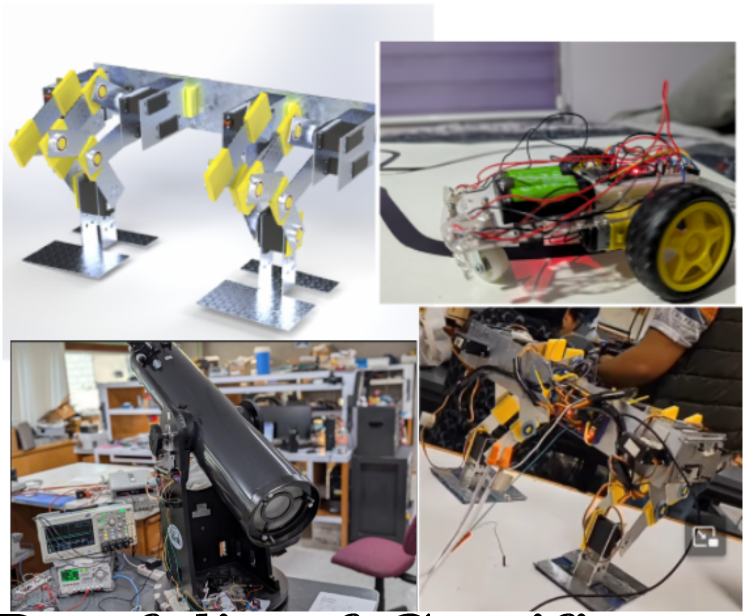
Student of:

**Autonomous University of Ciudad Juárez**

Institute of Engineering and Technology

Department of Electrical and Computer Engineering



# Portfolio of Certificates, Recognitions, and Projects of:

**Angel Hiram Alvidrez Hernandez**

---

**2024 - 2025**

# Índice


La Unidad Académica en Ensenada del Instituto de Astronomía y el Observatorio Astronómico Nacional San Pedro Mártir, Instituto de Astronomía, de la Universidad Nacional Autónoma de México, junto al Comité Organizador del

# XXXIV Verano Científico del Observatorio Astronómico Nacional San Pedro Mártir

extienden la siguiente

# *Constancia*

a

## Ángel Hiram Alvídrez Hernández

Por su valiosa participación como **estudiante** en el Programa del XXXIV Verano Científico OAN-SPM, atendiendo cursos, talleres y conferencias, y desarrollando el proyecto de investigación que **le fue asignado** en el período comprendido entre el 16 de junio y el 4 de julio de 2025.

INSTITUTO DE ASTRONOMIA
OBSERVATORIO ASTRONOMICO NACIONAL DE S. P. M
ENSENADA, B. C
JEFATURA

**Dr. Michael Richer**
Jefe de la Unidad Académica Ensenada
del Instituto de Astronomía,
Universidad Nacional Autónoma de México

**Dr. David Hiriart García**
Jefe del Observatorio Astronómico Nacional
San Pedro Mártir, Instituto de Astronomía,
Universidad Nacional Autónoma de México

**Dra. Lucía Adame Villanueva**
En representación del Comité Organizador
XXXIV Verano Científico OAN-SPM

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

INADET
INSTITUTO DE APOYO AL DESARROLLO TECNOLÓGICO

CENALTEC
CENTRO DE ENTRENAMIENTO EN ALTA TECNOLOGÍA

# SECRETARÍA DE EDUCACIÓN PÚBLICA
# SISTEMA EDUCATIVO NACIONAL
### DIRECCIÓN GENERAL DE CENTROS DE FORMACIÓN PARA EL TRABAJO

## INSTITUTO DE CAPACITACIÓN PARA EL TRABAJO
### ORGANISMO DESCENTRALIZADO

Unidad de capacitación: **CENTRO DE ENTRENAMIENTO EN ALTA TECNOLOGÍA** Con CCT: 08EIC0001B
**IAD-990630-MK7-0013**

OTORGA LA PRESENTE

# CONSTANCIA

ANGEL HIRAM ALVIDREZ HERNANDEZ

con Clave Única de Registro de Población: AIHA040515HCHLRNA2

En virtud de que acreditó los conocimientos, habilidades, destrezas y actitudes del curso

FUNDAMENTOS DE SEMICONDUCTORES

impartido en 90 horas, de conformidad al programa de **Extensión**.
De acuerdo a la información académica que obra en los archivos del Centro Educativo.

El presente se expide en JUÁREZ, CHIHUAHUA
A los DOCE días del mes de DICIEMBRE del dos mil VEINTICUATRO

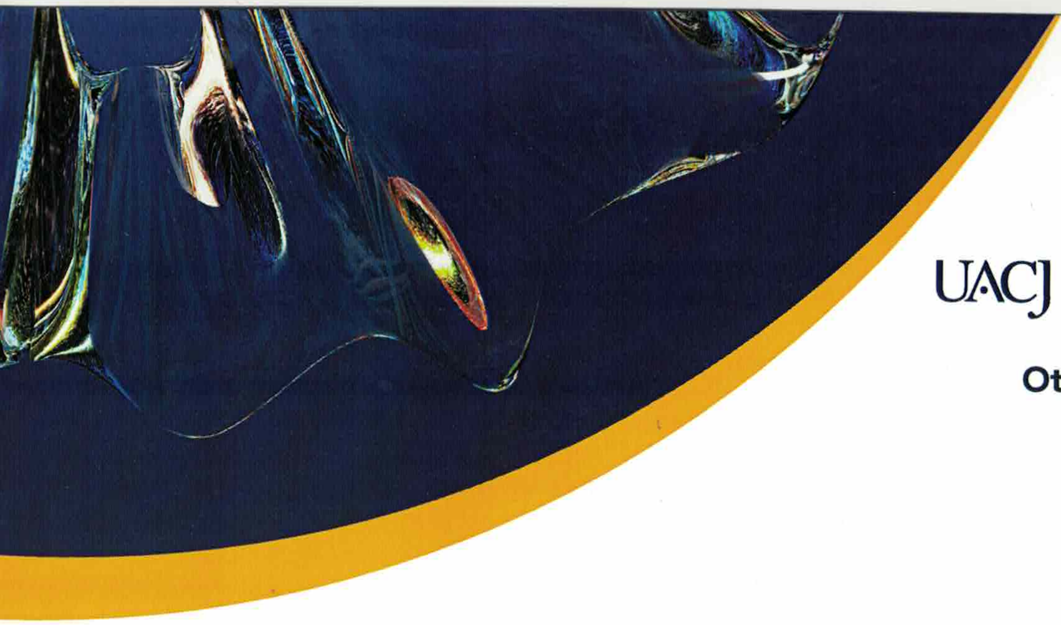
Folio D067705

DGCFT

BRENDA AIDEÉ RODRÍGUEZ PINALES
Director (a)

ESTADOS UNIDOS MEXICANOS
Sep Sems
D.G.C.F.T.
CENTRO DE ENTRENAMIENTO
EN ALTA TECNOLOGÍA
CCT. 08EIC0001B
STPS. IAD-990630-MK7-0013
SELLO

ESTA CONSTANCIA ES VÁLIDA EN LOS ESTADOS UNIDOS MEXICANOS Y NO REQUIERE TRÁMITES ADICIONALES DE LEGALIZACIÓN, NO DEBERÁ PRESENTAR BORRADURAS O ENMENDADURAS



UACJ | UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

**Otorga el presente**

# RECONOCIMIENTO

**A:** **Ángel Hiram Alvídrez Hernández**

del programa

**Ingeniería Mecatrónica**

**Por distinguirse como ESTUDIANTE INTEGRAL, con su excelente desempeño académico y por haber realizado actividades a favor de la comunidad durante el semestre enero-junio 2025.**

Ciudad Juárez, Chihuahua, noviembre 2025.
Por una vida científica, por una ciencia vital

Dr. Daniel Alberto Constandse Cortez
**Rector**

Dr. Salvador David Nava Martínez
**Secretario General**

Dr. Pedro Enrique Yáñez Camacho
**Director General de Bienestar Universitario**

DG BU DIRECCIÓN GENERAL DE **BIENESTAR UNIVERSITARIO UACJ**

UACJ INSPIRA EL FUTURO

52 ANIVERSARIO

UACJ | IIT

**La Universidad Autónoma de Ciudad Juárez**
**a través del Instituto de Ingeniería y Tecnología**

**Otorga el presente**

# RECONOCIMIENTO

**A**

## Ángel Hiram Alvídrez Hernández

**Por su participación como ponente de la Conferencia:**
**"¿Estamos solos en el universo?",**
**durante la XXX Semana de Ingeniería.**
**Septiembre 2024**

¨Por una vida científica,
Por una ciencia vital¨

**Dr. Juan Francisco Hernández Paz**
*Director del Instituto de Ingeniería y Tecnología*

UACJ | IIT

La Universidad Autónoma de Ciudad Juárez
a través del Instituto de Ingeniería y Tecnología

Otorga el presente

# RECONOCIMIENTO

## al Club de Astronomía

Por impartir la conferencia:
"Las muertes del universo".
Durante la XXIX Semana IIT

"Por una vida científica,
por una ciencia vital"

**Dr. Juan Francisco Hernández Paz**
Director del Instituto de Ingeniería y Tecnología

Autonomous University of Ciudad Juárez

Robotics Laboratory

Department of Electrical and Computer Engineering

# semester project

## Motors and controllers
## Edgar Alonso Martínez García

---

**Team Members:**

**Mechatronics Engineering:** Alan Quintana Quezada - 214621

**Mechatronics Engineering:** Angel Hiram Alvidrez Hernandez - 225581

**Mechatronics Engineering:** Viridiana Ahumada Velázquez - 225243

**Date:** November 21, 2025

# Index



# 1. Introduction

This project presents the development of an underactuated biped based on five-part mechanisms bars at each end, with a mechanical design modeled in CAD and an assembly perfected through successive stages of adjustment and improvement of the prototype aimed at achieving Controlled cyclical movements and initial walking tests. An Orange is also used. Pi as the main computer and an integrated IMU to record orientation and accelerations during static posture, cycloid trajectories and walking.

## 2. 5-bar linkage

Two five-bar mechanisms arranged in parallel were implemented, one at each end.
This is an underactuated mechanism with two degrees of freedom. The active joints
They are actuated by a pair of MG996R servomotors, while the joints
Passive motion was achieved using ball bearings. Two more servomotors were added per limb, in the connection to the hip and the connection to the robot's foot.

### 2.1. CAD Design

Figure 1 shows one of the five-bar mechanisms in isolation.
corresponding to the upper part of the limb. On the other hand, Figure 2 shows the five bars with a color code, described in Table 1, along with the lengths of each expressed in millimeters (mm). Once the CAD design of the mechanism is finished With 5 bars, the design of the entire bipedal robot, shown in Figure 3, was completed.

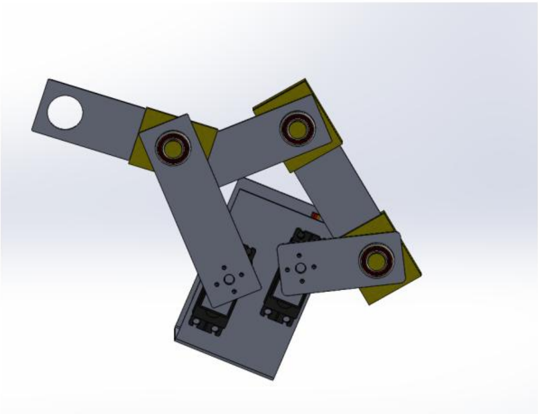


Figure 1: Design 1.

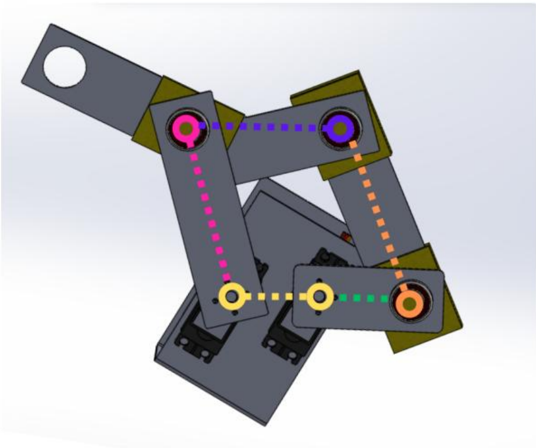


Figure 2: Definition of the 5 bars.

### 2.2. Kinematics of the mechanism

Once the five bars of the mechanism have been defined, the kinematic analysis is carried out with the objective to determine the position, in millimeters, of the upper tip of the mechanism from of the motor angles ÿ1 and ÿ4, as shown in Figure 4.

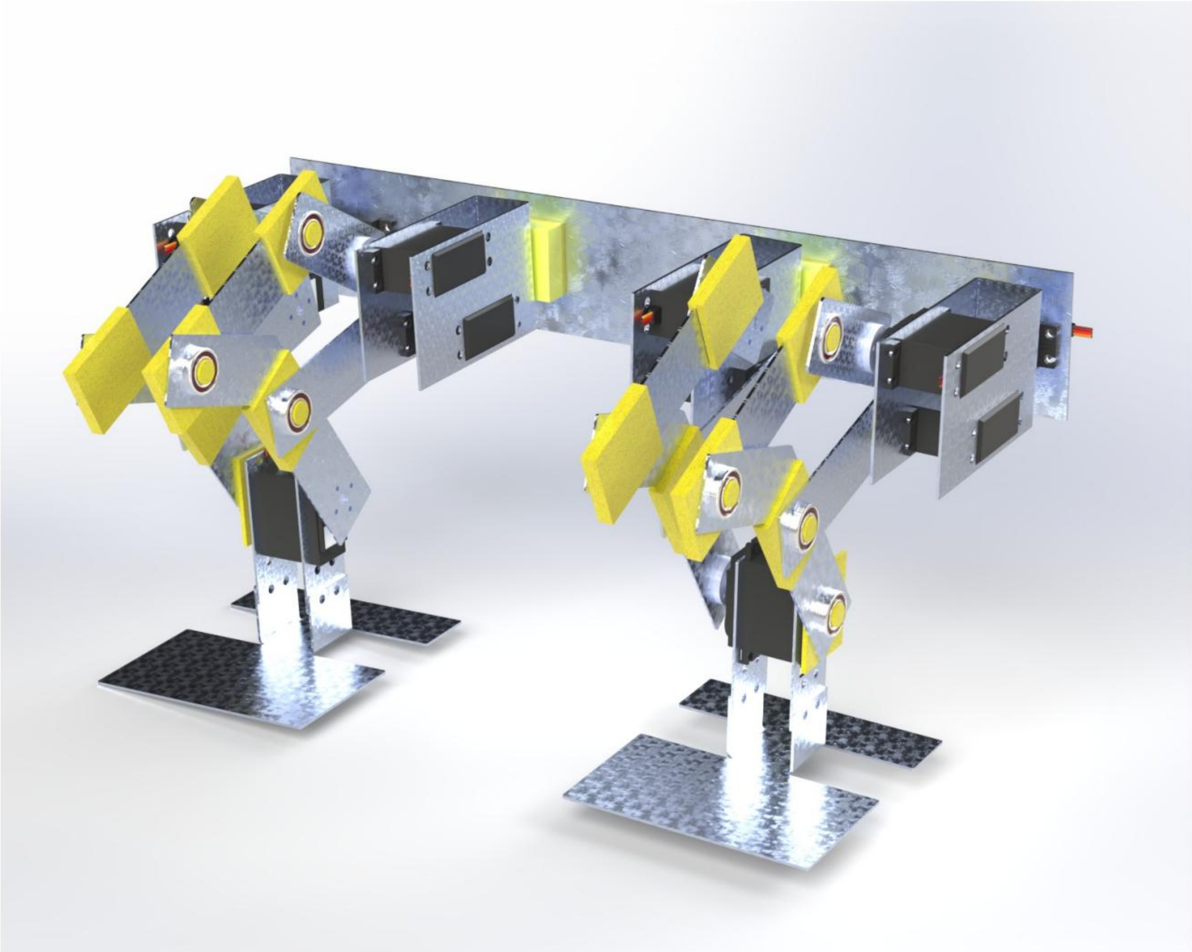


Figure 3: Design rendering.

