Indian Institute of Technology Hyderabad

Circuits Lab, 2023-24

Report On

Automatic Door Opening System

Project by:
EE23BTech11016,
EE23BTech11011,
EE23BTech11008,
EE23BTech11018
Instructors:
DR.SIVA RAMA KRISHNA VANJARI,
MR. THIRUMURUGAN

Abstract

The automatic door opening system project aims to design and implement a mechanism for opening doors without manual intervention. In this project, we propose a system that utilizes sensors to detect the presence of individuals near the door and automatically triggers the opening mechanism. The system incorporates infrared sensors to detect motion and actuates the door-opening mechanism accordingly. Additionally, we integrate a microcontroller, Arduino, to process sensor inputs and control the door-opening mechanism. The project aims to enhance convenience, accessibility, and efficiency in various environments, including commercial buildings, hospitals, and homes. Through this project, we demonstrate the feasibility and effectiveness of an automatic door opening system in improving accessibility and convenience while reducing the need for manual operation.

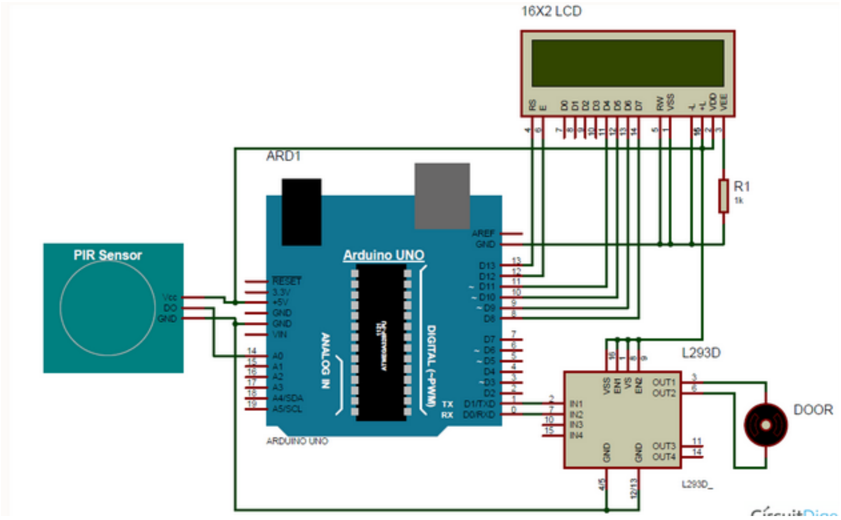
1 Introduction

Automatic door opening systems have become increasingly prevalent in various environments, ranging from commercial buildings and hospitals to homes and public spaces. These systems offer numerous benefits, including enhanced accessibility, improved convenience, and increased energy efficiency. The primary objective of this project is to design and implement an automatic door opening system that detects the presence of individuals near the door and triggers the opening mechanism without the need for manual intervention.

2 Materials and Methods

2.1 Apparatus

1. Arduino UNO
2. 16X2 LCD
3. PIR Sensor
4. Connecting wires
5. Bread board
6. 1 k resistor
7. Power supply
8. Motor driver
9. DC Motor



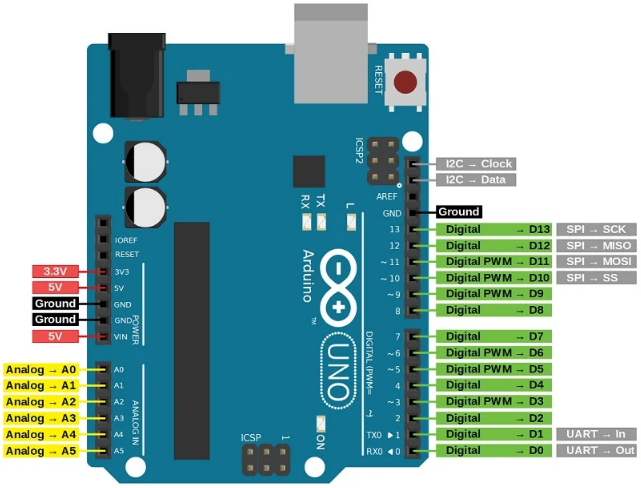
2.2 Detailed study of each component

2.2.1 Arduino UNO

The core component of the system, responsible for processing sensor inputs and controlling the door-opening mechanism.
The connections made from it are as shown below:

Section	Pins in ARD	Other Device	Pins in other device	Pin number (if any)
Power	+5V	PIR	V_{CC}	-
		LCD	+L, VDD	15, 2
		L293D	V_{CC1}, V_{CC2},EN1, EN2	16, 8, 1, 9
GND	GND	PIR	GND	-
		L293D	GND	4 and 12 OR 5 and 13
		LCD	RW, V_{SS}, -L, V_{EE}	5, 1, 16, 3
Analog	A0	PIR	Digital output	-
Digital	D13, D12, D11 TO D8	LCD	RS, Enable, D4 TO D7	4, 6, 11 TO 14
	D1/TXD, D0/RXD	L293D	IN1, IN2	2,7

Pin diagram of ARD is as shown below:



CONNECTIONS:

The Arduino consists of three main sections: Power, Analog, and Digital.

