiento	9
Distancia Frontal	9
Diagrama de Hardware	10
Protocolo de Comunicación	11
Sistema Embebido - Arduino	11
Funcionamiento	11
Aplicación Mobile	12
Funcionamiento	12
Problemas	12
Desarrollo del Proyecto.	13
Arduino	13
Aplicación Mobile	15

Misión

La misión de AutoMAC consiste en construir un dispositivo que deberá ser operado desde un dispositivo móvil conectado inalámbricamente, controlando su velocidad y la dirección del mismo.

Visión

Nuestra visión apunta a construir un dispositivo que sirva como base para distintos sistemas embebidos más complejos. Contando con la posibilidad de agregar más funcionalidades y convertir, por ejemplo, AutoMAC en la funcionalidad base de una aspiradora o un automóvil a control remoto.

Descripción General

AutoMAC es un dispositivo que se controla mediante una aplicación mobile, desde la cual le enviamos la acción a realizar: avanzar, retroceder y frenar, junto con la velocidad a la cual desplazarse. El dispositivo tiene la capacidad de detectar obstáculos y esquivarlos, a medida que se acerca al obstáculo se informa con una luz encendida a mayor intensidad que si estuviera lejos del obstáculo. También detecta luz cuando lo alumbramos, de la cual se aleja.

Hardware utilizado

Componentes

Sensores

- HC-SR04 (Sensor de ultrasonido)
- LDR (Light Dependent Resistor)
- FC03 x2 (Tacómetro)

Actuadores

- Led (Diodo emisor de luz - color rojo)
- Motor DC con (reductora) x2

Otros componentes

- Arduino UNO (ATMega 328P)
- Protoboard
- Puente H
- Módulo Bluetooth HC-05
- Una resistencia de 100 ohm para led
- Batería EB-L1G6LLU x2
- Cables varios para conexión con el protoboard

Descripción Componentes

Sensor HC-SR04

Este sensor, se maneja con un **TRIGGER** (disparador) de ultrasonido y un **ECHO**. El **TRIGGER** envía una señal de ultrasonido, esta rebota contra el objeto más próximo y vuelve hacia el dispositivo que lo captura por el **ECHO**. Entonces este sensor en realidad mide, el tiempo (en microsegundos) que tarda en volver la señal del ultrasonido por medio del **Clock** y realizando un cálculo se puede calcular la distancia (en milímetros), ya que la señal de ultrasonido viaja a una velocidad constante.



Sensor LDR (Light Dependent Resistor)

Un LDR es un resistor que varía su valor de resistencia eléctrica dependiendo de la cantidad de luz que incide sobre él. El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (en algunos casos puede descender a tan bajo como 50 ohms) y muy alto cuando está a oscuras (puede ser de varios megaohms).

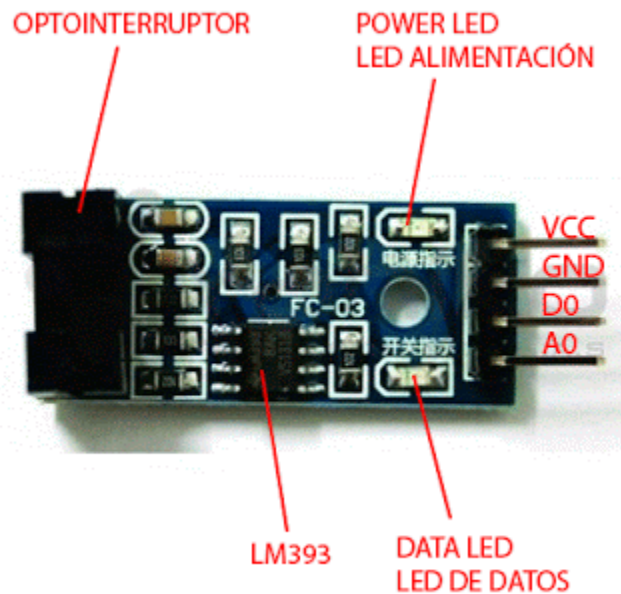
Los LDR se fabrican con un cristal semiconductor fotosensible como el sulfuro de cadmio (CdS). Estas celdas son sensibles a un rango amplio de frecuencias lumínicas, desde la luz infrarroja, pasando por la luz visible, y hasta la ultravioleta.



