

INFORME PARCIAL-PRACTICO

Santiago Nomesque – Jair Munevar

Universidad Sergio Arboleda

¿QUE USAMOS?

- Microcontrolador STM32 con programador.
- Software STM32CubeMX y IDE.
- Base de Robot Micromouse.
- Motor DC con reductor montado en la base del Robot.
- Encoders de cuadratura en cada uno de los Motores DC.
- PCB con circuito para control de motores, CPU principal, 4 Sensores IR, encoders.

¿QUE SE REALIZÓ?

En este proyecto de Sistemas Embebidos, el objetivo fue diseñar y programar un sistema de control basado en un microcontrolador STM32 para un robot micromouse. La meta principal era que el robot mantuviera una distancia constante programable en mm frente a una pared de color blanco. El primer paso en el desarrollo del sistema consistió en la medición de la distancia mediante los sensores IR frontales del robot. A través de estos sensores, se obtuvo una curva de respuesta que relacionaba los valores de los datos analógicos (ADC) con la distancia medida en mm. Los datos se procesaron y se graficaron en Excel para obtener una tabla de valores, que permitió realizar una interpolación de los datos y definir una función que calculara la distancia real a partir de los valores de ADC. Esto fue esencial para que el sistema pudiera reconocer de manera precisa la proximidad al obstáculo y ajustar su posición en consecuencia. También se realizó una función que controlaba la velocidad del robot, conocida como Feedforward. Esta función tomaba como entrada la velocidad deseada en mm/s y devolvía un valor de modulación de ancho de pulso (PWM) adecuado para que el robot avanzara a la velocidad especificada. De esta manera, el sistema podía ajustar el movimiento del robot según la distancia que debía mantener respecto a la pared. Para ello, se probó la respuesta de los motores a diferentes voltajes para asegurar que el rango de velocidades se mantuviera dentro de valores consistentes y controlados, y así se obtuvo la función que relacionaba la velocidad con el voltaje y establecía un ciclo útil de PWM. Adicionalmente, se usaron los encoders para obtener la posición del robot con respecto a los pulsos, de manera que la posición no se determinaba únicamente con los sensores IR, sino también de acuerdo con el desplazamiento realizado, logrando así un mejor control.

¿QUÉ SUCEDIÓ?

El primer problema que surgió fue que los LEDs receptores y transmisores no funcionaron correctamente. Inicialmente se pensó que la causa podría estar en la PCB, pero al revisar el código se evidenció un error que provocaba que los LEDs se encendieran de forma desordenada y se mezclaran entre sí. Tras identificar este fallo, se corrigió el código asignando adecuadamente las parejas de LEDs, logrando así el funcionamiento esperado.

Para la siguiente fase, se decidió utilizar los LEDs frontales para realizar pruebas de medición de distancia. Sin embargo, estos no funcionaron correctamente, ya que por alguna razón no realizaban la medición de forma óptima. En comparación con la práctica de IR realizada, los sensores no alcanzaban dichos valores, se intentó poniendo una superficie blanca casi que al ras de los sensores y esto mejoro un poco la señal, pero aun así no se logró leer y calibrar de forma óptima. Posteriormente para verificar el funcionamiento de los motores, se observó que solo uno operaba entonces se procedió a revisar conexiones físicas. Después de descartar problemas en la PCB, se revisó el código y se encontró que el TIM encargado de generar el PWM de los motores no estaba inicializado. Este problema se solucionó rápidamente inicializando las variables correspondientes, logrando así un funcionamiento adecuado. Posteriormente se midió el diámetro de la rueda, se determinó que era de 35 mm y con este dato, fue posible calcular la longitud del arco de la rueda, obteniendo un valor de 109.95 mm, que correspondía a la distancia recorrida en una vuelta completa. Luego se procedió a usar los encoders, intentando primero que dieran una vuelta exacta para saber cuántos pulsos eran una vuelta, luego de varias pruebas se registraron 780 pulsos por vuelta, lo que permitió calcular que cada pulso representaba un desplazamiento de 0.141 mm. Una vez se obtuvo esto, ya se logró mover las ruedas de manera precisa y así tener un control preciso del desplazamiento de cada rueda. Con esta información, se desarrolló un código que permitiera a la rueda completar exactamente una vuelta. Tras implementar el código, se realizaron pruebas para verificar la precisión del sistema, confirmando que el desplazamiento de la rueda era exacto. Los resultados obtenidos fueron satisfactorios, lo que evidenció el correcto funcionamiento y la precisión del sistema de control.