1. **Power Section**

   - It includes the reset pin (not utilized here), voltage pin, ground, and AREF (analog reference, not used here).
   - The 5V pin is connected to the power pins of the PIR sensor, LCD, and motor driver. The ground pin is connected to the respective grounds of these components. Other connections from this section are outlined below.
   - The backlight connections of the LCD, denoted as +L and -L, are connected to +5V and ground respectively. These connections power the backlight to illuminate the LCD panel.
   - The RW (read/write) pin of the LCD is permanently connected to ground on the Arduino, allowing write operations (sending commands).
   - Another connection involves linking the ground of the Arduino to V_{EE} of the LCD to control the display contrast. A 1k resistor is included in this connection to optimize contrast.
   - Additional connections to the motor driver IC (L293D) include the enable pins, ensuring the motor driver remains enabled whenever the Arduino is powered on. This enables motor power whenever the PIR sensor outputs 1.

2. **Analog Section**

   The only connection made in this section is to the PIR sensor, with its digital output connected to the A0 pin of the Arduino. This connection allows the Arduino to receive and process analog signals from the sensor.

3. **Digital Section**
   These connections can be categorized into three groups:
   - D13 and D12 of the Arduino are connected to RS and Enable pins of the LCD respectively. RS determines whether incoming data is interpreted as command or character data for display.
   - Connections from D11 to D8 of the Arduino are respectively linked to D4 to D7 of the LCD for later use.
   - Finally, D1/TXD (transmitting) and D0/RXD (receiving) of the Arduino are connected to IN1 and IN2 of the Motor Driver respectively. These connections control the motor direction based on the logic levels of IN1 and IN2.

TESTING:

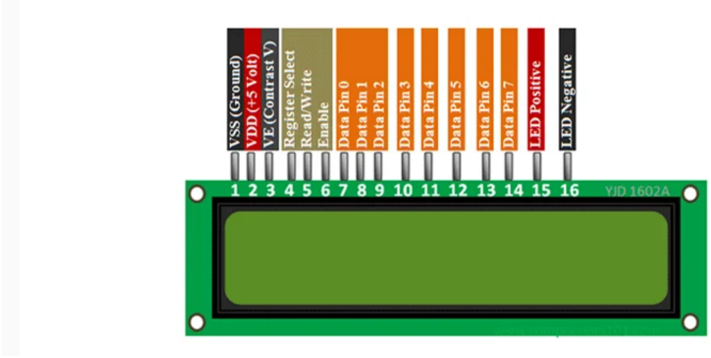
1. Begin by connecting the Arduino board to a laptop.
2. Open the Arduino IDE and test the board using the example code for blinking an LED.
3. Verify that the LED blinks as expected, indicating proper functionality of the Arduino board.

2.2.2 16X2 LCD

In an automatic door opening system, a 16x2 LCD (Liquid Crystal Display) can be used as a user interface to provide feedback, display status information, and enable user interaction.
The connections made from it are as shown below:

Pins in LCD	**Other Device**	**Pins in other device**	**Pin number (if any)**
V_{SS}, RW, -L, V_{EE}	ARD	GND	-
V_{DD}, +L	ARD	+5V	-
RS, E, D4 TO D7	ARD	D13 TO D8	-

Pin diagram of LCD is as shown below:



THEORY/PIN ANALYSIS:

1. VSS (Ground): This pin is connected to ground (0V) and serves as the ground reference for the LCD.
2. VDD (Supply Voltage): This pin is connected to a positive supply voltage (usually +5V) and provides power to the LCD.
3. VEE (Contrast Adjustment): This pin is used to adjust the contrast of the display. By applying a variable voltage to this pin, the contrast between the characters and the background can be adjusted.
4. RS (Register Select): This pin is used to select between data and command modes. When RS is LOW, the data sent to the LCD is treated as a command (e.g., clear screen, move cursor). When RS is HIGH, the data sent is treated as character data (to be displayed on the screen).
5. R/W (Read/Write): This pin is used to control the direction of data flow. When R/W is LOW, data is written to the LCD. When R/W is HIGH, data is read from the LCD. In most applications, R/W is connected to ground to enable write mode only.
6. E (Enable): This pin is used to enable the LCD module. When a HIGH-to-LOW pulse is applied to this pin, it triggers the LCD to execute the command or display the data present on the data bus.
7. D0-D7 (Data Lines): These are the eight data lines used to send data and commands to the LCD. In most applications, only four of these data lines (D4-D7) are used in 4-bit mode, with the remaining four being left unconnected or grounded. However, in 8-bit mode, all eight data lines are used.
8. L+ (Anode): This pin is connected to the anode of the LED backlight, providing power to illuminate the display.
9. L- (Cathode): This pin is connected to the cathode of the LED backlight, completing the circuit for illumination when connected to ground.

CONNECTIONS:

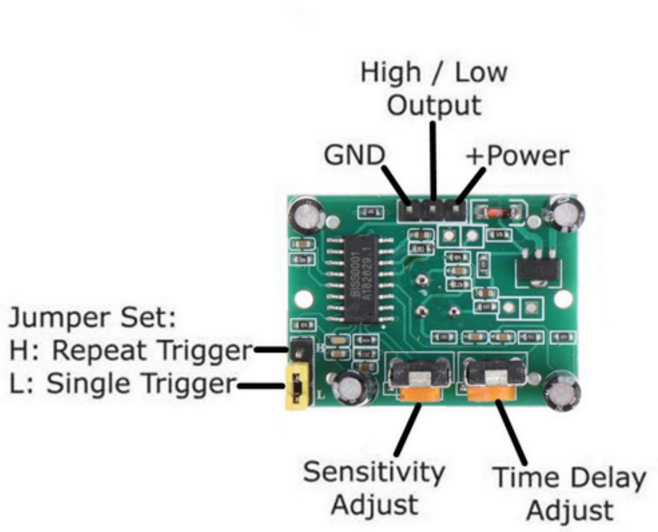
All connections to this device are established through the Arduino board, as detailed in the preceding section.

2.2.3 PIR Sensor

In an automatic door opening system project, a PIR (Passive Infrared) sensor plays a crucial role in detecting motion and triggering the door's opening mechanism.
The connections made from it are as shown below:

Pins in PIR	**Other Device**	**Pins in other device**	**Pin number (if any)**
Power Supply (V_{CC})	ARD	+5V (power)	-
	LCD	+L, VDD	15, 2
	L293D	V_{SS}, EN1, V_S, EN2	16, 1, 9, 8
Digital Output (DO)	ARD	Analog Output (analog)	-
Ground (GND)	ARD	GND (power)	-
	L293D	GND	5 and 12

Pin diagram of PIR Sensor is as shown below:

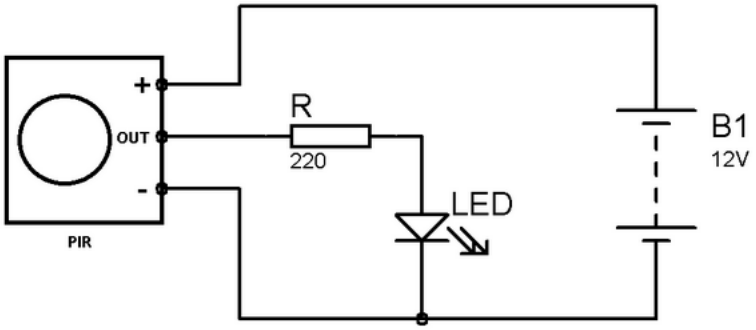


THEORY/PIN ANALYSIS:

1. Vcc: This pin is used to provide power to the PIR sensor. It typically requires a voltage supply within a specified range, often between 3V to 5V, depending on the sensor model.
2. GND: This pin is the ground reference for the PIR sensor and should be connected to the ground of the power supply.
3. Output (OUT): The output pin of the PIR sensor provides a digital signal indicating the presence or absence of motion detected by the sensor. When motion is detected within its detection range, the output goes HIGH (or LOW, depending on the sensor). When there is no motion, the output remains LOW (or HIGH).

