**Requirements for the firmware:**

The firmware can be defined from two ends. The user end and the belt end.

Firmware

Belt, Motors and Battery (Side B)

User Interaction (Side A)

**Side B definitions:**

1. **Things to control on each motor:**

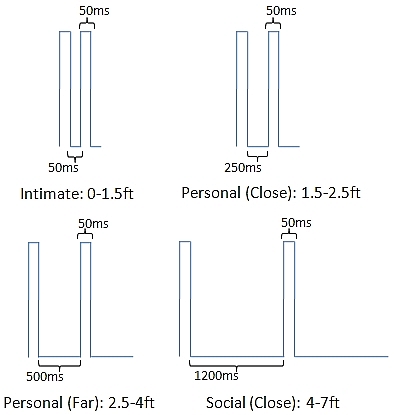
These are things the user should be able to control about each vibrator. An example format of user interface for this would be

**Vibrate Motor 180** (this parameter would depend on what configuration the belt is put on) **at Rhythm [A-E] at Magnitude [0-100] for Duration [0-3000(ms)]**

1. **Rhythm** (Rhythm corresponds to varying on-off cycles)

* The firmware at any time supports up to 5 learnt Rhythm patterns that will be referred to as A, B, C, D and E.
* The definitions for each of these patterns should be programmable into the controller at a later time if desired by user. (This requires a separate “learning” mode in the user interface).
* Each pattern, A through E, has three important parameters
  + On duration
  + Off duration
  + Length of each rhythm cycle

On



Length

Off

1. **Magnitude** (Strength of the vibration produced at each of the vibrator)

* There could be 4 unique strength levels for magnitude represented as 1, 2, 3 and 4.
* You can control the magnitude by dividing the “On” time into time slots and turning on and off the device within that short duration.
* The magnitude patterns 1, 2, 3 and 4 should be programmable from the user interface.

On

Off

1. **Duration** (The duration for which the vibrator should be working in its current rhythm and magnitude)

This is the duration for a single vibrator given in milliseconds. This controls how long the vibrator will continue to have the current rhythm and magnitude. This should be generally longer than the ‘length of each cycle’ parameter of the Rhythm to experience one full rhythm.

1. **Things to control on the entire belt:**

This controls the entire belt and the action of the each vibrator is somehow encoded into the patterns defined below. An example interaction of the user in the belt mode would be

**Move belt to Configuration [5, 7, SC, FC]**

1. **Configuration** ( The functional nature of the belt)

* This defines the way the belt is going to look functionally. There are four configurations that the belt can take 1, 2, 3 and 4.
* The user should be able to define what the configurations are. This should be done from the user interface. When each mode is chosen, there needs to be a fixed definition of what the motors will be referred to as. As accepted, we will use the angle to represent a vibrator.

1. **Spatio temporal pattern** (This refers to unique vibration pattern of the belt where more than one vibrator is involved)

* The belt supports one or more spatio temporal patterns.
* Each pattern is defined with respect to the origin vibrator of the pattern.

For example: **Vibrate belt in ST pattern 1 starting at Motor 180**.

* Each pattern is defined by what rhythm defines each motor and for what duration.

**Side A definitions:**

The user side can be broken down into two important segments.

1. **Console debug interface**

Console interface gives access to all the testing facilities and everyday functionality of changing the belt from one configuration to another. The complex task of programming the rhythm pattern and programming the spatio-temporal patterns will not be done.

1. **API interface**

This will be the all powerful interface. Access to all the facilities that we build into the belt would be accessed by using this interface. This should be able to do very simple tasks that are possible in the console mode and also complex tasks such as programming the patterns.

The API will also have a default GUI interface that can be used to visually alter patterns and program them into the belt. The GUI would be like a tool that users who do not want to mess with the API would use on their computer and then put the belt immediately to use.