

Mendoza - Argentina



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

PROTOTIPO DE ROBOT ASISTIDOR DE COSECHA

PROYECTO INTEGRADOR

Microcontroladores y Electrónica de Potencia
Profesor Titular: Ing. Eduardo E. Iriarte

Cantú Tsallis, Maximiliano – 11294

Año: 2022

RESUMEN

El objetivo principal del proyecto es lograr diseñar, simular y controlar el movimiento de un prototipo físico y a escala de un robot asistidor de cosecha, el cual sea capaz de seguir constantemente al operario durante el proceso, transportando el cajón de uva a una distancia y velocidad deseada. Esto es con el fin de evitarle al operario el trabajo de cargar, trasladar y descargar el cajón cada vez que necesite moverse, ya que esos movimientos suelen ser de distancias cortas y resulta agotador realizarlos repetidas veces en poco tiempo, generando un desgaste corporal y una pérdida de tiempo.

Para lograrlo, se desarrollará un modelo a escala con simplificaciones y diferencias respecto al proyecto final, las cuales se tendrán en cuenta como mejoras a futuro para completar la realización y fabricación del proyecto en su totalidad.

Índice

RESUMEN	1
INTRODUCCIÓN	3
DESARROLLO	5
1. Descripción de lógica y conexiones de hardware	5
2. Descripción de periféricos y código	10
3. Funcionamiento General	13
PUESTA EN MARCHA	15
ANÁLISIS DE INGENIERÍA	17
CONCLUSIONES	18
REFERENCIAS	19

INTRODUCCIÓN

El proyecto que se presenta a continuación tiene como objetivo la aplicación de los conocimientos adquiridos en la cátedra de Microcontroladores y Electrónica de Potencia de la carrera de Ingeniería en Mecatrónica en la Universidad Nacional de Cuyo.

La idea de realizar este proyecto surge de una experiencia propia vivida el año pasado, cuando tuve que trabajar haciendo temporada en la cosecha de uva de lunes a lunes todo el día. Mientras realizaba mi trabajo notaba que había acciones simples, pero repetitivas, y que con el correr de los días se tornaban cada vez más desgastantes por el cansancio y la fatiga que el cuerpo iba acumulando. Una de ellas era la de cargar y descargar el cajón de uvas a medida que se iba avanzando con la cosecha sobre la hilera de parcelas, lo cual era muy frecuente ya que como la distribución de los racimos no era homogénea, había sectores con gran cantidad de racimos para cosechar, lo cual equivale a mayor tiempo en una misma posición, pero habían otros sectores con muy pocos racimos en los que se perdía más tiempo trasladando el cajón, parcialmente lleno de uva, unos pocos metros para continuar el proceso. Es por esto que me surge la necesidad de pensar una forma para evitar dicho desgaste físico y pérdida de tiempo, totalmente desfavorable para el objetivo principal del trabajo, el cual era cosechar la mayor cantidad de uva de buena calidad y en el menor tiempo posible.

Como bien es sabido, en los pueblos o regiones en los que la actividad vitivinícola es uno de sus pilares económicos, la época de vendimia es una costumbre y una tradición compartida entre familiares, amigos y vecinos y de la cual todos forman parte en diferentes etapas del proceso, por lo tanto, no está bien visto el reemplazar completamente el proceso de cosecha por robots o máquinas automatizadas que lo realicen; y además, la correcta selección de la uva también es muy importante para obtener un buen producto final y de calidad. Por otro lado, también hay que tener en cuenta el factor económico, el cual también es muy importante, ya que no se justifica invertir grandes sumas de dinero por una máquina cosechadora si realizarlo manualmente es más barato y se obtiene un mejor producto final.

Teniendo en cuenta las limitaciones previamente mencionadas, es que pienso en los robots colaborativos, los cuales son destinados a trabajar en conjunto con un operario, pero sin quitarle el trabajo por completo al mismo. Logrando crear un robot que sea capaz de resolver el problema principal, el de cargar y trasladar los cajones repetidas veces en un periodo corto de tiempo, pero sin interferir en el proceso de selección y cosecha del fruto, sería lo ideal para lograr una cosecha más rápida y con una pequeña mejoría en cuanto al desgaste físico del operario; y sin necesidad de invertir grandes sumas de dinero en la compra de máquinas cosechadoras o vendimiadoras industriales, que rondan valores entre los US\$250.000 y los US\$350.000, aproximadamente.

La idea global del proyecto es implementar un robot diferencial de orugas, debido a que su aplicación sería principalmente en terrenos hostiles y dificultosos, no se requerirán altas velocidades de movimientos y si será necesaria una gran tracción en el terreno para trasladar cargas entre los 2 y los 20 kilogramos aproximadamente. El movimiento sería guiado por una tobillera colocada en el operario y con comunicación infrarroja con el autómat, de manera que pueda sensor y calcular la distancia a la cual se encuentra el operario, es decir la tobillera, y mediante una lógica determinar si es necesario moverse hacia adelante para seguir al operario o moverse hacia atrás para alejarse del mismo. Además, dependiendo de la dirección de la cual proviene la señal infrarroja, el autómat sería capaz de girar hacia un lado o hacia el otro, mediante la activación de uno de los motores únicamente; cuando el operario deba girar en una hilera o comenzar con una nueva. El robot contaría con un potenciómetro para que el operario pueda modificar manualmente y a su gusto la velocidad de movimiento hacia adelante y hacia atrás, según sea necesario; la velocidad de giro sería constante sin posibilidad de modificarse. Se opta por colocar un potenciómetro en lugar de panel táctil, teclado numérico o interfaz LCD, debido a que el ambiente también sería hostil y el robot debería ser lo más rudimentario y robusto posible, y también debido a que se debe tener en cuenta que el operario tendría guantes con suciedad colocados. Por último, el operario contaría con un pulsador

