

Wrench Arduino LED Mask

Introduction

The Wrench Light-Up Mask Project is a creative undertaking that aims to bring the popular video game character, Wrench, to life. The main inspiration for the project comes from Wrench's iconic mask that lights up and changes colors in the game Watch Dogs 2. The project involves building a real-life version of this mask, complete with working LED lights and a design that closely resembles the original.

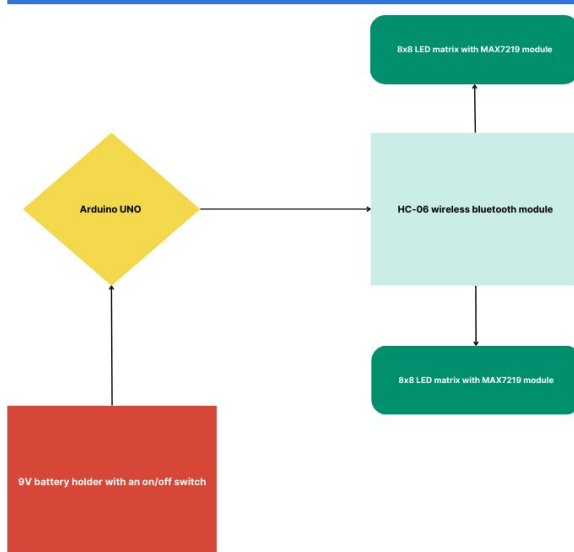
General Description

The Arduino UNO is the central component, controlling the wireless communication with the HC-06 module and the LED matrix display. The 9V battery holder provides power to the Arduino UNO and the LED matrix display. The on/off switch on the battery holder allows the user to turn the system on and off as needed.

The white LED matrix display with MAX7219 module is used to display visual information, and can be controlled by the Arduino UNO through the use of the LedControl library.

Overall, this system is designed to allow wireless communication between the Arduino UNO and a remote device using Bluetooth, while also providing visual feedback through the use of the LED matrix display.

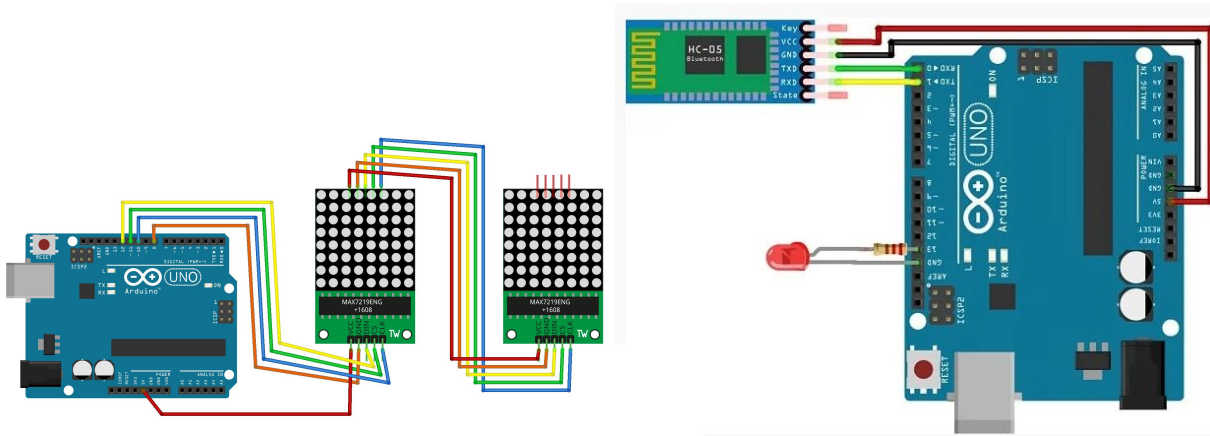
Wrench Mask Diagram



Hardware Design

These are the first components I believe I will require, though they may alter as project experiments progress:

- 3D printer
- HC-06 wireless bluetooth module
- Arduino UNO
- 9V battery holder with an on/off switch
- x2 white 8x8 LED matrix with MAX7219 module



Software Design

The pre-programmed LED matrix eyes and alphabet are kept in the memory of the Arduino as binary arrays. Each array has a key value associated with it that corresponds to the expression to which it is linked. Although this can be altered at the top of the code, the Arduino by default shows the X eyes when it is powered on because this has been set as "key1".