Sensor FC-03 (Tacómetro)

Con este módulo podemos controlar la velocidad de rotación de las ruedas de nuestro robot. Si colocamos una corona dentada que gira unida a nuestra rueda. También se podría usar como un interruptor óptico.

El funcionamiento básico de este sensor es el siguiente; Si se hace pasar cualquier cosa entre la ranura del sensor, este crea un pulso digital en el pin D0. Este pulso va de 0V a 5V y es una señal digital TTL. Luego con Arduino podemos leer este pulso.



Led (Diodo emisor de luz)

Usamos el led color Rojo para informar según la intensidad de la luz una correspondencia con la distancia a la que está encuentra el auto de un obstáculo.

Además, cuando se encuentra activo el modo Automático se utiliza el led “titilando” para informar la luz del ambiente es normal.



Motor DC con caja Reductora + Chasis

Se eligió un chasis con dos ruedas y cada una con un motor DC (CC) con caja reductora. Para alimentar cada uno de los motores usamos una batería portátil que le pueda dar el voltaje necesario.

Voltaje	Parámetros	3V DC	5V DC	6V DC
Parámetros del motor (sin caja reductora)	RPM	6000 RPM		
	Corriente	80-100mA		
Parámetros de la caja reductora	Reducción	48:1		
	Velocidad sin carga	125 RPM	200RPM	230RPM
	Velocidad con Carga	95RPM	160RPM	175RPM
	Torque de salida	0.8kg.cm	1.0 kg.cm	1.1 kg.cm
	Corriente	110-130mA	120-140mA	130-150mA
	Diámetro máximo de Rueda	6.5cm		
	Dimensiones	70mmx22mmx18mm		
	Peso	50g		
	Ruido	<65dB		



Para lograr calcular la velocidad que podía alcanzar nuestro auto utilizamos los datos de la tabla e hicimos los siguientes cálculos.

$$160 \text{ rpm} = 16,76 \text{ rad/s}$$

$$16,76 \text{ rad/s} * 0.035 \text{ m} = 0,5866 \text{ m/s}$$

$$0,5866 \text{ m/s} = 2,112 \text{ km/s}$$

Puente H Integrado

Un Puente en H es un circuito electrónico que permite a un motor DC girar en ambos sentidos, pero también puede usarse para frenarlo (de manera brusca), al hacer un corto entre las bornas del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta.

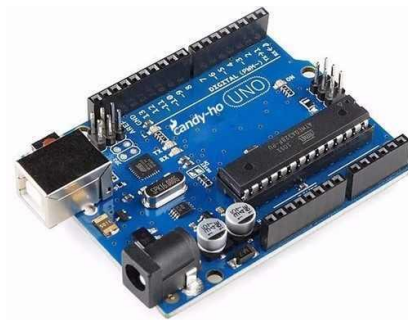
En nuestro proyecto lo utilizamos para lograr que el auto avance, retroceda y gire.



Arduino UNO

La placa Arduino UNO puede tomar información del entorno a través de sus entradas analógicas y digitales para controlar casi cualquier cosa como luces, motores y otros actuadores. El software de programación de esta placa es abierto, la distribución de sus pines es XBee. Tiene 6 pines de entrada analógica y además es compatible con campos extra para funciones como Bluetooth, infrarrojo y otros sensores.

