VOICE ACTIVATED ARDUINO BLINDS USING BLUETOOTH

Under the guidance of
(Dr. Sanju Saini Mam)

Submitted By:
Mansi soni (19001002038)
Prakash kumar (19001002050)
Atul bhoriya (19001002019)

Introduction on Automation

Automation is a term for technology applications where human input is minimized. The implementation of automation technologies, techniques and processes improve the efficiency, reliability, and/or speed of many tasks that were previously performed by humans.

Automation is being used in a number of areas such as manufacturing, transport, utilities, defense, facilities, operations and lately, information technology.

Advantages of automation:

- Greater productivity
- Better reliability
- Easier governance

Challenges:

- Cost
- Scope

OBJECTIVE OF PROJECT:

Creating a voice-activated system using dc motor leveraging an Arduino and bluetooth module that helps in automation of blinds.

In this project, we will be building a simple robotic subsystem that responds to the voice commands.

A microcontroller will collect inputs from a microphone to listen some wake words like "on" and "off," and then drive a small DC motor in the commanded direction.By this application we will be controlling the motion of blinds.

So the main feature that these blinds are going to have is the ability to open and close them from an app.

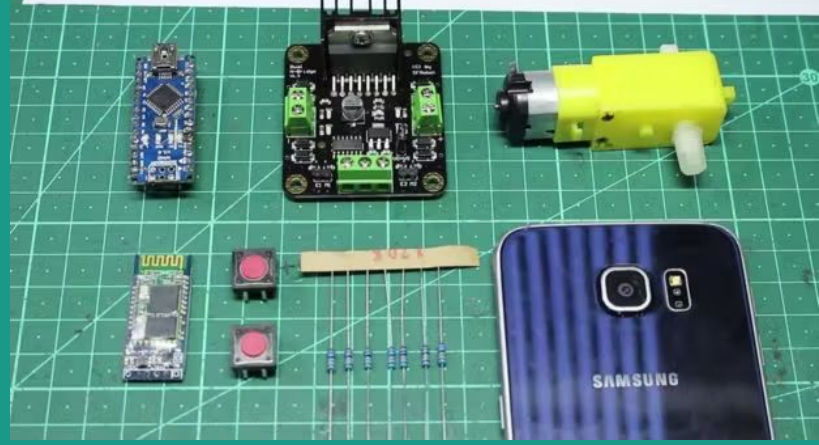
We have two blinds that we want to control with this device so we are going to be using two motors, one for each blind. When the command U is sent through the Bluetooth device it will spin both motors to the open position and if the command D is sent both motors will spin to the closed position.

Now one of the issues with Bluetooth based projects is that if the phone would die before the person got home, so they wouldn't be able to get use the device. Therefore in this project we are going to add push buttons to the device that let us open the blinds when the button is pressed.

This project features voice control and the ability to control individual motors allowing us to open one blind at a time.

REQUIREMENTS:

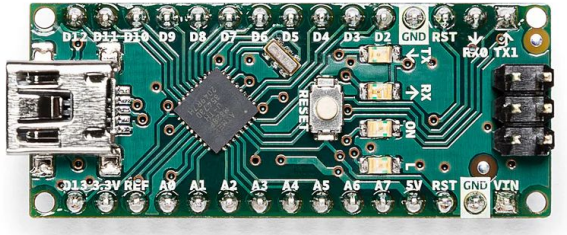
- ➔ Arduino Nano
- ➔ L293D Dual H-Bridge Motor Driver
- ➔ Hc - 06 Bluetooth module
- ➔ DC Motor
- ➔ Resistors
- ➔ Battery and Connector
- ➔ Push buttons
- ➔ LED



Tools and materials :

- Cardboard
- Wire
- soldering iron and solder
- hot glue gun and hot glue
- Box cutter

ARDUINO NANO



The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x).

Operating voltage: 5 volts

Input voltage: 5 to 20 volts

Digital I/O pins: 14 (6 optional PWM outputs)

Analog input pins: 8

DC per I/O pin: 40 mA

DC for 3.3 V pin: 50 mA

Flash memory: 32 KB, of which 2 KB is used by bootloader

Power consumption is 19 mA

Size of the printed circuit board is 18 X 45mm

Supports three communications like SPI, IIC, & USART

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX).

It has more or less functionality of the Arduino UNO but in a small form factor. The only major differences from UNO are the lack of a DC power jack, the usage of a Mini USB port instead of a USB B port, and the USB-TTL converter chip.

