



## ASSIGNMENT OF BACHELOR'S THESIS

**Title:** USB Flash Drive Writer  
**Student:** David Labský  
**Supervisor:** Ing. Miroslav Hroněk  
**Study Programme:** Informatics  
**Study Branch:** Computer engineering  
**Department:** Department of Digital Design  
**Validity:** Until the end of summer semester 2017/18

### Instructions

Design and implement an open source hardware box for writing a provided bootable disk image of a Linux distribution such as Fedora to USB flash drives. The box should have a button to start the write operation and an indicator of the progress.

- Study existing implementations.
- Consider and compare different types of hardware for use in the project.
- Write full instructions for construction of the box from scratch.
- Construct a functional physical prototype.

When making decisions, consider the following criteria for use in the Fedora Project:

- Ease of build: Even users without significant previous knowledge about used hardware should be able to build the device.
- Price and availability of the components on various continents.
- Extensibility.

### References

Will be provided by the supervisor.

L.S.

doc. Ing. Hana Kubátová, CSc.  
Head of Department

prof. Ing. Pavel Tvrđík, CSc.  
Dean

Prague February 13, 2017



CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF DIGITAL DESIGN



Bachelor's thesis

# USB Flash Drive Writer

*David Labský*

Supervisor: Ing. Miroslav Hrončok

16th of May, 2017



---

# Acknowledgements

A sincere thank you to my supervisor Miroslav Hrončok for his ideas, advice, and continued mentoring. Thank you to Petr Viktorin for his enthusiasm and pieces of hardware. Sorry I didn't reach out more! Many thanks to Marek Žehra for incredibly professional photos on short notice.

Thank you, Senti, for proofreading as well as keeping me virtual company during the final stretch. Thanks to Jiří Šebele for lending a hand, somehow we form a good team. A thank you goes to my parents, even if they are unable to read the work proper.

Thank you to the communities surrounding open hardware for their worthwhile endeavor and always being willing to help.

Lastly, I would like to extend my thanks to birds.



---

## Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on 16th of May, 2017

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2017 David Labský. All rights reserved.

*This thesis is a school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).*

### **Citation of this thesis**

LABSKÝ, David. *USB Flash Drive Writer*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2017. Available also from WWW: <https://github.com/Sanqui/fedorator-thesis>.



---

# Abstrakt

V této práci je od myšlenky k prototypu dovedeno zařízení zvané Fedorator, které slouží k zápisu operačního systému Fedora na USB flash disky. Je zde proveden detailní rozbor hardwaru včetně přehledu USB protokolu.

**Klíčová slova** USB flash disk, USB duplikační zařízení, Fedora, Raspberry Pi, open hardware



---

# Abstract

In this thesis, a device called Fedorator, capable of writing the Fedora operating system to USB flash drives, is brought from thought to a prototype. Detailed analysis over the hardware is performed, including an overview of the USB protocol.

**Keywords** USB flash drive, USB duplicator, Fedora, Raspberry Pi, open hardware



---

# Contents

<b>Introduction</b>	<b>21</b>
<b>1 Research and Analysis</b>	<b>23</b>
1.1 How Fedora Spreads Publicity . . . . .	23
1.2 Existing solutions . . . . .	24
1.2.1 Preloaded USB flash drives . . . . .	24
1.2.2 USB Duplicators . . . . .	25
1.2.3 Existing software . . . . .	27
1.2.4 Dedicated Open Source Hardware . . . . .	28
1.3 Rationale for a New Solution . . . . .	28
<b>2 Specification and Goals</b>	<b>31</b>
2.1 Function . . . . .	31
2.1.1 User . . . . .	32
2.1.2 Operator . . . . .	32
2.2 Looks . . . . .	32
2.3 Interface . . . . .	33
2.4 Conclusion . . . . .	35
<b>3 Analysis of Possible Solutions</b>	<b>37</b>
3.1 Hardware Options . . . . .	38
3.1.1 Raspberry Pi . . . . .	38
3.1.2 Arduino . . . . .	40

3.1.3	ESP8266 . . . . .	41
3.1.4	Teensy . . . . .	42
3.1.5	FPGA . . . . .	43
3.2	Display Options . . . . .	44
3.2.1	Character-based LCDs . . . . .	44
3.2.2	OLED . . . . .	44
3.2.3	TFT . . . . .	45
3.3	Other Component Options . . . . .	45
3.4	Casing Options . . . . .	45
3.4.1	Wood . . . . .	45
3.4.2	Plastic with Mold . . . . .	46
3.4.3	3D Printed . . . . .	46
3.5	Roundup . . . . .	47
3.6	Concepts . . . . .	48
3.6.1	Concept A . . . . .	49
3.6.2	Concept B . . . . .	50
3.6.3	Concept C . . . . .	51
3.7	Decision . . . . .	52
<b>4</b>	<b>Raspberry Pi-based Solution</b>	<b>53</b>
4.1	Specification . . . . .	53
4.2	Software . . . . .	53
4.2.1	Behavior . . . . .	54
4.3	Technical Implementation . . . . .	57
4.4	Case . . . . .	57
4.5	Construction of a Prototype . . . . .	59
4.6	Instructions . . . . .	61
4.7	Speed Measurements . . . . .	61
4.8	Conclusion . . . . .	65
<b>5</b>	<b>Interlude: USB Mass Storage</b>	<b>67</b>
5.1	Introduction to USB . . . . .	67
5.2	USB Specification . . . . .	68
5.2.1	USB Host . . . . .	68
5.2.2	USB Device . . . . .	69
5.2.3	USB Hardware . . . . .	69

5.2.4	USB Protocol . . . . .	69
5.3	USB Mass Storage Device Class . . . . .	70
<b>6</b>	<b>Microcontroller-based solution</b>	<b>71</b>
	<b>Conclusion</b>	<b>73</b>
	Future Work . . . . .	74
	<b>Bibliography</b>	<b>75</b>
<b>A</b>	<b>List of Acronyms</b>	<b>85</b>
<b>B</b>	<b>Supplemental Material</b>	<b>87</b>





---

## List of Tables

1.1	Examples of USB duplicators available on the market . . . . .	25
4.1	Fedorator speed measurements . . . . .	64



---

## List of Figures

1.1	Fedora logo and wordmark [33]	23
1.2	VINPOWER Black 1 to 11 USBShark [60]	26
1.3	Altarec USB Copy Cruiser Mini Duplicator [7]	26
1.4	Fedora Media Writer running on Linux [21]	27
2.1	Solid Machined Metal Knob offered by Adafruit [2]	34
3.1	Raspberry Pi 3 Model B [65]	38
3.2	Arduino Uno R3 [79]	41
3.3	Arduino Nano [56]	42
3.4	Teensy 3.6 [80]	43
3.5	Common OLED display available on Adafruit [1]	44
3.6	LulzBot Mini 3D printer [4]	46
3.7	Concept A	49
3.8	Concept B	50
3.9	Concept C	51
4.1	Fedorator: the standby screen	54
4.2	Fedorator: picking a release from the list	55
4.3	Fedorator: taking a look at the Fedora Workstation release	56
4.4	Fedorator: Fedora LXDE Destkop was sucessfully written to the flash drive	56
4.5	A screenshot of the OpenSCAD software with the case open. Both the source code and the rendered model can be seen	58

## LIST OF FIGURES

---

4.6	A render of the constructed case, courtesy of OpenSCAD . . . . .	59
4.7	A test print with a Raspberry Pi seated inside . . . . .	60
4.8	The case print complete with supports . . . . .	61
4.9	The Fedorator prototype with two USB flash drives plugged in . . .	62
4.10	The Fedorator prototype done flashing the Fedora Workstation system . . . . .	63
4.11	The various flash drives tested . . . . .	64
6.1	Teensy detecting a USB mass storage device . . . . .	72

---

# Introduction

In today's world, there are many free and open source operating systems based on the Linux kernel, called Linux distributions [90]. Despite Linux's relatively small market share (between 2.4% and 4.6%, to date [91]), it is being used by millions of people daily and has significant traction among developers [81] and technically minded people in general. The companies and communities behind Linux distributions are, of course, trying to raise the usage of their respective operating systems. One way to achieve that is through marketing stands at various technical conferences and trade shows. Besides persuasion and marketing items such as stickers, a great way to get people to try a new operating system is to simply hand it over to them.

The community behind the free Linux distribution called **Fedora** [32] has been giving out live DVDs which let people try and install Fedora at home. However, recently a decision was made and Fedora 25 is the last release which will be distributed on DVDs in the European region<sup>†</sup>. This is because as time advances, less computers actually have a DVD drive. In the past few years, laptops have been getting thinner and come without DVD drives [40]. Most computers nowadays are capable of booting from an USB flash drive just as well as from disk drives [30].

The goal of this thesis is to design and construct a device, dubbed the "Fedorator", which will enable writing Fedora to flash drives as a self-service. The primary use of the Fedorator would be on marketing stands, where it would

---

<sup>†</sup>The supervisor of my thesis has provided me with this information.

## INTRODUCTION

---

be available for visitors to use with any USB flash drive. This device should be relatively simple to construct using off-the-shelf components, even for a person who haven't seen it in person before.

In this thesis, I will firstly go on to consider the problem space of delivering bootable USB flash drives and look for existing solutions. I will then justify designing a new device to serve the purpose.

---

## Research and Analysis

In order to determine the best way to build the Fedorator device, as well as gather inspiration, I have done research into the problem space.

### 1.1 How Fedora Spreads Publicity



**Figure 1.1:** Fedora logo and wordmark [33]

The Fedora project has a worldwide Ambassador program. **Ambassadors** are people who act as representatives of Fedora. They have a good understanding of Fedora's principles and aim to spread the message to the public [31]. One of the tasks assigned to Fedora Ambassadors is organizing Fedora participation at events.

In the Czech Republic, Fedora holds stands on events such as DevConf [63], PyCon CZ [70], or Linux Days [53]. An Ambassador's task at the stand might

be to set it up, bring out the merchandise, and then speak with visitors, who may be familiar with Fedora and have questions, or have never heard about it before.

Fedora Ambassadors are not necessarily people of technical expertise. For example, a Fedora Ambassador might also be involved in the Fedora project as a translator.

## 1.2 Existing solutions

Several existing options for distributing USB flash drives were considered. These options may also offer inspiration for the Fedorator.

### 1.2.1 Preloaded USB flash drives

Many services offering bulk preloaded USB flash drives may be found on the Internet. With a large focus on marketing, in general they offer to preload materials such as PowerPoint presentations and PDF brochures [36].

Literally, they instruct to send the files by email or upload them. Hence, it stands a question whether they would be capable of producing bootable USB flash drives.

I went looking to order with the following parameters.

- Capacity of 2 GB
- 2 GB of preloaded data
- USB flash drive must be bootable
- No branding necessary
- 500 units

Most services, rather than show a cost upfront, appear to ask the user to fill in a form and get a personalized quote [54] [35] [69]. I have asked for a quote from two services and received no response. The company MemoTrek has an online quote calculator and offers for my specified parameters the price of 4.19 USD



(about 100 CZK) per flash drive [55]. However, they only offer data duplication of up to 1 GB, while most bootable images go over that volume.

For a bit of perspective, the price for 500 replicated DVDs come down to 0.88 USD (about 22 CZK) for one DVD [24]. We can see USB flash drives are over four times as expensive.

### 1.2.2 USB Duplicators

Rather than using a service and ordering flash drives already preloaded, we can acquire dedicated hardware and do the task ourselves. There are several products available on the market, called “USB Duplicators”, which claim to provide this functionality. Some examples are provided, with the number of ports and to-date cost in USD and CZK (only informative).