- Microcontrolador: ATMEGA328P
- Alimentación: 5 – 12 Vcc
- Frecuencia de operación: 16 MHz
- Puertos de entrada análoga: 6
- Puertos de entrada/salida digital: 13 (incluyendo puertos PWM)
- Capacidad de memoria flash: 32 Kb
- Salida PWM
- Salida de voltaje: 5 Vcc
- Switch reset
- Comunicación a la PC
- Software Arduino



Módulo Bluetooth HC-05

Es un módulo Bluetooth que tiene la particularidad de que puede ser configurado como maestro o como esclavo. Se lo implementó para la conexión con el dispositivo android bajo el protocolo RS 232 Serial, configurado como esclavo ya que la conexión se realizará con un solo dispositivo Android a la vez.

Este modelo cuenta con un botón de reset.

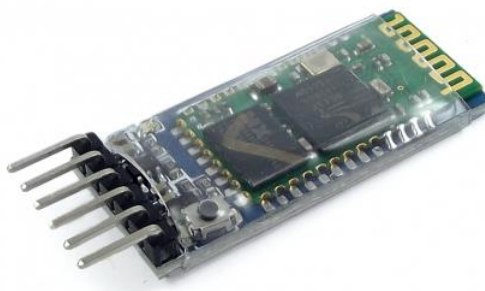
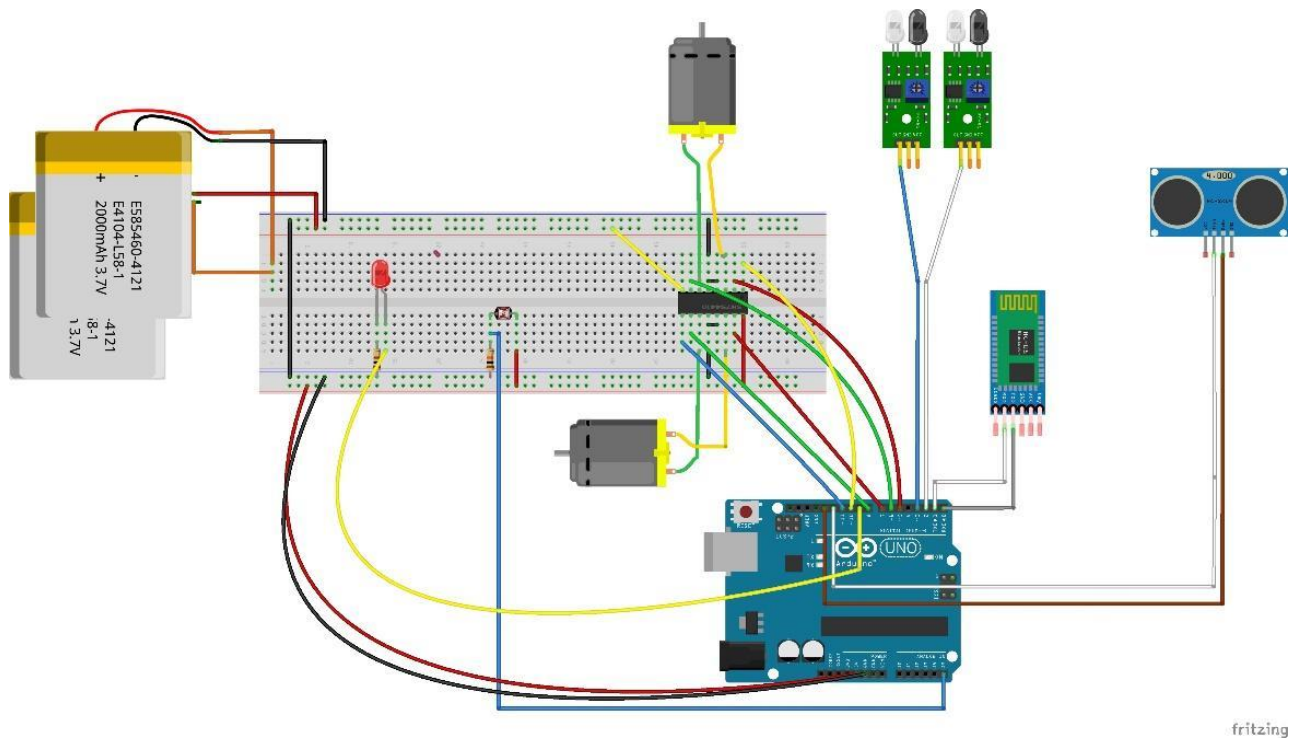


