Hand Force Controller User Manual

Lillian Boettcher

Grant Kveton

Dhaval Manojkumar Shirvi

Contents

Getting Started

The hand force controller and software consists of:

- A 3D printable platform
- Eight force sensors
- Multiplexer chip (such as 74HC4051 breadboard chip)
- Wires, wire jumpers, and/or other conductive connectors
- Arduino Uno
- USB cable
- Arduino code script
- Hand Force Controller Application

If you are manually assembling the hardware components and platform, the instructions are as follows:

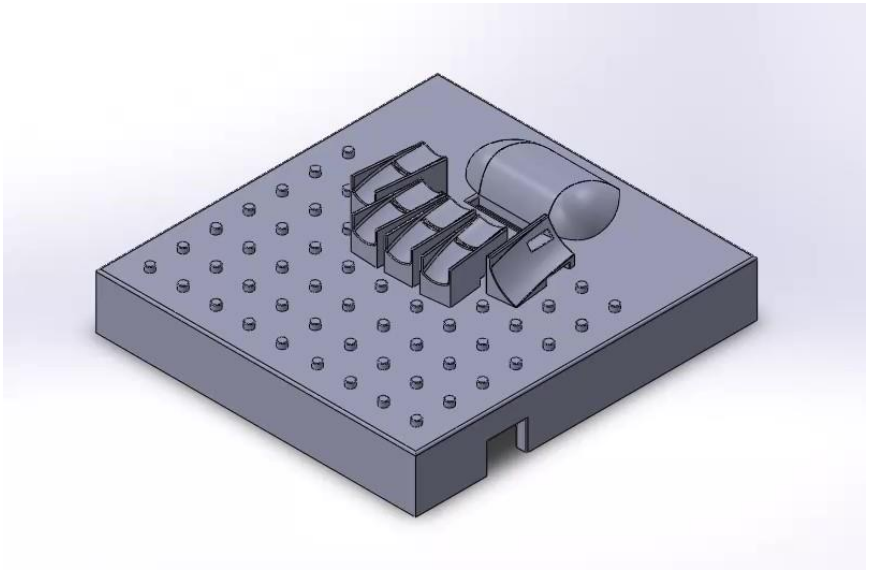
Manual Hardware Assembly

The force sensors need to be connected to the Arduino Uno by the use of wires, so a breadboard or other wire platform is required. The connections require having the first five force sensors connected to the first five analog ports on the Arduino. The remaining three force sensors should go through the multiplexer. The multiplexer connections are specific to whatever chip is used, as the type of chip used will define how the sensors will be connected.

Manual Platform Assembly

The components of the platform can be found in the repository as three separate files, a “finger” model, a “thumb” model, and a “base” model. To have a complete platform, the base and thumb are printed once while the finger model is printed four times.

The base consists of round pegs, which the thumb and finger models can fit into after they are printed. The wires to the sensors can be put through slits found in each of the fingers, including the thumb, or underneath, depending on what allows them to connect with where the Arduino is placed.



Arduino Script Functionality

The Arduino can be connected using a USB cable or a Bluetooth dangle. The script to be run on the Arduino Uno, main.ino, can be downloaded from the Github Repository in the

Arduino folder. The Arduino software can be downloaded from www.arduino.cc/en/Main/Software.

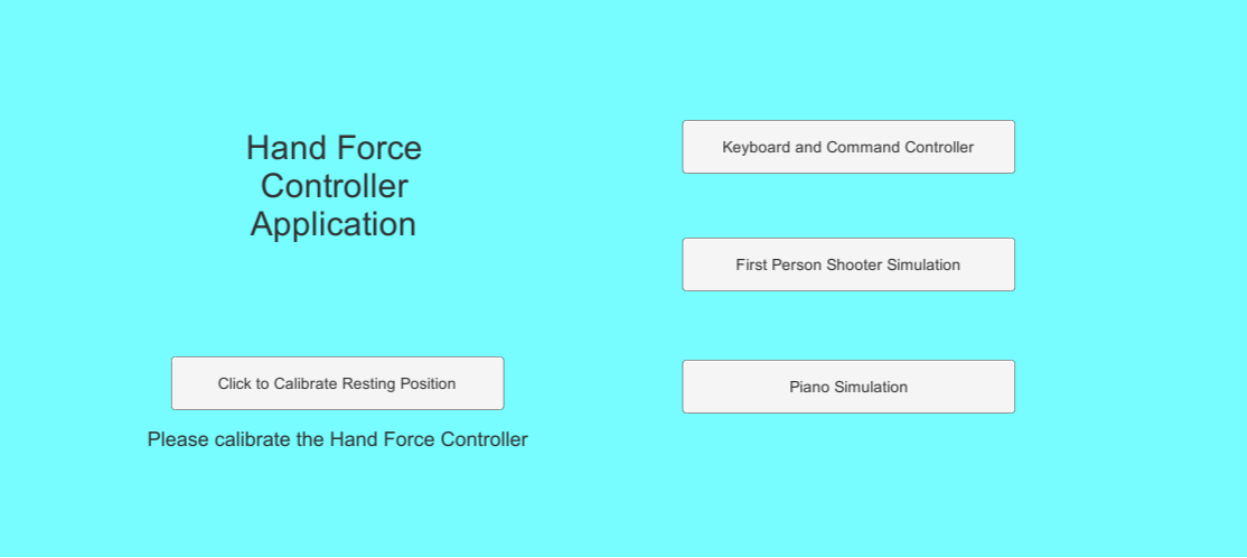
When the software is downloaded and installed for the correct operating system, open the software. In the File tab on the top right, and select Open. Then navigate to the downloaded script. When the script is open and the Arduino is connected to the computer (and recognized by the computer), click the Upload arrow at the top left. This will load the script onto the Arduino. If the Serial Monitor button on the top right of the software is clicked, a serial port reader will open. If the Arduino script is running and the force sensors are connected properly, the readings from each force sensor should appear on the port in live time.