**Table 1.1:** Examples of USB duplicators available on the market

Name	ports	USD	CZK	
Altarec USB Copy Tower SA Duplicator	118	\$15 749	388 225 CZK	[8]
TEAC 1 to 15 USB Drive Duplicator	15	\$1 206	29 728 CZK	[59]
VINPOWER Black 1 to 11 USBShark USB Flash Copy Tower Duplicator	11	\$890	21 668 CZK	[60]
Altarec USB Copy Cruiser Mini Duplicator†	10	\$315	7 764 CZK	[7]
Intelligent 9 Series (UB910G) - GOLD Factory series 1 - 9 Target USB Flash Memory/Pen Drive/External USB Hard Drive Duplicator	9	\$1 260	31 056 CZK	[58]
USB Stick Copystation†	7	Unk.	Unk.	[3]
EZ Dupe 2	3	\$81	1 997 CZK	[10]

In general, the products can be split into two categories: devices that work stand-alone and devices that require a computer to operate. Devices from the latter category are marked in the table with a dagger (†).

Besides their price, the trouble with these products is that they are proprietary. The hardware cannot be built from scratch and the software cannot be inspected or improved. Support for creating bootable USB flash drives is not in most



**Figure 1.2:** VINPOWER Black 1 to 11 USBShark [60]



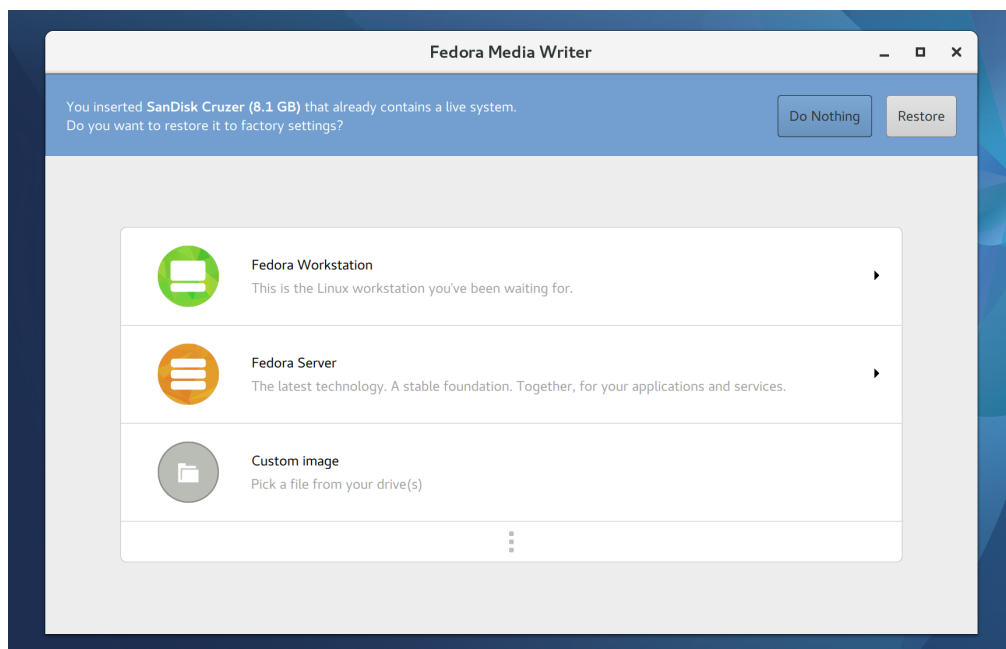
**Figure 1.3:** Aleratec USB Copy Cruiser Mini Duplicator [7]

cases guaranteed, and indeed people have been burned before [51]. The devices which are not stand-alone require a computer with specific software and drivers to operate. Linux support is not claimed, which can usually be interpreted as unavailable, leading to a sad and undesired irony of requiring a computer running an OS that is not Linux in order to produce and distribute USB flash drives containing Linux.

### 1.2.3 Existing software

A number of software exists to serve the purpose of creating live USB flash drives. The Fedora Project wiki describes several of them and primarily recommends the use of Fedora Media Writer [30].

#### 1.2.3.1 Fedora Media Writer



**Figure 1.4:** Fedora Media Writer running on Linux [21]

**Fedora Media Writer** is a desktop application developed by Martin Bříza and Jiří Eischmann. It covering the entire process from picking a Fedora release and downloading it, to copying to an USB flash drive. It was overhauled in 2016 to provide a new user friendly interface [26].

Fedora Media Writer is written in C++ using the Qt framework and supports Linux, Windows, and macOS [21]. It's a solid and well tested application, however, clearly suited for desktop usage rather than embedded. I have decided to create software for the Fedorator from scratch rather than attempt to add a new user interface to Fedora Media Writer. However, the user interface of Fedora Media Writer did prove an inspiration.

### 1.2.3.2 Other Specialized Software

Another piece of software called UNetbootin allows for writing a wide variety of live distributions onto flash drives. Its user interface is not as sleek as Fedora Media Writer's, lacking in logos and screenshots. The Fedora images installed by UNetBootin are known to exhibit issues [30] [77].

### 1.2.4 Dedicated Open Source Hardware

I have attempted a search to discover if somebody has created an open source and open hardware USB copier before. I found many small scripts for copying USB flash drives [64] [29] [20]. None had any sort of hardware design.

The blog post titled "A Simple USB Thumb Drive Duplicator on the Cheap" by Open Security Research [51] offers advice on how to use an USB hub and Linux commands to duplicate to several flash drives at once.

## 1.3 Rationale for a New Solution

There are several advantages to designing a solution from scratch.

- It will be a free and open solution under a permissive license, allowing others to contribute improvements.
- It allows us to create a customized interface with image options.
- It means we can design customized and attractive branding.
- The total cost may be lower.

There are also several disadvantages.

- A person needs to invest time into building the device.
- It may not be as compatible, or “battle tested” as commercial solutions<sup>†</sup>.
- We may run into other unexpected issues.

Despite available options, as we’ve seen, they do not offer all of the advantages of possible a new solution. I will go ahead and design a device called the Fedorator.

---

<sup>†</sup>Although as we have seen, commercial solutions are not always entirely compatible either.



---

## Specification and Goals

### 2.1 Function

The Fedorator should be a device that allows the layperson to load Fedora onto a generic USB flash drive. The flash drive may be provided by the person, or it can be given away by people who watch over the stand.

A Fedorator should allow the person to choose from several different images to load onto the USB drive. This choice should be provided by an intuitive menu highlighting the current choice and allowing the user to scroll. After the desired image is selected, a progress bar will be shown and the Fedorator will begin copying the image onto the flash drive. This image will be verified for correctness after the image is written.

Once this process is done, the Fedorator will call for the user's attention and inform them of the fact. At this point, the flash drive may be safely removed and the Fedorator has done its job. It shall revert to a standby state.

Should an error be detected at any stage, the process will be halted and the user will be informed of what happened. If appropriate, they may be asked to attempt to remove the USB stick and try again.

In case there is data present on the flash drive prior to writing the image, the user shall be informed and asked whether it's fine to overwrite.

The primary goal of the Fedorator is to function as an USB flash drive writer, therefore focus will be on the end user. However, I will also consider how the device will be operated by the person setting it up.

### 2.1.1 User

There is only one way a regular person interacts with a Fedorator. This use case is as follows.

- User approaches the device.
- User inserts USB flash drive.
- User follows instructions on screen to select an image.
- Fedorator loads the image on the provided USB flash drive.
- User is informed of the outcome of the process.
- User takes out their USB flash drive, which will contain the selected bootable image of Fedora on it.

### 2.1.2 Operator

A person needs to prepare the Fedorator for public usage. This person shall be called the operator. Often, the person will be a Fedora ambassador. This process will generally involve the following.

- Setting up the device may involve small amounts of assembly.
- The device needs to be connected to a power source.
- The operator should ensure the software and images present on it are up to date.

## 2.2 Looks

The expectation stands that a Fedorator will be used on Fedora stands situated on events. It may follow that it should be fairly sizeable in order to be accessible



and easily usable. A goal may be to attract the interest of passerbys, as well.

Another, possibly the primary goal, is for the device to be intuitive. Since it is sort of a novel concept, people will usually be unfamiliar with the purpose of the device. The looks should assist the user in understanding what the Fedorator does.

A number of features can be made prominent to express the purpose. The Fedorator should clearly convey a Fedora branding. This should be accomplished by having the casing primarily blue in color as well as bearing the Fedora logo. It would also be logical for the USB ports to be plainly visible as they should result in an intuitive explanation.

## 2.3 Interface

An user interface should be presented to the user. We shall consider the most trivial and minimalistic implementation of the Fedorator, and then add potentially desirable features with a rationale.

The trivial Fedorator requires nothing but an USB port. As soon as the user inserts a USB flash drive, it would write the image to it. This certainly works—but immediately we can see room for improvement. Even if the writing process is fast enough, the user will not know when has it finished. An indicator is therefore desirable.

The device can provide visual or aural information. Because the Fedorator may be used in a noisy environment, as conferences often are, a sound may be easy to miss. Therefore, we will choose a LED diode right next to the USB port. For example, the light may blink while the transfer is in progress, and emit constant light once it's complete. Alternatively, such information may be provided by a display.

The next point to consider is a way to start the process other than simply insertion of the USB flash drive. The user may not expect something to happen immediately when the flash drive is inserted. There is also risk in the device starting the write too early (e.g. before the USB flash drive is firmly in the slot).

## 2. SPECIFICATION AND GOALS

---

Waiting for the press of a button, or another, similarly explicit action from the user solves these issues.

Should we wish to allow the user to choose from several images to install, we will need to add more components. One possibility would be presenting several buttons, one for each image. This would be a poor choice in the context of future expansions and changes. A display would be much more flexible and future-proof.

In order to let the user make a selection on a display, we can provide a tactile interface, such as buttons. To navigate the interface comfortably, at least three buttons are required: serving as up, down, and accept.



**Figure 2.1:** Solid Machined Metal Knob offered by Adafruit [2]

An alternative to buttons would be a rotary encoder or a knob. It also allows the user to scroll easily and choose by pressing. It may be more comfortable for scrolling through a menu than repeatedly hitting buttons.

We can choose to forego buttons if we use a touchscreen. If sufficiently large and accessible, it will be comfortable for use.

Besides users performing the basic operation of writing to flash drives, another person who uses the device is the operator. The device may utilize the same interface for management of images stored inside, making it completely standalone, or this maintenance operation may require a computer.

## 2.4 Conclusion

In order to be fully featured, the Fedorator must entail the following hardware components.

- CPU.
- **Display**, providing information and conveying state.
- **Buttons, rotary knob, or a touchscreen.**
- At least one **USB port**.
- **MicroSD slot** or another way to store images.
- **Case**, to hide the components.



---

## Analysis of Possible Solutions

There are several things which should be taken in account when considering alternatives.

**Cost.** The device may be made by hobbyists without much funding, therefore, cheaper is better.

**Ease of assembly.** The people who will be building the device may not be professionals, so making it as simple to set up as possible is desirable. However, somebody who decides to fabricate a Fedorator is likely an enthusiast with some amounts of technical expertise, so there is no need to be overboard if it hampered other factors.

**Customizability and extensibility.** Users with necessary skill should have the freedom to make changes to the Fedorator. Therefore, the hardware shouldn't be locked down. In slang, it should be "hackable".

**Long term maintainability.** Fedora receives new releases approximately every six months [92], it is therefore necessary that the Fedorator is capable of keeping up with the times. The hardware should not impose any tight constraints and any software needs to be possible to upgrade.

## 3.1 Hardware Options

A device such as this needs a computer at its core. I shall go over a number of potential candidates, introducing them shortly and weighing their features.

### 3.1.1 Raspberry Pi



**Figure 3.1:** Raspberry Pi 3 Model B [65]

The **Raspberry Pi** is a small, inexpensive, and fully equipped single board computer. It was originally developed to facilitate better teaching of computer classes by enthusiasts at Cambridge [22]. The original £15 computer went on to be a much bigger success than anticipated and by November 2016, the total sales have reached ten million units [85].

Today the Raspberry Pi foundation continues to develop new versions of the computer [38]. The latest model is **Raspberry Pi 3 Model B**, released in February 2016. It boasts the following features [37].

- 1.2GHz 64-bit quad-core ARMv8 CPU,
- 802.11n Wireless LAN,

- Bluetooth 4.1 and Bluetooth Low Energy (BLE),
- 1GB RAM,
- 4 USB ports,
- 40 GPIO pins,
- HDMI port,
- Ethernet port,
- Combined 3.5mm audio jack and composite video,
- Camera interface (CSI) and Display interface (DSI),
- MicroSD card slot,
- VideoCore IV 3D graphics core.

