



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Plataforma ROVER para Aplicaciones de Robótica

Asignatura: Introducción a los Sistemas Embebidos

Carrera: Especialización en Inteligencia Artificial

Autor: Ing. Alfonso Rafel

Índice

Motivación	3
Descripción del Sistema.	3
Diagrama de Bloques	4
Conexionado de Hardware	4
Modelado de Software.	4
Próximos Pasos.	6
Links	7

Motivación

El objetivo del presente trabajo consiste en el armado de un prototipo móvil para aplicaciones de robótica e IA. La idea principal es lograr una plataforma base sobre la cual pueden agregarse sensores y funcionalidades para la prueba de distintos algoritmos, por ejemplo de localización, mapeo, visión por computador u/o aprendizaje automático.

Descripción del Sistema

La plataforma, nombrada MOBILE, consta de los siguientes componentes:

- Placa NÚCLEO F429ZI.
- 4 x motores de continua (3.3V - 5V).
- 1 x controlador de motores L298N.
- 2 x tacómetros ópticos LM393.
- 1 x sensor ultrasónico Hc-sr04.
- 1 x Potenciómetro 10kΩ
- 1 x fuente para protoboard MB102.
- 2 x baterías 18650 (3.6V c/u).

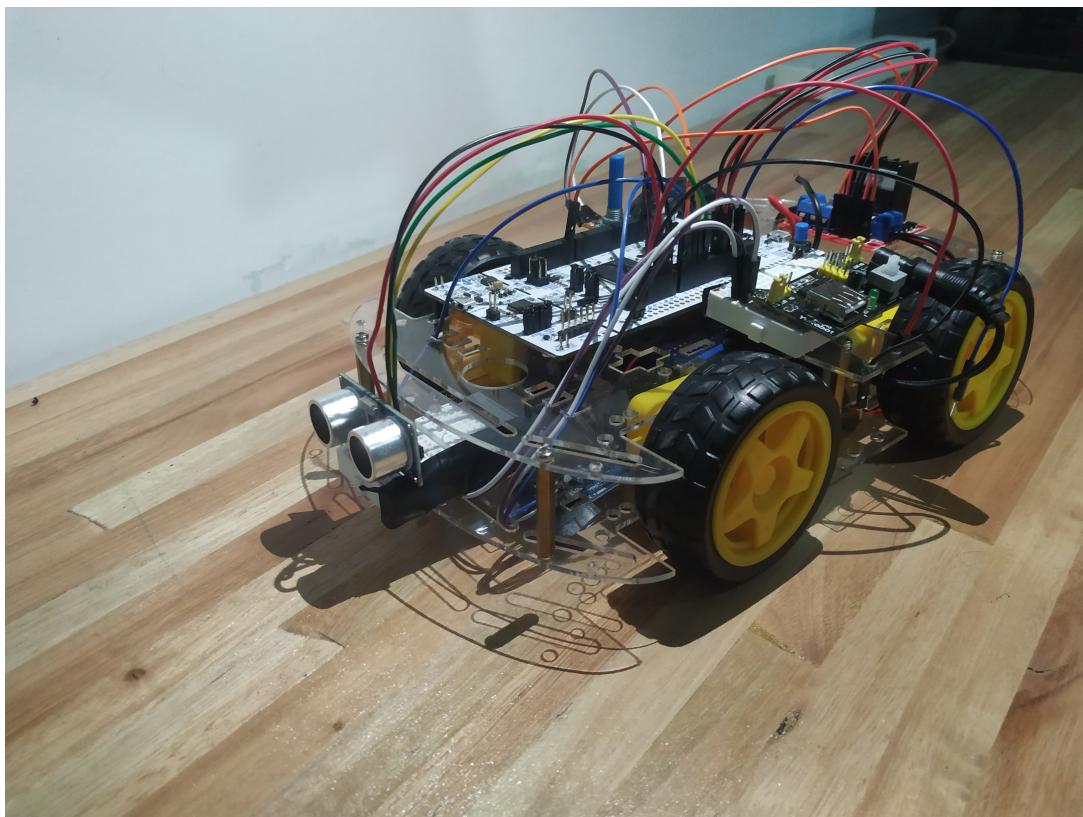


Figura 1. Plataforma MOBILE

Diagrama de Bloques