Table 1: Table Title

| Bar Length | | Color |
|---|---|---|
| L0 | 40.84 mm | Yellow |
| L1 | 43.5 mm | Green |
| L2 | 88.67 mm | Orange |
| L3 | 67.07 mm | Purple |
| L4 | 83 mm | Pink |

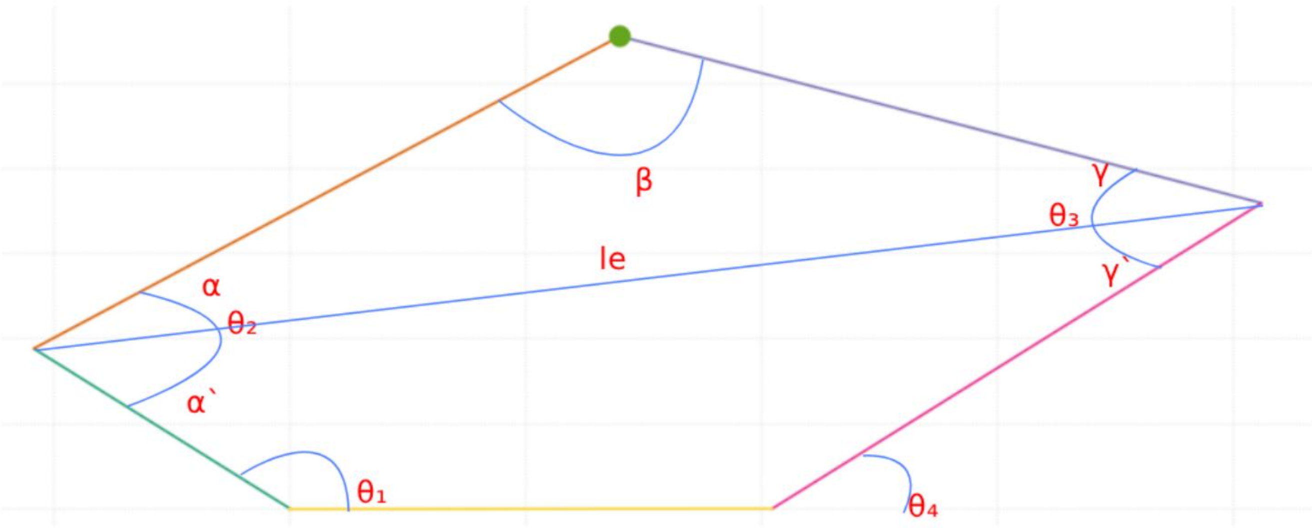



Figure 4: Visualization and nomenclature of the angles of the mechanism.

We begin by defining the position on the X and Y axes of the end of link L1 and L4 by the following expressions:

X1 = L1 cos(ÿ1); Y1 = L1 sin(ÿ1);

X4 = L4 cos(ÿ4); Y4 = L4 sin(ÿ4);

These positions are necessary to know the length of our "imaginary bar" Le, defined by:

Le = $\sqrt{(X1\ ÿX4)^2 + (Y1\ ÿY4)^2}$

This allows us to see if the combination of angles ÿ1 and ÿ4 is possible without breaking our mechanism, because if the sum of L2 and L3 is less than our newly calculated Le, then The motors would be unable to reach the indicated angles without destroying the mechanism or themselves same.

Continuing with the analysis, we can use the law of sines and cosines and rearrange it to find the interior angles ÿ, ÿ and ÿ with the expressions:

$$ÿ = \text{arc cos}\left(\frac{Le^2\ ÿL2^2\ ÿL3^2}{ÿ2\times L2\times L3}\right)$$

$$ÿ = \text{arcsin}\left(\frac{L3}{\text{You}}\times \sin(ÿ)\right)$$

$$ÿ = \text{arcsin}\left(\frac{L2}{\text{You}}\times \sin(ÿ)\right)$$

To continue, we need to define another triangle by drawing an imaginary line, which It goes from the top tip of L1 to the right tip of L0, as can be seen in the figure 5.

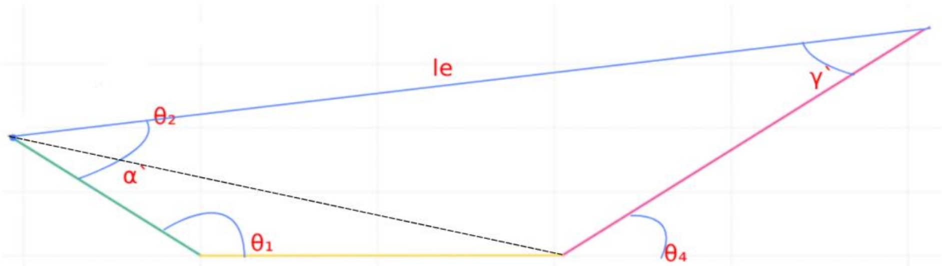



Figure 5: Visualization of L01 as the dotted line.

And using the law of cosines we can find the length of this imaginary line.

$$L01 = \sqrt{L0^2 + L1^2\ ÿ2L1L0\cos(ÿ1)}$$

with which we can calculate the local version of the angle ÿ using:

$$ÿ' = \text{arc cos}\left(\frac{L01^2\ ÿL4^2\ ÿLe^2}{ÿ2LeL4}\right)$$

which helps us to find the internal angle ÿ3 by adding ÿ with ÿ' and we will also need the complement of these angles. With which we can finally arrive at the positions Cartesian coordinates of the end effector of the 5-bar linkage with:

PX = L0 +L4 ×cos(ÿ4) +L3 ×cos(ÿ4 +ÿ3 ')

Py = L0 +L4 ×sin(ÿ4) +L3 ×sin(ÿ4 +ÿ3 ')

Once we know a condition that limits our mechanism, we can calculate all the possible locations for our mechanism and to have our workspace secure, shown in figure 6

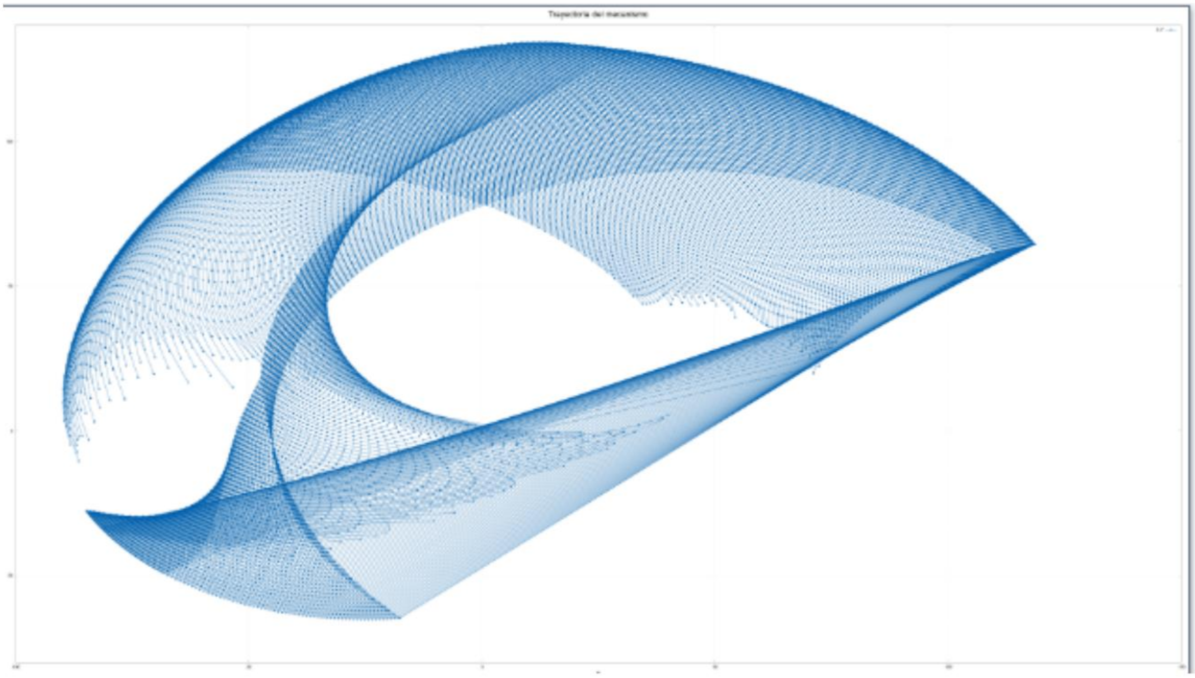


Figure 6: Workspace.

## 3. Cycloid

The cycloid is the shape of motion that the five-bar linkage follows to simulate a hike.

In order for the mechanism to follow the desired movement, the following equations were applied to function of time (t):

$$ÿ1 = a+c\times\sin(t) \tag{1}$$

$$ÿ4 = b+c\times\sin t + 2 \quad \frac{ÿ}{\phantom{x}} \tag{2}$$

where:

a : Initial angle of ÿ1,

b: Initial angle of ÿ4,

c: Oscillation amplitude.

## 3.1. Servo Movement

Once the joint movements were defined, the commands were sent to the servomotors. using an ESP-WROOM-32 and an expansion card with power supplies independent for control and power.

### 3.1.1. Master - Slave Communication

The computer embedded in the biped sends movement signals via serial port using the computer's USB ports and the control and sensing card.
A telegram is sent specifying which servomotor will be moved to what angle and with That's fast.

### 3.1.2. Angular velocity

For controlling the angular velocity of the servos, the initial choice was to use Libraries were used, but these did not provide good control of multiple servos simultaneously. To solve this This led to the development of an algorithm;

$$ÿt = ÿ0 + \sum_{k=1}^{N} ÿ \ \mathrm{sgn}(ÿi+N\ ÿÿi(k\ ÿ1))ÿÿi$$

Where ÿÿi are the angular intervals taken by the servo, calculated by:

$$ÿÿi = \frac{1}{veli}$$

The time intervals were calculated using a microcontroller time counter, instead of using interrupts, so that multiple servos can be controlled simultaneously. All of this was done to create a constant and smooth speed in the servomotors, by sending them to these angles by means of pulse width modulation, since it is the only data they can receive the servos.[1]

## 3.2. Generated cycloid

Once we understand the kinematics of our mechanism and have control of the actuators From this, the cycloid function was used and several iterations were performed in the parameters of equations 1 and 2 until a response is obtained that allows the robot to move maintaining its balance. The response of the end effector of the 5-bar linkage is shown in figure 7

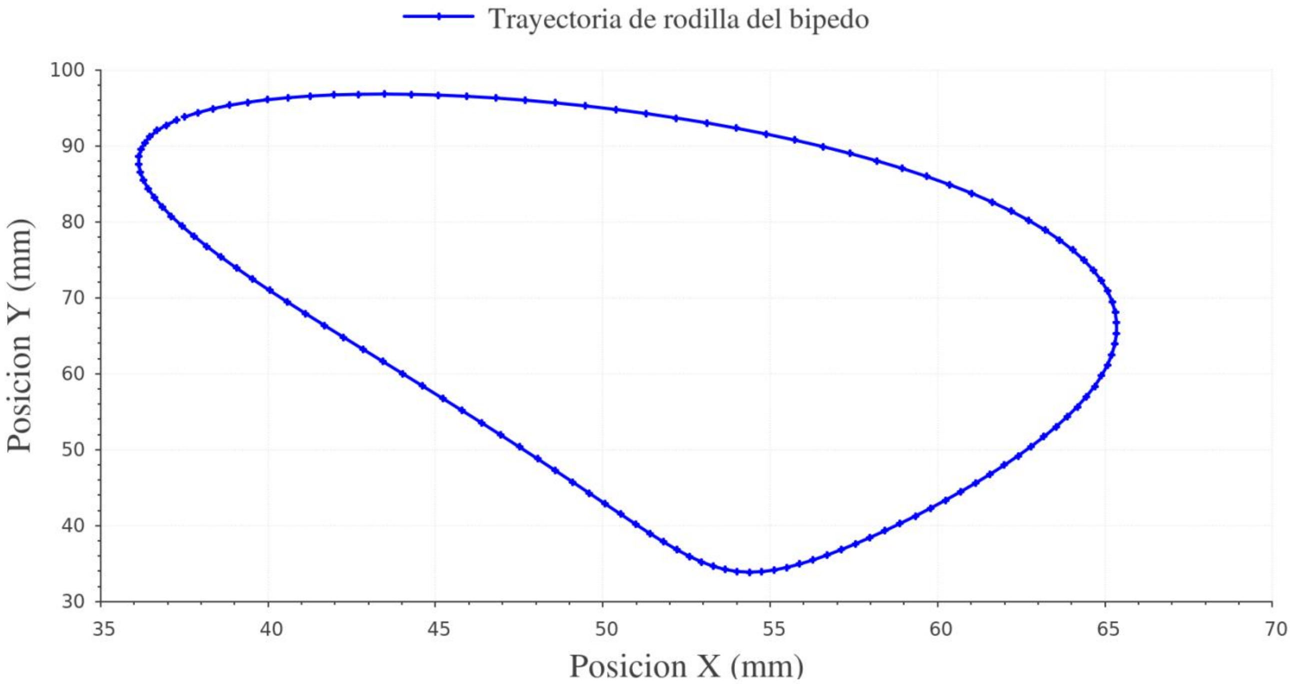



Figure 7: End effector of the 5-bar linkage.

## 4. Sensing

To prevent the Biped from falling, it needs to know that it is falling, so it is
I implement an IMU.
The selected IMU was the BNO80X, whose sensors are shown in Table 2

## 4.1. Data Fusion

The BNO08x is a System in Package (SiP) that includes a dedicated microcontroller and its its own sensors. The secret to its performance lies in the fact that the microcontroller executes the fusion algorithm within the chip, without the need for intervention from the main processor of the user. [2]

Table 2: IMU BNO08X

| SENSOR | MEASURES | MAIN PROBLEM |
|---|---|---|
| Accelerometer | Gravity and linear acceleration. | It is sensitive to vibration noise. |
| Gyroscope | Angular velocity (rotation). | He/She suffers from drift. |
| Magnetometer: | Measures the Earth's magnetic field (compass). | It is very sensitive to interference from metals. |

### 4.1.1. **Master - Slave** Communication

Once our IMU is defined, it will communicate with the ESP-32 using the I2C protocol, and It will be constantly reading data from the IMU. Until the Orange Pi sends it a send a telegram requesting the IMU data, then the ESP-32 will send the latest data that has been stored in memory. This communication algorithm is visually exemplified in the Figure 8.

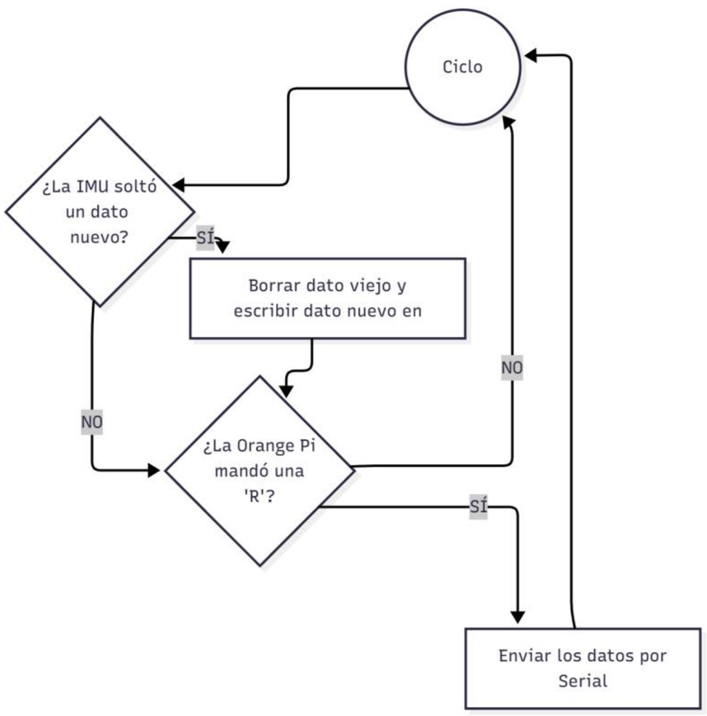



Figure 8: Slave ÿ Master communication sequence diagram.

# 5. Control Strategy

The robot actively monitors its tilt in real time to make safety decisions, alternating between a Walking mode and a Recovery mode.

The control algorithm implemented in C++ executes the following logical loop in each iteration, also shown in Figure 9:

1. **Data Acquisition (Feedback):** The computer sends the data acquisition telegram data to the sensing module. The IMU subsystem responds by sending the angle of Current pitch.

2. **Stability Assessment:** The algorithm checks if the robot is inside a safe operating range, experimentally defined as ±20° of inclination in the Pitch axis.

3. **Decision Making:**

   - **Stable Case** (|Pitch| < 20° ): If the pitch is safe, the system executes the next step of the cycloid trajectory. The coordinates (X,Y) are calculated, It solves the inverse kinematics and the servos move forward.

   - **Unstable Case (Fall Detected):** If the threshold is exceeded, the system is interrupted. temporarily stops the generation of the cycloid and activates a Safety Reflex:

     - If Pitch > 20° (Frontal Drop): The free limb moves rapidly forward to expand the support polygon.