DIAGRAMAS DE LUJO

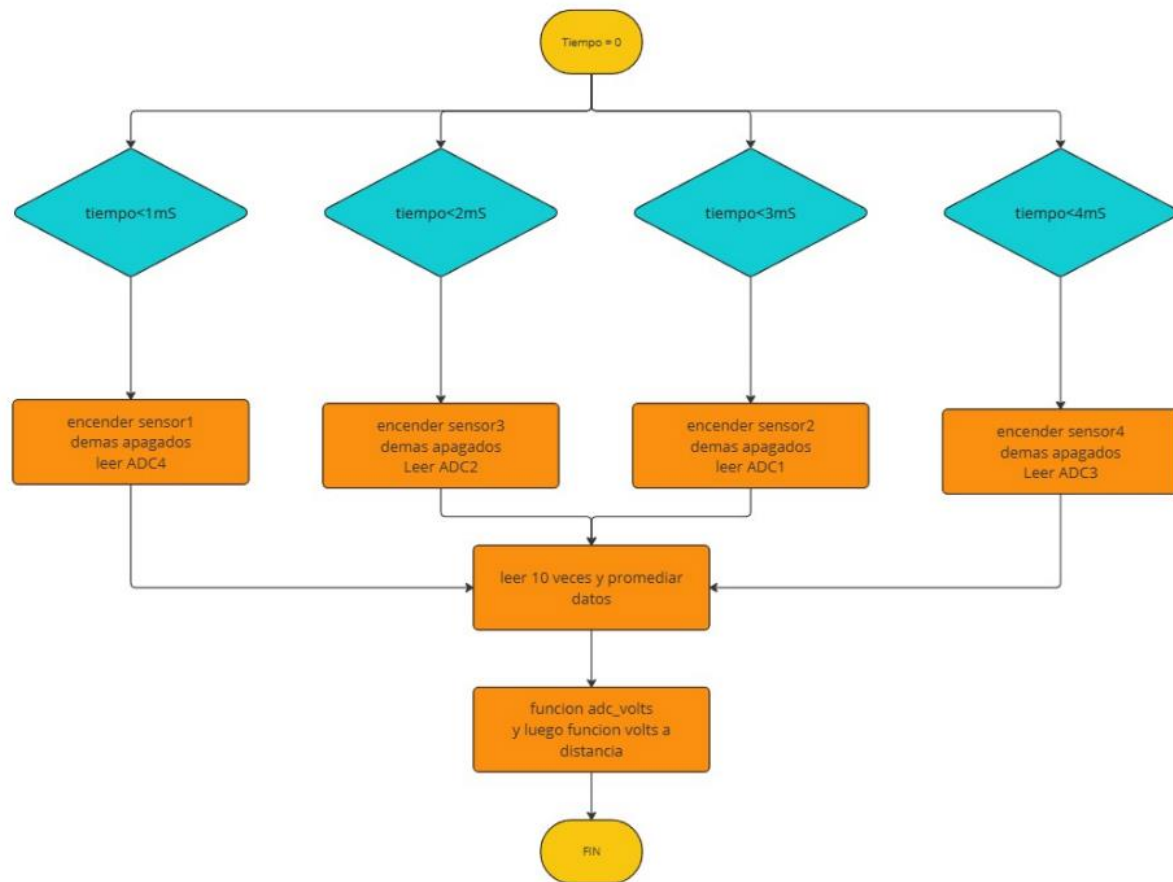


Imagen 1: Diagrama de flujo para el encendido de los diferentes sensores IR.

