SMART EMBEDDED ROBOTICS E-LAB



Written By

MOHD KHAIRUL ANUWAR BIN MOHD KHAIRI
DR. NOR ASHIQEEN BINTI JAMALUD-DIN
MARIA BINTI MANSOR

SMART EMBEDDED ROBOTIC

E-LAB

MOHD KHAIRUL ANUWAR BIN MOHD KHAIRI
DR. NOR ASHIQEEN BINTI JAMALUD-DIN
MARIA BINTI MANSOR

First Publication: May 2026
©Copyright 2026

This ebook is the original work of
Mohd Khairul Anuwar bin Mohd Khairi
Dr. Nor Ashiqeen binti Jamalud-din
Maria binti Mansor

Editor
Mohd Khairul Anuwar bin Mohd Khairi
Dr. Nor Ashiqeen binti Jamalud-din

Graphic Designer
Dr. Nor Ashiqeen binti Jamalud-din
Maria binti Mansor

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior permission of author/s or publisher. The author also does not guarantee that the content is suitable for the reader, but all the content is through the author's own experience and expertise.

Published by:

Politeknik Tun Syed Nasir Syed Ismail,
Hab Pendidikan Tinggi Pagoh,
KM1 Jalan Panchor
84600 Panchor, Muar, Johor, Malaysia

Cataloguing-in-Publication Data

National Library of Malaysia

A catalogue record for this book is available from the National Library of Malaysia.

eISBN 978-967-2736-47-9

PREFACE

This eBook, From Sensors to Intelligent Mobile Robot: Smart Embedded Robotic e-LAB, is designed as a comprehensive hands-on guide to support students in understanding and developing embedded robotic systems through a structured laboratory-based approach. It integrates fundamental concepts with practical applications, enabling learners to progressively build their knowledge and skills from basic components to a complete functional robot.

The content of this module is organised into six practical works, beginning with sensors and actuators, followed by motor control and ultrasonic sensing. Students are then guided towards more advanced applications, including line follower robot development, wireless control using Bluetooth, and finally, robot vision systems. This step-by-step progression allows students to experience the full development cycle of a mobile robot, from sensing and actuation to control, communication, and perception.

This eBook has been developed to support teaching and learning activities, particularly for students in engineering and technical programmes. It emphasises experiential learning, encouraging students to engage actively in problem-solving, experimentation, and system integration. The ultimate goal is to equip learners with the necessary technical competencies and practical skills aligned with current industry demands in embedded systems and robotics.

It is hoped that this module will serve as a useful reference for both students and educators, and contribute towards enhancing the quality of engineering education through practical and application-oriented learning.

AUTHOR



Mohd Khairul Anuwar bin Mohd Khairi

is a Lecturer in Electrical and Telecommunication within the Department of Electrical Engineering at Politeknik Ibrahim Sultan. He earned his Bachelor's degree in Electrical Engineering from Kolej Universiti Teknologi Tun Hussein Onn and completed his Master's degree in Technical and Vocational Education in 2005. With 20 years of experience in teaching Electrical and Electronic Engineering, he brings a wealth of knowledge and expertise to his role.



Dr. Nor Ashiqeen binti Jamalud-din

is a Lecturer in Mechanical Engineering at Politeknik Tun Syed Nasir Syed Ismail. She obtained her Bachelor's degree in Mechanical Engineering from Kolej Universiti Teknologi Tun Hussein Onn and later pursued her doctoral studies in the mechanical engineering field. With 18 years of extensive experience in teaching and academic development, she specialises in areas such as Strength of Materials, mechanical systems, and engineering workshop practice. She is actively involved in developing practical-based learning modules, particularly in embedded systems and robotics applications, aimed at enhancing students' hands-on skills and problem-solving capabilities in line with current industry needs.



Maria bin Mansor

is a Lecturer in Electrical and Instrumentation within the Department of Petrochemical Engineering at Politeknik Tun Syed Nasir Syed Ismail. She earned her Bachelor's degree in Education (Electrical Engineering) from Open University Malaysia (OUM) and completed her Master's degree in Electrical Engineering (Electronic System) in 2013. With 28 years of experience in teaching Electrical and Electronic Engineering, she brings a wealth of knowledge and expertise to her role.

CONTENT

NO.	ITEMS	PAGE
1	Introduction	1
2	Practical Task Overview	3
3	Arduino UNO R3 Microcontroller	3
4	Hardware and Software Preparation	5
5	Hardware Specification	8
6	Smart Embedded Robotic e-Lab	8
7	Operational Procedure	9
8	Practical Work Series	10
	Practical Work 1: Fundamentals of Sensors and Actuators	11
	Practical Work 2: Mobile Robot Design (Motor Driving System)	18
	Practical Work 3: Distance Measurement Using Ultrasonic Sensor	23
	Practical Work 4: Line Follower Mobile Robot Design	30
	Practical Work 5: Wireless Mobile Robot Control Using Bluetooth	37
	Practical Work 6: Introduction to Robot Vision System	44
9	References	53

1.0 Introduction

Embedded robotics has become an essential field in modern engineering, integrating electronics, control systems, and programming to develop intelligent machines capable of performing real-world tasks. With the rapid advancement of technology, the demand for skilled engineers who are able to design, build, and control robotic systems continues to grow across various industries, including manufacturing, automation, and smart systems.

This eBook is developed to provide a structured and practical approach to learning embedded robotics through a series of guided laboratory works. Rather than focusing solely on theoretical concepts, this module emphasises hands-on learning, allowing students to explore, experiment, and implement fundamental principles in a controlled laboratory environment. By engaging directly with hardware components and system integration, students are able to develop a deeper understanding of how individual subsystems function and interact within a complete robotic system.

The learning process begins with the fundamentals of sensors and actuators, which form the basis of any embedded system. Students are then introduced to motor driving techniques for mobile robot movement, followed by the application of ultrasonic sensors for distance measurement. As the module progresses, more advanced topics are explored, including line follower robot design, wireless control using Bluetooth communication, and robot vision systems. This progression reflects a complete development pathway, from basic input and output handling to intelligent decision-making and environmental perception.

From a technical perspective, an embedded robotic system operates through the seamless integration of hardware and software within a resource-constrained environment. At the core of the system is the microcontroller, which processes real-time data acquired from sensors and executes control algorithms accordingly. The accuracy of this process depends on effective signal conditioning, analog-to-digital conversion, and reliable digital interfacing. These inputs are then translated into physical actions through actuator driver circuits, such as H-bridge configurations, which enable precise control of motor direction and speed for robot navigation.

Building on this foundation, control strategies are implemented to ensure stable and responsive robot behaviour. Techniques such as pulse-width modulation (PWM) are used to regulate motor speed, while feedback from sensors enables closed-loop control. For example, in a line follower robot, sensor readings are continuously analysed to adjust motor output, allowing the robot to maintain its trajectory.

Similarly, ultrasonic sensing introduces time-of-flight measurement principles for distance estimation, supporting functions such as obstacle detection and basic navigation.

As the system evolves towards greater complexity, communication and perception capabilities are incorporated to enhance functionality. Wireless communication using Bluetooth modules allows for remote control and real-time data exchange, exposing students to serial communication protocols and system interfacing. This is further extended through the integration of robot vision, where image acquisition and processing enable the robot to interpret its surroundings. Such capabilities introduce fundamental concepts in visual perception and intelligent decision-making, ultimately bridging the gap between basic embedded systems and more advanced autonomous robotic applications.

Through this structured progression, students are expected to develop not only strong technical knowledge but also essential practical skills such as troubleshooting, system integration, and critical thinking. The module ultimately serves as a comprehensive foundation, preparing learners to design and develop intelligent mobile robotic systems aligned with current industry practices.

2.0 Practical Task Overview

You must be familiar with this manual in order to complete the exercises given within. With all these exercises, you will learn about:

i. Fundamentals of Sensors and Actuators
ii. Mobile Robot Design (Motor Driving System)
iii. Distance Measurement Using Ultrasonic Sensor
iv. Line Follower Mobile Robot Design
v. Wireless Mobile Robot Control Using Bluetooth
vi. Introduction to Robot Vision System

Upon completion of these laboratory sessions and exercises, students will be able to understand and apply the concepts of robot positioning, identification, and communication in mobile robot control in accordance with standard robotic system practices. Students will also be able to manipulate the application of sensors and actuators, as well as implement robot identification and communication techniques during practical work based on land mobile robot design. In addition, students will be able to organise and participate in mini competitions among peers, demonstrating their ability to design, develop, and operate land mobile robots effectively in a competitive environment.

3.0 Arduino UNO R3 Microcontroller

Arduino Uno R3 is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output) : 0-13
Analog Input Pins	6 (A0-A5)
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32KB (ATmega328) of which 0.5KB used by bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Clock Speed	16MHz

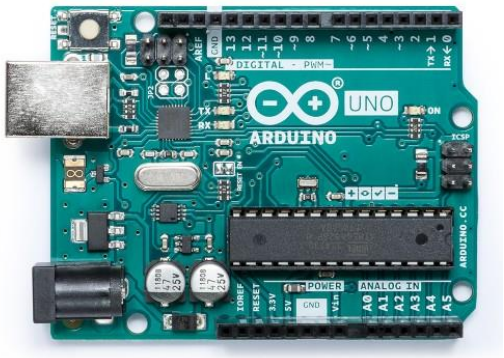


Figure 1: Arduino UNO R3 Board

Figure 1 shows the Arduino Uno Board. For more information please refer to the datasheet. The data sheet can be found at https://www.farnell.com/datasheets/1682209.pdf

4.0 Hardware and Software Preparation

This Arduino UNO R3 DIY Module Embedded Robotic covers the basic concept and application of microcontroller systems based on the Arduino Uno R3. Users will learn software and hardware development on Arduino development system such as Arduino Software (IDE) and understand how to do interfacing with external devices using suitable internal chip features. Users are also exposed to the new Microcontroller Unit (MCU) simulation software such as Proteus.

Hardware Setup

The basic hardware in this teaching kit are Arduino Uno R3, computer/laptop and USB cable. Basic hardware connection to a computer/laptop as shown in Figure 2.

