##########
Assignment
##########

Assignment 2
****

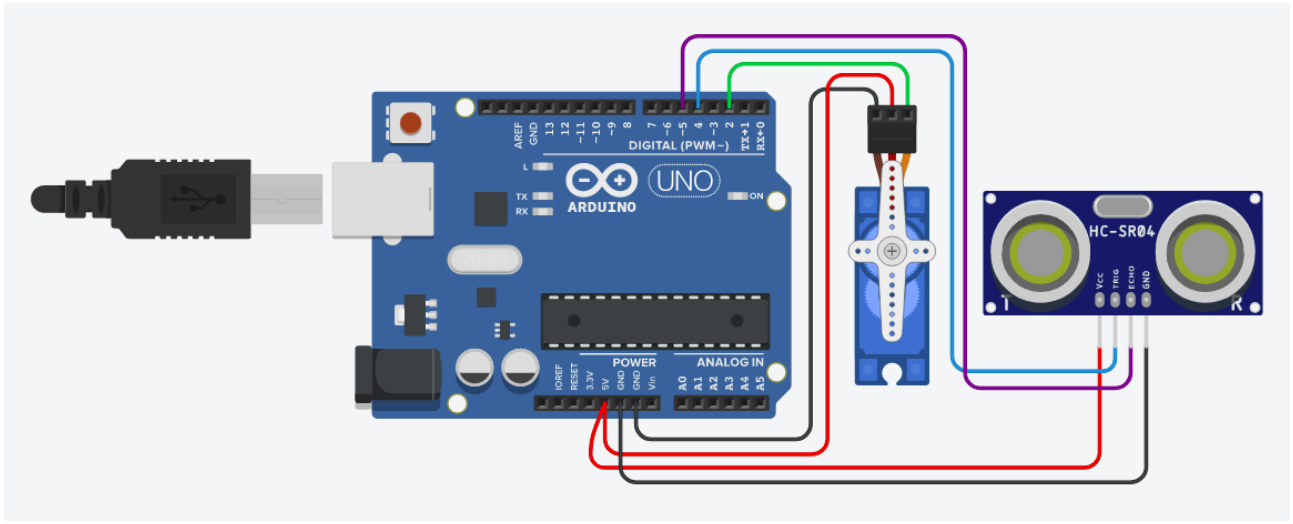
:Question: Automatic garage door opening system.

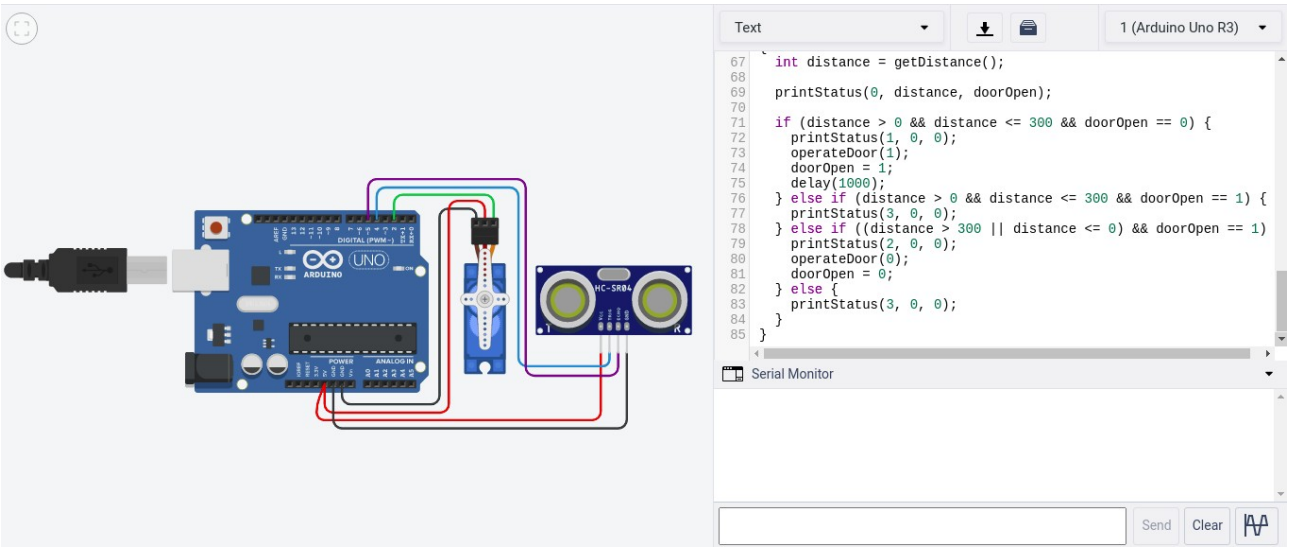
Code
====
Arduino C++ code::

