



Laboratorio de Microprocesadores - 86.07

## Tacho de basura inteligente

Profesor	Ing. Guillermo Campiglio
Cuatrimestre/Año:	2/2019
Turno de las clases prácticas:	Miércoles
Jefe de trabajos prácticos:	Ing. Pedro Ignacio Martos
Docente guía:	Ing. Fabricio Baglivo
Autores:	Aupi, Santiago José - P:100793 Demarco, Facundo Agustín - P:100620 Gayet, Denise Giselle - P:100828
Seguimiento del proyecto	

### Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P	

Coloquio	
Nota final	
Firma profesor	

# Índice

1. Introducción	1
2. Objetivos propuestos	1
3. Objetivos logrados	1
4. Descripción del Hardware	1
5. Descripción del Software	3
6. Conclusiones y posibles mejoras	5
7. Bibliografía y recursos	5

## 1. Introducción

A continuación, se presenta el proyecto correspondiente a la asignatura Laboratorio de Microprocesadores. El presente documento tiene como objetivo dar una total descripción del proyecto desarrollado, comparando las funcionalidades propuestas con las implementadas.

## 2. Objetivos propuestos

Se trata de una plataforma móvil con la capacidad de desplazarse hacia cierta ubicación y retornar a su lugar de origen mediante una conexión bluetooth *linkeada* a una aplicación en el celular. La misma posee dos comandos, uno para iniciar el avance del carro y otro para retornarlo a la posición inicial. Además esta plataforma servirá de base para sostener un cesto de basura que tendrá una tapa la cual se abrirá con la ayuda de un servo motor controlado por un sensor de proximidad.

## 3. Objetivos logrados

Se realizó la plataforma móvil que logra desplazarse hacia cierta ubicación siguiendo una línea negra, a partir de una orden de acercamiento enviada mediante bluetooth, pudiendo luego regresar a su lugar de origen a través de otro comando de retirada. Finalmente, el cesto de basura utilizado no tiene tapa, por lo que no se implementaron tanto el sensor de proximidad, cuyo propósito hubiera sido determinar cuándo abrir dicha tapa, como el servo motor propuesto en un principio.

## 4. Descripción del Hardware

Para elaborar la plataforma móvil se requirió en primer lugar de un microcontrolador que en este caso fue el Atmega 328P, el cual está embebido en la placa Arduino.

Para el movimiento del pequeño automóvil se hizo uso de dos motores de continua los cuales se prenden o apagan dependiendo la ruta que se deba seguir. Esto se determina con 4 sensores infrarrojos, dos que sirven para el camino de ida y dos que se implementaron para el de vuelta. El principio de funcionamiento es tal que si los sensores llegan a ver negro se apaga la tracción de los motores sobre las ruedas y si observa blanco se alimentan con una tensión de 5v a través de un puente-H. Este último se empleó no

solo para alimentar los motores, si no también para controlarlos, permitiendo avanzar o retroceder gracias a la posibilidad de cambiar la polaridad de los mismos.

Por otra parte, para establecer una orden sobre el dispositivo se implementó un modulo bluetooth hc-05, el cual recibía una señal desde una aplicación de celular creada por el programa del MIT '*app inventor*'.

El circuito se montó en una placa experimental. Cabe destacar que se alimenta por un lado a dos sensores y el puente-H que alimenta a los motores, y por el otro al Arduino, el módulo bluetooth y los dos restantes infrarrojos. Esto se hace así para evitar riesgos debido a que el puente-H requiere mayor corriente que el Arduino. En adición a esto se requirieron cables de protoboard macho-macho y macho-hembra para las conexiones entre sensores-Arduino entre Arduino-puente H y hc05-microcontrolador.

A continuación se muestra el esquemático:

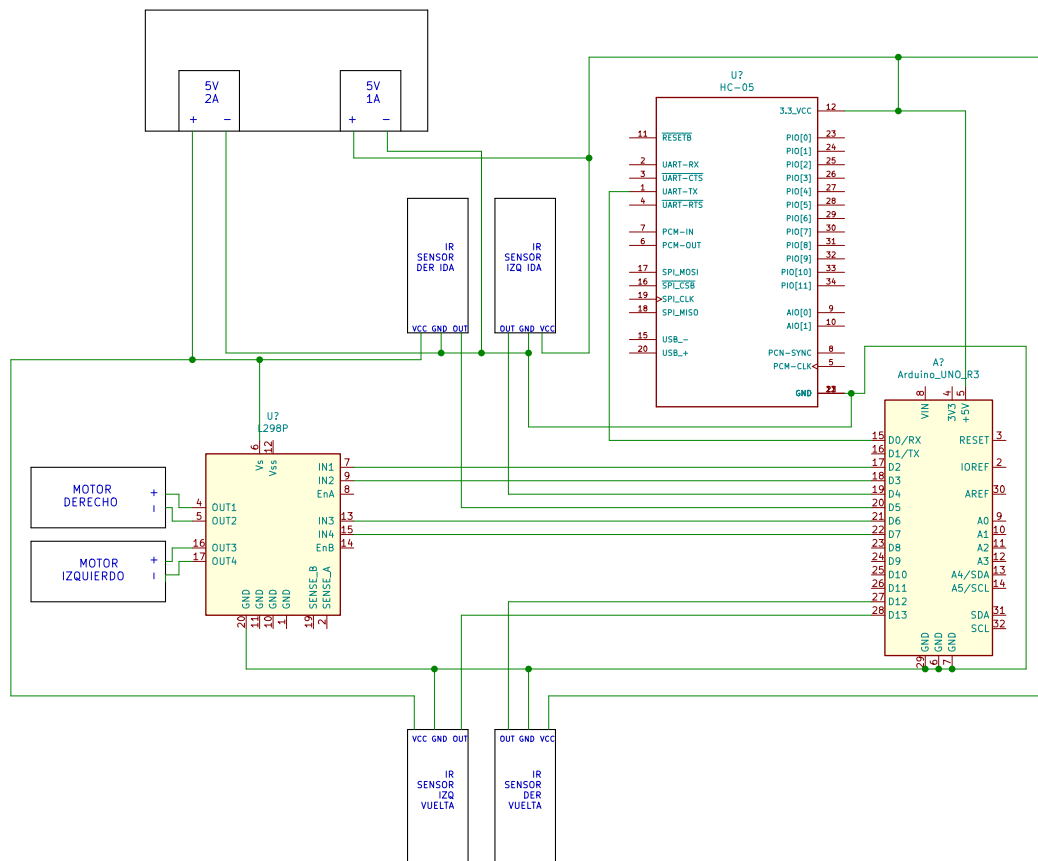


Figura 1: Esquemático del circuito

## 5. Descripción del Software

El software responsable del funcionamiento de la plataforma móvil puede dividirse principalmente en tres secciones. Por un lado la configuración básica del hardware a través de los *seteos* correspondientes siguiendo la hoja de datos del micro. Es decir, establecer qué pines del microcontrolador serán utilizados como entrada/salida. Por otro lado la configuración de la interrupción principal, USART, la cuál será responsable de poner en funcionamiento la plataforma cuando sea requerido. Y como última sección, el código correspondiente al control y funcionamiento de los motores. Estas tres secciones unidas dan lugar al software final que rige el comportamiento de la plataforma móvil de inicio a fin.

A continuación se explica en más detalle cada sección: Empezando por las configuraciones de los pines de entrada/salida. Dado que se utilizan 4 sensores infrarrojos que enviarán datos al microcontrolador y se deben manejar 2 motores a través de 2 pines cada uno, se tienen 4 pines configurados como entrada, para los sensores, y 4 configurados como salida, para los motores. En cuanto a la configuración de la interrupción USART, se habilitó la interrupción correspondiente la recepción de datos en USART (RXCIE), la misma se activa cuando hay datos sin leer en el buffer, es decir al enviar una señal a través del bluetooth, y se borra una vez leídos. Se estableció un paquete conformado por 8 bits de datos y 1 bit de stop para la comunicación. Luego, mediante la interrupción de RX, se da al inicio del movimiento. Dependiendo de si el usuario eligió que el cesto se acerqué o retorne a su lugar de origen se leerá un valor diferente en cada caso en el buffer y en base a ello se llama una rutina de avance o de retroceso. Mediante el PIN-D en el avance, y el PIN-B en el retroceso se leen los valores obtenidos por los sensores infrarrojos, es decir, los encargados de dictar el movimiento de la plataforma. Mediante los pines del PUERTO-D se ejecutan distintas combinaciones para controlar el sentido de rotación de los motores.

En la **tabla 1** se muestra la tabla de verdad que rige el movimiento de cada motor a través del puente-H.

En la **tabla 2** se exhibe una tabla de verdad que muestra las combinaciones del motor acorde a los valores leídos en los sensores.

