T.C.
Ankara University
Faculty of Engineering
Electrical and Electronics Engineering

Embedded Systems Project

Ali Ekber Yücel
20290512

<u>CONTENTS:</u>

1. Abstract:

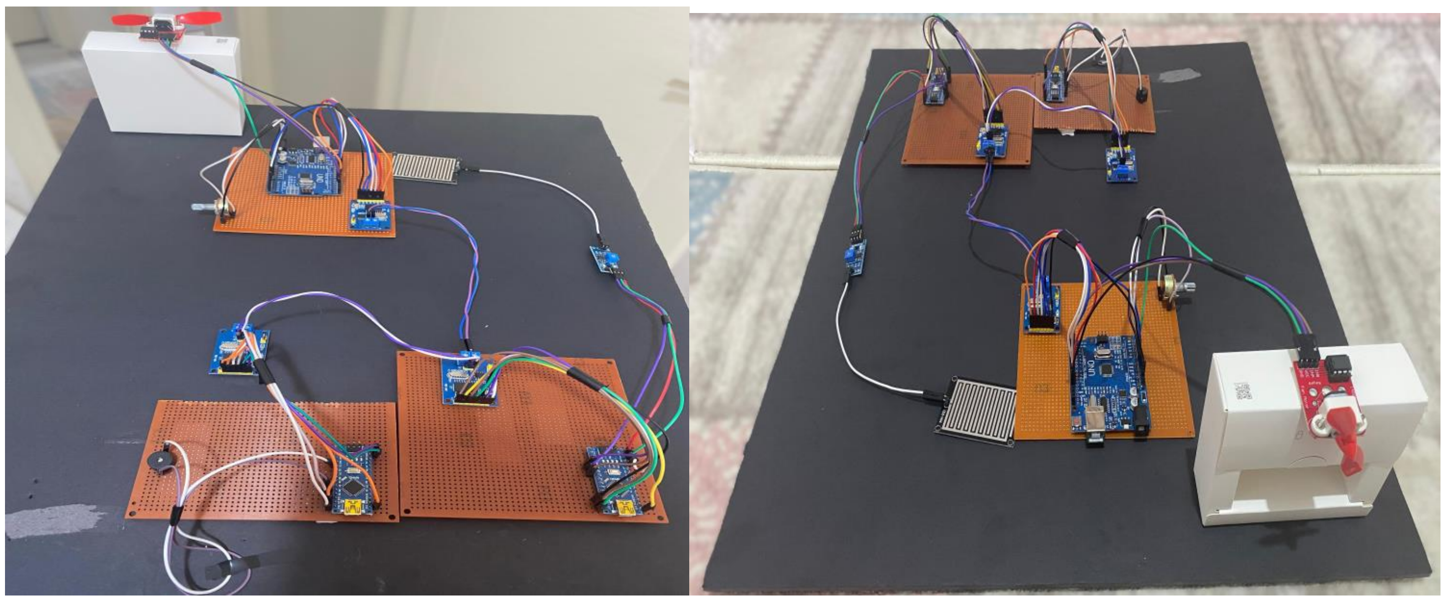
This project involves the design and implementation of a rain detection system using CAN bus communication to control a motor and a buzzer. The system reads rain sensor data and adjusts the motor speed while activating the buzzer when rain is detected. The primary goal is to demonstrate the integration of sensors and actuators using an embedded system with CAN bus communication.