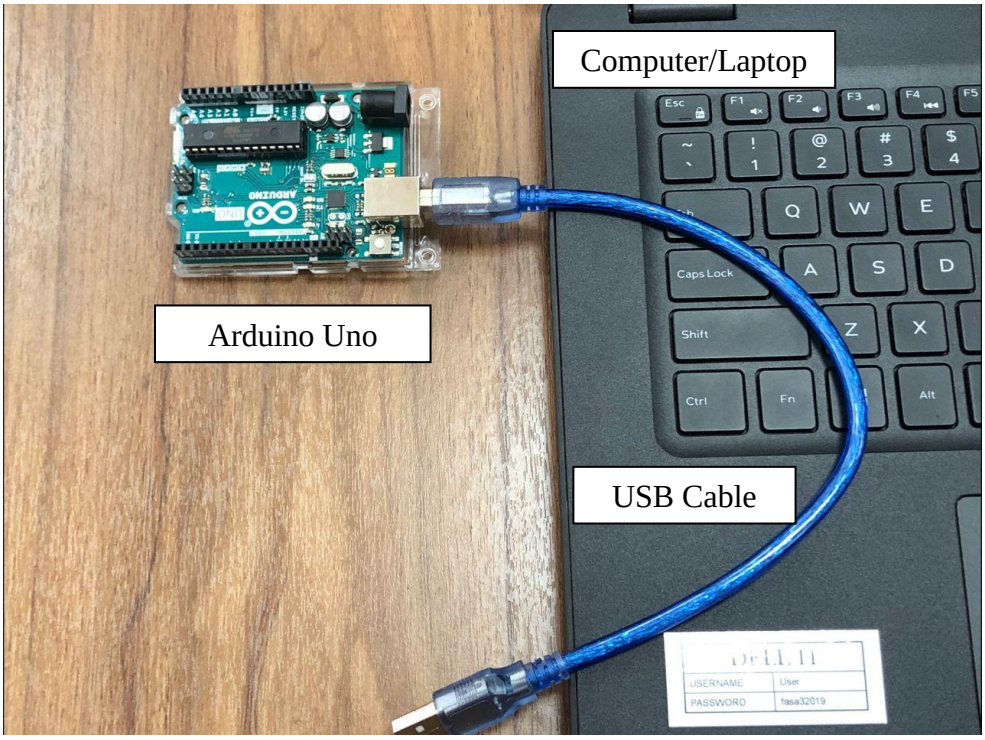


Figure 2: Basic Hardware Connection

Arduino Software (IDE)

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Arduino is an open-source physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing and MaxMSP). The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free at https://arduino.cc



Figure 3: Arduino

Active development of the Arduino software is hosted by GitHub. This organisation contains the official Arduino tools (IDE, CLI...), documentation and cores. See @arduino-libraries for the official libraries or https://github.com/arduino

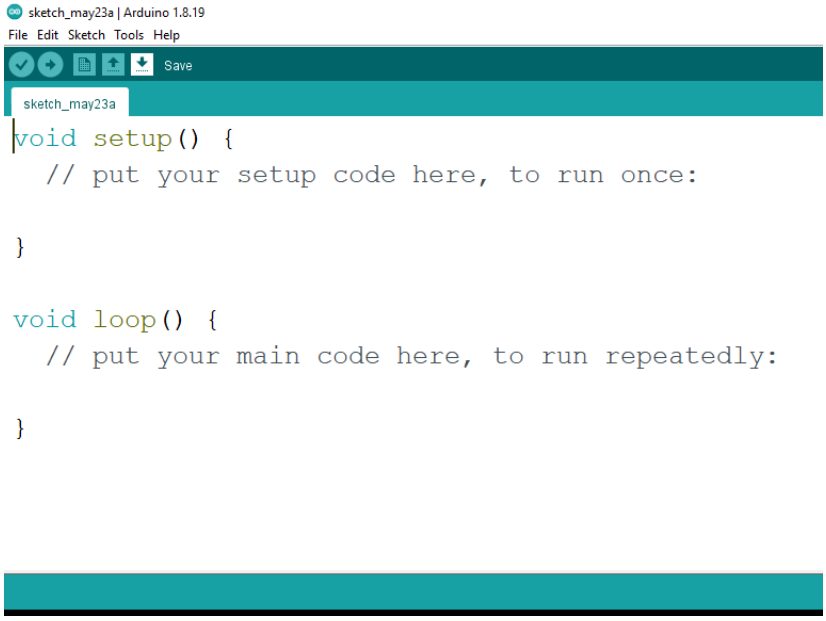


Figure 4: Arduino Editor

Proteus 7/8 Professional

Proteus is simulator software which is capable to simulate any circuit and scenario. Mainly, it is best for microcontrollers. Proteus professional design combines the ISIS schematic capture and ARES PCB layout programs to provide a powerful, integrated and easy to use tools suite for education and professional PCB Design.

The Proteus Professional are software for automated design of electronic circuits. The package is a system of circuit simulation, based on the models of electronic components in SPICE. A distinctive feature of the package Proteus Professional is the possibility of modelling of the programmable devices: microcontrollers, microprocessors, DSP and others.

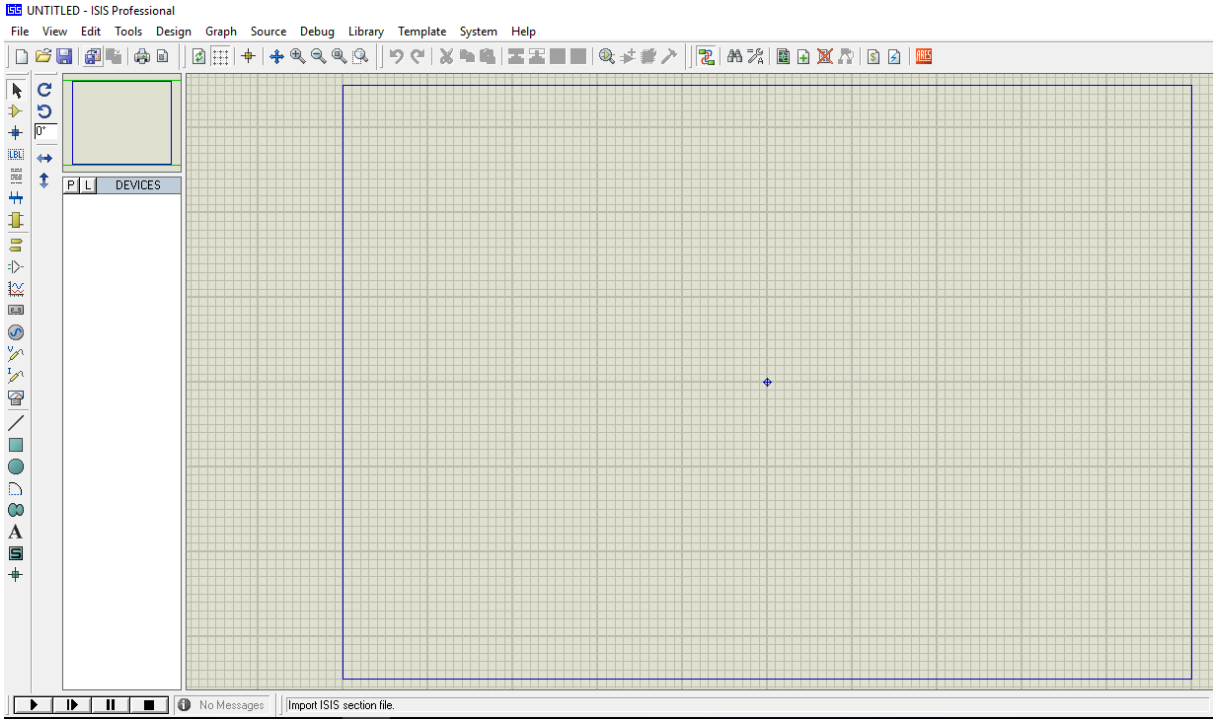


Figure 5: Proteus Workspace

5.0 Hardware Specification

Table 1 shows the hardware specification for Arduino UNO R3 DIY Kit.

Table 1: Arduino UNO R3 DIY Kit Hardware Specification

No	Components	Quantity
1	Arduino UNO R3	1
2	4WD Smart Robot Car Chassis	1
3	Ultrasonic Sensor HC-SR04	1
4	Auto-calibration Line Sensor	1
5	Bluetooth Module HC-05	1
6	Dual H-Bridge Driver L293D	1
7	Panasonic Eneloop Charger & Nimh Battery AA	8
8	Battery Holder support 5 x Battery AA	1
9	Pixy 2 CMUcam5 Image Sensor	1
10	M3 PCB Stand (Screw & Nut type) 10 mm & 15mm	20
11	Breadboard Mini (35mmx42mm)	2

