Group 16

BI-Modes Robot

Hardware Final Project

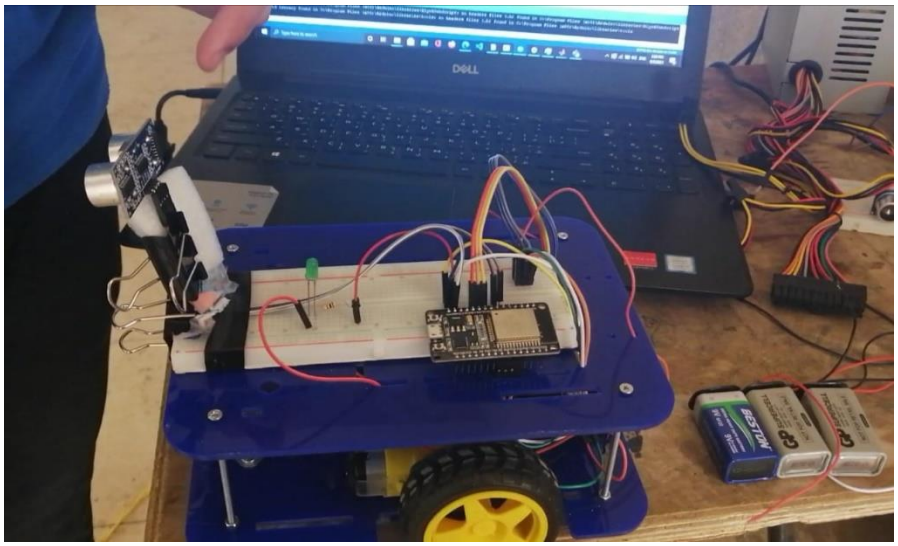
Mohamed Yasser Mohamed Ahmed	**18012553**
Monica Ihab ElDabaeh	**18012559**
Eslam Abdullatif Dyab	**18010333**
Islam Ashraf Muhammed Muhammed	**18010324**
Mai Bendary Soliman	**18011912**