2. Introduction:

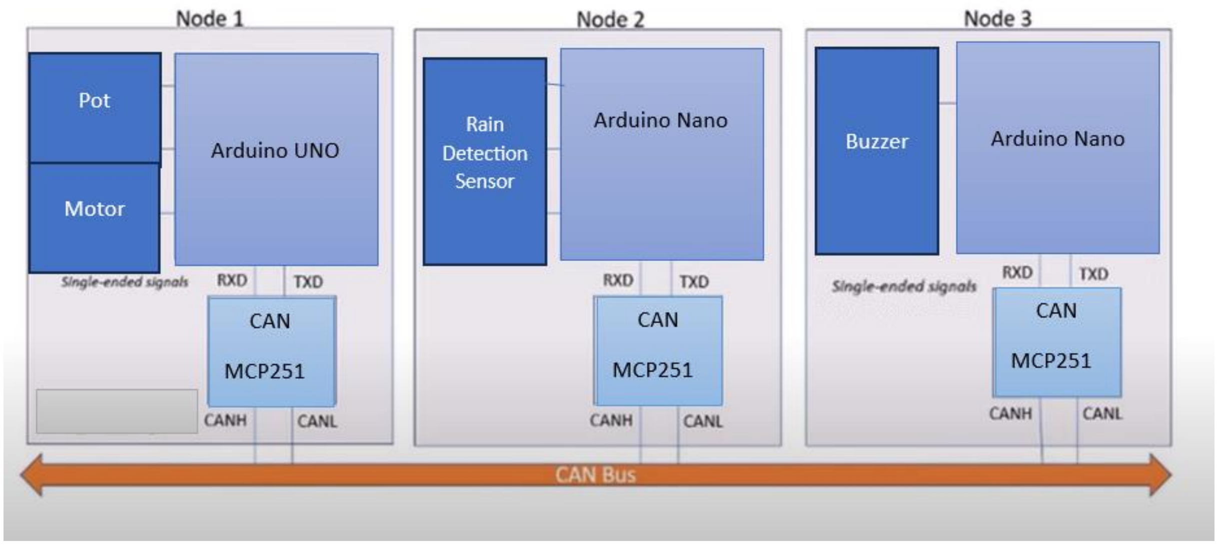
The main objective of this project is to design and implement an embedded system that can adjust the speed of an L9110 motor based on rain sensing and potentiometer input and utilizes CAN bus communication for efficient data transmission. The system includes a rain sensor, potentiometer, L9110 motor driver, MCP2515 CAN bus module and a buzzer, all controlled by Arduino microcontrollers. The basic functionality includes dynamically controlling the motor speed via a potentiometer and, if rain is detected, limiting the motor speed if it is above a certain value. Simultaneously, the system sends a CAN bus message to activate an audible warning indicating that rain has been detected. This project demonstrates the application of embedded systems in building responsive and interactive control systems with reliable communication protocols.

3. Project Design:

The system comprises a rain sensor, MCP2515 CAN bus modules, L9110 motor driver and fan, a buzzer, a potantiometer and Arduino microcontrollers. The rain sensor detects rain and sends data via the CAN bus module to the microcontroller, which activates the buzzer adjusts the motor speed and adjusts the motor speed if necessary.



3.1 The Block Diagram of The Project



3.2 Arduino UNO:

Arduino UNO's Connections:

L9110 Motor	Arduino UNO
VCC	Vin
GND	GND
INA	D3
INB	D4

POT	Arduino UNO
VCC	5V
Signal Pin	A0
GND	GND

Arduino UNO	MCP2515
VCC	5V
GND	GND
CS	D10
SO	D12
SI	D11
SCK	D13
INT	D5

Arduino UNO’s Software Implementation:

```
#include <SPI.h>
#include <mcp2515.h>

struct can_frame canMsg;
MCP2515 mcp2515(10);

const int motorPin1 = 3;
const int motorPin2 = 4;
const int potPin = A0;
int motorSpeed = 0;
const float reductionFactor = 0.9;

void setup() {
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(potPin, INPUT); /
  Serial.begin(115200); /
  mcp2515.reset();
  mcp2515.setBitrate(CAN_500KBPS, MCP_8MHZ);
  mcp2515.setNormalMode();
}

void loop() {
  if (mcp2515.readMessage(&canMsg) == MCP2515::ERROR_OK) { // Check if a CAN message is
received successfully
    if (canMsg.can_id == 0x036) { // Check if the CAN message ID matches the expected ID
      int rainDetected = canMsg.data[0]; // Read the rain detection value from the CAN message
      if (rainDetected == HIGH) { // Check if rain is detected
        if (motorSpeed > 100) { // If motor speed is greater than 100
          motorSpeed *= reductionFactor; // Reduce the speed
        }
      } else {
        int potValue = analogRead(potPin); // Read the potentiometer value
        motorSpeed = map(potValue, 0, 1023, 0, 255); // Map the potentiometer value to 0-255 range
      }
      analogWrite(motorPin1, motorSpeed);
      analogWrite(motorPin2, 0);

      Serial.print("Rain Detected: ");
      Serial.println(rainDetected); // Print rain detection value to serial monitor
      Serial.print("Motor Speed: ");
      Serial.println(motorSpeed); // Print motor speed value to serial monitor
    }
  }
  delay(10);
}
```