sin retención para setear el robot en un estado de reposo y que el mismo no lo siga hasta que se vuelva a presionar el mismo pulsador; esto sería útil en el caso que el operario desee alejarse del mismo para realizar alguna tarea temporal en la cual no requiera el seguimiento del robot.

Para el presente proyecto, se optaría por hacer algunas modificaciones a lo previamente mencionado debido a que la idea global completa sería considerada para presentarla como proyecto final de la carrera, implementándole conceptos y prácticas de diferentes materias. Se modificarán los sensores infrarrojos por sensores de luz tipo LDR, 3 para ser más específico, 1 sensor frontal, 1 sensor lateral izquierdo y 1 sensor lateral derecho. Las orugas se reemplazarán por 2 ruedas traseras conducidas por 2 motorreductores de CC controlados por 1 controlador L298N y 1 rueda "loca" frontal. El autómata sería alimentado por una batería externa recargable, la cual sea fácilmente extraíble y reemplazable si llega a ser necesario.

En este informe se expondrá la lógica de funcionamiento del robot, así como el código realizado y cargado en el microcontrolador, la implementación del prototipo, sus conexiones eléctricas y su montaje final con las partes mecánicas necesarias.

DESARROLLO

1. Descripción de lógica y conexiones de hardware

En el desarrollo del autómatas, primero se ideó la lógica de funcionamiento, la cual está dada en la Figura 1 y se tienen en cuenta los diferentes movimientos del robot en función de la intensidad y de la dirección de la luz incidente en los sensores. Se configuraron 4 valores constantes de luz para la comparación, los cuales generan una banda proporcional de velocidades de movimientos según la intensidad de luz recibida por el sensor frontal. Esta banda está dada por los valores constantes de $LUZ - MIN - MIN$, $LUZ - MIN$, $LUZ - MAX$ y $LUZ - MAX - MAX$, donde lecturas menores que $LUZ - MIN - MIN$ y $LUZ - MIN$ generarán un movimiento de avance a diferentes velocidades según el valor de dicha lectura; y lecturas mayores que $LUZ - MAX$ y $LUZ - MAX - MAX$ generarán un movimiento de retroceso a diferentes velocidades según el valor de dicha lectura. Estos movimientos deben ejecutarse hasta que el valor de la intensidad de luz se encuentre comprendido en los valores $LUZ - MIN$ y $LUZ - MAX$, los cuales determinarán la distancia a la cual el robot debe mantenerse del operario.

Para el giro, se realiza una comparación entre los valores de lectura de cada uno de los sensores laterales, con el del sensor frontal. En el momento en el que uno de los valores laterales sea mayor al frontal, el robot activa solo el motor correspondiente al lado contrario de la luz sensada, a una velocidad constante y fija, para generar un giro hacia el lado proveniente de la luz hasta que esta sea detectada por el sensor frontal, continuando con la lógica de avance y retroceso comentada anteriormente.

