BRAC UNIVERSITY

Inspiring Excellence

CSE360 Lab Project Report

Project Title: Surveillance Robot

Group Members:
1. Alif Rahman - 21201566
2. Mehedi Hasan -21101062
3. Ahmed Sayef-21201220
4. Mohammod Tahseen Mansoor-20101280

Introduction

The goal of this project is to create a surveillance robot capable of monitoring it's surroundings and seamlessly transmitting real time video data wirelessly to the user. The user is able to control the movement as well as the peripheral vision of the robot through the shifting of it's video capture unit. Furthermore, we have implemented a light source for illumination and clear video capture during nighttime. Lastly, the robot has a sensor capable of measuring the environmental temperature and humidity. After measuring, it sends live update to the user using bluetooth data transmission. By utilizing these features, the robot is able to monitor remote or inaccessible locations while being lightweight and power efficient.

Component List

1. **Arduino Uno Microcontroller**
2. **ESP32-CAM Module:** 2 MP OV2640 camera with an ESP32 SoC. Wi-Fi, GPIOs and PSRAM support.
3. **SG90 Servo Motors x2**
4. **L298N Dual H-Bridge Driver**
5. **DC Gear Motors x4 (RC Motors)**
6. **Power Subsystem:** Four 3.7 V Li-ion cells. Pairs are in series (7.4 V nominal) feeding an UBEC/ buck converter, stepping down to regulated 5 V (ESP32-CAM, Arduino) and 3.3 V (camera logic).
7. **DHT11 Temperature and Humidity Sensor**
8. **HC-05 Bluetooth Module**
9. **Structural Components:** Chassis, Wheels, Switches, LEDs and Wires.