     - If Pitch < ÿ20° (Backward Drop): The limb moves backward to counterbalance the fall.

4. **Resumption of Movement:** The system maintains the safety position until the IMU readings return to the ±20° range . Once stabilized, the algorithm resumes the execution of the cycloid walk from the point where it was interrupted.

This approach allows for a predictable cyclical walk, delegating stability to the design. mechanical and passive filtering of disturbances.

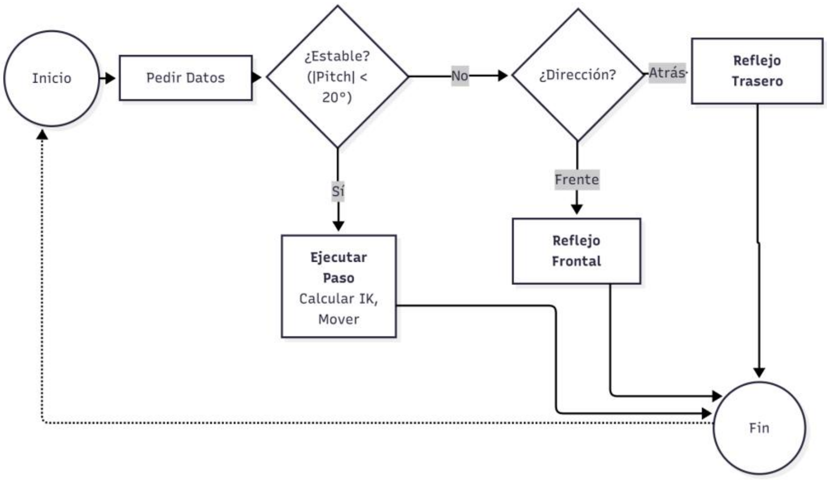



Figure 9: Walk sequence diagram

With this control architecture, even though it is somewhat "primitive", he managed to take a walk in the that the biped remained stable without needing support, thus managing to walk alone. [3].
While the walk was taking place, the IMU data was stored for later use. graphed, which can be seen in figure 10

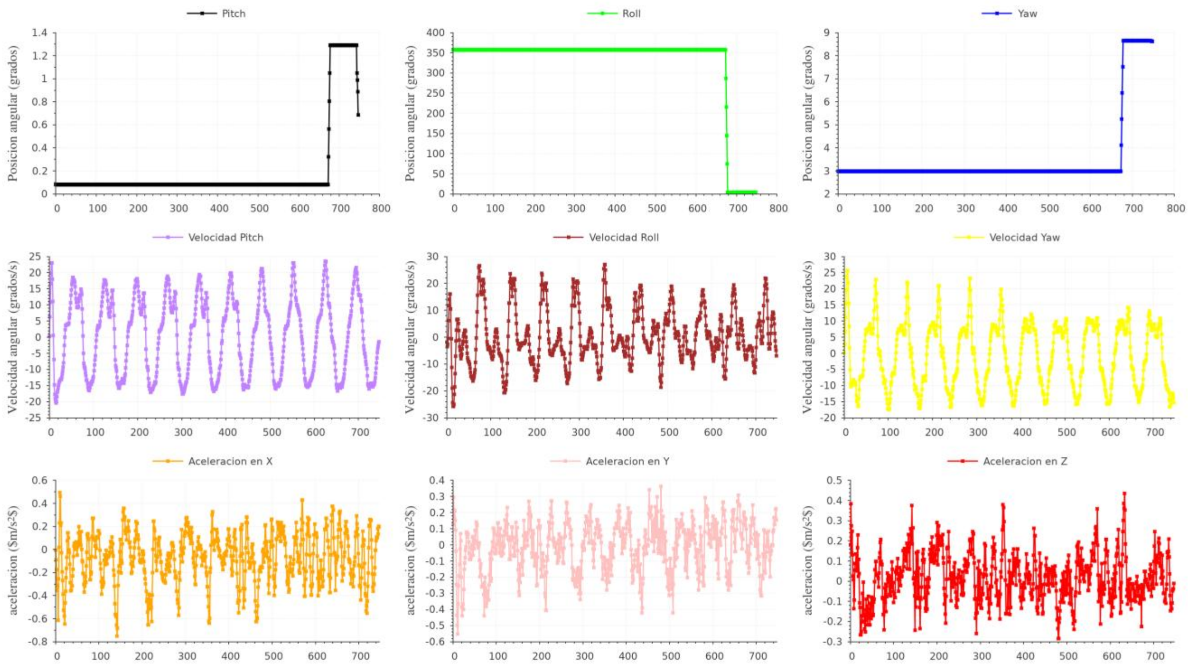



Figure 10: Walk sequence diagram

# 6. Conclusions

- Conclusion 1: The project to be carried out was the development of a bipedal walking robot, implementing a five-bar architecture for system kinematics, but
We had problems with the structure of the biped due to its dimensions and weights.
It was able to withstand it; the longest it took us to work on the biped was on the cycloid.
that while we were looking for the right angles we had to be using trial and error to
that he would walk the way we wanted, so that he would stay upright and
could follow its path without tripping over its weight

- Conclusion 2: In summary, the project was a great experience working on making the biped since it requires solving both mechanical engineering problems (weight and structure) such as the precision in the implementation of mathematical functions (the cycloid) for to have a good walk.

- Conclusion 3: Overall, the biped project achieved its main objective: generate a stable walk from a five-bar linkage controlled by a simple architecture, although there are still many improvements to be implemented.

  During development, several opportunities were identified that could optimize both the Performance, such as system efficiency, is a factor. For example, the capacity of neither the multiple serial ports available on the Orange Pi nor on the ESP32, which would have allowed to separate communication from sensing and control of servomotors, avoiding communication bottlenecks and potential blockages.

  Nor was the potential of the two cores of the ESP32 or the Orange Pi exploited, where One core could have been assigned exclusively to data acquisition and another to motion control, achieving truly parallel and smoother control.

  Another pending issue was the implementation of a Kalman filter, which would have allowed smooth the IMU data and generate a more accurate and controlled gait, instead of relying on a somewhat "brute force" strategy.

  Finally, I would have liked to perform a more in-depth dynamic analysis using the Lagrange's formalism, so that the biped could better take advantage of gravity and achieve a more natural and efficient movement.

  Even with these limitations, the project served as a solid foundation for understanding the integration between the mechanical, electronic and control parts of a bipedal robot, and leaves a a good starting point for more advanced versions in the future.

## References

[1] Tower Pro. Mg996r high torque metal gear dual ball bearing servo. https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf, 2018. Data sheet, Accessed: November 2025.

[2] CEVA Hillcrest Labs. bno08x series imu. https://www.digikey.com.mx/en/product-highlight/h/hillcrest-labs/bno080-imu, 2024. Accessed: November 2025.

[3] Angel Alvidrez. Video of bipedal walking. https://www.youtube.com/shorts/CoHYQln-zsA, 2025. YouTube Short.

Machine Translated by Google

**Upgrading the control electronics of an 8-inch Orion telescope Inches**

This project involved modernizing the electronics of an Orion telescope whose original controller had failed due to environmental wear. This telescope has two encoders per axis: a high-resolution encoder for trajectories and fine object tracking, and a low-resolution encoder for pointing. Since the motors and encoders were still in good condition, they were reused, and a new control system based on a modular architecture was implemented.

Two ESP32 microcontrollers were used, one for each axis (altitude and azimuth). These microcontrollers control the movements of each motor with the help of an L298N H-bridge and process the optical encoder signals in real time. This required designing an electronic conditioning stage for each encoder, which includes signal amplification using an LM324 in non-inverting configuration, offset correction with a potentiometer, and digitization using 74HCT05 inverter gates.

The conditioning circuit was tested on a breadboard and then transferred to a circuit The printout was made in the same laboratory.

Each ESP32 calculates trajectories based on encoder pulses and transmits position and status information to a Raspberry Pi via serial communication. This Raspberry Pi acts as the system's central node, running a Python script that receives coordinates and allows commands to be sent to the system.

In addition, Stellarium software is being integrated, which will allow users to select celestial objects in real time. Stellarium will send the coordinates directly to the Raspberry Pi. This feature is in its final stages of integration and testing.

Authors:

Ángel Alvídrez – Institute of Engineering and Technology, Autonomous University of Ciudad Juárez - aalvidrezhdz@gmail.com

Francisco Murillo - Advisor

# Audio amplifier transistor-based

Karla V. Suarez, Student, UACJ, IIT, Dept. of Electrical and Computer Engineering Angel H. Alvidrez, Viridiana A. Student, UACJ, IIT, Dept. Electrical and Computer Engineering, Velazquez, Student, UACJ, IIT, Dept. of Electrical and Computer Engineering

**Abstract—This document presents the design, construction, and simulation of a transistor-based audio amplifier using a voltage divider configuration. The circuit incorporates a 2N2222A NPN transistor and a TIP32C transistor. to amplify low-power audio signals intended for output devices such as loudspeakers. The methodology includes detailed calculations and procedures to achieve an efficient design Although the audio signals were successfully amplified, the prototype reveals areas for improvement, such as the management of thermal, distortion reduction and selection precision of components, which suggests future optimizations for a Improved performance and reliability. Among the topics covered These include transistor biasing, frequency response, and practical challenges in the implementation of electronic circuits.**

**Index Terms—Audio amplifier, transistors, divider voltage, electronics, transistor polarization, frequency, speakers, capacitors, resistors.**

## I. INTRODUCTION

A transistor-based audio amplifier is a electronic circuit designed specifically for au- Boosting the power of the audio signals it receives ensures their reproduction on output devices or speakers, allowing for higher audio volume. and clarity. As their name suggests, these devices They are based on the principle of amplification, where a transistor controls a larger current depending on a signal of much weaker input [1], [2]. This same principle often has different applications in areas of technology such as sound systems, musical instrumentation, sound in cell phones, inter alia.

One of the most important characteristics of an amplifier is Its great capacity to process low-power signals and convert them into stronger signals that can be controlled by loudspeakers. This is achieved through the use of transistors allowing for multiple configurations, among which the common emitter, common collector, and , Push-pull configurations. Each of these configurations They have specific characteristics that make them suitable for different amplification stages [3], [4].

### A. Operating principle of a transistor

The transistor falls into the category of being a device semiconductor that has three terminals; emitter, base and collector [1]. Figure 1 shows the visual diagram of bipolar transistors.

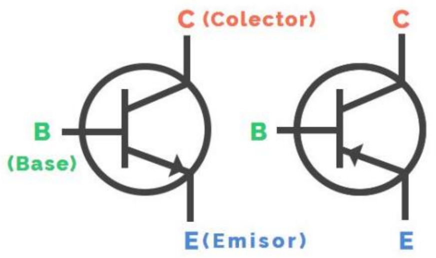



Fig. 1: Symbol diagram of bipolar transistors: NPN (left) and PNP (right), showing their terminals: Base (B), Collector (C) and Emitter (E) [5].

The main function of a transistor is to control the flow of current, acting as a switch or as an amplifier. When amplifying a signal, the transistor is able to increase the amplitude of a weak signal applied to its base, resulting in a higher power signal at the collector [1].

The relationship between a transistor and an amplifier is that the transistor has three terminals that can be designed... as amplifiers with specific gain, impedance and frequency characteristics [3].

Whereas a transistor is an individual component Capable of controlling and amplifying signals, an amplifier is a complete circuit that operates with transistors, among other components. components, in order to increase the amplitude of a signal Initially, according to the desired needs [4].

## II. THEORETICAL FRAMEWORK : AUDIO AMPLIFIER BY THROUGH A VOLTAGE DIVIDER

There are different configurations when amplifying a signal. This section discusses an audio amplifier based on a voltage divider, which is basically a circuit that uses this specific configuration to establish the necessary bias level in the transistor, allowing it to operate efficiently.

active region. This configuration is characterized by being simple and reliable, since the voltage divider, formed by two Resistors in series divide the power supply to generate a constant voltage that controls the behavior which the transistor counts [2].

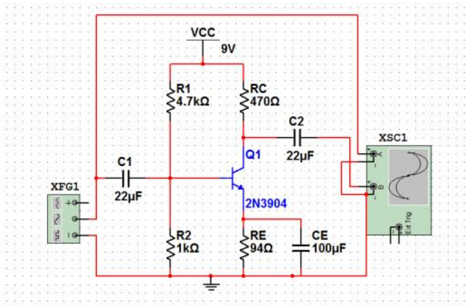



Fig. 2: Schematic of an audio amplifier based on a Voltage divider using a 2N3904 transistor. Includes resistors for polarization ( R1 and R2), resistor of collector (RC), emitter resistor (RE), a capacitor decoupling (CE) and coupling capacitors (C1 and C2) for signal input and output, powered by a 9V source [6].

Figure 2 shows the schematic of an amplifier audio based on a voltage divider using a transistor NPN model 2N3904. The circuit uses resistors for polarization, coupling and decoupling components, and a 9V power supply.

A voltage divider is typically composed of two resistors in series connected between the supply voltage (VCC ) and ground. The voltage at the junction of these resistors is used to bias the transistor's base. If R1 and R2 are the resistors of the divider, the bias voltage This is given by:

$$VB = \frac{R2}{R1 + R2} VCC \tag{1}$$

This voltage determines the base current (IB) of the transistor, which in turn controls the collector current (IC ) according to the relationship:

$$IC = ÿIB \tag{2}$$

where ÿ is the transistor's current gain.

## A. Components

The circuit also includes additional resistors and capacitors to improve its performance:

- RE and RC : The RE resistance in the emitter helps stabilize the operating point, while RC determines the amplifier gain.
- CE: The CE decoupling capacitor eliminates the component of alternating current in RE, increasing the gain of amplifier.
- C1 and C2: The coupling capacitors allow only The alternating current signals pass between the input and output stages, blocking any component of direct current.

The frequency response of the amplifier is determined by the capacitors and resistors in the circuit. lower cutoff frequency (fc) of a capacitor C in series with a resistance R, it is calculated as:

$$fc = \frac{1}{2ÿRC} . \tag{3}$$

This parameter is crucial to ensure that the amplifier operate in the frequency range (20 Hz - 20 kHz).

# III. METHODOLOGY

To carry out this project, a structured but practical process was followed [7], encompassing everything from the initial idea to the validation of the final design. The stages carried out are described below.

## A. Circuit design

For the circuit design of a transistor-based audio amplifier, an audio amplifier circuit using a voltage divider was used as a basis.

To obtain more efficient results in the project, the following steps were taken: two amplifiers, that is, the circuit design was divided into two phases.

1) First amplification stage: In the first stage For the amplification stage, the circuit sketch was made (Fig. 3) and then the relevant calculations were performed. In this phase, the guitar input is located, followed by the connection of a transistor in a voltage divider with capacitors.

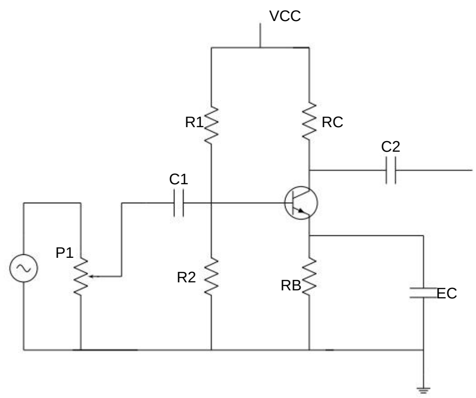



Fig. 3: Initial scheme of the first amplification stage.

This preliminary circuit defines the basic configuration for signal amplification, including the use of a transistor as the active element and passive components such as resistors and capacitors to stabilize and control the gain. component values, such as R1, R2, RE, C1, C2, and CE markings have not yet been determined, allowing for subsequent adjustment to meet design specifications.