Communication 2nd year

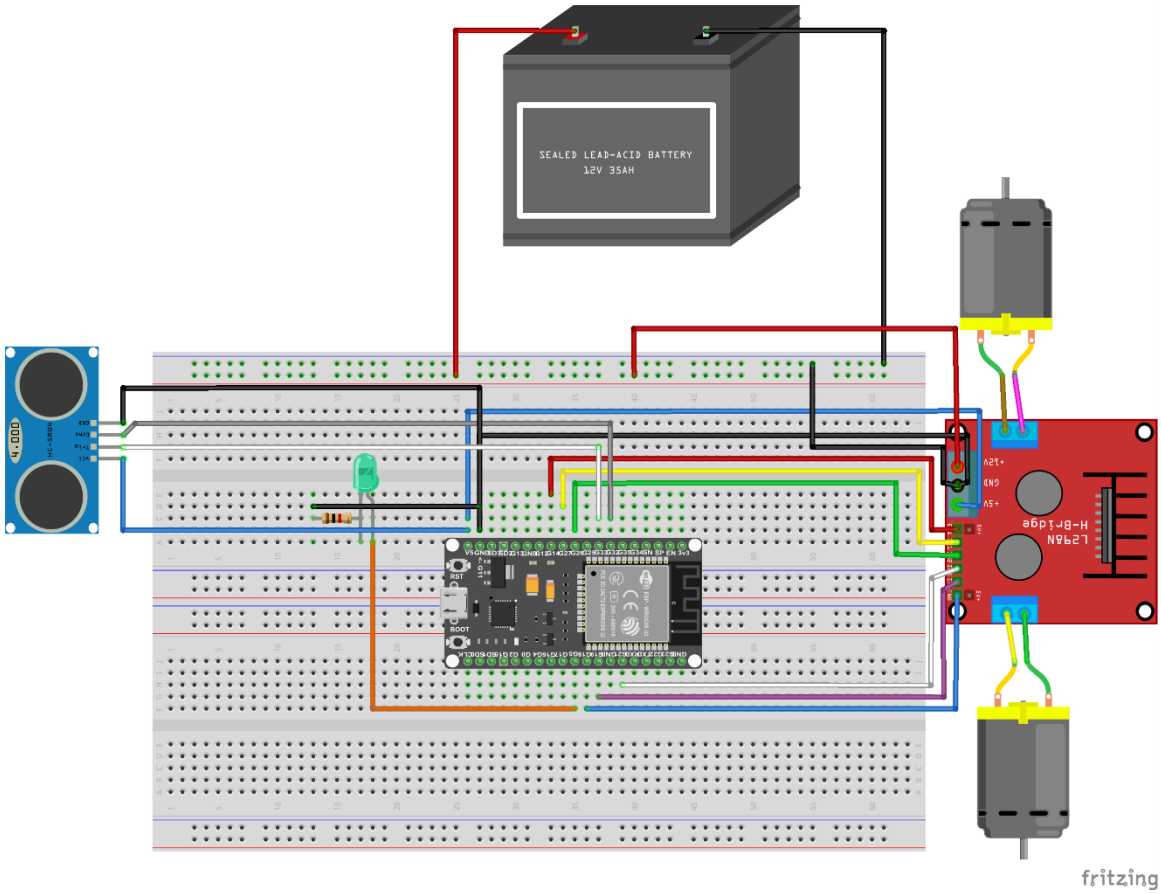
MICROPROCESSOR

• Aim of the hardware project:

- To construct a prototype for Surveillance Robot (which has 2 modes of operation), controlled by a mobile application.
- Practical implementation for
  - Microprocessors
  - C++ in Hardware
  - Robotics



- **Schematic of the circuit:**

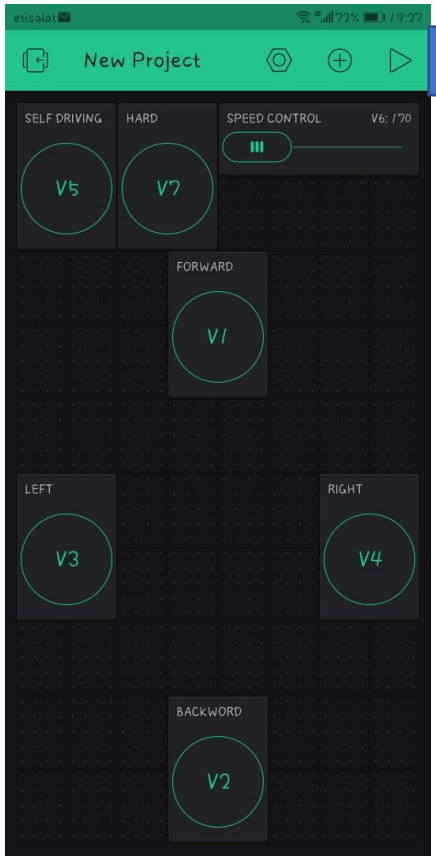


- **Procedure to use this circuit :**

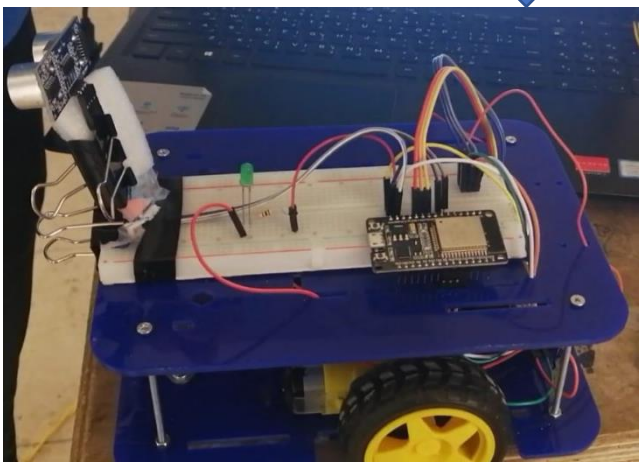
The robot can be controlled by a mobile application with two modes:

1- Self driven mode : in this mode the robot moves straight forward and when the sensor detects an object in front of the robot , the robot will stop , go back then turn right.
This is done by pressing the V5 button.