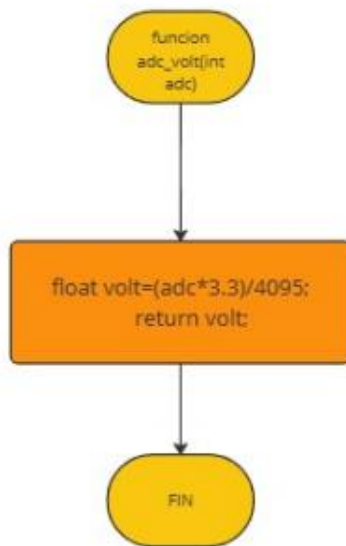


Imagen 2: Diagrama de flujo de la función para convertir ADC a voltaje.

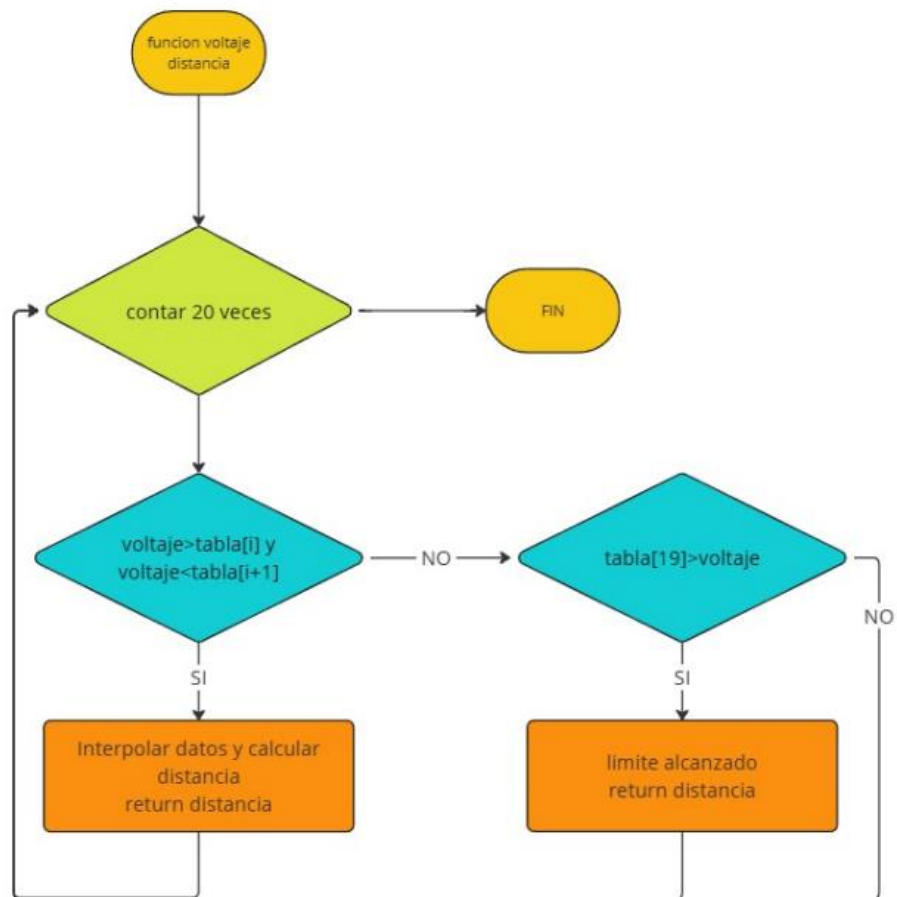


Imagen 3: Diagrama de flujo de la función para convertir voltaje a distancia.

