**VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**



**DESIGNING WIRELESS-EMBEDDED SYSTEM**

**MID-TERM REPORT**

**REMOTE-CONTROLLED ROBOT USING IOT**

**Instructor: PhD Đoàn Duy**

**Student:** Nguyễn Thành Tài - 21522568

CLASS: CE232.O21. MTCL

*HO CHI MINH CITY, March 2024*

**Tables of Contents:**

1. Introduction……...……………………………………………………2
2. Hardware System……………….………………………….…………4
3. Software ………………………………………………………………8
4. Implementation …….…………………………………………………9
5. Result ………...…………….………………………………………..21

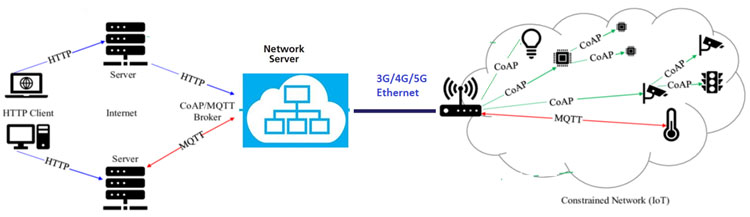
**I. Introduction**

* In the age of ever-evolving technology, the convergence of robotics and Internet of Things (IoT) has paved the way for innovative solutions in various domains. One such groundbreaking endeavor is the development of a remote-controlled camera robot, leveraging the power of IoT to revolutionize surveillance, exploration, and remote monitoring applications.



A Surveillance Robot

* The “Remote-Controlled Camera Robot Using IoT” project aims to address the need for flexible and efficient surveillance systems that can adapt to diverse environments and scenarios, empowering users with remote control and real-time monitoring capabilities.
* By integrating IoT principles into the design and functionality of the camera robot, this project offers unparalleled convenience and accessibility. Users can remotely manipulate the robot's movement and camera orientation through a user-friendly interface, accessible via smartphones, tablets, or computers.
* Key features of the Remote-Controlled Robot Using IoT project include:
* Wireless Connectivity: Leveraging IoT protocols for wireless communication, enabling remote control and data transmission over the internet.
* Pan-and-Tilt Camera Mechanism: Equipped with a versatile camera system capable of pan-and-tilt movements for comprehensive coverage and surveillance.
* Real-Time Monitoring: Providing users with live video feeds and sensor data, ensuring real-time monitoring and analysis of remote environments.



**IoT Wireless Network Protocols**

* In summary, the "Remote-Controlled Robot Using IoT" project represents a significant advancement in the realm of remote surveillance and exploration. By harnessing the synergy between robotics and IoT, this project sets a new standard for efficiency, accessibility, and functionality in remote-controlled camera systems, with promising implications across various industries and applications.

**II. Hardware System**

1. **NodeMCU ESP32 Module**

- NodeMCU ESP32 Kit is a Wi-Fi and Bluetooth transceiver KIT based on the ESP32 Wi-Fi SoC chip and the powerful CP2102 communication chip. Used for applications that need to connect, collect data and control via Wi-Fi, via Bluetooth, especially applications related to IoT. Using the latest ESP32-WROOM-32 Module with Dual core. With the easy-to-use design of the Arduino IDE's direct compiler for browser installation and code downloading, this makes using and installing applications via Wi-Fi and Bluetooth on the ESP32 very simple.

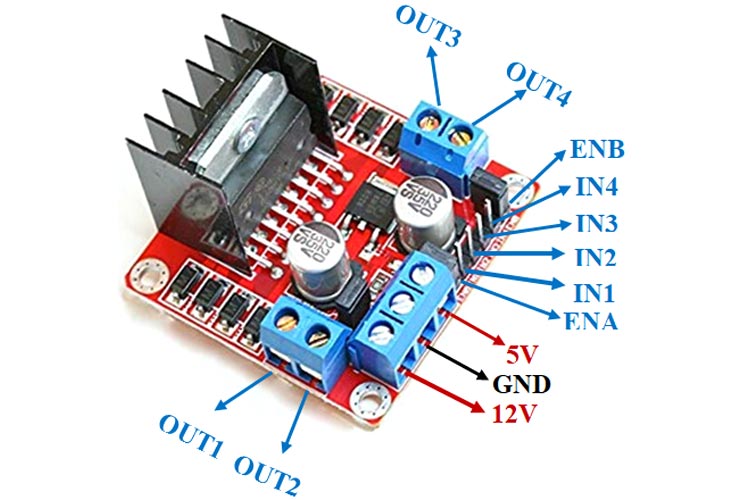
A black circuit board with a silver chip

Description automatically generated

**NodeMCU ESP32 Module**

1. **L298N Motor Driver Module**

* L298 DC Motor Driver motor control circuit can control 2 DC motors, maximum current of 2A per motor, integrated protection diode circuit and 7805 power IC to help supply 5VDC power to other modules.



