Princess Sumaya University for Technology
King Abdullah II Faculty of Engineering
Electrical Engineering Department

Embedded Systems
Final Design Project, Fall 2024
IoT Enabled Agricultural Car

Authors:

Omar Hourani	20210095	Communication Engineering/IoT
Sandra Shawish	20210343	Communication Engineering/IoT
Aya Dwikat	20210631	Communication Engineering/IoT

Supervisor:

Dr. Essam Qaralleh

January 14, 2025

Table of Contents

Abstract

The IoT Enabled Agricultural Car demonstrates a cutting-edge mechanism to improve agricultural practices, and that's by integrating autonomous technologies with Internet of Things (IoT). This system is designed to conduct real-time soil quality analysis, providing farmers with insights to improve agricultural productivity and sustain resources remotely.

This system is run by the PIC16F877A microcontroller, which controls and coordinates the functionality of different sensors and actuators. An H-bridge motor driver ensures accurate and reliable movement across different terrains. A bluetooth module allows wireless data communication for remote analysis.

This project offers a portable solution that merges innovation with efficiency, providing actionable data to modernize agricultural practices. By harnessing IoT and automation, this car provides a transformative tool for optimizing precision agriculture, solving issues like resource efficiency and environmental sustainability in the industry.

1 INTRODUCTION

The "IoT Enabled Agricultural Car" project is a showcase of embedded systems technology, featuring a smart car that is controlled remotely via bluetooth. Using the PIC16F877A microcontroller, the car integrates sensors for effective data collection and precise motor control. The inclusion of Bluetooth communication allows for remote control, demonstrating the versatility of the system. This project highlights the seamless collaboration between hardware and software components, offering insights into the innovative applications of embedded systems in agriculture. The car also sends data collected by sensors to the PC controlling it, then that data is visualised on a dashboard so users can monitor their agricultural land.

1.1 THEORY

Technology can be extremely beneficial to optimize agricultural practices and reduce waste/losses. Efficient power consumption is of utmost importance when working with remote IoT deployments, this is why we opted to use a relatively power efficient MCU, PIC16F877A, with sensors that consume minimal power. It is also important that this car be controlled remotely and to be able to work in an environment with no cellular connectivity.

1.2 OBJECTIVES

Data Collection: Connect an array of sensors that collect agricultural data that is relevant to farmers.

Real Time Monitoring: Develop a system that receives agricultural data, and at the same time transmits data to ensure that the car is moving seamlessly in real time through a Bluetooth module.

Integration with IoT: Implement an IoT technology that allows the data that have been collected by the sensors from the agricultural land to be illustrated on a user interface, enabling the users to improve their agricultural practices based on real, accurate insights.

Hardware Integration: Ensure flawless integration of components, focusing on the PIC16F877A microcontroller, sensors, and actuators, to create a robust and efficient agricultural car.

Scalability and Adaptability : Achieve a system for the agricultural car that can adapt to various scenarios and applications in the industry. While also being able to scale seamlessly to additional cars and sensors ensuring good performance across different agricultural operations.

2 Design

2.1 Hardware Design

The hardware design of our IoT Enabled Car is an absolute synergy between innovation and practicality. The whole car system is driven by the PIC16F877A microcontroller, which is selected for its efficiency and compatibility with IoT applications. This microcontroller's main task is managing the car's movements and interactions with the agricultural environment. This microcontroller with the network of various sensors including the ultrasonic sensor, moisture sensor, temperature sensor, pH sensor, and the light dependant resistor sensor (LDR) monitors and delivers real-time insights on the agricultural land.

The two-way transmission of the data is done through a Bluetooth module. The system transmits real time data from the agricultural land, such as soil acidity, moisture, and the surrounding temperature to the PC, visualised on an online dashboard. On the other hand, it receives commands from the PC through a user-friendly interface to control the car's and some of the sensors movements (pH sensor, moisture sensor), along with toggling the LEDs on and off as required.

