

Vehículo a Control Remoto

Informe de Avance

Taller de Proyecto I | Ingeniería en Computación

Arreche, Cristian
Blasco, Federico
Borini, Ángel
Paradiso, Martín

10 de octubre de 2019

Facultad de Ingeniería
Universidad Nacional de La Plata



Índice general

| | |
|---|-----------|
| 1. Introducción | 2 |
| 2. Objetivos | 3 |
| 3. Requerimientos | 5 |
| 3.1. Requerimientos Funcionales | 5 |
| 3.1.1. Requerimientos de Software | 5 |
| 3.1.2. Requerimientos de Hardware | 5 |
| 3.2. Requerimientos No Funcionales | 6 |
| 4. Cronograma | 7 |
| 5. Diseño de Hardware | 8 |
| 5.1. Diagrama en Bloques | 8 |
| 5.1.1. Recepción Inalámbrica | 8 |
| 5.1.2. Controlador de Motores | 10 |
| 5.1.3. Detector de Obstáculos | 11 |
| 5.1.4. Indicadores de Estado | 13 |
| 5.2. Cálculos de Corriente | 14 |
| 6. Diseño de Software | 16 |
| 6.1. Pseudocódigo de los Módulos | 16 |
| 6.1.1. Leer Bluetooth | 16 |
| 6.1.2. Detectar Colisiones | 17 |
| 6.1.3. Actualizar Motores | 17 |
| 6.1.4. Software Android | 18 |
| 6.2. Interfaz de Usuario | 20 |
| 6.2.1. Pasos a seguir para la configuración del sistema | 20 |
| 6.2.2. Utilización del Vehículo | 21 |
| 6.3. Librerías de Software | 21 |
| 7. División de Tareas del Grupo | 22 |
| 8. Bibliografía | 23 |
| 9. Anexo | 24 |
| 9.1. Datasheets | 24 |
| 9.2. Información general | 24 |
| 9.3. Aplicaciones Externas | 24 |
| 9.4. Esquemático Completo | 25 |

1. Introducción

Una de las principales aplicaciones que se le ha dado a la electrotecnia es el control de motores. En el ámbito de la electrónica se hace uso de pequeños motores con poca potencia que funcionan con corriente continua, como por ejemplo para mover autos a escala.

En los últimos años, con la aparición de plataformas de desarrollo como Arduino, se han desarrollado autos a control remoto artesanales; que proveen una gran diversidad de resultados debido a la plataforma abierta y variedad de componentes electrónicos.

La motivación del proyecto es obtener un producto que cuente con cierto nivel de complejidad y, al mismo tiempo, que sea visualmente gratificante. Consideramos que el control de un vehículo a control remoto conlleva una parte importante de investigación y desarrollo; y brinda un resultado que es tangible.

El proyecto es un vehículo móvil, equipado este con motores eléctricos que reaccionan dependiendo de las señales enviadas desde un mando inalámbrico del que depende el movimiento del automóvil. Además, colocaremos una serie de sensores para agregar mayor funcionalidad al proyecto reaccionando ante distintos eventos.

2. Objetivos

El objetivo principal del proyecto es el diseño de un vehículo a escala, que pueda ser controlado remotamente por los usuarios.

El proyecto se divide en 4 partes principales:

- Estructura física del vehículo
- Control de motores
- Control remoto
- Medición de sensores y respuesta en consecuencia

