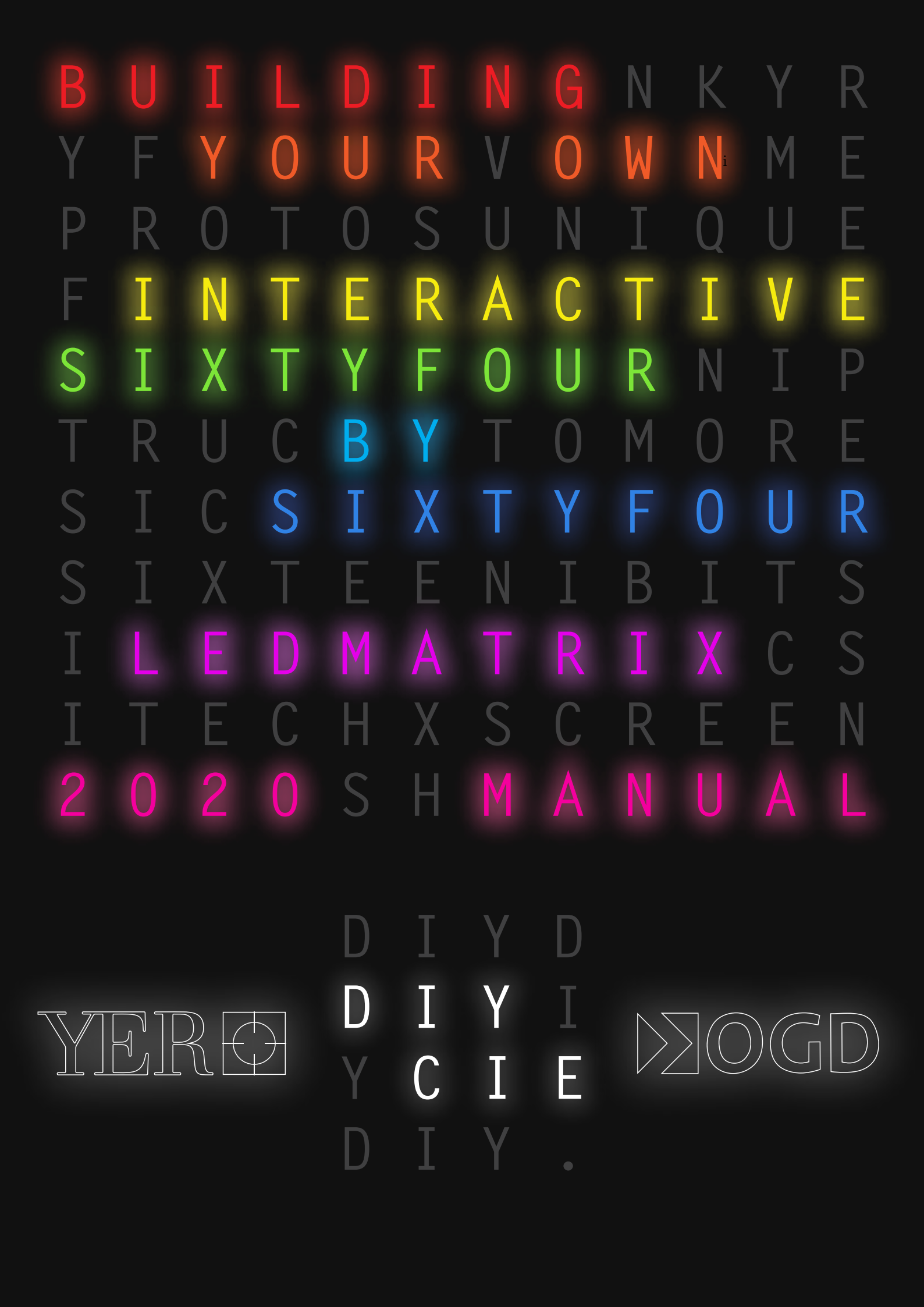

BUILDING NKYR
YFYOURVOWN ME
PROTOSUNIQUE
FINTERACTIVE
SIXTYFOURNIP
TRUCBYTOMORE
SICSIXTYFOUR
SIXTEENIBITS
ILEDMATRIXCS
ITECHXSCREEN
2020SHMANUAL
DIYD
DIYI
YCIE
DIY.

Preface

Welcome to DIYcie's 2020 workshop: How to build your own LED matrix!
In this manual we will explain how to build this year's project by walking you through the different steps that will lead to a working end product: A LED-matrix arcade machine!

This manual will however only be the first step in building your machine. After the workshop we will publish a handful of different case designs for the machine that can be downloaded and subsequently laser-cut at Alfred's office in the SmartXP (for the usual fees). From now on you can check the Proto wiki for all materials of this workshop and the SmartXP wiki for more information on laser-cutting.

This entire project is Open Source: the software, hardware and designs can be found online. With this open-source environment, we also stimulate you and others to add new ideas to our directory; so if you would like to add any code for the matrix you are free to do so, as it is with new designs, form factors or additional hardware implementations. This will also be further explained in the software portion of this manual.

If the manual isn't clear to you or you encounter any problems while following this tutorial please feel free to ask the committee members for help, that's why we are here! You can recognize us by our red and very stylish sweaters.

This workshop is made possible with help of the board of the Bachelor Creative Technology and Master Interaction Technology at the University of Twente, as well as OGD, YER, Ben's Electronics, Van Alles En Meer and Pi3g.

Enjoy!

The DIY-Cie
Stijn, Tijmen, Emiel, Jelle, Harald, Dennis, Jonathan and Maartje.

Table of Contents

Chapter 1

Getting Started

To start, check if you have all the parts and tools ready. These will all be necessary to complete this project. We have chosen to fit the matrix into the arcade style housing shown in Figure 1.1. Hereafter you can start assembling this housing.

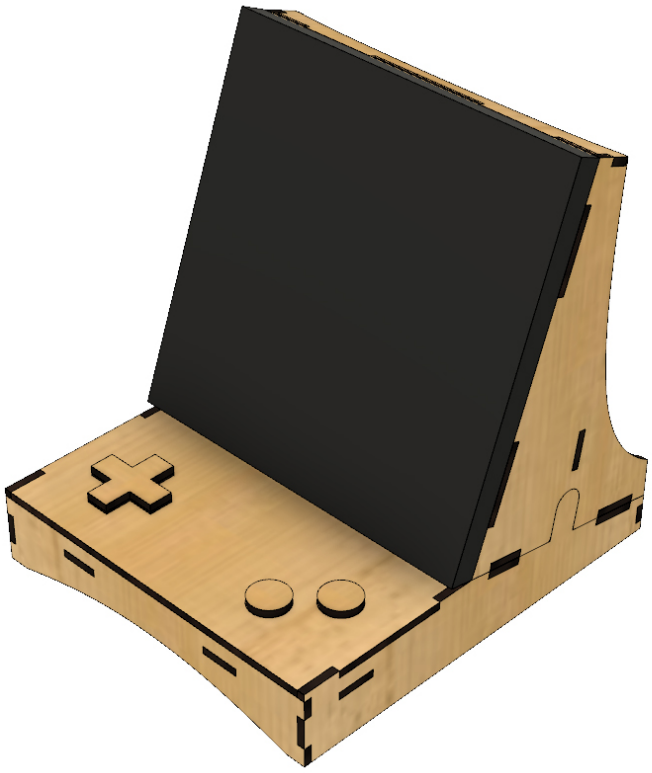


Figure 1.1: Overview of arcade style housing, the end-product

1.1 Parts Lists

Your kit should contain the following items:

- 1 Lasercut frame
- 1 64x64 LED matrix
- 1 SD card
- 1 LED matrix power cable
- 1 LED matrix data cable (ribbon)
- 1 Raspberry Pi Zero W unsoldered
- 1 Raspberry Pi header pins 2x20
- 1 Raspberry Pi shield RGB LED Matrix
- 1 Power Supply 5V 3A micro USB
- 7 M3 16mm nuts with bolts
- 4 M2.5 8mm nuts with bolts
- 7 12x12mm buttons
- 1 Perfboard 3x3.5cm (±10x10)
- 1 WF-Connector 8-pin (male end)
- 1 WF-connector 8-pin (20cm) (female end)
- ca. 70cm red wire
- ca. 70cm black wire

Before you start, please make sure all of these parts are in your kit. If this is not the case, tell one of the committee members which parts are missing from your kit and they will provide you these. This list, including order-links can also be found online.

1.2 Tools list

For this workshop you should also have the following tools at hand. We assume you brought a laptop, as it is required for the software part.

Tools

These are tools you need before starting the workshop:

- Laptop
- Micro-SD adapter
- Screwdriver set
- Pliers
- Ruler
- Sand-paper (optional)
- File (optional)
- Soldering iron
- Solder
- Wire stripper
- Wood glue
- Multimeter
- Breadboard

Workstations

We also set up a workstation which is provided with the following equipment (m = male end, f = female end):

- Micro-USB [m] to USB cable [f]
- Mini-HDMI [m] to HDMI adapter [f]
- USB-hub (optional)
- HDMI-to your screen [m/m]
- A screen
- Keyboard

1.3 General outline of the workshop

The workshop and this manual are set up in such a way that assembling the housing (Chapter 2), working on all the hardware (Chapter 3) and configuring the Raspberry Pi (Chapter 4) can be done in arbitrary order. However, the recommended order is as listed above. This is due to the facts that glue has to dry and working on the Pi inside the housing can be difficult. Because we will change code trough an SSH connection, the Raspberry Pi must be configured for this action and you should at least follow Section 4.1.2 before assembling

the Raspberry Pi. Other than that, feel free to go through this manual in the order you prefer.
During the workshop, configured SD-card will be provided. This means that you can follow this manual in order. More details will be provided in Chapter 4. Whenever you get stuck or something goes wrong, first check out Chapter 5, which contains the Troubleshooting Chapter. If this does not answer your questions, don't hesitate to ask someone of the committee.

1.4 3D-model

Besides all the visuals such as diagrams and photos, accompanying the steps in this manual, also a 3D-model is available. In this model, all the hardware and electronic components are included as well as the wires to be soldered. You can find this model online, the password will be provided during the workshop. Please note that the wiring in the model is modeled to the most ideal situation and that this wire management will most likely be cleaner than what is physically possible in your installation. To hide parts of the 3D model, you can right-click on the part and choose *Hide selected*.

Chapter 2

Housing

2.1 Assembling the housing

This chapters walks you trough assembling all the wooden pieces into a nice arcade style housing. Note that these pieces have two visually different sides, as you can see by their wood burn marks. It's your own design choice to point these sides in or out, but keep in mind that there is a difference between both sides. Also note that for some pieces, the orientation does matter, but this will also be pointed out in the relevant steps.
Unless stated otherwise, apply glue to all the surfaces that will be in contact. It is best to squeeze some glue out of the bottle onto a scrap piece of wood or paper and use another small scrap piece of wood or something similar to apply the glue to the joints. Also remove the excess glue that squeeze out of the joints while this glue is still wet, this is easier than removing it afterwards.
Note that the structure can feel unstable while you are assembling it. All the pieces give each other strength and create a good structural integrity. You can use painters tape to press all joints together while the glue is drying. It also might be helpful to check if the pieces fit together nicely prior to glueing them together.
The steps below will help you make the two parts shown in Figure 2.1. Follow the steps carefully and also read the short texts in each step.