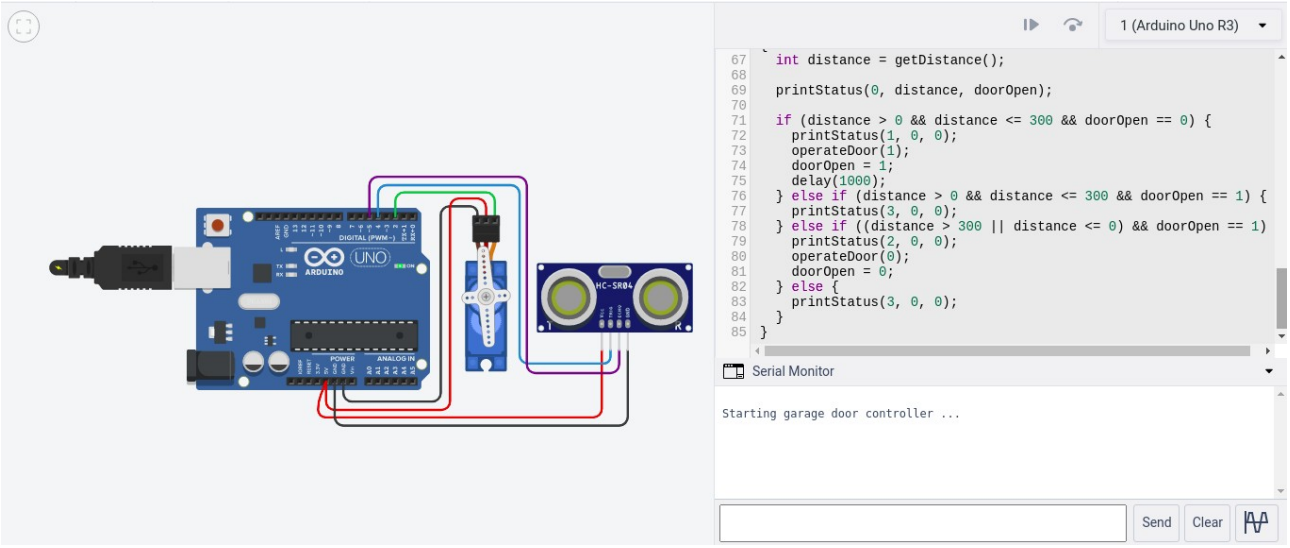
```
  /**
    * author  : Arunesh Gour.
    * reg. no.: 18BCG10024.
    */

  #include <Servo.h>
  Servo servo;

  int doorOpen = 0; // Door state: 0=close, 1=open.

  void setup()
  {
    pinMode(2, OUTPUT); // Servo
    pinMode(4, OUTPUT); // Ultrasonic.Trigger
    pinMode(5, INPUT); // Ultrasonic.Echo

    servo.attach(2); // Connecting servo.
    servo.write(0); // Closing door.
    digitalWrite(4, LOW); // Initializing trigger.
    Serial.begin(9600);

    Serial.println("");
    Serial.println("Starting garage door controller ...");
  }

  int getDistance () {
    digitalWrite(4, LOW);
    digitalWrite(4, HIGH);
    delayMicroseconds(10);
    digitalWrite(4, LOW);
    int duration = pulseIn(5, HIGH);
    int distance = (duration * 0.0343) / 2;

    return distance;
  }

  void operateDoor (int state) {
    // Assuming at 0deg, gate is closed,
    // while at 90deg, gate is opened.
    if (state == 0) {
```