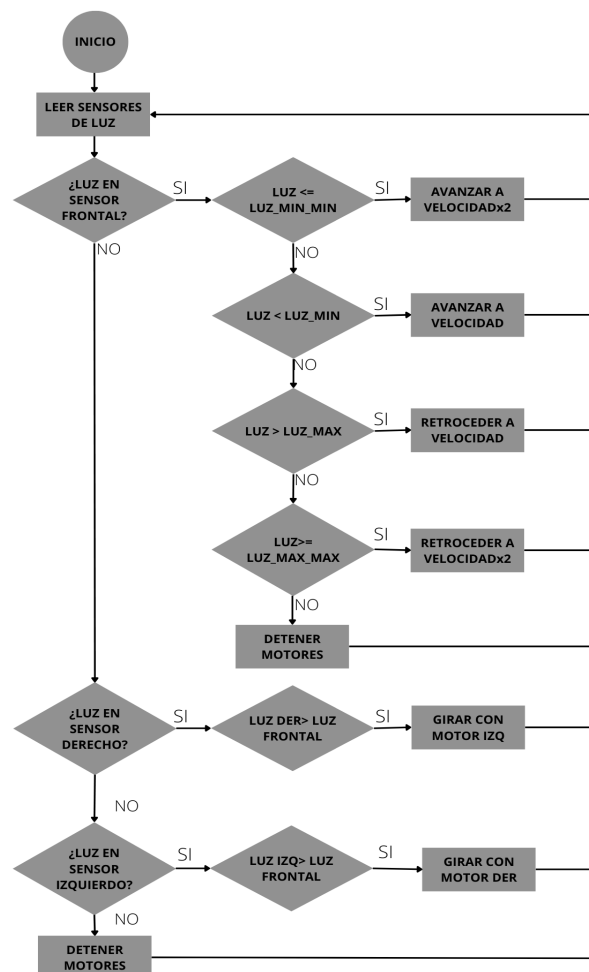


Figura 1: Diagrama de Flujo

Para implementar la lógica descrita anteriormente en un prototipo físico, se utilizará una placa Arduino UNO la cual contiene un microcontrolador ATmega 328P de la familia AVR, 1 controlador de motores L298N, 2 motores DC del tipo motorreductores acoplados a 1 rueda en cada uno, 3 sensores de luz LDR con una resistencia de pull-down en cada uno, 1 potenciómetro de 1kohm, 2 LED's y 1 pulsador sin retención. Este último componente se conecta con resistencia pull-up para evitar un falso flanco en alto si se rompe el pulsador y su aplicación en el autómata real sería colocado en la tobillera del operador, ya que es necesario que sea accesible de forma directa y remota, y no en el robot mismo; pero por razones prácticas, en el prototipo en cuestión se lo implementa en el mismo cuerpo del robot.

El correcto funcionamiento del autómata depende principalmente de los sensores de luz LDR (Resistencia Dependiente de Luz, por sus siglas en inglés) o fotorresistencias, los cuales cuentan con una resistencia cuyo valor varía en función de la luz que estén recibiendo. Es muy pequeña cuando recibe una luz muy intensa (unos pocos ohms), y aumenta su valor a medida que disminuye la intensidad de la luz que recibe. Para realizar la conexión de cada uno de los sensores, se implementa un circuito divisor de tensión con una resistencia de 10kohm, que permite traducir el cambio de resistencia a un cambio de voltaje que sea posible leer en el pin análogo del Arduino, sin dañar el microcontrolador.



Figura 2: Arduino UNO



Figura 3: L298N



Figura 4: Motor DC con rueda

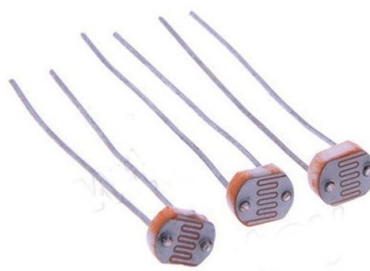


Figura 5: Sensores de luz LDR



Figura 6: Potenciometro



Figura 7: LED's



Figura 8: Pulsador sin retención

La conexión de los elementos electrónicos mencionados anteriormente, se pueden ver en la figura 9, en el cual se muestra la alimentación como una pila de 9V, pero en el prototipo se utilizan 2 pilas recargables de litio-ion de 3.7V y 7800mAh cada una, conectadas en serie, entregando una tensión de aproximadamente 8V cuando se encuentran completamente cargadas. Estas pilas se utilizan para alimentar el controlador L298N de los motores y por separado la placa arduino, utilizando un circuito de protección para la misma, compuesto por un diodo 1N4001 y un capacitor de 25V y 100uF.