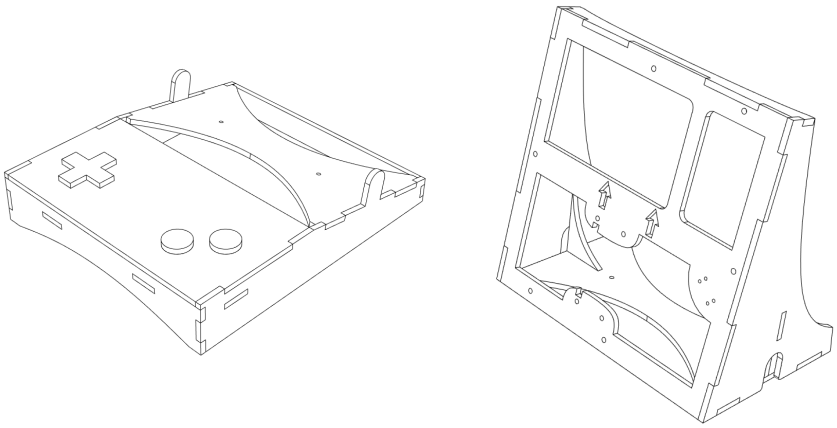
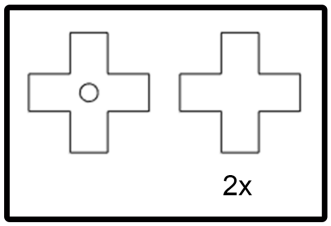
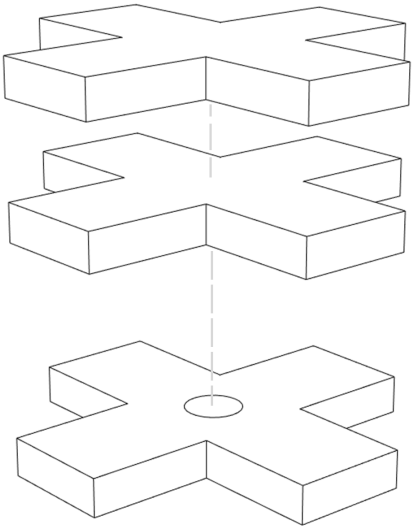


Figure 2.1: In this section, the base (left) and top (right) will be made

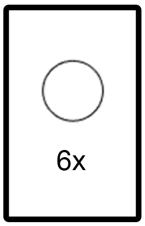
Footer

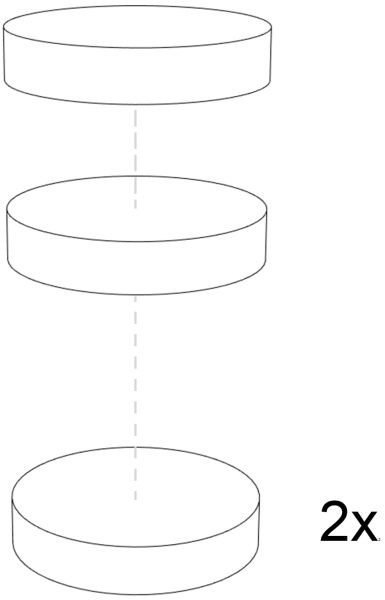
Step 1 We start by glueing together the D-pad. Note the position of the piece with the hole.



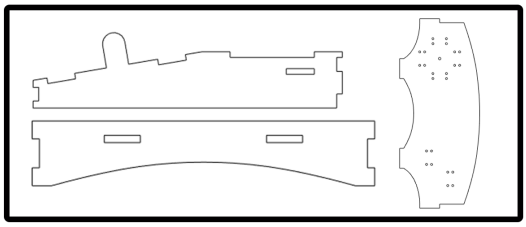


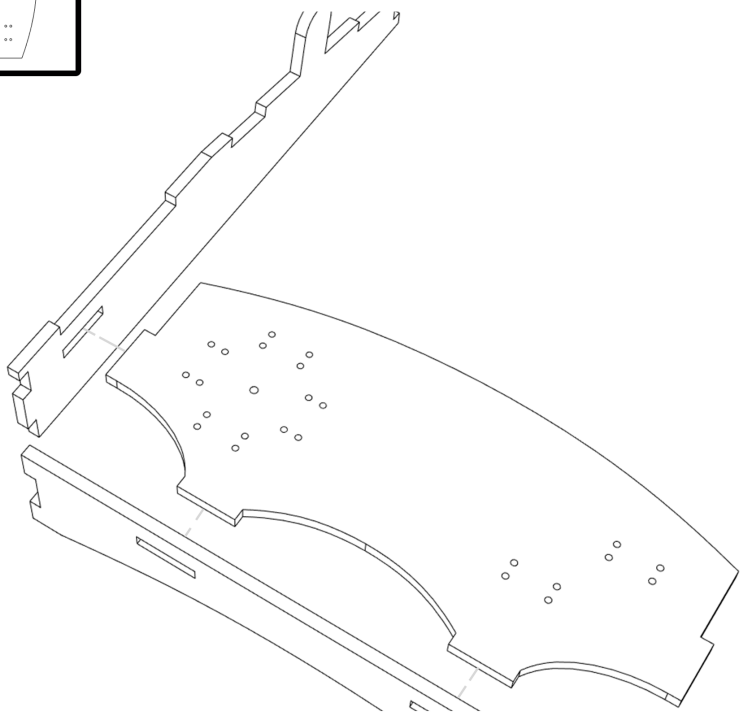
Step 2 Also glue together the two round buttons.



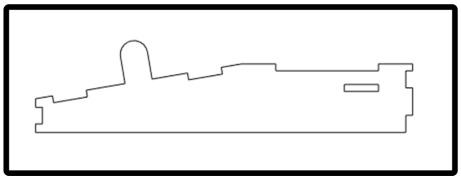


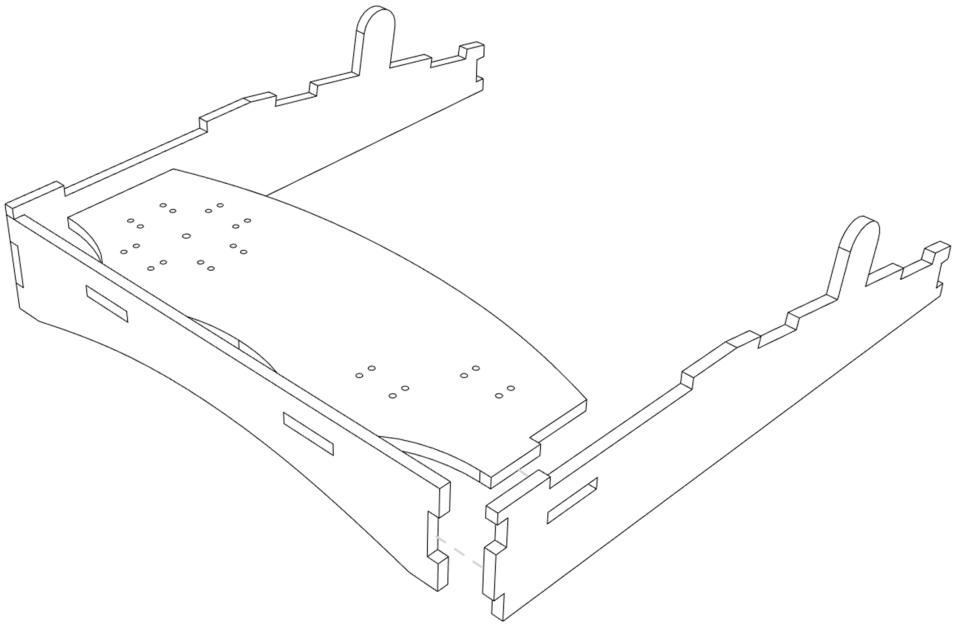
Step 3 Now we start with the assembly of one corner of the footer. At this point you can decide what button layout you want; the D-pad on the right or left side?



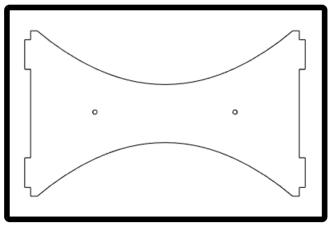


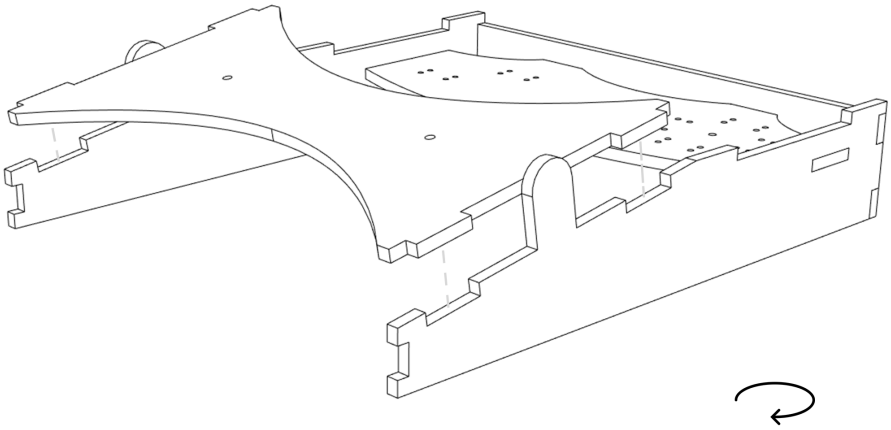
Step 4 Glue the right side in place.



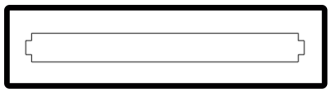


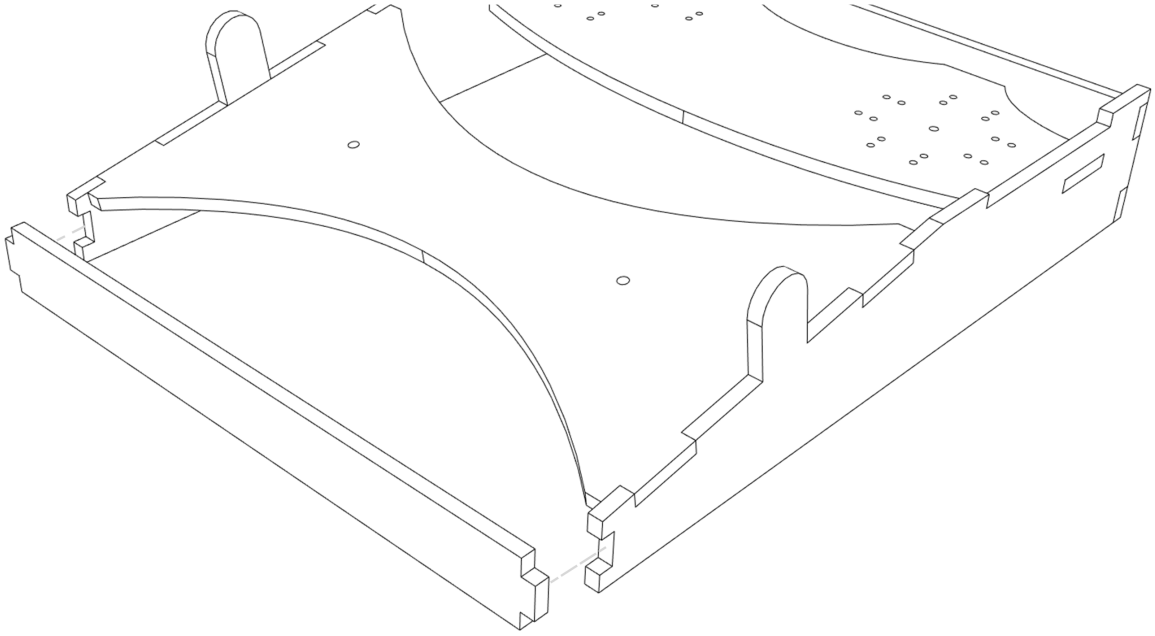
Step 5 We now glue in place the so-called *floor*. In your kit, there are two very similar pieces. Make sure you use the one that is wider in the middle; without the slots.



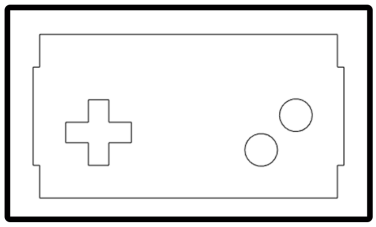


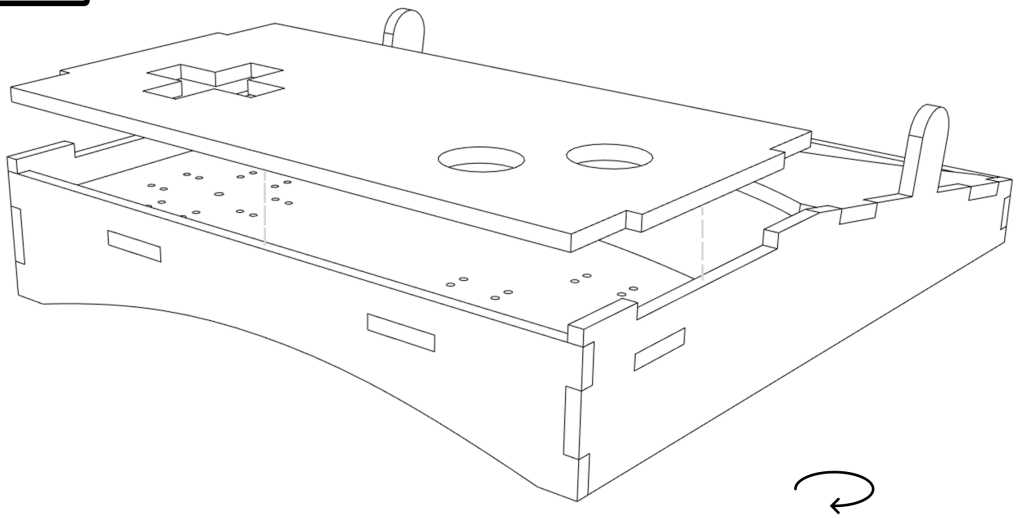
Step 6 Glue the back in place .