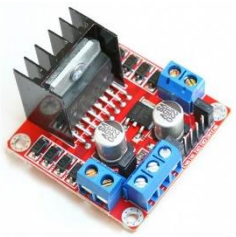
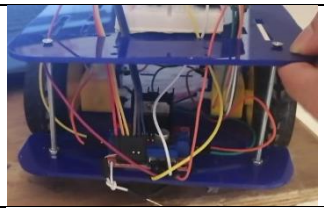
2- Controlled mode : in this mode we control the movement of the robot by four buttons in the APP. V1,V2,V3,V4 are the buttons for Forward, Backward, Left, and Right, respectively.
V7 when pressed it allows the functionality of rotating either to the right when pressing V4 or to the left when pressing the V3.



Instructions through hotspot

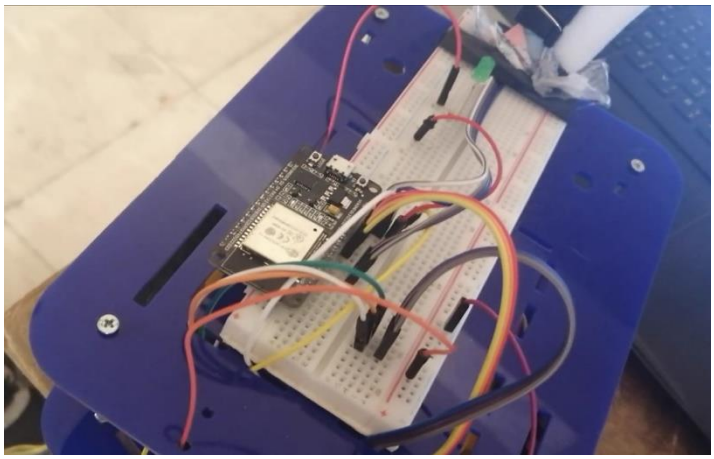


- **Components and Budget:**

Component	Photo	Budget
Esp32		200 LE
motor driver L298N		55LE
Ultrasonic sensor HC SR04		35LE
2 motors		80LE
Robot base		80LE
Jumper wires		0.25LE
Breadboard		25LE
Led		0.25LE
Power supply 300W		100LE
Total Cost		575.5LE

Challenges:

- **Control the robot via phone**: Solved by using blynk application.
- **Self-driven mode:** we faced a problem when the ultrasonic sensor detects an object with distance <= 23, we initially wanted the robot to rotate right, but it did not because the distance was still <=23. So, we made it go Backward for one second, so the distance is >23 and then turn right.
- **The robot initially was able to only rotate by pivoting on one wheel** (only one motor was working while the other was off). And to solve this, we used the analogWrite library to make the robot drift (turning while going forward). Then we made the two options available via the control button V7.
- **Starting speed:** the robot was not able to move at the initial speed we assigned. We solved this through trial and error, as we tried different initial speeds until the robot started moving comfortably at a duty cycle of 170, overcoming friction.
- **Circuit construction:** The esp32 kit cannot fit ion the breadboard from both sides, so we could not connect all the pins directly. We solved this by changing the wire connections as shown in the picture.



- **Controlling the wheel movements:** Every motor is controlled through one enable pin and 2 others. Those 2 pins control the movement of each wheel, and we had to understand the pattern through trial and error so the robot can move correctly according to the instruction given.

- **While testing the robot:** we initially used 3 batteries that died out in a couple of minutes. We solved this by using an external 300W power supply and wires.

Links:

Code link on github:

https://github.com/eslamdyab21/BI-Model-Robot

Video link on Google Drive:

Microprocessors Hardware Project.mp4 - Google Drive

References:

- Getting Started with the ESP32
- Getting Started with the ESP32 - Using the Arduino IDE (dronebotworkshop.com)
- https://ieeexplore.ieee.org/abstract/document/8673463

- Blynk Documentation
- https://docs.blynk.cc

- Motor Driver
- How to use the L298N Motor Driver - Arduino Project Hub

- ESP32 PWM with Arduino IDE (Analog Output)
- ESP32 PWM with Arduino IDE (Analog Output) | Random Nerd Tutorials

- Ultrasonic Range Detector
- Ultrasonic Range Detector Using Arduino and SR-04F - Arduino Project Hub
- https://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032189/meta