6.0 Smart Embedded Robotic e-Lab

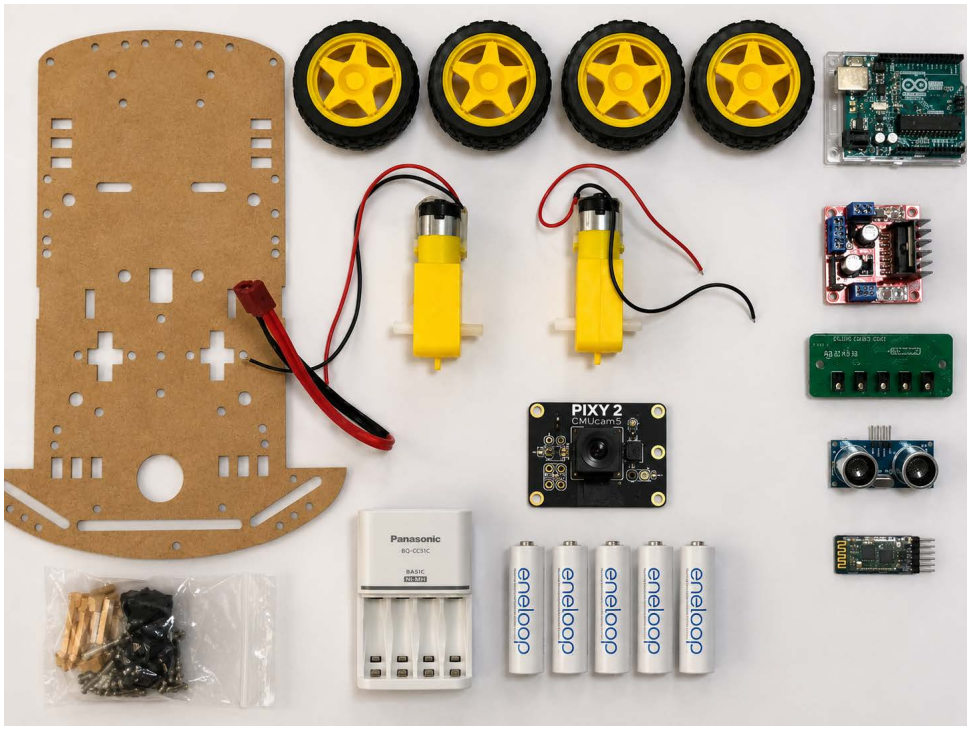


Figure 6: Hardware Specification

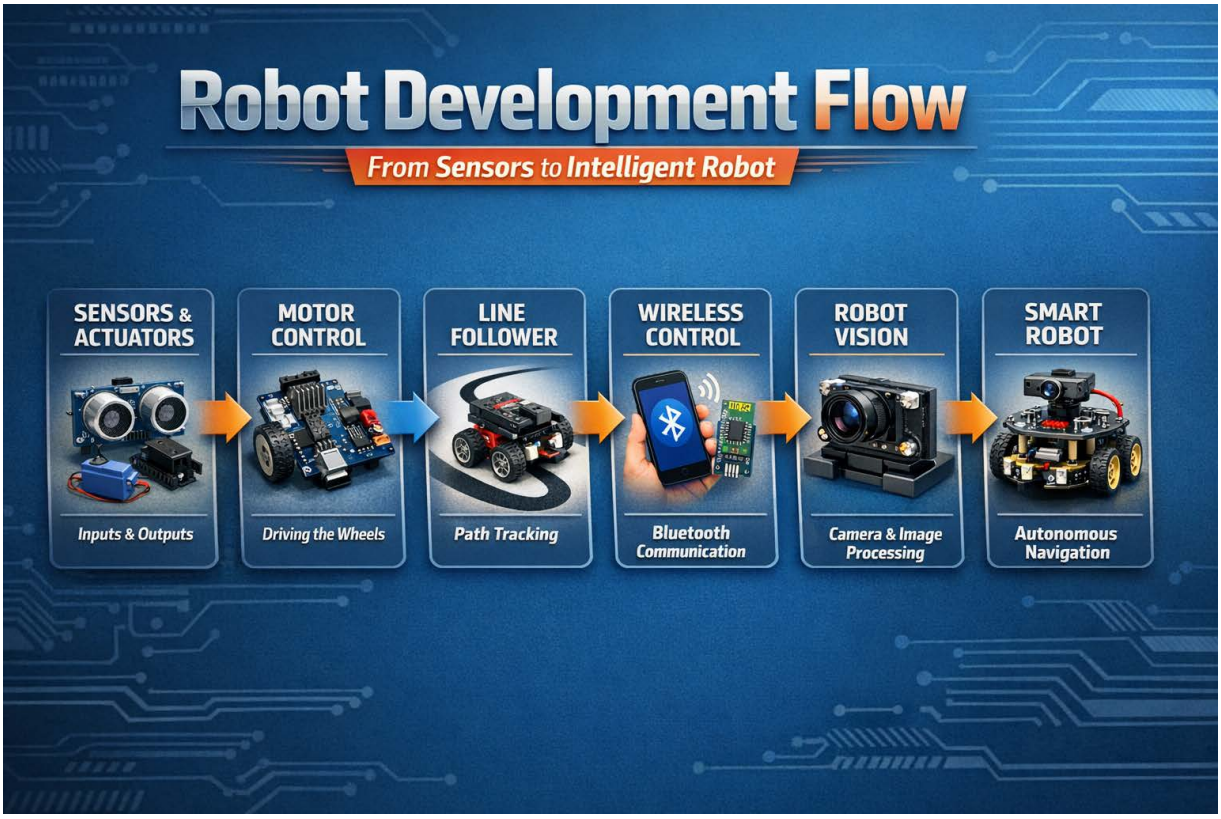
7.0 Operational Procedure

i. Prepare all the equipment and material use such as Arduino UNO R3, input-output (I/O) components, robot mobile and USB cable.
ii. Plug one end of the USB cable into the USB Arduino board. Plug the other end into a USB port on your PC.
iii. The lights indicator on the Arduino board will turn on. It shows that the equipment is in good condition.
iv. Open Arduino Editor workspace window. Select menu Tools – Board: “Arduino UNO” – Select Arduino UNO.
v. Then, select menu File – Preferences – menu Settings – checked box “Show verbose output during: √ compilation – then OK.
vi. Follow the next procedure to complete the practical work.
vii. Your PC must have Proteus and Arduino IDE installed.

PRACTICAL WORK

PRACTICAL WORK SERIES

1. PT1 : Fundamentals of Sensors and Actuators
2. PT2 : Motor Driving System
3. PT3 : Distance Measurement Using Ultrasonic Sensor
4. PT4 : Line Follower Mobile Robot Design
5. PT5 : Wireless Mobile Robot Control Using Bluetooth
6. PT6 : Introduction to Robot Vision System



PRACTICAL TASK 1

SENSORS AND ACTUATORS

1 LEARNING OUTCOMES (LO):

1. Manipulate the application of sensor and actuator

2 OBJECTIVE

a. Apply digital inputs to microcontrollers (Switch, IR)
b. Apply digital output to microcontrollers (LED)
c. Develop a program/ sketch to comply analog and digital sensor as input with the controller.
d. Identify types of input and output.

3 THEORY

Sensors

Data transfer from the sensor to the CPU can be either CPU-initiated (polling) or sensor-initiated (via interrupt).

Digital sensors: The signal produced or reflected by the sensor is binary.

4 EQUIPMENT / TOOLS

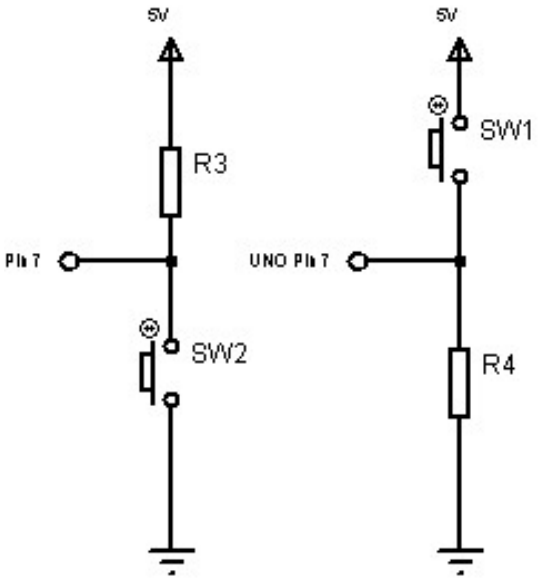
a. Personal Computer

b. Arduino IDE

c. Arduino UNO

d. Switch, LED

5 PROCEDURE



Active Low Active high

Figure 1: Switch connection

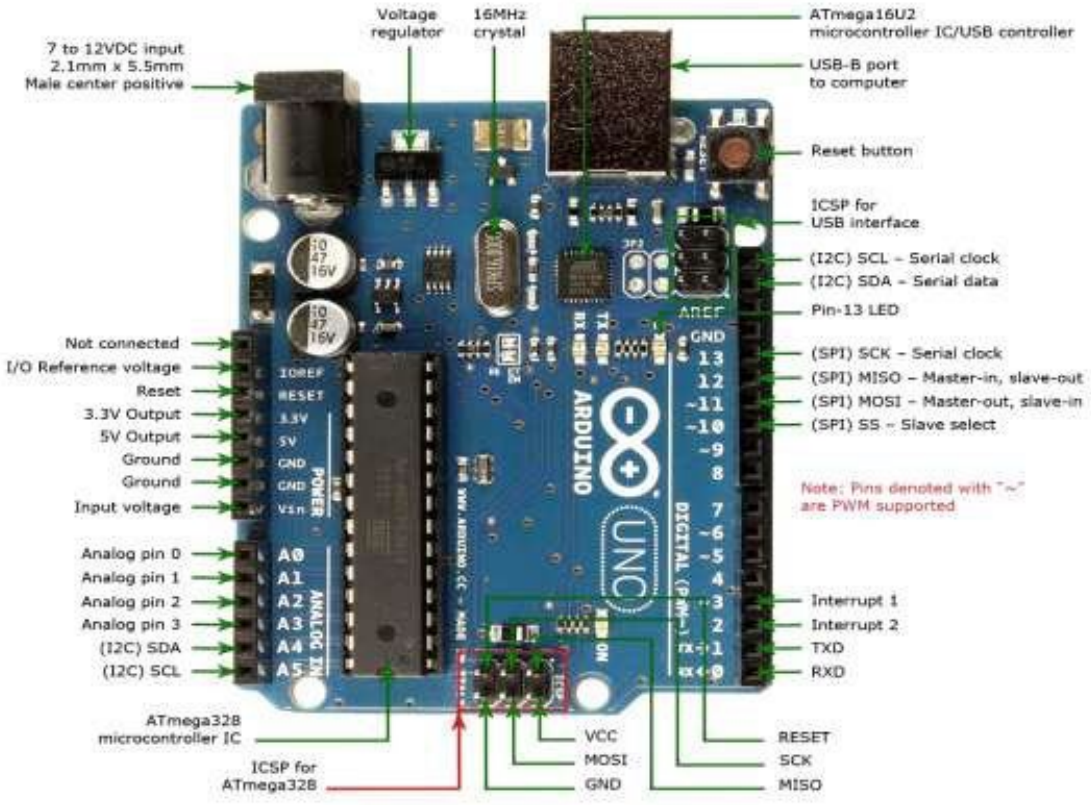


Figure 2: Pin diagram for Arduino Uno

Task 1 - Active LOW Input

a) Write a program to use digital input button (Active Low).
b) Observe the output and write down the Result.

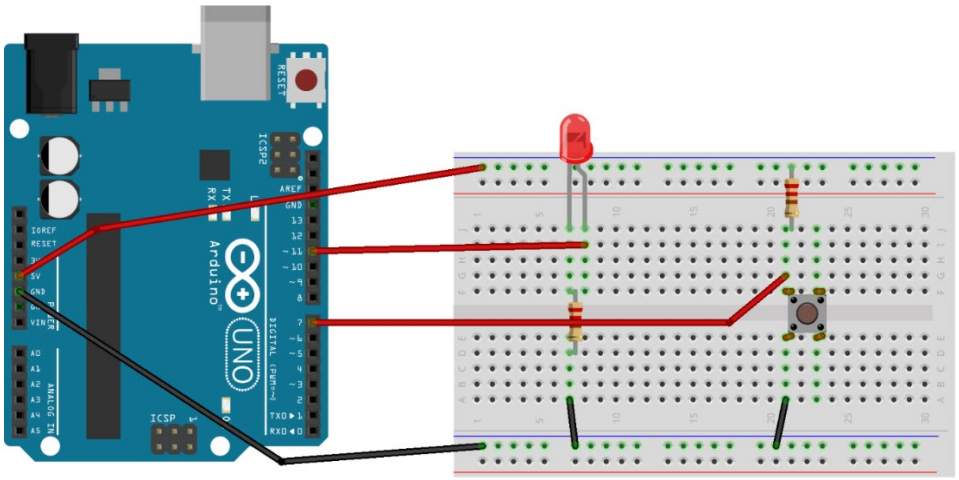


Figure 3: Wiring Diagram (Active Low)

Programming for TASK 1

Code to use or design your own code (Example):-

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 7;     // the number of the pushbutton pin
const int ledPin =  11;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup()
{
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
void loop()
```

```
{
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH)
  {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else
  {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Task 2 - Active HIGH Input

a) Write a program to use digital input button (Active High).
   - Use same programming as TASK 1

b) Observe the output and write down the Result.