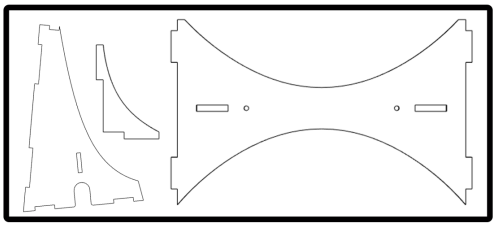
Step 7 For now, just place this last piece loose on top of the footer. **Do not glue this part** in place yet, since we still need to access the space underneath it.

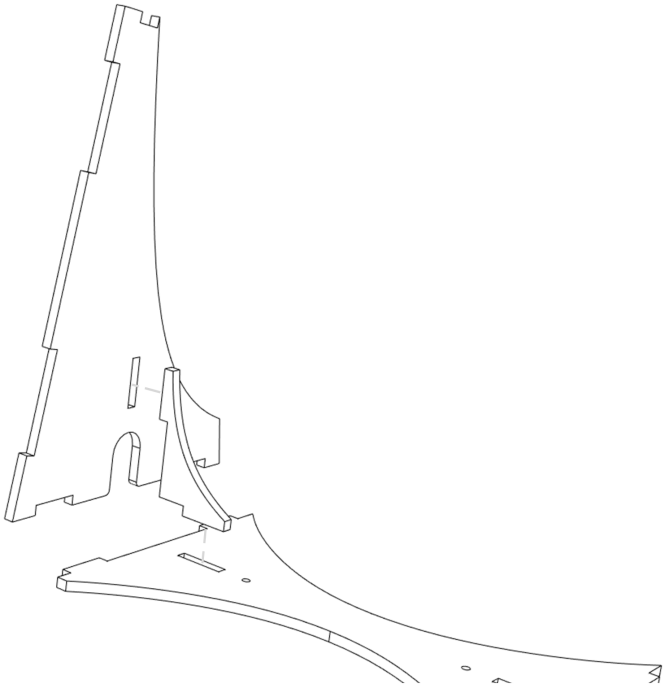




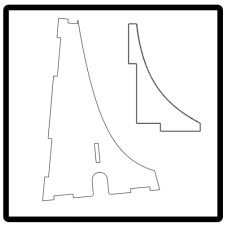
Screenstand

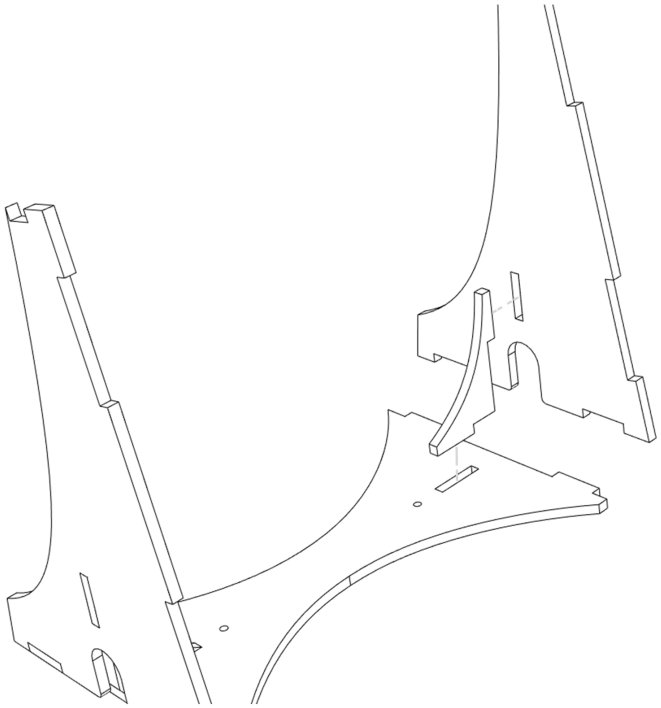
Step 8 Next we assemble the screenstand. Start as shown below.



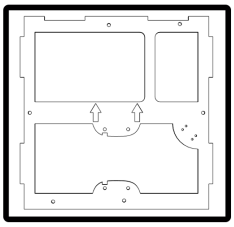


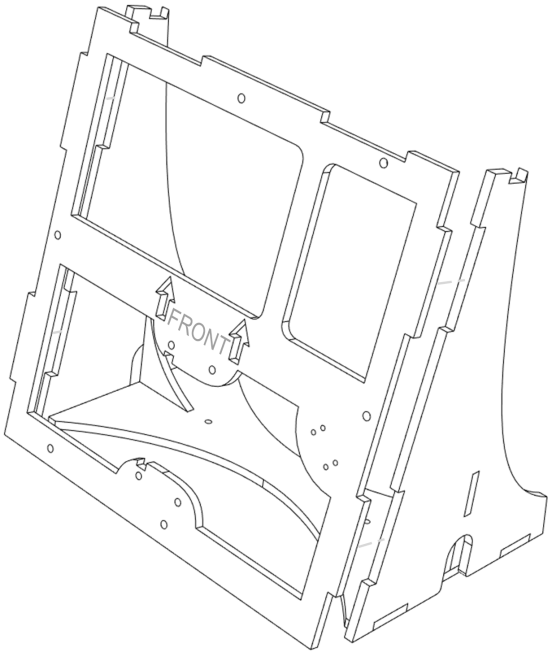
Step 9 Also glue the other side in place.



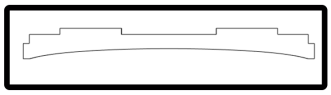


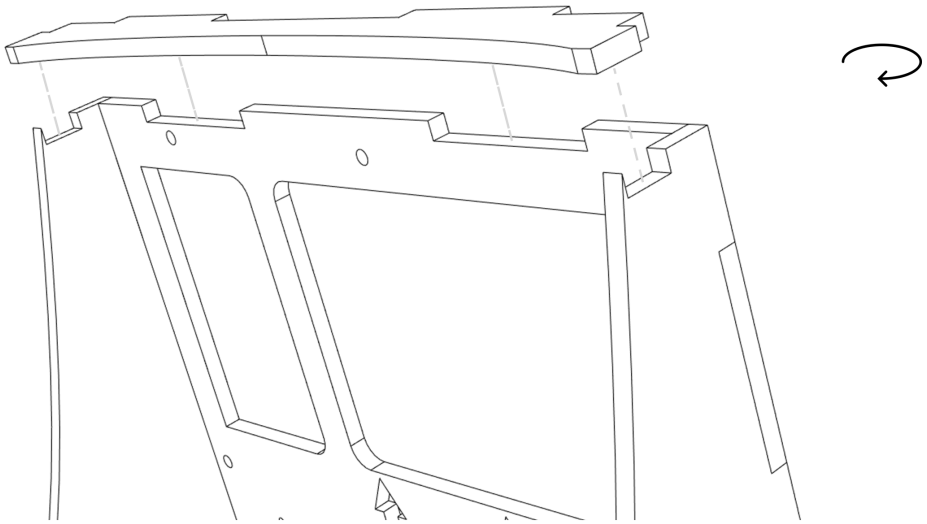
Step 10 Glue the front in place. Please note the orientation of this piece since it should be placed in exactly this way. See the front marking.





Step 11 Finish the screenstand by adding the top.





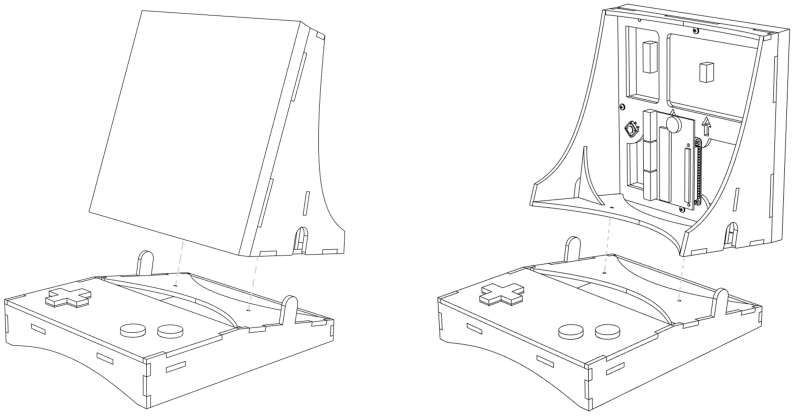


Figure 2.2: Two options in which the screen can be put

2.2 Different Form Factors

If you are not using the buttons or want to use your machine as a display, you can set up different form factors of the design, as shown in Figure 2.2. When you disconnect the button-connector and hide it behind the screen, you can set up your display-stand.

Chapter 3

Hardware

The device is split up into two parts, the screenstand and the footer. The screenstand contains the LED-matrix and its controller, the Footer contains the control buttons. You will be able to detach the screen from the body in a way where you take the controller with it but leave the buttons behind, for a more idle mode.
The Hardware in this machine is sketched in Figure 3.1. Here you can see that the footer contains a set of buttons and a connector piece. The screenstand contains the Raspberry Pi, the Matrix, one button and a connection wire for the buttons. This chapter will contain a lot of soldering. We will walk you through, step by step. An overview of all solder connections between different components, is shown in Figure 3.2.
If you're relatively new to soldering or need some more information on how to solder, look in Section 5.2.2 for tips and tricks.

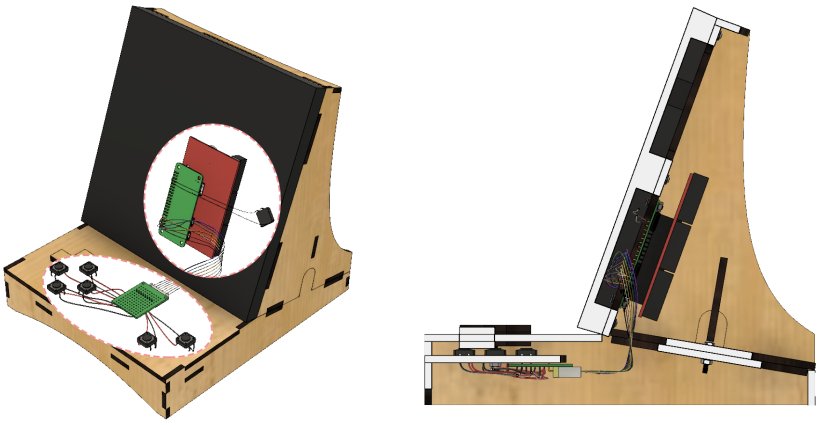


Figure 3.1: Location of all the components within the device

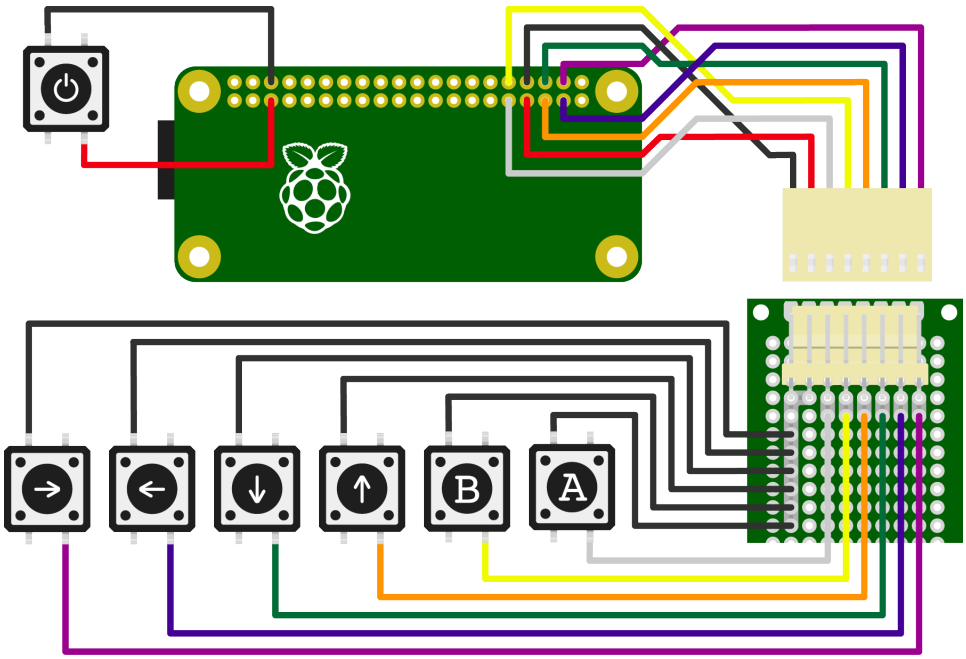


Figure 3.2: Overview of all the wiring in the device

3.1 The Footer

We start by setting up the footer containing the buttons and a WF-connector piece, all soldered to a small perfboard. The WF-connector is a white 8 pin male connector.

