Project documentation

Project title: Modern music station

Project members: Roman Fruzykov

Project mentors: Piotr Jóźwiak

Content:

The aim:

The aim is to follow all phases of development and create the music station using the knowledge and techniques learned during the labs.

Project description:

The music station itself will look like a simple box with cardholders and few leds and control buttons on the side panel. It is used for turning on the music by placing different cards on the top of the box. On the card user can place the photo of Artist or the Album. Also user can use side buttons for controlling the flow of the music. The led is displaying the status of the music box. So the basic workflow of music station is follows:

Use flow

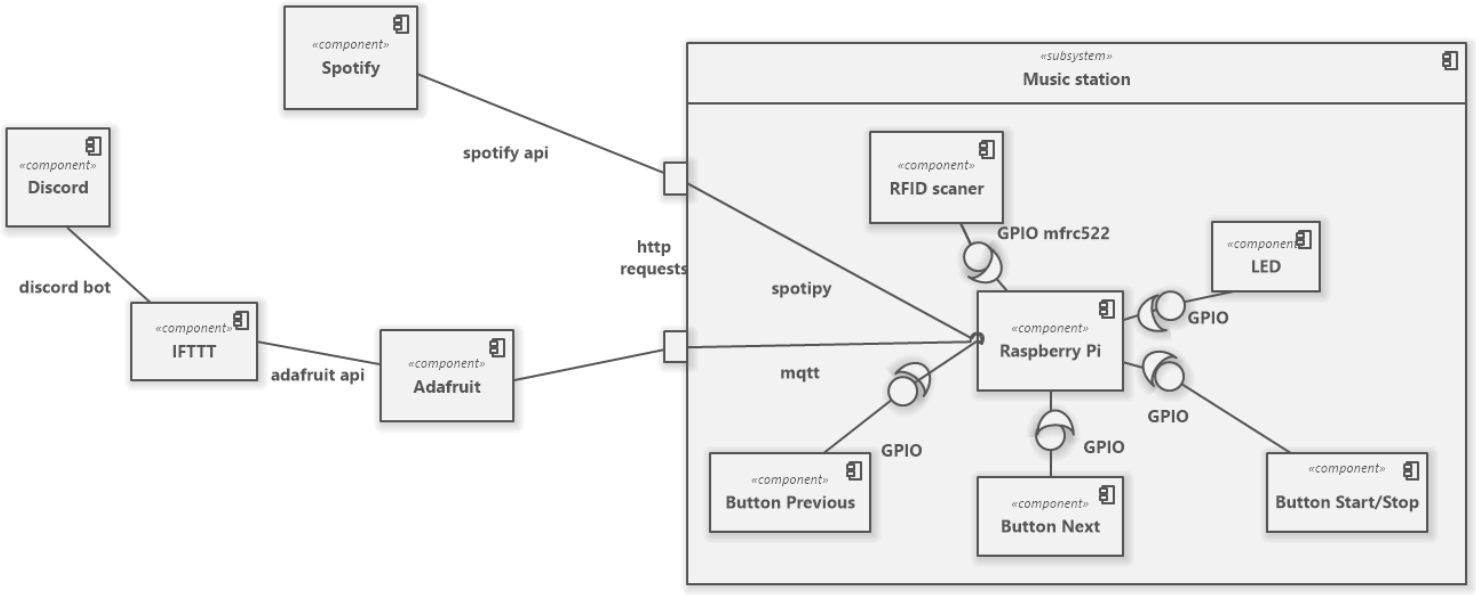
Basic use:

1. User turns on the music station.
2. User picks the music card form cardholder
3. User places the card on the top of the box
4. User places it back to the cardholder.
5. User uses the buttons to navigate between tracks.
6. User turns off the box