The design highlights a robust yet compact structure that safely integrates all of the components while ensuring smooth mobility and consistent monitoring. This systematic setup provides efficient interaction between sensors, actuators, and the microcontroller, optimizing the process of farming for different users with accurate insights and dependability.

2.2 Electrical Design

In the electrical design, power efficiency, reliability, and optimal performance were taken into consideration. The electrical connections were carefully done ensuring impeccable harmony between different components working together safely.

At the core of the system is the PIC16F877A microcontroller, which is responsible for all the interactions between all components and the external environment. These interactions include processing input from different sensors, controlling different motors, and communicating via the Bluetooth module. This microcontroller is powered by 5 volts, supplied through the H-bridge, which is powered using rechargeable lithium-ion batteries. These batteries provide a stable output of 3.7 volts each, the three batteries together give 11.1 volts, ensuring enough power for both the microcontroller and the DC motors.

To power the servo motor, which is in charge of the positioning of the moisture sensor and the pH sensor, an independent 5V regulated power supply set of batteries were used with a TIP110 transistor to manage its signal. This separation is crucial since the servo motor can draw significant current, especially when under load, which could cause voltage drops or fluctuation. By isolating its power supply, we prevent these fluctuations from affecting the stability of the microcontroller and the whole circuit, consequently it safeguards the microcontroller and the components from possible damage caused by power instability.

Additionally, the sensors including pH, moisture, ultrasonic, temperature (LM35), and LDR (Light Dependant Resistor), are powered using the micricontroller's analog input pins, their outputs are fed into the microcontroller's analog to digital converter (ADC). This enables the microcontroller to process real-time environmental data.

A voltage divider is used for the Bluetooth module to make sure it receives the right voltage levels for consistent communication, further enhancing power distribution across the system. As well as using 0.5 k ohms pull down resistors for each LED to regulate current, preventing overloading of the microcontroller's output pins.

Altogether, the electrical design for this system balances power distribution between all components, focusing on safety and accurate performance. With exclusive power supplies for key components, voltage regulators, dividers, and current limiting resistors, the system functions effectively, safeguarding against power inconsistencies while managing outstanding functionality in agricultural environments.