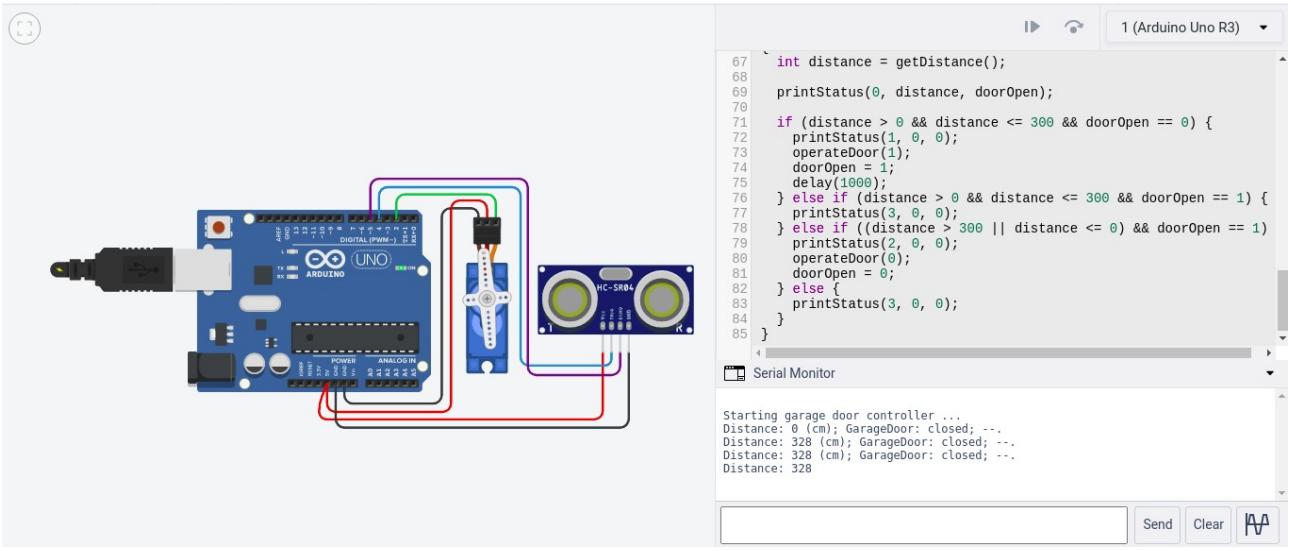
```
    servo.write(0); // Closing door.
    delay(1500);
  } else if (state == 1) {
    servo.write(90); // Opening door.
    delay(1500);
  }
}

void printStatus (int stage, int distance, int doorState) {
  if (stage == 0) {
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.print(" (cm); GarageDoor: ");
    if (doorState == 0) {
      Serial.print("closed; ");
    } else if (doorState == 1) {
      Serial.print("open; ");
    }
  } else if (stage == 1) {
    Serial.println("Opening door.");
  } else if (stage == 2) {
    Serial.println("Closing door.");
  } else if (stage == 3) {
    Serial.println("--.");
  }
}

void loop()
{
  int distance = getDistance();

  printStatus(0, distance, doorOpen);

  if (distance > 0 && distance <= 300 && doorOpen == 0) {
    printStatus(1, 0, 0);
    operateDoor(1);
    doorOpen = 1;
    delay(1000);
  } else if (distance > 0 && distance <= 300 && doorOpen == 1) {
    printStatus(3, 0, 0);
  } else if ((distance > 300 || distance <= 0) && doorOpen == 1) {
    printStatus(2, 0, 0);
    operateDoor(0);
    doorOpen = 0;
  } else {
    printStatus(3, 0, 0);
  }
}
```