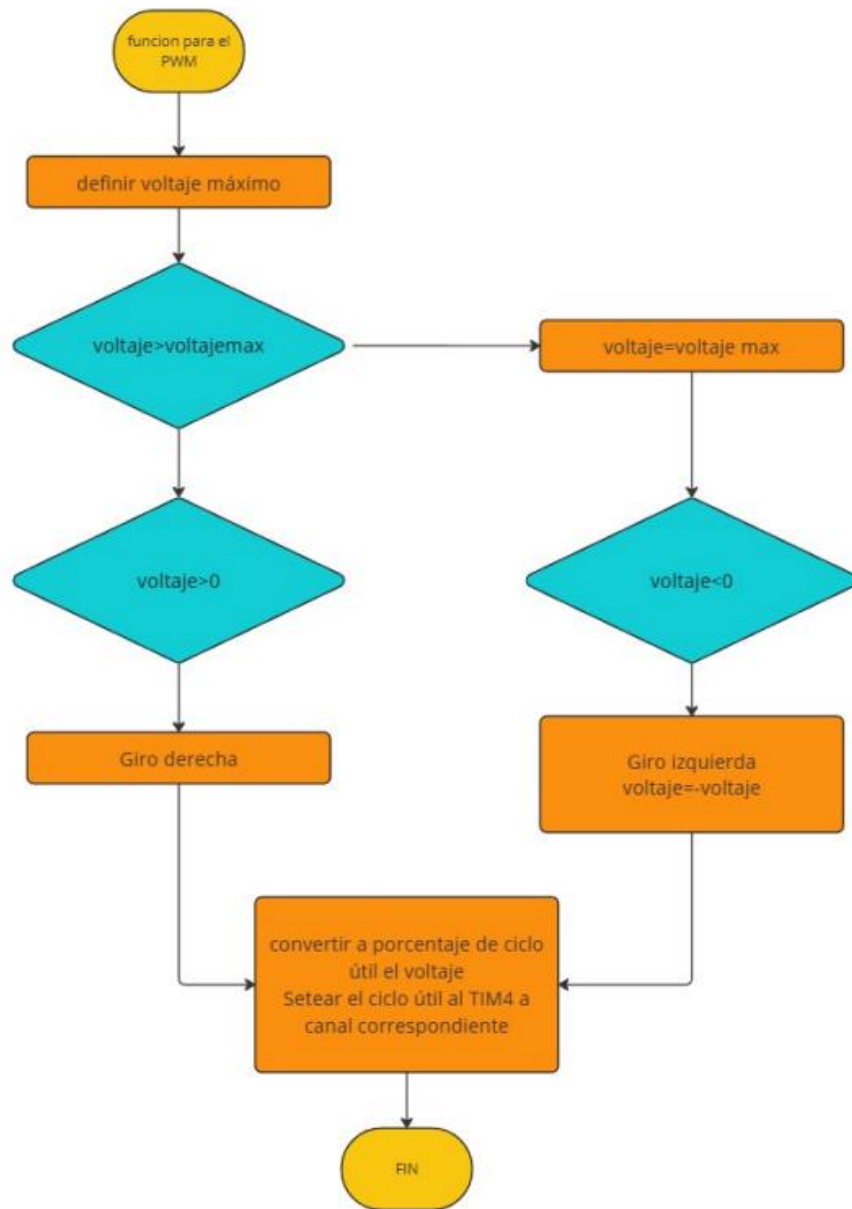


Imagen 4: Diagrama de flujo de la función del PWM.

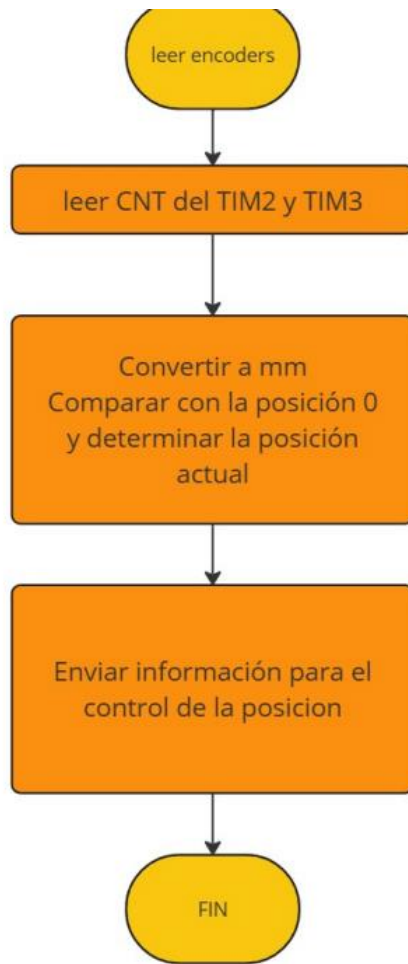


Imagen 5: Diagrama de flujo de la lectura de los encoders.