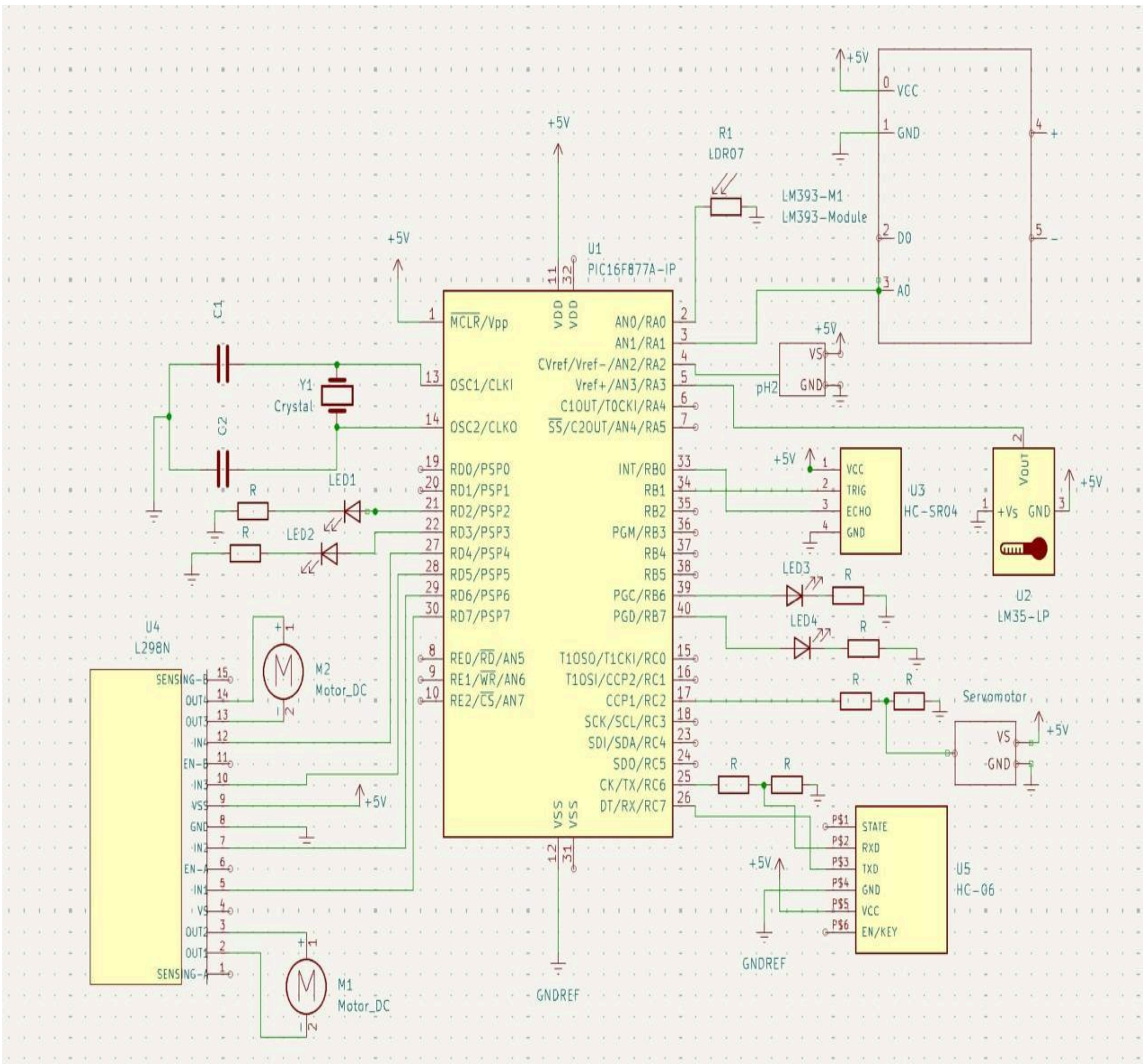


Figure 1 : IoT Enabled Agricultural Car Electrical Design Schematic

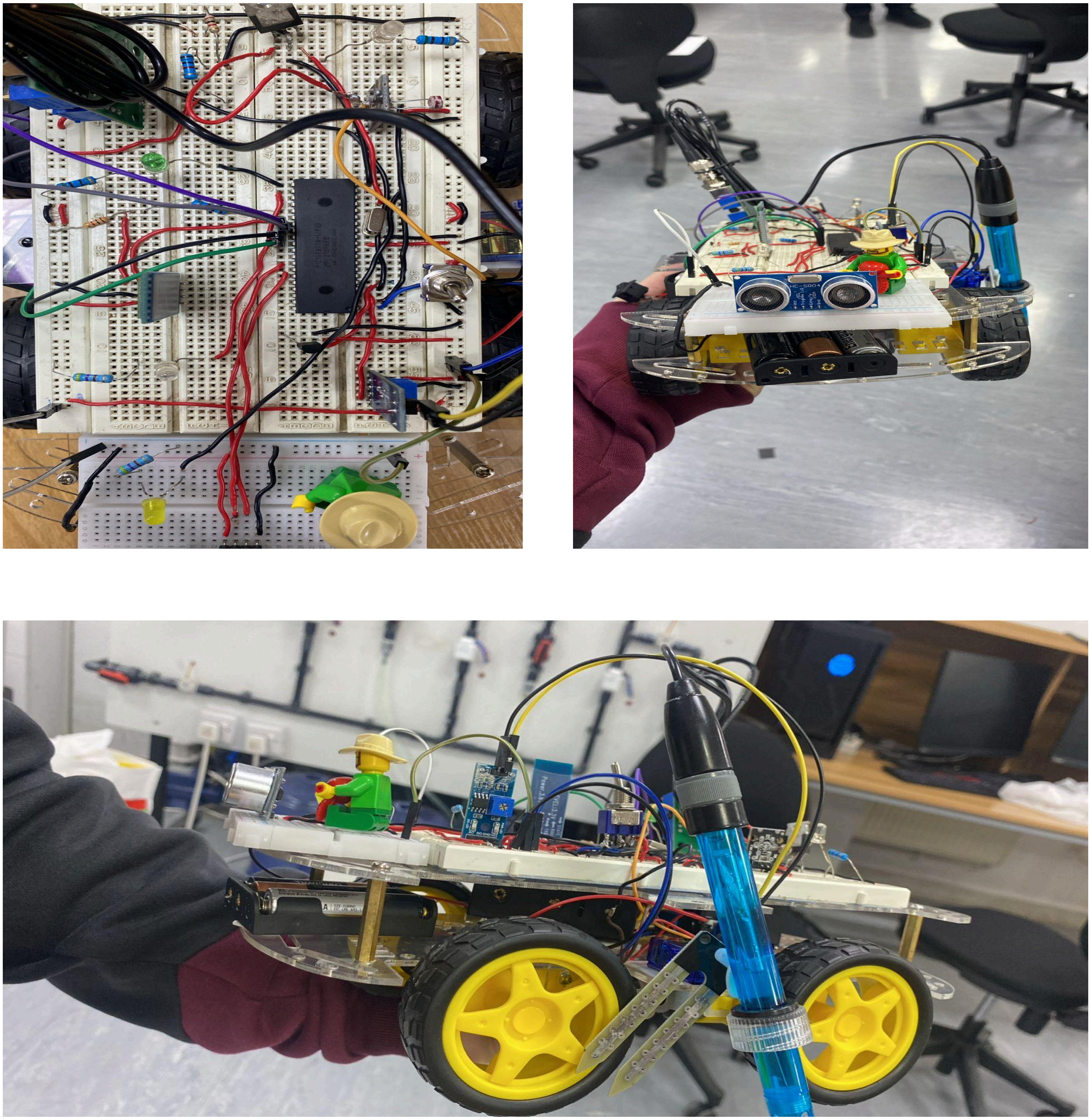


Figure 2 : IoT Enabled Agricultural Car Hardware Components

2.3 Software Design

The software system for the IoT Enabled Agricultural Car bridges the microcontroller hardware with an online platform, enabling seamless remote monitoring and control. It comprises three key components: PIC16F877A firmware, Python scripts, and a Bash script for setup and automation. Together, these components enable robust functionality and connectivity.