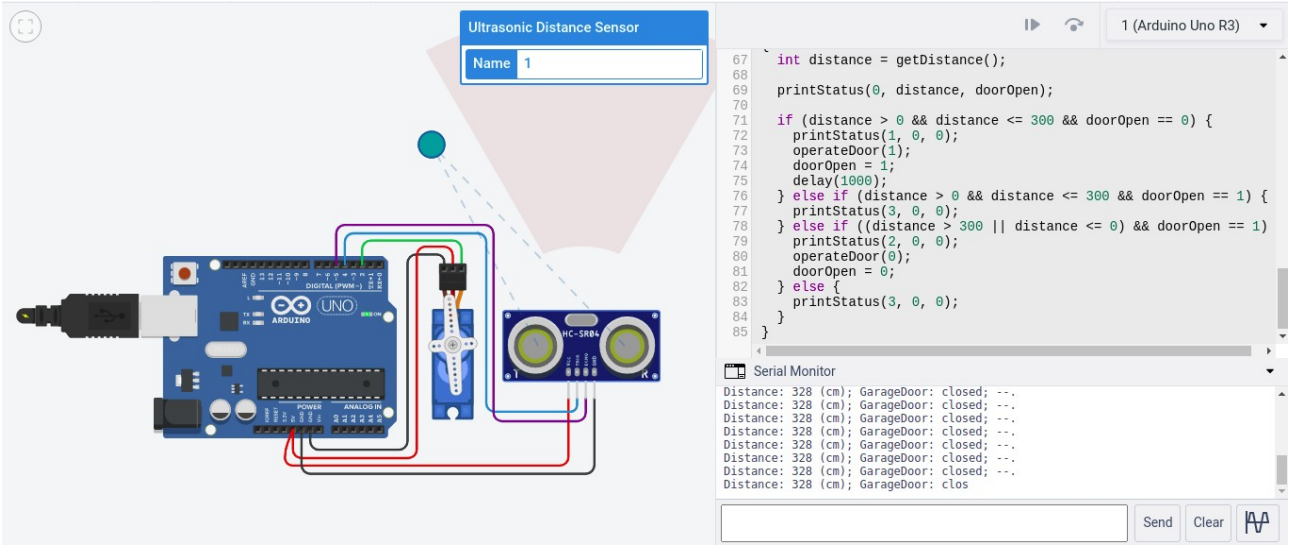
Output
======

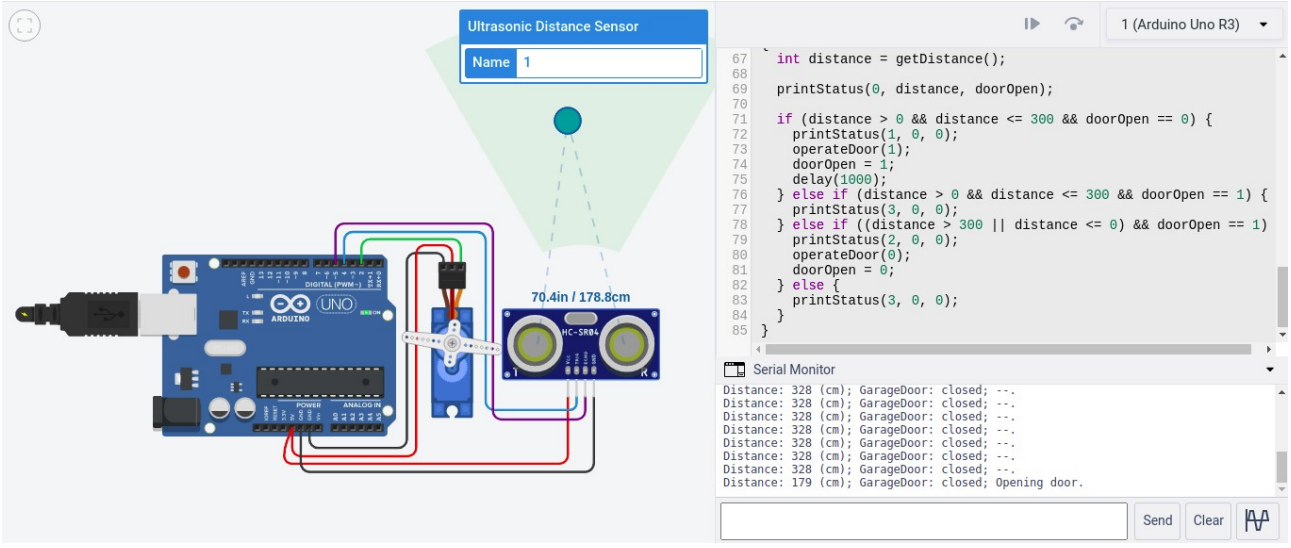

DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
HC-SR04
VCC
TRIG
ECHO
GND

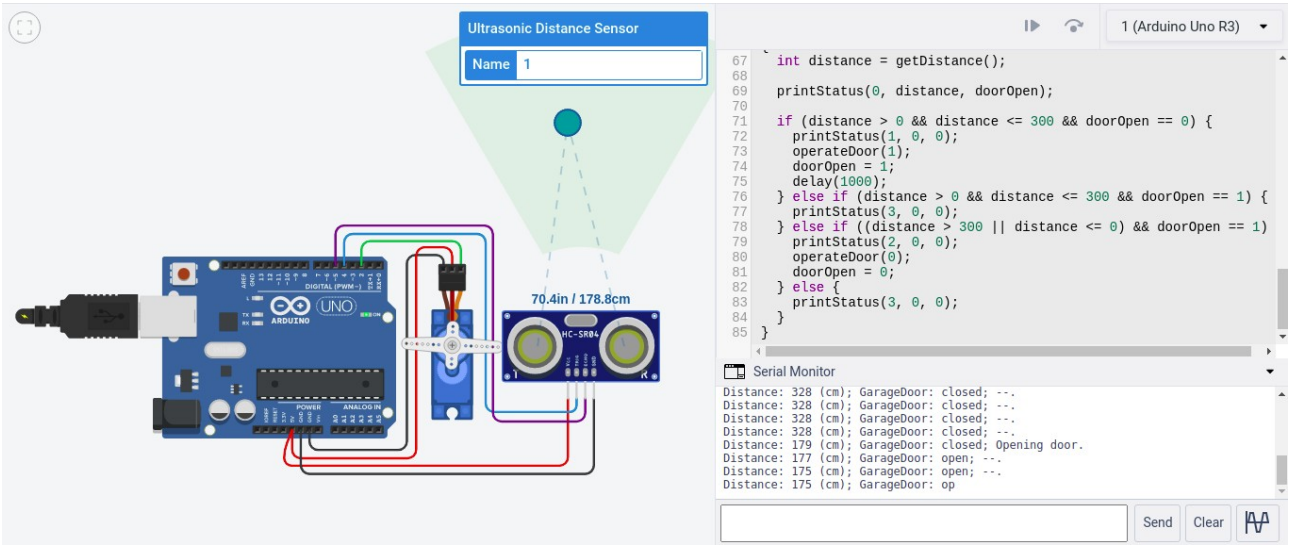