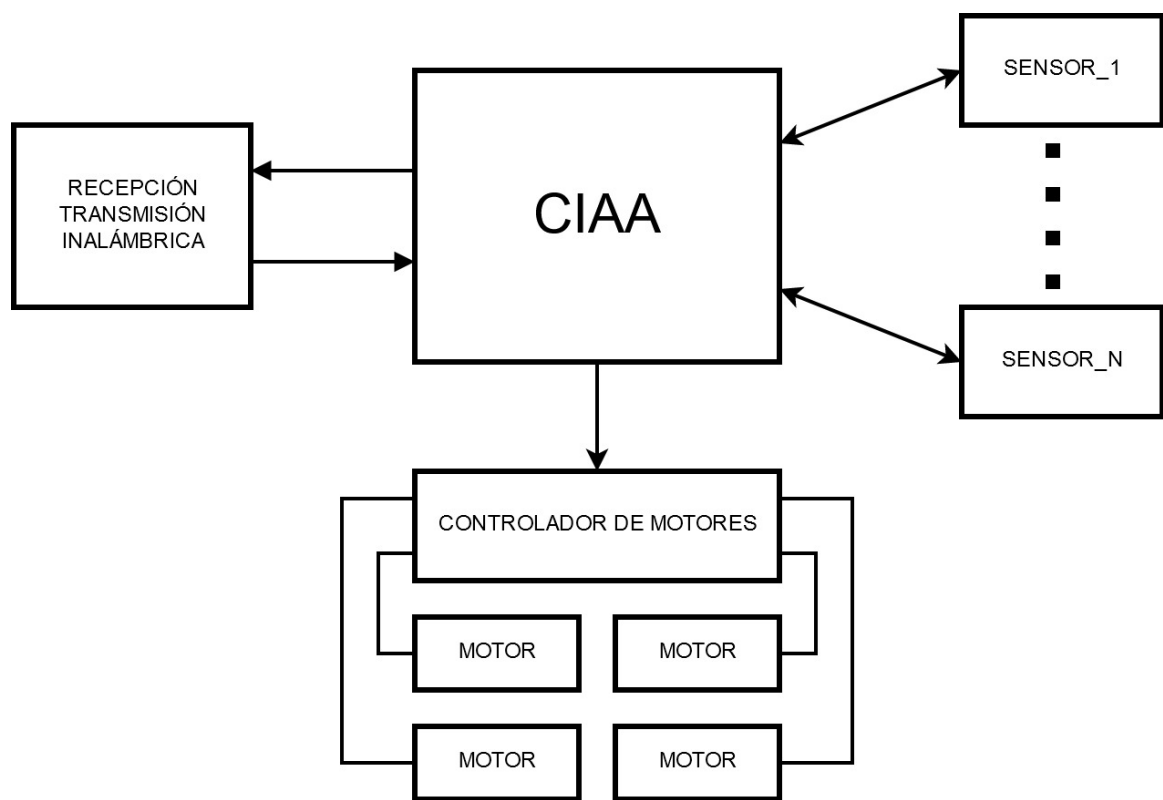


Figura 1. Diagrama en bloques del sistema

Estructura física del vehículo El desarrollo del vehículo implica tener una estructura sobre la cual se puedan montar los motores con sus respectivos controladores, sensores, la EDU-CIAA y la/s baterías necesarias para alimentar el sistema. La estructura deberá ser lo suficientemente rígida para soportar el peso del hardware mencionado anteriormente, y al mismo tiempo debe ser liviana para que los motores puedan transportarla sin la necesidad de una potencia excesiva.

Control de Motores Los motores deben permitir el movimiento en ambos sentidos, de manera de rotar el vehículo sin utilizar un complejo sistema de servos y bisagras. Los controladores se encuentran sobre el poncho, con salidas para cada motor, haciendo el sistema independiente de los motores y estructura del vehículo, como se puede observar en la *Figura 1*.

Control Remoto El objetivo es controlar el vehículo de manera remota a través de protocolo bluetooth. Debe permitir manejar el vehículo desde una distancia considerable (entre 10m y 20m), permitiendo realizar todos los movimientos: avanzar, retroceder, y girar hacia izquierda y derecha. También es deseable disponer de comandos auxiliares para agregar funcionalidad adicional, como por ejemplo, encender/apagar luces manualmente.

Medición de sensores y respuesta en consecuencia Como objetivos secundarios del proyecto, consideramos la colocación de algunos sensores:

- Sensor LDR, el cual ante la ausencia de luz se encienden luces delanteras y traseras del vehículo
- Sensores de proximidad por infrarrojo, los cuales controlan el vehículo ante los obstáculos con los que se encuentre

3. Requerimientos

Aquellos requerimientos indicados con **OPT** son completamente opcionales.

3.1. Requerimientos Funcionales

3.1.1. Requerimientos de Software

Control de Motores

1. El sistema debe accionar 4 motores individualmente, permitiendo girar en ambas direcciones
2. El sistema debe accionar motores en conjunto, permitiendo avanzar, retroceder y girar
3. El sistema debe permitir el bloqueo de movimiento en una dirección (por ejemplo debido a un obstáculo detectado por el sensor de proximidad)
4. **OPT** El sistema debe indicar mediante una serie de LEDs el estado de movimiento del vehículo

Control Remoto

1. El sistema debe establecer una conexión bluetooth con el sistema de control remoto
2. El sistema debe leer las indicaciones del control remoto y ejecutar las acciones correspondientes
3. El sistema debe indicar mediante un LED el estado de la conexión bluetooth

OPT Medición Sensores

1. Ante ausencia de luz externa, se deben encender las luces del vehículo
2. Ante la presencia de obstáculos, se debe frenar el desplazamiento en esa dirección

3.1.2. Requerimientos de Hardware

Control de Motores

1. El vehículo debe utilizar puentes H para motores de corriente continua
2. **OPT** El vehículo debe Indicar con un LED para cada motor si este está en movimiento

Baterías y cargador

1. El vehículo debe utilizar baterías de Litio para lograr la autonomía del vehículo
2. El vehículo debe indicar mediante un LED cuando haya baja tensión
3. El vehículo debe permitir cargar las baterías mediante un puerto USB