In the first amplification stage, a 2N2222A transistor is used as a preamplifier, which is expected to achieve a collector current gain. of 100, for this the appropriate resistors were chosen for ensure a collector current of 15 mA and a current of 150 µA base. To choose these resistors, the analysis began by ensuring that the collector and emitter pins have a voltage of 50% of our power supply to obtain a signal range without excessive distortion; in this case, 18V. The emitter resistor is expected to have 10% of the power supply voltage. Therefore, for

To determine the value of the necessary resistance, the following was carried out the following calculation.

$$RE = \frac{10\% \times 18V}{15mA} = 120ÿ \tag{4}$$

For the collector resistance, it was assumed that the current in the collector and emitter is the same ($IE = IC$ ), for to achieve 50% of the source voltage at the emitter-collector, The collector resistance must drop by 40% of the Using a similar calculation to the previous one, we obtain:

$$RC = \frac{40\% \times 18V}{15mA} = 480ÿ \tag{5}$$

For the resistors of the voltage divider located at the base of the transistor, we ensure that R2 is 10 times larger than that of the emitter, guaranteeing a current of smaller base than in the collector and emitter.

$$R2 = 10 \times Re = 1.2Kÿ \tag{6}$$

And for the last resistor, the analysis of the voltage that would drop across the base of the transistor was performed; this voltage is known by Kirchhoff's laws, since the voltage at the emitter pin of the We chose the transistor to be 10% of our power supply. voltage, and the voltage drop from base to emitter is of approximately 0.7V, with this voltage we can calculate the expected voltage drop across the first resistor and so calculate it with the voltage ratio between base, the drop expected and R2.

$$V\,be\ ÿ\ 0.7V,\ Ve = 1.8V \tag{7}$$

$$Vb = 1.8 + 0.7 = 2.5V \tag{8}$$

$$VR1 = 18\ ÿ\ 2.5 = 15.5V \tag{9}$$

$$R1 = \frac{15.5}{2.5} \times R2 = 10.33Kÿ \tag{10}$$

Finally, we calculated the power that each re- would dissipate. assistance to ensure we buy the right type.

$$PRE = 1.8V \times 15mA = 27mW \tag{11}$$

$$PRC = 7.2V \times 15mA = 108mW \tag{12}$$

$$PR2 = \frac{2.5V^{2}}{1200ÿ} = 5.2mW \tag{13}$$

$$PR1 = \frac{15.5V}{10.333Kÿ} = 23.25mW \tag{14}$$

2) Second amplification stage: For the amplifier For power, we chose the TIP32-C transistor because it has a higher current capacity and moderate power dissipation, but this can be improved up to 20 times with a Heat sink, this transistor was subjected to a collector current of $IC = 150mA$ and its base current was adjusted to $IB = 15mA$ to be connected to the first stage amplification.

The procedure used to determine the values of the components in the first amplification stage were applied, Similarly, the values corresponding to the components in the second amplification stage were calculated. This ensured continuity in the design and consistency in the circuit's performance.

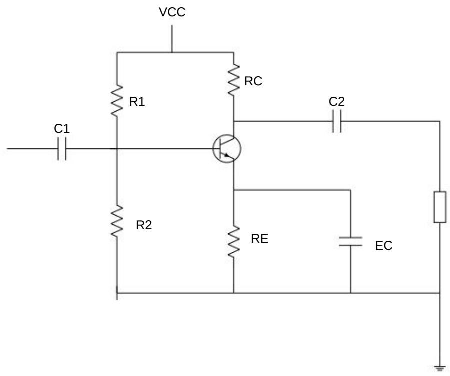



Fig. 4: Schematic of the final amplification circuit. This design shows the complete configuration of the second amplifier stage, where R1 and R2 form a voltage divider. to bias the transistor. Capacitors C1 and C2 act as couplers for the input and output signals, respectively, while CE provides higher gain By decoupling resistor RE, the circuit is designed to provide stable and efficient amplification.

3) Final circuit design: The final circuit design (Fig. 5.) was the result of an adjustment process based in the previous amplification stages. The main objective was to optimize the overall system gain and ensure that The circuit was stable and efficient. This design incorporates... components carefully selected to meet the specific amplification needs.

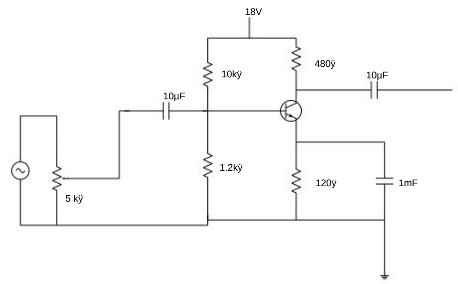
## B. Circuit simulation

The circuit simulation in its final stage was performed Using the free software Tinkercad, it is a tool Very useful for designing and simulating electronic circuits in a visual and interactive way. The implemented design includes a 2N2222A transistor and a TIP32C, which fulfill the role fundamental for audio amplification, which were selected thanks to their gain characteristics and power handling capacity (Fig. 6.).

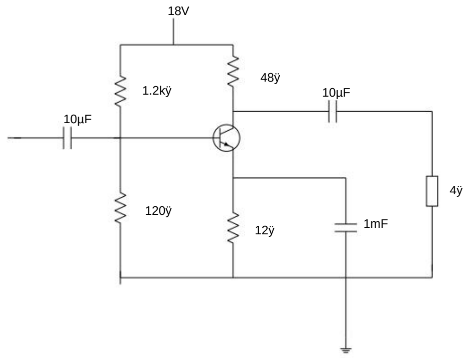
The circuit configuration includes the voltage divider mentioned earlier, formed by resistors at the base of the 2N2222A transistor, as this establishes the suitable polarization point for the transistor to function

Machine Translated by Google






(a) Circuit 1




(b) Circuit 2

Fig. 5: Two-stage amplification circuit. (a) First amplification stage: the initial circuit design is shown, where the values of the components, such as resistors and capacitors, were previously calculated to ensure an increase
(a) Efficient amplification of the initial signal. This stage establishes the basic gain of the system. (b) Second amplification stage: This stage on: sample determines the final circuit design, where the calculated component values optimize the overall gain and system stability. This stage reinforces the amplification achieved in the first stage, ensuring an output signal suitable for the desired purpose.

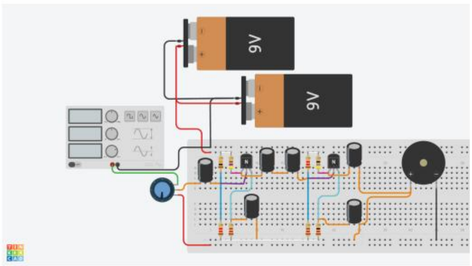



Fig. 6: Visualization of the simulation of the final stage of the amplifier circuit, made in Tinkercad.

Fig. 7: The transistor in the image has the same package (TO220) than the TIP32C used in the circuit design [8].

appropriately. On the other hand, capacitors were used of coupling (C1 and C2) to ensure a clean transfer of the alternating current signal between each layer, thus eliminating the unwanted direct current from other components. And because On the other hand, a decoupling capacitor (CE) was integrated which helps us to eliminate the voltage drop in the resistor of the emitter, thus maximizing profit.

For the power supply, it was determined to use 18V, so it was decided to use two 9V batteries each, the which supply the energy necessary for operation of the circuit.

### C. Circuit construction

After simulating the circuit and determining the optimal values For each component, it was assembled on the breadboard, and then tests and adjustments were carried out. necessary to ensure its proper functioning.

## IV. RESULTS

With the construction of the audio amplifier, practical tests were carried out to evaluate its performance under real-world conditions. The circuit was connected to a loudspeaker.

Radox 065-176 5-inch with 15W capacity and a With an impedance of 4ÿ, the device reproduced audio signals from an electric guitar. The results showed that it successfully amplified the audio signal, allowing its output at a perceptible volume. However, distortion was noticeable. at the output. Additionally, an increase in the temperature of the transistors and some resistors was observed during the Prolonged operation. Even with the use of a heatsink. The heat is not enough to maintain the temperature within an optimal range, highlighting the need for improvements in the circuit's thermal management This high dissipation Thermal factors affect the amplifier's efficiency and durability, requiring the use of more powerful heat sinks. Overall, the amplifier performs its function, but there are aspects that require improvement in output quality and performance. during prolonged use without getting too hot.

## V. ANALYSIS OF RESULTS

The audio amplifier built showed a performance Acceptable, successfully amplifying audio signals from an electric guitar. However, the analysis of results led us to identify areas for improvement to make its performance much more optimal and reliable.

Machine Translated by Google



For example, although the amplification was sufficient to reproduce a good volume of sound, distortion was detected. harmonic. Also, during prolonged use, the TIP32C transistor exhibited heating, along with the resistors. close together, even with a heat sink, and this could significantly reduce the component's lifespan and the amplifier's proper functioning. Or the robustness of the The design, although it allowed us to validate the functionality of the circuit, the protoboard is not very practical; ideally, it would be A small and compact design on a PCB, which has much more stable connections and avoids noise or false contacts. In addition to this, during the construction stage... While the components were being chosen, it was not possible to obtain exact 48 ÿ resistors, so two were used 100 ÿ resistors connected in parallel to obtain an approximate value of 48 ÿ. Similarly, for Instead of the 12 ÿ resistors, 24 ÿ resistors had to be used. in parallel. These are modifications that slightly affect the circuit behavior and areas for improvement. In short, although the objectives were met, improvement was needed. All these aspects will make the amplifier more efficient and stable, suitable for more demanding uses.

## VI. CONCLUSION

The design, simulation and construction of the transistor-based audio amplifier successfully achieved its main objective of amplifying audio signals for reproduction at an appropriate volume in on a output devices, such as speakers. The implementation of the amplification stages, using the 2N2222A and TIP32C transistors, allowed us to obtain a Adequate profit and satisfactory performance. However, During the results analysis, it was determined which areas should be improved for better functioning of the amplifier. During testing, heating was observed. of the TIP32C transistor, even integrating a heatsink heat, which compromises the thermal efficiency of the circuit. Furthermore, the output signal quality exhibits slight distortions. These issues are significant for modification. The design was for improved thermal resistance. Another important aspect was the selection of components, specifically The resistors. Because they weren't used. exact figures that were calculated due to the complications related to availability in stores, I've led to the use of parallel combinations, which has slight repercussions in the precision of the design. Finally, although the use of the protoboard is practical to check the functionality of the amplifier, for more reliable and durable use it was determined that the ideal is to transfer it to a PCB to make it more compact and with more precise connections to avoid false contacts. In conclusion, the development of this transistor-based audio amplifier enabled the exploration of Both the fundamental principles of amplification and the practical challenges that arise in the implementation of electronics were covered. Although the project achieved satisfactory basic functionality, it revealed several technical aspects that require improvements to increase the system's performance.

## REFERENCES


[1] AS Sedra and KC Smith, Microelectronic Circuits, 7th ed. New York, NY, USA: Oxford Univ. Press, 2015.

[2] RL Boylestad and L. Nashelsky, Electronics: Circuit Theory and Electronic Devices 11th ed. Mexico City, Mexico: Pearson Education, 2009.

[3] P. Horowitz and W. Hill, The Art of Electronics, 3rd ed. Cambridge, UK: Cambridge Univ. Press, 2015.

[4] A. Malvino and DJ Bates, Principles of Electronics NY, , 7th ed. New York, USA: McGraw Hill, 2016.

[5] Easy Myelectronics, "Transistor." [On-line]. Available: https://mielectronicafacil.com/componentes/transistor/. [Accessed: Nov. 18, 2024].

[6] SS Mohapatra, "How to increase the output power of this audio amplifier?", Electronica.guru, 2015. [Online]. Available in: https://electronica.guru/questions/84301/como-aumentar-la-potencia-de-salida-para-este-amplificador-d. [Accessed: 18-Nov-2024].

[7] "Design of transistor amplifiers with little mathematics," YouTube. https://www.youtube.com/watch?v=CkFdLhFLLLA (accessed: Nov. 20, 2024).

[8] "Heat sink for TO-220 transistor," Electronica Gabriel. [Online]. Available at: https://electronicagabriel.com/producto/disipador-de-calor-para-transistor-to-220/. [Accessed: 20-Nov-2024].

Autonomous University of Ciudad Juárez

Institute of Engineering and Technology

Department of Electrical and Computer Engineering

# Task 14 Controlling the position of a cart

Sensors and Instrumentation

Angel Alvidrez 225581:

Date: April 30, 2025

# Index

# 1. Objective

Develop a position control system for a motorized cart, using a an ultrasonic sensor and a PID controller implemented on Arduino, in order to maintain a fixed distance from a wall.

# 2. Introduction

In the field of mechatronics and automatic control, one of the fundamental challenges It is about ensuring that a system maintains a desired position or trajectory in the face of potential External disturbances. In this exercise, a position control system is implemented. for a cart with direct current motors, whose objective is to maintain a distance constant with respect to a wall using an HC-SR04 ultrasonic sensor.

The system is controlled by a PID (Proportional, Integral, Derivative) algorithm. which dynamically adjusts the speed and direction of the cart to minimize error between the current and desired distance. The latter is established remotely through via a Bluetooth interface, allowing real-time experimentation with different values reference.

The use of PID control in positioning systems is widely used in industry Due to its simplicity and effectiveness, this practice allows the student to understand to practically explain the operation of a closed-loop controller, as well as the importance of properly adjusting its parameters to obtain a stable response and precise.

# 3. Definitions and conceptual framework

## 3.1. Ultrasonic sensor

This sensor allows measuring distances through the time of flight of an ultrasonic pulse.

By sending a high-frequency sound pulse and detecting the echo reflected off an object, it is

It is possible to calculate the distance using the formula:

$$\text{distance} = \frac{340\ \text{m/s} \cdot t(\mu s)}{2} \cdot \frac{1\ \text{s}}{1{,}000{,}000\ \mu s} \cdot \frac{100\ \text{cm}}{1\ \text{m}}$$

Which, simplified, becomes

$$\text{distance (cm)} = t(\mu s) \times 0.01715$$

Figure 1: HC-SR04 Ultrasonic Sensor

## 3.2. PID Control

The PID (Proportional, Integral, Derivative) controller is one of the most used in system control. Its output is based on three components:

- Proportional (P): responds proportionally to the current error.
- Integral (I): accumulates the error over time to correct persistent errors.
- Derivative (D): anticipates the error trend to reduce oscillations.

The combination of these three actions allows for a balance between speed and stability. and accuracy. In this case, the PID is implemented with the ARDUINO PID v1 library in the ESP 32, configuring the parameters empirically until a behavior is achieved desired.

## 3.3. Bluetooth Communication

Bluetooth communication allows data to be sent and received wirelessly. In this In practice, the ESP32 integrated module is used along with the BluetoothSerial library. to receive the setpoint value in real time from an external device, allowing modify the position reference without reprogramming the system using a Android application created in MIT App Inventor.

## 3.4. Direct Current (DC) Motors

Direct current motors convert electrical energy into rotary motion. They are widely used due to their ease of control and low cost. In this practice, They used 6V motors, which allow for acceptable speeds for a laboratory or classroom environment.

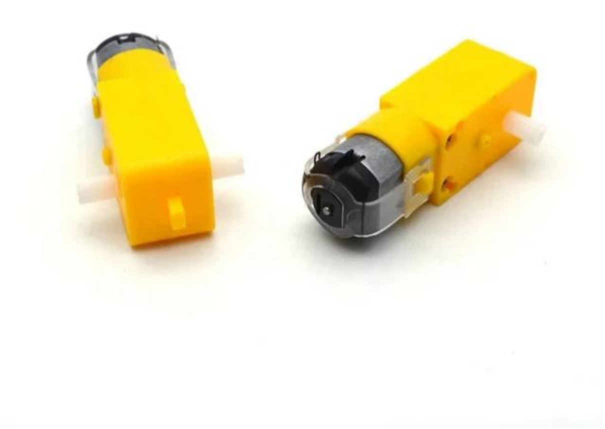


Figure 2: 6V DC Motor

3.5. PWM Signals

Pulse width modulation (PWM) is a technique that allows the speed of a DC motor to be controlled by adjusting the duty cycle.
cycle) of a digital signal. The higher the duty cycle, the greater the energy supplied to the engine, and therefore, its speed.

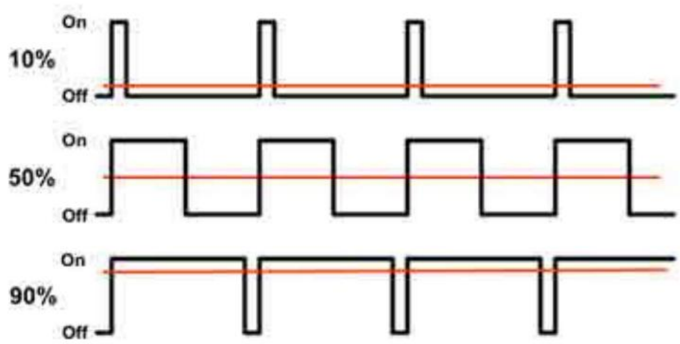



Figure 3: PWM Graph

3.6. Bridge H (L298N)

The H-bridge is a circuit that allows changing the polarity of the power supply to the motors, which allows their direction of rotation to be reversed. The L298N is a dual-bridge motor that

It allows you to control two motors independently, and can also control their speed using the PWM signal sent from a microcontroller such as the Arduino.