We can see that some of the necessary components are already included: four USB ports and a microSD slot. This means there is less assembly involved. For example, we may avoid the need to solder entirely.

The dimensions of a Raspberry Pi 3 Model B are 85.6 by 56.5 mm (without connectors). This may prevent it from being used in a more compact design.

Raspberry Pis are widely available and there are local distributors, or resellers in most places around the world [27] [28]. In the Czech Republic, a Raspberry Pi 3 Model 3 may be bought for between 1 046 and 1 359 CZK [73] [57] [9].

The Raspberry Pi is open hardware with the exception of the Broadcom SoC containing the video core. For an operating system, many Linux distributions provide support [62]. Being a computer running a full operating system, support for acting as a USB host device comes naturally.

Finally, due to the sheer popularity of Raspberry Pi, there are many compatible components and accessories available [74] [84], such as specially designated touchscreens [72].

These features, along with the factor of familiarity between hobbyists, make the Raspberry Pi a favorable option for the Fedorator. The disadvantages

include the price, which is being raised by a number of components that are not necessary for the Fedorator, as well as potentially the dimensions.

#### 3.1.2 Arduino

Looking for another platform for hardware development, one which may encompass a smaller form factor, we may be quickly brought to the popular **Arduino** brand.

Arduino as a platform stems from the work of Hernando Barragán, who designed the Wiring development platform in 2003 as a Master's thesis. His goal was to create a low cost and open platform accessible even to non-engineers<sup>†</sup> [19].

There is a broad variety of boards available in the Arduino family [13]. All of them come equipped with a microprocessor, usually from the AVR family, and a set of analog and digital pins. Some include an USB interface for communication with a computer.

Arduino boards are open source hardware available under a Creative Commons Share-Alike license [15].

##### 3.1.2.1 Arduino Uno

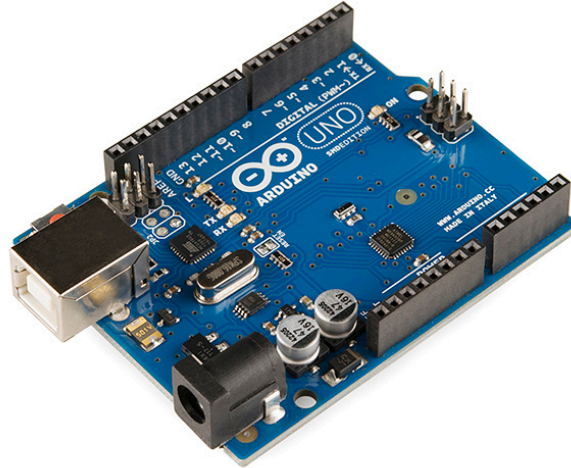
The Arduino Uno is the first board most will lay their eyes on. It is the most popular Arduino board and it's popular for getting started with hardware development [16]. It boasts the following features.

- ATmega328 microcontroller clocked at 16 MHz,
- 32 KiB flash memory,
- 2 KiB SRAM,
- 1 KiB EEPROM,
- 20 digital I/O pins.

---

<sup>†</sup>I include this bit of history especially because it is often misinterpreted.





**Figure 3.2:** Arduino Uno R3 [79]

The dimensions of the Arduino Uno are 53.4 by 58.6 mm. It is powered by the uncommon USB Type B connector.

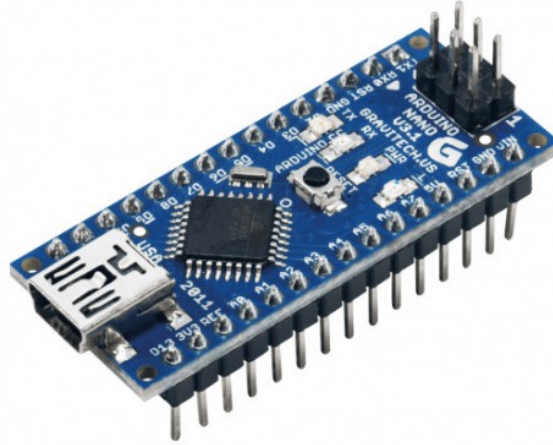
The nominal cost of Arduino Uno is 20.00 EUR (about 530 CZK) [14]. However, thanks to open source hardware, the market with clones has driven the price to as low as 3 USD for a fully compatible piece of hardware [5].

### 3.1.2.2 Arduino Nano

The Arduino Nano is a smaller board (18 by 45 mm) which nonetheless packs almost the same capabilities as the Uno model [12]. It is officially powered by Mini USB, but many clones provide micro USB instead.

### 3.1.3 ESP8266

The ESP8266 is a low-cost Wi-Fi chip [espressif-esp8266]. It is available on its own, but there are many boards putting it in a convenient package with an USB port, for example NodeMCU [61]. This is primarily a Wi-Fi module, but with a CPU frequency of 80 MHz [67], faster than the Arduinos, it may be suited for our purposes.



**Figure 3.3:** Arduino Nano [56]

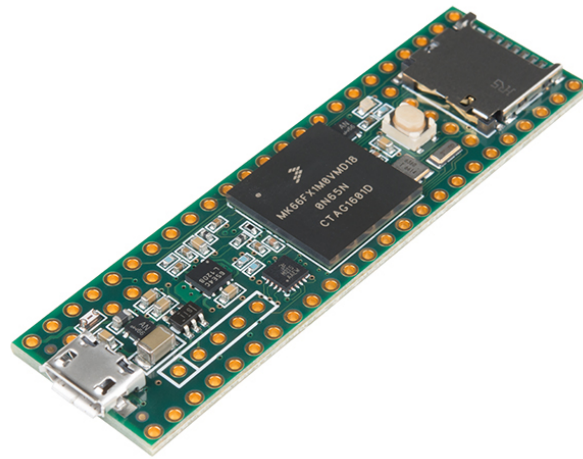
#### 3.1.4 Teensy

Teensy is a family of USB development boards boasting the ARM microprocessor. It is created by Paul Stoffregen and the latest versions, Teensy 3.5 and Teensy 3.6, were funded successfully through Kickstarter [83].

Teensy 3.6 comes equipped with the the following.

- 180 MHz ARM Cortex-M4 with Floating Point Unit,
- 1M Flash, 256K RAM, 4K EEPROM,
- USB Full Speed (12 Mbit/sec) port,
- USB High Speed (480 Mbit/sec) port,
- Native microSD card port,
- CAN bus ports, serial, SPI, I2C, I2S, Ethernet and more.

The size is stated as 2.4 by 0.7 inch (about 6 by 1.8 cm).



**Figure 3.4:** Teensy 3.6 [80]

Teensy 3.6 is officially available for 29.25 USD (about 715 CZK) without shipping [teensy-36], or, for instance, at local distributors in the Czech Republic for 1 198 CZK [78]. The cost is comparable with a Raspberry Pi.

The fast CPU, on-board USB host port, and MicroSD slot make the Teensy 3.6 a strong contender.

### 3.1.5 FPGA

One final option I considered is to forgo a CPU completely and design a custom chip instead. FPGA stands for field-programmable gate array and it's a class of integrated circuits which can be reconfigured at will. Thanks to allowing exact configuration of logic elements and memory blocks to serve a specific purpose, they're extremely efficient at signal processing [75].

By designing custom circuitry to handle copying of data from an SD card to a USB flash drive, the copier could be very fast, bottlenecked only by the connected devices.

### 3. ANALYSIS OF POSSIBLE SOLUTIONS

---

Popular FPGA manufacturers include big names such as Xilinx [93] and Intel [47]. The field of open hardware FPGA tools is lacking, although there do seem to be some contenders, like Papilio [39].

However, programming FPGAs is a task significantly more difficult than software programming, requiring the management of input and output signals and careful timing. I have deemed the task of implementing the USB protocol at such a low-level too complex and out of scope.

## 3.2 Display Options

The options for displays are truly diverse. We may choose between small and simple displays to large LCDs. A few representatives are provided.

### 3.2.1 Character-based LCDs

Simple single, two-, or four-line character based LCDs, typically blue in color, are a frequent sight in industrial devices. Indeed such a LCD would be able to handle the Fedorator in terms of functionality - there is sufficient space to display the status or progress. However, it's not a visually attractive alternative, being incapable of even displaying a logo.

### 3.2.2 OLED



Figure 3.5: Common OLED display available on Adafruit [1]

An interesting option are small, monochromatic OLED displays. A very common variant is 1.3 inches and 128x64 pixels in resolution [1]. Despite their small size, the contents are lit well, clearly visible and sharp. An OLED display would be a good choice for a compact design.

#### 3.2.3 TFT

TFT are a variant on LCDs with better image quality [lifewire-what-is-tft-lcd]. The term primarily refers to small and medium sized full-color displays which can be connected to DIY devices. Conveniently, TFT displays often come bundled with a resistive touchscreen controller attached to them.

### 3.3 Other Component Options

Each component increases the difficulty of assembly, as it imposes the necessity of buying the component, as well as of connecting it properly and seating it in place.

Basic components such as push-buttons and LEDs are widely available from local distributors.

Every device needs to be powered in some way. Because the Fedorator will be stationary, we shall let it be powered by cable, preferably a commonly available one like micro USB. There is no need to consider a battery.

### 3.4 Casing Options

Because the device will be made of several components, it's desirable to contain them in a case. There are not many reasonable options present, but I have shortly considered alternatives.

#### 3.4.1 Wood

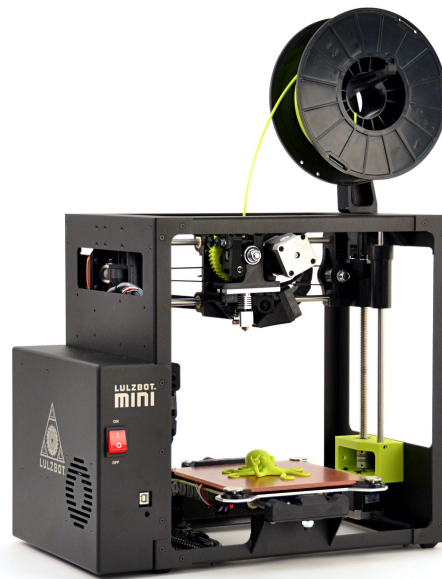
A wooden build would certainly look interesting, even if not modern. However, because it would have to be handmade, it's not a practical choice. Few people

have wood crafting skills and would invest time in building the moderately complex case.

#### 3.4.2 Plastic with Mold

The common manufacturing process for producing plastic parts is called injection molding. As the name suggests, this method requires the production of a mold, which is an expensive proposition unless a significant amount of devices is manufactured. It makes no sense to speak of molding when the numbers of parts produced is less than 1 000 [66].

#### 3.4.3 3D Printed



**Figure 3.6:** LulzBot Mini 3D printer [4]

The advent of 3D printing technology is a staple of today's day and age. Also called additive manufacturing, it allows for bringing objects designed on a computer into reality. Because an object can be printed in the matter of hours, 3D printing is ideal for prototyping.

In tech circles, 3D printers are a common find these days. For example, many hackerspaces have one available [42]. It is likely anybody wishing to build a

Fedorator would be able to attain access to one. A 3D printed case is simply the matter of getting ahold of a printer and enough printing material (plastic).

For a technically hobby project like this, a 3D printed case is the logical choice.

### 3.4.3.1 3D Printing Materials

A 3D printer needs to be provided with material to use. The two most commonly used materials are **PLA** and **ABS** [[all3dp-best-filament-types](#)]. Both are available in a variety of colors and look largely the same, but they do have a few distinctive properties [41].

**PLA** has a harder surface and is more prone to break when bent. It is well suited for objects such as household items and gadgets. PLA is made from plant material and is biodegradable.

**ABS** is the stronger material when printed at the correct temperature. It is well suited for mechanical parts. When printing, ABS is more prone to warping. Finally, ABS is not biodegradable.

Prices are largely the same between PLA and ABS filaments.

PLA as a material is slightly better suited for the purpose of being a static shell. However, ultimately, both materials should prove to work fine, so creators can choose whichever they have available.