ARDUINO IDE :

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Arduino Nano Pinout

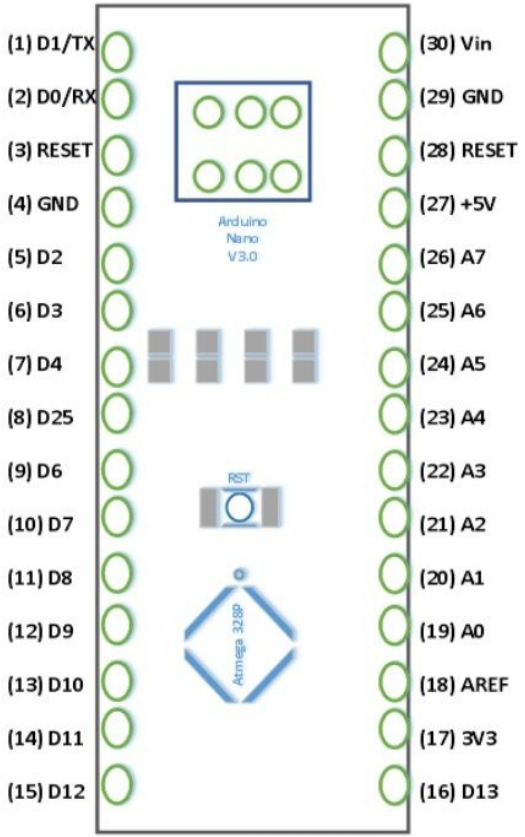
Power Pin (Vin, 3.3V, 5V, GND): These pins are power pins

Vin is the input voltage of the board, and it is used when an external power source is used from 7V to 12V.

5V is the regulated power supply voltage of the nano board and it is used to give the supply to the board as well as components.

3.3V is the minimum voltage which is generated from the voltage regulator on the board.

GND is the ground pin of the board



RST Pin(Reset): This pin is used to reset the microcontroller

Analog Pins (A0-A7): These pins are used to calculate the analog voltage of the board within the range of 0V to 5V

I/O Pins (Digital Pins from D0 – D13): These pins are used as an i/p otherwise o/p pins. 0V & 5V

Serial Pins (Tx, Rx): These pins are used to transmit & receive TTL serial data.

External Interrupts (2, 3): These pins are used to activate an interrupt.

PWM (3, 5, 6, 9, 11): These pins are used to provide 8-bit of PWM output.

SPI (10, 11, 12, & 13): These pins are used for supporting SPI communication.

Inbuilt LED (13): This pin is used to activate the LED.

IIC (A4, A5): These pins are used for supporting TWI communication.

AREF: This pin is used to give reference voltage to the input voltage

Arduino Nano Communication

The microcontroller using in Nano board (ATmega328) offers serial communication (UART TTL). This can be accessible at digital pins like TX, and RX. The Arduino software comprises of a serial monitor to allow easy textual information to transmit and receive from the board.

The TX & RX LEDs on the Nano board will blink whenever information is being sent out through the FTDI & USB link in the direction of the computer. The library-like SoftwareSerial allows serial communication on any of the digital pins on the board. The microcontroller also supports SPI & I2C (TWI) communication.

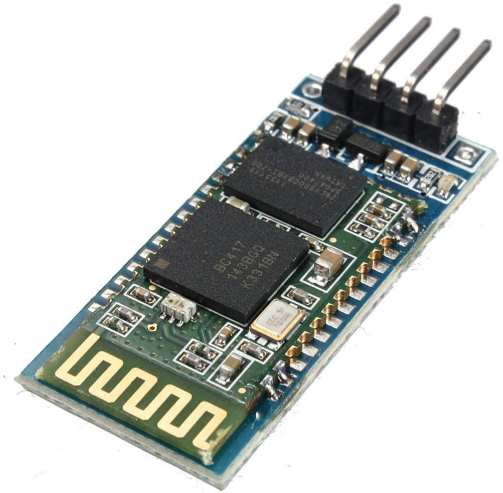
Applications of Arduino Nano:

Samples of electronic systems & products, Automation ,Several DIY projects, Control Systems, Embedded Systems, Robotics, Instrumentation

HC-06 bluetooth module

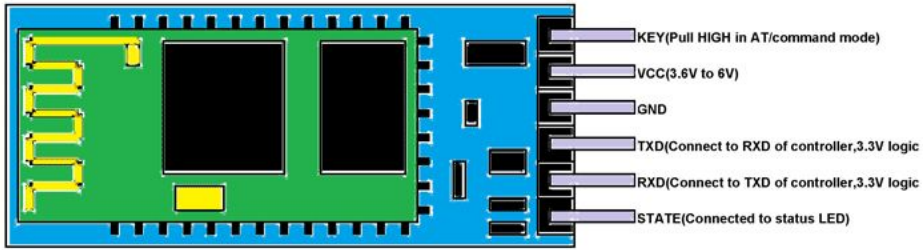
HC-06 is a Bluetooth module designed for establishing short range wireless data communication between two microcontrollers or systems.

The HC-06 is a class 2 slave Bluetooth module designed for transparent wireless serial communication. Once it is paired to a master Bluetooth device such as a PC, smartphone, and tablet, its operation becomes transparent to the user. All data received through the serial input is immediately transmitted over the air.