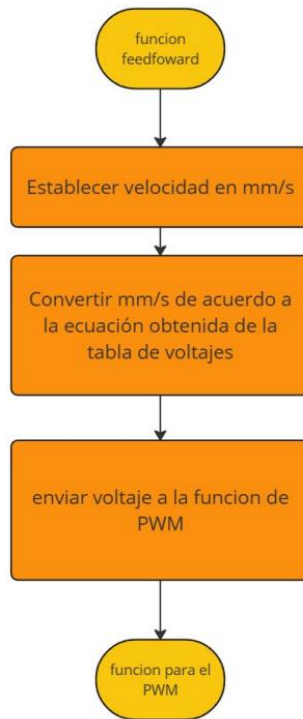


Imagen 6: Diagrama de flujo de la función feedforward.

SENSORES/DISTANCIA	S1	S4	VOLTSS1	VOLTS4
5	549	<u>1025</u>	0,44241758	0,82600733
10	337	<u>689</u>	0,27157509	0,5552381
15	281	<u>402</u>	0,22644689	0,32395604
20	171	<u>208</u>	0,1378022	0,16761905
25	139	<u>193</u>	0,11201465	0,15553114
30	128	<u>154</u>	0,10315018	0,12410256
35	117	<u>120</u>	0,09428571	0,0967033
40	100	<u>103</u>	0,08058608	0,08300366
45	97	<u>96</u>	0,0781685	0,07736264
50	94	<u>93</u>	0,07575092	0,07494505
55	90	<u>90</u>	0,07252747	0,07252747
60	88	<u>87</u>	0,07091575	0,07010989
65	87	<u>85</u>	0,07010989	0,06849817
70	86	<u>84</u>	0,06930403	0,06769231
75	84	<u>82</u>	0,06758499	0,06627154
80	81	<u>81</u>	0,06553432	0,06487639
85	79	<u>79</u>	0,06386884	0,06373834
90	78	<u>75</u>	0,06315928	0,06075185
95	76	<u>74</u>	0,06124542	0,05975478
100	74	<u>73</u>	0,0596337	0,05863142

Tabla 1: tabla distancia de los sensores y su voltaje correspondiente.

SENSOR 1

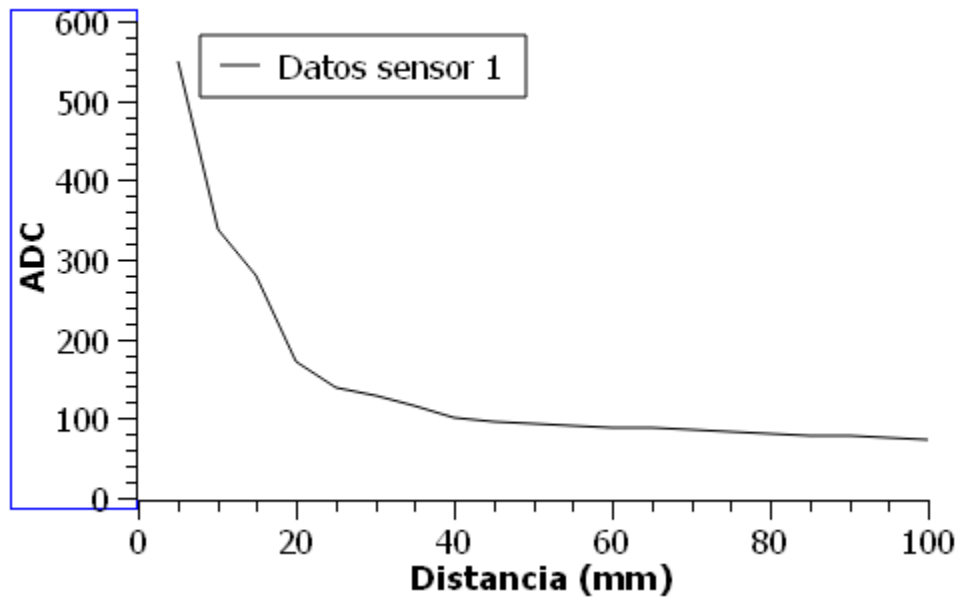


Imagen 7: Distancia vs ADC sensor1.