Text
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Send
Clear

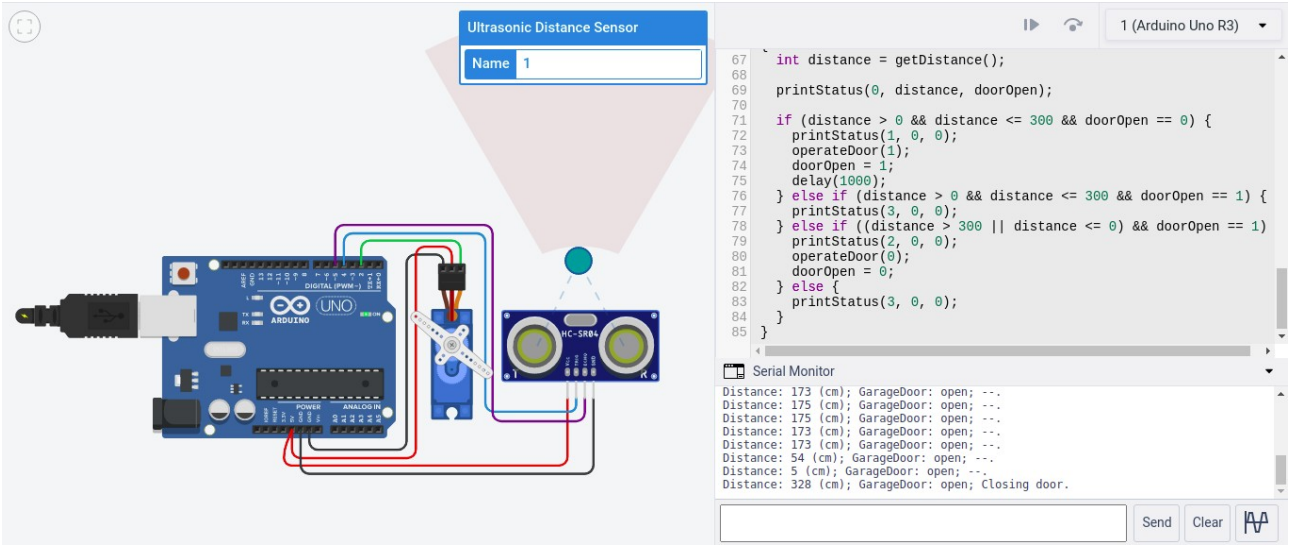

1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Starting garage door controller ...
Send
Clear