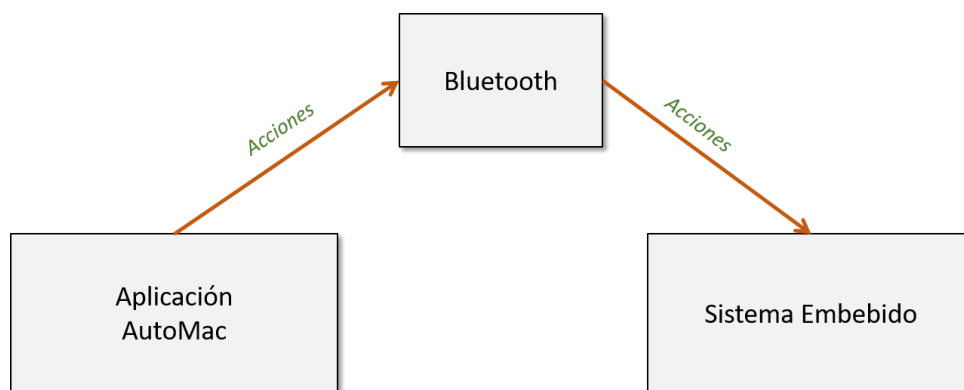
Diagrama de Conexión

Arduino

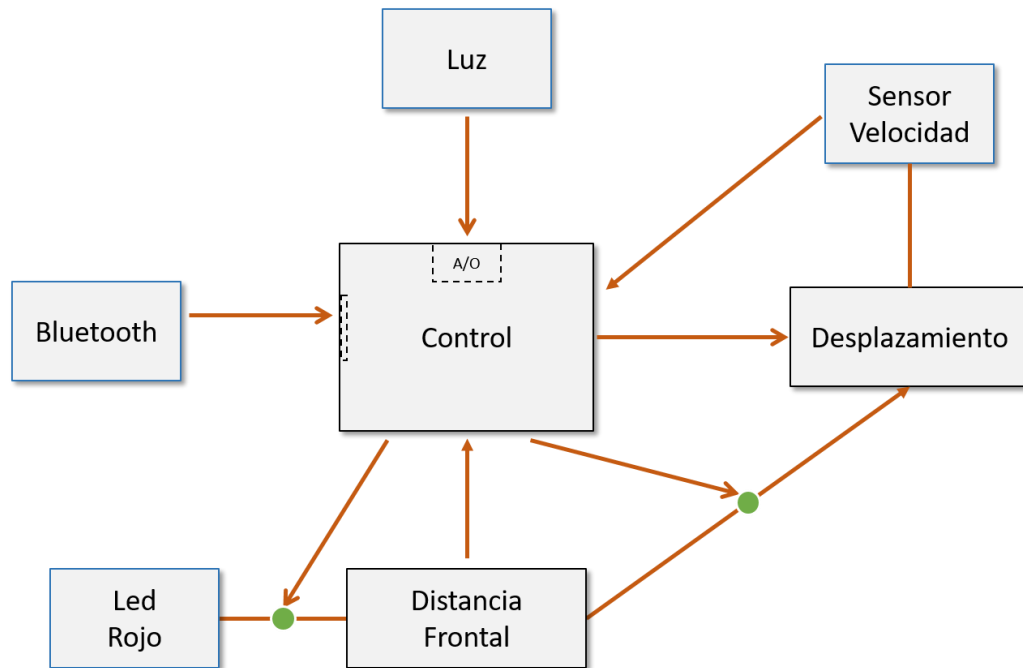


Diagramas

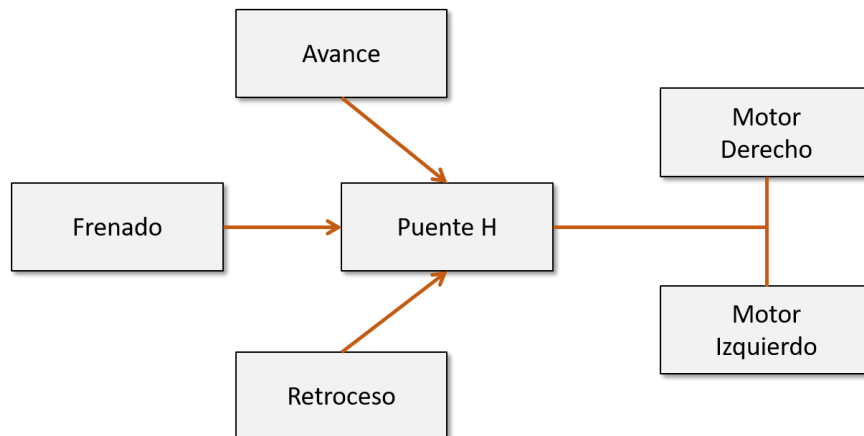
Diagrama de la Solución



Sistema Embebido



Desplazamiento



Distancia Frontal

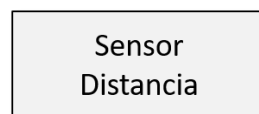