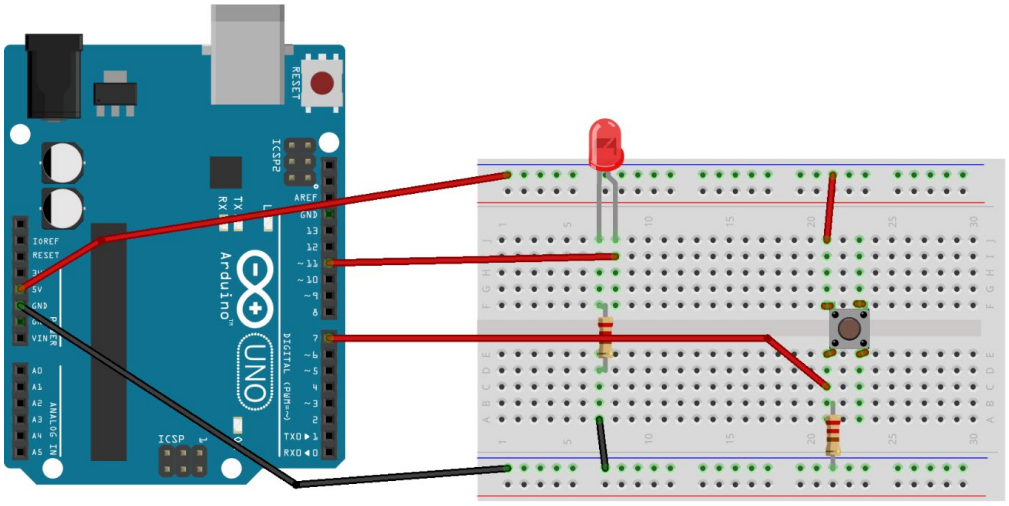


Figure 4: Wiring Diagram (Active High).

Task 3 – Analog sensor function

a) Write a program to use analog sensor function.
b) Observe the output and write down the Result.

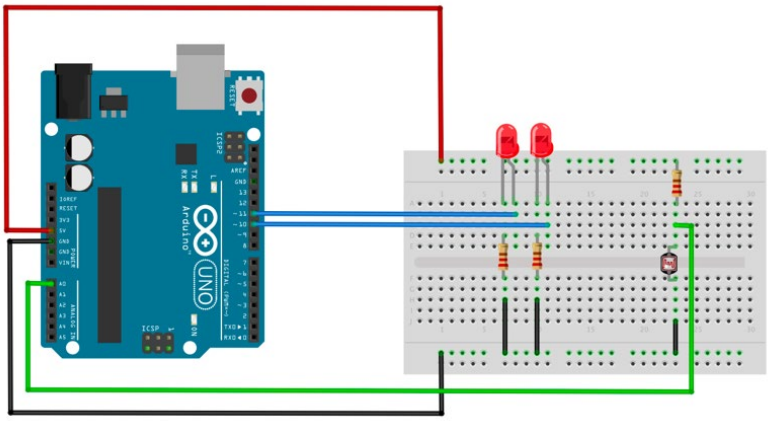


Figure 5: Wiring Diagram

Programming for TASK 3

```
// constants won't change. They're used here to set pin numbers:


const int ledPin1 =  11;      // the number of the LED pin
const int ledPin2 =  10;      // the number of the LED pin
int sensorPin = A0;
int sensorValue = 0;          // variables will change:


void setup()
{                                   // initialize the LED pin as an output:
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  Serial.begin(9600);                       //sets the data rate for serial data
transmission:
  }
void loop()
{                                           // read the state of the sensor value:
  sensorValue = analogRead(sensorPin);
  if (sensorValue > 400)
```

```
  {
                                    // turn LED1 on:
digitalWrite(ledPin1, HIGH);
digitalWrite(ledPin2, LOW);
delay(100);                         //delay period equal to 0.1 second
  }
  else
  {                                 // turn LED1 off:
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
    delay(100);
  }
  Serial.print("\n sensor = ");
  Serial.print(sensorValue);
}
```

6 RESULT

TASK 1 Push Button

Active Low

PUSH BUTTON	LED OUTPUT
Push	
Not Push	

TASK 2 Push Button

Active High

PUSH BUTTON	LED OUTPUT
Push	
Not Push	

TASK 3 Push Button

Active Low – 1 push button, 2 LED with delay function

LDR BRIGHTNESS	ANALOG VALUE	LED 1 OUTPUT	LED 2 OUTPUT
LOW			
HIGH			

7 Questions:

1. Explain how Active High and Active Low switch function.
2. What is the function of "//" symbol?

8 DISCUSSION

9 CONCLUSION

PRACTICAL TASK 2

MOBILE ROBOT DESIGN

(Apply Concept of Motor Driving Design)

1 LEARNING OUTCOMES (LO):

a. CLO2: design the concept of robot positioning, identification and communication in mobile robot control according to a standard robot organization regulation (C6, PLO3)
b. CLO3: manipulate the application of sensor and actuator, robot identification and communication during practical work based on land mobile robot design (P4, PLO5)

2 OBJECTIVE

a. Design/Show mobile robot driving method using DC motor (Differential drive).
b. Identify the concept of Stop, Forward, Reverse, Left and Right.
c. Develop program/ sketch to control the mobile robot.

3 THEORY

The differential drive design has two motors mounted in fixed positions on the left and right side of the robot, independently driving one wheel each. Since three ground contact points are necessary, this design requires one or two additional passive caster wheels or sliders, depending on the location of the driven wheels.

Differential drive is mechanically simpler than the single wheel drive, because it does not require rotation of a driven axis. However, driving control for differential drive is more complex than for single wheel drive, because it requires the coordination of two driven wheels.

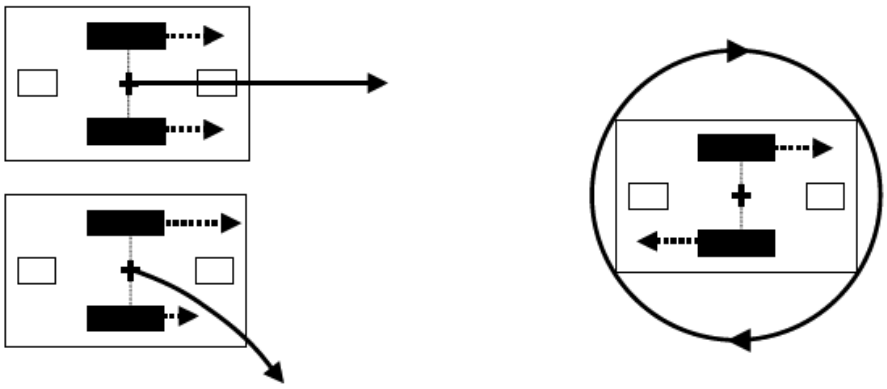
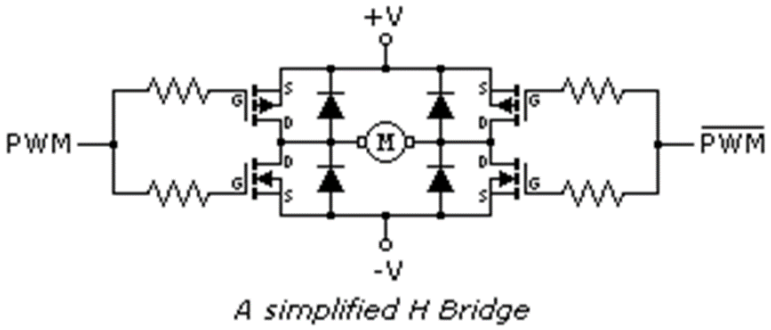


Figure 1: Differential drive



A simplified H Bridge

Figure 2: H- Bridge

Table 1: DC Motor Direction

IN1	IN2	Description
n/a	n/a	Motor A is off
0	0	Motor A is stopped (brakes)
0	1	Motor A is on and turning backwards
1	0	Motor A is on and turning forwards
1	1	Motor A is stopped (brakes)

4 EQUIPMENT / TOOLS

a. Personal Computer
b. Arduino IDE
c. Arduino UNO R3
d. LN298N motor driver
e. Motor, simple DC motor model

5 PROCEDURE

a) Construct the circuit Figure 3 base on the pin connection as stated on the Table 2

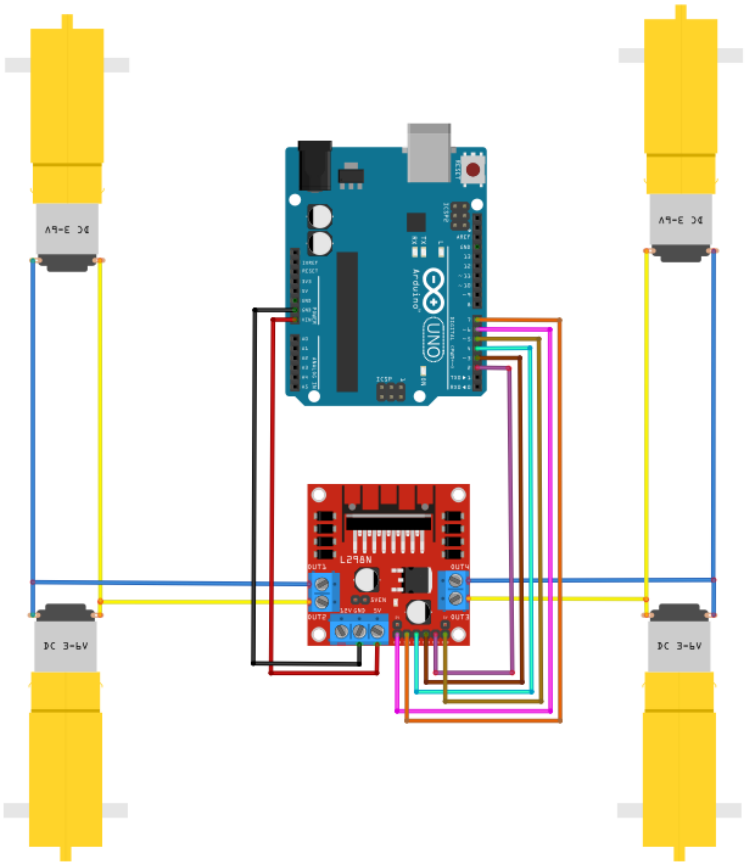


Figure 3: Wiring schematic

Table 2: Pin Diagram

Arduino UNO	L298N Motor Driver	Left Motor	Right Motor
Digital pin 7	IN1		
Digital pin 6	ENA		
Digital pin 5	ENB		
Digital pin 4	IN2		
Digital pin 3	IN3		
Digital pin 2	IN4		
	OUT4		Right terminal
	OUT3		Left terminal
	OUT2	Right terminal	
	OUT1	Left terminal	
GND	GND		
5V	5V (Parallel with 12v)		

b) Write a program given below and write the output result at Table 3

```
//pw2 dec50122

#include <stdint.h>

#define PWMA  6
#define PWMB  5
#define IN1   7
#define IN2   4
#define IN3   3
#define IN4   2
#define Speed 200

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.print("Starting...\n");  //for testing purposes
  pinMode(PWMA, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

}

void depan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void belakang() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kanan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kiri() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void henti() {
  analogWrite(PWMA, 0);
  analogWrite(PWMB, 0);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 1);
}
```

c) Observe and discuss the results.