## 3.5 Roundup

For the core of the device, I deem two distinct categories based on form factor.

A large device, complete with a larger display, will be served brilliantly by the Raspberry Pi computer. The goal of being easily extensible would also be fulfilled.

We may, however, choose to go for a smaller design with the OLED display. That would bar the Raspberry Pi from being an option. Over here, I see potential in the Arduino Nano device. If it turned out to be sufficiently powerful to

### 3. ANALYSIS OF POSSIBLE SOLUTIONS

---

drive a microSD card reader and a connected USB flash drive, the size would be quite convenient and the price very attractive—when we're speaking of clones. Should that turn out not to be the case, we may start considering stronger devices, such as the Teensy.

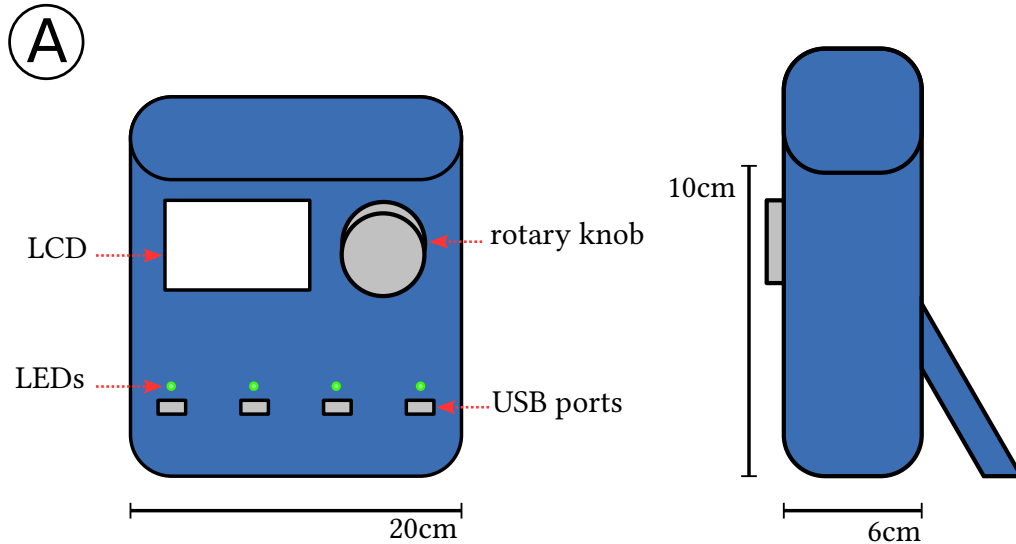
Whatever the design, a 3D printed case is a reasonable decision with regard to the typical fabricator of the device.

## 3.6 Concepts

Based on this analysis, I have created three distinct Fedorator concepts.



### 3.6.1 Concept A



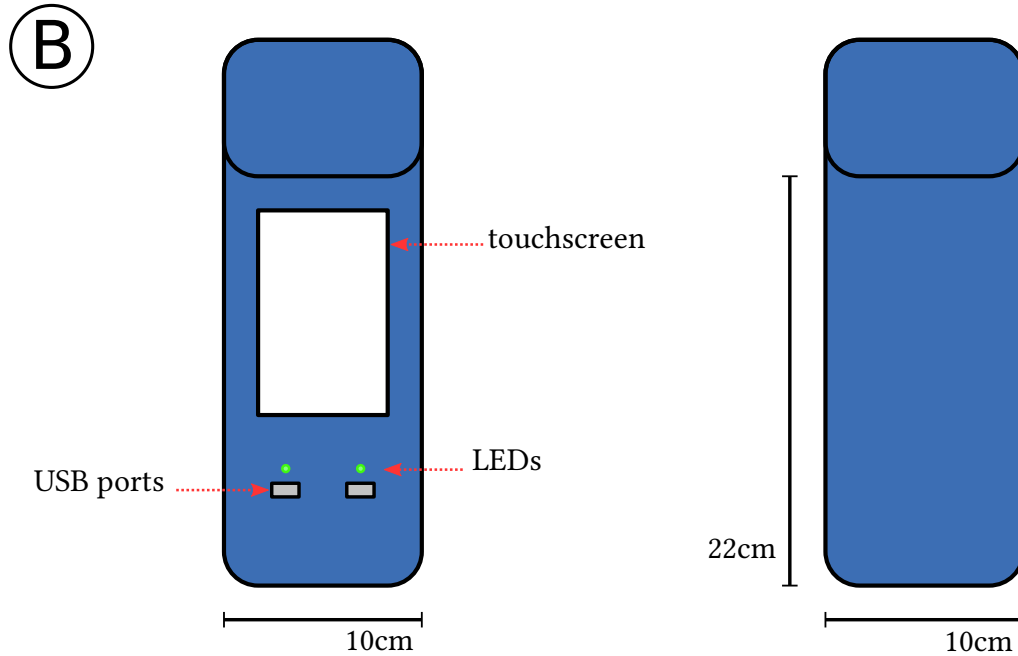
**Figure 3.7:** Concept A

The first concept is a moderately large device which is to be fixed in place and used from a standing posture. It provides an LCD with on-screen information on the status and progress. The rotary knob allows for controlling the menu and making a selection. It may be replaced with three buttons if so desired. There are four USB slots present, each with a corresponding status light. In order to be stable, the device needs support from the back.

Due to the size of the display and the number of USB ports, putting a Raspberry Pi computer at the core makes the most sense.

The Fedorator needs to be plugged into a power source, so there will be a cable running from the back.

### 3.6.2 Concept B



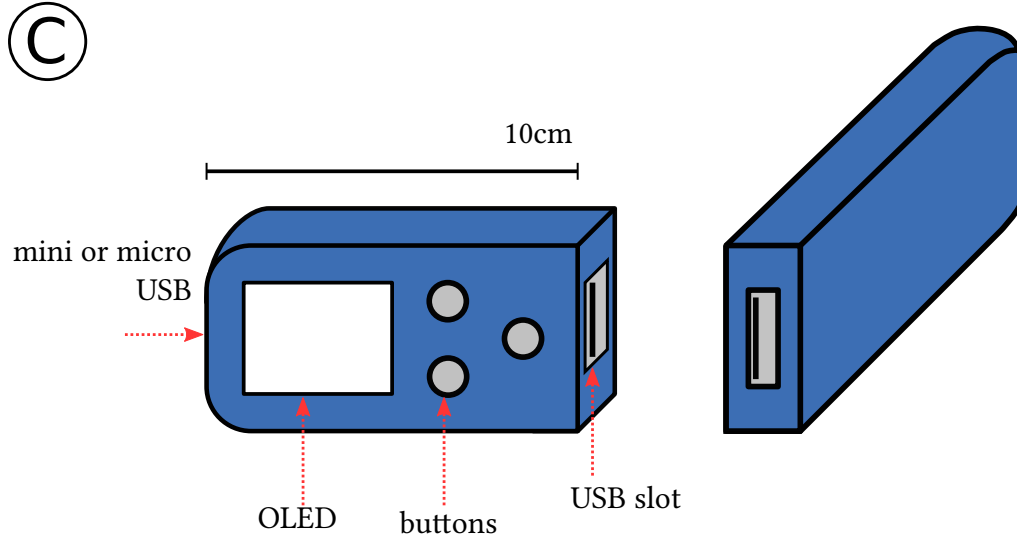
**Figure 3.8:** Concept B

The second concept is similar in some ways to the first one, but with some tweaks. The knob is removed, the sole control is a prominent touchscreen positioned vertically. As a consequence, the device can be thinner. There are now only two USB slots. The device is sleek and symmetric.

In order not to have stability issues, it may be necessary to ensure the Fedorator is heavy enough in order to keep the Fedorator in place as it's being used.

This design is also powered by a Raspberry Pi.

### 3.6.3 Concept C



**Figure 3.9:** Concept C

The third and final concept is a radical departure from the first two. Instead of standing still while being manipulated, this Fedorator is designed to be picked up and held in a hand. The OLED screen is small yet sharp and despite providing only a binary image it works well enough for our purposes. The three buttons allow for selecting the desired live image.

Due to its small size, this device needs a small system at its core. In an attempt to keep the costs low in contrast to other designs, I will attempt to use the Arduino platform.

This design still needs to be powered by cable. This cable will be permanently attached by the means of enclosure. This has the positive side effect of discouraging people from attempting to leave with the Fedorator.

Because it only has a single USB slot, it may be desirable to provide multiple instances of this Fedorator.

## **3.7 Decision**

I have showed five people these concepts and asked them for their feedback. In general, response to concept A was lukewarm. The device looks bulky, it is not very aesthetically pleasing. The number of ports is also excessive given the purposed use by onlookers at marketing stands. However, people showed excitement about both concept B and C. Because they differ significantly, I will strive to construct a prototype for each.

---

## Raspberry Pi-based Solution

I have decided to create a Fedorator utilizing the Raspberry Pi computer at its core. This design is based on the previously introduced Concept B (fig. 3.8).

### 4.1 Specification

The device consists of:

- Raspberry Pi 3 Model B. Compatibility with Raspberry Pi 2 Model B may be possible, but not a primary goal.
- A compatible 3.5 inch touchscreen (minimum resolution  $480 \times 320$  pixels) positioned vertically.
- A microSD card holding the system and several images.
- Two male-to-female USB cables.
- A 3D printed case.

### 4.2 Software

The software running on the Fedorator was written to support support both Raspbian, the Linux distribution designed and officially supported for Raspberry Pi [71], and the eponymous Fedora operating system. To facilitate

easier testing, it's also possible to run the software on a regular Linux desktop.

The current version of the Fedorator software may be found in the appendix. It is also made available online on GitHub, a code hosting service, with the GNU General Public License v2.0 [76]. On GitHub, continued development will take place.

The software running on the Fedorator is written in Python and uses the free open source Kivy framework [50] to provide a touchscreen interface. It is designed to start immediately after boot and stay running indefinitely.

### 4.2.1 Behavior

In standby, the screen shows prominently the Fedora logo as well as the name of the device (fig. 4.1). It prompts the user to insert a flash drive.

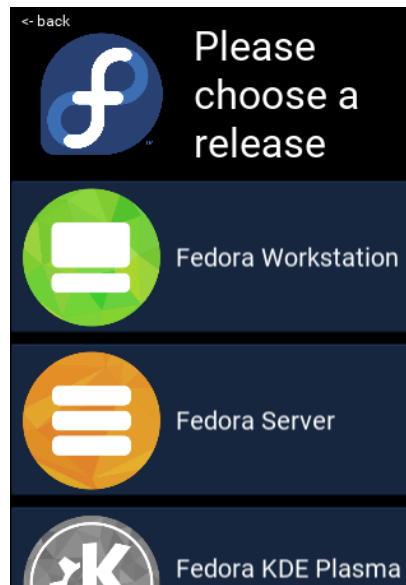


**Figure 4.1:** Fedorator: the standby screen

When the user inserts a flash drive, the text changes, stating “Tap to begin”. The size of the flash drive is also displayed at the bottom part of the screen.

Tapping will transition the screen into a menu listing releases available for writing (fig. 4.2). This list can be browsed by a swiping a finger or stylus across

the screen. Tapping the top left corner will always bring the user back to the previous screen, as indicated by the faint “back” text.



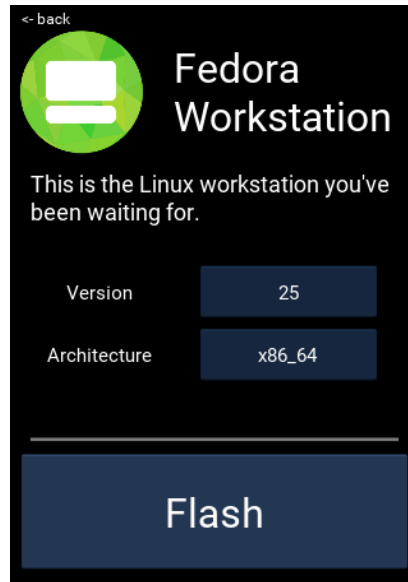
**Figure 4.2:** Fedorator: picking a release from the list

After tapping a desired release, a screen with details is presented (fig. 4.3). It contains a short official description of the release and offers to set the version and architecture, should the user wish for something other than the defaults.