Pin configuration:

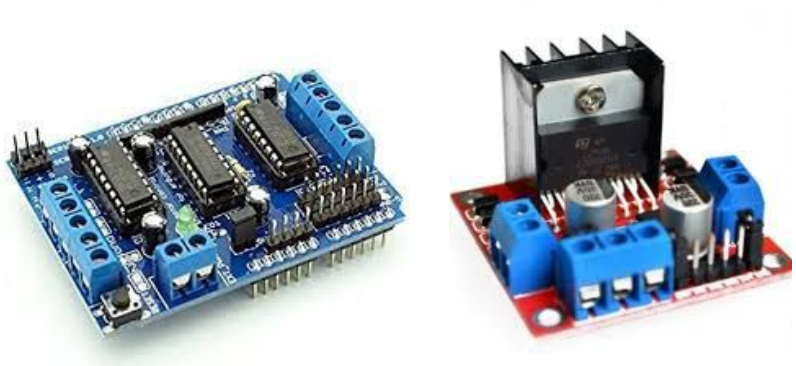
1. Key - The pin state determines whether the module works in AT command mode or normal mode [High=AT commands receiving mode(Commands response mode), Low or NC= Bluetooth module normally working]
2. Vcc - +5V Positive supply needs to be given to this pin for powering the module
3. Gnd- Connect to ground
4. TXD- Serial data is transmitted by module through this pin (at 9600bps by default), 3.3V logic



5. RXD- Serial data is received by module through this pin (at 9600bps by default),3.3V logic

6. State- The pin is connected to the LED on the board to represent the state of the module.

L293d motor driver



The L293D is a 16-pin Motor Driver IC which can control a set of two DC motors simultaneously in any direction. The L293D is designed to provide bidirectional drive currents of up to 600 mA (per channel) at voltages from 4.5 V to 36 V (at pin 8!).

L293D shield is a driver board based on L293 IC, which can drive 4 DC motors and 2 stepper or Servo motors at the same time. Each channel of this module has the maximum current of 1.2A and doesn't work if the voltage is more than 25v or less than 4.5v.