SENSOR 4

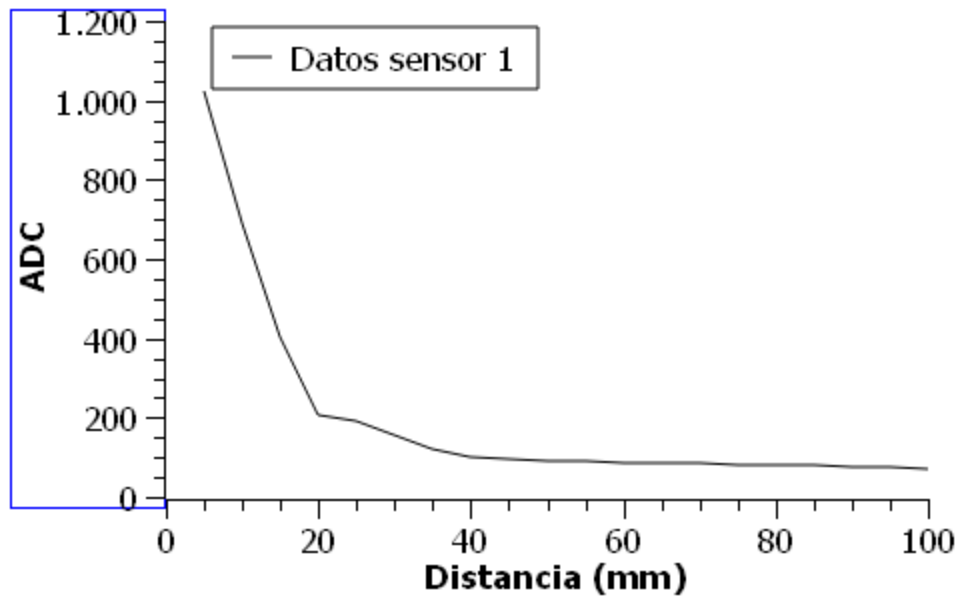


Imagen 8: Distancia vs ADC sensor4.

VOLTAJE	MOTOR1	MOTOR2
	mm/s	mm/s
1	26	29
2	98	104
3	174	181
4	238	244
5	279	299
6	331	341
7	380	399

Tabla 2: tabla de los mm/s con respecto al voltaje.