6 RESULT

Table 3: Results

Driver				**Left Motor**		**Right Motor**		**Description** Brakes/Forward/Backward/Turn Left/Right
IN1	IN2	IN3	IN4	Stop/CW/CCW		Stop/CW/CCW		
								forward
								backward
								turn left
								turn right
								brakes

7 QUESTIONS

1. Sketch and simulate a code to make motor left and motor right to rotate CW for 1 second and rotate CCW for 1 second. Make it run on infinity looping. (1mark).
2. Sketch a program to make robot move forward for 10 secs, stop for 1 secs, the turn right 180°, stop for 1 sec then move forward for 10 secs. (2marks)
3. Sketch a program to make robot spin CW for 5 secs, stop 1 secs and then spin CCW for 5 secs and completely stop.(2marks)

8 DISCUSSION

9 CONCLUSION

PRACTICAL TASK 3

MOBILE ROBOT DESIGN
(Distance Measurement Using Ultrasonic Sensor)

1 LEARNING OUTCOMES (LO):

a. CLO2: design the concept of robot positioning, identification and communication in mobile robot control according to a standard robot organization regulation (C6, PLO3)
b. CLO3: manipulate the application of sensor and actuator, robot identification and communication during practical work based on land mobile robot design (P4, PLO5)

2 OBJECTIVE

a. Use the ultrasonic sensor to measure distance.
b. Write a program Arduino to control mobile robot for obstacle avoiding.

3 THEORY

Ultrasonic sensor are commonly used to sense proximity and detect levels of high reliability in the non-contact detection such as presence, level, position, and distance of objects. In addition, ultrasonics are independent of light, smoke, dust, and color makes it is reliable for distance measurements. The obstacle detection by using ultrasonic methods are frequently used in mobile robot application such as drone which is unmanned aerial vehicle (UAV). HC-SR04 (Ultrasonic sensor) is one of ultrasonic sensor commonly used mobile robot can measure the distance of an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and waits for that sound wave to bounce back as shows in Figure 1. By recording the time taken between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sensor and the object.

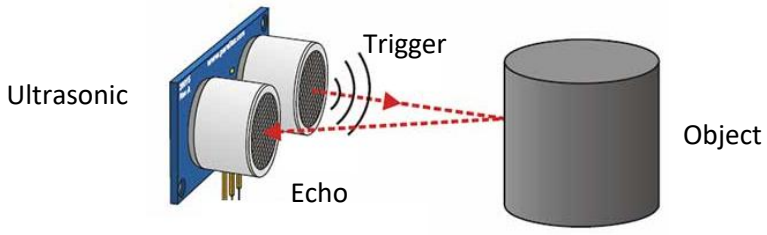


Figure 1: Ultrasonic sensor method

The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo as shows in Figure 2. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Arduino Board respectively and the trig and echo pins to any Digital I/O pin on the Arduino Board.



Figure 2: HC-SR04 Ultrasonic

In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled. For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2 as shows in Figure 3.

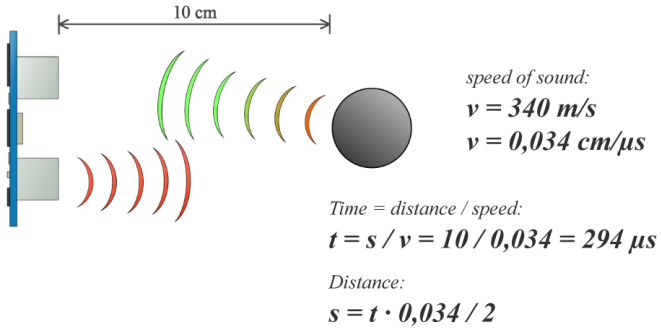


Figure 3: Distance measurement

4 EQUIPMENT / TOOLS

a. Personal Computer
b. Arduino IDE
c. Arduino UNO R3
d. LN298N motor driver
e. Motor, simple DC motor model

5 PROCEDURE

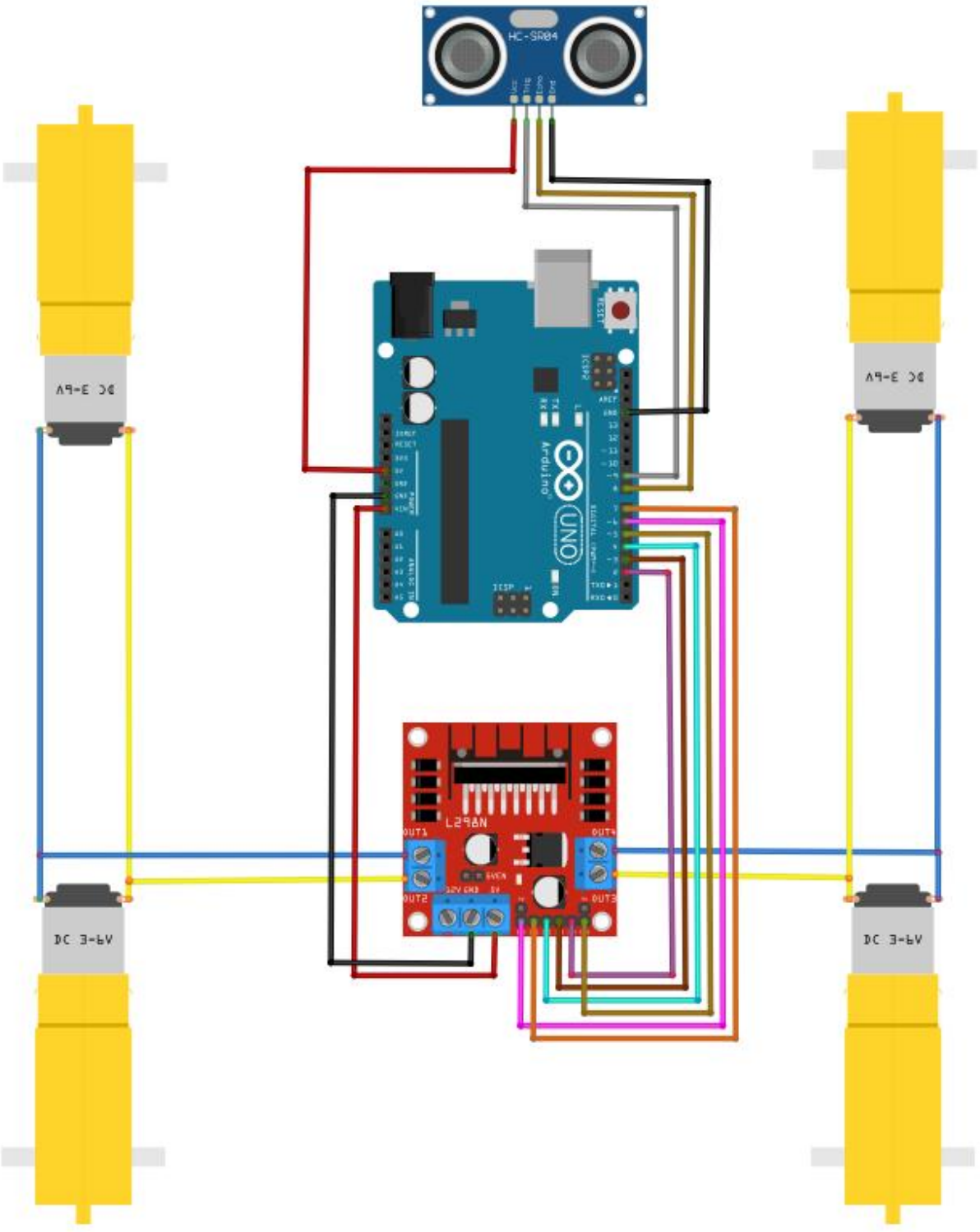


Figure 3: Wiring schematic

a) Construct the circuit Figure 3 base on the pin connection as stated on the Table 2

Table 2: Pin Diagram

Arduino UNO	**L298N Motor Driver**	**Left Motor**	**Right Motor**	**HC-SR04 Ultrasonic**
Digital pin 9				Trig
Digital pin 8				Echo
Digital pin 7	IN1			
Digital pin 6	ENA			
Digital pin 5	ENB			
Digital pin 4	IN2			
Digital pin 3	IN3			
Digital pin 2	IN4			
	OUT4		Right terminal	
	OUT3		Left terminal	
	OUT2	Right terminal		
	OUT1	Left terminal		
GND	GND			
5V	5V (Parallel with 12v)			

b) Write a program given below and write the output result at Table 3

```
#include <stdint.h>

#define PWMA  6 //utk pwm motor driver
#define PWMB  5 //sama mc atas
#define IN1   7 //utk kawal dc motor
#define IN2   4 //sama mcm atas
#define IN3   3
#define IN4   2
#define trig  9 //utk trigger signal ultrasonic
#define echo  8 //receive signal utrasonic
#define Speed 200 //kelajuan dc motor. ubah kat sini
long duration = 0;  //utk simpan nilai masa signal
uint16_t distance = 0;  //variable simpan nilai jarak objek

void setup() {  //setup function
  // put your setup code here, to run once:
  Serial.begin(115200); //start serial monitor with 115200 bit/s baud rate
  Serial.print("Starting...\n");  //for testing purposes
  pinMode(PWMA, OUTPUT);  //set pin as ouput
  pinMode(PWMB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT); //set pin as input
}

void loop() { //loop funtion
  // put your main code here, to run repeatedly:
 ultrasonic();
  if(distance<10)
    {
        henti();
        delay(1000);

        belakang();
        delay(500);

        kiri();
        delay(500);
    }
  else
    {
      depan();
    }
}
```

```
void ultrasonic()
{
    digitalWrite(trig, LOW);     // Clears the trigPin
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);   // Sets the trigPin on HIGH state for 10 micro seconds
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    duration = pulseIn(echo, HIGH);  // Reads the echoPin, returns the sound wave travel time in microseconds
    distance= duration*0.034/2;     // Calculating the distance
    Serial.print("Distance: ");    // Prints the distance on the Serial Monitor
    Serial.println(distance);
}

void depan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void belakang() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kanan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kiri() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void henti() {
  analogWrite(PWMA, 0);
  analogWrite(PWMB, 0);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 1);
}
```

c) Observe and discuss the results.