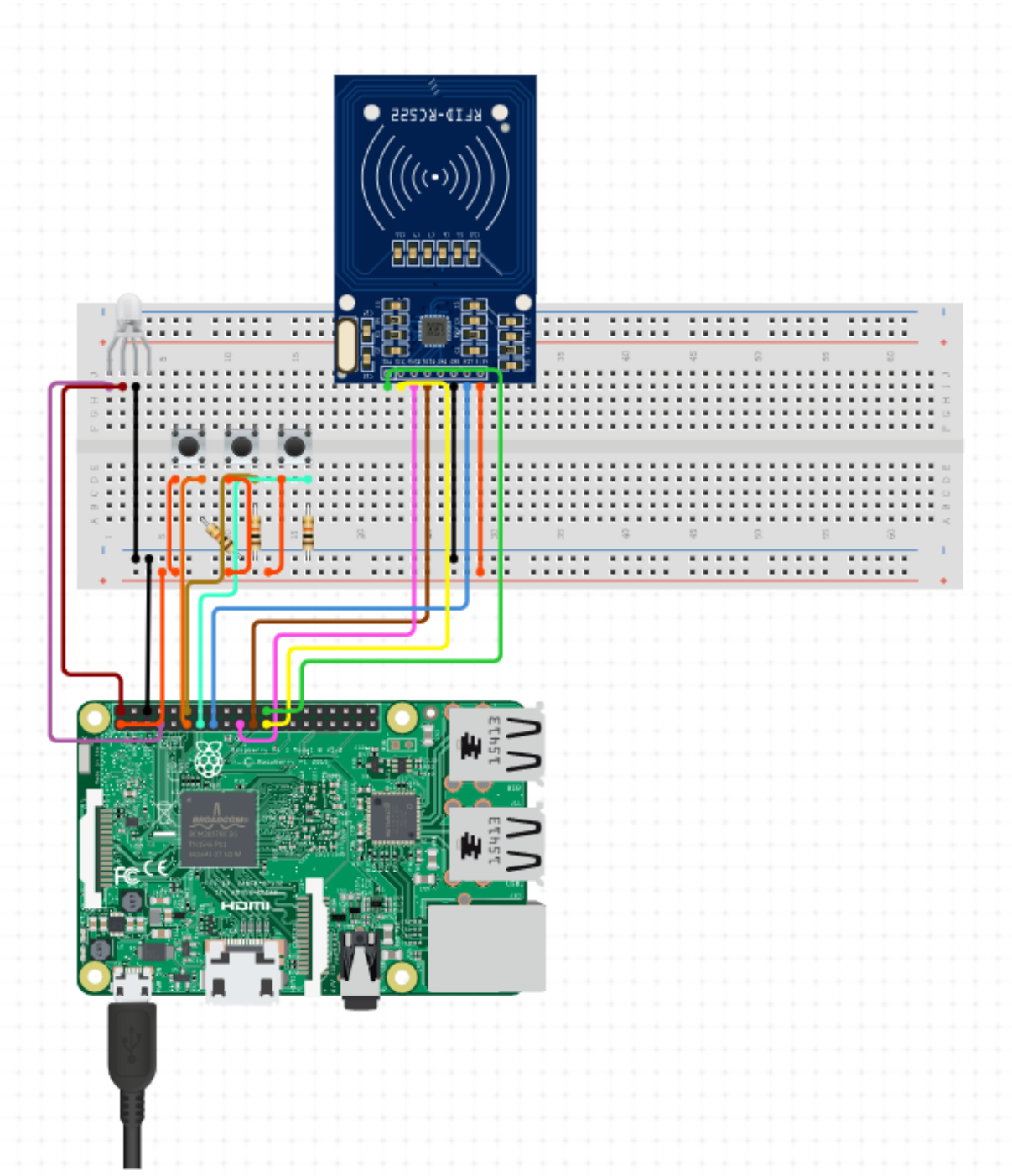
Card registration/ reassigning:

1. User turns on the music station.
2. User provides the link to an album/playlist/track via Discord
3. Inserts the card to card reader
4. User turns off the box

Project logical scheme:



Project physical scheme:



Parts needed:

Raspberry Pi 3 - Model B - ARMv8 with 1G RAM

LED - RGB Addressable, PTH, 5mm Diffused (5 Pack)