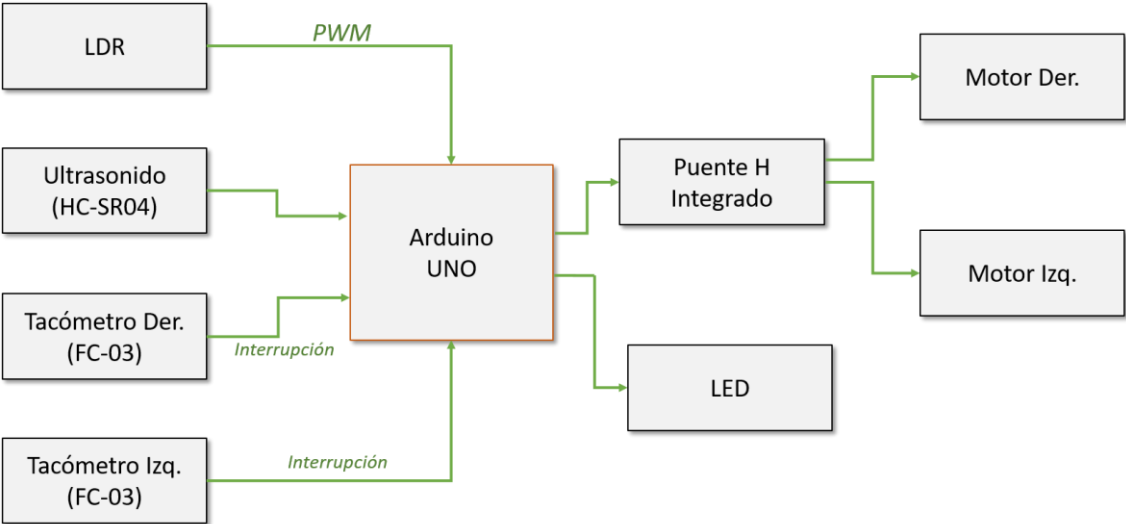


Diagrama de Hardware



Protocolo de Comunicación

Para la implementación de nuestra solución definimos un protocolo de comunicación entre la aplicación mobile y el sistema embebido. Para eso utilizamos letras en minúsculas y les definimos una funcionalidad a cada una.

