AI Lab-06 Tasks

Task 1:

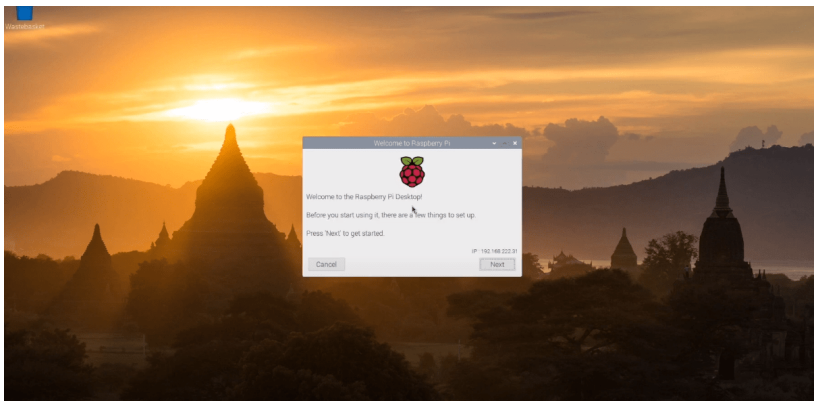
```
const int LEFT_SENSOR_PIN = A0;   // define the left sensor pin
const int RIGHT_SENSOR_PIN = A1;  // define the right sensor pin
const int LEFT_MOTOR = 5;         // define the left motor pin
const int RIGHT_MOTOR = 6;        // define the right motor pin

void setup() {
  pinMode(LEFT_SENSOR_PIN, INPUT);   // set the left sensor pin as an input
  pinMode(RIGHT_SENSOR_PIN, INPUT);  // set the right sensor pin as an input
  pinMode(LEFT_MOTOR, OUTPUT);       // set the left motor pin as an output
  pinMode(RIGHT_MOTOR, OUTPUT);      // set the right motor pin as an output
}

void loop() {
  int leftSensorValue = analogRead(LEFT_SENSOR_PIN);    // read the left sensor
value
  int rightSensorValue = analogRead(RIGHT_SENSOR_PIN);  // read the right sensor
value

  // adjust the motor speeds based on the sensor values
  if (leftSensorValue > 500 && rightSensorValue > 500) {
    // both sensors detect white, stop the motors
    digitalWrite(LEFT_MOTOR, LOW);
    digitalWrite(RIGHT_MOTOR, LOW);
  } else if (leftSensorValue > 500) {
    // left sensor detects white, turn right
    digitalWrite(LEFT_MOTOR, HIGH);
    digitalWrite(RIGHT_MOTOR, LOW);
  } else if (rightSensorValue > 500) {
    // right sensor detects white, turn left
    digitalWrite(LEFT_MOTOR, LOW);
    digitalWrite(RIGHT_MOTOR, HIGH);
  } else {
    // both sensors detect black, move forward
    digitalWrite(LEFT_MOTOR, HIGH);
    digitalWrite(RIGHT_MOTOR, HIGH);
  }
}
```

Task 2:

```
const int LED_PIN = 9;  // define the LED pin
```

```
int brightness = 0;     // initial brightness value
int fadeAmount = 5;     // amount to fade the LED by

void setup() {
  pinMode(LED_PIN, OUTPUT);  // set the LED pin as an output
}

void loop() {
  analogWrite(LED_PIN, brightness);  // set the LED brightness using PWM
  brightness += fadeAmount;          // change the brightness by the fade amount

  // reverse the fade direction if brightness reaches 0 or 255
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }

  delay(30);  // wait a short time to see the effect
}
```

Task 3:



Task 4:

```
import RPi.GPIO as GPIO
import time

# set the GPIO mode to BCM
GPIO.setmode(GPIO.BCM)

# define the GPIO pins for the LEDs
led_pins = [17, 18, 27, 22]

# set up the GPIO pins as outputs
for pin in led_pins:
```

```
    GPIO.setup(pin, GPIO.OUT)

# define the values to write to the LEDs
led_values = [
    [1, 0, 0, 0],  # turn on LED 1
    [0, 1, 0, 0],  # turn on LED 2
    [0, 0, 1, 0],  # turn on LED 3
    [0, 0, 0, 1]   # turn on LED 4
]

# loop through the LED values and write them to the GPIO pins
for value in led_values:
    for i in range(len(led_pins)):
        GPIO.output(led_pins[i], value[i])
    time.sleep(3)  # delay for 3 seconds

# cleanup the GPIO pins
GPIO.cleanup()
```

Task 5:

Raspberry Pi and Arduino are both popular platforms used for creating electronic projects and IoT devices. Although they have some similarities, they have distinct technical characteristics that make them suitable for specific tasks.

The Raspberry Pi is a complete computer that runs on a Linux-based operating system, with a powerful CPU, GPU, and other hardware components that enable it to run a wide range of software programs. Due to its versatility, it is used for various projects such as media centers, gaming consoles, and web servers.

On the other hand, the Arduino is a microcontroller platform that is designed for simple and repeated operations. It consumes less energy compared to the Raspberry Pi and is ideal for real-time control, sensing, and home automation projects.

One of the primary differences between Raspberry Pi and Arduino is their processing power. The Raspberry Pi's more potent CPU and GPU make it suitable for executing complex software programs and processing vast amounts of data. The Arduino, on the other hand, has a less complex microprocessor that is ideal for straightforward operations and real-time control.

Furthermore, their interactions with the outside world are different. The Raspberry Pi offers various input and output options, such as HDMI, USB, Ethernet, and GPIO pins, which facilitate sensor and device connectivity. Meanwhile, the Arduino has fewer input and output options but provides a wide range of analog and digital pins for sensing and control.

Both platforms can be used for creating AI-enabled gadgets, but they serve different purposes. The Raspberry Pi is suitable for AI applications that involve data processing and machine learning, such as smart cameras and chatbots. On the other hand, the Arduino is more appropriate for real-time control and sensing applications, such as smart home appliances and robots.

Finally, the combination of Raspberry Pi and Arduino can be used for creating a smart greenhouse project that utilizes both platforms for AI. The Raspberry Pi is used for monitoring environmental information, while the Arduino operates the watering and fertilizer systems in real-time based on the data gathered by the Raspberry Pi.