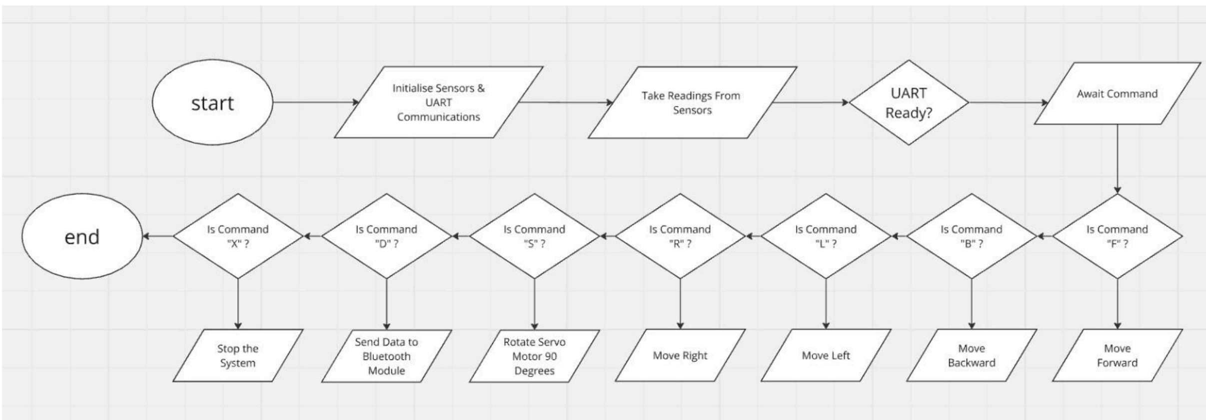


Figure 3 : Software Design Flowchart

PIC16F877A Code:

The code for the PIC16F877A microcontroller ensures efficient handling of peripherals and communication.

Key Features:

- **Modular Functionality:**

The firmware utilizes separate functions for each peripheral, such as sensors and actuators. This modular approach prevents conflicts and simplifies debugging.

- **UART Communication**:

The firmware supports UART-based communication, allowing the PIC to interact with the host PC via Bluetooth. This is critical for sending sensor data and receiving control commands.

- **Command Listening**:

The microcontroller listens on an open port for specific commands.

  - **Data Request Command**: Sends sensor data to the PC in **CSV format** when requested.
  - **Control Commands**: Executes actions on actuators based on received instructions.

Python Scripts:

The Python scripts act as the intermediary between the PIC microcontroller and the ThingSpeak platform. These scripts have separate versions for Windows and Unix-based operating systems to ensure compatibility across systems.

Key Features:

1. **Bluetooth Connectivity**:
   The script establishes a connection with the HC-06 Bluetooth module to facilitate communication with the PIC.
2. **ThingSpeak Integration**:
   Credentials for the ThingSpeak platform are defined within the script, enabling secure data upload. This allows sensor data to be visualized on an online dashboard.
3. **User Interface**:
   A simple user interface is initialized to control the PIC.

   This interface allows the host device's keyboard to send commands to the PIC.

4. **Data Polling and Processing**:

   The script sends a **data request command** to the PIC every 2 seconds.

   Received CSV data is processed and converted into **JSON format** for ThingSpeak compatibility.

5. **Data Upload**:
   Processed sensor data is sent to ThingSpeak for real-time monitoring.

6. **Dual Functionality**:
   The Python script facilitates both:

   Data collection and upload to the cloud.

   Remote control of the system via the host PC.

Bash Script:

The Bash script is a setup utility designed to automate the installation and configuration of dependencies for the Python script.

Key Features:

- **Virtual Environment Creation**:

  The script sets up a dedicated Python virtual environment to avoid dependency conflicts.

- **Dependency Installation**:

  Required libraries and modules are installed automatically within the virtual environment

- **Automation**:

  After running the script, the Python program can be executed without any additional setup, streamlining deployment.

Software Workflow:

The interaction between software components operates as follows:

1. **Data Collection**:

The PIC microcontroller collects data from connected sensors and listens for commands.

2. **Communication**:

Using UART, the PIC transmits sensor data to the host PC via Bluetooth.

3. **Data Processing**:

The Python script processes the received data and converts it into the required JSON format.

4. **Data Upload**:

The processed data is sent to ThingSpeak, where it is displayed on a dashboard.

5. **Control**:

Commands from the host PC, entered via the Python script's UI, are sent to the PIC to control actuators.