Letra	Función
a	Adelante
r	Reversa
b	Doblar Derecha Min
c	Doblar Izquierda Min
d	Doblar Derecha Medio
e	Doblar Izquierda Medio
g	Doblar Derecha Max
h	Doblar Izquierda Max
i	Doblar Reversa Derecha Min
j	Doblar Reversa Izquierda Min
k	Doblar Reversa Derecha Medio
l	Doblar Reversa Izquierda Medio
m	Doblar Reversa Derecha Max
n	Doblar Reversa Izquierda Max
f	Frenar
p	Encender Funcionamiento
o	Apagar Funcionamiento
z	Encender Led por detección de Luz
y	Apagar Led por no detección de Luz

Sistema Embebido - Arduino

Funcionamiento

El sistema embebido responde a las acciones que le envía la aplicación mobile a través de la conexión Bluetooth.

- Avanzar
- Retroceder
- Detenerse
- Encender
- Encender Led
- Apagar Led
- Variar velocidad de los motores.

Además, realiza el control de los valores captados por los sensores y toma decisiones a partir de ellos, esto incluye (Ordenado por prioridad):

- **Control de Distancia:** a través del sensor de ultrasonido se calcula la distancia a la que se encuentra un obstáculo.
Si esa distancia es de 40 cm se comienza a encender la luz del Led Rojo; aumentando la intensidad a medida que la distancia que se calcula va disminuyendo.
Si la distancia es menor de 15 cm se comienza a disminuir la velocidad hasta el frenado. Una vez que frena, se gira el auto para esquivar el obstáculo.
- **Control de Velocidad:** a través de los sensores Tacómetro que están en las ruedas se mide, en cantidad de pulsos a qué velocidad se encuentra cada motor. Eso se va a utilizar para la realimentación de los motores, esto se debe a que es necesario mantener la misma potencia en ambos motores y el auto se mantenga derecho.
- **Control Lumínico:** a través del sensor LDR se capta la intensidad de la luz del ambiente, con rangos de X y X.

Si capta un rango muy bajo de luz, se inicia el funcionamiento automático del auto.
Si capta un rango "normal" (limites que definimos según la luz del ambiente) no realiza ninguna acción.
Si capta un rango superior, el auto comienza a avanza en sentido contrario desde donde esté la luz.

Aplicación Mobile

Funcionamiento

La aplicación mobile es la encargada en primera instancia de establecer la conexión entre el sistema Android y el Sistema Embebido a través de una Conexión Bluetooth. Para ello muestra una lista de los dispositivos Bluetooth que el sistema tiene vinculados desde el cual se tiene que elegir el correspondiente al Sistema Embebido.

Una vez establecida la conexión, tenemos acceso a la pantalla de Controles de la aplicación, desde allí se van dirigir las acciones del auto.

La aplicación brinda la posibilidad de realizar las siguientes acciones:

- Encender y Apagar el auto a través de un componente Switch.
- Avanzar y Retroceder
- Frenar
- Variar la velocidad a puntos determinados.

Utilizamos lo sensores:

- Sensor Lumínico:

```
sensorManager.getDefaultSensor (Sensor.TYPE_LIGHT) ;
```

- Sensor de proximidad:

```
sensorManager.getDefaultSensor (Sensor.TYPE_PROXIMITY) ;
```

- Sensor Acelerómetro:

```
sensorManager.getDefaultSensor (Sensor.TYPE_ACCELEROMETER) ;
```

Tomamos esta decisión en base a que estos sensores se encuentran disponibles en la mayoría de los dispositivos móviles.

