VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

Voice Controlled LED’S Using Arduino

Team Members : S. Dattathreya - 21911A3555
T. Diya - 21911A3556
T. Rishitha -21911A3557
T. Venkat - 21911A3558

Faculty Sign

Section : AI-A

Voice Controlled LED’S
Using Arduino

Introduction:

In this Arduino Project I will show you how to use a mobile app for Android and the HC-05 Bluetooth module to create voice controlled LED's. Voice-controlled systems epitomize these attributes, allowing users to interact with devices through simple spoken commands. One popular application of this technology is the integration of voice control with LED lights using Arduino.

The Concept of Voice Control

Voice control refers to the ability to control devices and systems through spoken commands. This involves converting spoken words into digital signals that a computer or microcontroller can interpret and act upon. Advances in natural language processing and speech recognition have made voice control more accurate and user-friendly.

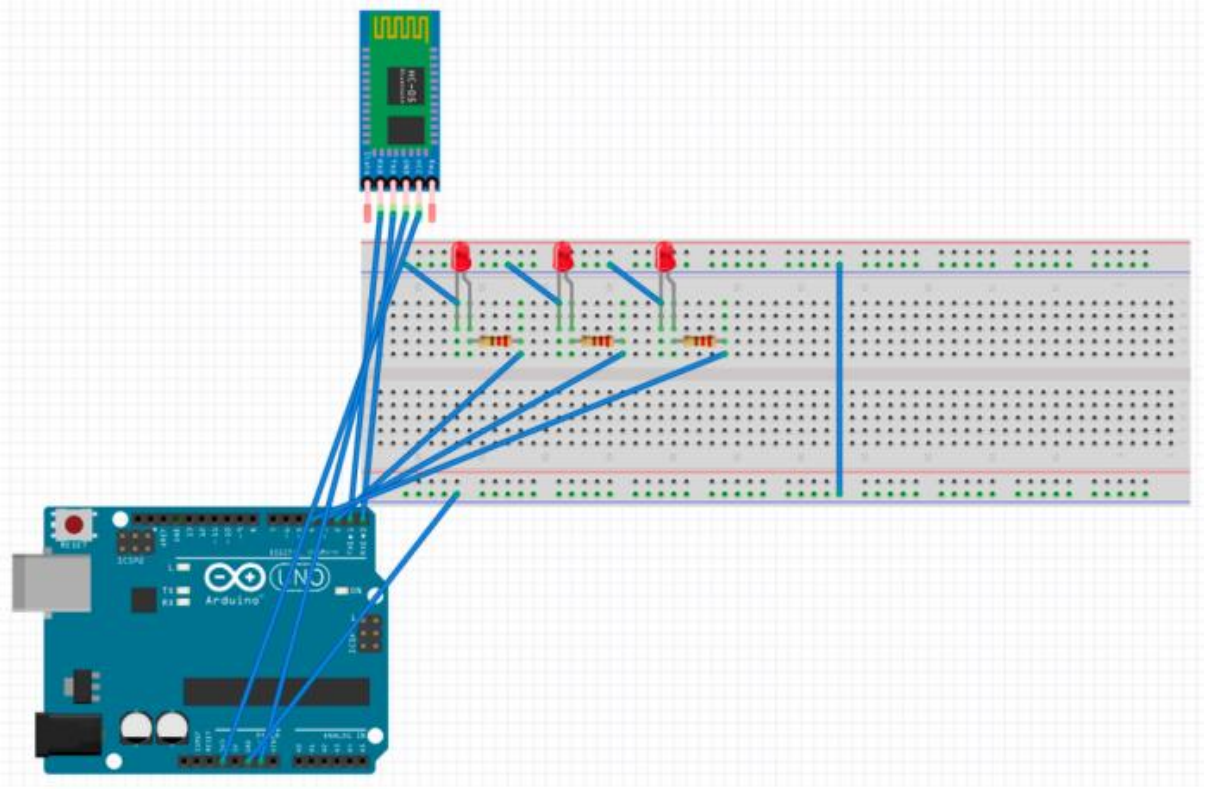
Voice Controlled LEDs

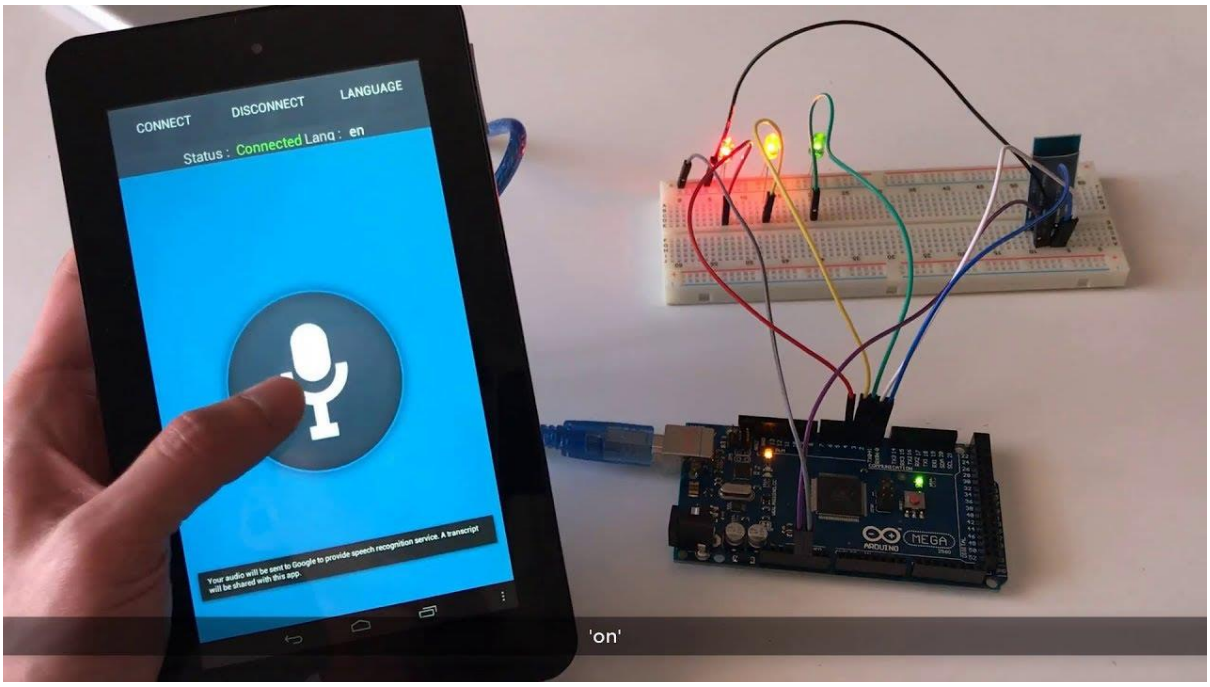
In a voice-controlled LED system using Arduino, users can turn on, off, or change the colour and brightness of LEDs using voice commands. This can be achieved through the integration of a microphone and a speech recognition module with the Arduino board. The system processes the voice commands and translates them into actions performed by the LEDs.

Components:

1. Arduino UNO
2. HC-05 Bluetooth Module
3. Breadboard
4. Jumper Cables
5. 330 Ohm Resistors
6. LED's

Circuit Diagram and Prototype:





Your audio will be sent to Google to provide speech recognition service. A transcript will be shared with this app.

Procedure:

Step 1: Connecting the HC-05 Bluetooth Module Step 1: Connecting the HC-05 Bluetooth Module.

