BlindCrown
Vibration Motor Power Control
Software-Enabled User Experience

Andrei-Cristian Ilies, 10LF222, 2024

Introduction

Our device is designed to help impaired people to avoid collisions and feel comfortable while using the BlindCrown. Comfort in this context can refer to various aspects, addressing potential challenges a user may face when using our device.

Examples of situations a user might face: getting clear results from the vibration motors, adjusting the intensity of vibrations to their preferences, and navigating in different environmental conditions.

Vibration Power Usage

It is very important for impaired people to be able to control the power of the vibration motors so that they can feel comfortable and face all the challenges they will encounter in real life.

For the examples in the previous section, this is how vibration motors power adjustment helps:

- **Getting clear results from the vibration motors:** By adjusting the vibration power, the user can ensure that feedback from the motors is strong enough to be easily perceived but not very intense in order to cause discomfort or confusion. This aspect improves the clarity of the feedback, making it more effective to give feedback to the user of the potential collisions or environmental information.

- **Adjusting the intensity of vibrations to users preferences:** Every user has a different sensitivity to vibrations. Providing the ability to control vibration power allows users to reduce sensory overload or discomfort.
- **Navigating in different environmental conditions:** Environmental factors, such as noisy surroundings, might require stronger vibration power to ensure they are noticed. In quieter environments, the user might prefer a lower vibration power. This adaptability helps the device remain practical and functional in many real-world scenarios.

PWM

[1] PWM (Pulse Width Modulation) is a technique for getting analog results with digital means. [2] It can reduce the total amount of electrical energy supplied to a resistive DC (Direct Current) device by simply changing the percentage of time that the device receives its full rated voltage while being rapidly switched on and off.

[2] PWM is an effective solution for many load types, but the implementation must be chosen properly and must consider the response speed and power requirements of each individual device.

Vibration Power Adjustment using PWM

In order to adjust the intensity of an ERM (Eccentric Rotating Mass) vibration motor using PWM, we need to do the following steps:

1. Connect the motor's control pin to one of the PWM pins (3, 5, 6, 9, 10, and 11) of the Arduino UNO R3 microcontroller [3] (pin 9 in our code snippet).

2. In the Arduino setup() function, use the pinMode() function [4] to set the motor control pin as an output so the microcontroller can send control signals to the motor.

3. Use the analogWrite() function [5] to generate PWM signals on the motor's control pin; this function determines the average voltage delivered to the motor.

Connectivity via Azure

We are going to use Azure for our device the same way as we did at the Azure laboratories. The integration involves sending messages from the client to the .NET service in Azure using fetch API.

Fetch API allows us to send HTTP requests from the browser to the backend, where the request will be processed and a response will be returned. Here is a detailed description of the communication process:

- **Client - Frontend**: The user will use the frontend by pressing one of the two buttons (the upper half will have +, the other half will have -) to adjust the vibration intensity. Pressing the '+' button will increase the vibration power, while pressing the '-' button will decrease it. Our frontend will be made in a simple way, using simple technologies such as **HTML**, **CSS** and **JavaScript**. Once the user interacts with the buttons, the client (browser) will send an HTTP request to the backend service via the Fetch API. This request will include data such as the user's selected vibration power or other parameters for device control.
- **Azure .NET Service - Backend**: The .NET service hosted on Azure will receive the request, process the data and return a response.
- **Connectivity with the BlindCrown**: The message is being send from IoTHub to BlindCrown.

Code Snippets

1: Arduino Code for PWM Motor Control [6]

```

int motorPin = 9; // PWM pin connected to the motor driver
int intensity = 128; // Set intensity (0-255)

void setup() {
    pinMode(motorPin, OUTPUT);
}

void loop() {
    analogWrite(motorPin, intensity); // Control motor intensity
}
```

2: Example of fetch request [6]

```

fetch("[api_url]", {
  method: "PUT",
  mode: "cors",
})
  .then((response) => response.json())
  .then((json) => console.log(json));
```

References

[1] https://docs.arduino.cc/learn/microcontrollers/analog-output/

[2] https://control.com/technical-articles/understanding-the-basicsof-pulse-width-modulation-pwm/

[3] https://docs.arduino.cc/tutorials/generic/secrets-of-arduino-pwm/

[4] https://docs.arduino.cc/language-reference/en/functions/digital-io/pinMode/

[5] https://docs.arduino.cc/language-reference/en/functions/analog-io/analogWrite/

[6] https://chatgpt.com/