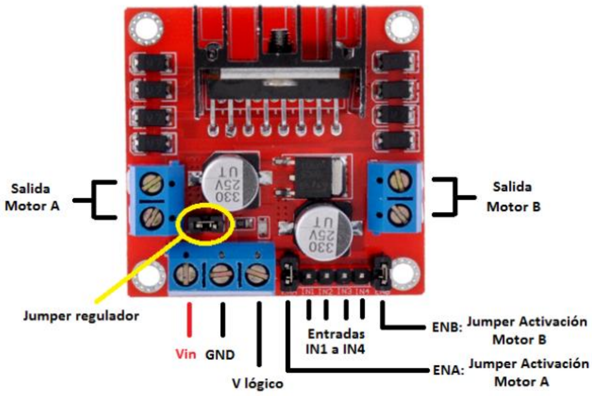



Figure 4: H-bridge L298N

## 4. Development

The position control system was implemented on an ESP32 board, which was responsible for both from the reading of the HC-SR04 ultrasonic sensor and from the control of the motors through the L298N H-bridge. The programming was done in the Arduino environment for Controlling an ESP32, using the PID_v1.h library to implement the controller PID and the BluetoothSerial.h library to receive data from a mobile device or computer via Bluetooth.

The program flow is as follows:

- Initialization of serial and Bluetooth communication.
- Pin configuration for motor control and ultrasonic sensor.
- Waits for a setpoint value sent via Bluetooth, which represents the distance desired distance in centimeters between the cart and the wall.
- Constant distance reading using the time of flight of the ultrasonic sensor.
- Calculation of the control signal using the PID algorithm.

- Application of the control signal to the motors, using PWM to adjust the speed according to the magnitude of the error.

- If the calculated PWM is very low (but not zero), a minimum value is imposed for to ensure that the engines respond.

The code was adjusted empirically, mainly the values of the constants of the PID and the minimum PWM threshold needed for the cart to move, because with values At very low speeds, the cart did not move. It was observed that, with the constants Kp = 1.0 and Ki = 0.0 and Kd = 0.2, the system's behavior was sufficiently stable and responded appropriate form for position corrections.

# 5. Code

This was the code used, as explained above:

Listing 1: PID control code for the cart

```
# include < PID_v1.h >
#include          "BluetoothSerial .h"
BluetoothSerial SerialBT;

// L298N motor pins
const int M1_A = 16                 ,   M1_B = 17;
const int M2_A = 18                 ,   M2_B = 19;

// HC - SR04 Pins
const int trigPin = 27                    ,   echoPin = 26;
long duration;
double readDistance() {
    digitalWrite(trigPin,                       LOW ) ;
    delayMicroseconds (2) ; // Optional, but can help stabilize
    digitalWrite(trigPin,                       HIGH);
    delayMicroseconds (10) ;
    digitalWrite(trigPin,                       LOW ) ;
    long duration = pulseIn ( echoPin                   ,   HIGH); // no timeout
    return duration * 0.01715;
```

```
}

// PID
double Setpoint , Input , Output ;
double Kp = 1.0 , Ki = 0.0 , Kd = 0.2;
PID myPID (& Input , & Output, & Setpoint, Kp, Ki , Kd , DIRECT);

// Minimum motor threshold
const int motorMinPWM = 135;

void setup() {
   Serial . begin (115200) ;
   while (! Serial ) delay (10) ;
   Serial.println ("\n--- Distance PID Control ---") ;

   SerialBT . begin (" ESP32_Carrito ") ; // Bluetooth Name

   // Motor pins
   pinMode ( M1_A , OUTPUT ) ; pinMode ( M1_B , OUTPUT ) ;
   pinMode ( M2_A , OUTPUT ) ; pinMode ( M2_B , OUTPUT ) ;
   // Ultrasonic pins
   pinMode (trigPin , OUTPUT ); pinMode ( echoPin , INPUT ) ;

   // Wait for first Bluetooth Setpoint
   Serial. println ("Waiting for Bluetooth Setpoint (1 byte in cm)... ");
   while ( SerialBT . available () == 0) delay (10) ;
   Setpoint = SerialBT . read () ;
   Serial. print ("Initial setpoint received: ") ;
   Serial . print ( Setpoint ) ;
   Serial . println (" cm") ;

   // Init PID
   Input = readDistance() ;
   myPID . SetMode(AUTOMATIC);
   myPID . SetOutputLimits ( -255 , 255) ;

   Serial. println ("Setup complete. Starting control... ");
```

```
}

void loop () {
    // Update Setpoint if a new one arrives
    if ( SerialBT . available () > 0) {
        Setpoint = SerialBT . read () ;
        Serial.print ("New Setpoint received: ");
        Serial . print ( Setpoint ) ;
        Serial . println (" cm") ;
    }

    // 1) Read distance
    Input = readDistance() ;
    Serial. print ( "Current Dist: ") ; Serial. print (Input); Serial. println ("cm" )

    // 2) Calculate PID
    myPID . Compute() ;

    // 3) Motor control
    int pwm = abs (( int ) Output ) ;
    bool forward = (Output > 0) ;

    if ( pwm > 0 && pwm < motorMinPWM ) pwm = motorMinPWM ; // Apply minimum PWM

    if (ahead) {
        analogWrite ( M1_A , pwm ) ; analogWrite ( M2_A , pwm ) ;
        analogWrite ( M1_B , 0) ; analogWrite ( M2_B , 0) ;
    } else {
        analogWrite ( M1_B , pwm ) ; analogWrite ( M2_B , pwm ) ;
        analogWrite ( M1_A , 0) ; analogWrite ( M2_A , 0) ;
    }

    // 4) Debug in serial monitor
    Serial. print ("SP:" ) Serial. print ( Setpoint ) ;
    Serial. print ("PV:") ; Serial. print (Input);
    Serial. print ("Err:") ; Serial. print ( Setpoint - Input ) ;
    Serial. print ("Out:" ) Serial. print ( Output ) ;
```

```
    Serial. print ("PWM:" ) Serial. println ( pwm ) ;


    delay (100) ;
}
```

## 6. Results

During system testing, a range of desired distances between 10 cm and 50 cm, dynamically transmitted via Bluetooth. Within this range, the cart maintained a stable position facing the wall, continuously correcting its distance by means of the PID controller.

It was observed that, when the desired distance was increased beyond 50 or 60 cm, the sensor The ultrasonic scale was beginning to show measurement errors. This is probably due to The sensor operates with a detection angle of approximately 30°, so At large distances, it may begin to detect the floor instead of the wall, affecting the reading accuracy.

Within the optimal range (10–50 cm), the system's behavior was smooth, without oscillations. very marked or very abrupt movements. It was also verified that the use of a threshold A minimum PWM (motorMinPWM) was required to prevent the motors from stalling. static with very small signals, especially when the error was low but still It needed correction.

The PID output signal adjusted well to the setpoint changes, allowing the cart adapts to new positions with a quick and stable transition. All the The system's behavior could be monitored in real time from the serial monitor.

# 7. Conclusion

The implemented system successfully controlled the position of a cart relative to to a wall using an ultrasonic sensor and a PID controller. The integration of the The Bluetooth module allowed dynamic modification of the setpoint, which facilitated experimentation. with different target distances without needing to modify the code or restart the system.

The ultrasonic sensor was found to be reliable within an approximate range of 10 to 50 cm. but it begins to fail at greater distances due to its conical detection shape, which can cause erroneous readings by picking up echoes from the floor or other objects. This highlights the importance of knowing the physical limitations of the sensors when designing a control system.

The use of the PID controller allowed for smoother behavior of the cart, adjusting its position continuously without needing to stop completely. Empirical adjustments to The constants Kp, Ki, and Kd were key to achieving stable and responsive performance.

Finally, this exercise was useful for reinforcing concepts of automatic control and electronics. and applied programming, showing how to integrate multiple components into a system functional.

Autonomous University of Ciudad Juárez

Institute of Engineering and Technology

Department of Electrical and Computer Engineering

# Task 16 Line-following cart

Sensors and Instrumentation

Angel Alvidrez 225581:

Date: April 30, 2025

# Index

# 1. Objective

Develop a functional prototype of a line-following cart using sensors infrared and DC motors controlled by PWM, using a ESP32 microcontroller. The system must be able to detect a trajectory in a track and follow it autonomously, demonstrating the principle of feedback between sensors and actuators.

# 2. Introduction

A line follower is a type of mobile robot that uses optical sensors to detect and follow a defined path, usually a line of contrasting color (black on (white background). These systems are one of the most common applications in robotics educational due to its ability to demonstrate fundamental concepts of electronics, control and programming in a practical way.

In this task, a prototype of a line-following cart was developed using a microcontroller. ESP32, infrared sensors consisting of a photodiode-phototransistor pair, and motors of Direct current controlled by pulse-width modulation (PWM). The system It was designed to follow a trail, reacting to changes in ground reflectivity. to adjust the vehicle's movement in real time.

# 3. Definitions and conceptual framework

## 3.1. Photodiode

The FD-IR333C is an infrared light emitting diode that typically operates at a voltage direct current of approximately 1.2V. Its function is to emit infrared radiation which will reflected by the surface under the sensor. Light surfaces reflect more light, while that dark cells absorb a large part of the radiation. In this project, the photodiode is connects in series with a 100ÿ current-limiting resistor, forming part of the sensor pair that detects the line on the track.

Figure 1: Transparent Photo Diode (FD-IR333C)

## 3.2. Phototransistor

The PT331C is a phototransistor that responds to infrared radiation emitted by the photodiode. When reflected infrared light strikes its base, the phototransistor allows the flow of current from the collector to the emitter. In this configuration, the collector is connected to a 10kÿ pull-up resistor towards 5V and also to the analog input of the ESP32, while the emitter is connected to ground. This arrangement allows for the detection of changes in the amount of reflected light as variations in output voltage, which the microcontroller interprets to determine the position of the cart with respect to the line.

Figure 2: Phototransistor (PT331C)

### 3.3. Direct Current (DC) Motors

Direct current motors convert electrical energy into rotary motion.

They are widely used due to their ease of control and low cost. In this practice,

They used 6V motors, which allow for acceptable speeds for a

laboratory or classroom environment.

Figure 3: 6V DC Motor

### 3.4. PWM Signals

Pulse width modulation (PWM) is a technique that

It allows you to control the speed of a DC motor by adjusting the duty cycle.

cycle) of a digital signal. The higher the duty cycle, the greater the energy supplied to the engine, and therefore, its speed.

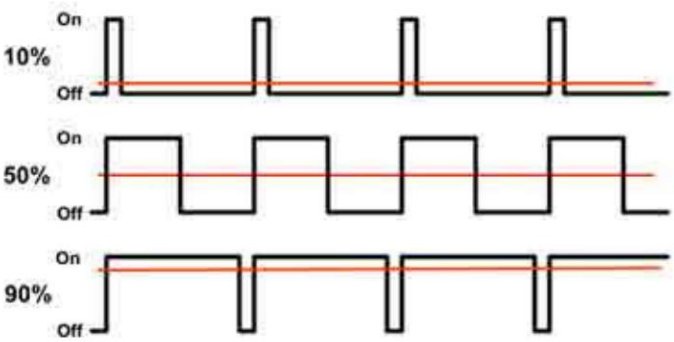



Figure 4: PWM Graph

## 3.5. Bridge H (L298N)

The H-bridge is a circuit that allows changing the polarity of the power supply to the motors, which allows their direction of rotation to be reversed. The L298N is a dual-bridge motor that It allows you to control two motors independently, and can also control their speed using the PWM signal sent from a microcontroller such as the Arduino.

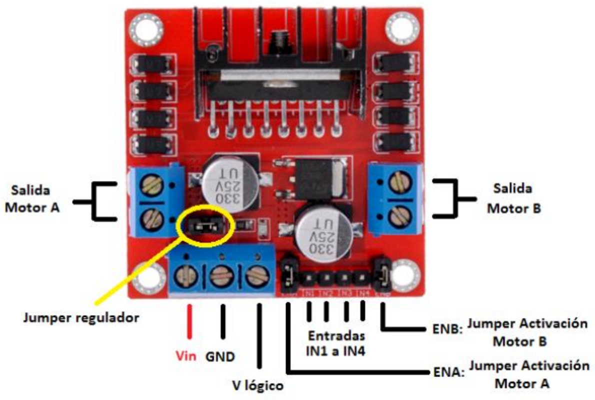



Figure 5: H-bridge L298N

# 4. Development

For line detection, two sensor pairs were used: each consisting of a FD-IR333C photodiode and a PT331C phototransistor.

For the photodiodes, a 100ÿ limiting resistor is used with a power supply of 5V and an approximate voltage drop of 1.2V, with both diodes in parallel which divides the current for each LED in half. Applying Ohm's Law:

Vled = 5V ÿ 1.2V = 3.8

Iled = $\frac{Vled}{R} \times \frac{1}{2} = \frac{3.8V}{100} \times \frac{1}{2}$ ÿ 20mA

This configuration ensures stable operation and adequate current consumption to protect both the LEDs and the ESP32.

The phototransistors were connected with the collector to a 100kÿ pull-up resistor which goes to 5V, and also to an analog input of the ESP32 (GPIO34 and GPIO35). The emitter of each phototransistor is connected directly to ground. Thus, when the sensor It detects a light surface (which reflects more IR), the phototransistor conducts and the voltage across The input pin decreases. When it detects a black line (less reflection), conduction decreases and the input voltage rises, allowing the position of to be clearly distinguished the line.

The system has two direct current motors, one for each wheel. These motors They were connected directly to the GPIO16, GPIO17, GPIO18 and GPIO19 pins of the ESP32 through a driver that allows H-Bridge switching for each motor.

Each motor receives two signals: one PWM control signal for speed, and another logic signal for the direction of rotation. The left motor is controlled by pins 18 and 19, and the right one by pins 18 and 19. through pins 16 and 17.

The main algorithm consists of constantly reading the analog values of the two sensors and compare their values against a threshold (approximately 1900). If the sensor The left side detects a line and the right side does not, the cart turns left; if the opposite occurs, Turn right. If both sensors detect a line or bottom, the cart stops.

Motor speed is controlled by PWM, with a value of 255 (maximum speed).

# 5. Code

This was the code used, as explained above:

Listing 1: Line follower cart code

```
# include <Arduino.h>


// Analog sensors
const int sensorIzq                    = 35;          // GPIO34 ( ADC1_CH6 )
const int sensorDer                    = 34;          // GPIO35 ( ADC1_CH7 )


// Engine A (left wheel)
const int motorA_IN1 = 18;                            // GPIO18
const int motorA_IN2 = 19;                            // GPIO19


// Engine B (right wheel)
const int motorB_IN1 = 16;                            // GPIO16
const int motorB_IN2 = 17;                            // GPIO17


// ======== PARAMETERS ========
const int THRESHOLD                    = 1900;
const int SPEED                        = 255;


void setup() {
    Serial . begin (115200) ;
    pinMode ( sensorIzq ,         INPUT ) ;
    pinMode (sensorDer)         , INPUT ) ;
    pinMode ( motorA_IN1        , OUTPUT ) ;
    pinMode ( motorA_IN2        , OUTPUT ) ;
    pinMode ( motorB_IN1        , OUTPUT ) ;
    pinMode ( motorB_IN2        , OUTPUT ) ;
}


void loop () {
    int leftVal = analogRead ( leftSensor ) ;
    int valDer = analogRead ( sensorDer ) -300;
```

```
    Serial. printf ("Left Sensor= %4d Right Sensor= %4d\n", leftVal,                valDer ) ;

if ( leftval < THRESHOLD && rightval > THRESHOLD ) {
        leftturn() ;
    }
    else if ( leftval > THRESHOLD && rightval < THRESHOLD ) {
        turn() ;
    }
    else {
        stall () ;
    }
delay (1) ;
}

void giraIzq () {
    analogWrite ( motorA_IN1          ,  0) ;            digitalWrite ( motorA_IN2          ,  LOW ) ;
    analogWrite ( motorB_IN1          ,  SPEED ) ; digitalWrite ( motorB_IN2            ,  LOW ) ;
}

void giraDor () {
    analogWrite ( motorA_IN1          ,  SPEED ) ; digitalWrite ( motorA_IN2            ,  LOW ) ;
    analogWrite ( motorB_IN1          ,  0) ;              digitalWrite ( motorB_IN2        ,  LOW ) ;
}

void stop() {
    digitalWrite ( motorA_IN1           ,  LOW ) ; digitalWrite ( motorA_IN2          ,  LOW ) ;
    digitalWrite ( motorB_IN1           ,  LOW ) ; digitalWrite ( motorB_IN2          ,  LOW ) ;
}
}
```

# 6. Results

Upon completion of the implementation of the line-following cart, its functionality was verified. on a track with a black line on a white background. The cart was able to identify correctly the line and react according to the programmed logic. When the sensor When the left side detected the line, the cart turned left; when it was the right side, It turned to the right; and when both sensors were out of line, the cart It stopped. The response was quick and stable during testing.