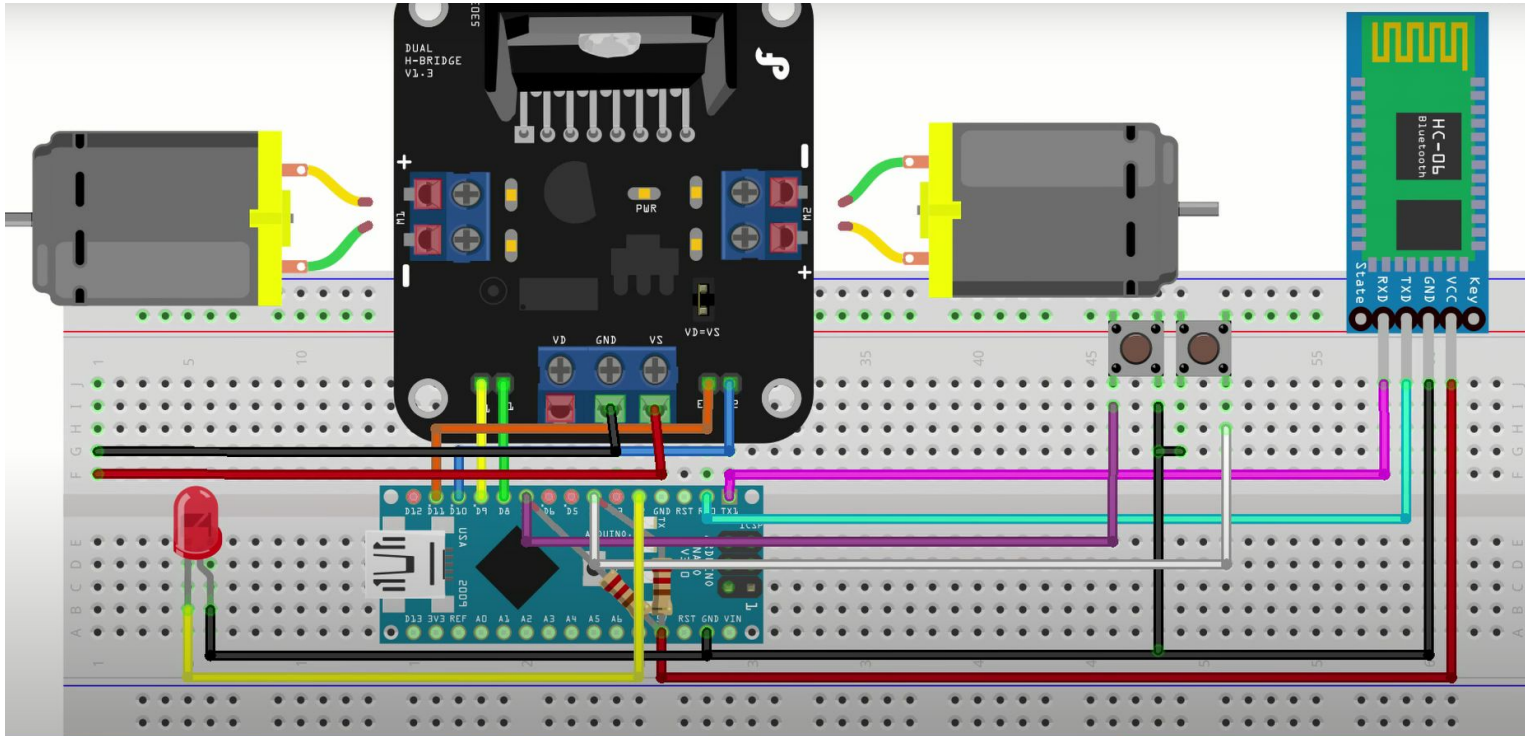
USE OF PUSH BUTTONS

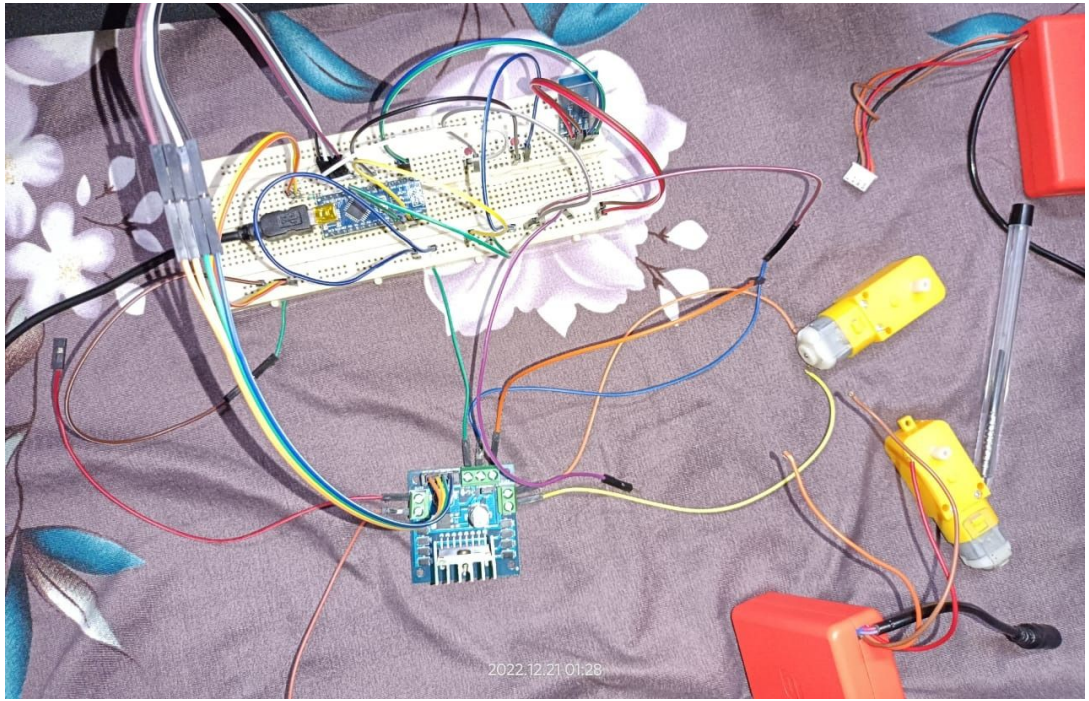
One of the issues that bluetooth based projects faces is that the phone battery dies so we will not be able to use the device after that.

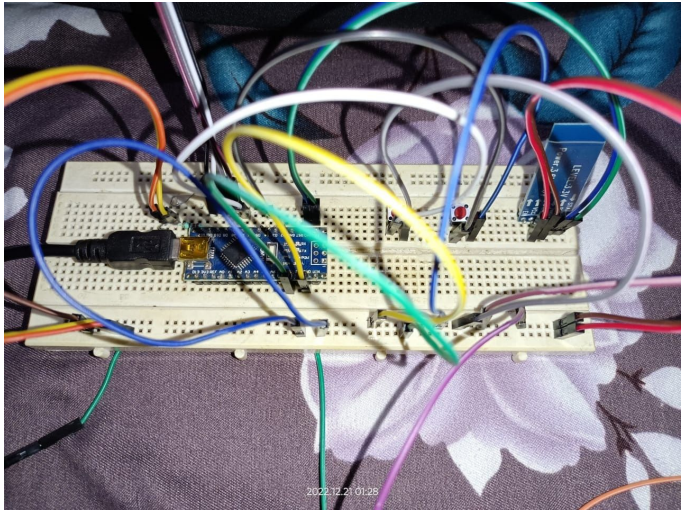
To solve this issue, push buttons are installed so that we can open the blinds by pressing these buttons.