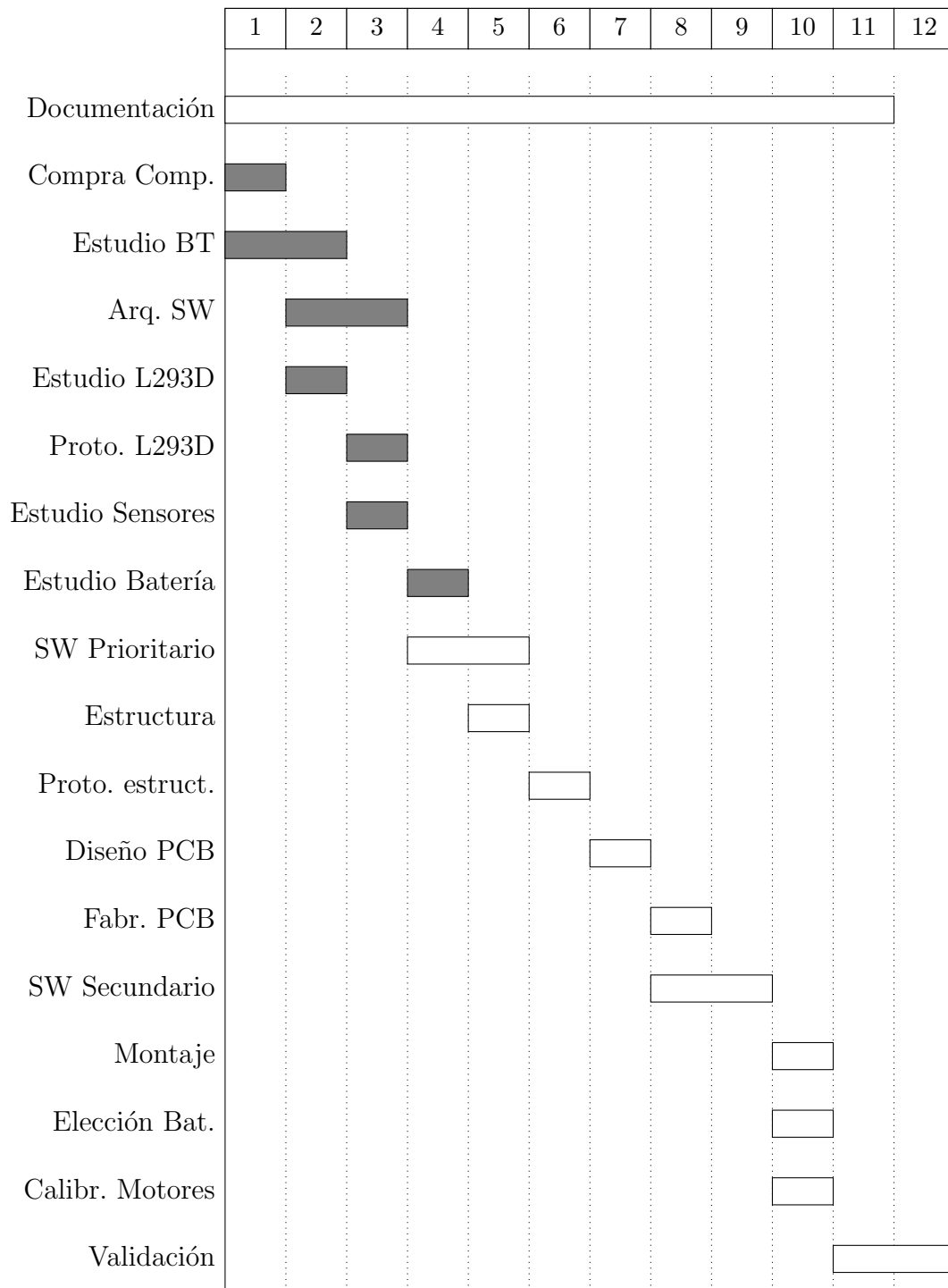
Componentes

1. Se debe contar con una llave de encendido/apagado y un LED que determine su estado
2. **OPT** El sistema debe poseer un buzzer para indicar eventos o fallas

3.2. Requerimientos No Funcionales

- Utilización de la placa de desarrollo EDU-CIAA
- Estructura sólida y liviana, donde se colocan los motores y la EDU-CIAA
- El conexionado a los motores debe ser a través de borneras para poder cambiar la estructura.
- Programación en lenguaje C
- Desarrollo de PCB en formato de poncho
- Mecanismo de control simple
- El sistema debe manejar respuestas en tiempo real.
- Fecha de finalización y entrega del proyecto el día 16/12

4. Cronograma



Las unidades de tiempo son semanas. Consideramos dos semanas de Validación como margen en caso de eventualidades que atrasen el proyecto.

5. Diseño de Hardware

5.1. Diagrama en Bloques

5.1.1. Recepción Inalámbrica

Para implementar el manejo por control remoto se utiliza el protocolo Bluetooth, más específicamente el módulo HM-10, que funciona como una comunicación serie inalámbrica. Del lado del usuario se utiliza la aplicación de Android "BLEJoystick", disponible en Google Play Store. También se cuenta con un LED para indicar la correcta vinculación entre la app y el módulo bluetooth.

Componentes a Utilizar

| Componente | Código | Valor | Cantidad | Función | Protocolo |
|------------|--------|----------|----------|--|-----------|
| Rx/Tx BLE | HM-10 | - | 1 | Recepción | UART 8N1 |
| LED | - | Rojo 3mm | 1 | Informar estado de la conexión Bluetooth | Digital |

Rx/Tx BLE

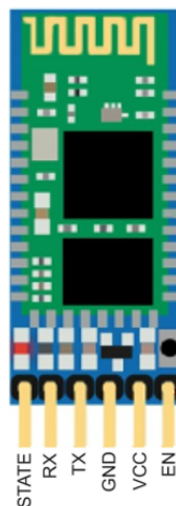


Figura 5.1: HM-10

El módulo Bluetooth a utilizar es el HM-10 provisto por la cátedra, el cual ya fue utilizado en asignaturas anteriores por lo cual se conocen sus características y su funcionamiento. Además, la sAPI de la CIAA provee una librería para comunicación Bluetooth, facilitando su programación.