IN1/IN3	IN2/IN4	Movimiento
0	0	frenar
1	0	avanzar
0	1	retroceder
1	1	frenar

Tabla 1: Comportamiento del Puente-H para cada motor

Sensor Izquierdo (PD5/PB5)	Sensor Derecho (PD4/PB4)	Movimiento
0	0	avanzar
0	1	girar a la derecha
1	0	girar a la izquierda
1	1	frenar

Tabla 2: Comportamiento de los motores según sea el valor leído en el sensor.

Y en la figura 2 se presenta el diagrama de flujo correspondiente al software descrito.

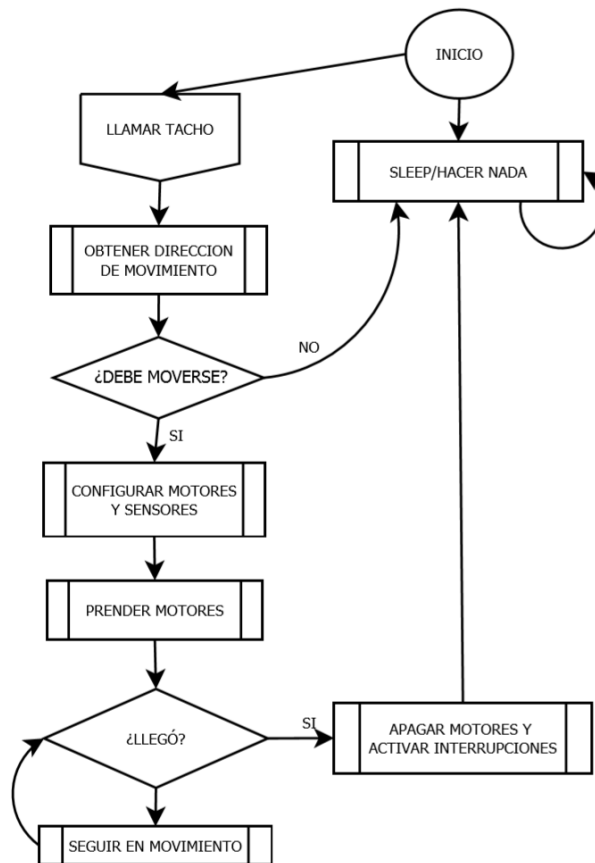


Figura 2: Diagrama de flujo del software implementado.

## 6. Conclusiones y posibles mejoras

Con el objetivo de construir el software del dispositivo se usó Assembler como lenguaje de programación con la ayuda de la interfaz Atmega Studio. El mismo es un lenguaje de bajo nivel dedicado al manejo del microcontrolador, lo cual fomentó el aprendizaje del funcionamiento del mismo, tanto a nivel de hardware como software.

En adición a esto, el uso de este micro requirió familiarizarse con su *datasheet* la cual brinda, por ejemplo, información correspondiente a cómo se deben configurar los pines tanto para salida como entrada o cómo modificar ciertos registros del micro para lograr utilizar las interrupciones tales como la USART y que funcionen de la manera deseada.

Por otra lado, al usar el módulo bluetooth se pudo entender en cierta medida cómo funciona no solo la comunicación en serie y como crear una aplicación para Android que mande información desde bluetooth, sino también cómo se programa una interrupción generada por el USART como ya se explicó anteriormente y la importancia de que tanto transmisor como receptor estén enviando y recibiendo datos bajo la misma convención.

Una de las posibles mejoras al proyecto sería la implementación de macros. De esa forma se evitaría tener código duplicado ya que sería posible ejecutar una misma rutina pero pasándole como parámetros los registros correspondientes cada vez.

Por otra parte, si no recibe ninguna orden de retirada por una cantidad corta de tiempo, 'Rolphie' volvería a su lugar usando el código de 'retirada'. Esto se podría implementar anidando interrupciones de forma tal que luego de que termine la rutina de avance se habilita el flag de interrupciones el cual posibilite al usuario enviar otro comando y si así no lo hiciera luego de cierto tiempo se llamara la rutina de retroceso saliendo finalmente de la interrupción inicial.

posibles mejoras: 1) Comando para detener el movimiento en cualquier momento. 2) Mejor ensamblaje de la conexiones.

## 7. Bibliografía y recursos

- Hoja de datos del microcontrolador:

[http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

- Set de instrucciones de Atmel:

<http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf>

- Hoja de datos del Puente-H.
- Hoja de datos del HC-05:  
<http://www.electronicaestudio.com/docs/istd016A.pdf>
- The AVR microcontroller and embedded system using assembly and C.