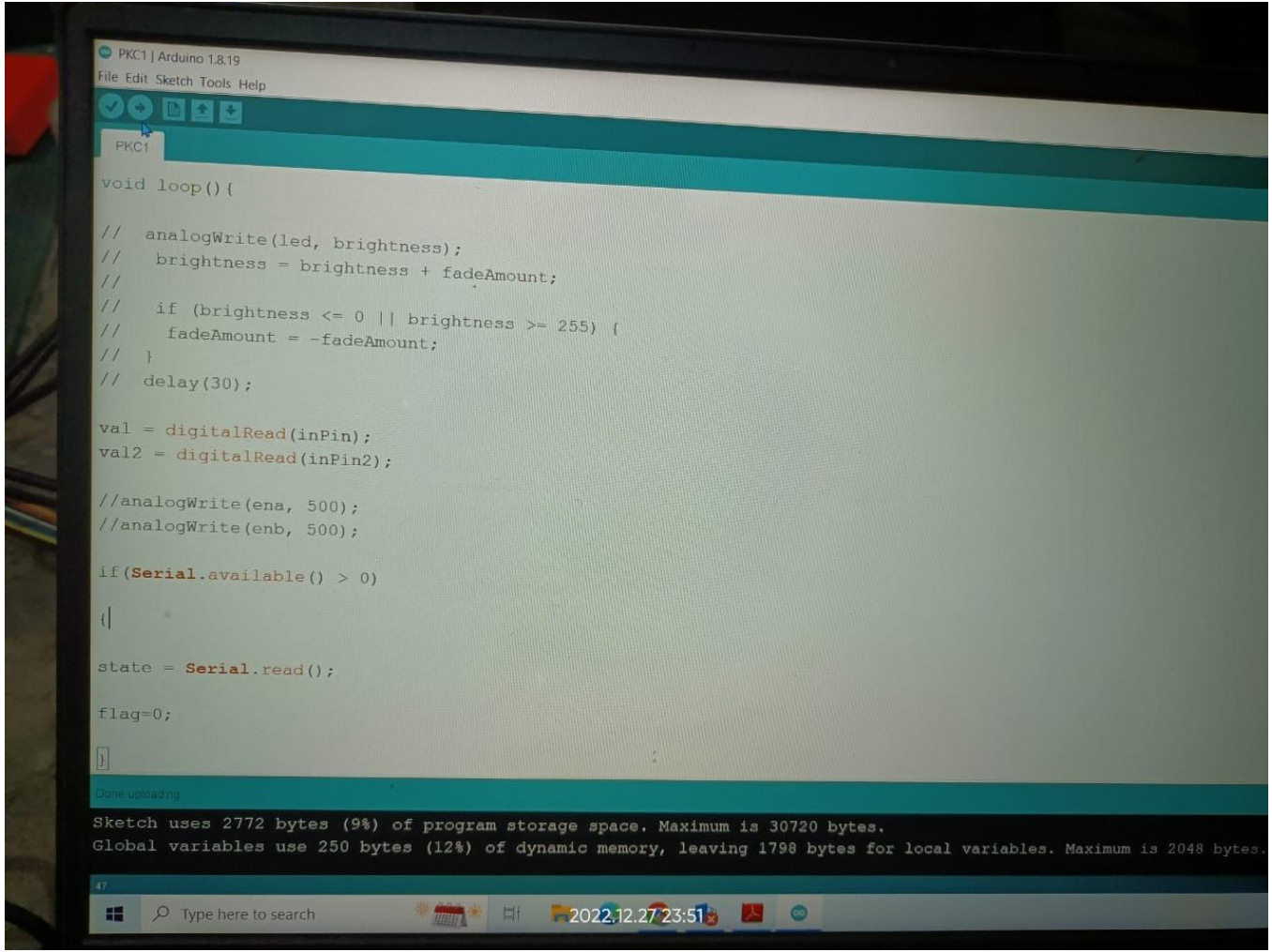
In order to control the motion of the blinds, two push buttons are used, one for the forward motion of motors and the other for the reverse motion.