Respecto a las características de Hardware, el HM-10 funciona a 3.3v, por lo que no hay que realizar ninguna adaptación para su funcionamiento.

LED

Para la alimentación del LED se utiliza el puerto GPIO1, realizando los siguientes cálculos se obtiene el valor de la resistencia que necesita para su conexión::

$$R = \frac{V - LED_V}{LED_I} = 420\Omega$$

$$V = 3,3v; LED_V = 1,2v; LED_I = 5mA;$$

Conexionado

HM-10 El módulo Bluetooth se comunica a través de un protocolo serie 8N1, a 9600 bps, y se configura a través de comandos AT. Las librerías de la CIAA ya proveen funciones para establecer una comunicación con un módulo HM-10.

LED Se conecta de manera directa a una salida digital del integrado, teniendo en cuenta la resistencia mencionada.

Esquemático

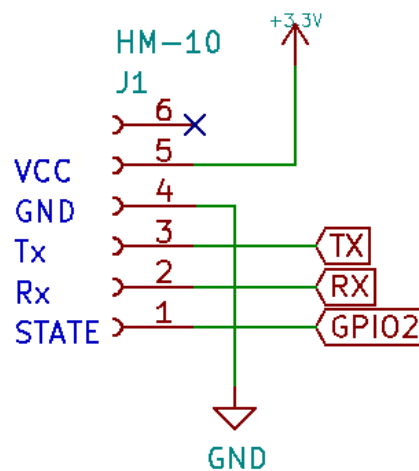


Figura 5.2: Conexionado del Módulo Bluetooth HM-10

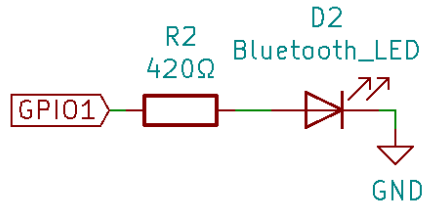


Figura 5.3: Conexionado del LED Bluetooth

5.1.2. Controlador de Motores

Para el control de motores se utiliza el circuito integrado L293D, en nuestro caso utilizamos la versión de STMicroelectronics, el cual provee dos puente H completos, permitiendo así controlar hasta dos motores en ambos sentidos.

Componentes a Utilizar

| Componente | Código | Valor | Cantidad | Función | Protocolo |
|------------|--------|-------|----------|-------------------|---------------|
| Puente H | L293D | - | 2 | Controlar motores | Analógico/PWM |

Puente H

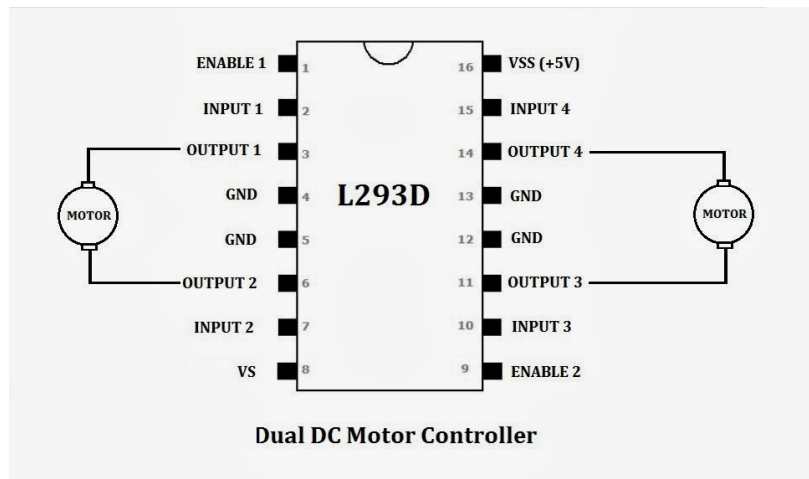


Figura 5.4: L293D

Al investigar sobre circuitos que provean puentes H para controlar motores de continua, rápidamente encontramos el L293D. Este tiene 4 canales, cada uno formando medio puente H. Dado que para girar motores en ambos sentidos se requiere un puente completo, se pueden controlar dos motores con un solo L293D.

En términos de conexionado provee un *enable* por cada par de canales, alimentación independiente para lógica y motores, y 6 pines centrales para disipación de calor.

Respecto a la alimentación, requiere entre 4.5v y 30v. Dado el elevado consumo es imposible alimentarlo directamente a través de la CIAA, por lo que se conecta de manera directa a la fuente de alimentación.

Soporta corrientes pico de 1.2A, por lo que es correcto colocar un disipador en caso de que los motores consuman mucha corriente.

Conexionado

L293D Las entradas de los L293D se controlan mediante PWM, permitiendo controlar la velocidad de cada motor de manera independiente. Esto requiere utilizar 8 PWM, dos para cada motor.