USB micro-B Cable - 6 Foot

3x Mini Pushbutton Switch

3x 10K Ohm Resistor

RFID - RC522 RF IC Card Sensor Module

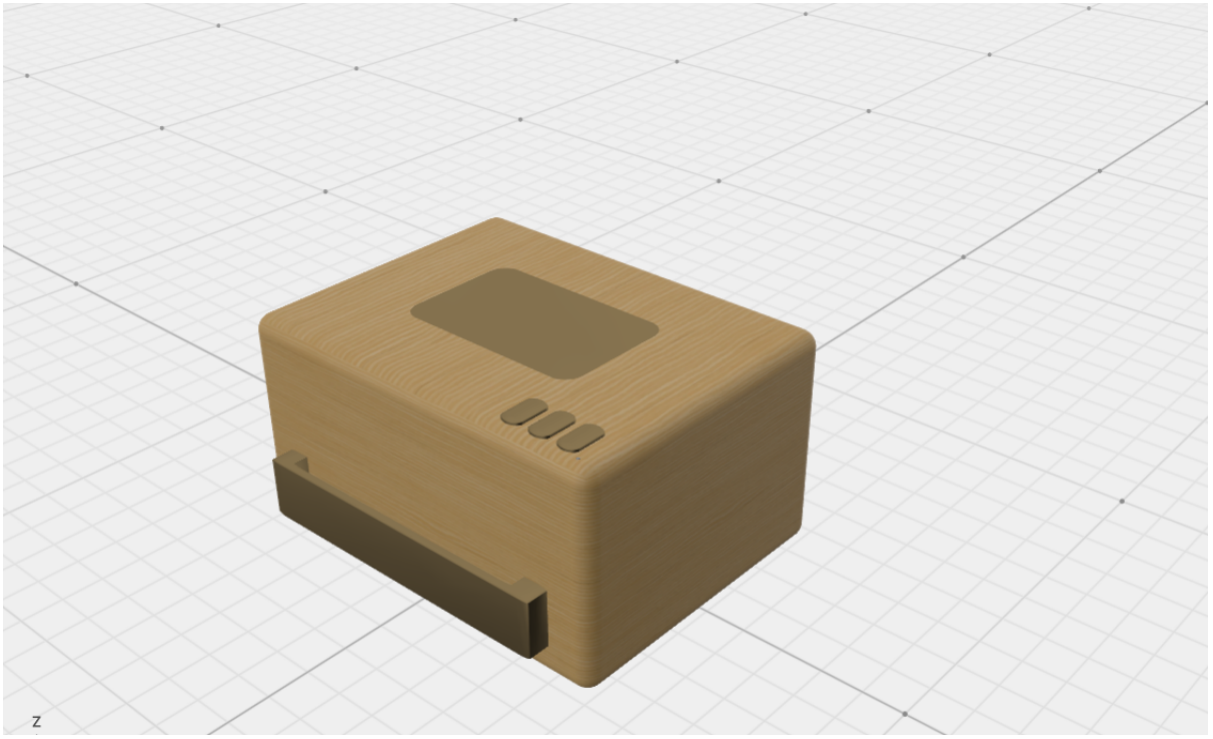