While certain facial expressions, like a wink, include numerous frames, others have just one with a blinking animation in between. While the Arduino waits for a bluetooth signal to update the expression, it will continue to loop the existing expression in the meantime.

Each byte in the 8-bit array used to encode the photos corresponds to one row of the image. Each byte's bit representation corresponds to the state of each pixel in the associated row of the image.

Some of them are:

- heart[]: This code represents an image of a heart.
- heart2[]: This code represents an image of a slightly more filled-in heart.
- heart3[]: This code represents an image of a fully filled-in heart.
- equal[]: This code represents the equal sign "=".
- tilde[]: This code represents a tilde "~".
- minus[]: This code represents a minus sign "-".
- dollar[]: This code represents a dollar sign "\$".
- exclamation[]: This code represents an exclamation mark "!".

- `O[]`: This code represents the letter "O".
- `smallo[]`: This code represents the letter "o".
- `at[]`: This code represents the "at" symbol "@".
- `semicolon[]`: This code represents a semicolon ";".
- `semicolonblink[]`: This code represents a blinking semicolon ";".
- `nineEye[]`: This code represents a nine-eyed character.
- `negation[]`: This code represents a negation sign "¬".
- `tripleDot[]`: This code represents three dots "...".
- `tripleDotBlink[]`: This code represents blinking three dots "...".
- `star[]`: This code represents a star "*".
- `bracket[]`: This code represents a bracket.
- `bracketBlink[]`: This code represents a blinking bracket.
- `invader1a[]` and `invader1b[]`: These codes represent two frames of an animation of an invader character.
- `invader2a[]` and `invader2b[]`: These codes represent two different frames of an animation of another invader character.
- `t[]`: This code represents the letter "T".
- `lenny[]`: This code represents the "Lenny" face meme.
- `musicNote[]`: This code represents a music note.
- `leftQuote[]` and `rightQuote[]`: These codes represent the left and right quotation marks, respectively.

Through the bluetooth receiver, the Arduino can be connected to an Android device, such as a phone or tablet. This authorizes you to instruct the Arduino to modify the values displayed. The Bluetooth module will show up with its default name, such as "HC-06," if the name has not been altered.

Once I install the app, I open it and I see a list of all paired devices. I press on the one corresponding to the name of the Bluetooth module and wait for the devices to connect. Now, I should be able to change the LEDs to different expressions by clicking buttons on the screen.

The Wrench Mask WD2, built using Arduino, utilizes the **delay** and **timers** functions to control the timing and duration of LED illumination, creating captivating visual effects. Additionally, it incorporates **serial communication** to establish a connection with a compatible device, allowing customization of the LED patterns in real-time.

Results

The result of my efforts is a fully functional, Bluetooth-controlled Wrench mask that offers an immersive experience. I am pleased to report that the mask includes 48 different set expressions that can be controlled with blinking and transition animations for a realistic look. The best part? It closely mirrors the in-game character, which was the primary goal of this project. The journey was long, but seeing the mask come to life made it worth the effort



Conclusions

I embarked on this journey to recreate the iconic light-up mask of Wrench, a character from the game Watch Dogs 2. There were already some fantastic tutorials out there, but I believed there was room for improvement. Inspired by these existing projects, I decided to build and code my own version. The process was challenging and took a month to complete, but I was satisfied with the outcome. The final product has several unique features, like being wirelessly controlled via a custom-built Dedsec themed Android application, having 48 preset expressions with blinking and transition animations, and the ability to draw custom expressions, display messages, and adjust speed and brightness. The mask is portable, powered by a single, easily replaceable 9V battery, and closely resembles the in-game character more than most other tutorials I have come across.

Download

🤖 [vlad_badii_arduino_wrench_mask.txt](#)

app for controlling on bluetooth: <https://github.com/bdykdz/wrench>

Bibliography/Resources

- Tutorial for working with these LED Matrices : <https://www.youtube.com/watch?v=H6Vs98-Cev0>
- Tutorial for creating an app to connect the Arduino using Bluetooth:
<https://www.youtube.com/watch?v=aE8EbDmrUfQ>
- Wrench Random Moments throughout the game so i can see more emojis to code in the matrix:
<https://www.youtube.com/watch?v=raY9ui5PEzU>
- Arduino_datasheet:
<https://docs.arduino.cc/static/32b4941b81a2c6a5308e0e9bd348d0e6/A000066-datasheet.pdf>

[Export to PDF](#)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/badii>



Last update: **2023/05/25 18:33**