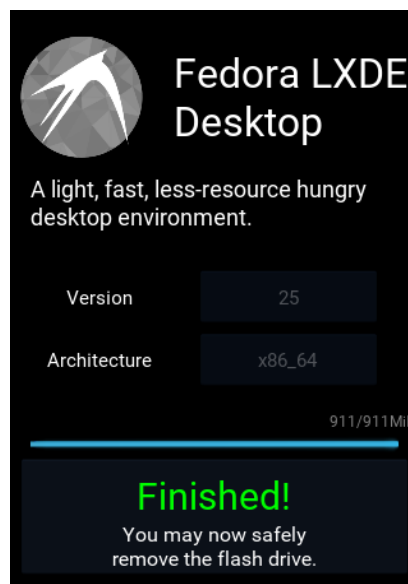
If the user is happy with the choice, they can press the sizeable “Flash” button. Before the flashing operation begins, a popup appears warning the user that all data on the flash drive will be destroyed. The user must acknowledge this with the “Continue” button, and that begins the process of writing the image to the flash drive.

A progress bar shows how far along the write is.

After the write is complete, the screen shows “Finished!”, pulsating in green to grab attention. It also tells the user they may now safely remove the flash drive. A single tap at this point will bring the Fedorator back to the standby screen.



**Figure 4.3:** Fedorator: taking a look at the Fedora Workstation release



**Figure 4.4:** Fedorator: Fedora LXDE Desktop was successfully written to the flash drive



The device returns to standby mode after a tap or after 30 seconds. If at any point before the write finishes the flash drive is removed, the software goes back to the standby screen immediately.

## 4.3 Technical Implementation

The Fedorator software leverages features provided by the Linux operating system for its feature set. Following the widely discussed UNIX “Everything is a File” philosophy [43], certain convenient directories provide information about the connected devices after the kernel takes care of them.

In particular, `/dev/disk/by-path` is used to watch for newly connected USB storage devices, and `/proc/partitions` is queried for information on the size of the medium.

The primary function is the flashing operation. After a write is initiated, the `umount` command is called on the target partition, because writing to a device which is mounted may result in corruption.

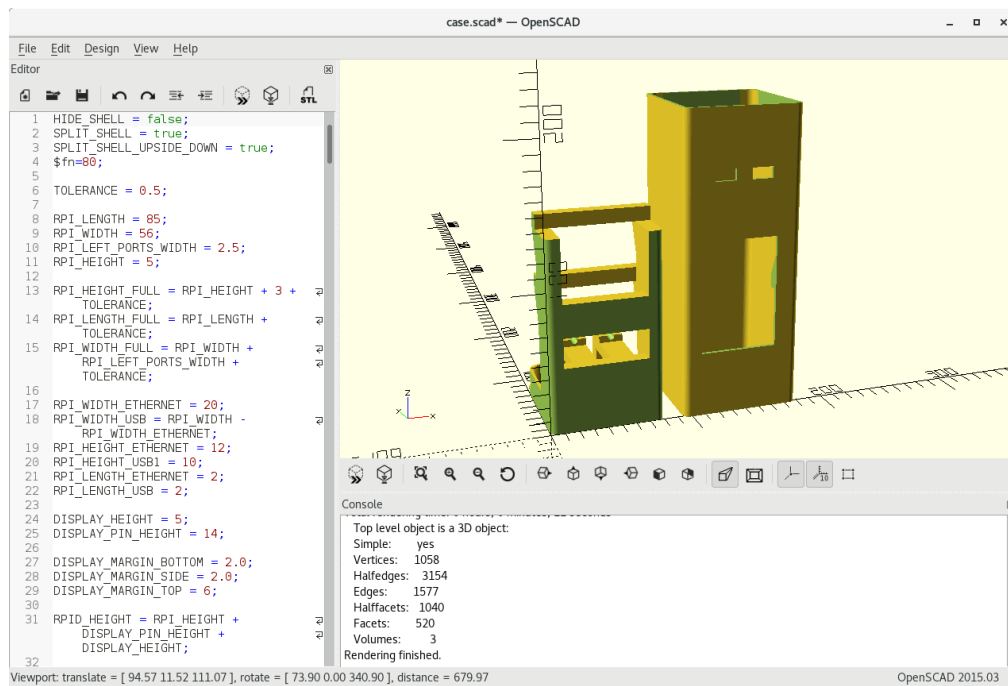
Next, the software performs a direct copy to the USB drive. This is akin to the `dd` utility. In order to achieve the best performance, the `sendfile()` [52] function is used. Because `sendfile()` performs copying directly in the kernel, it is faster than an equivalent task involving reads and subsequent writes.

After the copy finishes from the perspective of the system, it is actually not yet safe to remove the flash drive. Flash drives keep an internal buffer and may need more time in order to finish writing the data. By utilizing the `udisksctl power-off` command, we issue the system to shut down and unplug the USB device as soon as it stops being busy.

## 4.4 Case

I created the model for the case using the **OpenSCAD** software. In OpenSCAD, objects are designed using a descriptive language. One consequence is that when a value such as the length of a certain sub-component is used in several places, it may easily be defined once and used by its name repeatedly. This allows for easy changes to just about any relevant dimension and also enhances

## 4. RASPBERRY PI-BASED SOLUTION



**Figure 4.5:** A screenshot of the OpenSCAD software with the case open. Both the source code and the rendered model can be seen

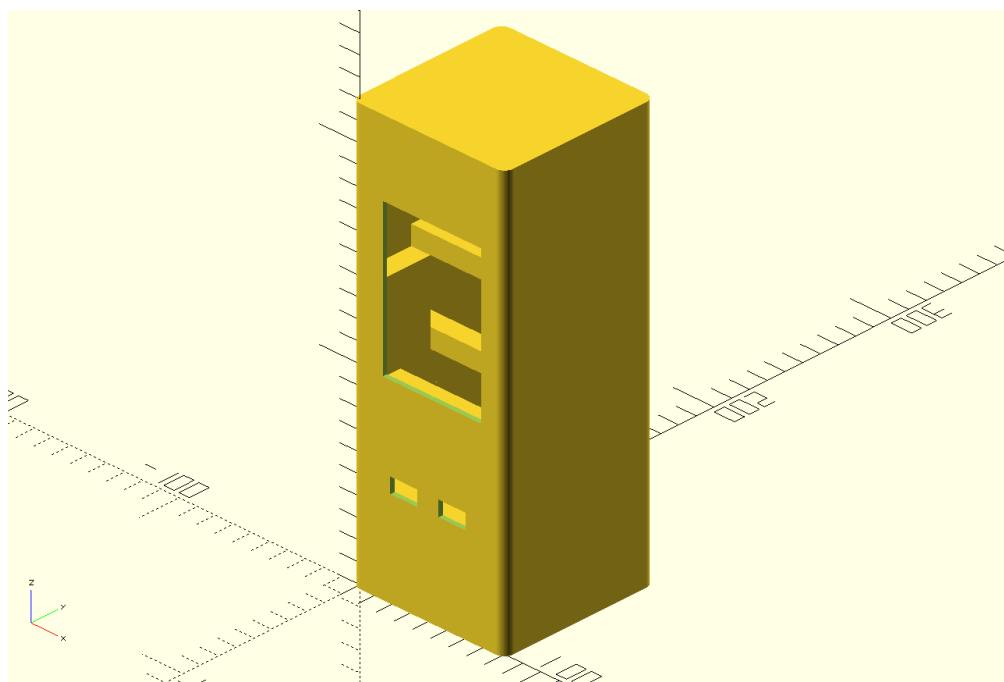
understanding of the source code. In proper terms, the model I designed is *parametric*.

OpenSCAD allows exporting the model to the STL format. This format can then be processed by a slicing program like Slic3r or Cura to prepare for printing.

The case I designed for the Fedorator is split into two components, which may be defined as inner and outer, or bottom and top. These two components are printed separately, but lock together when the device is constructed.

The **inner part** consists of the floor, a pair of supports designated for USB connectors, and a holder for the Raspberry Pi and display pair. The enclosure for the Raspberry Pi was designed with meticulous attention to dimensions, otherwise the device would not fit or move too much.

The **outer part** is majorly the shell and cap. There are four holes in the shell, one for the display, two for the USB ports, and one in the back for the power cable.



**Figure 4.6:** A render of the constructed case, courtesy of OpenSCAD

Both parts are designed in such a way that they fit snugly together. The top part can be carefully put over the bottom part and holds well unless taken by force.

In a misguided attempt at making the device easy to assemble, I had decided to forego any screws or fastening bolts in the design. This means the components inside hold by gravity, or are fixed by other means such as a tightening strap.

## 4.5 Construction of a Prototype

I have put a functional prototype of the Fedorator together.

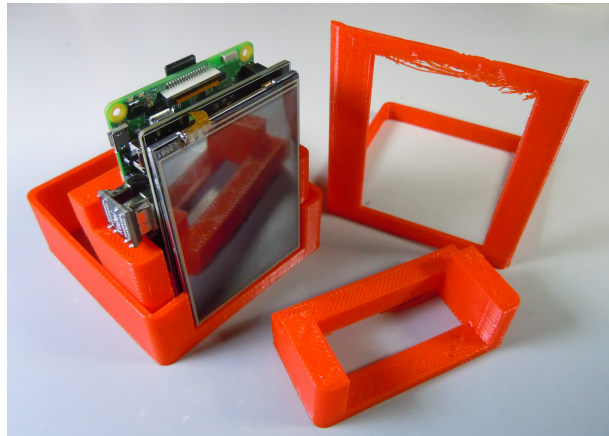
The prototype consists of all the required components. The screen is a 3.5 inch HDMI touchscreen display with support for up to  $1920 \times 1080$  pixels in resolution [6]. However, in the prototype, the resolution is kept as  $480 \times 320$  to ensure broadest support. Sadly, drivers for this display are proprietary and only

#### 4. RASPBERRY PI-BASED SOLUTION

---

provided for Raspbian [11]. Hence, the prototype runs the Raspbian operating system.

The case for the prototype was printed on a LulzBot Mini [4] 3D printer. The PLA material was used for filament.



**Figure 4.7:** A test print with a Raspberry Pi seated inside

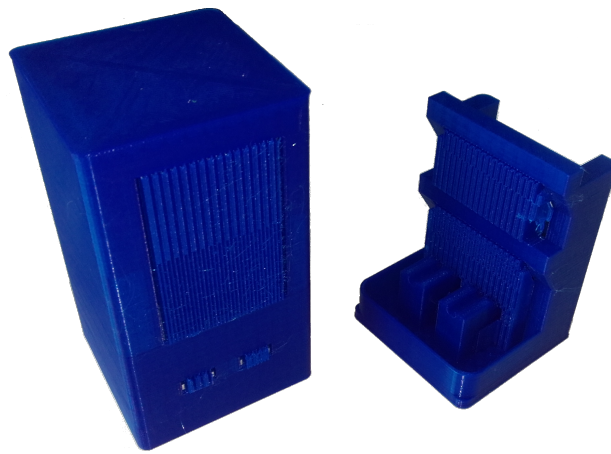
In order to test the model's dimensions and fitness, a series of test prints was done first. After certain parameters were verified, such as whether the Raspberry Pi is seated correctly, the full case was set to be printed. The poor behavior of the test prints with regard to bridges has led to the decision to print the full case with supports<sup>†</sup>.

Due to the printer's dimensions, the height of this prototype had to be lowered to 152 cm and both parts was printed individually. Each part took about ten hours to print.

The prints were successful and the device ready to assemble.

---

<sup>†</sup>Supports are brittle, easy to remove material printed under overhangs in the object.



**Figure 4.8:** The case print complete with supports

## 4.6 Instructions

Instructions on how to build a Fedorator are included in the BUILD.md file of the public repository, also present in the appendix.