CONNECTIONS:
All connections to this device are established through the Arduino board, as detailed in the preceding section.
TESTING

1. Connect the circuit as shown below.



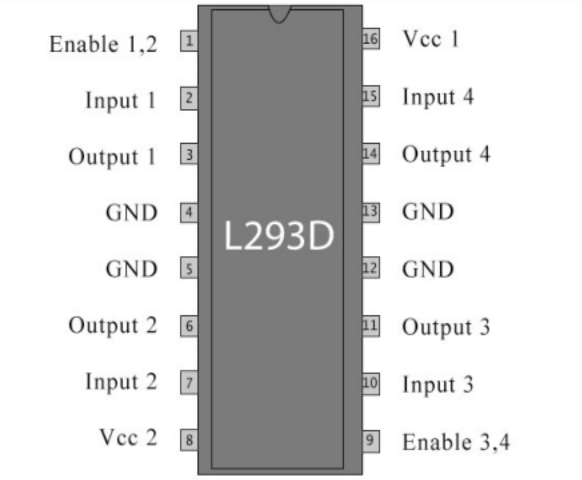
2. Place your hand near the sensor. It passes the test if LED glows.

2.2.4 Motor Driver: L293D (IC)

In an automatic door opening system project, the L293D motor driver IC is used to control the movement of the door mechanism, providing the necessary power and direction control for the motor.

Pins in IC	**Other Device**	**Pins in other device**	**Pin number (if any)**
Enables, and V_{cc}(1,8,9,16)	ARD	+5V (power)	-
Outputs(3,6)	DC Motor	-	-
Inputs(2,7)	ARD	D1, D0 resp.	-
GND(4,12)	ARD	GND	-

Pin diagram of L293D is as shown below:



THEORY/PIN ANALYSIS:

1. Enable 1, 2, 3, 4 (EN1, EN2, EN3, EN4): These pins are used to enable or disable the corresponding half-H drivers. When these pins are set HIGH, the respective half-H driver is enabled, allowing it to operate. When set LOW, the driver is disabled, effectively turning off the motor outputs.

2. Input 1, 2, 3, 4 (IN1, IN2, IN3, IN4): These pins are the input control pins for the respective half-H drivers. The combination of HIGH and LOW signals on these pins determines the direction and speed of the motor connected to the corresponding output.

3. Output 1, 2, 3, 4 (OUT1, OUT2, OUT3, OUT4): These pins are the outputs for the respective half-H drivers. They are connected to the motor terminals and provide the driving voltage and current to the motor.

4. Vcc1, Vcc2 (Vs, Vss): These pins provide the power supply voltage for the internal logic circuitry (Vcc1) and the motor supply voltage (Vcc2). Vcc1 typically ranges from 4.5V to 7V, while Vcc2 should match the voltage requirements of the connected motors.

5. Ground (GND): This pin is the ground reference for both the internal logic and the motor supply voltage.

CONNECTIONS:

1. Most of the connections to this device are facilitated by the Arduino (ARD), as previously outlined.

2. Additionally, OUT1 and OUT2 from the Arduino are connected to the motor, enabling its movement.

2.2.5 DC Motor:

(a) **Power and Control:** DC motors provide the necessary power to open and close doors of different sizes and weights. They can be controlled using electronic circuits, microcontrollers, or PLCs, making them suitable for automation.

(b) **Speed Control:** DC motors allow precise speed control, essential for opening and closing doors at different speeds based on environment, traffic flow, or safety needs.

(c) **Direction Control:** DC motors change rotation direction, enabling them to open and close doors as required, a crucial feature in automatic door systems.

(d) **Efficiency and Reliability:** DC motors efficiently convert electrical to mechanical energy, reducing energy waste and operating costs. They offer reliability for continuous operation in automatic doors.

(e) **Integration with Control Systems:** DC motors integrate seamlessly into complex control systems for features like remote operation, access control integration, and synchronization with other automation systems.

Overall, DC motors in automatic door openers provide efficiency, precise control, reliability, and integration with automation technologies, making them ideal for residential, commercial, and industrial applications.

2.3 Programming

2.3.1 Code

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

#define PIR_sensor 14

#define m11 0

#define m12 1

void setup()

{

  lcd.begin(16, 2);
  pinMode(m11, OUTPUT);
  pinMode(m12, OUTPUT);
  pinMode(PIR_sensor, INPUT);
  lcd.print("    Circuits    ");
  lcd.setCursor(0,1);
  lcd.print("   Lab Project   ");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.clear();
  lcd.print("    Automatic    ");
  lcd.setCursor(0,1);
  lcd.print("   Door Opener   ");
  delay(3000);
  lcd.clear();
  delay(2000);
}
void loop()
{
  if(digitalRead(PIR_sensor))
  {
    lcd.clear();
    lcd.print("    Gate Opened    ");
  digitalWrite(m11, HIGH);          // gate opening
  digitalWrite(m12, LOW);
  delay(1000);
  digitalWrite(m11, LOW);           // gate stop for a while
  digitalWrite(m12, LOW);
  lcd.clear();
  lcd.print(”␣Closed␣”);
  digitalWrite(m11, LOW);            // gate closing
  digitalWrite(m12, HIGH);
  delay(1000);
  digitalWrite(m11, LOW);             // gate closed
  digitalWrite(m12, LOW);
  delay(1000);
}

else
{

  lcd.setCursor(0,0);
  lcd.print(”␣␣␣No␣Movement␣␣␣”);
  lcd.setCursor(0,1);
  lcd.print(”␣␣␣Gate␣Closed␣␣␣”);
  digitalWrite(m11, LOW);
  digitalWrite(m12, LOW);
  }