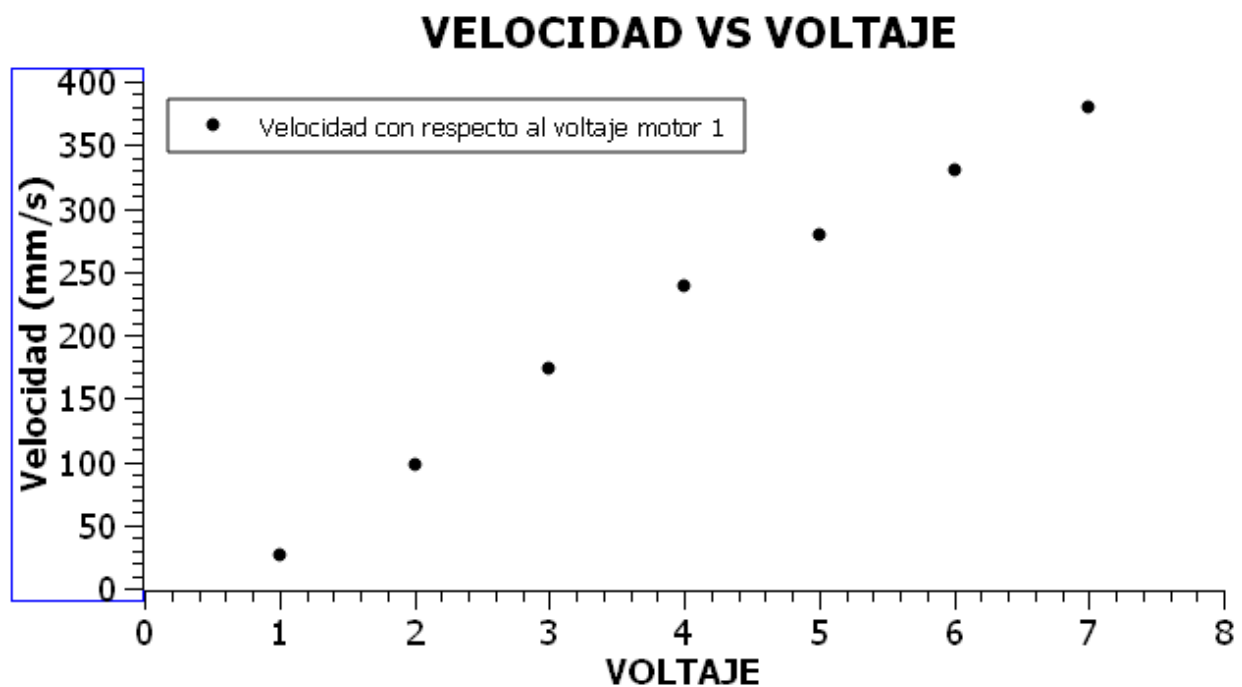


Imagen 9: Voltaje vs velocidad motor1.

Para este último, cabe aclarar que las pruebas se realizaron partiendo desde el voltaje de 7 voltios esto hizo que el motor a llegar a voltajes de 2 y menores, siguiera girando, pero si el estado inicial es apagado, los motores únicamente arrancan con un voltaje mayor o igual de 2,5 voltios

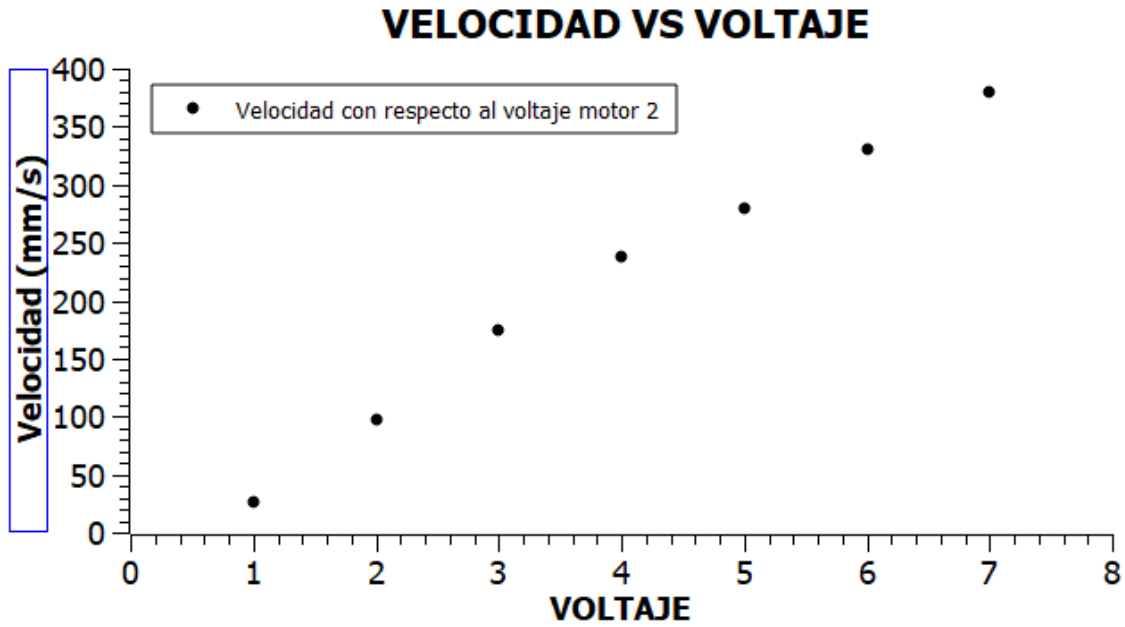


Imagen 10: Voltaje vs velocidad motor2.