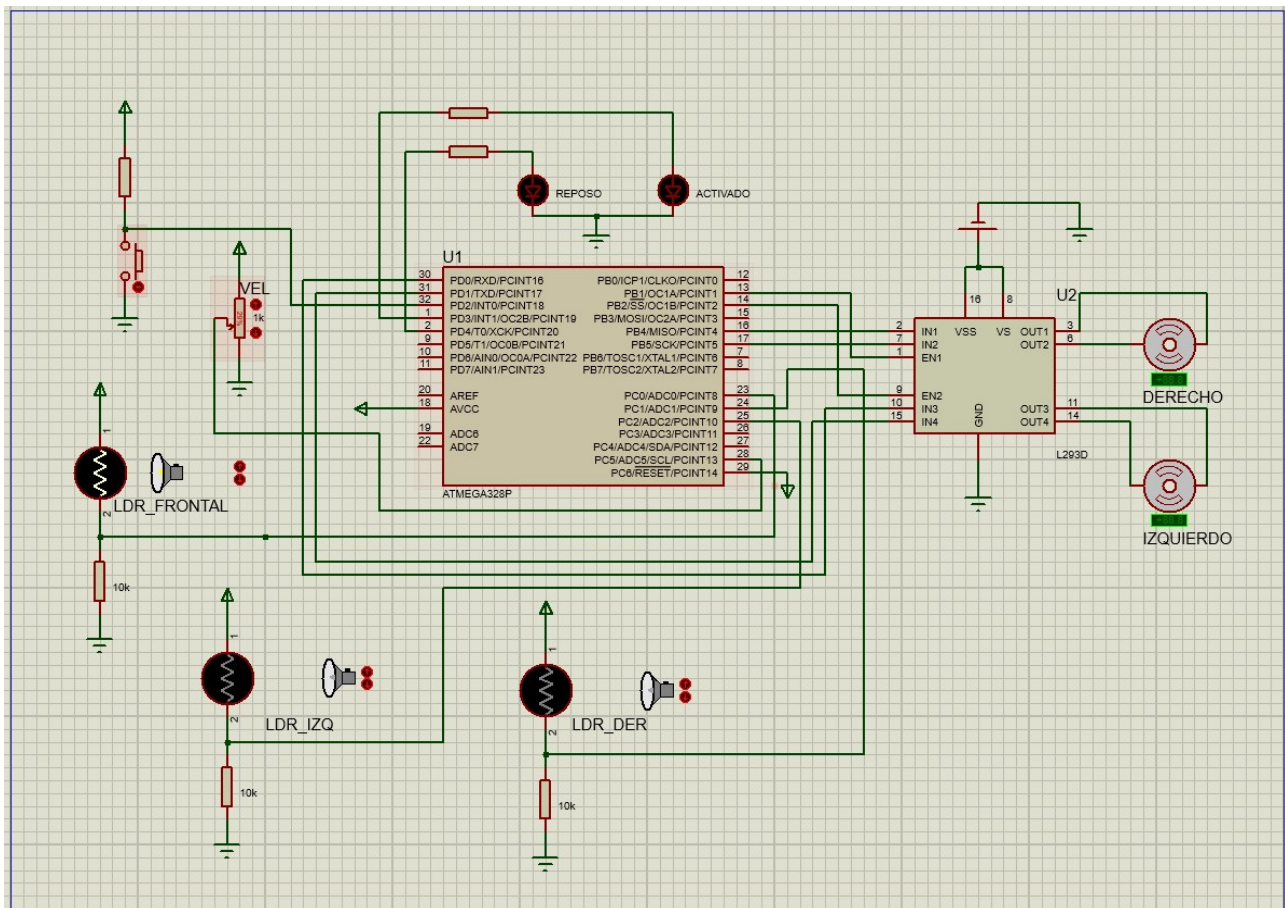


Figura 9: Diagrama de conexiones



Figura 10: Pilas 18650

2. Descripción de periféricos y código

En esta sección se describirán los periféricos utilizados del microcontrolador, según el esquema de entradas y salidas que se puede visualizar a continuación:

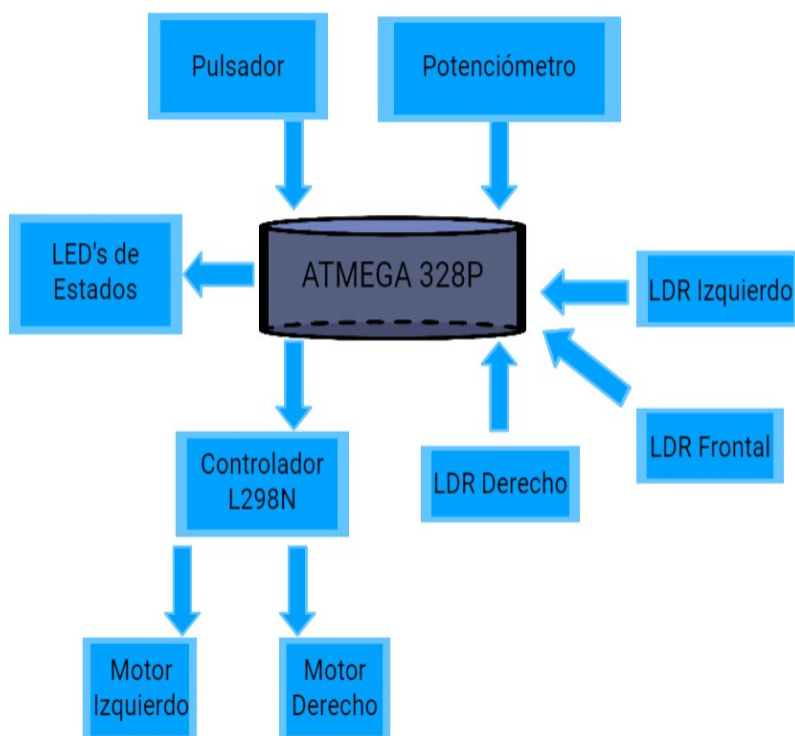


Figura 11: Esquema de E/S

Tal como se describió anteriormente, y como se puede ver en la figura 11, el microcontrolador realiza 1 lectura digital proveniente del pulsador de estado, 4 lecturas analógicas provenientes de los sensores de luz y del potenciómetro, y siguiendo la lógica explicada previamente, escribe señales digitales en el controlador de los 2 motores y en los 2 led's de estado.

Por un lado, la lectura digital del pulsador para el cambio de estado se realiza mediante una rutina de servicio de interrupción (ISR) en el bit INT0, configurada en modo CLEAR, en el cual solo se produce la interrupción al detectar un flanco de subida y permite cambiar el estado en el que se encuentra el autómata. En estado ACTIVADO, funciona según la lógica explicada previamente; y en estado de REPOSO, no responde ante ningún cambio de luz ni velocidad, es decir, se encuentra encendido pero detenido completamente.

Continuando con las lecturas, para realizar las lecturas analógicas, se utiliza su conversor analógico digital (ADC) de 10 bits. Primero se inicializa, se habilita para realizar la lectura y luego de cada una de estas, se deshabilita con el registro correspondiente para poder realizar correctamente la lectura siguiente, la cual proviene de un sensor diferente, es decir, un canal diferente. Para la utilización del ADC, previo a la inicialización, se lo configura para realizar conversiones a una frecuencia de 125kHz colocando su divisor de frecuencia en 128, para que dicho valor de frecuencia esté comprendido entre los 50kHz y 200kHz.