***L298N Motor Driver Module***

A screenshot of a computer

Description automatically generated

1. **DC Geared Motor**

* DC Gear motor is also called DC Geared Motor. It consists of an electric DC motor and a gearbox or gearhead; these gearheads are used to reduce the DC motor speed, while increasing the DC motor torque.

A yellow and silver electric motor

Description automatically generated

**DC Geared motor**

* Operating voltage: 3V – 12V
* Maximum Torque: 800g/cm at 3V

1. **LED**

* We will use an additional LED in the robot to be able to use it for lighting in the dark.

A clear led with two thin sticks

Description automatically generated

**LED 5mm**

* Operating Voltage: 2.8 – 3.2 V
* Current: 5mA – 40mA

1. **Servo SG90 Motor**

* The servo mounted on the front of the robot will help in the future.

A blue plastic device with a white valve

Description automatically generated

**SG90 Servo Motor**

* Pin configuration for the servo:
* Red wire (VCC): Connect to a 5V pin.
* Black wire (GND): Connect to a GND pin on the ESP32.
* Yellow wire (signal): Connect to 3.3V signal or 5V signal.

1. **DC LM2596 Module**

* LM2596 module is used to control the servo motor (operates with 5V voltage)

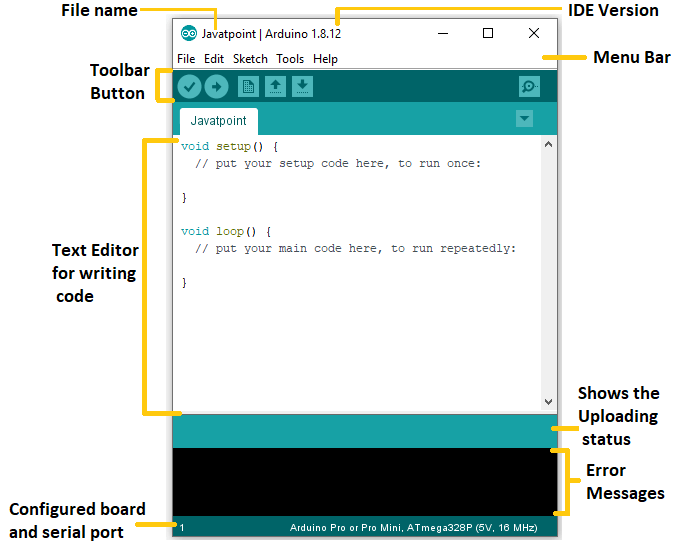
A blue circuit board with a digital display

Description automatically generated

* Input voltage: 3V – 30V.
* Output voltage: 1.5V – 30V.
* Maximum current: 3A
* Power: 15W