ESQUEMÁTICO

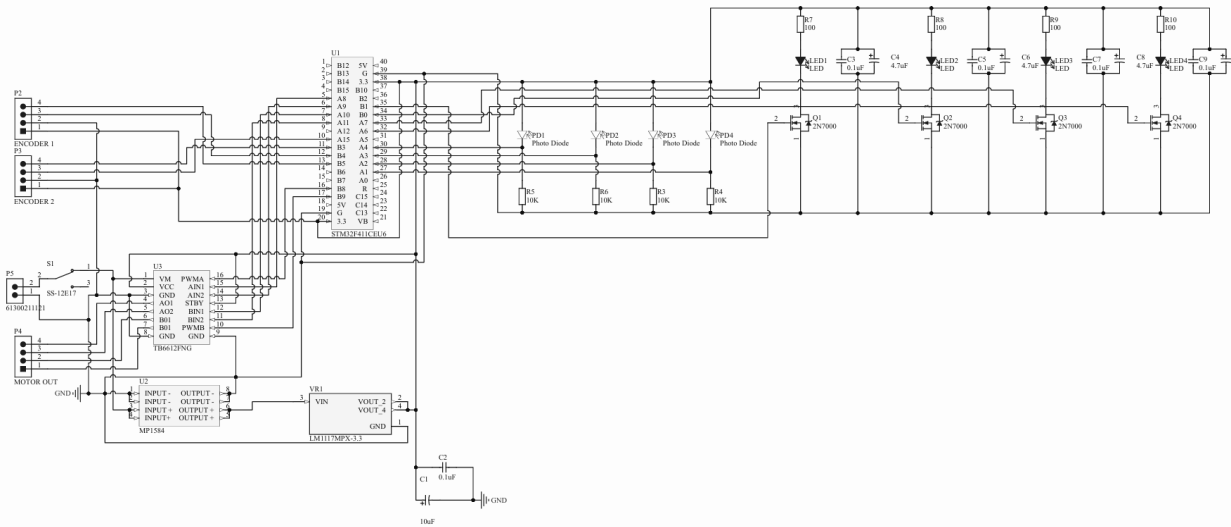


Imagen 11: esquemático micro mouse.

CONCLUSIONES

- La calibración precisa de los sensores IR es fundamental para obtener mediciones confiables de distancia. Aunque se presentaron dificultades para alcanzar valores óptimos, el ajuste de los sensores con superficies reflectantes mejoró la lectura, evidenciando que las condiciones de calibración afectan directamente el desempeño del sistema.
- La detección temprana de fallos, tanto en el código como en los componentes físicos, fue esencial para corregir problemas con los LEDs y el control de los motores. Esto demuestra que las revisiones periódicas de código y conexiones son cruciales en proyectos de sistemas embebidos.
- Medir el diámetro de la rueda y calcular su desplazamiento por pulso del encoder permitió un control exacto de la distancia recorrida. Este cálculo mejoró la precisión del sistema de desplazamiento y fue determinante para el éxito del control posicional del robot.
- La implementación de un control de velocidad tipo Feedforward con modulación de ancho de pulso (PWM) permitió ajustar la velocidad del robot con precisión. Esto resalta la relevancia de los métodos de control de velocidad en sistemas que requieren desplazamientos regulados y predecibles.
- La creación de funciones específicas para el control de distancia y velocidad facilitó el ajuste y funcionamiento del robot. Este enfoque modular ayuda a gestionar y actualizar el sistema de forma eficiente, permitiendo a su vez un mayor control sobre el desempeño de cada componente del robot.