The infrared sensors responded with clear and distinguishable readings: high values when there was no line (white background) and low values when detecting the black line. This It allowed a reliable detection threshold to be established in 1900, with minimal calibration. in addition to the right sensor (-300) to compensate for differences in sensitivity. Furthermore, The PWM control provided a constant and suitable speed for the test without causing no sudden movements or deviations off the track.

During testing, the cart completed multiple routes without losing its path or needing External intervention. The system proved efficient for basic monitoring tasks. in line, fulfilling the objectives established for the practice.

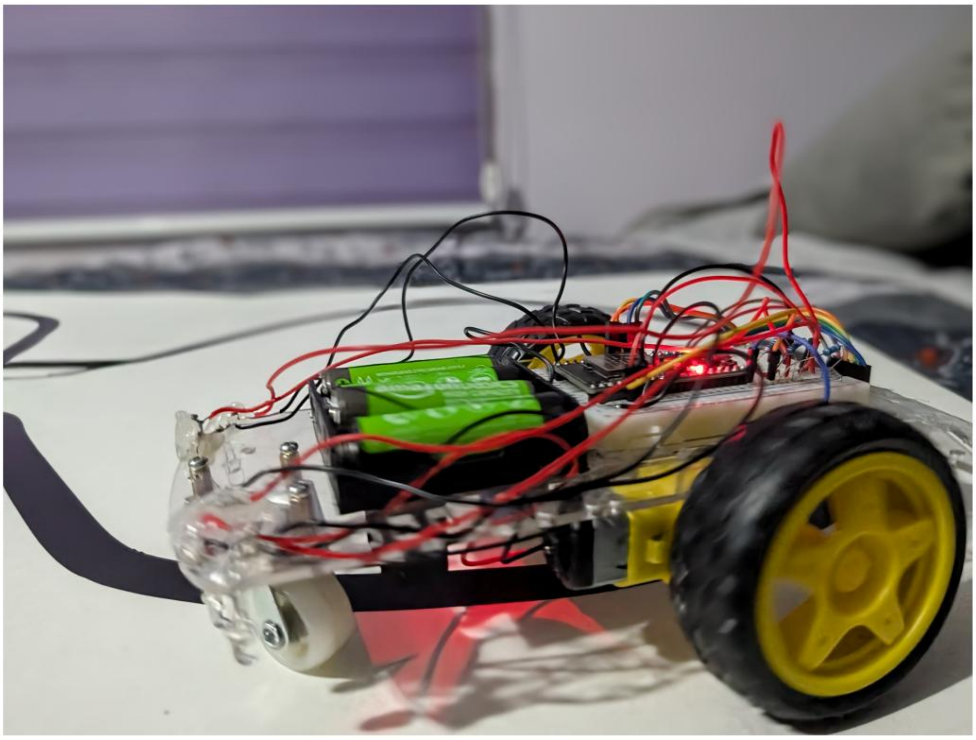


Figure 6: Cart in operation

# 7. Conclusion

The line-following cart project met the established objectives in a

satisfactory. The combination of infrared sensors (FD-IR333C and PT331C), the microcontroller The ESP32 and direct current motors made it possible to create an autonomous navigation system capable of following a line with precision.

The use of pulse width modulation (PWM) facilitated efficient control of the Motor speed, which improved the cart's stability. Analog reading The infrared sensors allowed for the reliable detection of variations between background white and the black line, which made it possible for the cart to react correctly to the different situations that arose on the track.

The system showed consistent behavior in the tests, performing precise turns. and stopping when necessary. Although the system is functional, problems can arise. improvements, such as the implementation of adaptive cruise control or the incorporation of additional sensors to improve accuracy on tighter curves.

Overall, the project was successful in its implementation and represents a solid foundation. for future robotics and automation projects.

Autonomous University of Ciudad Juárez

Institute of Engineering and Technology

Department of Electrical and Computer Engineering

# Task 15 Turning a light bulb on and off with applause

Sensors and Instrumentation

---

Angel Alvidrez 225581:

Date: May 6, 2025

# Index

# 1. Introduction

In this practical exercise, a system capable of turning on and off is designed and implemented. A 110V spotlight is activated by applause detection. This is achieved using a chain signal processing that ranges from sound capture to switching
Using a triac with an Arduino Uno, the circuit is expected to:

- Detect the sound of applause and convert it into a processable electrical signal.
- Filter and amplify only the frequency band where applause predominates (3 kHz - 6 kHz).
- Generate an adjustable digital pulse for the Arduino, based on the intensity of the applause.
- Count the number of claps in a 3-second interval to decide when to turn on. (2 claps) or off (4 claps) of the spotlight.
- Switch a triac (2N6071) with an MOC3022 optocoupler to isolate the signal from 110 VAC line control.

The gain is adjusted with a potentiometer in the feedback network (Rf), allowing to control the amplitude of the signal coming from the KY-038 microphone. The circuit with +5V from the Arduino and -5V generated with an L7905 regulator connected to a 9V battery.
The filter was implemented with a second TL081 designed to attenuate all frequencies outside the range of interest, ensuring that only applause generates a significant signal.
A third TL081 functions as a simple window comparator. The trigger threshold is Adjust with another potentiometer so that the output saturates and delivers a digital level (HIGH) when the filtered signal exceeds the reference level.
The comparator output is connected to pin 2 of the Arduino, which detects rising edges. to increase an applause counter.
A 3-second interval is measured (starting with the first clap) within which: 2 claps the Arduino turns on the simulation LED (and sends the signal to the MOC3022 to turn on the triac).

Four claps turn the Arduino off (and stop triggering the triac). Values exceeding Applause or the end of the 3 seconds restarts the count. The Arduino will control an MOC3022. which opto-isolates the control signal and triggers the 2N6071 triac, responsible for switching the light 110 VAC.

# 2. Definitions and conceptual framework

## 2.1. KY-038 Microphone

It is a sound sensor module with analog and digital output. In this project, only the sound sensor module is used. It uses the analog output, which provides a signal proportional to the intensity of the sound captured. The sensor consists of an electret condenser microphone and a small integrated amplifier. The signal is weak and requires additional amplification to be useful. in signal processing.

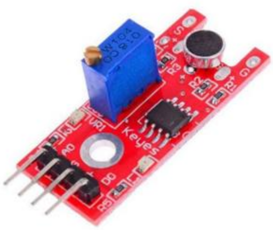


Figure 1: HC-SR04 Ultrasonic Sensor

## 2.2. TL081 operational amplifier

It is a low-noise operational amplifier with differential input and single output.

He used this component for all analog stages of the circuit:

- Initial amplification of the microphone signal
- Bandpass filter
- Level comparator

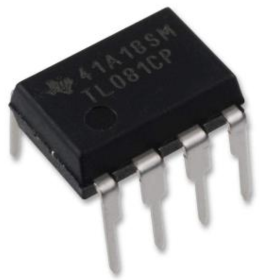


Figure 2: Op Amp TL081

## 2.3. Active bandpass filter

Applause has a strong energy component between 3kHz and 6kHz. To be sure
that only these frequencies pass through and the system does not react to ambient noise or
For other voices, an active bandpass filter with op amps was designed.

This filter attenuates signals outside that band, dissipating low noise (speech, ambient).
and high (sparking, interference).

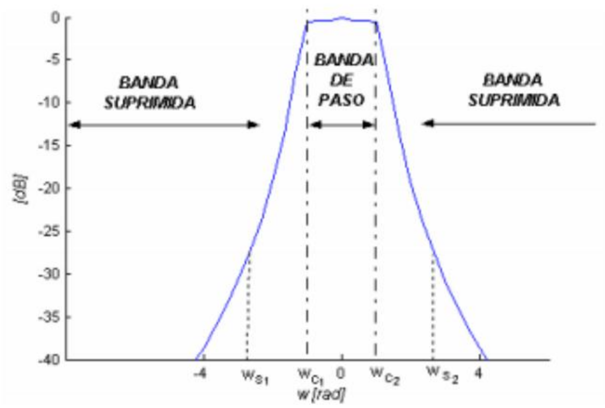



Figure 3: Output of a bandpass filter

## 2.4. Comparator with adjustable reference level

The filtered signal passes to a comparator built with another TL081, which is compared against an adjustable reference voltage with a potentiometer, when the signal exceeds that When the threshold is reached (by applause), the op amp saturates and its output abruptly changes to a high level (5V), acting as an applause detector.

The output is a digital signal that the Arduino can easily read.

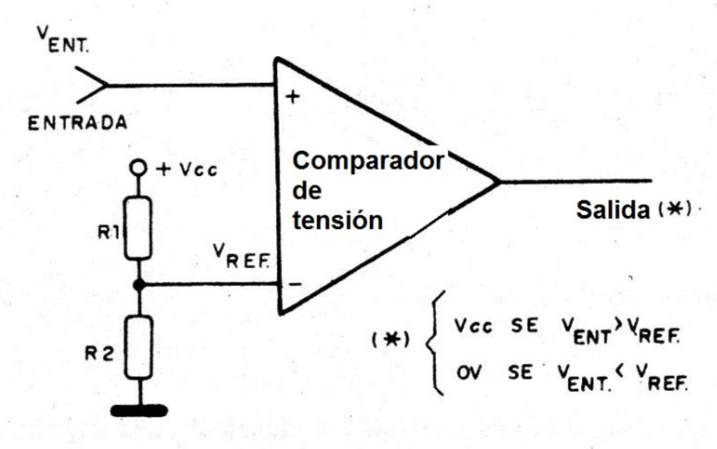



Figure 4: Example of a voltage comparator

## 2.5. MOC3022 Optocoupler

The MOC3022 is an optocoupler with TRIAC output designed to control loads alternating current without the need for a direct connection between the control part and the part of power. It consists of an internal LED diode which, when activated, triggers a TRIAC integrated.

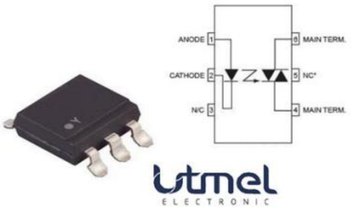



Figure 5: MOC3022 Optocoupler

## 2.6. TRIAC 2N6071

The 2N6071 is a TRIAC, a semiconductor device that allows current control It alternates between both polarities of the AC waveform. It functions as a controlled switch. by a small firing signal on its door

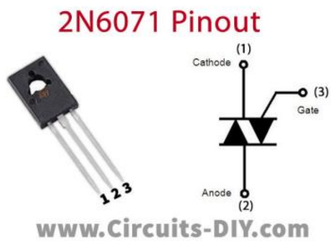



Figure 6: Triac 2N6071

## 3. Development

For the circuit design, everything was divided into functional blocks, each with a specific task. Clear. The first is the sound capture stage, which was done with a KY-038 microphone. This module has two outputs, one digital and one analog; for this project the following was used: It's analog because it delivers a signal proportional to the sound intensity. That signal It is of very low amplitude, barely in millivolts, so a stage was needed amplification to be able to process it properly.

## 3.1. Amplification with op amp (TL081)

Amplification was achieved using a TL081 in a non-inverting configuration, powered by a symmetrical ±5V power supply. We take the +5V directly from the Arduino, and for the At –5V we use an L7905 regulator with an external –9V source. In this stage also A precision potentiometer was placed as a variable feedback resistor, for adjust the amplifier's gain to control how loud a applause to make it detected.

## 3.2. Active bandpass filter (3kHz – 6kHz)

Before choosing the frequencies for the filter, it was decided to use the Audacity program. to analyze the frequencies present in applause, so an applause was recorded and applied the frequency analysis tool.

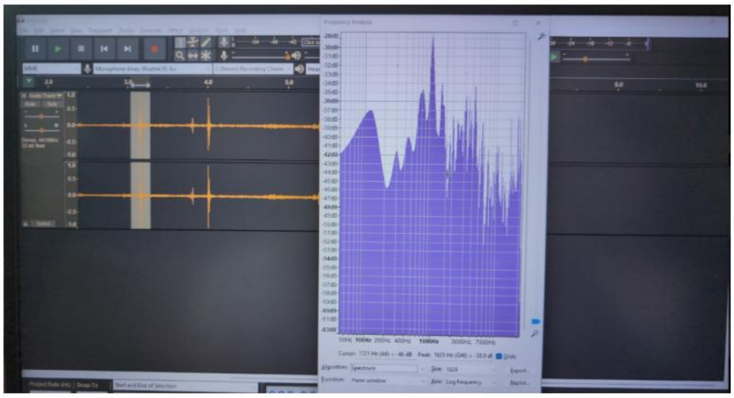


Figure 7: Frequency analysis in Audacity

Seeing that a large part of the applause frequencies were applied in the same program, a filter with a curve simulating the one that the circuit filter will perform.

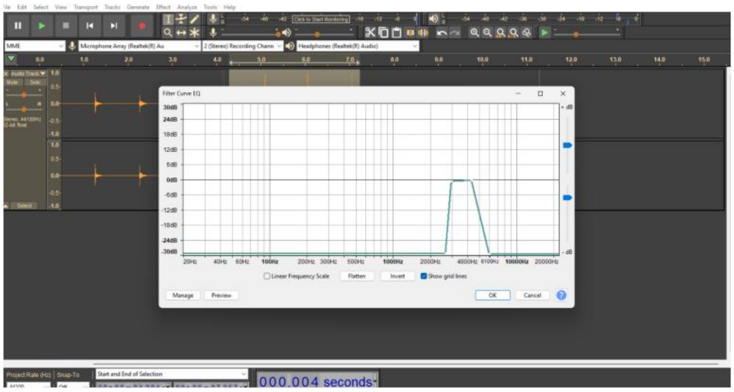


Figure 8: Filter simulation in Audacity

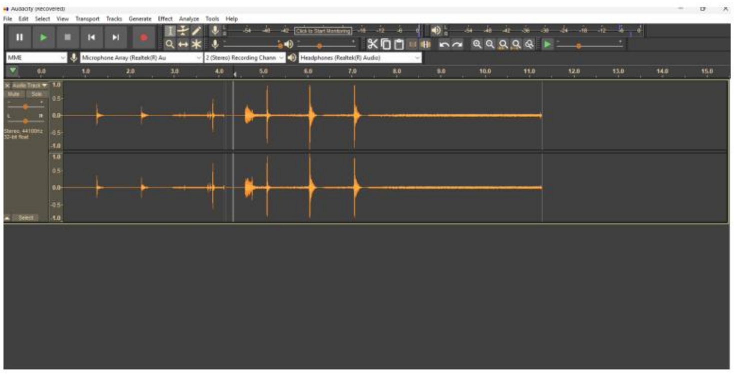


Figure 9: Filter output in Audacity

Once the filter is applied, it can be observed that the other background sounds are filtered out. of the audio and there are some peaks in the audio spectrogram, shown in figure 9 demonstrating that applause prevails after the filter.

Once frequencies for the filter were selected, the design of the filter began.

The capacitor was selected practically, choosing a value of 1nF since it is a an easy-to-obtain value. From that value, the necessary resistances were calculated for center the filter on the mid-frequency of:

$$f_r = \sqrt{f_H \cdot f_L} = \sqrt{6\ \text{kHz} \cdot 3\ \text{kHz}} = 4.242\ \text{kHz}$$

The interim band was defined as:

$$B = f_H \text{ ÿ } f_L = 6 \text{ kHz ÿ } 3 \text{ kHz} = 3 \text{ kHz}$$

And the filter quality, such as:

$$Q = \frac{f_r}{B} = \frac{4,242}{} = 1,414\ 3$$

With this data, and using the configuration of an active bandpass filter with amplifier

Operationally, the corresponding formulas were used to calculate the values of the

resistors, with the filter constant given by:

$$\frac{1}{RC} = 2\text{ÿ}B$$

$$R = \frac{1}{2\text{ÿ}BC} \text{ ÿ } \frac{0.1591}{1nF} \text{ ÿ } 53.033k\text{ÿ } 3kHz \times$$

The following expression was used for the resistance

$$f_r = 2\,\frac{\sqrt{1 + \frac{R}{R_r}}}{\text{ÿ } 2 \times \text{ÿ} \times R \times C}$$

$$R_r = R \left( \left( \frac{f_r \cdot RC}{0.1125} \right)^2 \text{ ÿ } 1 \right)^{-1}$$

$$R_r \text{ ÿ } 17,684 \text{ kÿ}$$

## 3.3. Threshold comparator

After the filter, the signal was sent to a third stage: a voltage comparator made

also with a TL081. Here the filter signal was compared against a reference voltage

adjustable with a potentiometer. If the signal exceeded that threshold, the comparator saturated.

its high voltage (close to 5V), generating a digital output that represented the detection

from a round of applause. This output was connected directly to a digital input of the Arduino.