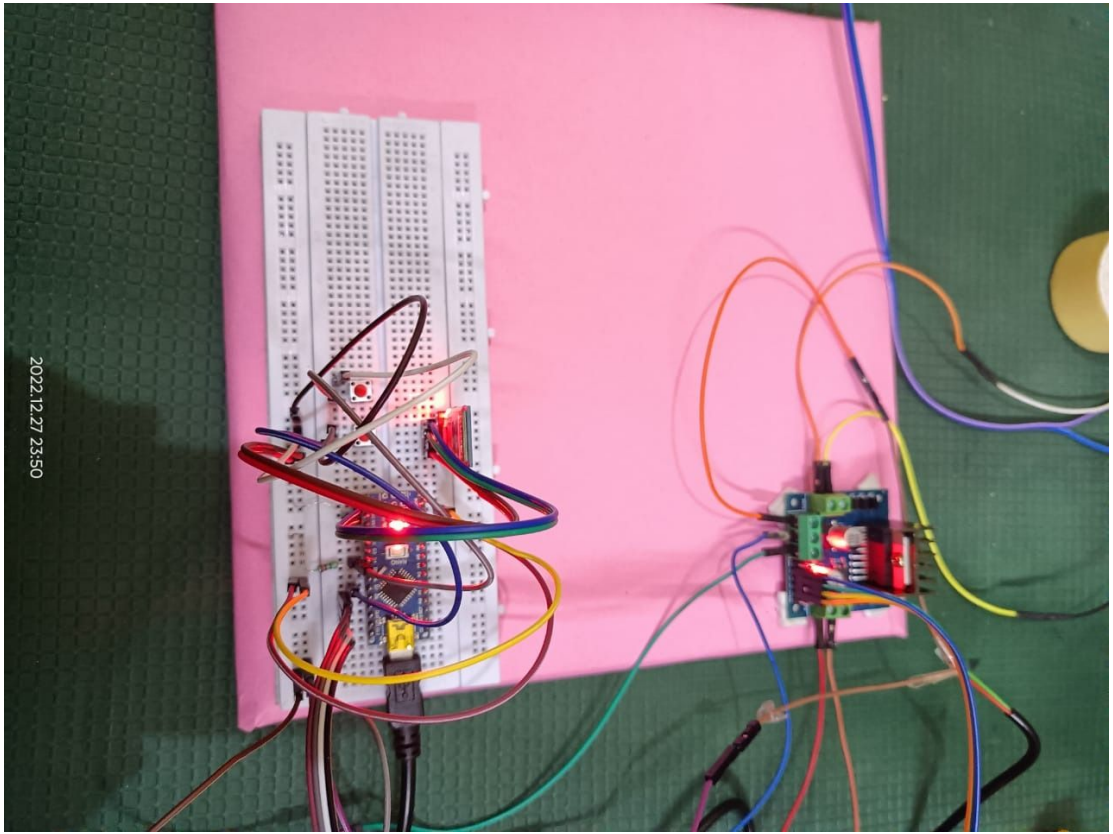
CIRCUIT DIAGRAM

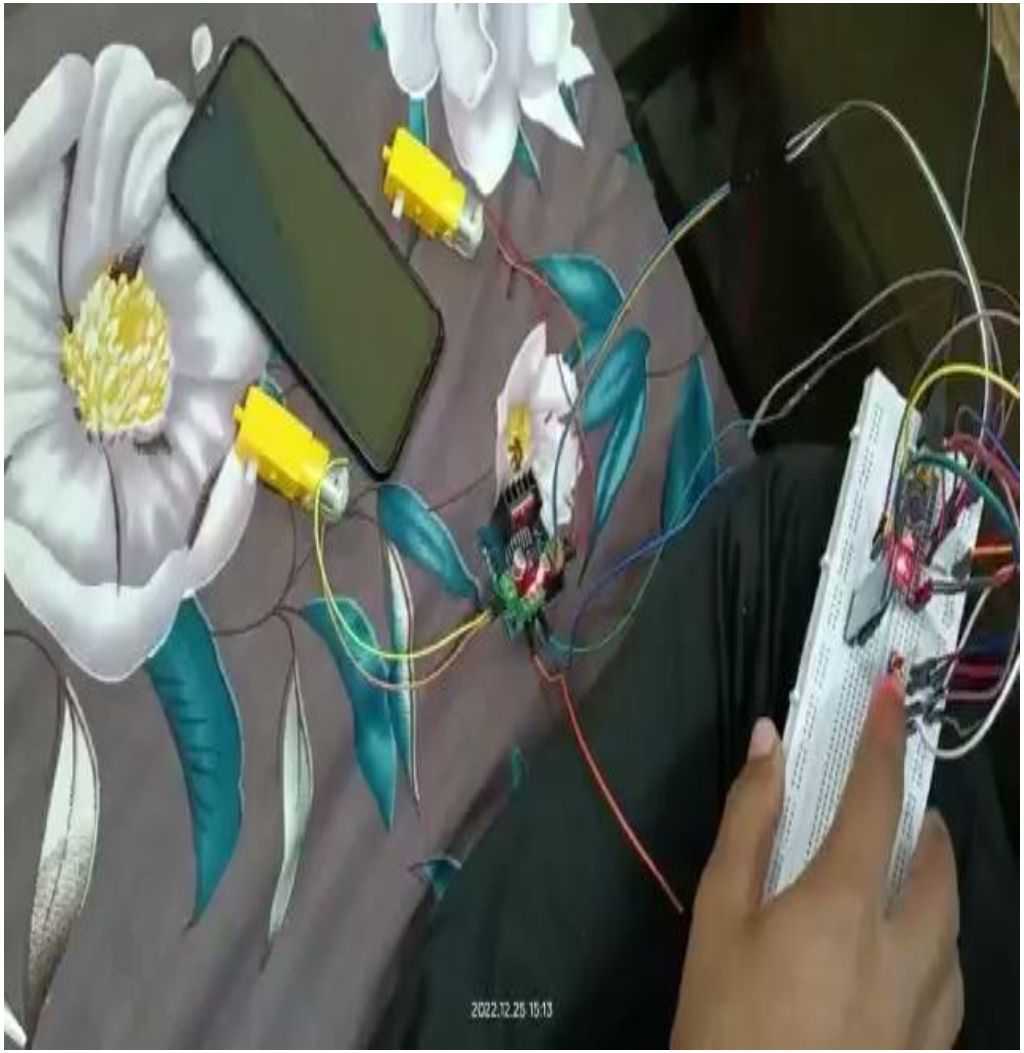



2022.12.21 01:28


2022.12.21 01:28


PKC1 | Arduino 1.8.19
File Edit Sketch Tools Help
PKC1
void loop(){
// analogWrite(led, brightness);
// brightness = brightness + fadeAmount;
//
// if (brightness <= 0 || brightness >= 255) {
// fadeAmount = -fadeAmount;
// }
// delay(30);
val = digitalRead(inPin);
val2 = digitalRead(inPin2);
//analogWrite(ena, 500);
//analogWrite(enb, 500);
if(Serial.available() > 0)
{
state = Serial.read();
flag=0;
}
Done uploading.
Sketch uses 2772 bytes (9%) of program storage space. Maximum is 30720 bytes.
Global variables use 250 bytes (12%) of dynamic memory, leaving 1798 bytes for local variables. Maximum is 2048 bytes.
Type here to search
2022.12.27 23:51