Los valores de los 3 sensores son utilizados tal cual son leídos, pero, el valor del potenciómetro con el cual se setea la velocidad de avance y retroceso, es necesario escalarlo ya que para valores bajos no se logra ningún efecto en la velocidad de los motores debido al par resistente que se debe vencer para que el robot se mueva. Para lograr determinar la escala a aplicar, se procedió a medir con un multímetro la tensión a la cual comenzaban a moverse, ya que para valores de tensión menores, no se movían y solo se escuchaba una especie de pitido proveniente de los mismos. Teniendo la lectura de dicha tensión y sabiendo que el potenciómetro es de 1kohm, es posible

calcular la resistencia para la cual los motores vencen el par resistente y así encontrar la fórmula 2.1 que permite escalar el valor leído, para que un valor bajo de lectura sea suficiente para mover los motores a una velocidad mínima.

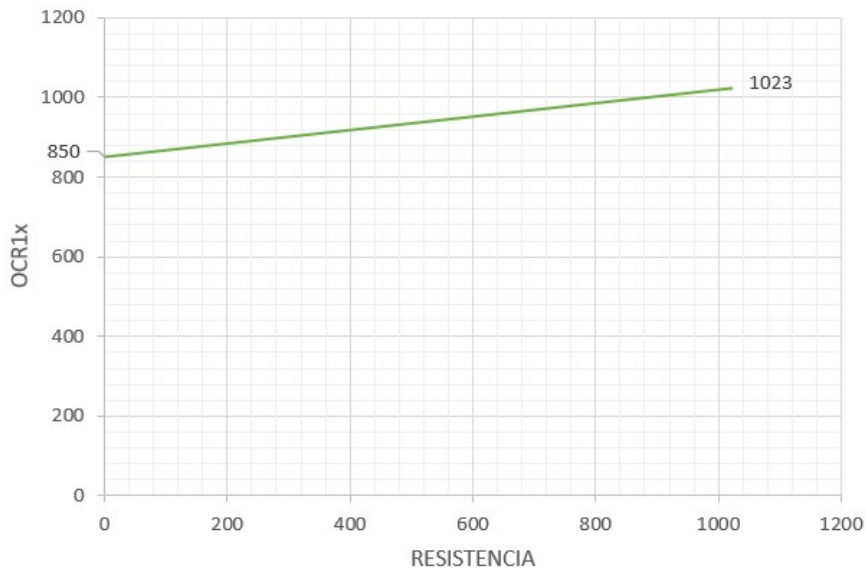


Figura 12: Resistencia potenciómetro vs. OCR1x

$$\text{OCR1x} = R_{\text{potenciómetro}} * 0.17 + 850 \quad (2.1)$$

Por otro lado, para modificar y escribir el valor de velocidad de cada uno de los motores, se utiliza el Timer1 en modo FAST PWM - 10 bits, donde el periodo estará dado por un valor fijo de 1023 y el ancho de pulso será modificado según el valor escalado leído por el potenciómetro y la lógica de movimiento. El timer se configurará en el modo indicado, con un prescaler de 1 y en modo CLEAR; y se utilizarán sus 2 registros OCR1x para generar la señal PWM de 15,6kHz aproximadamente, el OCR1A para el motor izquierdo y el OCR1B para el motor derecho. El valor de la frecuencia PWM mencionada surge de realizar el siguiente cálculo:

$$f_{\text{PWM}} = \frac{f_{\text{clk}}}{N * (1 + \text{TOP})} \quad (2.2)$$

Donde:

f_{PWM} : Frecuencia del PWM

f_{clk} : Frecuencia de clock del microcontrolador = 16MHz

N: Prescaler = 1

TOP: Valor máximo del contador del timer = 1023

Para obtener un valor de (2.2) de 15,6kHz se optaron por usar los valores de prescaler y TOP previamente mencionados, ya que simplifican los cálculos con el uso del potenciómetro y se obtiene un valor de frecuencia del PWM aceptable para los motores y el driver utilizados. Además, en la práctica el movimiento de ambos motores es de manera correcta.

En los movimientos de avance y retroceso, se escribirán los mismos valores en ambos registros, es decir, las ruedas girarán a la misma velocidad. Previo a la escritura de cada registro, se realizará una saturación de cada uno de ellos, para asegurarnos que los valores no superarán el máximo admitido correspondiente a 1023. En cambio, para realizar los giros, se escribirá solo uno de los registros, según el sentido de giro deseado, escribiendo un 0 en el otro.

Para finalizar, la escritura digital en los LED de estado, se realiza según el estado en el que se encuentra el autómata, modificado por el pulsador previamente explicado. Cuando se encuentra en REPOSO, se escribirá un 1 en el bit correspondiente al LED amarillo y un 0 en el bit correspondiente al LED verde; al cambiar al estado de ACTIVADO, se pone en 0 el LED amarillo y en 1 el verde.