BreadBoard // NOT REALLY

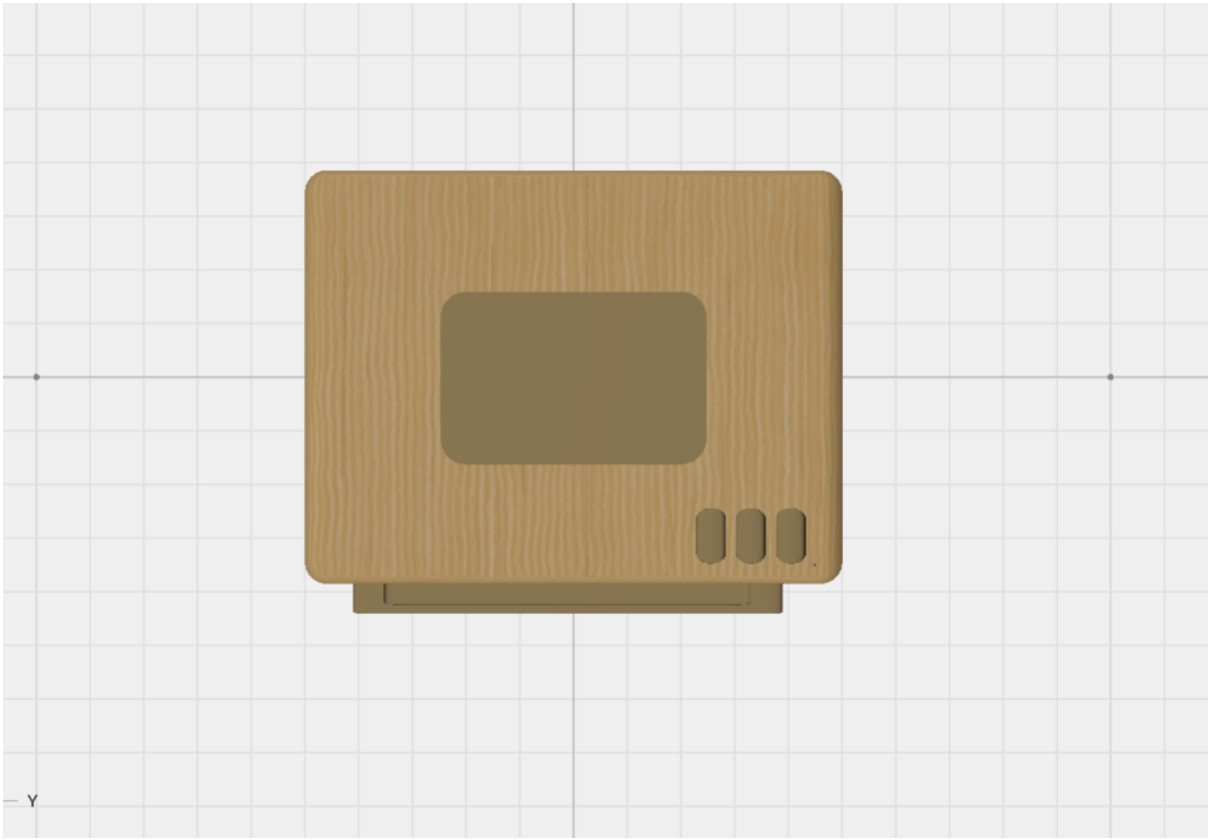
Jumper Wires Pack - M/M

Jumper Wires Pack - M/F

Box for the setup

Box renders:

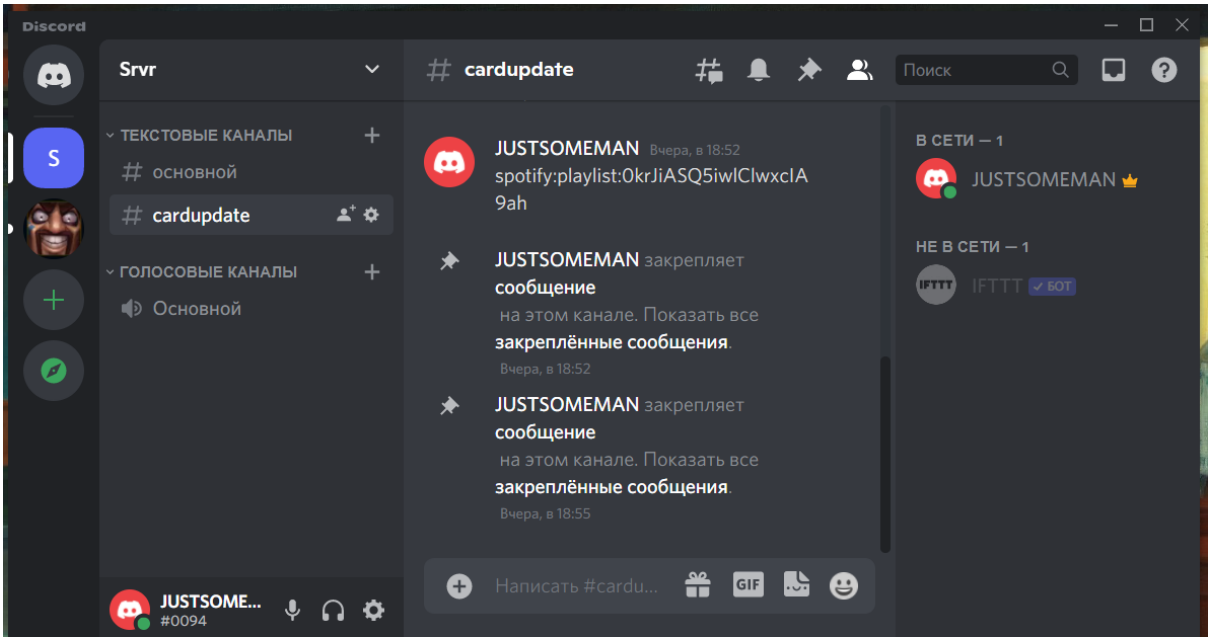




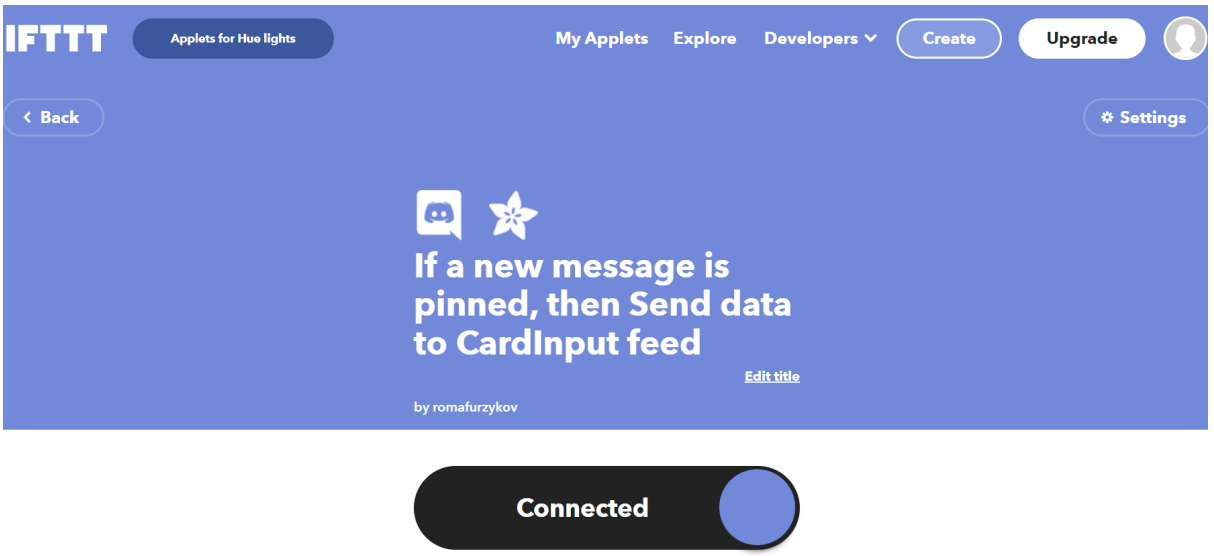
Detailed description of Discord - Raspberry connection

1. In discord I have added the new Server (Srvr) and new Text channel (cardupdate)

   Here was added a bot from IFTTT for listening to the new pinned messages in this channel.



2. The message itself is sent to the CardInput feed on Adafruit.io



3. On the python side I have a subscriber that listens to the CardInput feed changes

   And it sets up a special mode for the next card to input.

```
mqtt.Client.connected_flag=False
client = mqtt.Client("Subscriber1")                # creating client object
client.on_connect = on_connect                     # defining function o handler on connected
client.on_message = music.on_message

client.username_pw_set(ADAFRUIT_USER, password=ADAFRUIT_KEY)
client.connect(BROKER_HOST, port=PORT, keepalive=60)
client.loop_start()
```

Python libs used:

- ast
- json
- requests
- time
- paho.mqtt.client
- datetime

For adaptation to real project also will be needed:

- RPi.GPIO
- mfrc522

Led codes:

Green - All set and good.
Green/Yellow - Waiting for the card to insert the new data to it.
Yellow - Spotify services connection error.