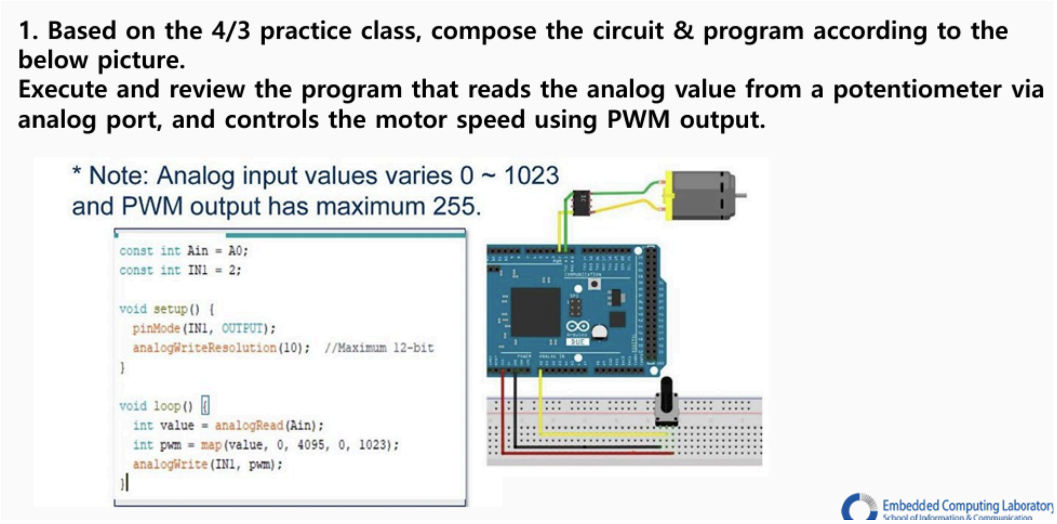
Class : 2025-1 Embedded System Design—BEU
Team Number : 11
Student ID : 12244853, 12244858
Name : Elton Satiyev, Ikram Mammadov
Submission Date : 2025-04-09

Homework1.

Problem



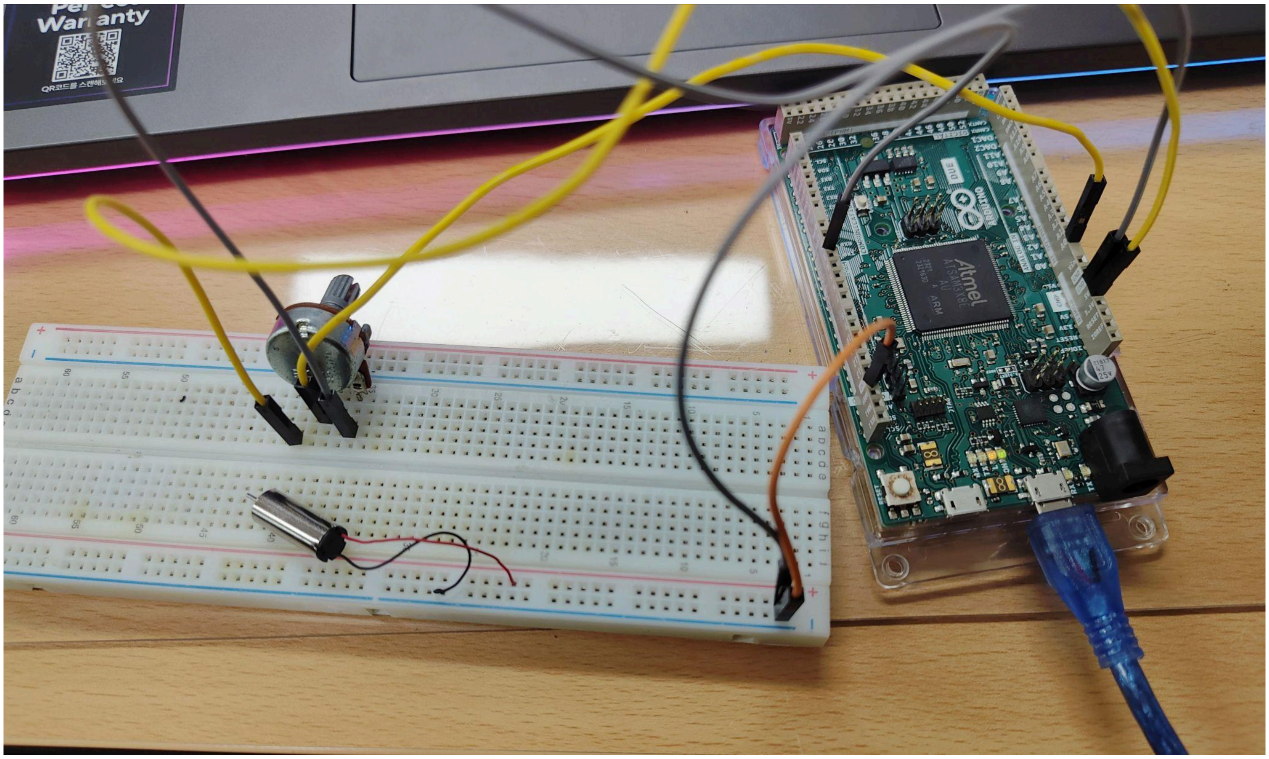
1. Based on the 4/3 practice class, compose the circuit & program according to the below picture.
Execute and review the program that reads the analog value from a potentiometer via analog port, and controls the motor speed using PWM output.