3. Funcionamiento General

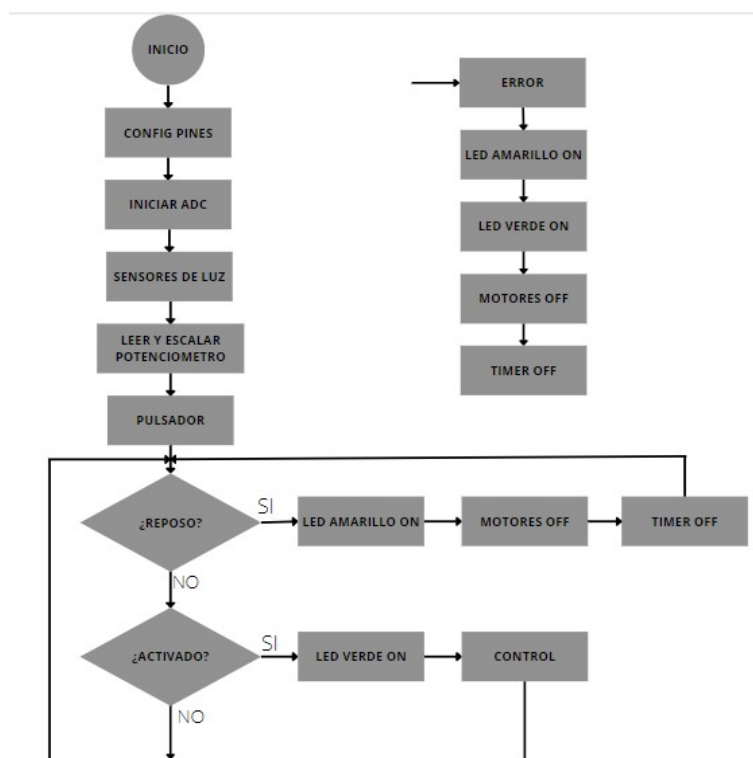
En esta sección se explicará la secuencia de funcionamiento que seguirá el robot, según la programación realizada para el mismo.

Primero se encenderá con un interruptor tipo ON-OFF que alimentará el circuito de la placa arduino y del controlador de los motores. Al iniciar el programa principal, se configuran los pines de salida de las señales PWM, la interrupción del pulsador de estado, se inicializa el conversor ADC, se setea el valor de la velocidad de giro y por último se habilitan las interrupciones globales. Luego de realizar todas las configuraciones previas, se indica que el robot se encuentra inicialmente en estado de REPOSO para que los valores de lecturas se realicen cuando se tenga lista la fuente de luz con la que se guiará al mismo.

Al encontrarse el usuario preparado, se procede a pulsar el pulsador para pasar al estado ACTIVADO y que comience el movimiento según la lógica planteada en la figura 13. El programa continua ejecutando cíclicamente las consignas que se encuentran dentro del bucle y dentro del estado ACTIVADO, es decir, comparando los valores de luz y asignando los valores del ancho de pulso a los registros OCR1X según la lectura en el potenciómetro.

Si se presiona nuevamente y en cualquier momento el pulsador de estado, el robot vuelve instantáneamente al estado de REPOSO, encendiendo el LED indicador correspondiente y apagando el timer que controla los motores.

Por último, el robot cuenta con un estado de ERROR, el cuál solo se presentará en el caso de que la lectura de los valores analógicos de los sensores LDR sean anómalos, es decir, superiores a 1023 o inferiores a 0. A este estado se lo considera como un estado de seguridad frente a fallas o comportamientos inesperados por parte de los sensores, del microcontrolador o cualquier otro factor que afecte la lectura real de la intensidad de luz. Al presentarse alguno de los casos mencionados, el robot se detendrá, se encenderán los 2 LED's indicadores (el verde y el amarillo, simultáneamente) y se mantendrá en ese estado hasta que se realice un reseteo por hardware, es decir, apagarlo y encenderlo nuevamente manualmente con el interruptor ON-OFF mencionado. Es importante tener en cuenta que si sucede una falla, previo a resetearlo por hardware, es prioritario revisar y solucionar el motivo de la falla, para que no vuelva a ocurrir.