Esquemático

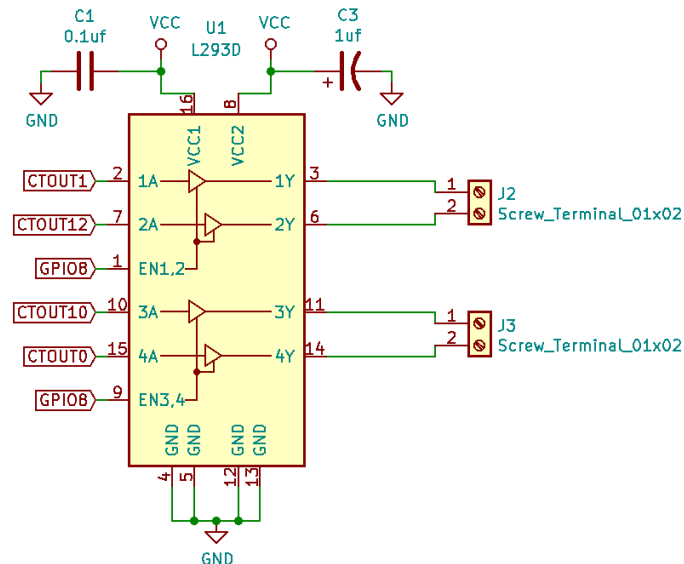


Figura 5.5: Conexionado de un L293D

5.1.3. Detector de Obstáculos

Para detectar obstáculos se utiliza el sensor de proximidad de Arduino HC-SR04, que permite captar objetos entre 2cm y 400cm, con un ángulo de 15°. Estos rangos resultan ideales para detectar obstáculos y actuar en consecuencia en un vehículo a control remoto.

Componentes a Utilizar

| Componente | Código | Valor | Cantidad | Función | Protocolo |
|-------------------|---------|-------|----------|------------------------------|-----------|
| Sensor Proximidad | HC-SR04 | - | 1 | Medir distancia a obstáculos | Digital |
| Buzzer | - | - | 1 | Indicar mediante sonido | Digital |

Sensor de Proximidad

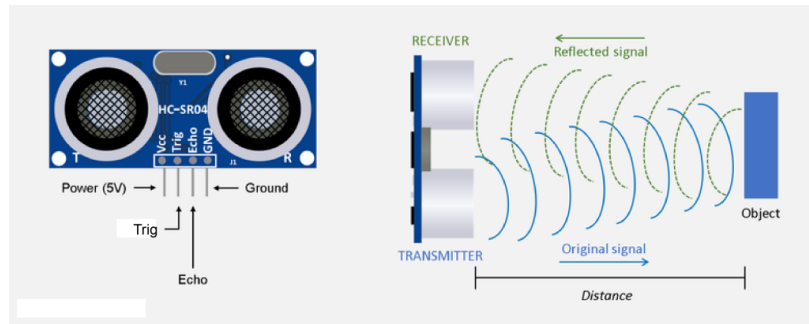


Figura 5.6: HC-SR04

El HC-SR04 es un sensor ultrasónico que permite calcular la distancia de los objetos que se encuentran en frente a él, con un ángulo de visión de 15°. Utiliza una onda sonora de 40kHz, inaudible para el oído humano, que se emite cuando se coloca un 1 en el pin *trigger*. Cuando el módulo detecta el rebote del pulso emitido, coloca un 1 en el pin *echo*. Midiendo el tiempo entre la emisión del pulso y la recepción del eco, puede determinarse la distancia al objeto.

Buzzer

Se utiliza como componente auxiliar para indicar la presencia de obstáculos mediante sonido.

Conexionado

HC-SR04 Dado que requiere medir con precisión el tiempo transcurrido, es necesario utilizar un timer en modo *input capture* para lograr la mayor precisión posible. El *trigger* se generará con una salida GPIO común.

Buzzer Dado que el consumo del buzzer supera el máximo tolerado por los puertos GPIO de la CIAA, se utiliza un transistor para alimentar el buzzer y asegurar su correcto funcionamiento sin dañar ningún componente de hardware. Se utiliza el pin de GPIO7 como salida para el LED.

Esquemático

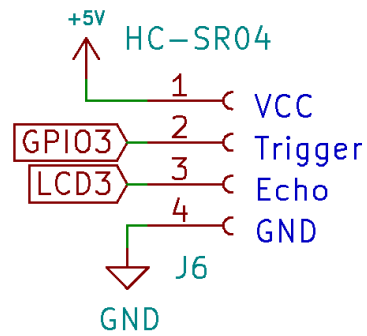


Figura 5.7: Conexión del HC-SR04

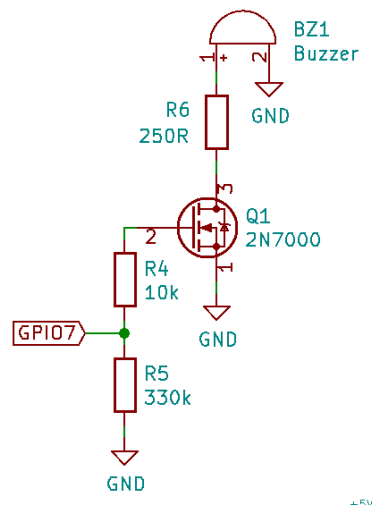


Figura 5.8: Conexión del Buzzer

5.1.4. Indicadores de Estado

Se utiliza como componente auxiliar para indicar el estado del vehículo, un LED que indica si el vehículo está encendido.