}
```

2.3.2 Explaination

(a) **Library Inclusion**:

- The code includes the necessary libraries, such as LiquidCrystal.h, to interface with the LCD display.

(b) **Pin Definition**:

- Pin assignments for the LCD, PIR sensor, and gate motor are defined using preprocessor directives.

(c) **Setup Function**:

- The setup function initializes the LCD and configures pins for input and output.
- Initial messages are displayed on the LCD to indicate system readiness.

(d) **Loop Function**:

- The loop function continuously monitors the PIR sensor for motion detection.
- Upon detecting motion, the system activates the gate motor to open the gate and displays relevant messages on the LCD.
- After a delay, the gate motor is stopped temporarily before closing the gate.
- The system repeats this process as long as motion is detected.

2.4 Sequential Mechanism

The automatic door opening system operates through the following sequence:

(a) **Initialization**:

- The system powers on, initializing all components.
- The LCD display shows an initial message indicating system readiness, such as ”Automatic Door Opener.”

(b) **Monitoring for Motion**:

- The PIR sensor continuously monitors its surroundings for motion.

(c) **Detection of Motion**:

- Upon detecting motion, the PIR sensor sends a signal to the Arduino Uno.

(d) **Decision Making**:

- The Arduino Uno processes the signal from the PIR sensor to determine if someone is approaching the door.
- If motion is detected, the Arduino Uno decides to initiate the door-opening sequence.

(e) **Initiating Door Opening**:

- The Arduino Uno sends signals to the motor driver to activate the door-opening motor.

(f) **Door Opening**:

- The motor rotates, smoothly opening the door.
- The LCD display shows a message indicating that the door is opening.

(g) **Monitoring Door Position**:

- The system continuously monitors the position of the door to ensure it reaches the fully open position.

(h) **Door Fully Opened**:

- Once the door reaches the fully open position, the motor stops, and the Arduino Uno receives confirmation.

(i) **Waiting for Motion to Cease**:

- If no motion is detected for a predefined period, the system waits for motion to cease.

(j) **Detection of No Motion**:

- If no motion is detected for the predefined period, the PIR sensor sends a signal to the Arduino Uno indicating the absence of motion.

(k) **Closing the Door**:

- Upon receiving the signal indicating the absence of motion, the Arduino Uno initiates the door-closing sequence.
- It sends signals to the motor driver to activate the motor in the reverse direction, causing the door to close.

(l) **Door Closing**:

- The motor rotates in the reverse direction, closing the door.
- The LCD display shows a message indicating that the door is closing.

(m) **Monitoring Door Position**:

- The system continuously monitors the position of the door to ensure it reaches the fully closed position.

(n) **Door Fully Closed**:

- Once the door reaches the fully closed position, the motor stops, and the Arduino Uno receives confirmation.

(o) **Idle State**:

- The system returns to an idle state, ready to detect motion and initiate the door-opening sequence again when necessary.

NOTE: We haven't connected a door in here. So the clockwise and anticlockwise rotation of door shows the opening and closing of door.

3 Conclusion

In conclusion, the Automatic Door Opener project using Arduino Uno has demonstrated the successful implementation of an electronic system for automating door operations.

(a) **ADVANTAGES:**

i. Functionality: The system effectively detects approaching individuals using the pir sensor and activates the DC motor to open the door, providing a hands-free experience.

ii. Arduino Uno Integration: By utilizing the Arduino Uno microcontroller, we were able to program the logic and control the hardware components.

(b) **LIMITATIONS:**

i. Range Limitation: The pir sensor's range may be limited in certain environments or weather conditions, affecting the system's reliability at longer distances.

ii. Speed and Responsiveness: Depending on the DC motor's specifications and programming, the door's opening and closing speed may need optimization for smoother operation.

iii. Safety Considerations: While the system improves convenience, ensuring safety measures, such as obstacle detection and emergency stop mechanisms, is critical for real-world deployment.

4 References

https://circuitdigest.com/microcontroller-projects/automatic-door-opener-project-using-arduino