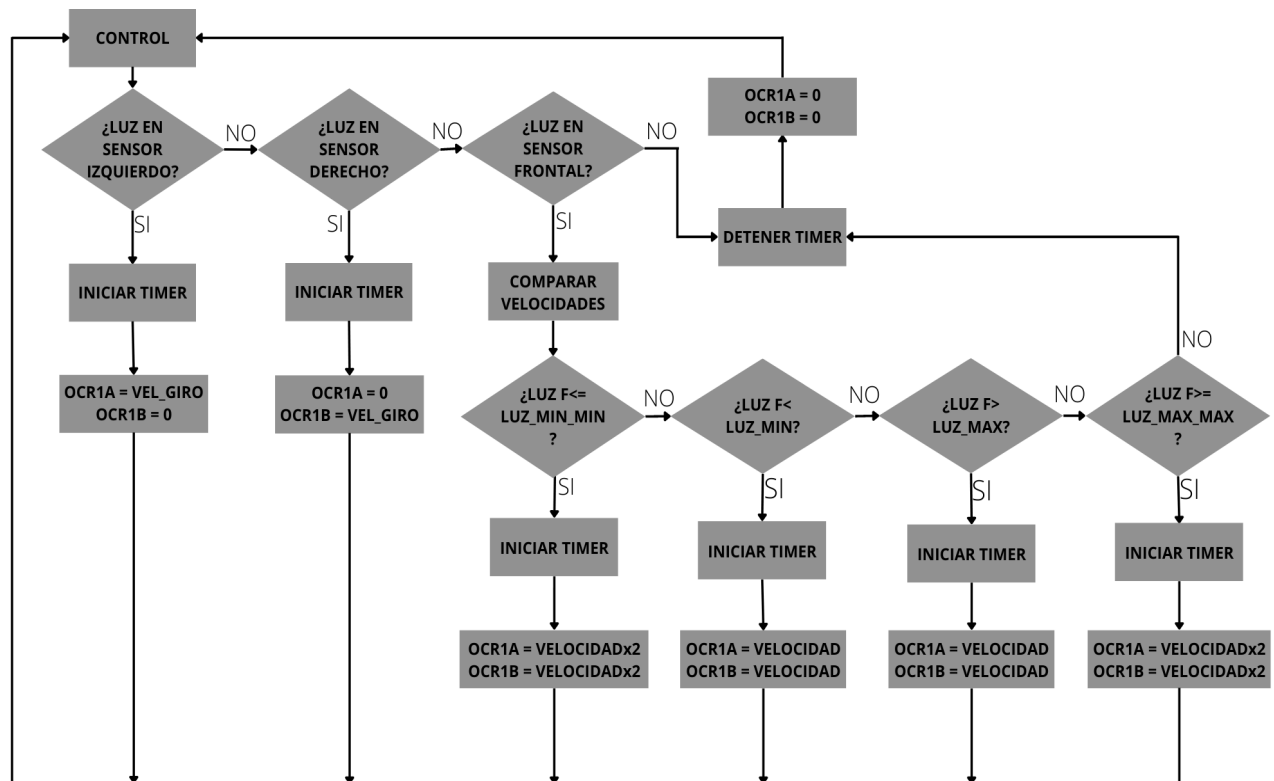


Figura 13: Diagrama de Flujo completo

PUESTA EN MARCHA

En esta sección se comentarán los resultados obtenidos luego de la puesta en marcha del robot, con una fuente de luz y modificando su velocidad y estados.

A continuación se puede ver una foto del prototipo final obtenido con la implementación de todo lo mencionado en este informe y sus conexiones tanto eléctricas como de control.

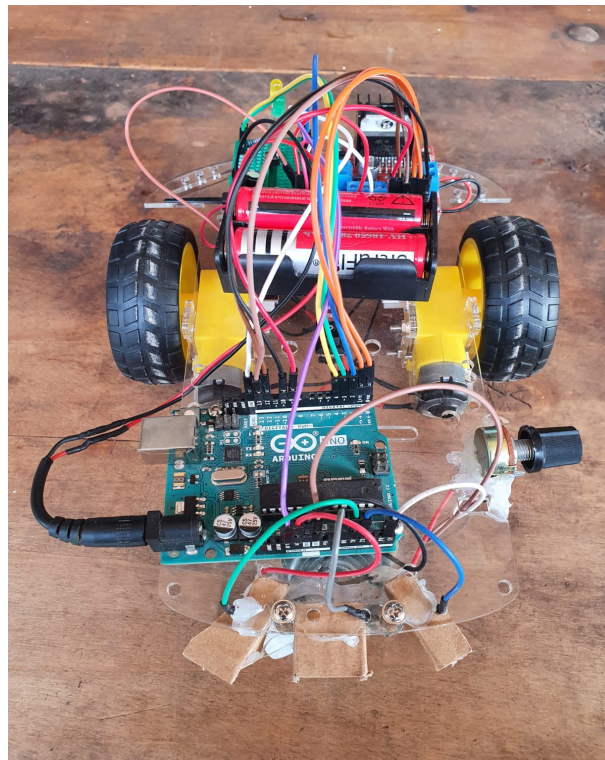


Figura 14: Prototipo Final 1

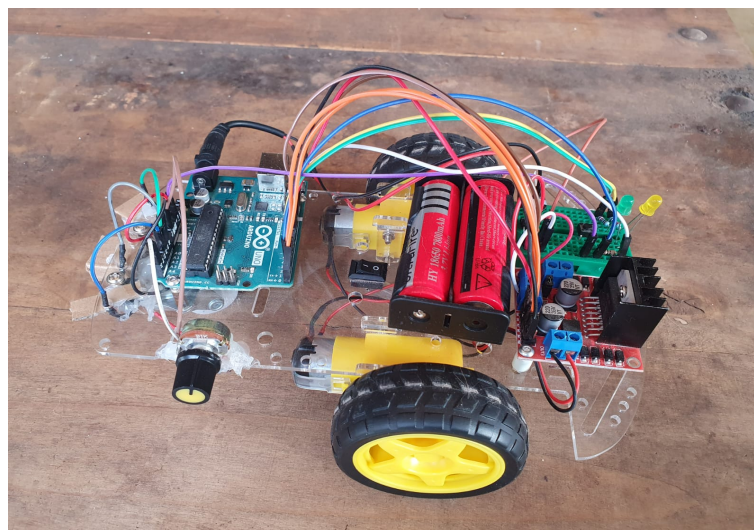


Figura 15: Prototipo Final 2

Las pruebas del funcionamiento real del prototipo de robot asistidor de cosecha se realizaron en una habitación prácticamente oscura, para que no hayan luces que confundan al mismo y modifiquen su funcionamiento, mas allá de que el mismo cuenta con una especie de ojeras en cada sensor, para

que se tenga ponga influencia de la luz ambiental en la lectura de los sensores LDR. Además, se utilizó una linterna de celular, colocado en un palo extensible con agarre de celulares en su extremo, típicamente conocido como "palo selfie".