Esquemático

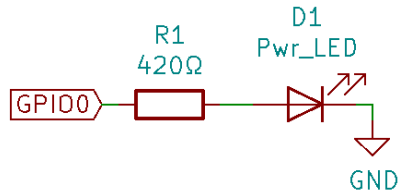


Figura 5.9: Conexión del LED de encendido

5.2. Cálculos de Corriente

| Subsistema | Tensión [V] | Corriente [mA] |
|--------------|-------------|----------------|
| Principal | 3.3 | 100 |
| UART2 (BT) | - | 0.5 |
| SCT * 3 | - | 0.16 * 3 |
| GPIO * 6 | - | 5 * 6 |
| HM-10 | 3.3 | 10 |
| L293D * 2 | 5 | 32 * 2 |
| Motor * 4 | 5 | 100 * 4 |
| Total | 3.3 | 140 |
| Total | 5 | 500 |

Como se puede observar, dado la variedad de dispositivos y componentes, se requieren dos valores de tensión diferentes. Además, al tratarse de un vehículo manejado inalámbricamente, es necesario el uso de baterías. Utilizando baterías de Litio, que poseen tensiones de entre 3.7v y 4.2v, es necesario utilizar un convertor DC-DC step-down para obtener tensiones de 3.3v, y un convertor step-up para obtener tensiones de 5v.

Esquemático

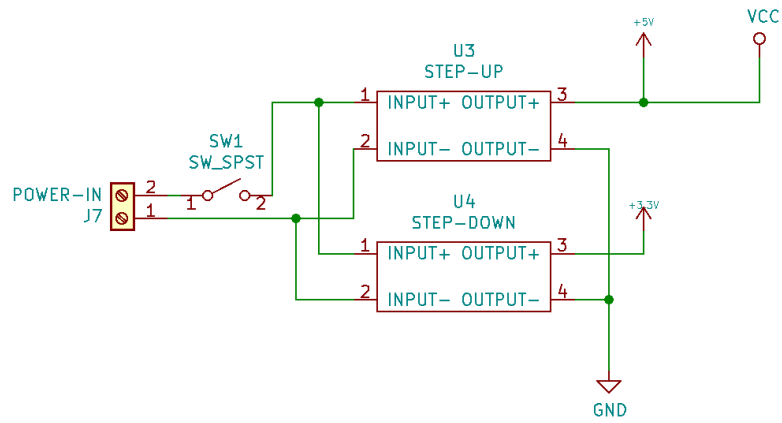


Figura 5.10: Conexión de la alimentación

6. Diseño de Software

Se optó por utilizar una arquitectura Time-Triggered cooperativa, donde las tareas se corresponden con los subsistemas de Hardware:

- Control de motores
- Recepción inalámbrica
- Detector de obstáculos

Además de la interrupción de tiempo, propia de la arquitectura Time Triggered, se necesita una interrupción de tiempo independiente utilizada por el sensor de proximidad para medir con precisión la distancia a la que se encuentran los obstáculos.

El pseudocódigo del programa principal se resume a lo siguiente:

```
interrupcion 50ms:
```

```
    flag_tiempo = 1
```

```
variables compartidas:
```

```
    accion_vehiculo # Enumerativo que indica el desplazamiento que  
                    # se debe realizar
```

```
programa principal:
```

```
    inicializacion motores  
    inicializacion bluetooth  
    inicializacion detector de colisiones  
    prender led power
```

```
super loop:
```

```
    si flag_tiempo = 1:  
        leer bluetooth  
        detectar colisiones  
        actualizar motores  
        flag_tiempo = 0
```

6.1. Pseudocódigo de los Módulos

6.1.1. Leer Bluetooth

```
si hay mensajes nuevos:
```

```
    leer el mensaje
```

```
    si es una A o a:
```

```
        motor = avance
```

```

si es una C o c:
    motor = retrocede
si es una D o d:
    motor = giro izquierda
si es una B o b:
    motor = giro derecha
si es un 0:
    motor = giro libre
si es una F o f:
    motor = freno

```

6.1.2. Detectar Colisiones

El eco producido por el módulo se genera -aproximadamente- entre 60µS y 12mS. Creemos que no es necesario realizar las mediciones cada estos periodos tan cortos de tiempo, dado que, puede ser contraproducente para los tiempos del sistema considerando el objetivo del sensor. Por esto, el procesamiento del se realiza cada 250ms:

```

inicializar detector de colisiones:
    tick_colision = 0

detectar colisiones:
    tick_colision++
    si tick_colision = 5:
        tick_colision = 0
        tomar tiempo
        prender input capture
        enviar señal de comienzo de medicion

si llego el echo:
    calcular distancia
    si hay un obstaculo cerca:
        bloquear movimiento hacia adelante

```

6.1.3. Actualizar Motores

Los motores se controlan en dos niveles: a bajo nivel se puede controlar cada motor de manera independiente, indicando la dirección de giro. Una capa más “alta” permite controlar el vehículo más facilmente, brindando funciones para indicar como se debe mover el vehículo.