* Note: Analog input values varies 0 ~ 1023
and PWM output has maximum 255.

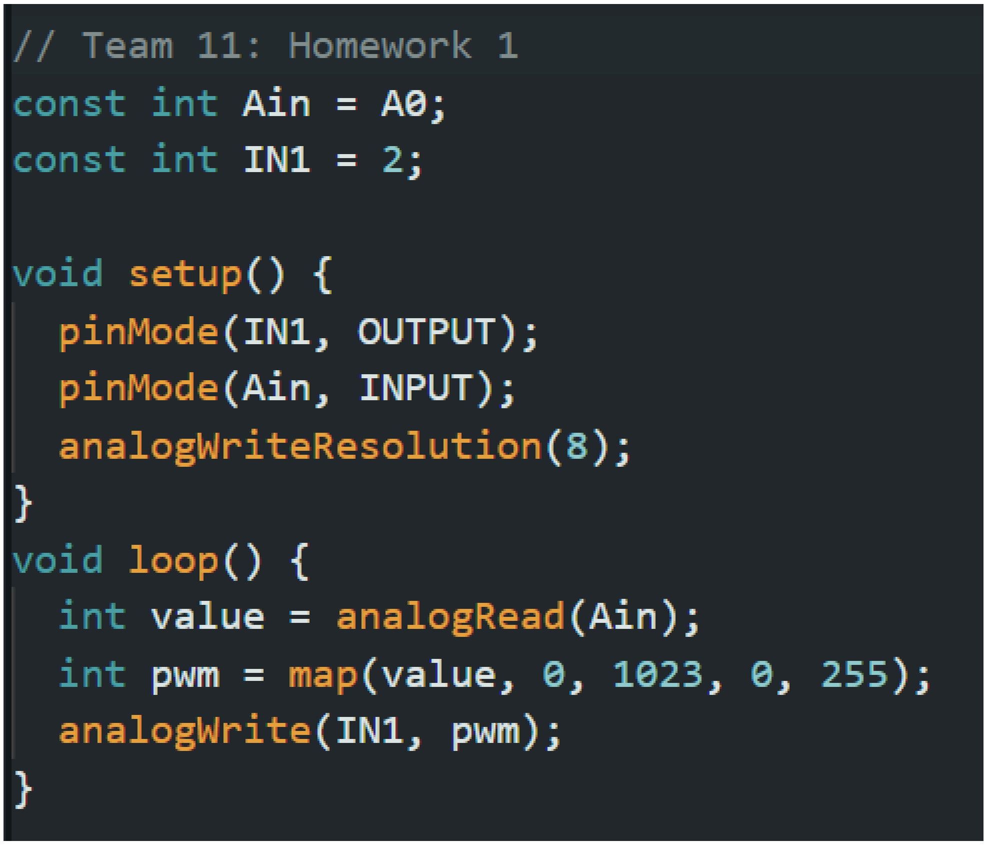
```
const int Ain = A0;
const int IN1 = 2;

void setup() {
  pinMode(IN1, OUTPUT);
  analogWriteResolution(10);  //Maximum 12-bit
}

void loop() {
  int value = analogRead(Ain);
  int pwm = map(value, 0, 4095, 0, 1023);
  analogWrite(IN1, pwm);
}
```

Embedded Computing Laboratory
School of Information & Communication

Circuit



Code



```
// Team 11: Homework 1
const int Ain = A0;
const int IN1 = 2;

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(Ain, INPUT);
  analogWriteResolution(8);
}
void loop() {
  int value = analogRead(Ain);
  int pwm = map(value, 0, 1023, 0, 255);
  analogWrite(IN1, pwm);
}
```

Result

The solution was completed and demonstrated live during the class session. The TA verified the functionality of the system, confirming that the potentiometer correctly controlled the motor speed through PWM.