Circuit Diagram

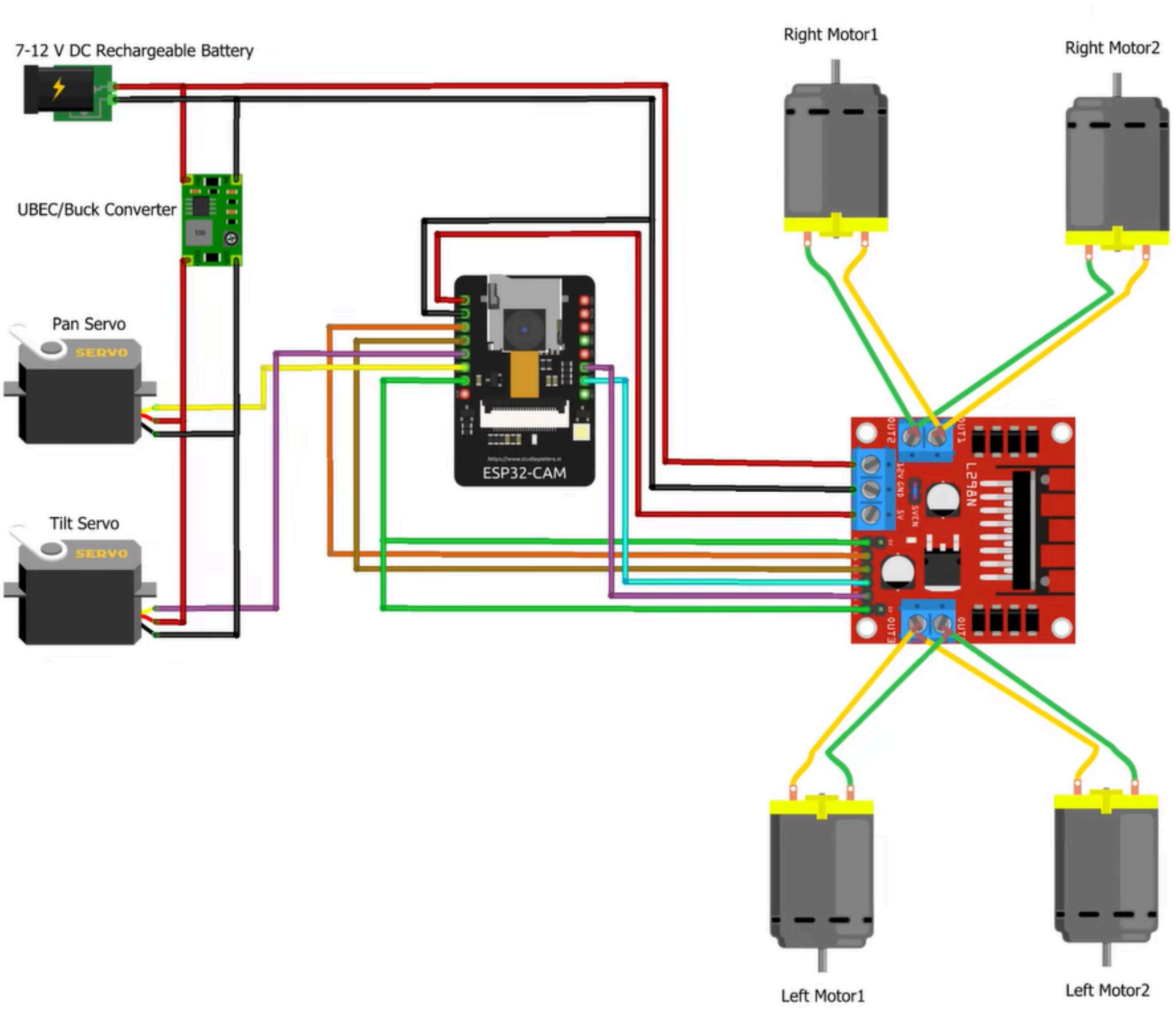

Figure-1: ESP32-CAM Module Connection with Navigation and Camera System

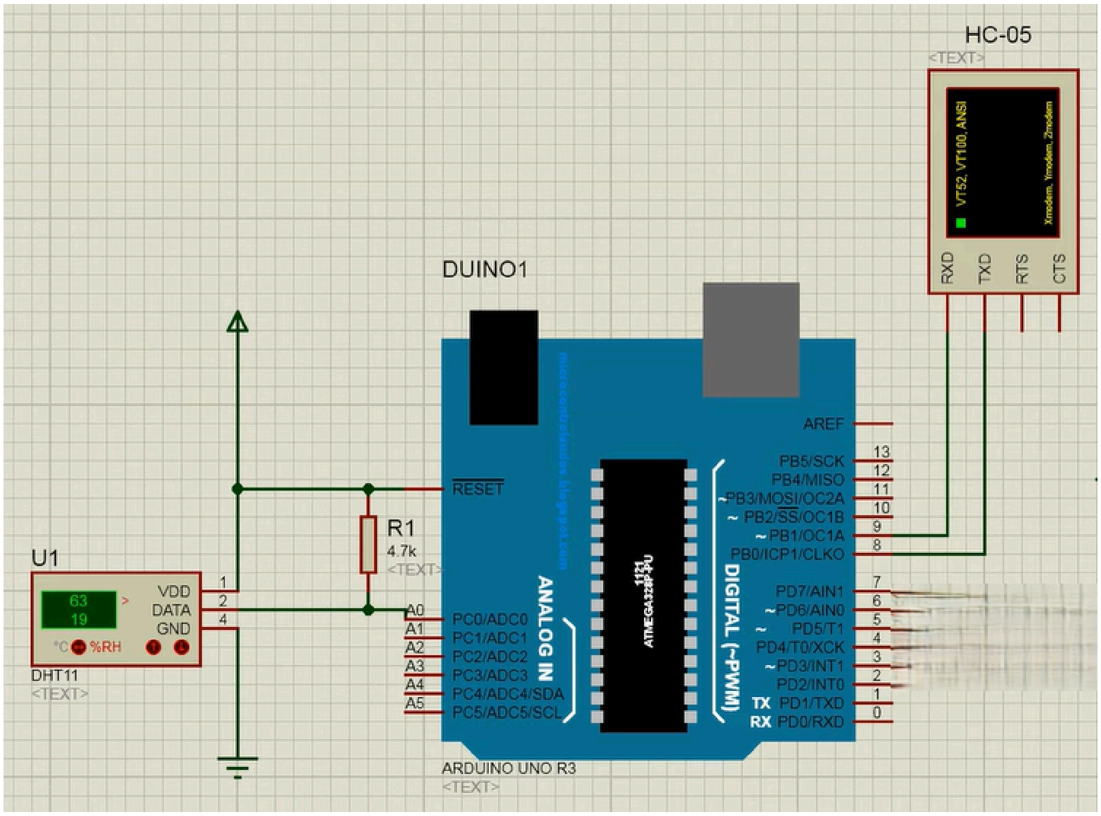

Figure-3: Arduino Uno Connection with DHT11 Temperature and Humidity Sensor and HC-05 Bluetooth Module

Hardware Configuration

1. Configure the power and level interface by wiring the servos, ESP32-CAM, L298N, and UBEC (buck converter) according to Figure 1 & 2.