Figura 16: Celular + Palo Selfie

Se procedió a encender el robot, colocar el celular con su linterna encendida en una posición enfrentada al mismo y luego se presionó el botón de cambio de estado para activarlo. Manteniendo el potenciómetro al mínimo, el robot lo primero que hizo fue moverse hacia atrás para cumplir con la consigna de mantener una distancia constante con el celular.

Al comenzar a mover el celular, agarrándolo mediante el palo mencionado, el robot continuó siguiéndolo según se esperaba, probando giros abiertos y cerrados para cada lado, avances alejando el celular y retrocesos acercándole el mismo. Con la intensidad de la luz que emite la linterna del celular utilizado, se verificó que el robot mantenía una distancia de entre 50cm y 100cm aproximadamente.

Por último, se comprobó que al modificar la velocidad de avance y retroceso con el potenciómetro correspondiente, se veían reflejadas dichas modificaciones en sus movimientos, comprobando que la lecturas analógicas se realizaban correctamente y que la respuesta de los motores era adecuada según la lógica de funcionamiento implementada en el microcontrolador.

Luego de varias pruebas, se verificó que en ninguna de las mismas se presentó una falla o un comportamiento anómalo en el robot y en sus lecturas, ya que nunca se habilitó el estado de ERROR, es decir, nunca se vieron los led verde y amarillo encendidos simultáneamente.

ANÁLISIS DE INGENIERÍA

En esta sección se plantean las mejoras necesarias que permitirán llevar este prototipo a un proyecto real, con un funcionamiento acorde a su implementación práctica.

- Modificar los sensores LDR por sensores infrarrojos que reciban una señal infrarroja proveniente de una tobillera colocada en el operario. En la misma se encontrará el botón de cambio de estado, que se comunicará con el robot y el microcontrolador mediante una señal de radio frecuencia, por lo tanto, en el robot habrá un módulo receptor de RF. Al realizar la modificación de los sensores, no será necesaria una habitación oscura para su funcionamiento, ya que será utilizado en terrenos exteriores, y el factor de la luz externa ya no será tenido en cuenta ni será influenciable.
- El tamaño final será bastante mayor ya que le debe caber una caja de cosecha de 500mm x 300mm x 270mm, aproximadamente.
- Las ruedas laterales y la rueda libre frontal serían reemplazadas por orugas con eje delantero y trasero, para poder desplazarse correctamente en terrenos hostiles y desnivelados.
- Se realizará un análisis dinámico que sería utilizado en la selección de los motores para que pueda soportar cargas extras de hasta 20kg aproximadamente, ya que es lo máximo que suele pesar un cajón de cosecha lleno de uva, además de su propio peso con sus baterías incluidas.

CONCLUSIONES

En este proyecto se lograron aplicar gran cantidad de los temas estudiados y practicados en la cátedra Microcontroladores y Electrónica de Potencia. Por un lado, se implementó el manejo y configuración de timers generando señales PWM, las cuales nos permitieron manejar los motores, modificando su velocidad mediante la modificación de su ancho de pulso. Además, se utilizaron interrupciones tipo INT y se implementó el uso del conversor analógico digital, leyendo valores analógicos de intensidad de luz y de resistencia variable de un potenciómetro.

Los conocimientos adquiridos en clases teóricas y prácticas, permitieron que se pudiera llevar a cabo este proyecto, junto a un trabajo de investigación y profundización en ciertos temas; aprendiendo a leer, interpretar y utilizar las hojas de datos de los fabricantes de los elementos utilizados.

Se logró materializar un prototipo del proyecto, pudiendo aprender sobre conexiones, alimentación y funcionamiento del mismo; ya que fue evidente que no es lo mismo trabajar con una simulación computacional que con un proyecto físico, en el cual las condiciones ideales no existen dado que hay una mayor probabilidad de error y de presencia de factores externos.

Al comprobar el funcionamiento del prototipo físico se pudo constatar que el mismo funciona de acuerdo a lo implementado en el código y en las conexiones, logrando un correcto movimiento guiado por una fuente de luz.

Tal como se mencionó al inicio del presente informe, este proyecto es un prototipo inicial de uno de mayor complejidad y tamaño, por lo tanto las mejoras a realizar serán apuntadas para implementar y obtener un proyecto final de acuerdo a la idea original.

REFERENCIAS

- <https://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>
- Instructivo Atmel Studio y AVR
- ATmega328P - 8-bit AVR Microcontroller with 32K Bytes In System - DATASHEET
- L298N Datasheet
- <https://www.agrofy.com.ar/vendimiadora>
- Videos, diapositivas y apuntes de clase