3.1.1 Buttons

Most buttons on your machine are located in the Footer, all housed in the double layer. The buttons will be connected to a pcb containing an 8-pin-WF-male-connector. We will solder one extra button, which is housed in the screenstand. Connecting wires to these button and securing them to the housing goes in the same fashion for all buttons.

Step 0 Start by cutting 7 pieces of red wire and 7 pieces of black wire of approximately 10cm and strip both ends ($\pm 3mm$). You should now make sure you have the following materials ready, see Figure 3.3.

- PCB (perfboard)
- 7 pieces of red wire ($\pm 10cm$)
- 7 pieces of black wire ($\pm 10cm$)
- 8-pin-WF-male-connector
- 7 Buttons

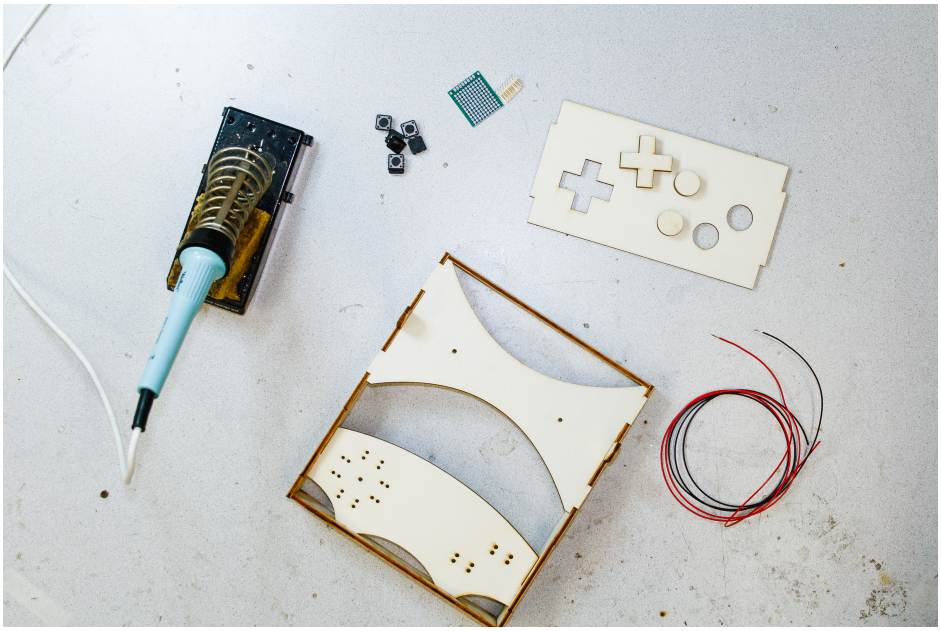


Figure 3.3: All materials for the footer.

Step 1 Solder a red wire and a black wire to all 7 buttons, connecting them diagonally opposite as shown in figure 3.5 and Figure 3.4.

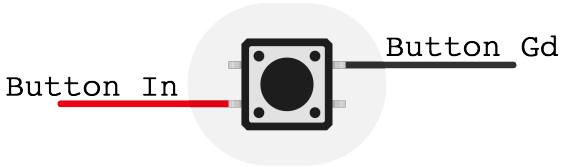


Figure 3.4: The way the buttons should be soldered

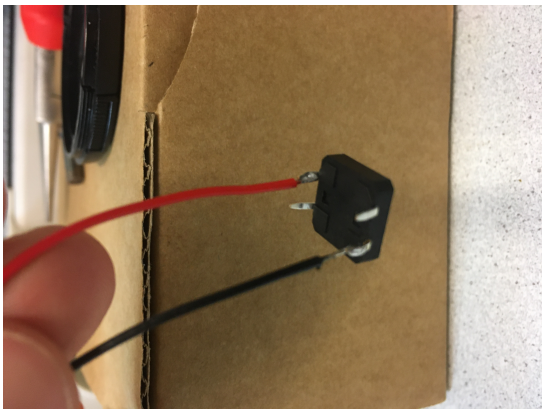


Figure 3.5: Buttons soldered

Step 2 Now push six of the buttons with wires connected through the designated holes in the footer. See figure 3.6. Make sure to put the correct wire through the correct opposite hole in such a way that the two unsoldered legs of the buttons fall into the other two holes. Set one button aside for the steps in Section 3.2.1. The layout of the buttons is shown in Figure 3.7, however, note that you can flip this layout.

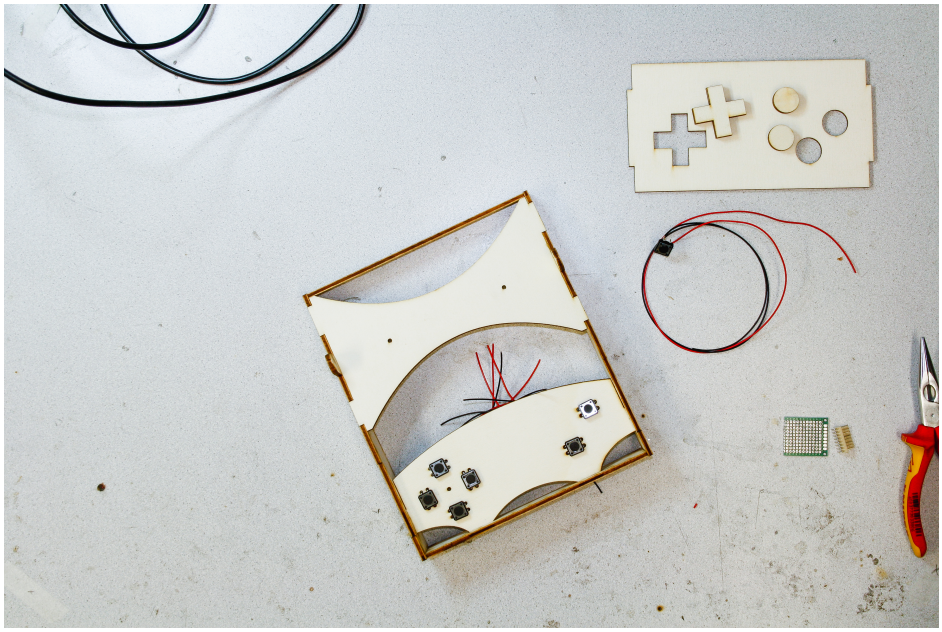


Figure 3.6: Insert the buttons in the footer

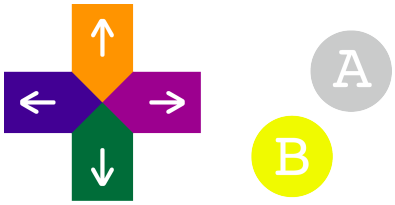


Figure 3.7: The layout of the buttons

Step 3 Now solder the 8-pin-male-connector to the perfboard as shown in figure 3.8 and 3.9. Put the legs of this connector through the holes in row *C* or *D* on the perfboard (or *U/V* on the flip side). Please take a good look at the schematic in Figure 3.9, this is how the buttons will be connected on the perfboard.

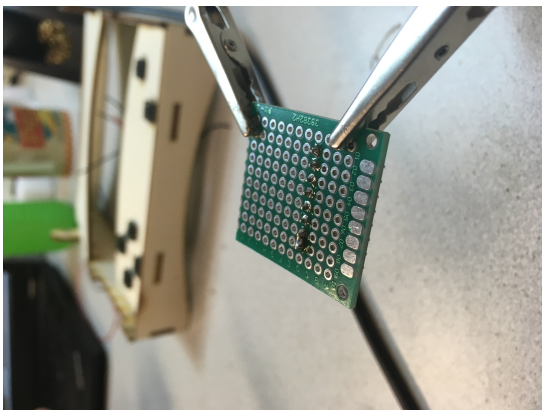


Figure 3.8: 8-pin-male-connector on perfboard

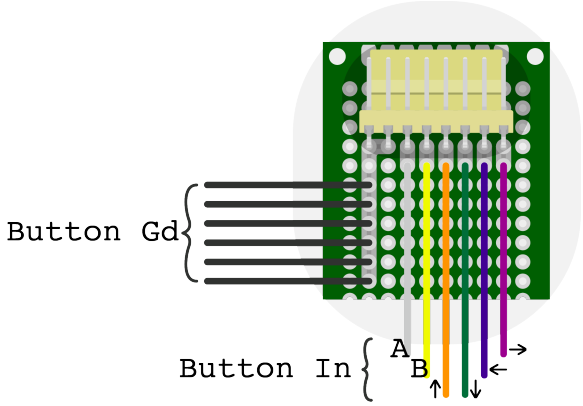


Figure 3.9: Soldering the buttons and connector to the perfboard, see Figure 3.7.

Step 4 Bridge the first and second input (the black and red input), as shown in figure 3.9.

Step 5 Now connect all the wires to the perfboard following the schematic. Your board should now look as in figure 3.10. You could take a look in the 3D model to check your connections. Now your buttons should be ready to use. However, it is wise to check your soldering for short circuits before connecting it to the Raspberry Pi. Read through the troubleshooting in Chapter 5 for more information.

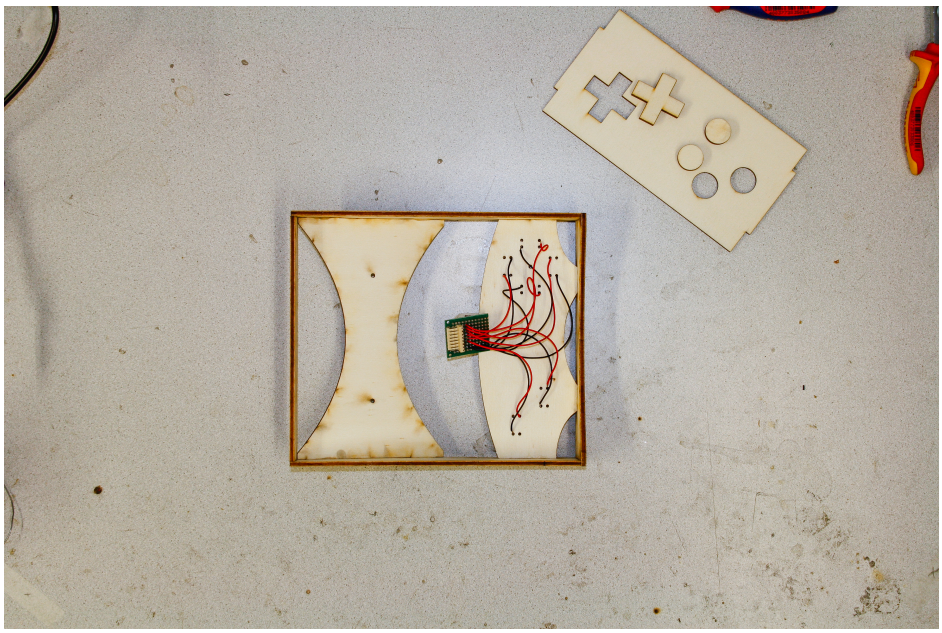


Figure 3.10: The footer all connected

3.2 The Screenstand

Next up is the screen stand. This contains the LED-matrix and your main computer: the Raspberry Pi. We start by working on the Raspberry Pi. To the Pi's GPIO pins, both a shield and buttons will be connected. Because of this, we need to split the headers from the Pi, where a part of them is facing up and a part is facing down. For this part it is very important that you work precisely and that you follow the manual closely; you wouldn't want to break the Raspberry Pi. If you feel uncomfortable, please ask a committee member or another participant for some help or guidance.