2022.12.27 23:50


2022.12.25 15:13

Steps to connect the circuit:

Step 1: let's start by connecting the motor controller to the Arduino:

Pin 8 connects to M1 on the motor controller
Pin 9 connects to E1 on the motor controller
Pin 10 connects to M2 on the motor controller
Pin 11 connects to E2 on the motor controller

Step 2: Now connect the Bluetooth module to the Arduino:

5 Volt Pin connects to VCC on the Bluetooth module
Ground Pin connects to Ground on the Bluetooth module

Step 3: Upload the code and then connect

Rx connects to Tx on the Bluetooth module
Tx connects to Rx on the Bluetooth module

Step 4: Connect the push buttons

Connect a resistor from Pin 7 on the Arduino to 5 volts on the Arduino
Connect a resistor from Pin 4 on the Arduino to 5 Volts on the Arduino
Connect one leg of the button to Pin 7 and the other leg to ground
Connect one leg of the next button to Pin 4 and the other leg to ground

Step 5: Now we are going to connect an LED to pin 4 which will show that the device has power:

Pin 4 goes to the cathode (long leg of LED), Ground goes to anode (Short leg of LED)

Step 6: And lastly, we are going to connect the motors to the motor controller by screwing them into the screw terminal. To gives these motors enough power to turn the somewhat stiff blind mechanism we are going to need at least a 9 Volt 1 Amp wall plug power supply. Once you've found a power supply we are going to want to connect it to our Arduino and the Motor Controller.

It connects to the Arduino as follows:

Positive (+) connect to the VIN Pin on the Arduino , Ground (-) connects to the Ground Pin on the Arduino

It connects to the Motor controller as follows:

Positive (+) connects to VS on the motor controller , Ground (-) connects to GND on the motor controller

Now we can plug the motor controller in and give it a test, if everything lights up we can move onto the next step!

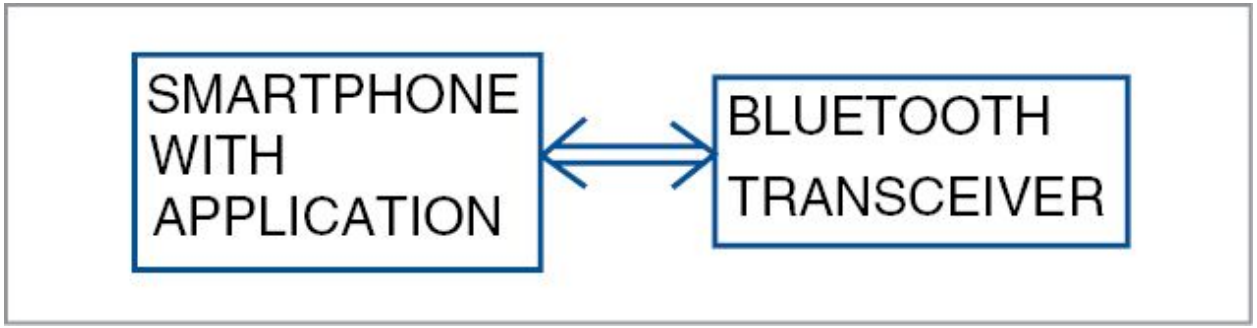
To connect the motors to our blinds we are going to need to take the stick and hook that is normally used to open the blinds and break the hook off, we will then glue that to the shaft of the motor and hook it onto the blinds movement mechanism.

Now we can't just leave the motor dangling there so we are going to have to glue it to the wall.

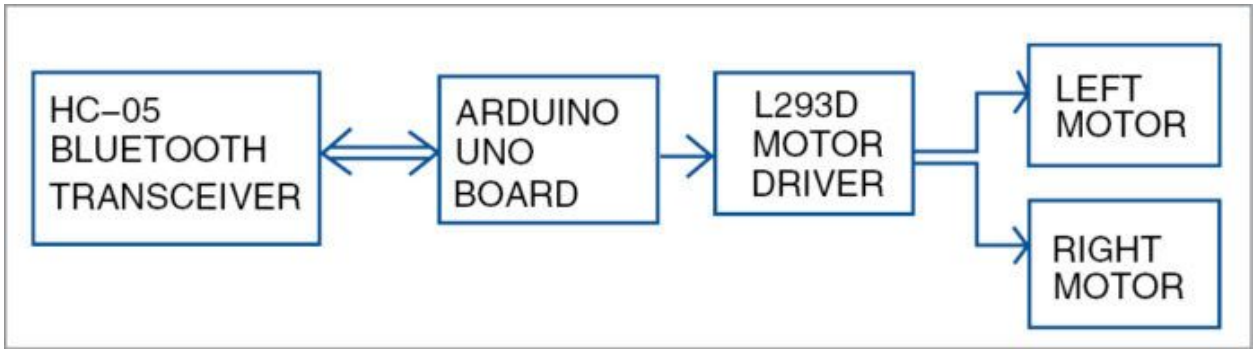
WORKING OF PROJECT

- Voice Commands are processed by phone, and speech-to-text conversion is done within the app using Google's speech-recognition technology.
- Text is then sent to the receiver side via Bluetooth.
- Text received via Bluetooth is forwarded to the Arduino board using UART serial communication protocol.
- Arduino code checks the text received.
- Whenever the text is a matching string, Arduino controls the movements of the blinds.

