

Bluetooth module LED voice control

I. Summary

In this project, we will use an HC-05 Bluetooth module and a smartphone to send voice commands to control LEDs to receive voice commands. We are using an “Arduino Bluetooth Voice Controller” android app which you can download from play store. The speech is received by a microphone and processed by the voice module using voice command. The voice command is given by using mobile to the Bluetooth which has certain features like controlling which LEDs open and close.

II. Objectives

- To learn how to use the HC-05 module for controlling Arduino via Bluetooth communication.
- Learn how to control LEDs using Bluetooth modules and Arduino.

III. Industry-Based Applications

Automation plays a key role in the world of technology. Home automation allows us to control household electrical appliances such as light, door, fan, AC, etc. It also provides home security and an emergency system. Home automation does not only reduce human efforts, energy efficiency and time-saving. The main objective of home automation and security is to help handicapped and old aged people who will enable them to control home appliances and alert them in critical situations.

IV. Project Methodology

When we control the voice, speech recognition is a technology where the system understands the words given through speech. The main aim of this project is to control LEDs through the human voice. In this system, we have used a voice recognition module to recognize the voice of the user for controlling the LEDs turning on and off. The advancement of use in this project is to control home appliances by using voice command to this project can also work as a home automation system. The command was set in the Arduino board. The home appliances can be controlled by two methods, by giving voice commands or by using mobile as a remote controller.

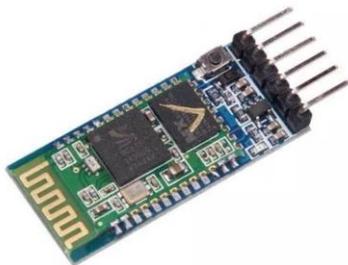
Arduino is a simple integrated development environment that runs on any PC and allows users to write programs for Arduino in C or C++ language. The entire programs are installed in an Arduino controller for any commands.

Components:

- 1x Arduino Uno



- 1x HC-05 Bluetooth Module



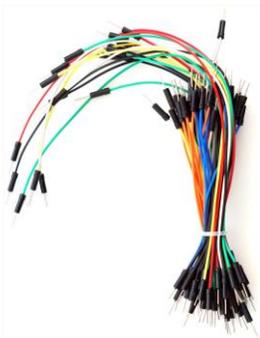
- 3x 330 Ohm Resistors



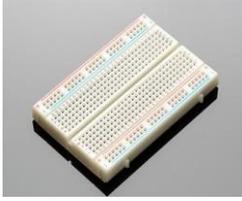
- 3x LED's



- Jumper Wires

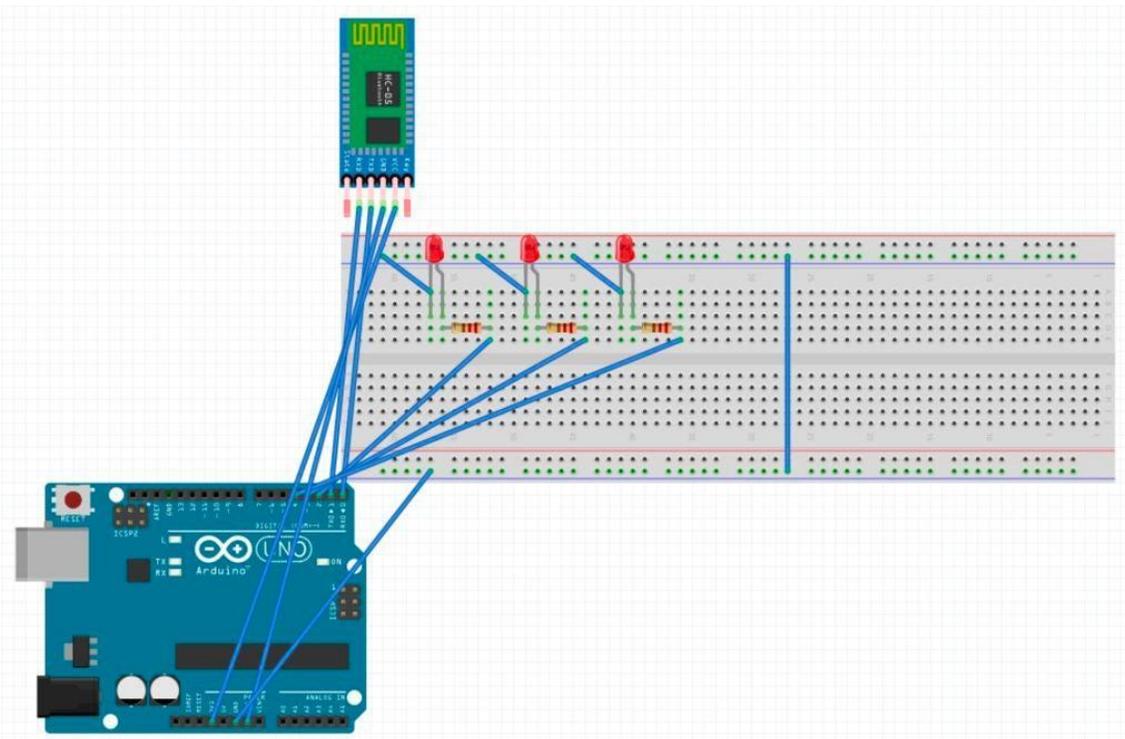


- 1x Breadboard



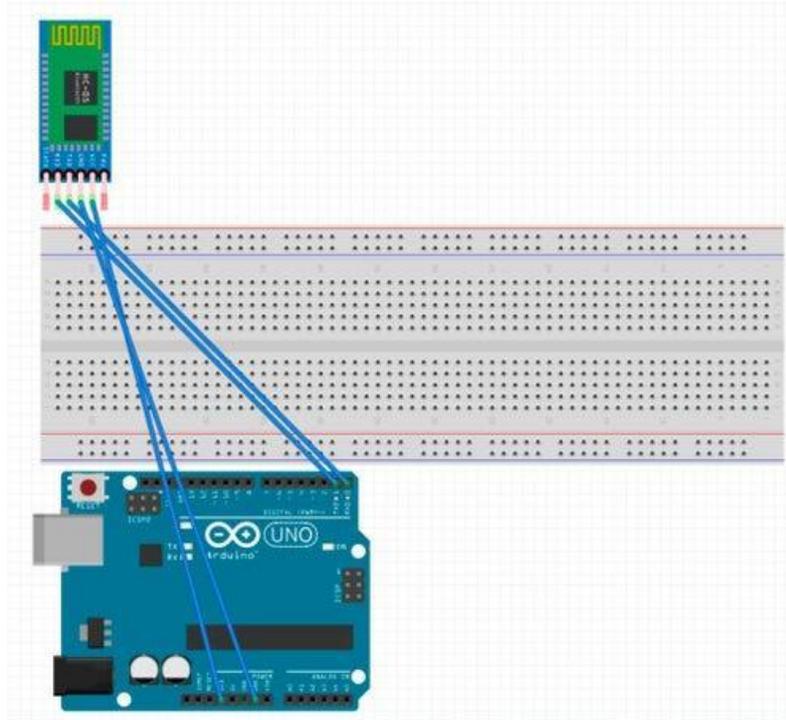
Project

Procedure

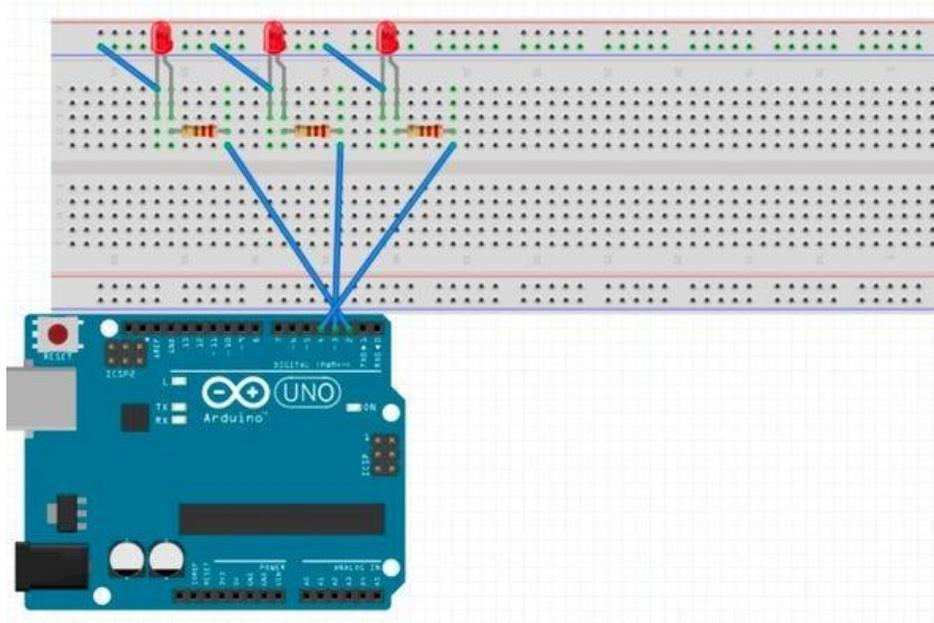


The standard HC-05 Bluetooth module has six pins. However, we will only be using 4 of the pins in this project. The pins we will be using are the VCC pin, the GND pin, the TXD pin, and the RXD pin.

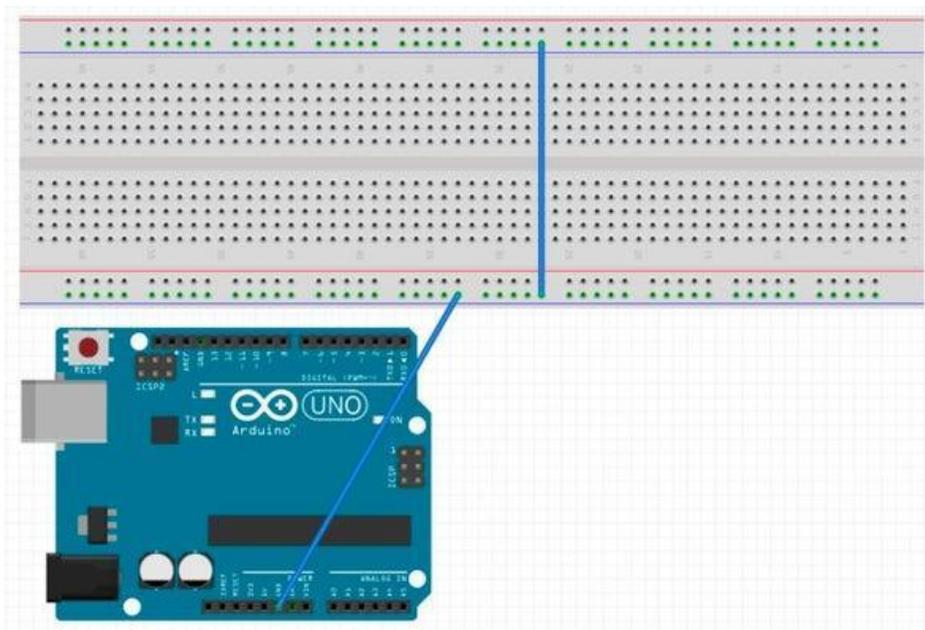
The VCC pin of the Bluetooth module connects to the +3.3V of the Arduino. The GND pin of the module connects to the GND of the Arduino. TX pin of the Bluetooth module connects to digital pin 0 (RXD), and the RXD pin connects to digital pin 1 (TXD)



The next step in creating our voice-controlled LEDs. Connect the LED to the Arduino through the breadboard. First, connect the short end of the LED to the 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.



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. Before you connect a power source to your Arduino, make sure that the GND of the Arduino is connected to the ground of the breadboard.

https://www.youtube.com/watch?v=EZBVh1Ztk4A&feature=emb_title

Code:

```
#include <SoftwareSerial.h>

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  
  
}  
  
}
```

To test the project download the app for Android: BT Voice Control for Arduino; Created by SimpleLabsIN. 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 to it. When you speak a certain command from the code into your device through the app, specific LED's should turn on and off.

I. Reference

1. Instructables. (2017, October 16). Voice Activated Arduino (Bluetooth Android). Retrieved May 1st, 2020, from <https://www.instructables.com/id/How-to-Make-Voice-Controlled-LEDs-Using-Arduino/>