

# Readme for Button Simulator

This is a general instruction about constructing a prototype for **Button Simulation** via FDVV models. **Please read thoroughly before starting the hardware prototype or software implementation.**

All the materials, code, data can be found at: <https://userinterfaces.aalto.fi/button-design>

Our paper (Button Simulation and Design via FDVV Models, CHI'20) can be found at:  
<https://dx.doi.org/10.1145/3313831.3376262>  
<https://arxiv.org/abs/2001.04352>

The document is created by Yi-Chi Liao ([yi-chi.liao@aalto.fi](mailto:yi-chi.liao@aalto.fi)) on 2020 Jan, 31st.

## 1. Introduction

In our paper (<https://dx.doi.org/10.1145/3313831.3376262>), we present a novel FDVV (force-displacement-vibration-velocity) model for better capture the physical properties of a button. Based on the model, we demonstrate render the tactility of modelled buttons using a button simulator. This project folder contains all the materials needed to replicate the system. For further understand the research contributions of our work, please check our paper (Button Simulation and Design via FDVV Models, CHI'20). We hope open-sourcing this project can motivate future researchers, engineers, and designers investigate the fundamentals of buttons and towards better haptic designs.

## 2. File Overview

The sub-folders contain in this folder is organized as such:

- 1. paper\_acmdl:** paper, auxiliary material, paper source code (latex).
- 2. button\_data:** contains the data of six buttons that we have modeled. There are 6 folders, each contains files of a corresponding button.
- 3. simulator\_prototype:** 3D print models, laser cutting layers, assembling instruction, and how to make the circuits.
- 4. source\_code:** programs run on 2 microprocessors – Adafruit ItsyBitsy and Arduino Uno.

### 3. General Instruction

#### 1. Prepare all the electronic components

##### Microprocessors:

Adafruit ItsyBitsy M0

<https://www.adafruit.com/product/3727>

Arduino Uno X 2

[https://www.amazon.com/s?k=arduino+uno&ref=nb\\_sb\\_noss\\_2](https://www.amazon.com/s?k=arduino+uno&ref=nb_sb_noss_2)

##### Linear position sensor:

MHR 250

<https://www.electronicsdatasheets.com/manufacturers/measurement-specialties/parts/mhr-250>

LVM-110 signal conditioner

<https://www.te.com/usa-en/product-CAT-PSI0006.html>

MAX17681 DC-DC converter

[https://katalog.we-online.de/en/icref/MAX17681-DB-MAX17681EVKIT-Rev-1\\_Buck\\_1](https://katalog.we-online.de/en/icref/MAX17681-DB-MAX17681EVKIT-Rev-1_Buck_1)

##### Linear force actuator:

Moticont HVCM-025-022-003-01

<http://www.moticont.com/HVCM-025-022-003-01.htm>

DRV8871 DC Motor Driver Board

[https://www.yeint.fi/elektroniikka/development-tools/internet-of-things/adafruit-drv8871-dc-motor-driver-br?gclid=Cj0KCQiAvc\\_xBRCYARIsAC5QT9kZnoEcDxlqUGjKIAIgjoiY3VakaAh1fXZxD4W5rhSpwIII-mDv4u8aAnmFEALw\\_wcB](https://www.yeint.fi/elektroniikka/development-tools/internet-of-things/adafruit-drv8871-dc-motor-driver-br?gclid=Cj0KCQiAvc_xBRCYARIsAC5QT9kZnoEcDxlqUGjKIAIgjoiY3VakaAh1fXZxD4W5rhSpwIII-mDv4u8aAnmFEALw_wcB)

##### Vibration motor:

Tectonic TEAX13C02-8/RH

<https://www.parts-express.com/tectonic-teax13c02-8-rh-13mm-exciter-8-ohms--297-214>

Adafruit Wave Shield for Arduino Kit

<https://www.adafruit.com/product/94>

## **Others:**

SG90 tower pro

<https://www.banggood.com/TowerPro-SG90-Mini-Gear-Micro-Servo-9g-For-RC-Airplane-Helicopter-p-1009914.html>

Sparkfun logic level shifter

<https://www.sparkfun.com/products/12009>

16V Power supply X 1

9V Power supply X 1

## **2. 3D print models and laser cutting layers**

Print all the 3D components (stl files) under the “3. simulator\_prototype/3D\_print\_models”.