6 RESULT

Table 3: Results

Ultrasonic Sensor	**Left Motor**	**Right Motor**	**Description** Brakes/Forward/Backward/Turn Left/Right
	Stop/CW/CCW		
No Obstacle			
Detect Obstacle			

7 QUESTIONS

1. Calculate the distance of Ultrasonic measurement when v is 300 m/s and and t is 200us. (5 Marks)
2. Sketch and simulate a code to make your mobile robot move backward 5 sec continue with turn right 1 sec when the mobile robot detect obstacle. (10 Marks)

8 DISCUSSION

9 CONCLUSION

PRACTICAL TASK 4

MOBILE ROBOT DESIGN
(Line Follower Robot)

1 LEARNING OUTCOMES (LO):

a. CLO2: design the concept of robot positioning, identification and communication in mobile robot control according to a standard robot organization regulation (C6, PLO3)
b. CLO3: manipulate the application of sensor and actuator, robot identification and communication during practical work based on land mobile robot design (P4, PLO5)

2 OBJECTIVE

a. Use Infra-Red (IR) line sensor.
b. Write a program Arduino to control mobile robot for follow the line.

3 THEORY

A **line follower robot**, or a **LFR** in short, is a simple autonomous robot that optically tracks a line made on the surface of the floor. That means you have an arbitrary line drawn on the floor and the robot tracks it by moving right along it! The line is sensed using a piece of hardware called a line sensor. A line sensor can be easily made using a low cost **IR Rx/Tx pair**. The IR Tx emits IR radiation and the Rx helps in receiving the waves. See the figure below for working of line sensor.

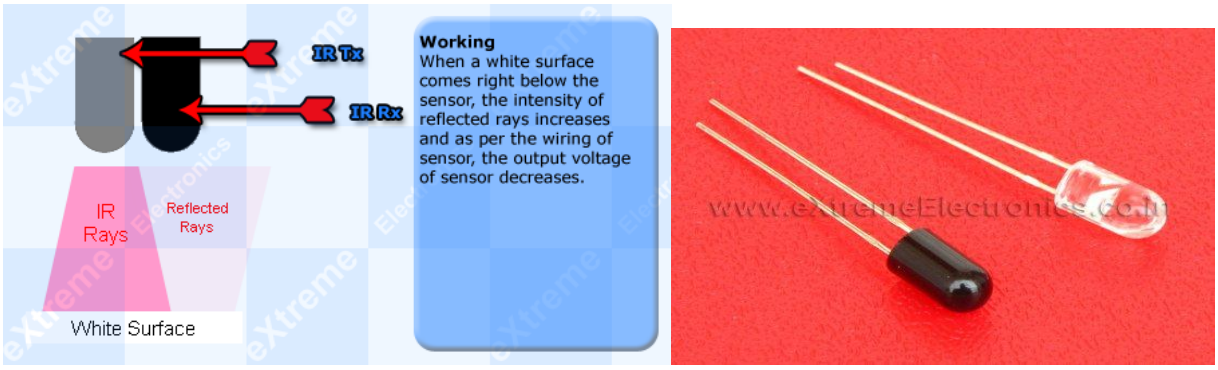


Figure 1: Transmitter and receiver Infra-red

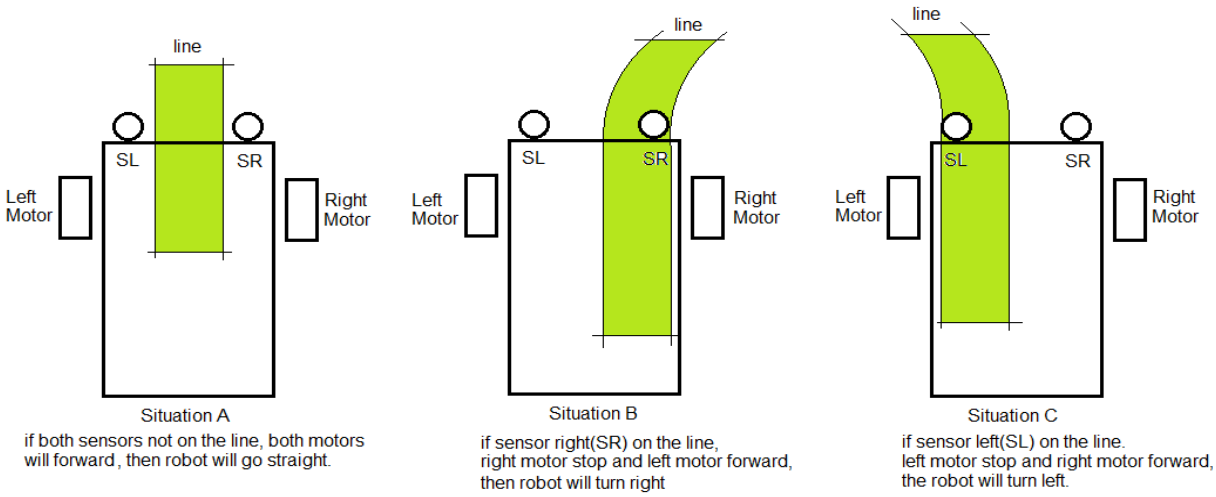


Figure 2: Line following method

A basic movement method of line following robot as shows in Table 1. The robot will turn left when sensor left (SL) is '1' and sensor right (SR) is '0'. Then, the robot will turn right when SL is '0' and SR is '1'. Otherwise, the robot will move forward when SL and SR is '0'.

Table 1: Line following robot movement

Sensor		**Motor movement**
SL	SR	
1	0	Left
0	1	Right
0	0	Forward

4 EQUIPMENT / TOOLS

a. Personal Computer
b. Arduino IDE
c. Arduino UNO R3
d. LN298N motor driver
e. Motor, simple DC motor model
f. Infrared sensor

5 PROCEDURE

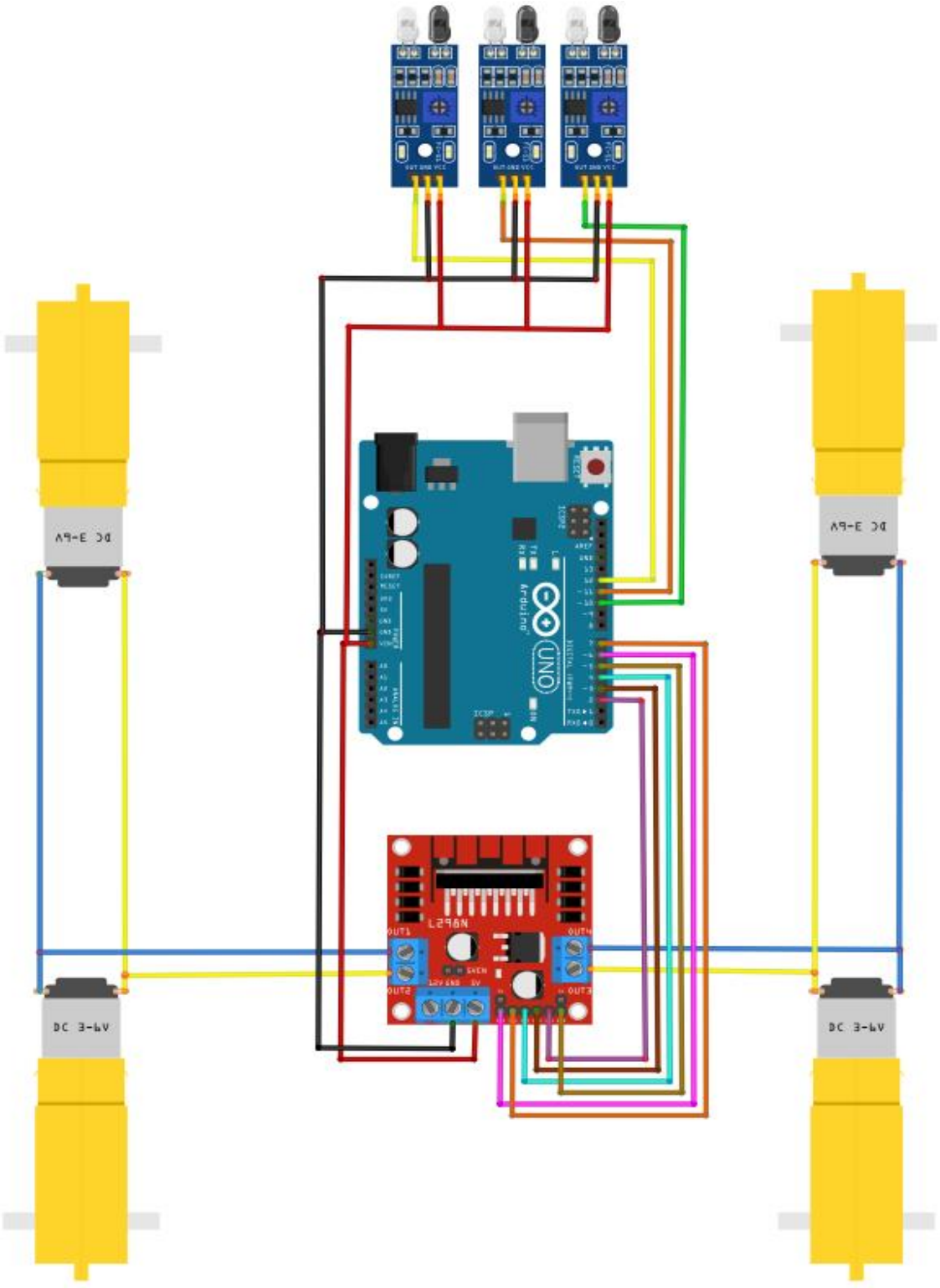


Figure 3: Wiring schematic

a) Construct the circuit Figure 3 base on the pin connection as stated on the Table 2

Table 2: Pin Diagram

Arduino UNO	**L298N Motor Driver (1)**	**Left Motor (1)**	**Right Motor (2)**	**Infrared Sensor 1**	**Infrared Sensor 2**	**Infrared Sensor 3**
Digital pin 12						D0
Digital pin 11					D0	
Digital pin 10				D0		
Digital pin 7	IN1					
Digital pin 6	ENA					
Digital pin 5	ENB					
Digital pin 4	IN2					
Digital pin 3	IN3					
Digital pin 2	IN4					
	OUT4		RIGHT TERMINAL			
	OUT3		LEFT TERMINAL			
	OUT2	RIGHT TERMINAL				
	OUT1	LEFT TERMINAL				
GND	GND			GND	GND	GND
5V	5V			Vcc	Vcc	Vcc

b) Write a program given below and write the output result at Table 3

```
//PW4 DEC50122
//ULTRASONIC SENSOR

#include <stdint.h>

#define PWMA  6
#define PWMB  5
#define IN1   7
#define IN2   4
#define IN3   3
#define IN4   2
#define sen1  10  //utk ir1
#define sen2  11  //utk ir2
#define sen3  12  //utk ir3
#define laju  200 //utk speed robot

uint8_t volatile sen1_state = 0;  //utk simpan data ir
uint8_t volatile sen2_state = 0;
uint8_t volatile sen3_state = 0;

void setup() {
  serial.begin(115200); //utk serial monitor
  pinMode(PWMA, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(sen1, INPUT);
  pinMode(sen2, INPUT);
  pinMode(sen3, INPUT);
}

void loop() {
  sen1_state = digitalRead(sen1); //utk baca data sensor semasa
  sen2_state = digitalRead(sen2);
  sen3_state = digitalRead(sen3);

  if(sen1_state == LOW && sen1_state == HIGH && sen3_state == LOW)
      {
        depan();
      }
  else if(sen1_state == HIGH && sen1_state == LOW && sen3_state == LOW)
      {image.png
        kiri();
```

```
        }
    else if(sen1_state    HIGH && sen1_state    HIGH && sen3_state    LOW)
        {
          kiri();
        }
    else if(sen1_state    LOW && sen1_state    LOW && sen3_state    HIGH)
        {
          kanan();
        }
    else if(sen1_state    LOW && sen1_state    HIGH && sen3_state    HIGH)
        {
          kanan();
        }
    else  henti();
  }
```

```
void depan()  {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void belakang()  {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kanan()  {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kiri()  {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}
```

c) Observe and discuss the results.