Problemas

A continuación, vamos a nombrar algunos inconvenientes que fueron surgiendo a lo largo de todo el proyecto, algunos se pudieron solucionar y otros no:

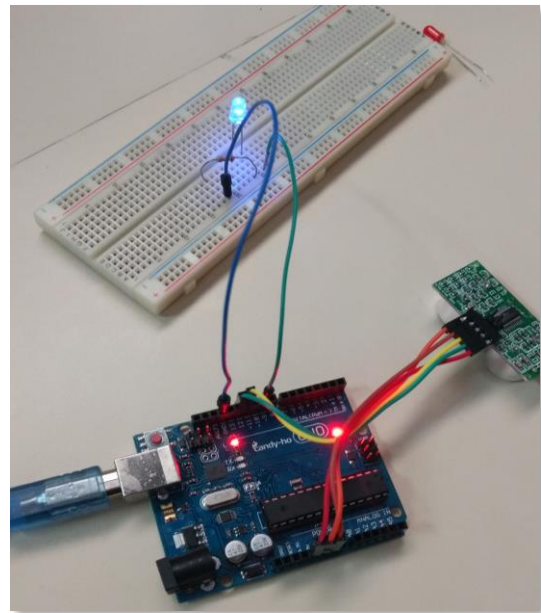
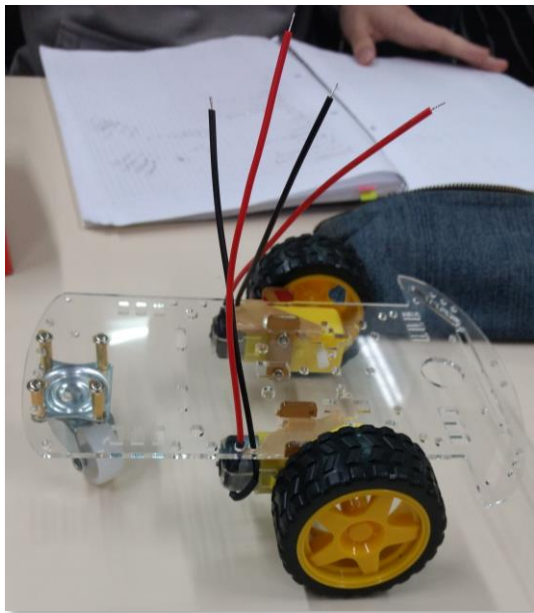
- Al comienzo del proyecto decidimos probar cada uno de los sensores que íbamos a utilizar por separado para corroborar que funcionaban correctamente. Al momento de probar el sensor de ultrasonido (**US100**) no podíamos lograr que funcione, luego de muchos intentos vimos que teníamos que desconectar el **Jumper** que estaba ubicado en la parte inferior del mismo. Esta se debía a que solo en el modo Serial UART es necesario que el Jumper esté conectado al sensor.
- Con el sensor de ultrasonido tuvimos además otros inconvenientes ya que no podíamos lograr calcular de forma correcta la distancia a partir de los datos que nos devolvía el sensor. Después de investigar cuál podía ser la razón decidimos que lo mejor era cambiar el modelo de mismo, entonces conseguimos un Sensor de Ultrasonido modelo HC-SR04.