Control Commands:

Command	Task
F	Moves Forward
B	Moves Backward
L	Moves Left
R	Moves Right
S	Rotates Servo Motor 90 Degrees
D	Sends Data to Bluetooth Module
X	Stops the system

Table 1: Control Commands table

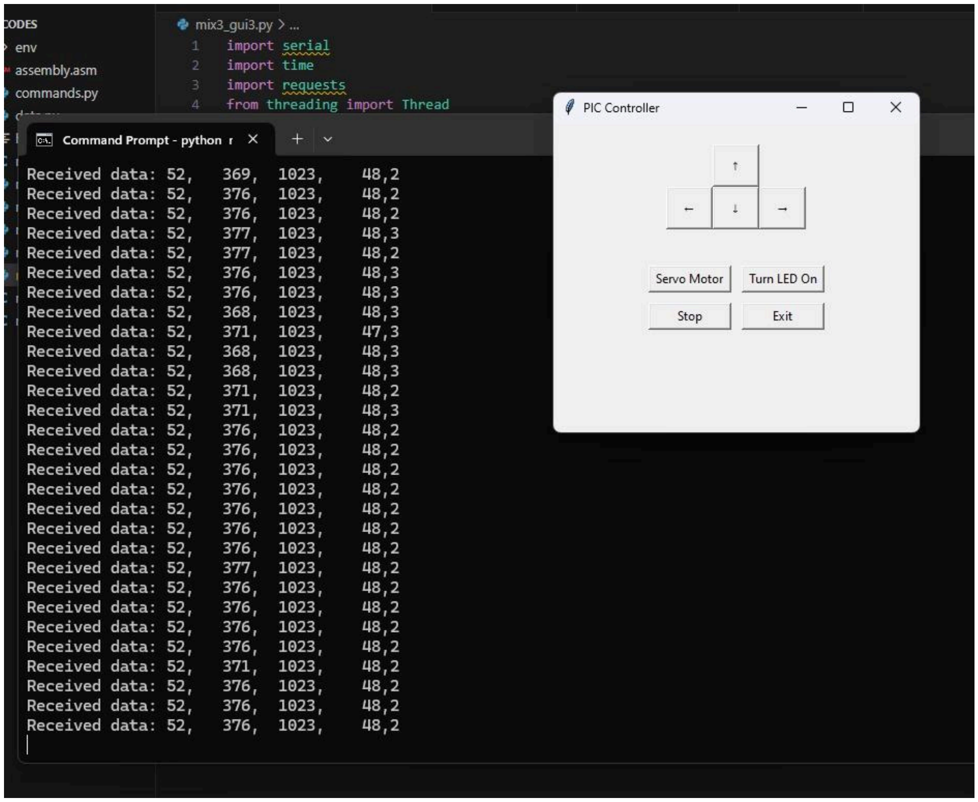


Figure 4: Virtual Remote Control

3 Problems and Recommendations:

1. Servo Motor with PIC::

   Issue: a pic was damaged when connected to the servo because the servo drew 500 mA from the pic that can only output a maximum of 25 mA.
   Recommendations: Using a separate 5 volts to power the servo and we connected it to the circuits common ground as well as using a TIP110 and a voltage divider to adjust the signal's voltage to match the input requirements of the servo

2. Communication:

   Issue: Initially, we faced problems in the UART serial communication in the Bluetooth module
   Recommendations: we used three libraries that are allowed according to the requirements which are : Conversions (For data type conversions) , C_String (For making strings through character arrays), and UART (For UART communications, for reading and writing).

3. Voltage drop on the BreadBoard:

   Issue: we faced a problem with a breadboard where is continually had a voltage drop that affected the workflow
   Recommendations: after figuring that out we continually checked the voltage on the breadboard after connecting different components.