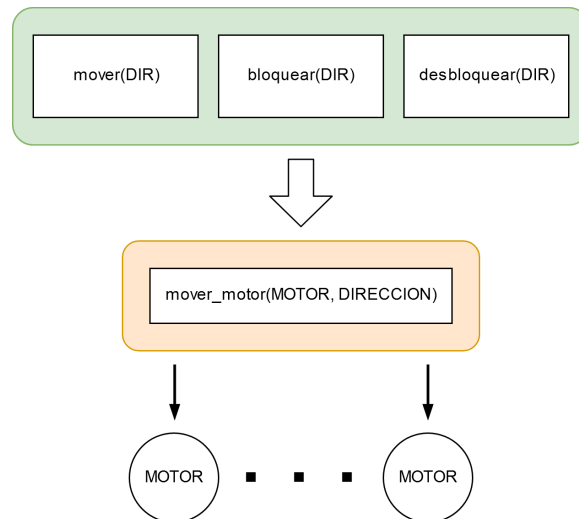


Figura 6.1: Capas de los Motores

Por lo que el pseudocódigo se reduce a lo siguiente

inicializacion de motores:

setear movimiento en libre

actualizacion de motores:

ver accion_vehiculo

mover el vehiculo en la direccion solicitada

A modo de ejemplo se explica como mover en una sola direccion
si movimiento es hacia adelante:

motor trasero izquierdo girar horario

motor trasero derecho girar antihorario

motor delantero izquierdo girar horario

motor delantero derecho girar horario

...

A bajo nivel se controla cada motor individualmente:

si movimiento es horario:

colocar canal 1 en alto

colocar canal 0 en bajo

si movimiento es anti horario:

colocar canal 0 en bajo

colocar canal 1 en alto

6.1.4. Software Android

BLEJoystick BLEJoystick es la aplicación a utilizar para el control inalámbrico del vehículo, está disponible para múltiples dispositivos android y se puede descargar

de manera gratuita desde Google Play Store. Esta aplicación se conecta al módulo bluetooth HM-10 mediante el protocolo bluetooth low energy 4.0 (BLE). Cuenta con 8 botones, donde el funcionamiento de cada uno es el siguiente:

- Cuando se presiona por primera vez, envía una letra en mayúscula.
- Mientras siga presionado el mismo botón, envía la misma letra en minúscula de manera constante.
- Cuando se suelta el boton, envía un 0.

A continuación se describe cada uno de los botones y su letra correspondiente:

| Botón | Letra inicial | Letra constante |
|-----------|---------------|-----------------|
| Arriba | A | a |
| Derecha | B | b |
| Abajo | C | c |
| Izquierda | D | d |
| Triángulo | E | e |
| Círculo | F | f |
| Equis | G | g |
| Cuadrado | H | h |

6.2. Interfaz de Usuario

6.2.1. Pasos a seguir para la configuración del sistema

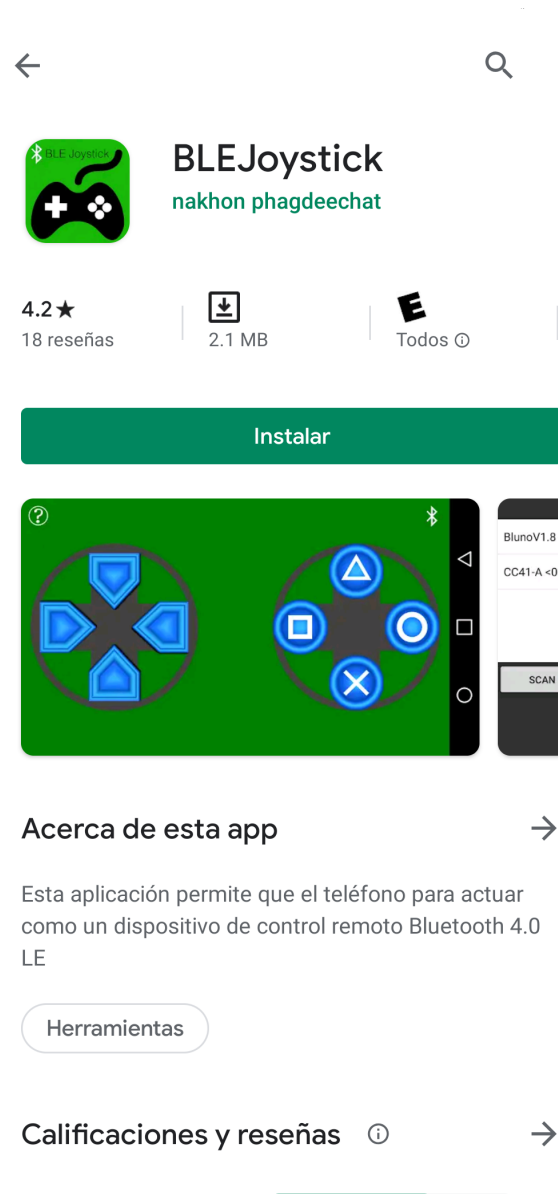


Figura 6.2: Aplicación en el Play Store

1. Ingresar a Google Play Store
2. Descargar la aplicación BLEJoystick
3. Iniciar la aplicación.
4. Pulsar el ícono de vinculación bluetooth y seleccionar *vehículo*.