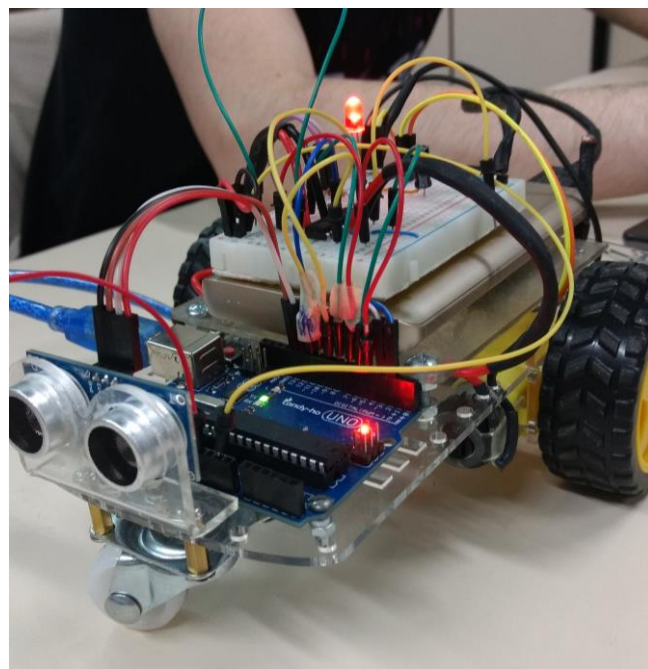
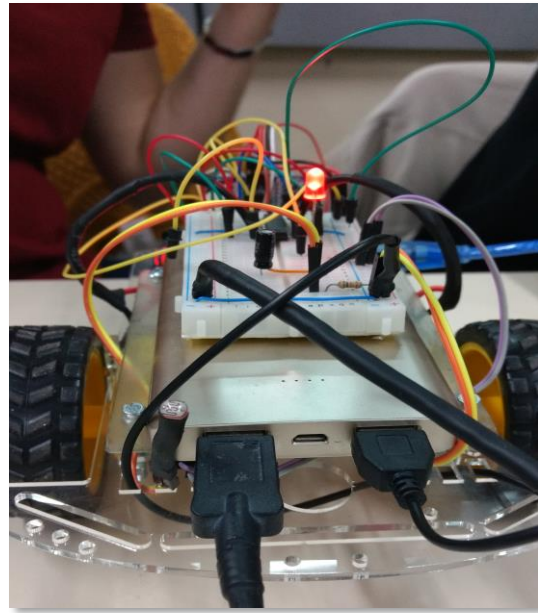
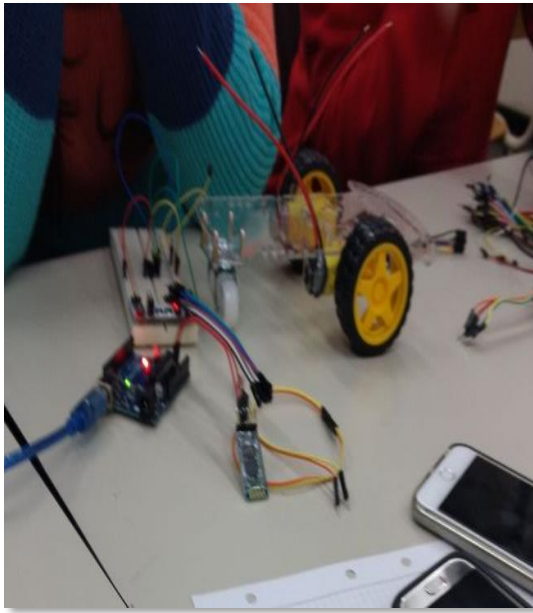
- Por otro lado, se nos presentó una situación con los motores, ya que al entregarle a cada uno la misma cantidad de pulsos PWM uno iba a mayor velocidad que el otro. Al querer solucionar esta nos resultó difícil encontrar la exactitud de pulsos PWM que le teníamos que enviar a cada uno. Entonces para que el auto vaya derecho tuvimos que regular el PWM para que ambos motores vayan a la misma velocidad siendo ayudado también por la realimentación que nos brindan los datos obtenidos desde los Tacómetro ubicados en cada una las ruedas.
- Por el lado de Android, tuvimos inconvenientes al momento de realizar la comunicación Arduino-Android. El objetivo era enviar los pulsos que median los Tacómetros de las ruedas a Android para que este pueda realizar los cálculos correspondientes y así obtener la velocidad promedio a la que estaba el auto.
Al momento de realizar la conexión la pudimos hacer, pero los valores que se enviaban desde Arduino no llegaban en Tiempo Real a Android, tenía un retardo que no pudimos tomarlo como correcto. Por lo tanto, decidimos que íbamos a tomar los pulsos de las ruedas manualmente una única vez y esos valores los íbamos a dejar estáticos dentro de la aplicación Mobile.

Desarrollo del Proyecto.

A continuación, están algunas imágenes de cómo se fue desarrollando el proyecto, haciendo foco en el Sistema Embebido.

Arduino





Aplicación Mobile