- **Building a Fedorator:** A short intro tells the reader about the Fedorator being open hardware.
- **Components:** The necessary components are listed, with a suggestion on where to buy them.
- **Software:** Contains instruction on how to set up the Fedorator software on the Raspberry Pi.
- **Hardware-specific setup:** Gives some tips on things that may differ between configurations.
- **Assembly:** Contains a description of the case, instructions on printing it and how to put it all together.

## 4.7 Speed Measurements

To test performance in different situations, I have measured how long the Fedorator takes to write the Fedora Workstation image (1.34 GiB) to several



**Figure 4.9:** The Fedorator prototype with two USB flash drives plugged in



**Figure 4.10:** The Fedorator prototype done flashing the Fedora Workstation system

#### 4. RASPBERRY PI-BASED SOLUTION

---

different flash drives. Several measurements were done in order to gain the average. There was negligible deviation.

**Table 4.1:** Fedorator speed measurements

Vendor	Name	Capacity	Average time
Sony		15.8 GiB	6 m 07 s
Sony	MicroVault	1.9 GiB	6 m 27 s
LaCie	iamaKey V2	15.1 GiB	2 m 08 s
PetalIT		14.6 GiB	2 m 46 s
Silicon Motion	Flash Drive	15.0 GiB	4 m 26 s
Silicon Motion	Flash Drive	3.7 GiB	3 m 38 s
Corsair	Flash Voyager	3.8 GiB	4 m 34 s
Unknown <sup>†</sup>		3.8 GiB	6 m 59 s

We can see that the time varies a lot depending on the speed of the particular flash drive. Even flash drives from the same vendor can differ in their writing speed.



**Figure 4.11:** The various flash drives tested

---

<sup>†</sup>This flash drive is inscribed with the text “Ústav leteckého zdravotnictví Praha” and Linux identifies it only as “Unknown counterfeit flash drive”.



## 4.8 Conclusion

I have devised and designed a device fulfilling the specified requirements for a Fedorator. A prototype was built and is functional. It is ready to be tested in the field.

Overall, creating the device was a success, although there is still work to be done. I'm not satisfied with the components used for the prototype, in particular, the proprietary screen. I would like to be able to recommend a standard screen, so I will be searching for alternatives.

Another issue is the lack of fixation for the components. During regular usage, the screen may wiggle a small amount or be at a slightly off angle. The next iteration on this design will include screws, fastening bolts, or a similar "tried and tested" solution.

Testing has shown the current version of the software works according to the specifications. It is built with provisions for future improvements. Continued development may bring features such as the addition of secondary data partitions to the resulting flash drives.



---

## Interlude: USB Mass Storage

In the previous chapter, I have created a USB flash drive copier solution using the Raspberry Pi computer. Thanks to the computer running a full operating system, all the details of the USB protocol were handled transparently and kept away from me. However, for a solution closer to bare metal, this will not be the case. In order to have any hope of understanding the problem space, I committed to study the parts of the USB protocol relevant to understanding mass storage.

### 5.1 Introduction to USB

USB, standing for Universal Serial Bus, is an industry standard defining the hardware and protocols surrounding connection and communication of electric devices to computers[34].

The USB 1.0 specification was introduced in 1996 and defined data transfer rates of 1.5 Mbit/s and 12 Mbit/s [46]. In 2000, the USB 2.0 specification was released, which a significantly higher maximum data transfer rate of 480 Mbit/s[88].

The USB 3.0 specification, released in 2008, defines a new SuperSpeed transfer mode. This mode is claimed to support a transfer speed of up to 5 Gbit/s in full-duplex, with 350 MB/s throughput in current implementations [44]. SuperSpeed plugs are distinctively identified by blue inserts.

USB flash drives are data storage devices implementing the USB interface. Many USB flash drives already implement USB 3.0 [18], however, all have backwards compatibility as long as they use a traditional USB-A port. For our purposes, it's reasonable to assume people will use any old flash drive they have lying around, rather than specifically brand new ones.

In order to create a device behaving as an USB host, I have studied the Universal Serial Bus Specification revision 2.0 [88]. Even if I won't be working with USB at a particularly low level, an overview is essential in order to understand behavior and judge parameters such as speed.

The device I will be designing will only need to support the connection of USB flash drives to a single USB port, so only relevant information is taken in consideration.

## 5.2 USB Specification

USB is a cable bus providing data exchange between a host device (usually a computer) and several connected peripherals which may be attached and used independently. USB defines a host-scheduled and token-based protocol for communication with peripherals.

A USB system is prescribed as a tiered star topology involving hosts, hubs, and functions (logical devices). USB allows for complex asymmetrical topology, but in our situation, only a root hub will be present.

### 5.2.1 USB Host

A single host is present in any USB system. The Host Controller, implemented in hardware or software, has many responsibilities. It initiates connections and manages all communication and bandwidth between software and connected devices.

The USB host also supplies power to connected devices.

### 5.2.2 USB Device

A USB device is a peripheral connected to the host utilizing the USB protocol. There is a number of class codes defined in the USB standard [86]. Common classes include the HID (Human interface device) class encompassing keyboards and mice, the printer class, the video class, and indeed, the mass storage class.

### 5.2.3 USB Hardware

USB uses a four-wire cable, encompassing a twisted pair of data wires and as a  $V_{\text{BUS}}$  and GND pair delivering a nominal +5 V power to devices.

The clock is transmitted along with data using a clock encoding scheme. There are three data rates available:

- Low-speed, defined at 1.5 Mb/s with limited capacity.
- Full-speed, defined at 12 Mb/s.
- High-speed, defined at 480 Mb/s.

### 5.2.4 USB Protocol

On the low level, USB functions as a polled bus. Transactions, as initiated by the Host Controller, usually involve the transmission of up to three packets. The token packet allows devices to identify whether they're the intended recipient of the packet.

To facilitate communication, the host and device negotiate device endpoints. A device may be composed of several independent endpoints. Endpoints are simplex—data flow is defined in a single direction.

A USB pipe is an association between an endpoint on a device and software on the host. In other words, pipes are what the software handles and uses for data transmission. Pipes can be configured to behave in *message* or *stream* modes, which either do or do not have a USB-defined structure, respectively.

Pipes define a transfer type and hold several characteristics, such as directionality, the transfer type, the amount of bandwidth allocated, and maximum data

payload sizes. A pipe called the *Default Control Pipe* is always present and is used for device configuration.

USB offers four transfer types for a pipe:

- **Control Transfer:** Simple request/response communication used to communicate commands and status.
- **Isochronous Transfer:** Continuous communication without guarantee for errors intended for real-time data.
- **Interrupt Transfer:** Timely and reliable latency-bound communication.
- **Bulk Transfer:** Large and bursty, but non-time sensitive data transmission, utilizing any bandwidth available.

Understandably, bulk transfer is the type used for mass storage.

### 5.3 USB Mass Storage Device Class

The USB mass storage device class, or MSC for short, is a set of protocols allowing an USB device to enable the host to directly access a data storage. A device implementing this protocol usually appears as an external hard drive. All sorts of devices implement MSC, including hard drives, optical drives, solid state drives, card readers, or instruments such as cameras and mobile phones [17].

A number of subclasses and transport protocols is defined for the MSC. MSC allows for certain protocols defined outside of USB to be used, such as SCSI or totally obscure ones like QIC-157 or SFF-8070i. The USB defined transports are Bulk-Only transport (BBB or BOT) and the obsolete Control/Bulk/Interrupt (CBI) transport, only in use for USB floppy disk drives [89] [87].

As of USB 3.0, there are provisions for a USB Attached SCSI protocol (UASP) protocol. Particularly high performance mass storage devices may use that standard [48], but few devices have adopted it so far [49]. In particular, as late as 2012, almost all USB flash drives on the market implemented only the BBB protocol [45].

---

## Microcontroller-based solution

I made an attempt at designing and creating a compact variant of the Fedorator modelled after Concept C (fig. 3.9), dubbed Fedorator Tiny. I set to work with some components I already had ready. I began working on a version consisting of an Arduino Nano device, a monochrome 128x64 OLED display, a microSD card reader, and the USB Host Mini board [23]. For drawing on the display, I used the `esp8266-oled-ssd1306` library [25].

The limitations of the Arduino device cropped up quickly, when the code, together with drivers for the display and SD card reader, stopped fitting inside the small 32 KiB flash memory.

I reproduced the same work on a NodeMCU device, but had troubles with the SD card library.

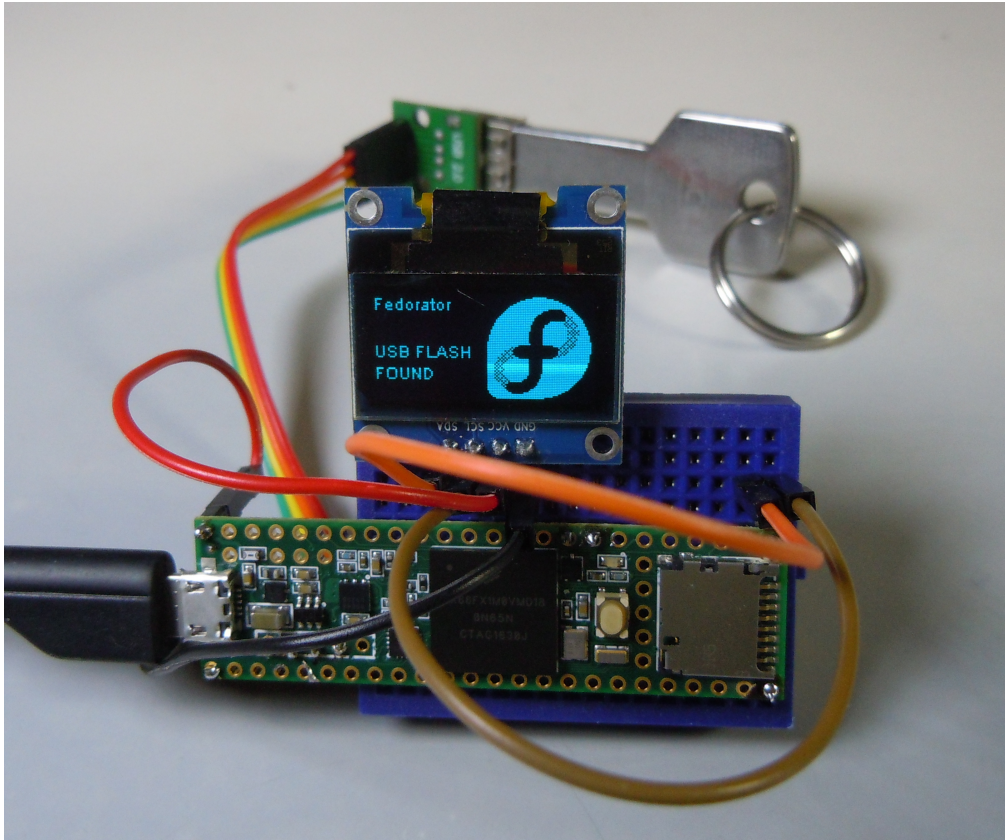
I'm now convinced that the ideal core for the Fedorator Tiny would be **Teensy 3.6**. With a SD card slot on board as well as a built in USB host capability, it requires no external active components to fulfill all requirements.

There is a library for working with the USB host port in development by the Teensy creator, `USBHost_t36` [82]. In the present, it only supports keyboard and MIDI devices. The best course of action here would be to add mass storage functionality to this library. I have begun the work on this by studying the USB and SCSI protocols. Code-wise, I performed a small trial in the form of detecting the connection of an USB flash drive as seen on figure ??, but full support would be the subject of another work of a significant scale.

## 6. MICROCONTROLLER-BASED SOLUTION

---

All work was done in C using the PlatformIO Core framework, which provides a unified build system for a large number of development platforms [68]. Source code may be found in the appendix.



**Figure 6.1:** Teensy detecting a USB mass storage device



---

## Conclusion

In this thesis, I was tasked with a problem and an idea for a solution. Due to the fact that DVD drives have stopped being common in computers, giving out DVDs for people to try out the Fedora operating system was no longer a viable option. Because mass producing USB flash drives is significantly more expensive, an alternative was conceived in a device that would write to flash drives on demand dubbed the Fedorator.