Problem Explanation

This homework requires developing an Arduino program that reads the analog value from a potentiometer and controls the speed of a DC motor using PWM. The PWM signal must be scaled correctly based on the motor's required input range. The provided buggy code failed to match the resolution and mapping needed to drive the motor correctly.

Observations in the Buggy Code

- **Incorrect Mapping Range** – The analog input was mapped from `0–4095` to `0–1023`, which doesn't match the motor's expected input range of `0–255`.
- **Mismatched PWM Resolution** – `analogWriteResolution(10)` set the PWM output range to `0–1023`, exceeding the supported range of the DC motor.
- **Missing Pin Configuration** – `Ain` was not explicitly set as an `INPUT`, which, while optional, is good practice for clarity and consistency.
- **Motor Not Responding Properly**

Fixes in the New Code

- **Adjusted PWM Resolution** – Changed `analogWriteResolution(8)` to ensure the output PWM values are in the correct range (`0–255`).
- **Correct Mapping Logic** – Updated the `map()` function to convert the analog input from `0–1023` to `0–255`, matching the motor's requirements.
- **Defined Input Pin** – Added `pinMode(Ain, INPUT)` for completeness and better coding style.
- **Stable Motor Control** – With the corrected mapping and resolution, the motor's speed responded properly to changes in the potentiometer.

Key Points

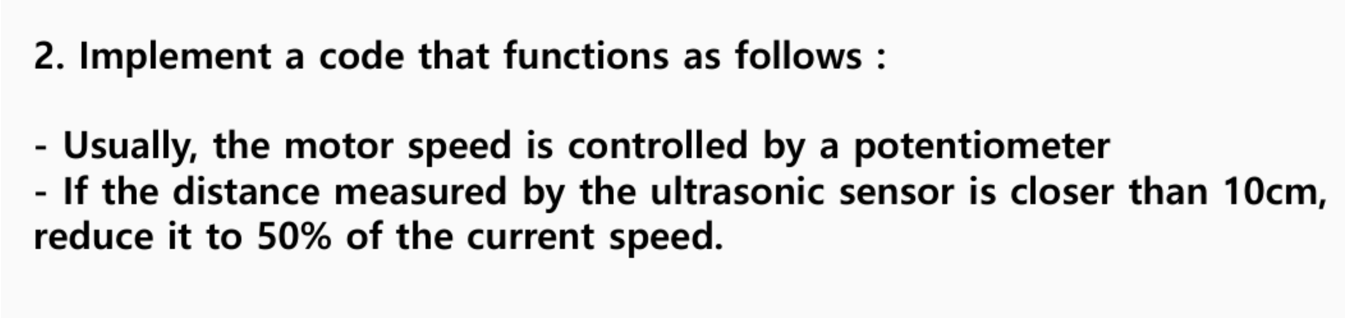
- **Resolution Matching** – Aligning the ADC input and PWM output resolution is critical, especially on Arduino Due, which supports 12-bit ADC and configurable PWM.
- **Component-Based Scaling** – The mapping should be based on the characteristics of the motor and potentiometer, not just default Arduino ranges.
- **Reliable Analog Control** – Ensuring that the `analogRead()` and `analogWrite()` ranges are consistent leads to stable and predictable hardware behavior.

Conclusion

By identifying the mismatch between input and output resolution and correcting the mapping logic, a functional and responsive motor control system was achieved. This task highlighted the importance of resolution awareness and correct scaling when working with analog inputs and PWM outputs in embedded systems.

Homework2.