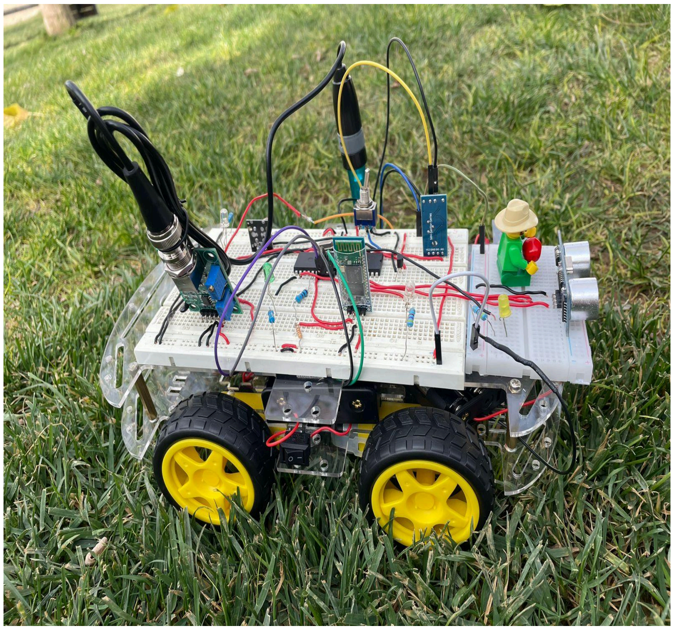
4. pH sensor:

   Issue: pH sensor configuration and readings
   Recommendations: we used an arduino microcontroller and the pH buffer solutions to correctly configure the pH sensor and find the correct equation to convert from voltage to pH readings.

4 Results and Discussion:

Our IoT Enabled Agricultural Car provided impressive results! The system successfully inspected key parameters, including soil moisture, pH, temperature and more, visualising logical readings on the dashboard. The car's movement, LED control, and the servo motor positioning worked seamlessly, showcasing the reliability and the receptiveness of the system. The real-time collected data was transmitted to the PC via Bluetooth, enabling efficient monitoring and control. These successes showed that our car system performed well, with smooth operation and accurate data processing, highlighting its potential for utilizing it in optimizing agricultural applications.

Figure 5 : Platform Dashboard



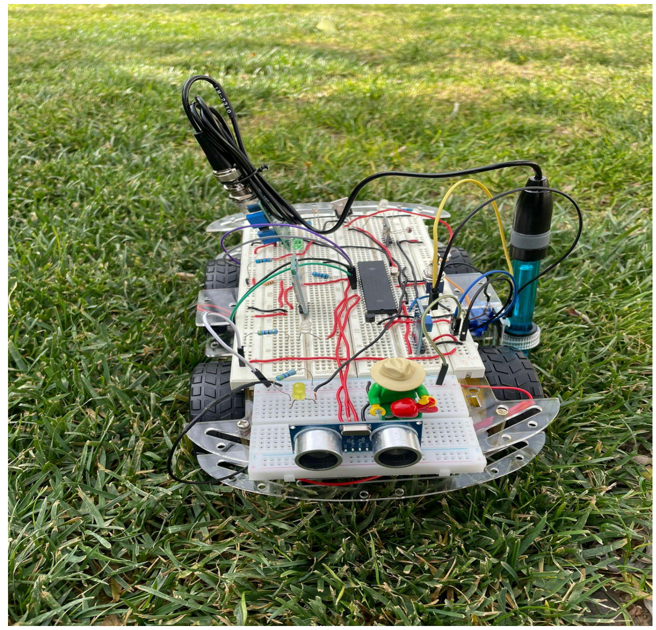


Figure 6 : IoT Enabled Agricultural Car Structure

Youtube Video Link : IoT ENABLED AGRICULTURAL CAR

Github Link : IoT_Enabled_Agricultural_Car/README.md at main · OmarHourani0/IoT_Enabled_Agricultural_Car · GitHub

5 Conclusion:

In conclusion, the IoT Enabled Agricultural Car provides a groundbreaking solution for the future of farming. By combining the power of IoT technology and an array of advanced sensors, this system provides farmers with real-time data, and actionable insights to assess and improve soil quality with outstanding accuracy.

The outcome of this project extends beyond increased harvests; it drives better resource management, limits waste, and supports a more smart and sustainable farming method.

6 References and Tools Used

1- Course Slides
2- PIC16F877A Datasheet
3- PSUT eLearning site, http://www.elearning.psut.edu.jo/