Laser cut all the layers needed under “3. simulator\_prototype/Laser\_cutting\_layers” with at least 3 mm thick materials.

## **3. Physical prototype**

Follow the instructions in “3. simulator\_prototype/Prototype Assembling Instruction.pdf” to construct the physical prototype.

## **4. Circuits**

Follow the instructions in “3. simulator\_prototype/Circuit Installation.pdf” to connect all the circuits.







## **5. Software installation**

In order to upload the code onto the microprocessors, you have to download the Arduino IDE. Furthermore, you must follow the steps here to have the driver of ItsyBitsy M0 Express board: <https://learn.adafruit.com/introducing-itsy-bitsy-m0/setup>

After confirming that the connection from laptop to both microprocessor is ready, find the programs in “4. source\_code” folder.

The “button\_simulation\_chi20” folder contains the code “button\_simulation\_chi20.ino” that should be uploaded to the main board – Adafruit ItsyBitsy M0. The “button\_wavehc\_chi20” folder contains the code “button\_wavehc\_chi20.ino” which is for the Arduino Uno that attached with soundwave shield. All the file under the folder “all\_sound\_wave” should be uploaded to the SD card embedded on the Arduino soundwave shield.



 cherry\_mx\_black\_data.txt  
 cherry\_mx\_brown\_data.txt  
 cherry\_mx\_clear\_data.txt  
 cherry\_mx\_red\_data.txt  
 hp\_pr1101u\_data.txt  
 macbook\_pro2011\_data.txt

```

1 // Replace this section below with the button you want to simulate.
2 // =====
3 // ===== THIS IS CHERRY MX BROWN =====
4 // ===== Set the travel range according to your simulation target =====
5 int travel_range = 4000;
6
7 // ===== Button Data =====
8 float button_v0_Down[] = {10.000,29.730,32.670,32.340,32.330,32.760,32.910,33.560,34.270,34.680,35.070,35.290,36.280,36.340,36.540,3
9 float button_v0_Up[] = {10.000,10.000,14.440,21.670,22.440,22.480,23.320,22.740,23.270,23.190,23.520,24.480,24.400,24.790,25.240,2
10 float button_v1_Down[] = {10.000,21.550,24.498,25.902,27.474,29.267,30.915,32.765,34.425,35.547,36.345,36.858,37.678,38.061,38.532,3
11 float button_v1_Up[] = {10.000,10.000,15.750,21.118,23.264,25.044,27.221,28.624,30.282,31.038,31.502,32.145,32.293,32.681,33.127,3
12 float button_v2_Down[] = {11.347,13.369,16.325,19.464,22.619,25.773,28.920,31.970,34.580,36.415,37.620,38.426,39.076,39.781,40.523,4
13 float button_v2_Up[] = {11.504,13.761,17.061,20.566,24.088,27.609,31.121,34.509,37.295,38.887,39.485,39.810,40.186,40.572,41.015,4
14 float button_v3_Down[] = {10.991,12.478,14.652,16.962,19.282,21.602,23.933,26.359,29.085,31.761,33.743,35.377,36.785,38.010,39.271,4
15 float button_v3_Up[] = {11.558,13.898,17.317,20.950,24.599,28.248,31.885,35.358,38.047,39.248,39.407,39.578,40.237,41.418,42.883,4
16 float button_v4_Down[] = {10.635,11.587,12.980,14.459,15.945,17.432,18.945,20.748,23.590,27.106,29.867,32.329,34.495,36.238,38.019,3
17 float button_v4_Up[] = {11.613,14.034,17.573,21.333,25.110,28.887,32.648,36.206,38.800,39.609,39.328,39.346,40.288,42.264,44.752,4
18 float button_v5_Down[] = {10.520,11.300,12.441,13.652,14.870,16.088,17.328,18.810,21.223,24.526,27.575,30.148,31.959,33.415,35.297,3
19 float button_v5_Up[] = {11.860,14.650,18.729,23.063,27.416,31.770,36.101,40.154,42.962,43.798,43.852,44.227,45.412,47.981,50.477,5
