

# Project Description

This project is a combination of a music playing website I call YourMusic and a remote controlling device that allows the user to control some functionalities of the website remotely. These functionalities include changing background color among ten different variations to customize for the user's preference, controlling the volume of the music playing along with pause and play options.

The system is generally composed of the following components:

- Arduino MKR 1010
- Potentiometers and buttons
- Arduino code (ino file) controlling the arduino MKR 1010
- Server.js: the javascript code that acts as a server between the client and the arduino
- Client.js: the javascript code that governs the index.html
- Index.html: the webpage
- Style.css: the css file that gives the style of the website.

The website is a basic music playing website. It contains a table like arrangement of songs with the album arts corresponding to each song. Each song on the website has a song control beneath it that the user can use to control the music playing. The songs are accessed locally, but this project can be extended for websites that use music APIs such as Spotify or Soundcloud.

The hope for this project was to allow the user to control the volume of the audio playing, or pause it remotely when they are not in close proximity with their desktops or laptops. The remote device allows to do this even from long distances as long as it is connected to the same network as the website. The volume controller is a knob that rotates from one side to another. Beyond this, the device allows to adjust the background color of the website and customize it to the user's preferences. The background controller is also a knob that rotates from one side to another. The initial aim of the project also includes a system that detects whether the website is playing music above a maximum volume for a certain amount of time and sends an email notification to the user.

Now let's get to how this all works. First, we are going to cover the communication from the device. The device contains two knobs and two buttons controlled by an arduino. The knobs on the device are potentiometers, which are resistors with adjustable resistances (by rotating the knob). Hence, they can be programmed to send distinct signals depending on their resistance. The buttons also send a separate message when pressed and released. The arduino communicates with the server via a UDP protocol. This protocol allows the arduino to send packets to the server defined in the server.js script and also receive packets.

So a UDP packet (a signal) is sent to the server each time an event occurs. An event is defined as a change of state in either the rotation of either knob or a button pressed or released. So the size of the UDP is four, the first byte representing the volume knob, the second byte representing the background knob, the third representing the play mode and finally the fourth representing the pause mode.

After receiving these four bytes, the server reads them byte by byte and sends specific messages to the client via a socket. A socket is a channel created by a server to send signals in an efficient manner. So one example of a message sent to a client is "Volume knob rotation, 4". The first part of the message lets the client know what variable to change and the second part is the data which the variable is going to be assigned. By this manner, the client receives the data and assigns different variables the values sent from the server side.

The client contains code that governs the elements of the index.html (website). So for instance, during a volume change event the volume of all the music playing in the website will be set to the value of a variable in the client code named 'volume'. Also on another case, when the event corresponds to changing background, the background of the website will be set to the value of another variable in the client code named 'background'. Accordingly, during each event, the message from the server will be read and different variables, which in turn control elements of the website, will be modified.

The css file simply gives the style of the website and allows it to have a vivid and appealing nature. Simply saving the css file in the same directory as the html file will automatically link the two files.