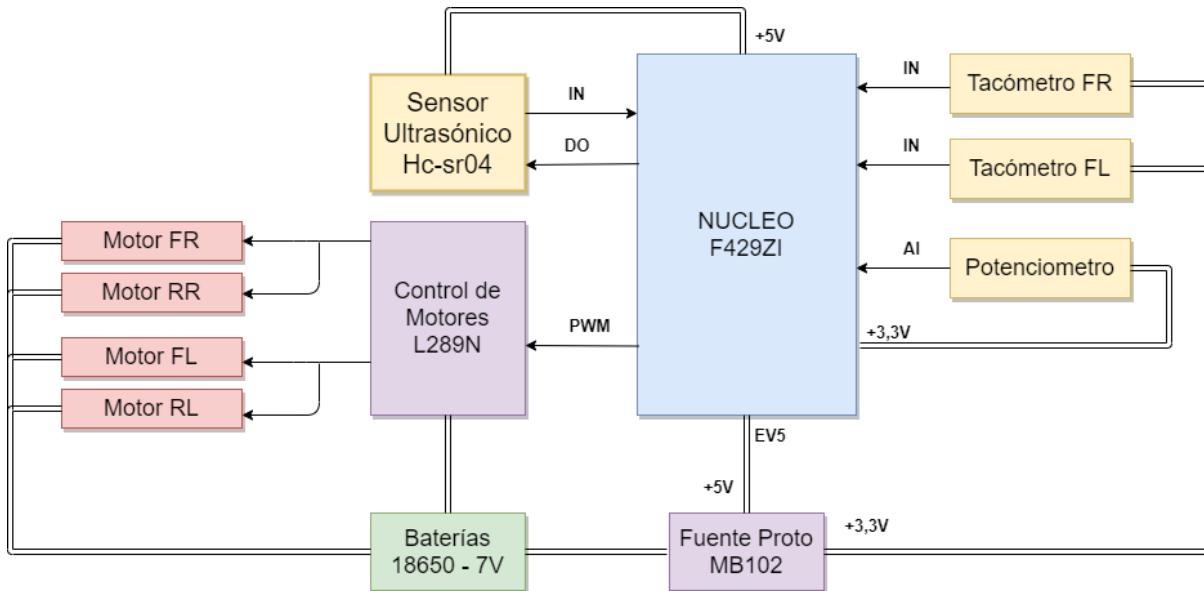


Figura 2. Diagrama de Bloques

Conexionado de Hardware

NUCLEO	L298N
PB4	ENA
PF2	IN1
PE3	IN2
PE5	IN3
PE6	IN4
PA0	ENB

NUCLEO	L298N
FR, RR (+)	OUT1
FR, RR (-)	OUT2
FL, RL (+)	OUT3
FL, RL (-)	OUT4

NUCLEO	Hc-sr04
A3	Trig
A4	Echo

Cuadro 1. Conexiones L298N, Hc-sr04 y NUCLEO.

NUCLEO	LM393 FR
A1	D0

NUCLEO	LM393 FL
A2	D0

NUCLEO	Pote
A0	V _{out}

Cuadro 2. Conexiones Tacómetros, Potenciómetro y NUCLEO.

Modelado de Software

El control del vehículo se centra en el módulo mobile, el cual se interconecta con todos los demás módulos. A continuación el diagrama y la descripción global de uno de ellos:

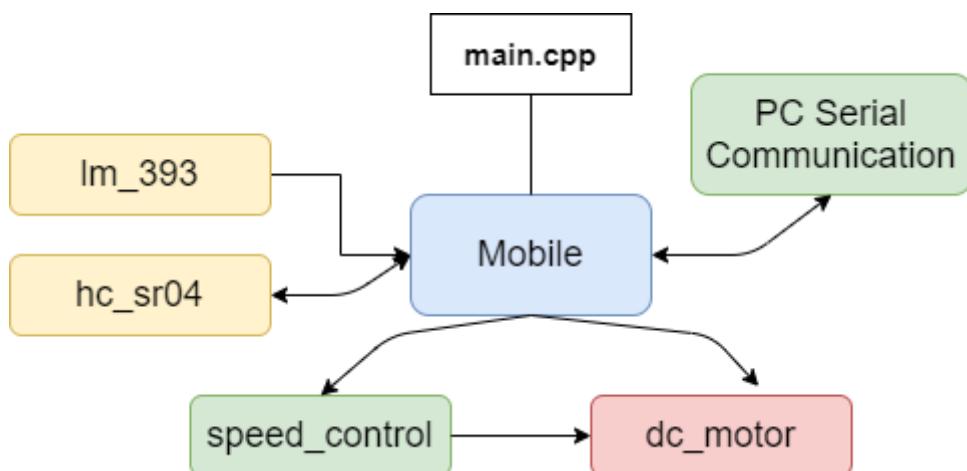


Figura 3. Diagrama de Software

Módulo	Descripción de la Funcionalidad	Rol
mobile	Controla las funcionalidades de los subsistemas y el tipo de manejo	Sistema
hc_sr04	Lee las mediciones del sensor ultrasónico y calcula la distancia	Driver
lm_393	Lee las mediciones del sensor óptico y calcula la velocidad	Driver
pc_serial_com	Controla la comunicación con la PC	Subsistema
dc_motor	Controla el estado de cada motor	Driver
speed_control	Lee las mediciones del set point y controla el PWM	Driver

Cuadro 3. Funcionalidades de los módulos.