The purpose of the device isn't just to write flash drives—that could be easily achieved with just a computer, or perhaps a commercial piece of hardware. The Fedorator is intended to be present on marketing stands and draw attention and interest.

I have first specified how this device would behave and what parameters should its look fulfill. Next, I researched the options for the device's components—the hardware as well as the case. I analyzed these options with regard to cost, ease of assembly, extensibility, and long term maintainability. Based on this analysis, I created three distinct concepts. Due to positive feedback, I went ahead and attempted to construct two of them.

I turned a concept of a Fedorator driven by a Raspberry Pi into a reality. I decided on the components, wrote the software and designed a case. Then I showed the solution to be reasonable by building a functional prototype. Constructing the prototype has allowed me to determine which things to improve on in the next iteration.

I also strived to design and build a smaller variant on the Fedorator device. I have done the research and performed some trials, but failed to secure a solution that would not require a large time investment. This would be a good topic for a follow-up master's thesis.

## **Future Work**

The Fedorator prototype that was built is ready to be showcased at local events in order to garner feedback, and there are a few noted enhancements that can be made.

The source code for all the work I have done is available on GitHub. Thanks to the nature of open source, this means anybody can contribute improvements. The Fedorator project has a future at the hands of me and other enthusiasts.

---

## Bibliography

1. ADAFRUIT. *Monochrome 1.3" 128x64 OLED graphic display* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.adafruit.com/product/938>.
2. ADAFRUIT. *Solid Machined Metal Knob - 1" Diameter* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.adafruit.com/product/2056>.
3. AKL TECHNOLOGY ENGINEERING AND SERVICES. *USB Stick CopyStation* [online]. 2016 [visited on 2017-05-14]. Available from: <http://akl-it.com/home/products/duplication-system/copystation/usb-stick-copystation/>.
4. ALEPH OBJECTS, Inc. *LulzBot Mini* [online]. 2016 [visited on 2017-05-15]. Available from: <https://www.lulzbot.com/store/printers/lulzbot-mini>.
5. ALIEXPRESS. *1pcs New 2016 - UNO R3 ATmega328P/CH340G MicroUSB. Compatible for Arduino UNO Rev 3.0 (hei)* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.aliexpress.com/item/1pcs-New-2016-UNO-R3-ATmega328P-CH340G-MicroUSB-Compatible-for-Arduino-UNO-Rev-3-0-hei/32688321693.html>.
6. ALIEXPRESS. *3.5 inch HDMI LCD TFT Touchscreen Display 1920 1080 for Raspberry pi 2 and Pi 3 Model B* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.aliexpress.com/item/3-5-inch-HDMI-LCD-TFT-Touchscreen-Display-1920-1080-for-Raspberry-pi-2-and-Pi/32774253132.html>.

## BIBLIOGRAPHY

---

7. ALTAREC. *1:10 USB 3.0 Copy Cruiser Mini Duplicator for Windows and Mac* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.aleratec.com/1-to-10-usb-3-0-thumb-flash-drive-duplicator-copy-cruiser-mini-330113.html>.
8. ALTAREC. *1:118 USB Copy Tower SA Duplicator - Stand-Alone 1:118 USB Flash Drive Duplicator* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.aleratec.com/330118.html>.
9. ALZA.CZ. *RASPBERRY Pi 3 Model B* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.alza.cz/raspberry-pi-3-d4078690.htm?catid=18851862#prislusenstvi>.
10. AMAZON. *EZ Dupe 2 Copy Portable USB Duplicator and Tester EZ2USB* [online]. 2017 [visited on 2017-05-01]. Available from: <https://www.amazon.com/EZ-Portable-Duplicator-Tester-EZ2USB/dp/B003N4D7BE>.
11. AMBER. *Install 3.5" HDMI Touch Screen Linux Driver on Raspberry Pi* [online]. 2017-01-18 [visited on 2017-05-15]. Available from: <http://osoyoo.com/2017/01/18/install-3-5-hdmi-touch-screen-linux-driver-on-raspberry-pi/>.
12. ARDUINO. *Arduino Nano* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.arduino.cc/en/Main/arduinoBoardNano>.
13. ARDUINO. *Arduino Products* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.arduino.cc/en/main/products>.
14. ARDUINO. *Arduino Uno Rev3* [online]. 2017 [visited on 2017-05-15]. Available from: <https://store.arduino.cc/arduino-uno-rev3>.
15. ARDUINO. *Frequently Asked Questions: What do you mean by open-source hardware?* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.arduino.cc/en/Main/FAQ#toc3>.
16. ARDUINO AG. *Arduino UNO* [online]. 2017 [visited on 2017-05-15]. Available from: <http://www.arduino.org/products/boards/arduino-uno>.
17. ARM KEIL. *MSC: Mass Storage Class* [online] [visited on 2017-05-15]. Available from: [https://www.keil.com/pack/doc/mw/USB/html/\\_m\\_s\\_c.html](https://www.keil.com/pack/doc/mw/USB/html/_m_s_c.html).
18. AUSTIN, Patrick; AGUILERA, Ray. *The Best USB 3.0 Flash Drive* [online]. 2017-04-13 [visited on 2017-05-15]. Available from: <http://thewirecutter.com/reviews/the-best-usb-3-0-thumb-drive/>.

19. BARRAGÁN, Hernando. *The Untold History of Arduino* [online]. 2016 [visited on 2017-05-15]. Available from: <https://arduinohistory.github.io/>.
20. BJONNH. *udev\_serialcopier: A stupid quick and dirty udev-based USB copier*. [online]. 2016-06-26 [visited on 2017-05-15]. Available from: [https://github.com/bjonnh/udev\\_serialcopier](https://github.com/bjonnh/udev_serialcopier).
21. BŘÍZA, Martin et al. *Fedora Media Writer* [online]. 2017-05-11 [visited on 2017-05-15]. Available from: <https://github.com/MartinBriza/MediaWriter>.
22. CELLAN-JONES, Rory. *A 15 pound computer to inspire young programmers* [online]. 2011 [visited on 2017-05-12]. Available from: [http://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a\\_15\\_computer\\_to\\_inspire\\_young.html](http://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html).
23. CIRCUITS@HOME. *USB Host Shield* [online]. 2016 [visited on 2017-05-15]. Available from: [https://www.circuitsathome.com/arduino\\_usb\\_host\\_shield\\_projects/](https://www.circuitsathome.com/arduino_usb_host_shield_projects/).
24. DISC MAKERS. *Quoter* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.discmakers.com/quoter/default.aspx>.
25. EICHHORN, Daniel. *esp8266-oled-ssd1306* [online]. 2017-02-17 [visited on 2017-05-15]. Available from: <https://github.com/squix78/esp8266-oled-ssd1306>.
26. EISCHMANN, Jiří. *LiveUSBCreator as Primary Downloadable* [online]. 2016-04-26 [visited on 2017-05-15]. Available from: <https://fedoraproject.org/wiki/Changes/LUCasPrimaryDownloadable>.
27. ELINUX WIKI CONTRIBUTORS. *RPi Buying Guide* [online]. 2016-10-03 [visited on 2017-05-15]. Available from: [http://elinux.org/RPi\\_Buying\\_Guide](http://elinux.org/RPi_Buying_Guide).
28. ELINUX WIKI CONTRIBUTORS. *RPi Buying Links By Country* [online]. 2016-03-09 [visited on 2017-05-15]. Available from: [http://elinux.org/RPi\\_Buying\\_Links\\_By\\_Country](http://elinux.org/RPi_Buying_Links_By_Country).
29. ELMINALIRZAYEV. *AUC - AUTOMATICLY USB COPIER* [online]. 2017-03-01 [visited on 2017-05-15]. Available from: <https://github.com/elminalirzayev/auc-automaticly-usb-copier>.
30. FEDORA PROJECT. *How to create and use Live USB* [online]. 2017-03-08 [visited on 2017-05-15]. Available from: [https://fedoraproject.org/wiki/How\\_to\\_create\\_and\\_use\\_Live\\_USB](https://fedoraproject.org/wiki/How_to_create_and_use_Live_USB).

## BIBLIOGRAPHY

---

31. FEDORA PROJECT. *Ambassadors* [online]. 2017 [visited on 2017-04-30]. Available from: <https://fedoraproject.org/wiki/Ambassadors>.
32. FEDORA PROJECT. *Get Fedora* [online]. 2017 [visited on 2016-12-12]. Available from: <https://getfedora.org>.
33. FEDORA PROJECT. *Fedora graphic standards manual* [online] [visited on 2017-05-15]. Available from: [https://fedoraproject.org/w/uploads/5/5e/Logo\\_FedoraLogoGSM.pdf](https://fedoraproject.org/w/uploads/5/5e/Logo_FedoraLogoGSM.pdf).
34. FISHER, Tim. *Universal Serial Bus (USB): Definition of USB* [online]. 2016-09-01 [visited on 2017-05-15]. Available from: <https://www.lifewire.com/universal-serial-bus-usb-2626039>.
35. FLASHBAY. *Flashbay* [online]. 2017 [visited on 2017-05-15]. Available from: <http://www.flashbay.com>.
36. FLASHBAY. *USB Flash Drive Data Preloading* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.flashbay.com/services/data-preloading>.
37. FOUNDATION, Raspberry Pi. *Raspberry Pi 3 Model B* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
38. FOUNDATION, Raspberry Pi. *Raspberry Pi Products* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.raspberrypi.org/products/>.
39. GADGET FACTORY. *Papilio Wiki* [online]. 2013 [visited on 2017-05-15]. Available from: <http://papilio.cc/>.
40. GORDON, Whitson. *How to Survive Without a DVD Drive In Your Laptop* [online]. 2012 [visited on 2016-12-12]. Available from: <https://lifehacker.com/5922047/how-to-survive-without-a-dvd-drive-in-your-laptop>.
41. GRIESER, Franz. *PLA vs ABS: Filaments for 3D Printing Explained & Compared* [online]. 2016-01-14 [visited on 2017-05-15]. Available from: <https://all3dp.com/pla-abs-3d-printer-filaments-compared/>.
42. HACKERSPACES CONTRIBUTORS. *3D Printer* [online]. 2013-11-16 [visited on 2017-05-15]. Available from: [https://wiki.hackerspaces.org/3D\\_Printer](https://wiki.hackerspaces.org/3D_Printer).

43. HANRIGOU, Philippe. *In UNIX Everything is a File* [online]. 2012-03-11 [visited on 2017-05-15]. Available from: <https://web.archive.org/web/20120320050159/http://ph7spot.com/musings/in-unix-everything-is-a-file>.
44. HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. *USB 3.0 Technology: Performance Advantage on HP Workstations* [online]. 2012 [visited on 2017-05-15]. Available from: <http://www8.hp.com/h20195/v2/GetDocument.aspx?docname=4AA4-2724ENW>.
45. HOFFMANN, Gerd. *qemu usb storage emulation* [online]. 2017-01-24 [visited on 2017-05-15]. Available from: <http://git.qemu.org/?p=qemu.git;a=blob;f=docs/usb-storage.txt>.
46. HUI PAN, E. *1394 Monthly Newsletter*. Information Gatekeepers Inc. Available also from: <https://books.google.cz/books?id=fRvbxgH4wmsC>.
47. INTEL CORPORATION. *Intel® FPGAs* [online] [visited on 2017-05-15]. Available from: <https://www.altera.com/products/fpga/overview.html>.
48. JOHNSON, Dave. *What's the Difference Between USB UASP And BOT* [online]. 2012-03-18 [visited on 2017-05-15]. Available from: <http://www.electronicdesign.com/embedded/whats-difference-between-usb-uasp-and-bot>.
49. KALE, Gaurav. *Check if your USB 3.0 device supports USB Attached SCSI (UAS) Protocol* [online]. 2017-08-23 [visited on 2017-05-15]. Available from: <http://winaero.com/blog/check-if-your-usb-3-0-device-supports-usb-attached-scsi-uas-protocol/>.
50. KIVY CONTRIBUTORS. *Kivy* [online]. 2017 [visited on 2017-05-15]. Available from: <https://kivy.org/>.
51. LEE, Tony; KEMELHAR, Matt. *A Simple USB Thumb Drive Duplicator on the Cheap* [online]. 2012-07-16 [visited on 2017-05-15]. Available from: <http://blog.opensecurityresearch.com/2012/07/simple-usb-thumb-drive-duplicator-on.html>.
52. *Linux Programmer's Manual - sendfile(2)*. 2016. Available also from: <http://man7.org/linux/man-pages/man2/sendfile.2.html>.
53. LINUXDAYS. *LinuxDays 2017* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.linuxdays.cz>.