One, who was in charge of processing the events. These 3 previous stages were simulated.

in Proteus 10 to observe its operation shown in figure 11.

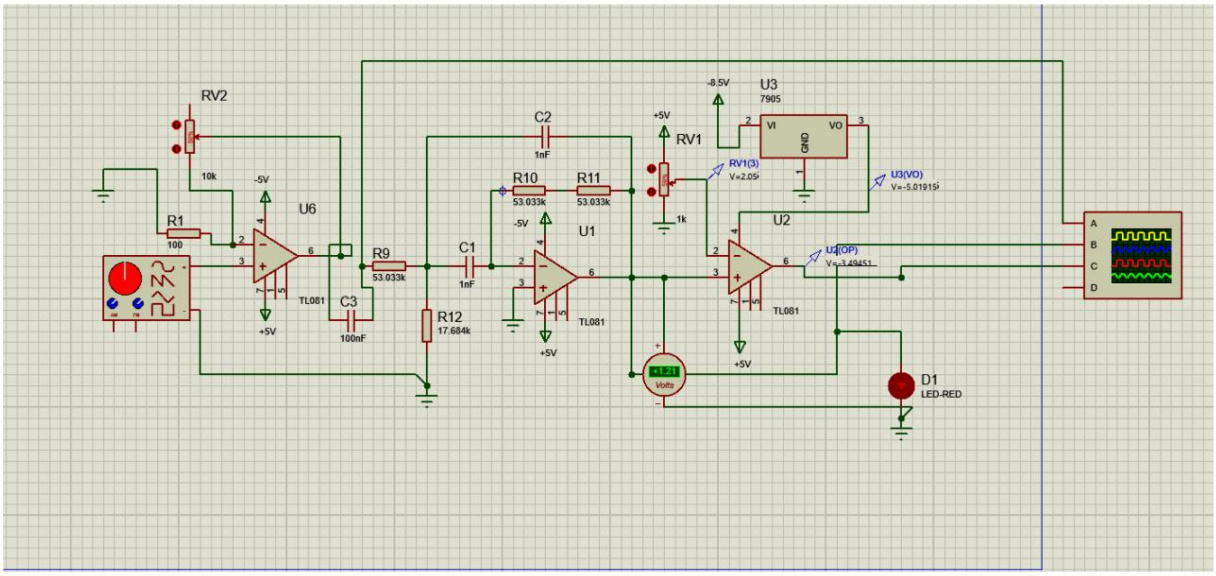



Figure 10: Frequency analysis in Audacity

## 3.4. Digital Processing

The Arduino was programmed to detect rising edges, that is, when the signal

It went from low to high. Upon detecting applause, it initiated a 3-second window to

count the following.

If there were 2 claps within that time, the light would turn on. If there were 4, it would turn off.

In the event that more than 3 seconds pass without reaching a valid amount or if a valid amount is detected

After more than 4 claps, the counter would automatically reset.

## 3.5. Power Stage

The final stage was triggering the 110V light bulb with a 2N6071 TRIAC. To isolate the

To control the mains voltage from the Arduino, an MOC3022 was used as an optocoupler.

It sent the signal to the internal LED of the MOC3022, and when it was activated, the TRIAC

It fired, allowing alternating current to pass to the light bulb.

All of this was integrated onto a single punched card with the three TL081s, the MOC3022, the

TRIAC and passive components, leaving the Arduino and the power supply connected by wires.

# 4. Code

This was the code used, as explained above:

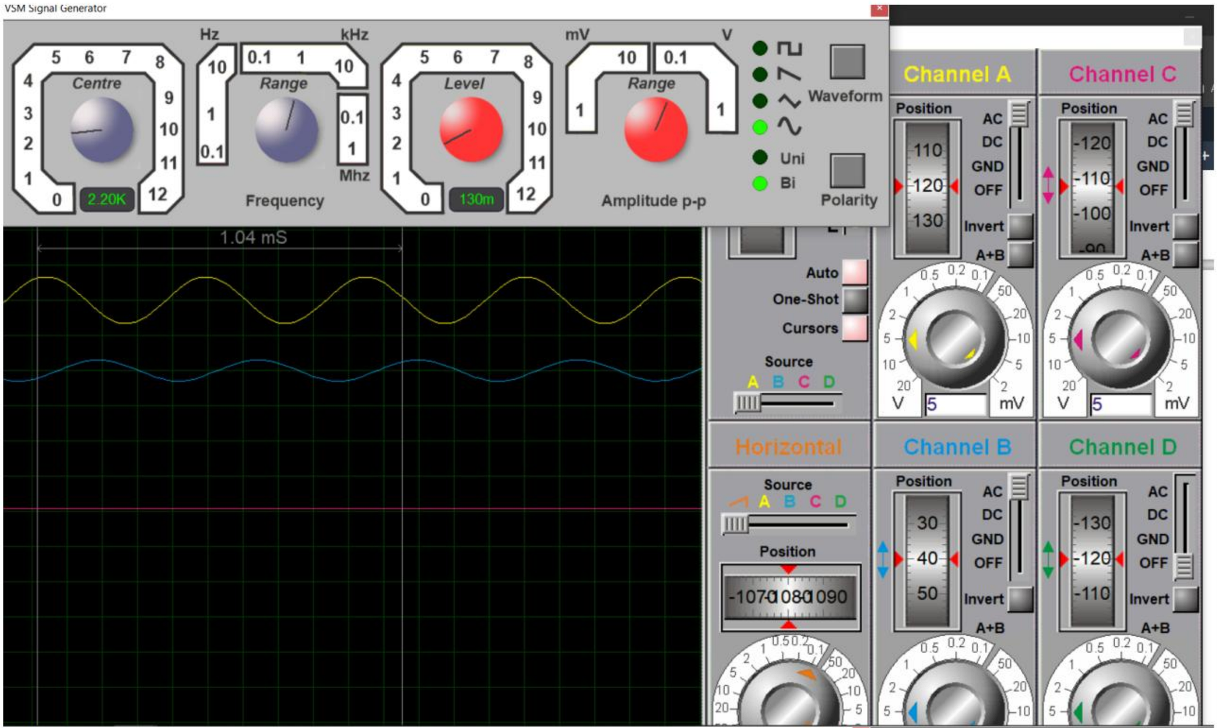
Listing 1: PID control code for the cart

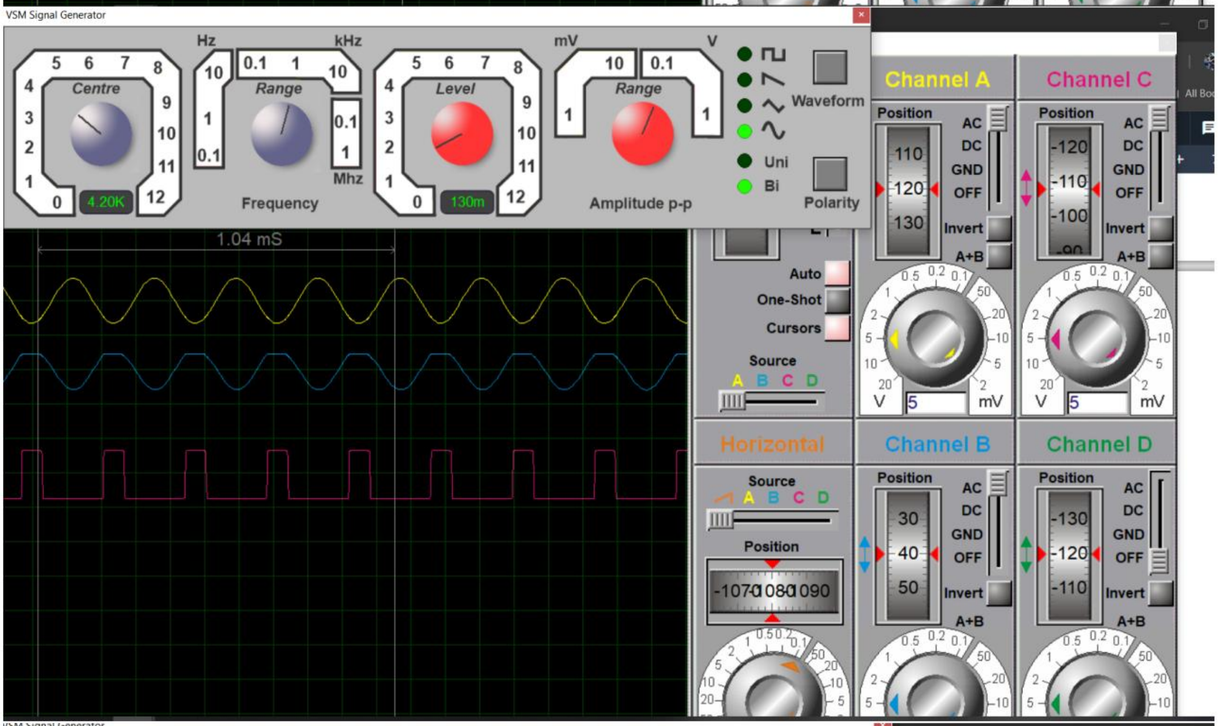
```
const int pinEntrada = 2;                    // Comparator input
const int pinLED = 4;                        // LED (simulates a light bulb)

int applausecounter = 0;
unsigned long startTime = 0;
bool counting = false ;
bool lightOn = true ;

void setup() {
    pinMode ( pinEntrance ,     INPUT ) ;
    pinMode ( pinLED ,     OUTPUT );
    digitalWrite ( pinLED ,          HIGH);       // The light bulb starts on
    Serial . begin (115200) ;
    Serial. println ("Applause counter started... ");
}

void loop () {
    static bool previousState = LOW ;
    bool currentState = digitalRead(inputPin);

    // Detect rising edge
    if ( currentState == HIGH && previousState == LOW ) {
        if (! counting ) {
            startTime = millis();
            counting = true ;
        }

        applause counter ++;
        Serial.print ("Applause detected! Total: ");
        Serial. println (ApplauseCounter);

        // Immediate actions
```

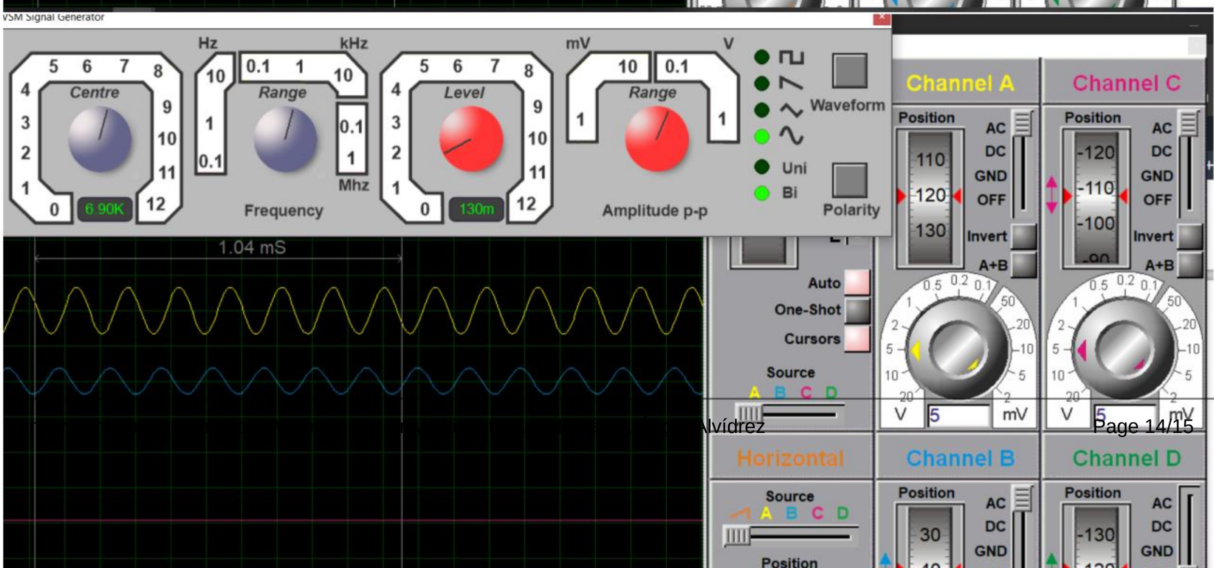
```
        if ( applausecounter == 2 && ! lightOn ) {
            digitalWrite ( pinLED ,                HIGH);
            lightOn = true ;
            Serial . println (" >> LIGHT ON <<") ;
        }
        else if ( applausecounter == 4 && lightOn ) {
            digitalWrite ( pinLED ,                LOW ) ;
            lightOn = false ;
            Serial. println (" >> FOCUS OFF <<") ;
        }
        else if (applausecounter > 4) {
            Serial. println (" >> More than 4 claps . Resetting counter . << " ) ;
            applause counter = 0;
            counting = false ;
        }

        delay (200) ; // Anti-bounce
    }

    previousState = currentState;

    // If the waiting time expires
    if ( counting && ( millis () - startTime >= 3000) ) {
        Serial . print (" >> Time expired . Counted applause : ") ;
        Serial. print ( applauseCounter ) ;
        Serial. println (". Resetting counter . <<") ;
        applause counter = 0;
        counting = false ;
    }
}
```

# 5. Results

The circuit was initially simulated in Proteus, replacing the microphone with a signal generator. In this simulation the behavior was correct: the filter allowed the signal path in the desired range and attenuated out-of-band frequencies as it I expected. However, several details arose during the physical implementation that forced to modify the design. The KY-037 module used as a microphone introduced an offset of approximately 2.5V in the signal, which directly affected the operation of the filter. To mitigate this effect, a ceramic capacitor was incorporated at the input of the circuit in order to block that DC component, an adjustment that was also applied subsequently in the Proteus simulation to reflect the real behavior.


VSM Signal Generator
Centre
2.20K
Range
Frequency
Level
130m
Amplitude p-p
Waveform
Uni
Bi
Polarity
1.04 mS
Channel A
Channel C
Horizontal
Channel B
Channel D


VSM Signal Generator
Centre
4.20K
Range
Frequency
Level
130m
Amplitude p-p
Waveform
Uni
Bi
Polarity
1.04 mS
Channel A
Channel C
Horizontal
Channel B
Channel D


VSM Signal Generator
Centre
6.90K
Range
Frequency
Level
130m
Amplitude p-p
Waveform
Uni
Bi
Polarity
1.04 mS
Channel A
Channel C
Horizontal
Channel B
Channel D

Furthermore, it was detected that the comparator, in its low output state, It introduced a small negative voltage that was undesirable at the time of excitation. the TRIAC gate. This problem was solved by placing a diode that blocks that value negative and ensures that the control signal remains clean and uninterrupted in the negative state at rest. With these adjustments, both simulated and practical performance will be improved. It was stable and the circuit fulfilled its objective of detecting acoustic signals within the specific frequency range.

# 6. Conclusion

This project allowed the integration of multiple concepts of analog and digital electronics to Develop a functional system for detecting specific frequencies using a filter active bandpass filter with TL081 operational amplifiers and a trigger circuit with Arduino-controlled TRIAC. Throughout the development process, its importance was verified. to consider not only the theoretical design, but also the practical details that arise when implementing real components, such as the unexpected offset of the KY-037 module or the comparator behavior in low state.

Simulation in Proteus was a key tool for validating the design before the physical weaponry, but it was also evident that certain elements do not behave in a identical between the simulation and the practice. This forced direct adjustments to the circuit, such as including a capacitor to eliminate DC offset or a diode to Ensure signal levels compatible with TRIAC triggering. These simple solutions But effective ones show how a good understanding of the behavior of each block is important. Functional allows you to solve problems without redoing the entire design.

Finally, a system was achieved that detects signals in the desired range and responds properly switching on an alternating current load, demonstrating practical use of active filters, comparators and power devices in a control application automated.

Autonomous University of Ciudad Juárez

Institute of Engineering and Technology

Department of Electrical and Computer Engineering

# Task 18 IoT Irrigation System

Sensors and Instrumentation

---

Angel Alvidrez 225581:

Date: May 21, 2025

# Index

# 1. Introduction

Water scarcity and inefficient use of water resources are persistent problems. in urban agriculture and gardening. In this context, automated irrigation systems They represent an efficient solution for reducing water waste and improving care of the plants.

The present project aims to develop an intelligent irrigation system using an ESP32 development board, soil moisture sensors, and the Blynk platform for Remote monitoring. This system allows real-time detection of humidity levels in the ground and activate irrigation automatically when needed. In addition, the user can View the plant's status from your mobile device, thanks to Wi-Fi connectivity and the user-friendly interface offered by Blynk [1].