3.2.1 Connections to the Raspberry Pi

Step 0 The Raspberry Pi will be connected on the back of the screen, where there is a designated spot for it. During this process, do not connect the Pi to its power. Now first make sure you have the following materials ready.

- Raspberry Pi Zero W [headerless]
- 2x20 header pin
- 2 2x1 header pin (could be provided as one 4x1 header pin)
- 8-pin rainbow Female cable
- LED screen power Cable
- Button with wires soldered (see Section 3.1.1)
- 4 M2.5 8mm bolts with nuts

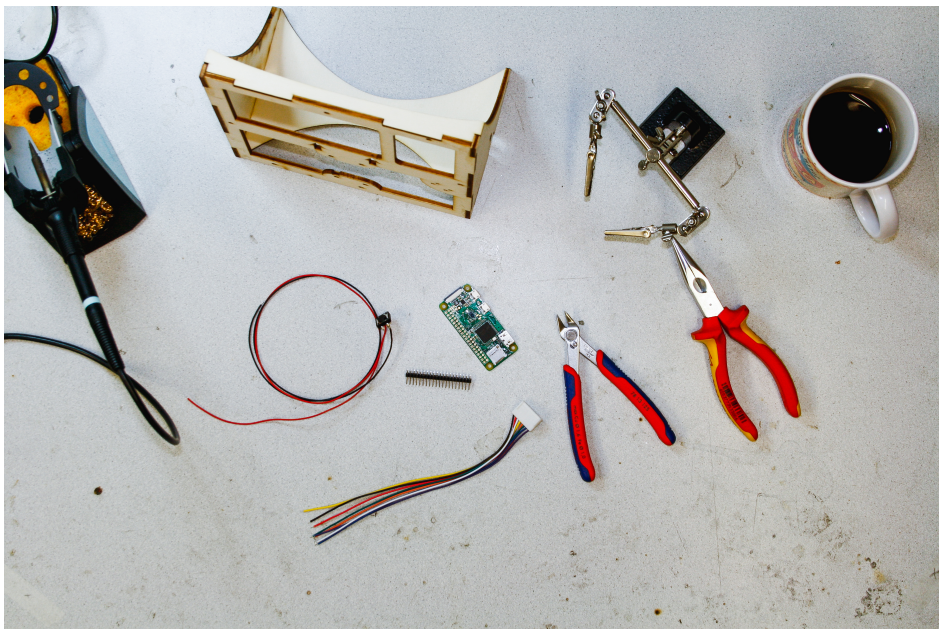


Figure 3.11: Materials for the first part of the screen stand

Step 1 First you need to split your 2x20 header pin in a set of 2x5, 2x12 and 2 times 2x1. See figure 3.12 and Table 3.1. Since this often goes wrong when cutting smaller sets, we provided a few extra 2x1 pin sets.

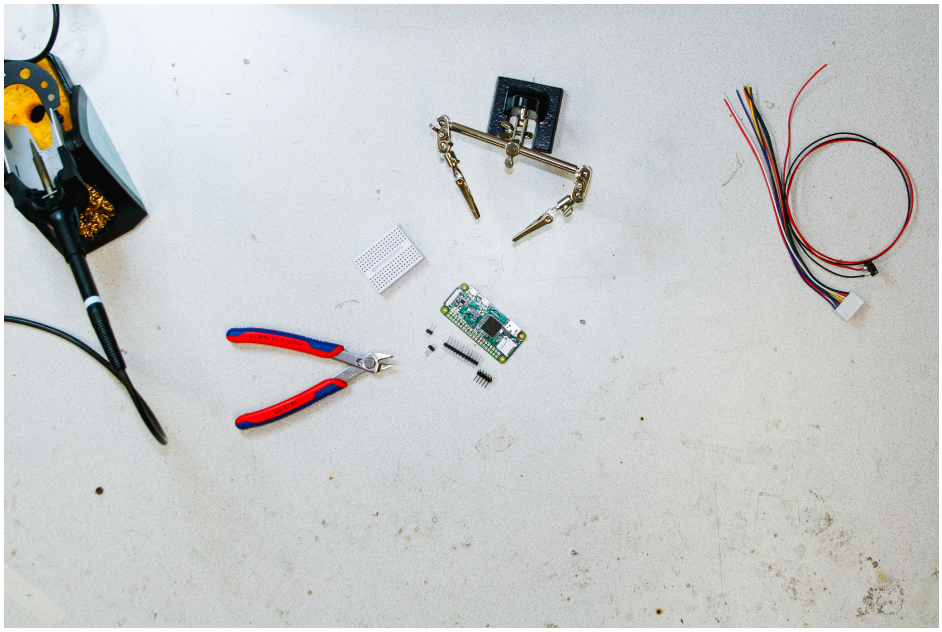


Figure 3.12: Cut the header pin

Step 2 We are going to replicate Figure 3.14. Here we will start with the large pin set of 12x2. Again pay close attention to Figure 3.14, 3.13 and Table 3.1, to where the *upward* facing pins are located. Note that the pin closest to the SD card slot (on the inside) is pin 1 and that pin 40 is at the USB connector end (on the outside). Upwards is defined as pointing the same way as all the connectors (HDMI, USB, SD, etc) on the Pi, and that downwards is thus facing the flat side of the Pi. If you plug the header pins in a breadboard, soldering straight becomes a lot easier, see figure 3.15.

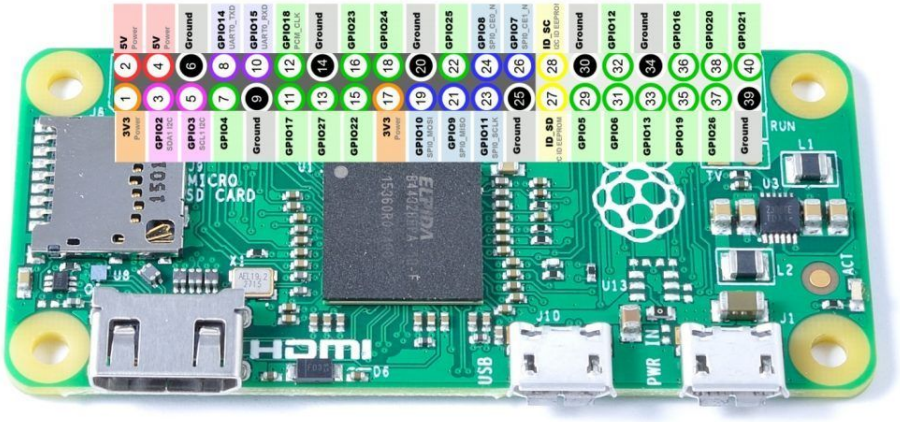


Figure 3.13: Raspberry Pi Pin layout

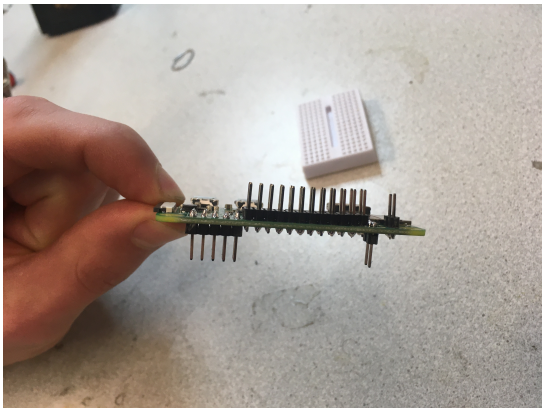


Figure 3.14: Raspberry Pi Soldered

Pi Pins	Headers	Facing	Usage
1 and 2	2x1	Upwards	Providing power to the shield
3 and 4	*Not Connected*		
5 and 6	2x1	Downwards	Power Button on the screenstand
7 to 30	2x12	Upwards	Communication with the shield
31 to 40	2x5	Downwards	Connection with the buttons in the footer

Table 3.1: Direction of header pins on the Raspberry Pi

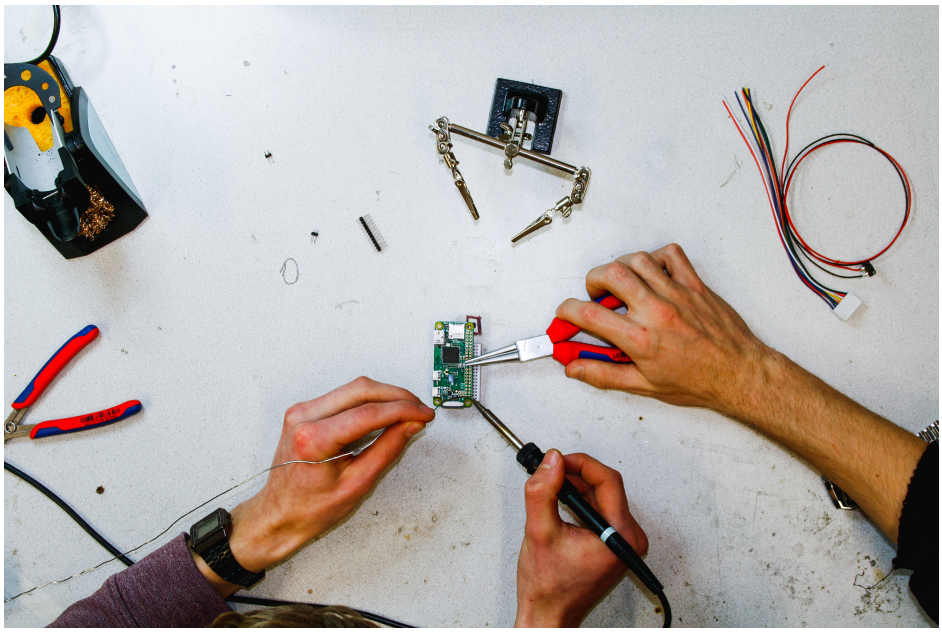


Figure 3.15: Soldering the pins

Step 3 Once you have replicated the Pi of Figure 3.14, you can optionally slightly (±5°) bend the downwards facing pins a little outward, see Figure 3.16.

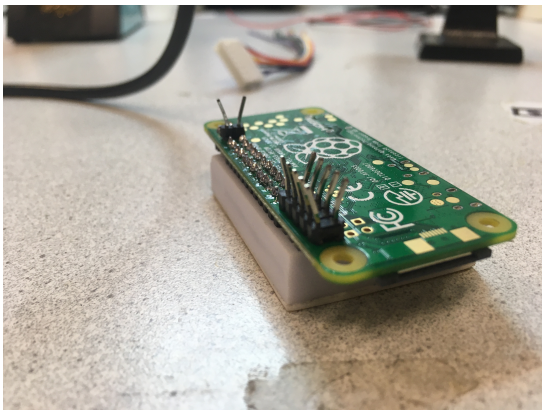


Figure 3.16: Raspberry Pi pins bend

Step 4 Now take the power cable for the led screen. This cable has one connector with two red and two black wires coming out of it. We are only going to use one set of wires, thus strip only one black and one red cable. Optionally pre-solder these wires.

The connection we are going to make now allows the screen to get directly power from the power adapter, instead of through the Raspberry Pi, through the shield and through the ribbon cable addressed later.

The wires will be soldered to the back of the pi, see Figure 3.17 (Note that in the picture **Step 5** is already done, doing the current step first is preferred). In this figure, two pads underneath the outermost USB connector are pre-soldered with some tin. Also pre-solder these pads. Please take your time doing these actions, and do not add too much solder tin. Again, if you feel uncomfortable to do this step, pleas ask a committee member or a fellow participant to help you with this step.

Connect one of the red wire to the inner most pad and one of the black wire to the outer most pad, as illustrated in Figure 3.18. It does not matter which red and which black wire you take. Let the wires face inwards to the Pi as shown in the figure.