## BIBLIOGRAPHY

---

54. MEMORYSUPPLIERS.COM. *Custom USB Flash Drives & Portable Chargers* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.memorysuppliers.com/>.
55. MEMOTREK. *MemoTrek* [online]. 2016 [visited on 2017-05-15]. Available from: <http://www.memotrek.com/usb/usb-flash-drives-clip-n-easy>.
56. MICRO ROBOTICS. *Arduino Nano 5V* [online] [visited on 2017-05-15]. Available from: <https://www.robotics.org.za/arduino-nano-5v-nano3>.
57. MINIDROID. *Raspberry Pi 3 Model B 1GB* [online]. 2015 [visited on 2017-05-15]. Available from: <https://eshop.minidroid.cz/vsechno-pro-raspberry-pi/470-raspberry-pi-3-model-b.html>.
58. NEWEGG. *Intelligent 9 Series (UB910G) - GOLD Factory series 1 - 9 Target USB Flash Memory/Pen Drive/External USB Hard Drive Duplicator* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.newegg.com/Product/Product.aspx?Item=9SIA25H20W5993>.
59. NEWEGG. *TEAC 1 to 15 USB Drive Duplicator Model USBDUPLICATOR/15* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.newegg.com/Product/Product.aspx?item=N82E16827139069>.
60. NEWEGG. *VINPOWER Black 1 to 11 USBShark USB Flash Copy Tower Duplicator Model USBShark-11T-BK* [online]. 2017 [visited on 2017-05-01]. Available from: <http://www.newegg.com/Product/Product.aspx?Item=N82E16827196197>.
61. NODEMCU. *NodeMcu* [online]. 2014 [visited on 2017-05-15]. Available from: [http://nodemcu.com/index\\_en.html](http://nodemcu.com/index_en.html).
62. OPENSOURCE.COM. *What is a Raspberry Pi?* [online] [visited on 2017-05-15]. Available from: <https://opensource.com/resources/what-raspberry-pi>.
63. ORGANIZERS, DevConf.cz. *DevConf.cz* [online]. 2017 [visited on 2017-05-15]. Available from: <https://devconf.cz/>.
64. PAREKH, Sameer. *usbmK: Mass USB copier* [online]. 2013-06-06 [visited on 2017-05-15]. Available from: <https://github.com/sameerparekh/usbmK>.
65. PIMORONI. *Raspberry Pi 3* [online] [visited on 2017-05-15]. Available from: <https://shop.pimoroni.com/products/raspberry-pi-3>.



66. PLASTICS, Rex. *How Much Do Plastic Injection Molds Cost?* [online]. 2013-06-15 [visited on 2017-05-15]. Available from: <https://rexplastics.com/plastic-injection-molds/how-much-do-plastic-injection-molds-cost>.
67. PLATFORMIO. *Platform espressif8266* [online]. 2017 [visited on 2017-05-15]. Available from: <http://docs.platformio.org/en/latest/platforms/espressif8266.html>.
68. PLATFORMIO. *PlatformIO Core* [online]. 2017 [visited on 2017-05-15]. Available from: <http://docs.platformio.org/en/latest/core.html>.
69. PREMIUM USB. *Data Loading* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.premiumusb.com/usb-preloading>.
70. PYCON CZ. *PyCon CZ* [online]. 2017 [visited on 2017-05-15]. Available from: <https://cz.pycon.org/>.
71. RASPBERRY PI FOUNDATION. *Raspbian* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.raspberrypi.org/downloads/raspbian/>.
72. RASPBERRY PI FOUNDATION. *Raspberry Pi Touch Display* [online] [visited on 2017-05-15]. Available from: <https://www.raspberrypi.org/products/raspberry-pi-touch-display/>.
73. RPISHOP.CZ. *Raspberry Pi 3 Model B 64-bit 1GB RAM* [online]. 2017 [visited on 2017-05-15]. Available from: <http://rpishop.cz/kategorie/283-raspberry-pi-3-model-b-64-bit.html>.
74. RS COMPONENTS LTD. *Raspberry Pi Boards and Official Accessories* [online]. 2017 [visited on 2017-05-15]. Available from: <http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi>.
75. SADROZINSKI, H.F.W.; WU, J. *Applications of Field-Programmable Gate Arrays in Scientific Research*. CRC Press, 2016. ISBN 9781439841341. Available also from: [https://books.google.cz/books?id=5zkzM9\\_Hd9MC](https://books.google.cz/books?id=5zkzM9_Hd9MC).
76. SANQUI. *Fedorator* [online]. 2017-05-15 [visited on 2017-05-15]. Available from: <https://github.com/Sanqui/fedorator>.
77. SILENT. *Error Installing Fedora 24* [online]. 2016-09-17 [visited on 2017-05-15]. Available from: <https://unix.stackexchange.com/questions/310457/error-installing-fedora-24/310463>.

## BIBLIOGRAPHY

---

78. SNAIL INSTRUMENTS. *Teensy 3.6 without headers* [online]. 2017 [visited on 2017-05-15]. Available from: <http://www.snailshop.cz/teensy/2208-teensy-3-6-without-headers.html>.
79. SPARKFUN ELECTRONICS. *Arduino Uno - R3* [online]. 2013-01-21 [visited on 2017-05-15]. Available from: <https://www.flickr.com/photos/sparkfun/8406865680/>.
80. SPARKFUN ELECTRONICS. *Teensy 3.6* [online] [visited on 2017-05-15]. Available from: <https://www.sparkfun.com/products/14057>.
81. STACK OVERFLOW. *Stack Overflow Developer Survey 2017 - Platforms* [online]. 2017 [visited on 2017-04-30]. Available from: <https://stackoverflow.com/insights/survey/2017#technology-platforms>.
82. STOFFREGEN, Paul. *USBHost\_t36: Experiments with K66 USBHS Host Port* [online]. 2017-03-30 [visited on 2017-05-15]. Available from: [https://github.com/PaulStoffregen/USBHost\\_t36/](https://github.com/PaulStoffregen/USBHost_t36/).
83. STOFFREGEN, Paul. *Teensy 3.5 & 3.6* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.kickstarter.com/projects/paulstoffregen/teensy-35-and-36>.
84. THE PI HUT. *Raspberry Pi Store* [online]. 2017 [visited on 2017-05-15]. Available from: <https://thepihut.com/collections/raspberry-pi-store>.
85. UPTON, Eben. *Ten millionth Raspberry Pi, and a new kit* [online]. 2016 [visited on 2017-05-12]. Available from: <https://www.raspberrypi.org/blog/ten-millionth-raspberry-pi-new-kit/>.
86. USB IMPLEMENTERS FORUM, Inc. *USB Class Codes* [online]. 2016-06-15 [visited on 2017-05-15]. Available from: [http://www.usb.org/developers/defined\\_class](http://www.usb.org/developers/defined_class).
87. USB IMPLEMENTERS FORUM, Inc. *Universal Serial Bus Mass Storage Class: Bulk-Only Transport*. 2000. Available also from: [http://www.usb.org/developers/docs/devclass\\_docs/usbmassbulk\\_10.pdf](http://www.usb.org/developers/docs/devclass_docs/usbmassbulk_10.pdf).
88. USB IMPLEMENTERS FORUM, Inc. *Universal Serial Bus Specification*. 2000.
89. USB IMPLEMENTERS FORUM, Inc. *Universal Serial Bus Specification: Specification Overview*. 2000. Available also from: [http://www.usb.org/developers/docs/devclass\\_docs/Mass\\_Storage\\_Specification\\_Overview\\_v1.4\\_2-19-2010.pdf](http://www.usb.org/developers/docs/devclass_docs/Mass_Storage_Specification_Overview_v1.4_2-19-2010.pdf).

90. VERMEULEN, Sven. *Linux Sea: What is Linux?* [online]. 2009–2015 [visited on 2016-12-12]. Available from: [http://swift.siphos.be/linux\\_sea/](http://swift.siphos.be/linux_sea/).
91. W3COUNTER. *Browser & Platform Market Share* [online]. 2004-2016 [visited on 2016-12-12]. Available from: <https://www.w3counter.com/globalstats.php?year=2016&month=11>.
92. WILLIAMSON, Adam et al. *Fedora Release Life Cycle* [online]. 2017-03-17 [visited on 2017-05-15]. Available from: [https://fedoraproject.org/wiki/Fedora\\_Release\\_Life\\_Cycle](https://fedoraproject.org/wiki/Fedora_Release_Life_Cycle).
93. XILINX, Inc. *FPGAs & 3D ICs* [online]. 2017 [visited on 2017-05-15]. Available from: <https://www.xilinx.com/products/silicon-devices/fpga.html>.



---

## List of Acronyms

<b>ABS</b>	Acrylonitrile Butadiene Styrene
<b>ARM</b>	Advanced RISC Machine
<b>AVR</b>	Alf (Egil Bogen) and Vegard (Wollan)'s RISC Processor
<b>BBB</b>	Bulk/Bulk/Bulk
<b>BOT</b>	Bulk-Only Transport
<b>CAD</b>	Computer-Aided Design
<b>CAN</b>	Controller Area Network
<b>CBI</b>	Control/Bulk/Interrupt
<b>CPU</b>	Central Processing Unit
<b>CSI</b>	Camera Serial Interface
<b>CZK</b>	Czech Koruna
<b>DSI</b>	Display Serial Interface
<b>DVD</b>	Digital Versatile Disc
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>FPGA</b>	Field-Programmable Gate Array
<b>GND</b>	Ground
<b>GNU</b>	GNU's Not Unix!
<b>GPIO</b>	General-purpose Input/Output
<b>HDMI</b>	High-Definition Multimedia Interface
<b>HID</b>	Human Interface Device
<b>I/O</b>	Input/Output

## A. LIST OF ACRONYMS

---

I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
LAN	Local-Area Network
LCD	Liquid-Crystal-Display
LED	Light-Emitting Diode
MIDI	Musical Instrument Digital Interface
MSC	Mass Device Class
OLED	Organic Light-Emitting Diode
PDF	Portable Document Format
PLA	Poly(Lactic Acid)
QIC	Quarter Inch Cartridge
RAM	Random-Access Memory
RISC	Reduced Instruction Set Computing
SCSI	Small Computer System Interface
SD	Secure Digital
SFF	Small Form Factor
SoC	System on a Chip
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
STL	STereoLithography
TFT	Thin-Film-Transistor
UASP	USB Attached SCSI Protocol
USB	Universal Serial Bus
USD	United States Dollar

---

## Supplemental Material

The complete source code of this thesis and the projects described within can be found on the attached medium.

The same content is also available online on GitHub. Versions at the submission of this thesis bear the tag thesis.

<b>Thesis</b>	<a href="https://github.com/Sanqui/fedorator-thesis">https://github.com/Sanqui/fedorator-thesis</a>
<b>Fedorator</b>	<a href="https://github.com/Sanqui/fedorator">https://github.com/Sanqui/fedorator</a>
<b>Fedorator Tiny</b>	<a href="https://github.com/Sanqui/fedorator-tiny">https://github.com/Sanqui/fedorator-tiny</a>

	README.md	.....	brief description of contents
	BT_Labsky_David_2017.pdf	.....	thesis in the PDF format
	thesis/	.....	source of thesis in X <sub>Y</sub> -L <sup>A</sup> T <sub>E</sub> X
	fedorator/	.....	repository for Fedorator
		src/	.....
		case/	.....
	fedorator-tiny/	.....	repository for Fedorator Tiny
	photos/	.....	photos of Fedorator, courtesy of Marek Žehra

**Directory structure B.1:** Contents of the attached medium