6 RESULT

Sensor 1	Sensor 2	Sensor 3	Mobile Robot Movement

Table 3: Results

7 QUESTIONS

1. Sketch a code to make your mobile robot using 5 sensor which sense 3 sensor to turn right, 3 sensor sense to move forward and 3 sensors to turn left. In this case the speed of your mobile robot should slower than used 3 sensors so that the robot can sense accurately.

8 DISCUSSION

9 CONCLUSION

PRACTICAL TASK 5

MOBILE ROBOT DESIGN
(Wireless Mobile Robot Control Using Bluetooth HC-05)

1 LEARNING OUTCOMES (LO):

a. CLO2: design the concept of robot positioning, identification and communication in mobile robot control according to a standard robot organization regulation (C6, PLO3)
b. CLO3: manipulate the application of sensor and actuator, robot identification and communication during practical work based on land mobile robot design (P4, PLO5)

2 OBJECTIVE

a. Use Bluetooth HC-05 sensor to communicate with android and Arduino.
b. Write a program Arduino to control wireless mobile robot.

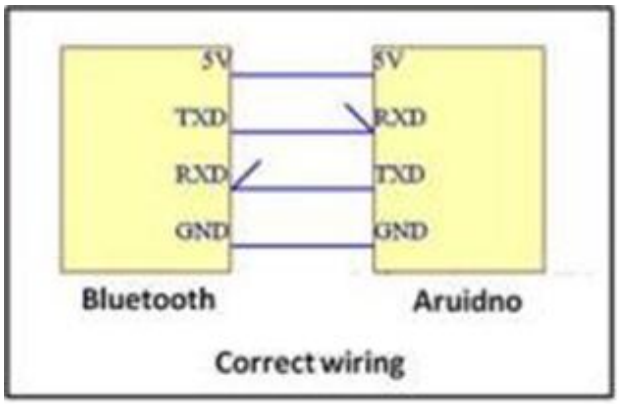
3 THEORY

The HC-05 Bluetooth module is the most economical and easiest way to go wireless (via Bluetooth).
This module makes it easy for you to wirelessly extend your serial interface, so you can control any program running on your Laptop/android/handphone with serial port interface.

The 4 pins are:

- VCC (Power 3.3 – 6V);
- GND;
- TXD;
- RXD;

Default pairing code: 1234
Default baudrate: 9600



Usually this Bluetooth module is used with the Arduino, but it can be used with others microcontrollers devices.

4 EQUIPMENT / TOOLS

a. Personal Computer
b. Arduino IDE
c. Arduino UNO R3
d. LN298N motor driver
e. Motor, simple DC motor model
f. Bluetooth HC-05 sensor
g. Android Phone with Arduino Bluetooth Controller apps.

5 PROCEDURE

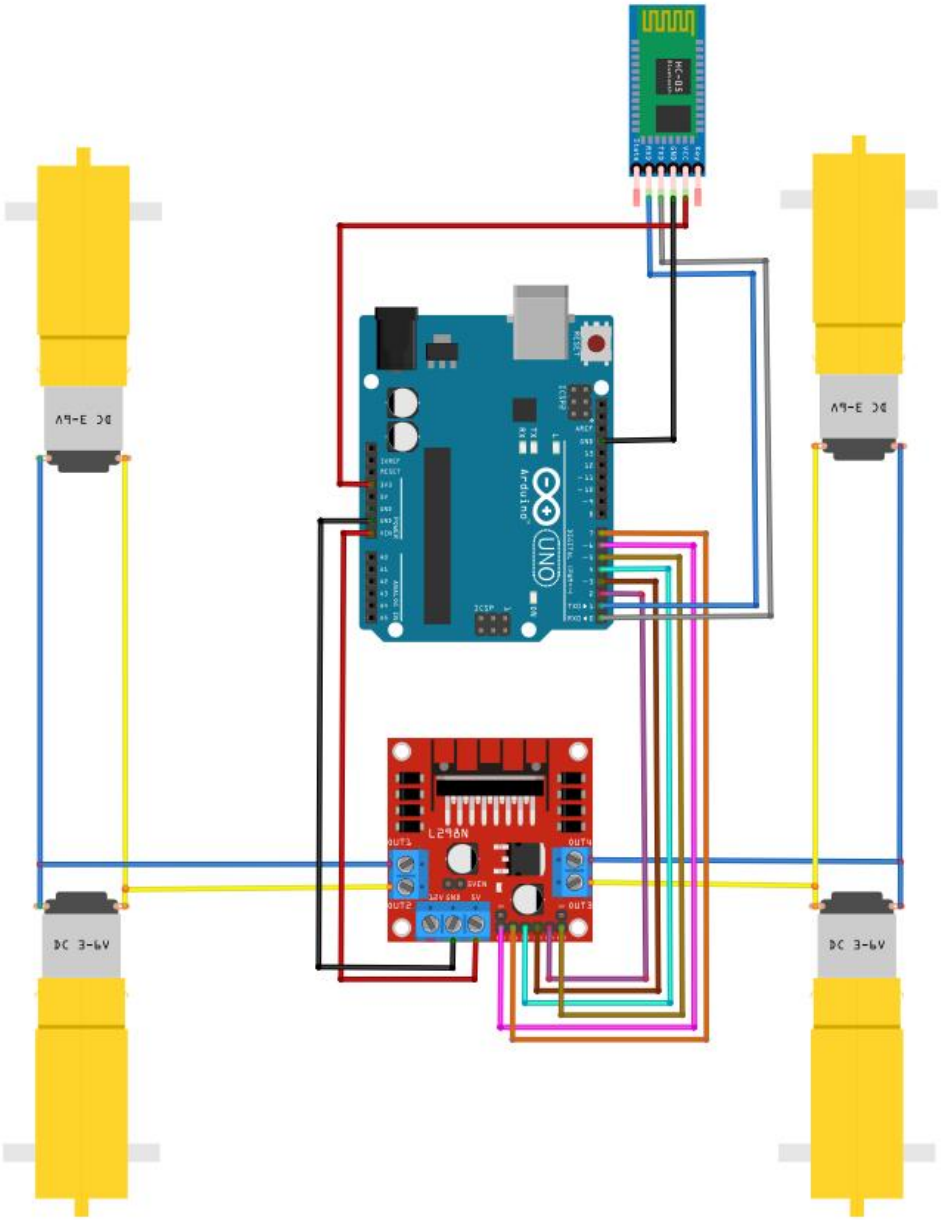


Figure 1: Wiring schematic

a) Construct the circuit Figure 1 base on the pin connection as stated on the Table 1

Table 1: Pin Diagram

Arduino UNO	**L298N Motor Driver (1)**	**Left Motor (1)**	**Right Motor (2)**	**Bluetooth Module**
Digital pin 0 (RX)				TX
Digital pin 1 (TX)				RX
Digital pin 12				
Digital pin 11				
Digital pin 10				
Digital pin 7	IN1			
Digital pin 6	ENA			
Digital pin 5	ENB			
Digital pin 4	IN2			
Digital pin 3	IN3			
Digital pin 2	IN4			
	OUT4		RIGHT TERMINAL	
	OUT3		LEFT TERMINAL	
	OUT2	RIGHT TERMINAL		
	OUT1	LEFT TERMINAL		
GND	GND			GND
5V	5V			
3v3				VCC

b) Write a program given below and write the output result at Table 2

```
//pw5 dec50122
//HW - uno, L298 x 1, hc-05
#include <stdint.h>
#define PWMA  6 //utk pwm motor driver
#define PWMB  5 //sama mc atas
#define IN1   7 //utk kawal dc motor
#define IN2   4 //sama mcm atas
#define IN3   3
#define IN4   2
#define Speed 200 //kelajuan dc motor. ubah kat sini

uint8_t rx_data = 0;  //data yg diterima dr hc-05

void setup() {  //setup function
  // put your setup code here, to run once:
  Serial.begin(9600); //start serial monitor with 9600 bit/s baud
rate
  Serial.print("Starting...\n");  //for testing purposes
  delay(333);   //utk initialise serial communication
  pinMode(PWMA, OUTPUT);  //set pin as ouput
  pinMode(PWMB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}

void loop() { //loop funtion
  // put your main code here, to run repeatedly:
```

```
  if(Serial.available() > 0){     //cek ada x communication dgn
bt.
    rx_data = Serial.read();   //data dr bt akan disimpan di
rx_data
    if(rx_data == 'U')
      depan();
    else if(rx_data == 'D')
      belakang();
    else if(rx_data == 'L')
      kiri();
    else if(rx_data == 'R')
      kanan();
    else
      henti();
  }
}

void depan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void belakang() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
```

```
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kanan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kiri() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void henti() {
  analogWrite(PWMA, 0);
  analogWrite(PWMB, 0);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 1);
}
```

c) Install and setting Apps on mobile phone.
d) Observe and discuss the results.

6 RESULT

Button	Label	Robot Movement
		
		
		
		
		
		
		
		

Table 2: Results

7 QUESTIONS

1. Sketch and simulate a code to make your robot move when button pressed and stop when button released.

8 DISCUSSION

9 CONCLUSION

PRACTICAL TASK 6

ROBOT VISION

1 LEARNING OUTCOMES (LO):

a. CLO2: design the concept of robot positioning, identification and communication in mobile robot control according to a standard robot organization regulation (C6, PLO3)
b. CLO3: manipulate the application of sensor and actuator, robot identification and communication during practical work based on land mobile robot design (P4, PLO5)

2 OBJECTIVE

a. To learn robot vision
b. To understand the basic robot vision and autonomous robot

3 THEORY

Image processing in mobile robot

Image tracking, surveillance systems, and robotic platforms are fields that have been well studied in the past decade. However, in the majority of surveillance and video tracking systems, the sensors are stationary. The stationary systems require the desired object to stay within the surveillance range of the system. If the object goes beyond this range. It no longer becomes tractable. One solution to this problem is to design the system as a mobile system that uses an infrared range sensor, and a visual-spectrum camera, to track the object and avoid obstacles.

