Bee Button

These are the bee buttons, these buttons can be controlled with the use of MQTT. They send u if they have been pressed, double pressed or long pressed. And in turn, U can tell them to do light effects or play music. You controll the buttons via serial commandos, which every programming language supports. All files related to the project can be found on project files. A more detailed guide can be found at guide. If u have any questions please mail to m.c.pijnappel@student.utwente.nl.

Mesh Dongle

The commands that you want to send to the mesh go via the Mesh dongle. You can send the commands over serial to this dongle with a baudrate of 115200. The screen on the fongle tells some basic information about the mesh it is connected to. It also tells how many buttons are connected to it.

Normal Button

- Blinking Red -> Not connected to the mesh.
- Blinking Green -> Connected to Mesh.

SD card

On the SD card, there are a few things.

- Folder "Sounds": here u can store ur **MP3** sound files that u want to play.
- config.txt: Here are some configs u can change related to networking and some default parameters, without having to upload new code.
- README.md: that is what u are reading at the moment.
- Firmware.bin: it is not on there at the moment, but u can update the firmware by putting a firmware.bin file on the SD card.

config.txt

The config file is meant to quickly change some things without needing to edit the code.

- **MESH_SSID**: This is the name of the mesh, which has to match among the buttons u use, otherwise, they cannot find each other. U can also see this with ur phone/computer, but u cannot connect to it without a special app.
- **MESH_PASSWORD**: This is the password your network will use between the buttons, so not ur laptop. This also has to match between the buttons.
- **MESH_PORT**: This is the port where they communicate and also has to match. U can have multiple networks match on the MESH_SSID and MESH_PASWORD, but change the port. They won't be able to communicate with each other that way.
- **Volume**: This is the volume level the button start with. The scale is 1-21, can be handy to change this on startup. It can also be changed later with MQTT commands, but it won't be remembered between startups.
- **BRIGHTNESS**: The brightness value the LEDs start with, the value is in a range between 0 and 255. Just as with the volume, u can change this with a command but it will not be remembered between reboots.

Commands

These are all the commands that are available for the buttons with the use of serial commands. You start with either a nodeid which you can request with a command, or with 'Broadcast'. This can be used to send a commando to all nodes or to a specific node. After this you send ':' and the commando you want to send. Between each command layer you also put a ':'. At the end of command you have to put a '`' to indicate the command is done.

A command can be 'Broadcast:Full:Purple`', this will turn all connected buttons purple. To test command you can use the serial monitor from for example the arduino IDE.

nodelist

To get a list off all nodes connected to the dongle you can send the command 'nodelist`'. This will return a string that start with "nodeList' followed by all the nodes seperated by a '/'. For example 'nodeList/498425705/3171683153' The first id is from the dongle itself and can be ignored. This this list contains only the button with id 3171683153.

Colors

These colors are available to use.

- Red
- Blue
- Green
- Purple
- Yellow
- Orange
- Cyan
- Pink

These are the commands;

- `:Play:[song]` - This is how u play a song on a button. [song] has to be replaced with the name of a song u have put in the sounds folder of the SD card without the ".mp3".
- `:Rainbow` - This will put a rainbow effect on the button.
- `:Full:[color]` - This will put a solid color on the button. Replace [color] with one of the colors.
- `:Circle:[color]` - This will put a circle effect on the button in a color.
- `:Blink:[color]` - This will have the button blink a color.
- `:Clear` - Turns of the LEDs.
- `:Brightness` - With this, u can change the brightness of the button on a scale from 0 - 255.

- `:Volume` - With this u can change the volume of the speaker on a scale from 0-21.
- `:Battery:[subcommand]` - With this, u can get information about the battery.
    - `Percentage` - Gives the battery percentage.
    - `Voltage` - Gives the battery voltage.
    - `Show` - Gives back the battery voltage and percentage, but also show the battery capacity with the LEDs.

Extra pins available in the buttons

When u screw open the button there are some pins extra available to use. U do have to edit the code to use them. The code is written with platformIO and the code is on the GitHub. Installing platformio is fairly easy with Visual Studio Code.

- IC3/IC4: These are pins for the I2C protocol, lots of digital sensors can be connected to these pins like accelerometer/gyroscope.
- Analoge Pins: If u want to do something analoge it can be done with these pins. Make sure it is all 3.3V, since that is what the controller has, has maximum input and supplies.
- LED Strip PINS: If u want to use more LEDs u can connect a neopixel/ws2812b strip on these pins. At the moment this might not work with every buttons since some LEDs are broken and might not pass on the signal to the end. The LED strip would be LED number 13. But in the case of some buttons this might be off since some jumper wires are soldered over LEDs.