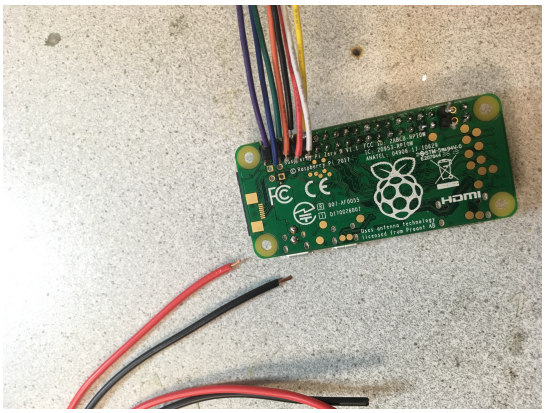


Figure 3.17: Preparing the Pi to connect the screen power Connection

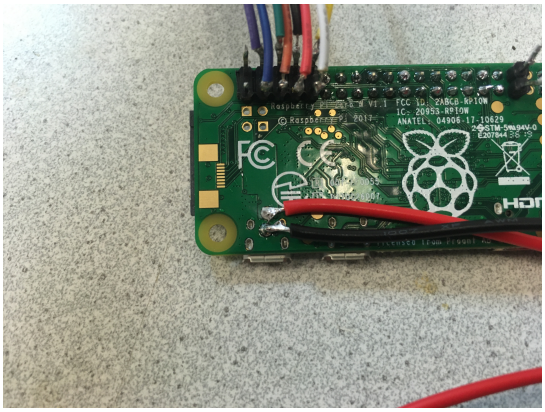


Figure 3.18: Led screen power cable connected to the Pi

Step 5 Now we can solder the 8-pin Female cable to the pins of the Pi, as in figure 3.20. How they are connected exactly, is shown in Table 3.2 and Figure 3.19. As in **Step 2**, solder the wires directly to the pins soldered in the Pi. Make sure you don't short circuit the pins, which happens when you solder pins together.

Button	Wire Colour	Pin	GPIO	WiPi
Ground	Black	Pin 39/34	-	-
Bridge	Red	Pin 33	23	23
A (upper)	White	Pin 31	22	22
B (lower)	Yellow	Pin 32	26	26
Up	Orange	Pin 35	24	24
Down	Green	Pin 36	27	27
Left	Blue	Pin 37	25	25
Right	Purple	Pin 38	28	28
Power/Menu	Red	Pin 5	9	9
Ground	Black	Pin 6	-	-

Table 3.2: Button connections to the Pi; see Figure 3.13

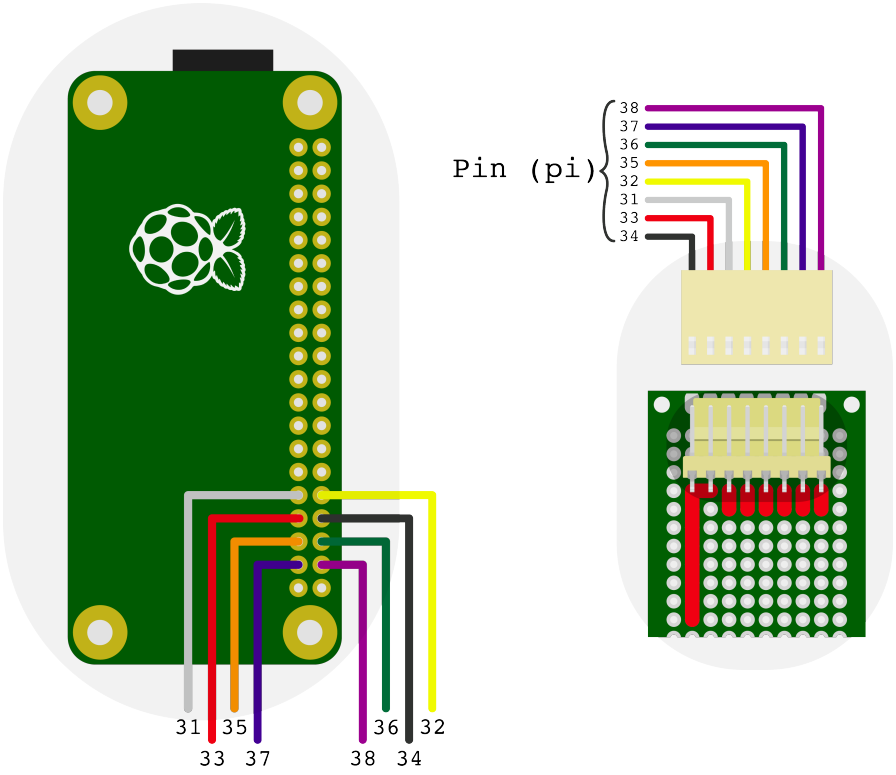


Figure 3.19: Overview of connection of each cable to the correct pin

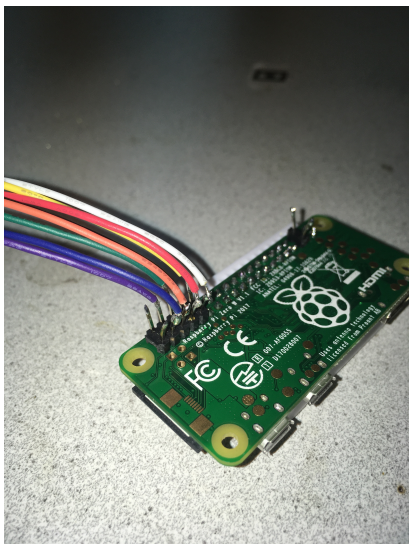


Figure 3.20: Cables on Raspberry Pi Soldered

Step 6 It is now time to mount the Raspberry Pi in the screen stand. Do this with the four M2.5 bolts and nuts, see Figure 3.22 (note that we only connected two bolts).

The last connection to make is the power button. Take the remaining button from Section 3.1.1 and put it through the designated holes next to the pi, see Figure 3.21. Solder the red wire to pin 5 and the black wire to pin 6 (see Table 3.2) just as you did with the other pins. Again, just solder the wires directly to the pins.

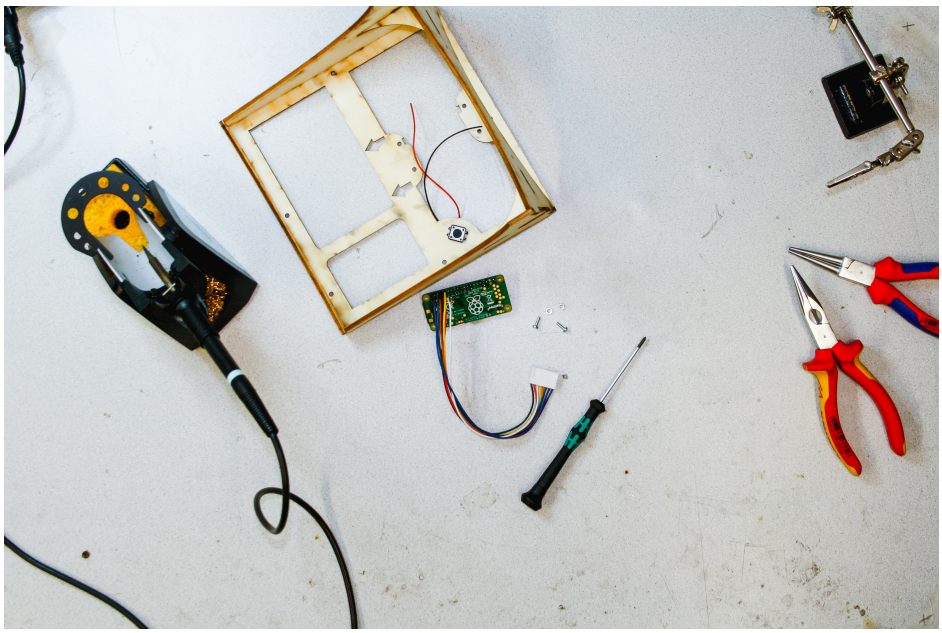


Figure 3.21: Preparing the connection of the last button

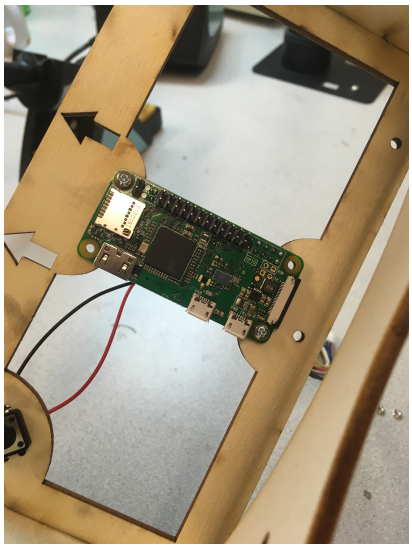


Figure 3.22: The Raspberry Pi mounted in the screenstand

Step 7 Just as in the previous section, it is wise to first check your soldering for short circuits before connecting the Raspberry Pi to the power adapter. Read through the troubleshooting in Chapter 5 for more information.

3.2.2 Connections to the Shield

Now all connections to the Raspberry Pi are made and it is mounted to the screenstand. Next up is connecting the Led screen to the Pi. This is done with the help of a shield which makes this connection very simple. However, a small configuration on the shield should be made to make it work in the way we want it to. Go through the following steps to make this modification.

Step 0 Gather these materials (see Figure 3.23):

- Shield
- Short piece of wire $\pm 1cm$
- Gray Ribbon Cable
- 4 M3 bolts
- Led Matrix

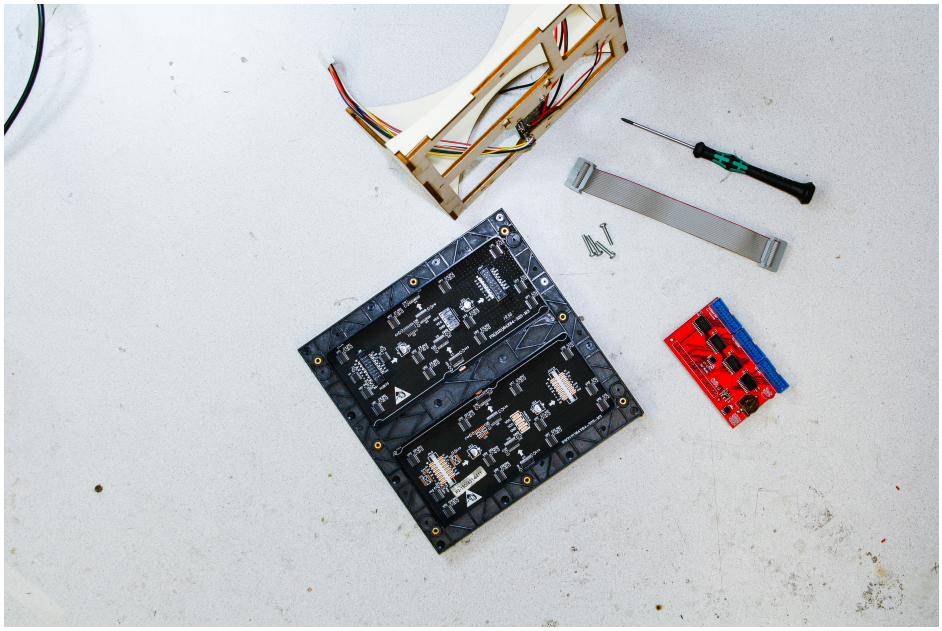


Figure 3.23: Materials to connect the Led Matrix

Step 1 Strip both ends of the short piece of wire. Connect one end to *P8* and the other end to Ǝ (the tilted *E*). See Figure 3.24.

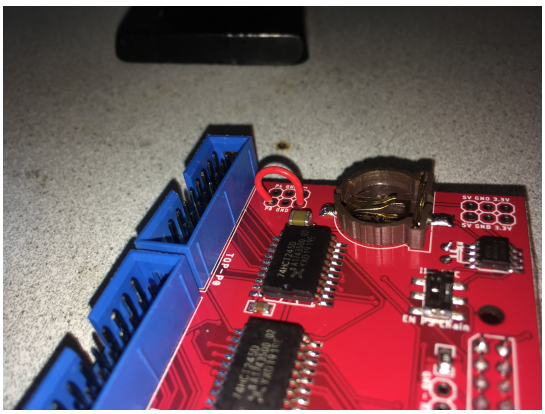


Figure 3.24: Making the jumper connection on the shield

Step 2 With the 4 M3 bolts, mount the Led matrix to the screenstand. Align the arrows on the Led matrix with the arrows cut into the wood.

Step 3 Press the shield on top of the Raspberry Pi, it should cover the whole Pi. Connect the gray ribbon cable to *TOP-P0* and to the *JIN* on the screen. Also connect the power cable of the screen. See how the result should be in Figure 3.25. Note that the rainbow coloured wires are going through the slot in the bottom of the screenstand.