Block diagram of transmitter side

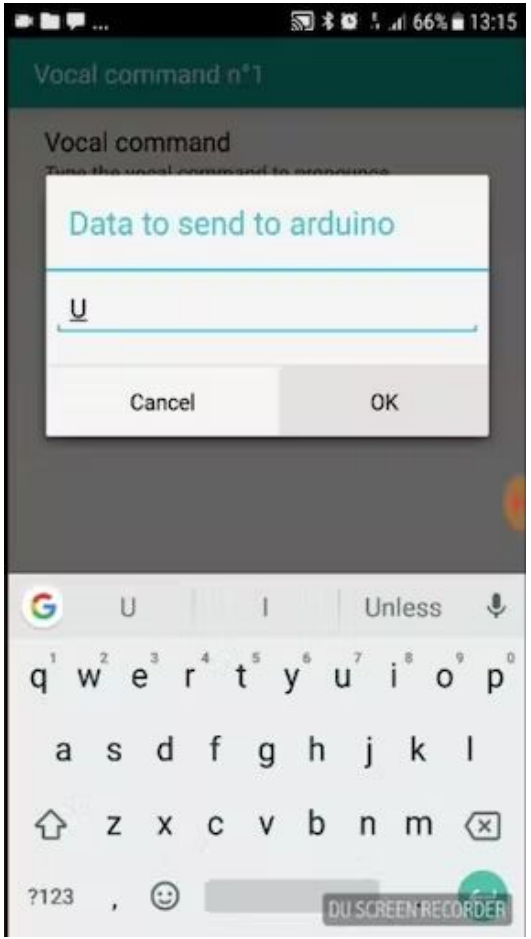


Block diagram of the receiver side



Commands given to the app :

- If we want both blinds open, we can either push the top button or we can say **let there be light** in the app, If we want both blinds closed we can push the bottom button or say **let there not be light** in the app.
- If we only want one of the two blinds open at a time we can go into the terminal section of the app and send the command **“K”** to open the left blind
- And **“I”** to close the left blind or **“W”** to open the right blind and **“L”** to close the right blind.



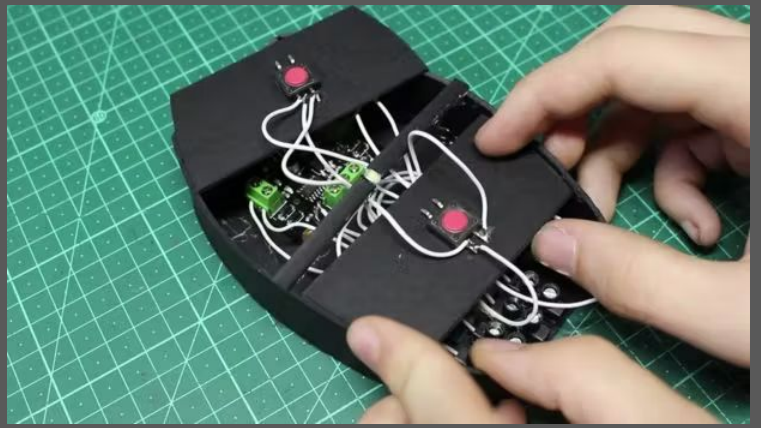
The app we are going to be using is called "Arduino Bluetooth Control" by Broxcode, the reason we chose this app is because it has not only a terminal feature but also a feature allowing us to send commands to the Arduino when a chosen phrase is said.

To set it up we are going to pair our Bluetooth device with our smartphone and send U and then D through which should make the motors spin clockwise and then counter-clockwise.

CONCLUSION

This project features voice control and the ability to control individual motors allowing us to open one blind at a time.


Voice Activated
Arduino Blinds



References

- Xiaoling Lv, Minglu Zhang and Hui Li, "Robot control based on voice command," 2008 IEEE International Conference on Automation and Logistics, 2008, pp. 2490-2494, doi: 10.1109/ICAL.2008.4636587.
- https://research.google.com/pubs/archive/46554.pdf
- https://how2electronics.com/wireless-voice-controlled-robot-car-using-arduino/
- L293D motor driver with arduino: https://youtu.be/zhWGeJnIfeY
- Motor connections : https://yainnoware.blogspot.com/p/motor-conncetions.html
- https://www.instructables.com/Simplest-Home-Automation-Using-Bluetooth-Android-S/
- M. F. Riza and N. S. Salahuddin, "Control Home Devices with Voice Commands via a Smartphone," 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019, pp. 1-7, doi: 10.1109/ICIC47613.2019.8985796.