5. Corroborar la correcta vinculación mediante el LED indicador.
6. Sistema listo para usar.

6.2.2. Utilización del Vehículo

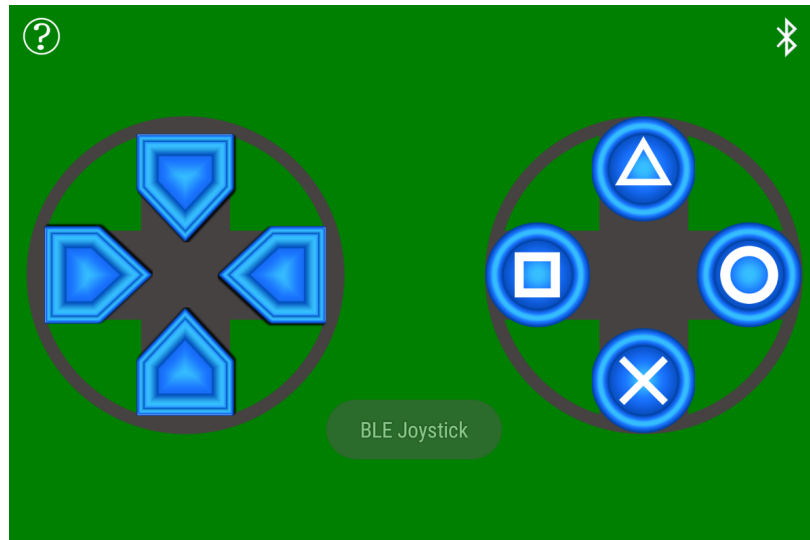


Figura 6.3: La aplicación BLEJoystick

A continuación se detalla y explica el funcionamiento de cada uno de los botones disponibles en la aplicación

| Botón | Función |
|-----------|--------------------------|
| Arriba | Desplazar hacia adelante |
| Derecha | Girar hacia la derecha |
| Abajo | Desplazar hacia abajo |
| Izquierda | Girar hacia la izquierda |
| Círculo | Frenar |

6.3. Librerías de Software

sAPI Dada la extensión de la librería provista por la CIAA, solo se requiere usar esta. Más específicamente se utilizarán las librerías `sapi_pwm.h`, `sapi_gpio.h` y `sapi_timer.h`.

7. División de Tareas del Grupo

| Tarea a realizar | Alumno/s |
|--|-----------------------------------|
| Obtención de electrónica | Arreche, Blasco, Borini, Paradiso |
| Estudio de protocolos Bluetooth | Blasco, Paradiso |
| Arquitectura de software | Arreche, Blasco, Borini, Paradiso |
| Estudio de L293D | Arreche, Borini |
| Prototipo L293D | Arreche, Paradiso |
| Investigación de sensores | Arreche, Paradiso |
| Investigación de baterías y carga | Borini, Blasco |
| Desarrollo de software prioritario | Arreche, Blasco, Borini, Paradiso |
| Diseño y fabricación de estructura | Blasco, Borini |
| Proto. de estructura con motores y L293D | Arreche, Blasco, Borini, Paradiso |
| Diseño de la PCB | Blasco, Paradiso |
| Fabricación de PCB | Paradiso |
| Desarrollo de SW secundario y sensores | Arreche, Blasco |
| Montaje final | Arreche, Blasco, Borini, Paradiso |
| Elección de baterías | Borini, Paradiso |
| Calibración de motores | Blasco, Borini |
| Validación | Arreche, Blasco, Borini, Paradiso |
| Documentación | Arreche, Paradiso |

8. Bibliografía

1. Hoja de datos del microcontrolador LPC4337 utilizado por la EDU-CIAA-NXP. Disponible en: https://www.nxp.com/docs/en/data-sheet/LPC435X_3X_2X_1X.pdf
2. Hoja de datos de L293D. Disponible en: <http://www.ti.com/lit/ds/symlink/1293.pdf>

9. Anexo

9.1. Datasheets

- Transistor 2n7000 Disponible: <https://www.onsemi.com/pub/Collateral/2N7000-D.PDF>
- L293D Texas Instrument Disponible: <http://www.ti.com/lit/ds/symlink/l293.pdf>
- L293D STMicroelectronics Disponible: <https://www.st.com/resource/en/datasheet/l293d.pdf>
- HC-SR04 <http://raspoid.com/download/datasheet/HCSR04>
- HC-SR04 <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf4>
- HC-SR04 <https://www.electroschematics.com/hc-sr04-datasheet/>

9.2. Información general

- HM-10 https://github.com/ciaa/firmware_v3/blob/master/examples/c/sapi/bluetooth/hm10_uart_bridge/EDU-CIAA-NXP\%20y\%20BLE\%204.0\%20HM10.pdf

9.3. Aplicaciones Externas

- BLEJoystick <https://play.google.com/store/apps/details?id=iyok.com.blejoystick>

9.4. Esquemático Completo