1. **Software**
2. **Arduino IDE**

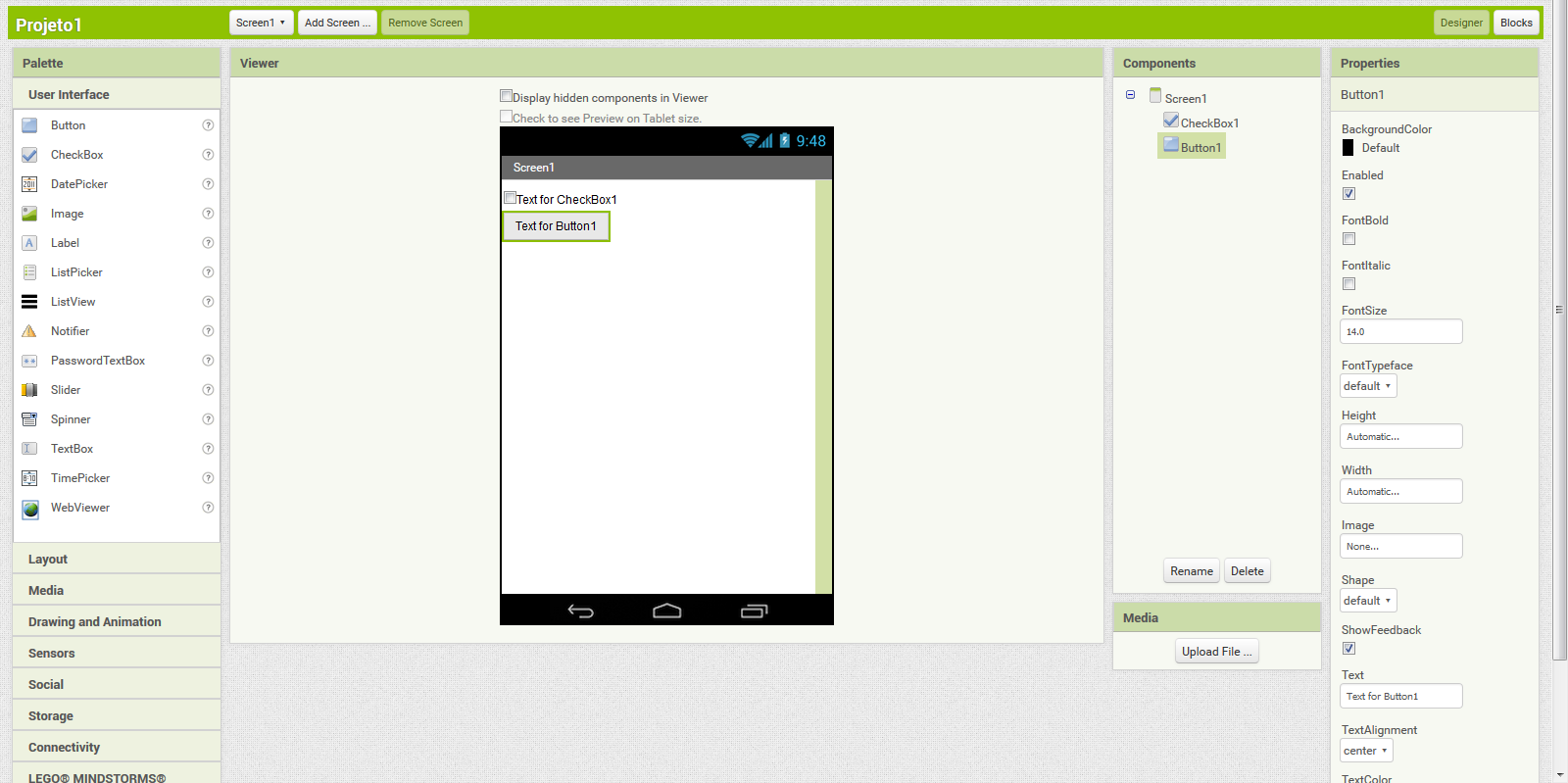
* The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.



**Overview of Arduino IDE**

1. **MIT App Inventor**

* MIT App Inventor (App Inventor or MIT AI2) is a high-level block-based visual programming language, originally built by Google and now maintained by the Massachusetts Institute of Technology. It allows newcomers to create computer applications for two operating systems: Android and iOS.



**Overview of MIT App Inventor**

1. **Implementation**
2. **Project Diagram:**

A diagram of a wireless connection

Description automatically generated

* Users can use their smartphone, computer, … to control the ROBOT by CONTROL SIGNAL through WIRELESS CONNECTION.
* ROBOT will receive the CONTROL SIGNAL which sent through WIRELESS CONNECTION and operate activities from CONTROL SIGNAL
* The STATUS OF ROBOT (CAMERA, …) will be sent back to USER to help they can track the ROBOT.
* In this project, we will use HTTP Protocol with ESP32 Access Point Mode to make a WIRELESS CONNECTION between the ROBOT and USER DEVICES. And the USER will use a DEVICE APP which we will create to control the ROBOT.

1. **Robot Circuit:**

The table below shows connections between ESP32 GPIOs to other components of the ROBOT:

|  |  |
| --- | --- |
| ENA (DC Motor Driver pin) | ESP32 GPIO 13 |
| ENB (DC Motor Driver pin) | ESP32 GPIO 14 |
| IN1 (DC Motor Driver pin) | ESP32 GPIO 15 |
| IN2 (DC Motor Driver pin) | ESP32 GPIO 16 |
| IN3 (DC Motor Driver pin) | ESP32 GPIO 17 |
| IN4 (DC Motor Driver pin) | ESP32 GPIO 18 |
| LED pin | ESP32 GPIO 4 |
| Servo Motor pin | ESP32 GPIO 19 |

A diagram of a circuit board

Description automatically generated

**Robot Circuit**

* We will use 3 18650 Li-on batteries to power the DC Motor Driver Module L298N and use 5V Supply Power pin of DC Motor Driver Module L298N to power the ESP32.
* The robot will have 4 motors, each of the 2 motors will be placed at the left and the right side of the robot.
* The robot also has an LED which is placed in the front of the robot.
* We also use the LM2596 module to provide 5V power to the servo motor.