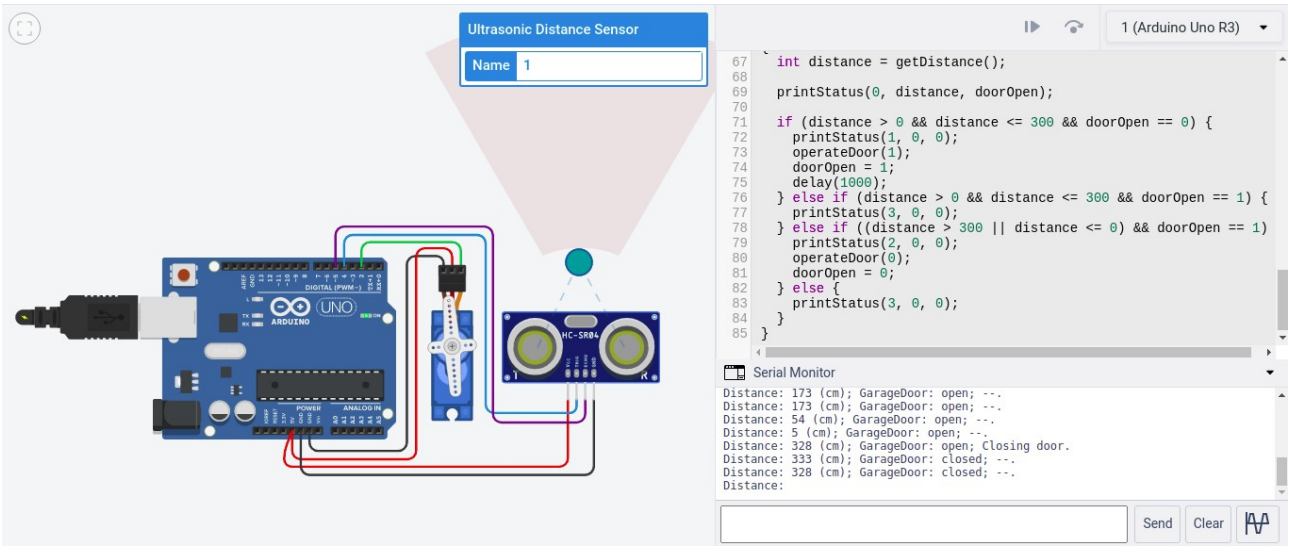

1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Starting garage door controller ...
Distance: 0 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328
Send
Clear


Ultrasonic Distance Sensor
Name 1
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: clos
Send
Clear


Ultrasonic Distance Sensor
Name 1
70.4in / 178.8cm
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 179 (cm); GarageDoor: closed; Opening door.
Send
Clear


Ultrasonic Distance Sensor
Name 1
70.4in / 178.8cm
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance: 179 (cm); GarageDoor: closed; Opening door.
Distance: 177 (cm); GarageDoor: open; --.
Distance: 175 (cm); GarageDoor: open; --.
Distance: 175 (cm); GarageDoor: op
Send
Clear


Ultrasonic Distance Sensor
Name 1
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Distance: 173 (cm); GarageDoor: open; --.
Distance: 175 (cm); GarageDoor: open; --.
Distance: 175 (cm); GarageDoor: open; --.
Distance: 173 (cm); GarageDoor: open; --.
Distance: 173 (cm); GarageDoor: open; --.
Distance: 54 (cm); GarageDoor: open; --.
Distance: 5 (cm); GarageDoor: open; --.
Distance: 328 (cm); GarageDoor: open; Closing door.
Send
Clear


Ultrasonic Distance Sensor
Name 1
1 (Arduino Uno R3)
int distance = getDistance();
printStatus(0, distance, doorOpen);
if (distance > 0 && distance <= 300 && doorOpen == 0) {
printStatus(1, 0, 0);
operateDoor(1);
doorOpen = 1;
delay(1000);
} else if (distance > 0 && distance <= 300 && doorOpen == 1) {
printStatus(3, 0, 0);
} else if ((distance > 300 || distance <= 0) && doorOpen == 1)
printStatus(2, 0, 0);
operateDoor(0);
doorOpen = 0;
} else {
printStatus(3, 0, 0);
}
}
Serial Monitor
Distance: 173 (cm); GarageDoor: open; --.
Distance: 173 (cm); GarageDoor: open; --.
Distance: 54 (cm); GarageDoor: open; --.
Distance: 5 (cm); GarageDoor: open; --.
Distance: 328 (cm); GarageDoor: open; Closing door.
Distance: 333 (cm); GarageDoor: closed; --.
Distance: 328 (cm); GarageDoor: closed; --.
Distance:
Send
Clear