Bluetooth door

Made by:

Name	ID
Mohamed Ali Sayed	191900165
Islam Osama	191900044
Kirolos Osama	191900016
Abanoub Ibrahim	191900276

Abstraction:

A controlled door from telephone using Bluetooth sensor is a project that involves using a Bluetooth module to remotely control a door lock from a mobile phone. This project is useful in situations where you want to provide secure access to a room or building while also having the ability to control access remotely.

Objectives:

a controlled door from telephone using Bluetooth sensor is a simple and effective solution for providing secure access control to a room or building while also allowing for remote control via a mobile phone.

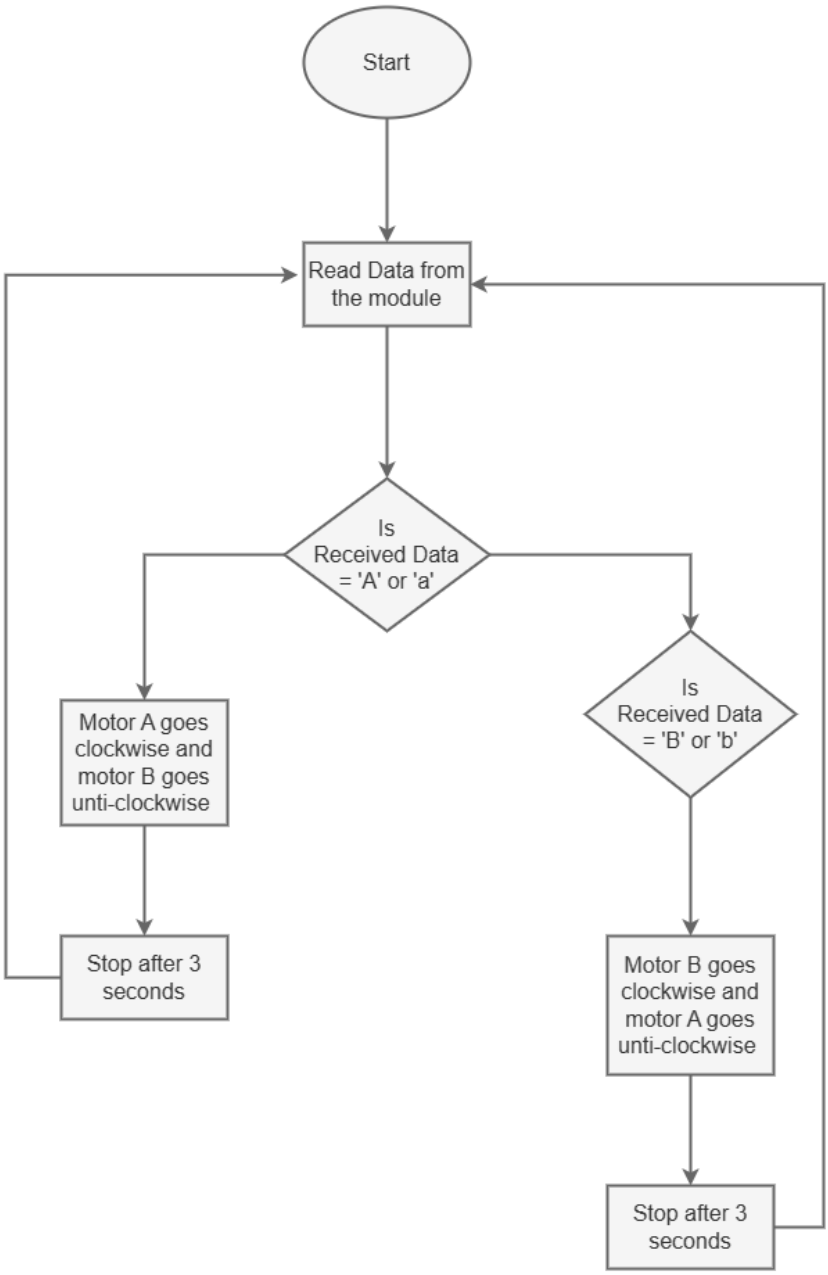
Methodology:

To implement this project, you will need to connect a Bluetooth module to the Atmega32 microcontroller and then program the microcontroller to read commands from a mobile phone and control a door lock accordingly. You will also need to install a door lock mechanism that can be controlled by the microcontroller

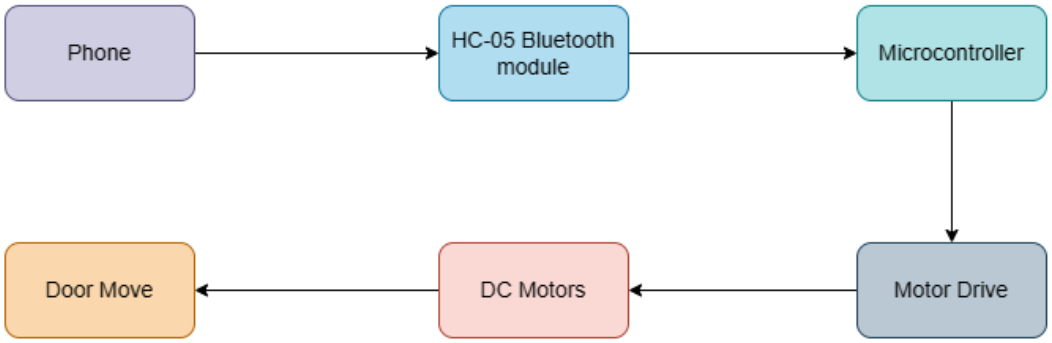
Tools:

- Atmega32
- HC-05 Bluetooth module
- Small door
- Motor Drive L298N
- DC Motor

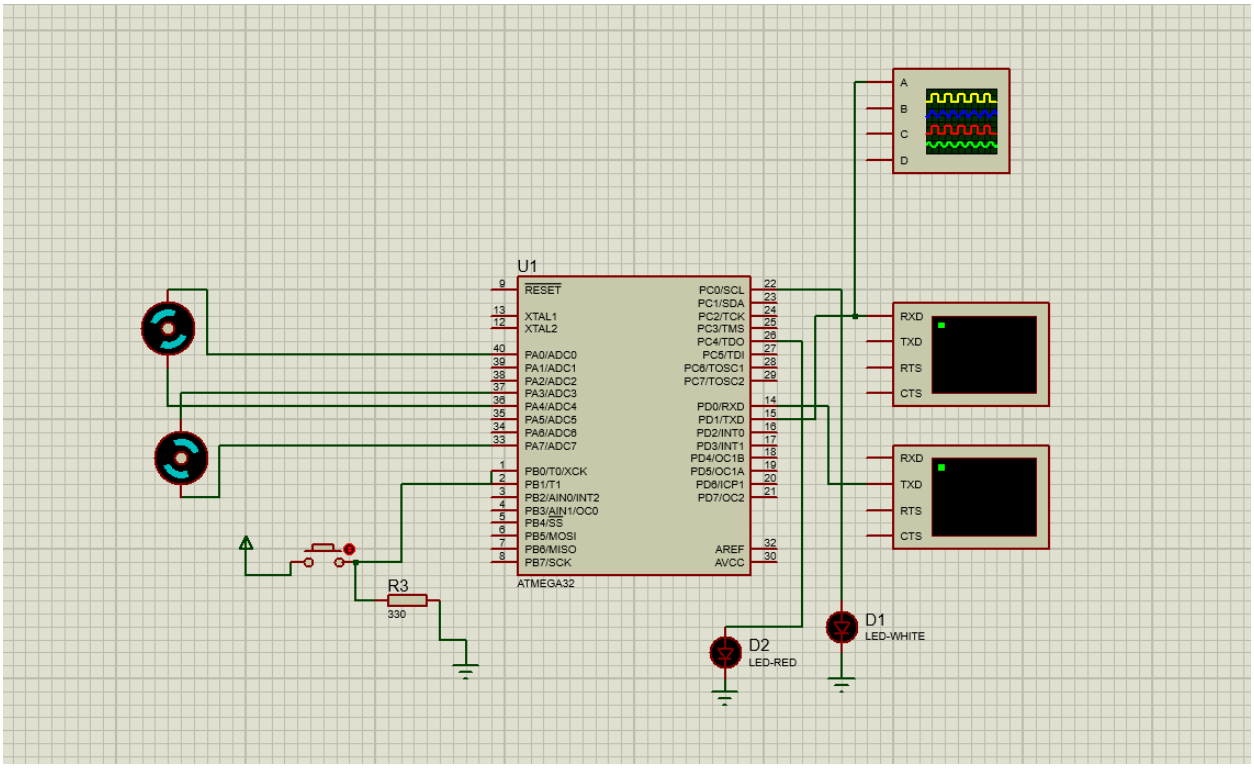
FlowChart:


Start
Read Data from the module
Is Received Data = 'A' or 'a'
Motor A goes clockwise and motor B goes unti-clockwise
Stop after 3 seconds
Is Received Data = 'B' or 'b'
Motor B goes clockwise and motor A goes unti-clockwise
Stop after 3 seconds

Block Diagram:



Simulation:



Code:

```
/*
 * Door.c
 *
 * Created: 5/2/2023 6:27:04 PM
 * Author : Mohamed ali
 */
#include <avr/io.h>
#define F_CPU 8000000UL // Define CPU clock frequency for delay function
#include <util/delay.h>
#define BAUD_RATE 9600 // Define the desired baud rate
#define BAUD_PRESCALER ((F_CPU / (BAUD_RATE * 16UL)) - 1) // Calculate baud rate
prescaler value

void USART_Init(unsigned int baud_prescaler) {

	UBRRH = (baud_prescaler >> 8); // Load upper 8 bits of the baud rate value into
the UBRRH register
	UBRRL = baud_prescaler; // Load lower 8 bits of the baud rate value into the UBRRL
register
	UCSRB = (1 << RXEN) | (1 << TXEN); // Enable receiver and transmitter
	UCSRC = (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1); // Set frame format: 8 data
bits, 1 stop bit, no parity
}

unsigned char USART_ReadChar() {
	while ((UCSRA & (1 << RXC))==0); // Wait until data is received
	return UDR; // Return received data
}

/*void USART_SendChar(unsigned char byte) {
	while ((UCSRA&(1<<UDRE))==0); // Wait until data is received
	UDR=byte; // Return received data
}*/


int main(void) {
	USART_Init(BAUD_PRESCALER); // Initialize USART with the calculated baud rate
prescaler value
	unsigned char received_char;
	DDRA=0xFF;
	DDRC=0xFF;
	DDRB  &= ~(1<<PORTB0);

	while (1) { // Infinite loop
		//USART_SendChar('A');
		//USART_SendChar('B');
		if (PINB & (1 << PORTB0)) {
			PORTC |= (1 << PORTC0);
			_delay_ms(5000);
			PORTC &= ~(1 << PORTC0);
		}
		received_char=USART_ReadChar();

		if((received_char=='A')||(received_char=='a')){
```

```
                        PORTA=0b10100101;// portA for motors
                        PORTC=0x0F;//portC for leds
                        _delay_ms(5000);
                        PORTA=0x00;
                        PORTC=0x00;

                }
                if ((received_char=='B')||(received_char=='b'))
                {
                        PORTA=0b01011010;// portA for motors
                        PORTC=0xF0;//portC for leds
                        _delay_ms(5000);
                        PORTA=0x00;
                        PORTC=0x00;
                }
        }

        return 0;
}
```