20 float button_v6_Down[] = {10.405,11.013,11.902,12.846,13.795,14.744,15.710,16.872,18.856,21.946,25.283,27.967,29.422,30.591,32.576,3
21 float button_v6_Up[] = {12.106,15.266,19.885,24.793,29.723,34.653,39.554,44.101,47.124,47.987,48.376,49.107,50.536,53.699,56.202,5
22 float button_v7_Down[] = {10.207,10.518,10.973,11.456,11.941,12.426,12.922,13.546,14.732,16.824,19.573,22.457,24.599,26.258,28.543,3
23 float button_v7_Up[] = {12.024,15.062,19.503,24.221,28.960,33.700,38.465,43.472,49.176,54.297,56.439,56.860,58.504,61.864,64.398,6
24 float button_v8_Down[] = {10.009,10.023,10.043,10.065,10.086,10.108,10.134,10.221,10.607,11.701,13.863,16.948,19.777,21.924,24.509,2
25 float button_v8_Up[] = {11.943,14.858,19.120,23.648,28.197,32.748,37.375,42.843,51.228,60.607,64.503,64.613,66.472,70.029,72.595,7
26 float button_v9_Down[] = {10.000,10.000,10.000,10.000,10.000,10.000,10.000,10.000,10.000,10.000,10.000,11.439,14.955,17.590,20.476,2
27 float button_v9_Up[] = {11.861,14.654,18.738,23.075,27.434,31.795,36.285,42.214,53.280,66.917,72.567,72.366,74.440,78.194,80.791,8
28
29 // ===== Vibration settings =====
30 bool if_vib = true;
31 int vib_point_1 = 1600;
32 int vib_point_2 = 1900;
33
34 int vib_file0 = 20;
35 int vib_file1 = 20;
36 int vib_file2 = 18;
37 int vib_file3 = 18;
38 int vib_file4 = 4;
39 int vib_file5 = 4;
40 int vib_file6 = 5;
41 int vib_file7 = 5;

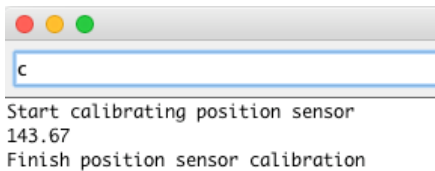
```

## 7. Calibration of the button

There are three calibrations you may use to help improve the accuracy.

### a. Real-time position calibration

While running the system, you can open the Serial Monitor (when the serial port is toward ItsyBitsy) and type character “c”. It will re-calibrate the position sensing reference, and use the current displacement as the new origin (0 mm) for future displacement sensing.



Because the position sensor has (most) linear properties, meaning the sensor value is linear to the actual displacement, such a calibration will be sufficient for most of the time.

However, if you found the displacement reading very unusual, another round of full calibration will be required. Please see the following point.

### **b. Full calibration of the position sensor**

If you view the program “button\_simulation\_chi20.ino”, you will find these two parameters, which determine the relation between the actual displacement and the reading. You need to change the values of these two parameters to have a very accurate sensing.

```
float pos_intercept = 14565.7522;    // These two parameters (pos_intercept, pos_slope) may require
float pos_slope = -3.9103;          // manual calibration using linear regression of the position sensor.
```

In order to do so, you must equip another precise tool for poking the button with accurate depth capability. Meaning, you have to gather a set of <depth, sensor value> data, and then conduct a linear regression.

### **c. Calibrating the travel range control**

For better control the travel range, you need to manually set different degrees of the sg90 servo motor and record a set of <servo degree, travel range> data then run a linear regression. Afterward, modify the parameters below in the “button\_simulation\_chi20.ino” program.

```
float travel_intercept = 158.77735;
float travel_slope = -0.021463;
```