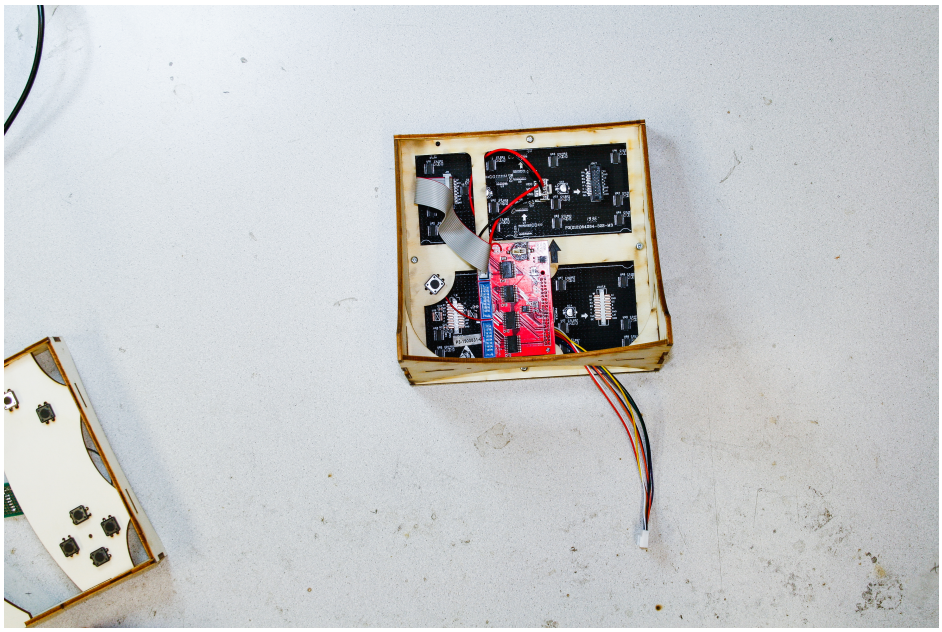


Figure 3.25: How your completed screenstand should look like

3.3 Completing the Machine

There are only a few steps left to finishing the hardware part of this workshop: completing the buttons, connecting the screenstand to the bottom and connecting the power.

Step 0 For these last steps, you need the parts below. See also Figure 3.26.

- Completed base
- Completed screenstand
- M3 bolt with 2 nuts
- Remaining wooden parts
- Power adapter

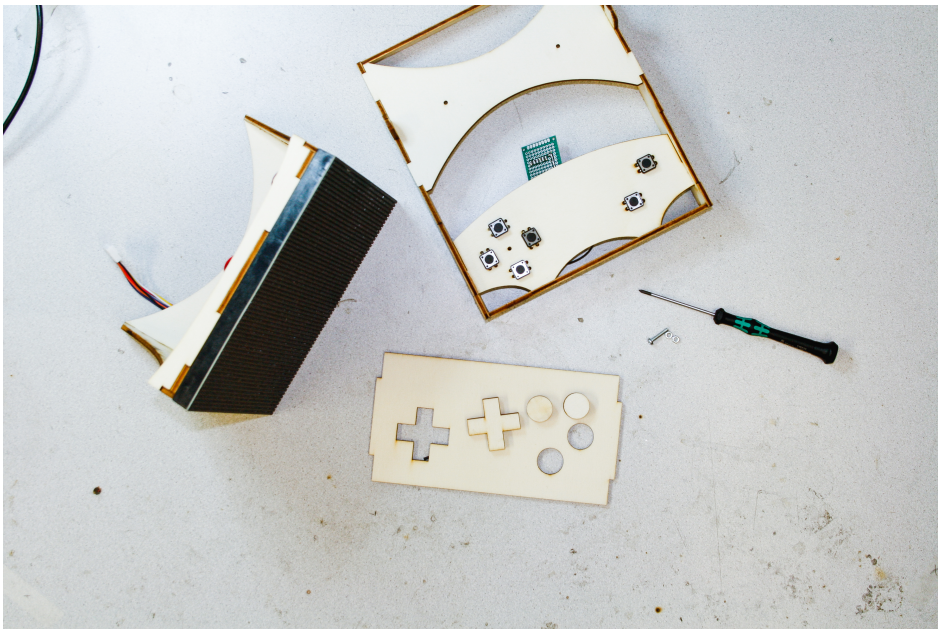


Figure 3.26: Parts needed for the last part

Step 1 First thread one of the nuts onto the M3 bolt, then insert this bolt from underneath into the hole in between the four buttons. Thread the second nut also onto the bolt, locking it's position. You can use this nut to tune the height of the D-pad buttons.

Step 2 Use the connectors to connect the buttons of the base to the Raspberry Pi, as shown in Figure 3.27.

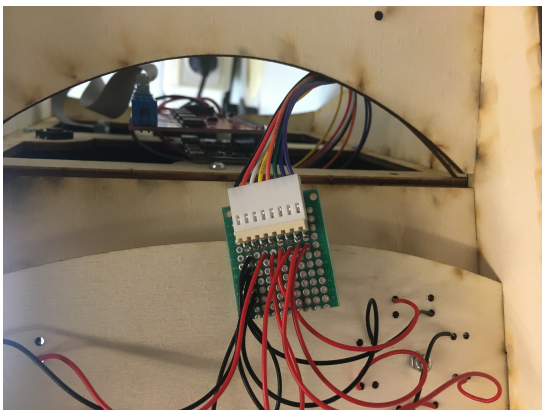


Figure 3.27: Connecting the Raspberry Pi to the buttons

Step 3 Check your connections from each button to the Raspberry Pi, as explained in the troubleshooting in Chapter 5.

Step 4 The screenstand can be pressed on top of the base. Optionally, these two parts can be secured with two M3 bolts and nuts.

Step 5 In the next step we will connect the power to the Pi. However, before you do this, please follow the steps in the *Double Check!* section in Chapter 5.

Step 6 Connect the power adapter to the outer most USB connector (see Figure 3.13).

You are now done with the hardware of this build.

Chapter 4

Software

The last part during this workshop is setting up the software on the Raspberry Pi. The dedicated operating system for the Pi is Raspbian, which is based on Linux. During the workshop, you will be provided a prepared SD-card. This means you can start at Section 4.1.2. If you are following these steps at home, start by installing Rasbian on the Pi.

4.1 Installing Raspbian Linux on the Pi

First you will have to download Raspbian Lite from the official Raspberry Pi website here. When you unzip the downloaded zip file, you will notice a *.img* file inside. This image file needs to be written onto the SD-card, for this we will use balena Etcher which you can download here.

4.1.1 Write an image

Step 1 Make sure the SD-card is connected to your laptop and visible in the devices list in explorer on Windows or Finder on Mac.

Step 2 Open balena Etcher and press the select image button. Navigate to the location where you unzipped the Raspbian Linux download and choose the *.img* file.

Step 3 Make sure the SD-card is the selected device.

Step 4 Press flash and wait while balena Etcher does its magic.

4.1.2 Configuring the SD-card

To make things easier, we will configure the OS to accept SSH and to connect to the wifi-network on a laptop before booting the Pi itself.

Step 1 Open a explorer or finder window and navigate to the *boot* directory on the SD-card.

Step 2 Place a file called *ssh* in this directory. Make sure the file has no file-extension, so no *.txt*, *.docx* or anything. To make sure it has no file-extension you should enable file name extension in the view tab of explorer.

On Mac this can be done in the advanced tab in the finder preferences by checking show all filename extensions. If you that the file accidentally does have a file-extension you should rename the file and remove the file-extension.

Step 3 Next, create a text file called *wpa_supplicant.conf* in the boot folder on the SD-card. Make sure the file-extension is *.conf* and not *.txt*.

Step 4 Add the code from Listing 4.1 to the file replacing `YOUR_SSID` and `YOUR_PASSWORD` with the ssid and password of the preferred wifi network.

Step 5 You can add multiple networks in this file. Every time the Pi boots, the information from this file gets read.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=NL

network={
 ssid="YOUR_SSID"
 psk="YOUR_PASSWORD"
}
```

Listing 4.1: Wifi settings

During the workshop in the SmartXP, you can use the ssid *ProtoEvent* with the key *protosownmatrix*. Make sure your own laptop is also connected to this network. If you added both files to the SD-card, you can (during the workshop) continue to Section 4.1.4.

4.1.3 Booting the Pi

The SD-card is now ready, and it's time to boot the Pi and finalize the setup.

Step 1 Safely remove the SD-card from your laptop and put it in the Pi.

Step 2 Connect the monitor using the HDMI adapter and the keyboard using the usb adapter.

Step 3 Connect the Raspberry Pi power brick to the Pi and wait for it to boot.

Step 4 The raspberry will show a command-line login screen. Login using default username and password of the pi, these are *pi* and *raspberry* respectively.

Step 5 **VERY IMPORTANT** Change the password using the `passwd` command. It will ask you for the current password (*raspberry*) and for the new password.

Step 6 In the future we will control the pi remotely, for this we need the ip address of the pi. This can be retrieved by running the `ip addr show` command. There is a possibility multiple ip addresses are shown, you will need the one after `inet` by `wlan0`, most likely starting with *192.168* or *130.89*. Don't forget to write this ip address down for future reference.

4.1.4 Getting your IP address

During the workshop, you get a prepared SD-card on which already some programs are installed. You can take the SD-card out of your laptop and put it into the Raspberry Pi. Assuming you already build the case and connected all electronics, you can connect the power cable. It may take a while, but a menu should appear on the LED-matrix. On of the options in the menu is ip-address. Select this option and note down the ip-address. You can use this in the next step.

4.1.5 Connecting to the Pi using SSH

To work on the pi from your laptop, you will use SSH. This allows you to interact with the command-line interface of the Pi over wifi.

Step 1 On your laptop, open CMD/Command Prompt on Windows or Terminal on Mac.

Step 2 Run the following command, replacing ip with the IP-address of your Pi: `ssh pi@[ip-addres]`, where you use the ip-addres found before. The `pi` refers to the username on the Raspberry Pi.

Step 3 You will now be asked for your password to login. Use the password you chose during the setup of the pi. If you received an already prepared SD-card during the workshop, use the password *DiyciePassword*. **VERY IMPORTANT** Change the password using the `passwd` command.

Step 4 *Optional* To make sure the pi you are connected to is actually your own Pi you can run the following commands. We will use the built-in ACT LED of the Pi to get some visual feedback from the connected Pi. First we have to set the LED to manual mode, this can be done using the following command: `echo none | sudo tee /sys/class/leds/led0/trigger`. To then turn on the LED you can use this command: `echo 0 | sudo tee /sys/class/leds/led0/brightness`. And to turn it off again, run `echo 0 | sudo tee /sys/class/leds/led0/brightness`. You can check the LED of your Pi to see if its behaviour corresponds to the commands you run. If it does, you are connected to the correct Pi.

All further steps are to be completed over ssh.

4.2 Linux

Just like Windows and MacOS, Linux is an operating system. Windows and MacOS rely heavily on their graphical user interface (GUI), while Linux is often used without a GUI even being installed. All actions can be performed using the command-line, which is similar to comand prompt or cmd on Windows and Terminal on MacOS. All configuration on the Linux OS can be done using just the command-line and a keyboard.

4.2.1 Basic Linux commands

To make using Linux an as easy as possible experience some of the most used commands including their usage and functionality are described below. Square brackets are used to show [parameters] that have to be filled in. When typing in the command-line, you can always use tab to use to built-in auto-complete feature to complete commands or directories, or show multiple possible options for completion.

```
man [command]
```

Listing 4.2: Shows a manual page of the given command including the function

For example, `man ls` shows the usage of the ls command. Which turns out to be a useful command, it lists all directories and files of your current folder. By adding an -a argument, also hidden en dot-files will be show: `ls -a`. Arguments are extra options given to a command in order to slightly change the output of it.