The standard HC-05 Bluetooth module has six pins. However, we will only be using 4 in this project. The pins we will be using are the VCC pin, the GND pin, the TXD pin, and the RXD pin. The VCC pin of the Bluetooth module connects to the +3.3V of the Arduino. The GND pin of the module connects to GND of the Arduino. TX pin of the Bluetooth module connects to digital pin 0 (RXD), and the RXD pin connects to digital pin 1 (TXD).

Step 2: Connecting the LED's.

The next step in creating our voice controlled LED's is to connect the LED's to the Arduino through the breadboard. First, connect the short end of the LED to ground. Next, connect the long end of each of the LED's to a 330 ohm resistor. Finally, connect the resisted current from the LED's to digital pins on the Arduino. In this project, we will be connecting one LED to digital pin 2, another to digital pin 3, and the last LED to digital pin 4.

Step 3: Supplying Power.

For this project, we can supply power to the Arduino through any +5V power source. You can use a USB port from your computer to power the Arduino, but in this project, I will be using a 5V portable battery. Before you connect a power source to your Arduino, make sure that the GND of the Arduino is connected to ground of the breadboard.

Step 4: Get the Code.

Get the source code and upload the source code.

Step 5: Using the Mobile App Step.

You can begin to test your project by downloading the app for Android: BT Voice Control for Arduino. Next, click on the menu bar in the upper right hand corner and select 'Connect Robot.' On the new window that appears, select the HC-05 bluetooth module and connect it. Congrats! You have completed your voice controlled LED Arduino project. Now, when you speak certain commands from the code into your device through the app, specific LED's should turn on and off.

Source Code:

```
#include 'SoftwareSerial.h' //Replace (' ') with (< >)
SoftwareSerial BLU(0,1);
String voice;
int Green = 4; //Connect To Pin #4
int Yellow = 2; //Connect To Pin #2
int Red = 3; //Connect To Pin #3
void allon()
{
        digitalWrite(Red, HIGH);
        digitalWrite(Yellow, HIGH);
        digitalWrite(Green, HIGH);
}
void alloff()
{
        digitalWrite(Red, LOW);
        digitalWrite(Yellow, LOW);
        digitalWrite(Green, LOW);
 }
void setup()
{
        Serial.begin(9600);
        BLU.begin(9600);
        pinMode(Red, OUTPUT);
        pinMode(Yellow, OUTPUT);
        pinMode(Green, OUTPUT);
}
void loop()
{
        while (Serial.available()) //Check if there is an available byte to read
        {
                delay(10); //Delay added to make thing stable
                char c = Serial.read(); //Conduct a serial read
                if (c == '#')
                {
                        break; //Exit the loop when the # is detected after the word
                }
                voice += c;
        }
if (voice.length() > 0)
        {
                if(voice == "*turn on all LED")
                {
                        allon();
                }
                else if(voice == "*turn off all LED")
                {
                        alloff();
                }
                else if(voice == "*switch on red")
                {
                        digitalWrite(Red,HIGH);
                }
                else if(voice == "*switch on yellow")
                {
                        digitalWrite(Yellow,HIGH);
                }
                else if(voice == "*switch on green")
                {
                        digitalWrite(Green,HIGH);
                }
                else if(voice == "*switch off red")
                {
                        digitalWrite(Red,LOW);
                }
                else if(voice == "*switch off yellow")
                {
                        digitalWrite(Yellow,LOW);
                }
                else if(voice == "*switch off green")
                {
                        digitalWrite(Green,LOW);
                }
                voice=""; //Reset variable
        }
}
```

Conclusion/Result:

The voice-controlled LED project using Arduino successfully integrates Bluetooth communication and digital control to create an intuitive and interactive lighting system. The project enables the user to control the state of multiple LEDs(red, yellow, green)using predefined voice commands sent via a Bluetooth module .By employing the SoftwareSerial library, the project establishes a reliable serial communication link between a mobile device and the Arduino, enabling users to send voice commands to control the state of multiple LEDs. The project effectively processes incoming voice commands to execute precise actions, such as turning on or off.