Main Software Application

The Hand Force Controller Application can be downloaded from the Github repository. When it is downloaded into a folder, it will contain an executable and the application assets. There are two ways to access the functionality: opening the executable (the simplest method) or by opening the scene assets (for developers).

The scenes, which can be individually by developers, can be opened if Unity 3D is downloaded by the user. Then the source code and assets in the program can be viewed, exactly as they were released as an executable.

When accessing the application through the executable, a main screen will be displayed with calibration and buttons for accessing different parts of the application.



Calibration

The Hand Force Controller can be easily calibrated on the main screen of the software application. There is a single button for the calibration, underneath the title text. Calibrating can be done by having the hand that will be used rest on the Hand Force Controller and then using the other hand, while the controller hand is resting on the controller, to click the calibration button.

For the calibration button to work, please make sure that the controller is plugged in. If the controller is not plugged in, an error message will display prompting the user to plug in the controller to the computer. Additionally, the Arduino script needs to be running on the Arduino, else the software will not be able to receive information from the Arduino to sense the force sensor values being created by the user.

When the calibration process is complete, the response text underneath the button should display “Calibrated!” The calibration should save when testing on one of the simulations. If in the simulation the force sensors do not respond as desired, as in they are too sensitive or not sensitive enough, this can be fixed by recalibrating the sensors through the same process.

Piano Simulation

The piano simulation is opened as a Unity scene via the main software application’s main page. The simulation displays a piano from a top orthogonal view along with five color-

coded fingers. The red finger corresponds to the left-most sensor and the blue finger corresponds to the fifth sensor, which will be the thumb and pinky finger or vice versa depending on whether or not the user is using their left hand or right hand. The next two sensors, which are default to be placed underneath the sides of the palm of the selected hand, indicate left or right movement. When the sixth sensor is pressed, all five fingers move left. When the seventh sensor is pressed, all five fingers move right. The eight sensor is default considered to be placed at the back of the hand with some strap or board keeping it pressed against the back of the hand. Upon putting pressure on it through raising the back of the hand, the five fingers on the piano move forward. Upon release of pressure, the fingers move back before the black keys and onto only white keys.

Each key press corresponds to an audio output of the corresponding key. When the hand is not moved left, right, or forward, the default notes that can be sounded are C4, D4, E4, F4, and G4. This is because the left-most finger starts by default on middle C, or C4. If the hand does not start on C4, then you may need to check your hardware or recalibrate the sensors so the hand does not move without user intent.

To visually see what notes are being pressed, when a note plays it changes its color. When a note stops playing, it reverts to its original color. Black keys when played are set to change from black to purple when played and white keys are set to change from white to pink when played. A note changing color should correspond to note audio being played. If no notes are heard when keys are pressed, then you may need to check that your audio is turned on for the software application.

FPS Simulation

A first-person shooter (FPS) consists of the user taking a virtual weapon, pointing, and shooting that weapon at objects or enemies in a first-person view. This part of the application demonstrates the Hand Force Controller's ability to be used for playing FPS games and to be used when developing games, as this FPS was developed.

The goal of the game is to get to the end with the user's gun weapon. During the game, zombies will attempt to approach and damage the player. The user can attack these zombies to defeat them by shooting them with ammo. The ammo is collected as glowing objects along the way. The user wins if they can get to the end without losing all of their health.

The controls to play are as follows by default (referring to sensors as 0-7):

- Move forward: 2
- Move backward: 6
- Move left: 1
- Move right: 3
- Rotate left: 5
- Rotate right: 7

- Jump: 4
- Fire: 0

The controls can be altered by accessing the scene in Unity by programmers and developers. There is an "HF Config" script that can be viewed in the Unity editor. The values are set to the above controls by default, but they can change by changing the values set in the HF Config variables, where they can be set to values from 0 to 7 for sensors 1 through 8.

Keyboard Emulation

The Hand Force Controller can be used to emulate computer commands to control games and different computer software. This is done by choosing which gestures, or which sensor inputs, correspond to which keyboard or computer mouse commands. After any desired gesture commands are set, the controls can go into effect by pressing "Play."

What this does is opens an executable Python file, which runs how the sensors should be used based on the commands set on the computer emulation screen. This means that when the user performs a gesture on the Hand Force Controller that is set to activate a computer commands, such as press a keyboard key, the command will take effect. This allows the Hand Force Controller to be used to play computer games and control other computer and software actions. For example, say a game the user wants to play has the attack button set to the spacebar on the computer keyboard. If the user, on the keyboard emulation settings, set the first force sensor to activate the spacebar, then the user can press the first force sensor to attack in the game.