```
ls
```

Listing 4.3: Lists all directories and files in the current directory

Is you have listed your subdirectories, you can go into that folder with the help of the `cd` command. Changing the current working directory to the specified `[directory]`. The specified directory is relative to the current working directory, unless preceded by a forward slash. You can go deeper into multiple folders by entering longer directories separated with forward slashes, e.g. `cd Documents/data`. You can use your tab key to complete your typing.
Inside a folder, you can create a new folder. This is done with the `mkdir` command. The `[directory]` can be a name for a directory to place in the current working directory or a relative or absolute path to the new directory.
Going back, a directory up, is done by typing `cd ..`

```
cd [directory]
```

Listing 4.4: Change the current working directory

```
mkdir [directory_name]
```

Listing 4.5: Create a new folder

The last important en use full task in Linux is creating and editing files. For this we can utilize a program called Nano. Nano is a command-line text-editor. It is just like a normal text editor but you can only use the keyboard to navigate the cursor. You can open a file with `nano [file_to_edit]`. If the file does not exist yet, it will get created. To save the file you are working in you can press CTRL+O followed by ENTER. To exit the editor you can press CTRL+X, if the file has not been saved, it will prompt for you to save the file.

```
nano [file_name]
```

Listing 4.6: Edit or create a file

4.3 Installing the LED-matrix software

This workshop includes pre-written code to make interfacing with the LED-matrix easier. This code is hosted on GitHub. During the workshop, you can continue to **Step 9** in Section 4.3.1. The next steps will guide through downloading the code.

Step 1 To download code from GitHub, we first need to install git. This can be done using the package manager included with the Raspbian OS. You can install git using the following command: `sudo apt-get install git -y`. You may be required to first run the commands `sudo apt-get update` and `sudo apt-get upgrade`.

Step 2 Next we have to download (or clone, as it is called when using git) the code. This can be done by running `git clone https://github.com/DIYCie/LED-matrix`. This will download the code into a folder called *LED-matrix* placed in your current directory. Remember that you can navigate to another folder using `cd` if you want it to be placed in a different directory.

4.3.1 Preparing the software

Right now you have downloaded the code to interface with the LED-matrix, but you can't run it yet. You will have to download the needed libraries.

Step 1 Use `cd` to navigate to the *LED-matrix* folder. Run `cd LED-matrix` if the folder is in your current directory. If it is located elsewhere, navigate to its actual relative or absolute path instead.

Step 2 To be able to easily download all needed libraries and run the software, you will need some essential applications. These can be installed by running `sudo apt-get install nodejs -y`.

Step 3 To download the libraries, simply run `npm install`. This will download all the libraries needed to run the software.

Step 4 Disable the sound module by running `sudo nano /boot/config.txt` and change the line `dtparam=audio=on` to `dtparam=audio=off`.

Step 5 Also add `dtoverlay=gpio-no-irq` to */boot/config.txt* to disable GPIO interrupts. Save the file with Ctrl-X, followed by pressing y and enter.

Step 6 Your user will need to be a member of the gpio group. Run `sudo usermod -a -G gpio root`.

Step 7 You also need to configure udev with the following commands:

```
$ sudo -i
$ cat >/etc/udev/rules.d/20-gpiomem.rules <<EOF
SUBSYSTEM=="bcm2835-gpiomem", KERNEL=="gpiomem", GROUP="gpio",
     MODE="0660"
EOF
$ exit
```

Step 8 Reboot the Pi with `reboot`. On your laptop, you need to reconnect to your pi with `ssh pi@[ip-addres]`.

Step 9 Test if everything went correctly by running the software. Start it by running `sudo npm start`, this might ask you for the password of the pi you set up earlier. The LED-matrix should now be activated and you should be able to run the example programs on it. If this is not the case, read the software part in Chapter 4 or ask one of the DIYCie members.

Step 10 To return to the menu, press the button on the back of the matrix, next to the Raspberry Pi.

Step 11 The last step is to set the software to automatically start when the Pi is booted. This can be easily done by running `sudo crontab -e`, this will open op a file in the text editor. Add `@reboot sudo npm start --prefix /home/pi/LED-matrix/` to the end of the file (if applicable change `/home/pi/LED-matrix/` to location of your cloned repository).

4.3.2 Writing your own programs

Now the fun part starts. The code supplied to you is mostly a framework that allows you to write and run your own programs on the LED-matrix. You can write your own code in JavaScript. In the *apps* folder, an example program is included. As you can see there, it is structured similarly to processing code. There is a setup and draw function. Adding your own program to the software is quite easy:

Step 1 Copy the template program from *templates/AppTemplate.js* program and save it as *[ProgramName].js* in the *apps* folder and change the class name, name of the app and description to fit your new app.

Step 2 Add code in the *setup()* and *draw()* functions. The *this.matrix* object is used to interface with LED-matrix, a list of all its functions and usage can be found here. A few of the most important ones are:

- *fgColor(color)* sets the color for all the drawing functions run after this. color can be a hex code like *0xC0FF33* or an RGB object like {*r: 0, g: 254, b: 0*}.
- *bgColor(color)* is used to set the background color for drawing functions run after this. The color can have the same formats as in the *fgColor* function.
- *brightness(number)* sets the brightness for all drawing functions run after this. The brightness value can be anywhere between 0 and 255.
- *drawCircle(x, y, radius* draws a circle at the position (x,y) with a specific radius.
- *drawLine(x0, y0, x1, y1)* draws a line from point (x0,y0) to (x1,y1).
- *drawRect(x, y, width, height)* draws a rectangle at (x,y) width a specific width and height.
- *setPixel(x, y)* draws one single pixel at (x,y).
- *drawText(text, x, y)* draws a given string of text at (x,y).

These functions can be used like this: *this.matrix.fgColor({r: 0, b: 255, g: 125})*

Important: Never use *this.matrix.sync()* as this will break the application framework.

Step 3 Start the application by reloading the apps using either the reload menu item or by exiting the software and running `sudo npm start` and launch your own program from the list on the LED-matrix.

Chapter 5

Troubleshooting

5.1 FAQ

The workshop is over, how do I contact the committee?

You can find our contact information on the Proto website. Here you can also find the committee members who could be willing to help you.

I received a broken part in my kit, what now?

We did our best to check all the parts before putting them in your kit. However, it can occur that you received a broken part. Please let one of the committee members know, they will help you getting a new part. However, we are not responsible for damage to your parts due to soldering, transportation, etc.

How do I connect my device to the my own WiFi?

To connect your Raspberry Pi to your home network, or to eduroam in case you are living on campus, please follow the steps described in Section 5.3.1.

I was unable to attend the workshop, but want to build this device at home. Where can I buy all the parts?

A list of ordering links will be provided on the wiki/GitHub/elsewhere.

What programming language is used for the framework used in this build?

The framework is written in JavaScript running in Node.js, based on code written by Alex Eden. The matrix however is in the background controlled bu C++ code.

Look, I made this, where can I share it?

Nice, we find it really nice that you have made something yourself to add to the device. This could be a new program but also a new design for a casing. How

to share this and where to find this is described in the next chapter, Chapter ??.

I accidental connected two pins, how can I fix this?

In the Section 5.2.2 below, some things are said about fixing mistakes. Essentially, it boils down to two simple steps. First clean your soldering tip with a cleaning sponge. Secondly, let your soldering tip go over your unintended connection or in between the two shorted pins, and let some solder stick to your tip. Repeat these two actions, until you fixed the mistake.

5.2 Hardware

5.2.1 Double Check!

Before you connect power to your Raspberry Pi, we suggest to check the things below to make sure nothing breaks or blows up. Although this last event is unlikely to happen, the possibility is still there.

- Check if you have put the right headers in the right place in the Raspberry Pi. Note the visible pins in Figure 3.22 in relation to the arrows in the wood. Also note the empty holes in the second row.
- Check all your soldering as described in Section 5.2.3.
- Make sure all connections are as shown in Figure 3.2 and as described in Table 3.2.
- Double check if the two wires soldered to your Pi, are as shown in Figure 3.18. Also make sure you have connected them to the correct USB input (the outer most). This will also be the USB input to which you connect the power.
- Check the orientation of the shield on your Raspberry Pi. Both the shield and and the Pi should be directed in the same direction with respect to the pins. Or stated differently, the shield should fully cover the Pi.
- Check the jumper you made on the shield (see Section 3.2.2), also look for short circuits.
- Check the connections form the shield to the Led matrix as described in Section 3.2.2, both the red/black power cable and the grey ribbon cable.
- Make sure your SD-card is inserted into the Pi.
- Check if the connector on the perfboard is connected to the rainbow cable. Note that this cable has a correct orientation.
- Look through the arrows, cut out in the back of the screenstand, make sure you see two corresponding black arrows on the back of the Led matrix.

5.2.2 How to solder

If you haven't soldered before or you need a recap, we recommend this instructable tutorial. We are soldering onto the Raspberry Pi circuit board, described in Chapter 3. In the instructable guide, this is explained in step 4, take some special notice to that step. Also check step 8 for fixing mistakes. If you still feel uncomfortable soldering or have any questions, don't be afraid to ask your neighbour or a committee member.

5.2.3 Checking your connections

After you have soldered wires or pins to buttons, a perfbord or to the Raspberry Pi, we recommend crosschecking your connections with a multimeter.

Step 0 For these steps you only need a multimeter.

Step 1 Put the multimeter in continuity testing mode. This is indicated with a speaker symbol ( ) and often combined with the diode test ( ). How to select this mode differs per multimeter, but it should be quite straight forward.

Step 2 In the unlikely case that your multimeter does not have this continuity mode, you can also put it in direct resistance measuring mode (indicated with the Ω symbol). Instead of getting audible feedback, you should watch the screen for resistances of infinity, which means the pins are not connected.

Step 3 Put your black probe in the COM input, and your red probe in the input with the same symbol as you have selected above.

Step 4 Put one end of the probe against one pin and the other to the one next to it. The moment continuity is discovered within the two points, the meter will automatically emit an audible tone. This is an excellent and great way for finding short circuits, or open circuits.

Step 5 Cross check every solder joint. As long as you do not hear a beep, you are good. In the case you do hear a beep, check first if this maybe should be the case. For example, all grounds on a Pi are connected.

Step 6 If pins, not intended to be, turn out to be interconnected, check step 8 in Section 5.2.2 how to fix such mistakes.

5.3 Software

5.3.1 How do I connect my Pi to Eduroam?

Connecting the Pi to eduroam is not as simple as connecting it to a normal wifi network. Instead of just placing a text file on the SD-card, you will have to configure eduroam on the Pi itself using some commands. The networking service of the Pi has to be stopped to configure it to connect to eduroam, so you won't be able to complete these steps over SSH.

Step 1 Connect a monitor and keyboard to the Pi using a HDMI-miniHDMI adapter and a miniUSB-USB adapter.

Step 2 Login using the username *pi* and the password you set earlier.

Step 3 Stop the networking service by running `sudo service networking stop`.

Step 4 Run: `sudo nano /etc/wpa_supplocant/wpa_supplicant.conf`. This opens a command-line text editor in which you can navigate using the arrow keys.

Step 5 Remove the whole *network=...* block and copy the text from Listing 5.1 and fill in YOUR_SNUMBER and YOUR_PASSWORD.

Step 6 To run the new configuration, run: `sudo wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf -B`. Your raspberry pi will now connect to the eduroam network.

Step 7 The command in **Step 6** has to be run every time the raspberry pi boots. To make it run automatically, add it to the bottom of the */.bashrc* file. To open the file in a text-editor, run: `sudo nano ~/.bashrc` and add the command to the bottom. Save the file using CTRL+O and then ENTER and close the text-editor using CTRL+X. Now the raspberry pi will automatically connect to eduroam at every boot.

```
network={
        ssid="eduroam"
        scan_ssid=1
        key_mgmt=WPA-EAP
        eap=PEAP
        identity="YOUR_SNUMBER@utwente.nl"
        password="YOUR_PASSWORD"
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
}
```

Listing 5.1: Wifi settings for Eduroam

5.3.2 System Setup

Before running `npm start`, you may need to be required to change some settings.

Step 1 Disable sound module. Run `sudo nano /boot/config.txt` and change the line `dtparam=audio=on` to `dtparam=audio=off`.

Step 2 Also add `dtoverlay=gpio-no-irq` to */boot/config.txt* to disable GPIO interrupts.

Step 3 Your user will need to be a member of the gpio group. Run `\$ sudo usermod -a -G gpio root`. You also need to configure udev with the following commands:

```
$ sudo -i
$ cat >/etc/udev/rules.d/20-gpiomem.rules <<EOF
SUBSYSTEM=="bcm2835-gpiomem", KERNEL=="gpiomem", GROUP="gpio",
     MODE="0660"
EOF
$ exit
```

Step 4 Reboot the Pi with `reboot` and reconnect to the pi with ssh.