1. **Software Implementation:**
   1. **Network model.**

A diagram of a computer process

Description automatically generated

**Network connection between Robot as Server and Control Device as Client**

* 1. **Robot Software Implementation**

As mentioned above, we will use Arduino IDE to code the robot.

* + 1. **Define pins.**

**A number and numbers on a white background

Description automatically generated with medium confidence**

* + 1. **Initialization ESP32 Wi-Fi mode.**

We use ESP32 Access Point mode with HTTP protocol to make a Wi-Fi connection between user devices and the robot.

**A computer code with text

Description automatically generated with medium confidence**

The ESP32 will have an access point whose name is “ESP Wi-Fi” and password is “12345678”.

* + 1. **Set up the robot.**

In this setup() function, we will set up all the components of the robot (Motor Driver, LED, Servo Motor, ESP32 Wi-Fi).

**A computer code on a white background

Description automatically generated**

* + 1. **Control robot’s motors.**

We use ESP32’s GPIOs to control the Motor Driver Module. The control signal of Motor Driver Module is shown in the table below.

**A screenshot of a computer

Description automatically generated**

Based on the signal table, we have built 5 main functions for the robot's motors including GO STRAIGHT, GO BACK, TURN LEFT, TURN RIGHT and STOP. ENA pin and ENB pin will be used to adjust the speed of the motors.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

* + 1. **Handle Root**

- This function checks if the URL that the client requested from the Server has an argument "State" or not, and the Server will always send code 200 to the client.

**A screen shot of a computer code

Description automatically generated**

* + 1. **Control robot based on data.**

- This function will operate the robot based on the data received from the application (client).

**A white text with blue and black letters

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

* + 1. **Loop**

- The ESP32 always handles clients. And data variable will take the value of argument “State” which server received from client. And the robot will operate continuously according to the data variable.

**A computer code with text

Description automatically generated**

* 1. **Robot Control App Implementation**
     1. **Design the application’s interface.**

- We use 4 buttons to control the motors of the robot, 3 buttons to control the servo motor angle and use 1 switch to control the LED of the robot.

**A screen shot of a phone

Description automatically generated**

* + 1. **Design the backend of the application.**

- For buttons, every time a button is pressed (Touch Down or Touch Up), the application will request objects in the Web Server using the GET method on the URL path on the Server (currently ESP32).

For example: When we press the "=>" button (Turn Right), the application will request the object using the GET method from the URL "192.168.4.1/?State=right", then when the Server (ESP32) receives the request from the application, it will control the robot to turn right

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

- Same for Switch, if switch is on, the LED of the robot will turn on and if switch is off, the LED of the robot will turn off.

**A screenshot of a computer program

Description automatically generated**

1. **Hardware Implementation**

Based on the construction circuit diagram, we have built a realistic robot model.

**A white box with wires and wires

Description automatically generated**

**A machine with wires and wires

Description automatically generated**

1. **Result**
2. **Achievement**

- After assembling the robot and installing the control application on the phone, we connected to the Wi-Fi network created from the ESP32, we were able to control the robot through the app.

- Here is a video below about our achievement we achieved with our project:

<https://drive.google.com/file/d/1EJGGFPsUwXPEVzB8Adq72gkce85Lyp76/view?usp=drive_link>

- The robot's operating range is the ESP32's Wi-Fi operating distance (about a few dozen to a few hundred meters depending on the environment) from the robot to the control device.

1. **Difficulties**

- Our limited knowledge of HTML/CSS/JavaScript, which is used to set up a personal website for us to control the robot. That’s why we must use the MIT App Inventory application for ease to create a robot control application.

- Because our project mentions the use and streaming of cameras based on the ESP32 CAM module, but because we have not found it yet and are still trying to implement camera streaming on the MIT App Inventory.

1. **Discussion**

- Because this is a mid-term report, we still need to improve a lot on our project. So below are some project goals that we hope to achieve in the final report:

* + Possible to use an additional ESP32 CAM module placed right in front of the robot to stream the camera to the user.
  + Using the MIT App Inventor application to control the robot can be replaced with a personal website that we program ourselves to control the robot.