El archivo mobile.cpp se estructura de la siguiente manera:

- Primero que nada se instancian los sensores y motores disponibles mediante las clases definidas en cada uno de los módulos lm_393, hc_sr04 y dc_motor. Esto nos permite trabajar con la cantidad que queramos dándonos flexibilidad a la hora de agregar, por ejemplo, otro sensor de ultrasonido para la parte trasera del prototipo.
- Luego se inicializan los módulos a través de la función mobileSystemInit():
 - Comunicación Serie con la PC, la cual permite elegir el modo de conducción.
 - Medición de sensores, la cual se realiza a través de interrupciones de la clase InterruptIn, y temporización a través de la clase Timeout. La medición se lleva a cabo de manera continua independiente de la temporización principal del programa. La distancia se mide en *mm* y las velocidades en *rpm*.
 - Inicialización de los PWM y del estado de los motores instanciados.
- Luego se comienza a realizar la actualización del sistema a través de mobileSystemUpdate(), mediante una temporización de 100ms. El proceso de actualización puede verse en el diagrama ??.
 - Se actualiza la información recibida a través de la PC, en particular el método de manejo. De esta manera se puede modificar esta variable ciclo a ciclo.
 - Se leen las mediciones mediante los métodos de las clases de los sensores y se almacenan en variables.

- Se setea el modo de conducción del prototipo. Dentro de esta función se evalúa el método seleccionado a través de la comunicación serie y se utilizan las mediciones anteriores para la toma de decisiones.

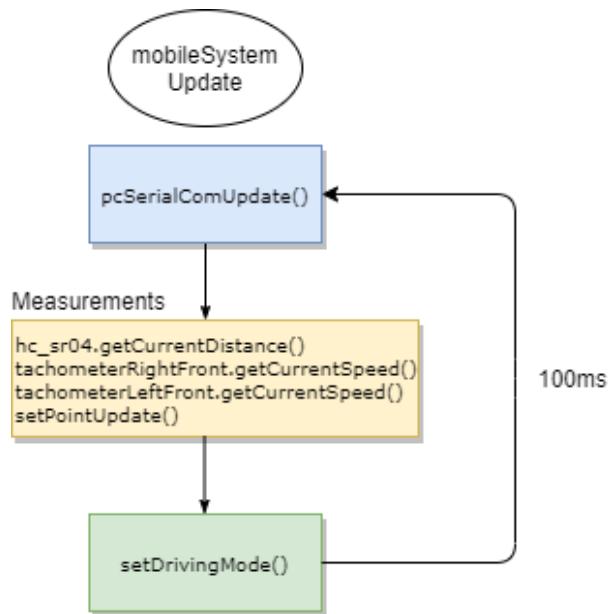


Figura 4. Diagrama de Flujo de `mobileSystemUpdate()`.

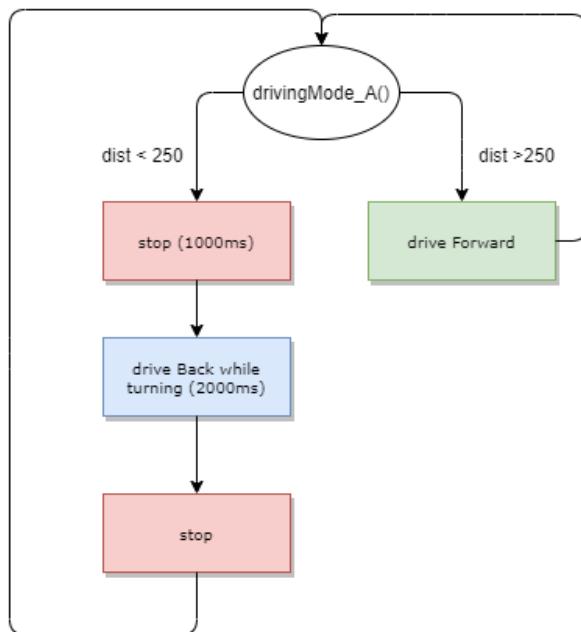


Figura 5. Diagrama de Flujo de `drivingMode_A()`.

Próximos Pasos

Se detallan a continuación los próximos pasos para mejorar la plataforma:

- Agregar place de balanceo y carga para las baterías.
- Implementación de PID para el control de velocidad.
- Agregar sensores ópticos en las ruedas traseras.
- Agregar otro controlador de motores para lograr un control individual.
- Relación entre velocidad y distancia para lograr mayor precisión en el frenado.
- Agregar display gráfico para visualizar el estado del sistema con el vehículo en funcionamiento.
- Desarrollo de funciones de conducción basadas en probabilidad.

Links

- Repositorio en Github: <https://github.com/AlfonsoRafel/mobile>
- Video en Drive: <https://drive.google.com/drive/folders/1wHx3dhAk0ciyH5XRI0-yYBtzN31v02P->