3.3 Arduino Nano:

Arduino Nano's Connections:

Rain Sensor	Arduino Nano
VCC	Vin
GND	GND
D0	D2

Arduino Nano	MCP2515
5V	VCC
GND	GND
D10	CS
D12	SO
D11	SI
D13	SCK
D4	INT

Arduino Nano's Software Implementation:

```
#include <SPI.h>
#include <mcp2515.h>

struct can_frame canMsg;
MCP2515 mcp2515(10);

const int sensorPin = 2;
const int intPin = 4;
int sensorValue = 0;

void setup() {
  pinMode(sensorPin, INPUT);
  pinMode(intPin, INPUT);
  Serial.begin(115200);
  mcp2515.reset();
  mcp2515.setBitrate(CAN_500KBPS, MCP_8MHZ);
  mcp2515.setNormalMode();
}

void loop() {
  sensorValue = digitalRead(sensorPin); // Read sensor value

  canMsg.can_id  = 0x036;    // Set message ID
  canMsg.can_dlc = 1;        // Set data length
  canMsg.data[0] = sensorValue == LOW ? 1 : 0;  // Rain sensor value (1: Rain
detected, 0: Not detected)

  mcp2515.sendMessage(&canMsg); // Send CAN message

  delay(500);
}
```

3.4 Arduino Nano 2:

Arduino Nano 2 Connections:

Buzzer	Arduino Nano
VCC	D8
GND	GND

Arduino Nano	MCP2515
Vin	VCC
GND	GND
D10	CS
D12	SO
D11	SI
D13	SCK

Arduino Nano 2 Software Implementation:

```
#include <SPI.h>
#include <mcp2515.h>

struct can_frame canMsg;
MCP2515 mcp2515(10);

const int buzzerPin = 8;

bool rainDetected = false;
unsigned long lastRainTime = 0;
const unsigned long rainDelay = 750;

int repeatCount = 0;

void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(115200);
  mcp2515.reset();
  mcp2515.setBitrate(CAN_500KBPS, MCP_8MHZ);
  mcp2515.setNormalMode();
}
```

```
void loop() {
  if (mcp2515.readMessage(&canMsg) == MCP2515::ERROR_OK) {
    if (canMsg.can_id == 0x036) {
      int rainDetectedVal = canMsg.data[0];
      if (rainDetectedVal == 1) {
        if (!rainDetected || millis() - lastRainTime > rainDelay) {
          // First rain detection or detection after a certain time
          digitalWrite(buzzerPin, HIGH);
          delay(1000);
          digitalWrite(buzzerPin, LOW);
          lastRainTime = millis(); // Update last rain time

          repeatCount++; // Increment counter

          if (repeatCount >= 5) { // Stop when counter reaches 5
            while (true) {
              if (mcp2515.readMessage(&canMsg) == MCP2515::ERROR_OK) {
                if (canMsg.can_id == 0x036) {
                  int newRainDetectedVal = canMsg.data[0];
                  if (newRainDetectedVal != 1) {
                    // No rain detected, reset counter
                    repeatCount = 0;
                    break;
                  }
                }
              }
            }
          }
        }
        rainDetected = true;
      } else {
        rainDetected = false;
      }
    }
  }
  delay(10);
}
```

4. Summary:

This project presents the development of a rain detection and motor speed control system using CAN bus communication. The system comprises a rain sensor, potentiometer, L9110 motor driver, MCP2515 CAN bus module, buzzer, and Arduino microcontroller. The motor speed is primarily controlled by a potentiometer. If the motor speed is above a specific value and rain is detected, the system limits the motor speed and sends a CAN bus message to activate a buzzer. The implementation involved configuring the MCP2515 for CAN bus communication, integrating the rain sensor and potentiometer for input, and programming the Arduino to adjust motor speed and activate the buzzer based on sensor data. Testing confirmed the system's ability to reliably detect rain, adjust motor speed, and communicate via CAN bus.

5. Conclusion:

The project successfully achieved its goal of creating an embedded system that dynamically adjusts motor speed based on potentiometer input and rain detection, with CAN bus communication enabling efficient data transmission and system integration. The system effectively limits motor speed when rain is detected and the speed is greater than a specific value while also activating a buzzer to indicate rain presence. This demonstrates the practical application of embedded systems in developing interactive and responsive control mechanisms. Overall, the project highlights the robustness and reliability of using CAN bus communication in embedded systems.