2. Set Serial.begin(115200), map TXD0/RXD0, and upload the Arduino sketch to the ESP32-CAM using the Uno functioning as a USB-TTL serial adapter to configure the Uno-ESP32 firmware.

3. Launch the web server by using WiFi to establish a Wi-Fi access point in the setup() function.SoftAP (password, SSID). Launch an AsyncWebServer and configure AsyncWebSocket events for control commands and video streaming.

4. Control servo pan and tilt on GPIO pins 14 and 15 which can be managed with the ESP32Servo library. To control the motor and the onboard LED, set up LEDC PWM channels with an 8-bit resolution and a 1 kHz frequency.

5. Use ESP32-CAM to serve an HTML page with a <canvas> for video display and <input type='range'> sliders for movement control. Send control values (key-value pairs) to the ESP32 using WebSocket.send().

6. Initialize UART for HC-05 communication at 9600 baud rate. DHT11 is set to periodically read temperature and humidity which is then transmitted through Bluetooth as a string ("temp;humidity;") to mobile app.

Implemented Communication Protocols

1. UART : Used between Arduino Uno and Bluetooth module.

2. Wi-Fi (SoftAP) : Used by ESP32-CAM to Host a webpage/access point.

3. WebSocket : HTML interface for camera and movement control.

Code Implementation

1. Navigation and Camera Control Psuedo-code:

```
// Import required libraries

IMPORT ESP Camera, WiFi, AsyncWebServer, Servo, etc.

// Define constants for pins, directions, channels, and Wi-Fi credentials

// Define HTML interface elements (live video, directional controls, sliders)

// Motor control

FUNCTION rotateMotor(motorNumber, direction):

  sets motor direction based on input (FORWARD, BACKWARD, STOP)

FUNCTION moveCar(direction):

  calls rotateMotor() to move car in specified direction (UP, DOWN, LEFT, RIGHT, STOP)

// WebSocket control handlers

FUNCTION onCarInputWebSocketEvent(server, client, eventType, data):

  handles car control, speed, light, and servo based on received WebSocket data

FUNCTION onCameraWebSocketEvent(server, client, eventType, data):

  manages camera WebSocket connection for video streaming

// Camera setup and streaming

FUNCTION setupCamera():

  initializes camera with necessary pins and settings

FUNCTION sendCameraPicture():
  captures a camera frame and sends it to connected WebSocket clients

// Hardware setup

FUNCTION setUpPinModes():

  initializes servos, motor pins, light pin, and configures PWM channels

// Main setup function

FUNCTION setup():

  initializes hardware, sets up WiFi access point, configures web server, WebSocket, and camera

// Main loop

FUNCTION loop():

  handles WebSocket cleanup and sends camera stream to clients
```

2. Weather Sensing Unit Complete Code:

```
#include <SoftwareSerial.h>
SoftwareSerial bt(8, 9); // RX, TX

#include "dht.h"
#define dataPin A0
dht DHT;

int temp;
int hum;

void setup() {

 Serial.begin(9600);
 bt.begin(9600);
 Serial.println("Ready");
}

void loop(){
  int readData = DHT.read11(dataPin);

  hum = DHT.humidity;
  temp = DHT.temperature;

 bt.print(temp); //send distance to MIT App
 bt.print(";");
 bt.print(hum); //send distance to MIT App
 bt.println(";");

  delay(1000);
}
```

Future Improvements

1. Implement a water tank and hydropump on the robot which could be controlled by tilting and panning just as the camera. Therefore, pinpoint a fire hazard and remotely take care of it.
2. Implement autonomous obstacle avoidance and movement using computer vision of ESP32 or external microprocessor.
3. In-built storage system for archiving camera recording data.