The idea was inspired by previous open-source projects, such as the one developed by Viral Science [2], which showed a simple but functional approach to monitoring plants through the integration of sensors and controllers connected to Blynk Starting From this base, a customized solution was developed, adapted to the available resources. and specific needs of the environment where it will be implemented.

## 2. Theoretical Framework

The development of intelligent systems for monitoring and controlling plants has taken of great relevance in the field of automation and the Internet of Things (IoT). These systems allow for the optimization of resource use, such as water, improving efficiency and facilitating plant care through the use of interconnected sensors and actuators.

### 2.1. ESP32 Microcontroller

The ESP32 is a dual-core microcontroller developed by Espressif Systems, which It integrates Wi-Fi and Bluetooth connectivity, and offers high performance with low power consumption. energetic [3]. Thanks to its versatility and processing capacity, it is widely Used in IoT projects that require wireless communication and real-time control real, like the smart irrigation system presented in this project.

Figure 1: ESP32 Development Board

### 2.2. Soil Moisture Sensor

The soil moisture sensor used measures the electrical conductivity between two electrodes immersed in the substrate. The resistance varies depending on the content of water, allowing the soil moisture level to be determined by means of an analog signal that the microcontroller can interpret. This data is essential to activate the water pump and maintain adequate irrigation.

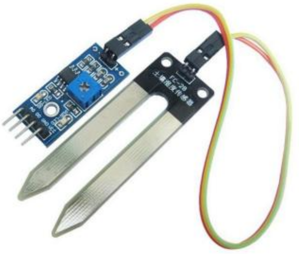


Figure 2: HR0040 Soil Moisture Sensor

## 2.3. DHT11 Sensor

The DHT11 sensor is a digital device that measures temperature and relative humidity. of the air. It has a capacitive humidity sensor and a thermistor for temperature, delivering values in digital format with sufficient resolution for applications basic environmental conditions. In the system, this sensor allows monitoring of atmospheric conditions. that influence the state of the plant.

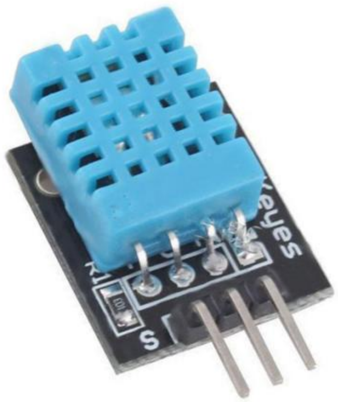


Figure 3: Air humidity and temperature sensor

## 2.4. PIR Sensor (Motion Sensor)

The PIR (Passive Infrared) sensor detects movement based on changes in the Infrared radiation emitted by objects in its field of view. It is used to detect the presence or passage of people or animals near the plant, enabling alerts and reports in the mobile application when the sensor is active.

Figure 4: PIR motion sensor

## 2.5. 5V Relay Module

The 5V relay module allows you to control AC or DC devices continuous higher voltage and current, such as the water pump, using the signal of Low voltage generated by the microcontroller. This module acts as a switch electronic shielding that isolates the microcontroller from high electrical loads, ensuring the system security.


Figure 5: 5V relay module

## 2.6. Water Pump and Flow Control

The water pump used is a 5V DC pump, designed for supply a constant flow of water suitable for watering small plants and pots. The key feature of this pump is its nominal flow rate of approximately 1.6 liters per minute [4] (0.0267 liters per second), which allows for precise calculation of the Total water consumption during operation. The pump is activated by a 5V relay module that functions as an electronic switch, controlled directly by the ESP32 to turn the irrigation on or off according to the conditions detected by the sensors.

Precise control of water flow is essential to avoid both insufficient and excessive irrigation. excess, optimizing the use of water resources and promoting healthy development of the plants. In addition, the system calculates and displays the accumulated consumption in real time. of water during pump operation, facilitating efficient monitoring.

Figure 6: 9V Water Pump

## 2.7. Blynk Platform

Blynk is an IoT platform that allows the rapid creation of graphical interfaces for the Remote control and monitoring of embedded devices. Through its mobile application, the Users can view data, receive notifications, and control actuators in real time real through customizable widgets [1]. In this project, Blynk facilitates monitoring of the environmental conditions and the remote management of the irrigation system.

## 2.8. LCD Screen

The system uses a 16x2 character alphanumeric LCD screen, which is based on the standard HD44780 controller. This screen allows important information to be displayed. such as temperature, humidity, soil moisture, and water consumption in a clear and user-friendly.

Figure 7: 16x2 LCD Screen

## 2.9. I2C Communication and PCF8574 Module

To simplify the connection between the ESP32 microcontroller and the LCD screen, the following is used:

An I2C adapter module based on the PCF8574 chip. This module reduces the number of Pins needed to control the screen, using only two data lines, SDA and SCL.

The PCF8574 chip acts as a port expander that allows control of the pins of the LCD via the I2C protocol, facilitating communication and reducing complexity of the wiring in the project.

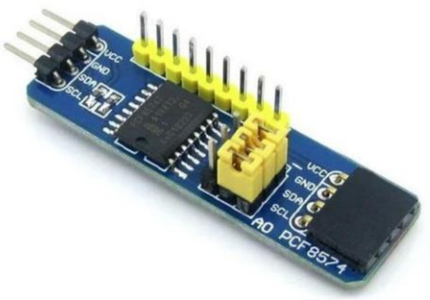



Figure 8: PCF Module

## 3. Development

The smart irrigation system was developed using an ESP32 microcontroller and sensors. environmental sensors, a relay module, and an LCD screen for data display. The main elements and the implemented operating logic are described below.

### 3.1. Microcontroller and communication

An ESP32 was used due to its ability to connect to WiFi networks and run the Blynk platform, which allows remote monitoring and control of the system through a Mobile application. Communication with sensors and actuators is done via pins digital and analog, as well as via the I2C bus for the LCD screen.

### 3.2. Sensors and actuators

The system integrates the following devices:

- Soil moisture sensor connected to analog pin GPIO36, which measures the percentage of moisture present.
- DHT11 temperature and humidity sensor connected to GPIO4, which provides environmental conditions.
- PIR sensor connected to GPIO27, detects movement for additional functions monitoring.
- 5V relay module connected to GPIO26, which controls the power on/off the water pump.
- Physical button on GPIO33 to manually activate or deactivate the relay.

### 3.3. Program Logic

The code uses the Blynk library to connect the device to the cloud and enable the Remote interaction. It also uses timers to periodically read the sensors. and update the LCD screen.

The water pump is activated or deactivated depending on the state of the physical or virtual button. During Its operation calculates the approximate water consumption based on the time that It remains on and has an estimated flow of 1.6 liters per minute.

### 3.4. Data Visualization

The LCD screen displays real-time temperature, ambient humidity, and humidity of the soil and accumulated water consumption. The information is updated every second. to maintain a clear and useful user interface.

### 3.5. System Operation

The system initiates the WiFi connection and synchronizes with the Blynk platform. Configuration begins. the pins and sensors, and periodic tasks are set up using timers for reading and updating data.

In the main loop, the state of the PIR sensor is evaluated to activate indicators of During movement, Blynk's tasks are executed and user interactions are monitored. through physical and virtual buttons.

This modular architecture allows the system to be flexible, efficient, and scalable for future improvements.

# 4. Code

This was the code used, as explained above:

Listing 1: PID Speed Code

```
// Viral Science www . viralsciencecreativity. com www. youtube. com /c/ viralscience
//Blynk IOT Smart Plant Monitoring System

# define BLYNK_TEMPLATE_ID " TMPL2I5BNgZpU "
# define BLYNK_TEMPLATE_NAME " Intelligent Irrigation System                "
# define BLYNK_AUTH_TOKEN " sP_bnZb903gxCO6MNO - h4YMteKdv925c "

# include < Wire .h >
# include < LiquidCrystal_I2C .h >
# include < WiFi .h >
# include < WiFiClient .h >
# include < BlynkSimpleEsp32 .h >
# include < DHT .h >

# define BLYNK_PRINT Serial

// LCD
LiquidCrystal_I2C lcd (0 x20          ,  16 ,  2) ;

// I2C Pins
const int SDA_PIN = 21;
const int SCL_PIN = 22;

// WiFi and Blynk
char auth [] = BLYNK_AUTH_TOKEN ;
char ssid[] =          " *********** ";
char pass [] =         " ******** ";

// Blynk Timer
BlynkTimer timer;

// Pins
```

```
# define SOIL_SENSOR_PIN        36
# define PIR_PIN                27
# define RELAY_PIN_1            26
# define PUSH_BUTTON_1_PIN 33
# define VPIN_BUTTON_1          V12
# define VPIN_CONSUMO           V13

// DHT11
# define DHTPIN 4
# define DHTTYPE DHT11
DHT dht ( DHTPIN     ,   DHTTYPE ) ;

// States
int PIR_ToggleValue = 0;
int relay1State            = LOW ;
int pushButton1State = HIGH ;

// Consumption
unsigned long onTime = 0; // in seconds
float consumptionLiters = 0.0;
const float LITERS_PER_SECOND = 1.6 / 60.0; // 0.0267

void setup() {
    Wire.begin(SDA_PIN)       ,   SCL_PIN ) ;
    Serial . begin (9600) ;

    lcd . begin (16     ,   2) ;
    lcd . backlight () ;
    lcd . setCursor (0      ,   0) ;
    lcd. print ("Initializing ");
    for ( int a = 5; a <= 10; a++) {
        lcd.setCursor(a       ,   1) ;
        lcd . print (".") ;
        delay (500) ;
    }
    lcd . clear () ;
```

```
  pinMode ( PIR_PIN , INPUT ) ;
  pinMode ( RELAY_PIN_1 , OUTPUT ) ;
  digitalWrite ( RELAY_PIN_1 , relay1State ) ;
  pinMode ( PUSH_BUTTON_1_PIN , INPUT_PULLUP ) ;

  dht . begin () ;
  Blynk.begin(auth , ssid, pass, " blynk . cloud ", 80) ;

  timer . setInterval (100 L , soilMoistureSensor ) ;
  timer . setInterval (2000 L , DHT11sensor ) ;
  timer . setInterval (500 L, checkPhysicalButton);
  timer . setInterval (1000 L , calculateConsumption ) ; // every second
}

// Soil moisture sensor reading
void soilMoistureSensor() {
  float value = analogRead ( SOIL_SENSOR_PIN ) ;
  value = map ( value , 0 , 4095 , 0 , 100) ;
  value = ( value - 100) * -1;
  Blynk . virtualWrite ( V3 , value);
  lcd . setCursor (0 , 1) ;
  lcd . print ("S:") ;
  lcd.print(value);
  lcd . print ("            ") ;
  Serial . print (" SOIL MOISTURE: ") ;
  Serial . println ( value ) ;
}

// DHT11 sensor reading
void DHT11sensor () {
  float h = dht . readHumidity () ;
  float t = dht . readTemperature () ;
  if ( isnan ( h ) || isnan ( t ) ) {
    Serial . println (" Error reading DHT11 ") ;
    return ;
  }
```

```
    Blynk. virtualWrite ( V0 Blynk .          ,   t) ;
    virtualWrite ( V1                          ,   h) ;

    lcd . setCursor (0             ,   0) ;
    lcd . print ("T:") ;
    lcd . print ( t ) ;
    lcd . print ("C ") ;
    lcd . print ("H:") ;
    lcd . print ( h ) ;
    lcd . print (" % ") ;
    Serial. print ("Temp: "); Serial. print ( t ) ;
    Serial. print ("C Hum: "); Serial. print ( h ) ; Serial. println ("%") ;
}

// PIR sensor
void PIRsensor() {
    bool value = digitalRead ( PIR_PIN ) ;
    Serial. print ("PIR Sensor Value: ") ;
    Blynk . virtualWrite ( V5                  ,   value);
    Serial . println ( value ) ;

    if ( value ) {
        Blynk. logEvent ( " pirmotion ", " WARNING ! Motion Detected !") ;
        WidgetLED LED ( V5 ) ;
        LED . on () ;
    } else {
        WidgetLED LED ( V5 ) ;
        LED . off () ;
    }
}

// Physical B
void checkPhysicalButton() {
    if ( digitalRead ( PUSH_BUTTON_1_PIN ) == LOW ) {
        if ( pushButton1State != LOW ) {
            relay1State = ! relay1State ;
            digitalWrite ( RELAY_PIN_1 , relay1State ) ;
```

```
            Blynk. virtualWrite ( VPIN_BUTTON_1 , relay1State ) ; delay (500) ;

        }
        pushButton1State = LOW ;
    } else {
        pushButton1State = HIGH ;
    }
}

// PIR status from Blynk
BLYNK_WRITE ( V6 ) {
    PIR_ToggleValue = param . asInt () ;
}

// Virtual button connected to relay
BLYNK_CONNECTED () {
    Blynk. syncVirtual ( VPIN_BUTTON_1 ) ;
}

BLYNK_WRITE ( VPIN_BUTTON_1 ) {
    relay1State = param . asInt () ;
    digitalWrite ( RELAY_PIN_1 , relay1State ) ;
}

// Function to calculate water consumption
void calculateConsumption() {
    if ( relay1State == HIGH ) {
        timeOn ++;
        consumptionLiters = ignitionTime * LITERS_PER_SECOND ;
        consumptionLiters = round ( consumptionLiters * 100.0) / 100.0; // round to 2 decimals
    }

    Blynk. virtualWrite (VPIN_CONSUMPTION , consumptionLiters ) ;

    lcd . setCursor (11 , 1) ;
    lcd . print ("L:") ;
    lcd. print ( consumptionLiters ) ;
```

```
    lcd.print (" ); // clear leftover text

    Serial.print ("WATER CONSUMPTION: ");
    Serial. print ( consumptionLiters ) ;
    Serial . println (" L") ;
}

// Main loop
void loop () {
    if ( PIR_ToggleValue == 1) {
        lcd . setCursor (5        ,   1) ;
        lcd . print ("M:ON ") ;
        PIRsensor() ;
    } else {
        lcd . setCursor (5        ,   1) ;
        lcd. print ("M:OFF ") ; WidgetLED
        LED (V6);
        LED . off () ;
    }

    Blynk . run () ;
    timer.run();
}
```

# 5. Results

During the implementation and testing of the smart irrigation system, it was possible to monitor real-time soil moisture, ambient temperature and humidity, as well as approximate water consumption based on pump operating time.

The consumption calculation was based on an estimated flow of 1.6 liters per minute, obtained from technical specifications of the pump. However, for more accurate measurements, It would have been ideal to use a flow meter that would allow recording the actual flow of water. However, flow meters are often expensive devices to integrate into projects. due to the low budget, the indirect method based on the following was chosen at this stage: the relay's operating time.

The data obtained showed an adequate response of the system to the conditions of the soil and environment, with the ability to manually activate or deactivate the pump The LCD screen provided clear and continuous information to the user, facilitating system status monitoring without the need for a remote connection.

Overall, the results confirm that the system meets the stated objectives. for efficient irrigation monitoring and control, with room for future improvements in accuracy of water consumption.

# 6. Conclusion

The development of the intelligent irrigation system based on ESP32 proved to be a solution effective for the automated monitoring and control of soil conditions and environment, optimizing water use through precise pump activation.

The use of sensors such as the DHT11 and the soil moisture sensor allowed us to obtain data reliable devices that were successfully integrated with the Blynk platform for monitoring remote control, while local control via a relay and a physical button increased the system versatility.

Although the calculation of water consumption was approximate due to the absence of a For the flow meter, the method based on operating time proved sufficient to estimate the volume used, while maintaining affordable costs.

The implementation of the LCD screen and the PCF module facilitated direct interaction with the user with the system, improving the experience and on-site control.

Finally, the project opens the door for future improvements, such as the incorporation of More precise sensors and the integration of intelligent algorithms for further optimization irrigation, contributing to a sustainable use of water resources.

# References

[1] Blynk Inc. Blynk documentation. https://docs.blynk.io/en, 2024. Documentation official for integrating IoT sensors and devices with Blynk.

[2] Viral Science - The home of Creativity. New blynk iot smart plant monitoring system. https://www.youtube.com/watch?v=PTJ9sAk2c2I&t=1s&ab_channel=ViralScience-ThehomeofCreativity, 2022. Video guide for implementation Plant monitoring with ESP32 and Blynk.

[3] Espressif Systems. Esp32 series datasheet. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 2023. Technical specifications of the ESP32 microcontroller used in the system.

[4] Electrozone Mexico. Mini submersible water pump 3-5v mb-35. https://electrozone.com.mx/tblarticulos_fulltext.php?field=Descripcion&key1=MB-35, 2023. Technical specifications of the pump used in the project.