The most important method in the Arduino library is getBlocks(), which returns the number of objects Pixy has detected. You can then look in the pixy.ccc.blocks[] array for information about each detected object (one array member for each detected object.) Each array member (i) contains the following fields:

- pixy.ccc.blocks[i].m_signature The signature number of the detected object (1-7 for normal signatures)
- pixy.ccc.blocks[i].m_x The x location of the center of the detected object (0 to 316)
- pixy.ccc.blocks[i].m_y The y location of the center of the detected object (0 to 208)
- pixy.ccc.blocks[i].m_width The width of the detected object (1 to 316)
- pixy.ccc.blocks[i].m_height The height of the detected object (1 to 208)
- pixy.ccc.blocks[i].m_angle The angle of the object detected object if the detected object is a color code (-180 to 180).
- pixy.ccc.blocks[i].m_index The tracking index of the block
- pixy.ccc.blocks[i].m_age The number of frames the block has been tracked.
- pixy.ccc.blocks[i].print() A member function that prints the detected object information to the serial port

4 EQUIPMENT / TOOLS

a. Personal Computer
b. Arduino IDE
c. Arduino UNO R3
d. LN298N motor driver
e. Motor, simple DC motor model
f. Pixy2 camera

5 PROCEDURE

PART A: Image Processing Configuration

1. Download and install PixyMon software:

   https://pixycam.com/downloads-pixy2/

2. Connect Pixy2 camera to PC (micro USB)
3. Select Action > Clear all signatures
4. Select Action > Set signature 1
5. Point camera & Select object (s=1)
6. Stop program

PART B: Circuit diagram & Arduino setting

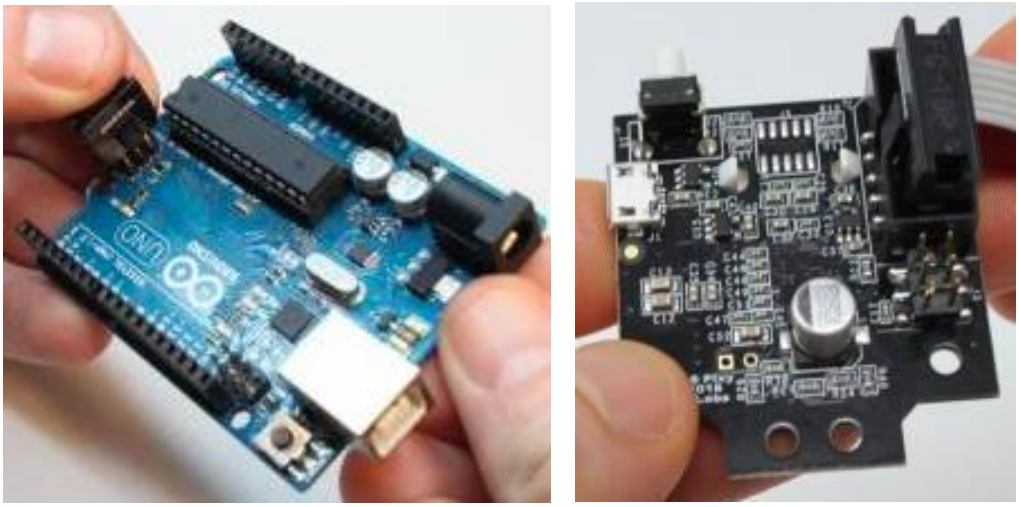


Figure 1: ICSP connection between Arduino UNO and Pixy2 camera

a) Construct the circuit Figure 1 base on the pin connection as stated on the Table 1

Table 1: Pin Diagram

Arduino UNO	L298N Motor Driver (1)	Left Motor (1)	Right Motor (2)	Pixy2
Digital pin 0 (RX)				
Digital pin 1 (TX)				
Digital pin 12				
Digital pin 11				
Digital pin 10				
Digital pin 7	IN1			
Digital pin 6	ENA			
Digital pin 5	ENB			
Digital pin 4	IN2			
Digital pin 3	IN3			
Digital pin 2	IN4			
	OUT4		RIGHT TERMINAL	
	OUT3		LEFT TERMINAL	
	OUT2	RIGHT TERMINAL		
	OUT1	LEFT TERMINAL		
GND	GND			
5V	5V			
3v3				
ICSP				ICSP

b) Download **arduino_pixy2-1.0.3 .zip**. Import Pixy library to Arduino IDE by selecting **Sketch** > **Include Library** > **Add .zip Library** > Browse & Select zip file.

c) Write a program given below and write the output result at Table 2.

```
//pw6 dec50122
```

```
//HW - uno, L298 x 1, pixy2 camera

#include <stdint.h>
#include <Pixy2.h> //utk guna pixy2 camera

// This is the main Pixy object
Pixy2 pixy;
int cont = 0;
int signature, x, y, width, height;
float cx, cy, area;

#define PWMA  6 //utk pwm motor driver
#define PWMB  5 //sama mc atas
#define IN1   7 //utk kawal dc motor
#define IN2   4 //sama mcm atas
#define IN3   3
#define IN4   2
#define Speed 200 //kelajuan dc motor. ubah kat sini

void setup() {  //setup function
  // put your setup code here, to run once:
  Serial.begin(9600); //start serial monitor with 9600 bit/s baud rate
  Serial.print("Starting...\n");  //for testing purposes
  pixy.init();
  delay(333);   //utk initialise serial communication
  pinMode(PWMA, OUTPUT);  //set pin as ouput
  pinMode(PWMB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
```

```
void loop() { //loop funtion
  // put your main code here, to run repeatedly:
  int x;
  pixy.ccc.getBlocks();// grab blocks!
  static int i = 0;
  int j;
  uint16_t blocks;
  char buf[32];
  blocks = pixy.ccc.getBlocks();// grab blocks!

  if (blocks)// If there are detect blocks, print them!
  {
    signature = pixy.ccc.blocks[0].m_signature;
    height = pixy.ccc.blocks[0].m_height;
    width = pixy.ccc.blocks[0].m_width;
    x = pixy.ccc.blocks[0].m_x;
    y = pixy.ccc.blocks[0].m_y;

    Serial.print("sig: ");
    Serial.print(signature);
    Serial.print(" x:");
    Serial.print(x);
    Serial.print(" y:");
    Serial.print(y);

        if(x<100)
          {
          kiri(); delay(100);
          }
        else if(x>200)
          {
```

```
          kanan(); delay(100);
          }
        else if(y<100)
          {
          depan(); delay(100);
          }
        else if(y>150)
          {
          belakang(); delay(100);
          }
        else if(x<=100&&x>=200&&y<=100&&y>=150)
          {
          henti(); delay(10);
          }
        else
          {
            henti(); delay(10);
          }
   }
}


float mapfloat(long x, long in_min, long in_max, long out_min, long
out_max)
{
  return (float)(x - in_min) * (out_max - out_min) / (float)(in_max -
in_min) + out_min;
}


void depan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
```

```
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void belakang() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kanan() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 0);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
}

void kiri() {
  analogWrite(PWMA, Speed);
  analogWrite(PWMB, Speed);
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 0);
}

void henti() {
```

```
  analogWrite(PWMA, 0);
  analogWrite(PWMB, 0);
  digitalWrite(IN1, 1);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 1);
  digitalWrite(IN4, 1);
}
```

d) Observe and discuss the results.

6 RESULT

Table 2: Results

Object Movement	Robot Movement
Static	
Close to robot	
Away from robot	
Right	
Left	

7 QUESTION

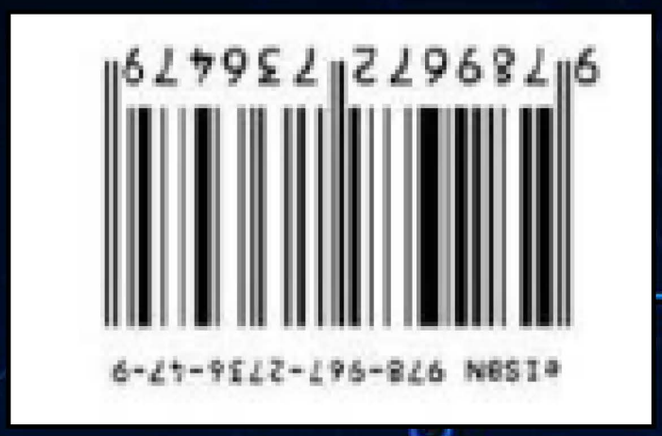
1. Give example application of robot vision.

8 DISCUSSION

9 CONCLUSION

REFERENCES

1. Alba-Flores, R. (2007). Laboratory Enhancements for Improving Embedded Systems Education. *Proceeding of the 2007 American Society for Engineering Educational Annual Conference & Exposition*. American Society for Engineering Education.
2. Cicolani, J. (2021). Beginning robotics with Raspberry Pi and Arduino: Using Python and OpenCV (2nd ed.). Apress.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
4. Jawaid, I., Javed, M. Y., Jaffery, M. H., Akram, A., Safder, U., & Hassan, S. (2020). Robotic system education for young children by collaborative-project-based learning. *Computer Applications in Engineering Education*, *28*(1), 178–192.
5. Joseph, L. (2022). Mastering ROS2 for robotics programming. Packt Publishing.
6. Larson, J., Jordan, S. S., Lande, M., & Weiner, S. (2020). Supporting Self-Directed Learning in a Project-Based Embedded Systems Design Course. *IEEE Transactions on Education*, *63*(2), 88–97.
7. Lynch, K. M., & Park, F. C. (2017). Modern robotics: Mechanics, planning, and control. Cambridge University Press.
8. Margolis, M., et al. (2021). Arduino cookbook (3rd ed.). O'Reilly Media.
9. Rosa, A. H. R., Ferreira, R. V., & Pereira, C. A. (2021). Integrated PBL and HIL practices for real-time simulations applied in technical and engineering teaching using embedded systems. *Przeglad Elektrotechniczny*, *97*(1), 46–52.
10. SSiciliano, B., & Khatib, O. (Eds.). (2023). Springer handbook of robotics (3rd ed.). Springer Nature.
11. Tian, J. (2021). Optimization of Embedded Mobile Teaching Model Based on Network Streaming Media Technology. *Complexity*, *2021*(4).
12. Uzun, A. (2020). Using Educational Robotics as a Cognitive Tool for ICT Teachers in an Authentic Learning Environment. *International Education Studies*, *13*(4), 27.
13. Valvano, J. (2023). Embedded systems with ARM Cortex-M microcontrollers (latest revision editions used in courses)


9789672736479
eISBN 978-967-2736-47-9