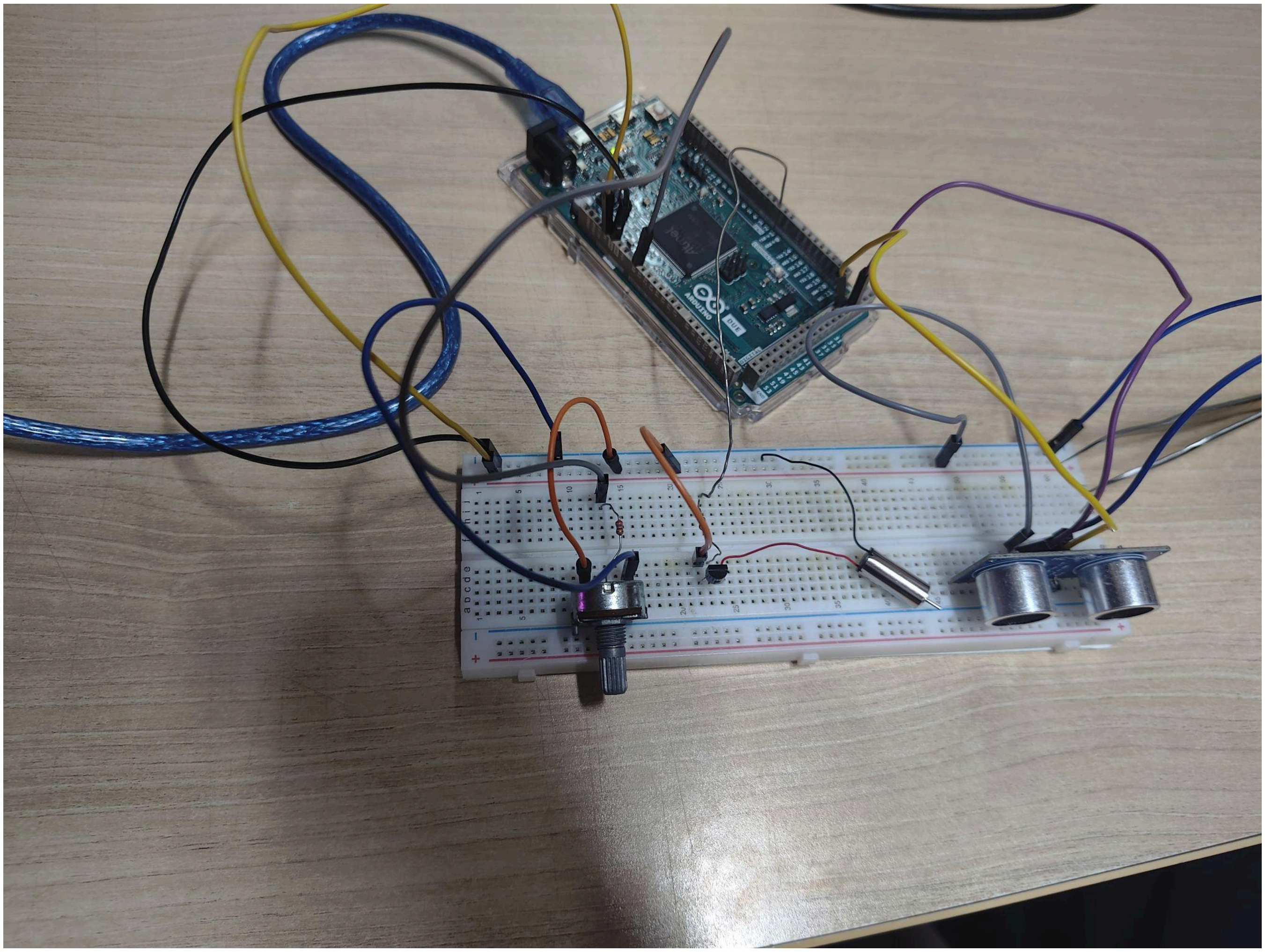
Problem



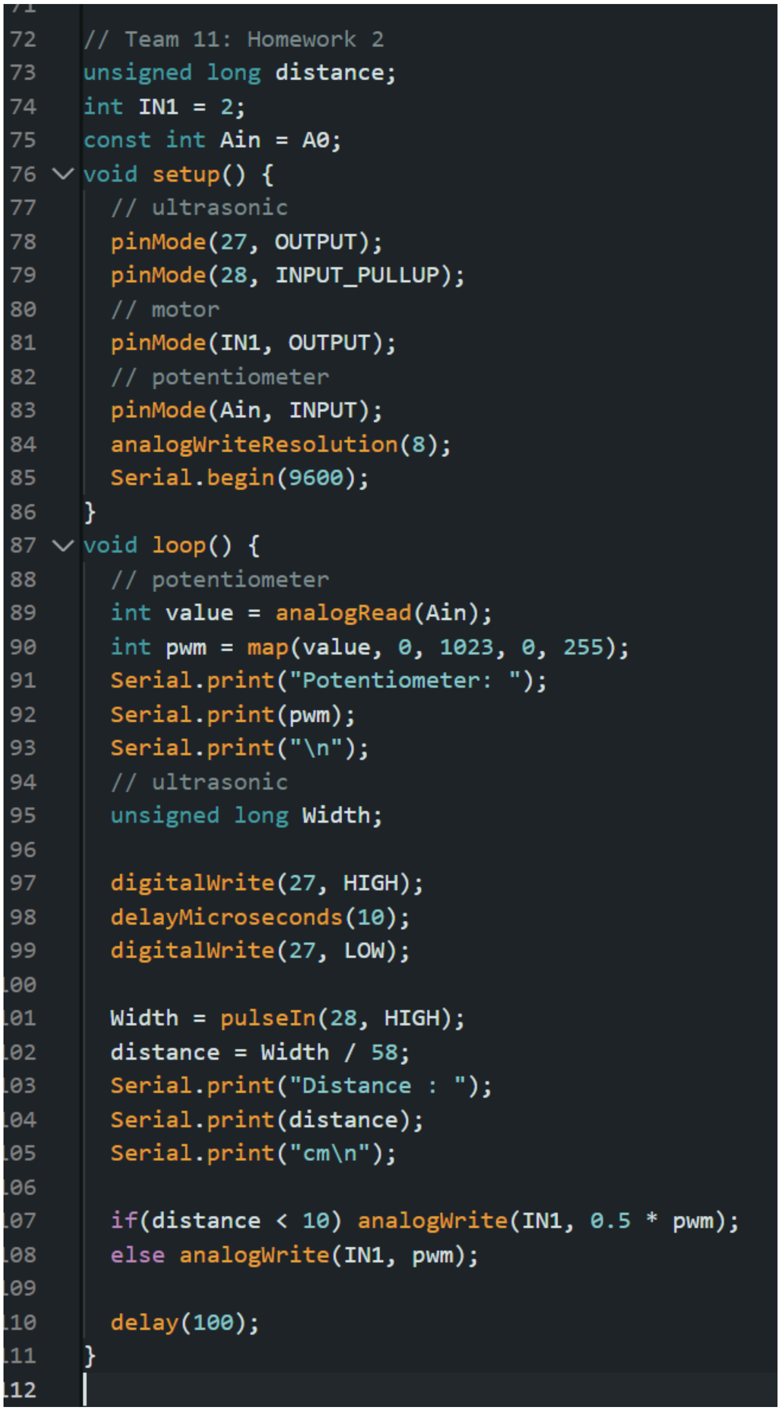
2. Implement a code that functions as follows :

- Usually, the motor speed is controlled by a potentiometer
- If the distance measured by the ultrasonic sensor is closer than 10cm, reduce it to 50% of the current speed.

Circuit



Code



```
// Team 11: Homework 2
unsigned long distance;
int IN1 = 2;
const int Ain = A0;
void setup() {
  // ultrasonic
  pinMode(27, OUTPUT);
  pinMode(28, INPUT_PULLUP);
  // motor
  pinMode(IN1, OUTPUT);
  // potentiometer
  pinMode(Ain, INPUT);
  analogWriteResolution(8);
  Serial.begin(9600);
}
void loop() {
  // potentiometer
  int value = analogRead(Ain);
  int pwm = map(value, 0, 1023, 0, 255);
  Serial.print("Potentiometer: ");
  Serial.print(pwm);
  Serial.print("\n");
  // ultrasonic
  unsigned long Width;

  digitalWrite(27, HIGH);
  delayMicroseconds(10);
  digitalWrite(27, LOW);

  Width = pulseIn(28, HIGH);
  distance = Width / 58;
  Serial.print("Distance : ");
  Serial.print(distance);
  Serial.print("cm\n");

  if(distance < 10) analogWrite(IN1, 0.5 * pwm);
  else analogWrite(IN1, pwm);

  delay(100);
}
```

Result

week5_hw2.mp4

HW2_DISCUSSION

Problem Explanation

This homework required developing an Arduino program to control the speed of a DC motor using a potentiometer and to reduce the speed by 50% when an object is detected closer than 10 cm by an ultrasonic sensor.

Solution Approach

- **Potentiometer-Based Speed Control –** Read analog values from the potentiometer and mapped them to the PWM range (0–255) to set the base speed of the motor.
- **Ultrasonic Distance Measurement** – Measured the distance using a standard trigger/echo method with `pulseIn()`, converting the pulse width to centimeters.
- **Dynamic Speed Adjustment** – Implemented a condition: if an object is within 10 cm, the motor speed is reduced to 50% of the potentiometer-defined value using `analogWrite(IN1, 0.5 * pwm)`; otherwise, full speed is applied.

Key Points

- **Sensor Fusion –** The code successfully integrates two inputs (potentiometer and ultrasonic) to influence a single actuator (DC motor).

Conclusion

By combining analog control from the potentiometer with obstacle-aware logic using the ultrasonic sensor, this implementation effectively simulates an intelligent motor system that adapts to its environment.