

UNIVERSIDAD DE



VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍAS ESPECÍFICAS DE
TELECOMUNICACIÓN, MENCIÓN EN TELEMÁTICA

**Wireless adaptation for a standalone CNC
controller**

**Draadloos uitbreiden van een stand-alone
CNC-besturingseenheid**

Autor:

Daan Ver Eecke

Tutor de UVa:

Dr. Jesús Arias Álvarez

Tutor de Vives (Brugas, Bélgica):

Mr. Tom Cordemans

Valladolid, Junio 2019

TITLE: **Wireless adaptation for a standalone CNC controller**

AUTHOR: **Daan Ver Eecke**

TUTOR: **Dr. Jesús Arias Álvarez**

DEPARTMENT: **Department of Electronics**

COURT

CHAIRMAN: **Dr. Martín Jaraíz Maldonado**

MEMBER **Dr. Jesús Arias Álvarez**

SECRETARY: **Dr. Jesús M. Hernández Mangas**

SUBSTITUTE: **Dr. Ruth Pinacho**

SUBSTITUTE: **Dr. Pedro López**

DATE: **21 of June 2019**

QUALIFICATION:

Preface

With this bachelor's thesis I finish my three years as an electronics student at the university of applied sciences Vives in Bruges, Belgium. A thesis is the perfect way to prove that I can apply the knowledge I have gained over the past three years in a large project. I would like to thank a number of people who helped me to complete this thesis as well as the people who have supported me in the past years.

First of all I want to thank my two mentors from the University of Valladolid, Dr. Jesús Arias Álvarez and Dr. Jesús M. Hernández Mangas. If I encountered a problem during my thesis, I could always count on you and your experience.

I would also like to thank my internship promoter, Mr. Tom Cordemans. Thank you for the support during my Erasmus. If I had a problem with administration during my Erasmus, I could always count on you. Without you it would not have been possible to go on Erasmus to Valladolid.

In addition, I would like to thank all my teachers who have taught me a lot in the past three years. The tips and tricks were very useful during the development of this thesis.

I also want to thank my classmates and friends for all the wonderful memories of the past three years. You provided the necessary relaxation during this period. A special thanks to Elias Verstappe for sharing with me the Erasmus experience.

Ultimately, I want to thank my parents who support me in everything I do. Since high school you encourage me to do my best for school and I want to thank you for that. I also want to thank you for supporting me in my Erasmus experience, this was a very educational experience.

Abstract

English

Togheter with Elias Verstappe I worked on this thesis. We are two students at the university of applied sciences Vives in Bruges, Belgium. This thesis has been worked out at the University of Valladolid in Spain where we are both on Erasmus. A Bachelor's thesis aims to solve a specific problem. In our case this is a problem which is related to electronics because we are both students electronics.

The aim of this thesis is to control an old CNC machine with an electronic system designed by Elias and I. As technological developments follow each other in rapid succession in recent years, some outdated devices are no longer supported by the latest control systems.

Dr. Jesús Arias Álvarez and Dr. Jesús M. Hernández Mangas, both professors at the University of Valladolid, suggested the following problem: They currently have an old CNC machine that is controlled by an old MS-DOS computer. Our assignment is to control the CNC machine on the basis of a self-designed control unit. The CNC machine must also be capable of being controlled via a website. To replace the functionality of the old MS-DOS computer, we designed a printed circuit board ourselves. You can control the machine in two different ways: on one hand everything can be operated by means of a touchscreen, on the other hand the CNC machine can be fully controlled on the basis of a website.

Now that the project is finished, I can conclude that the functionality of a program on an old computer can be applied fairly easily in a website. The only additional costs that the website entails are the working hours of the person who develops the website. Once the WiFi module has been purchased and the website has been developed, the website does not incur any additional costs.

This thesis shows that many old machines can be controlled via a printed circuit board and can also be controlled by a website. This method could be applied to other machines where newer technology is desired.

Dutch

Samen met Elias Verstappe heb ik deze thesis uitgevoerd. Wij zijn twee studenten aan de Hogeschool Vives in Brugge, België. Deze thesis is uitgewerkt aan de Universiteit van Valladolid in Spanje waar wij beide op Erasmus zijn. Een bachelor proef heeft als doel om een praktisch en concreet probleem op te lossen in de praktijk. In ons geval is dit een probleem met betrekking tot de afstudeerrichting elektronica.

Deze thesis heeft als doel om een oude CNC machine aan te sturen met een door ons ontwikkeld elektronisch systeem. Aangezien de laatste jaren technologische ontwikkelingen zich snel na elkaar opvolgen, worden sommige verouderde machines niet meer ondersteund door de nieuwste besturingssystemen.

Dr. Jesús Arias Álvarez en Dr. Jesús M. Hernández Mangas, beide professoren aan de universiteit van Valladolid, hebben ons volgend probleem voorgesteld: Momenteel beschikt men over een oude CNC-machine die aangestuurd wordt door een oude MS-DOS computer. Onze opdracht is om aan de hand van een zelf ontworpen besturingseenheid de CNC-machine aan te sturen. De CNC-machine dient ook aangestuurd te kunnen worden via een website. Om de functionaliteit van de oude MS-DOS computer te vervangen hebben we zelf een printplaat ontworpen. Het besturen van de machine kan op twee verschillende manieren: enerzijds kan alles bediend worden door middel van een touchscreen, anderzijds kan de CNC-machine volledig bestuurd worden aan de hand van een website.

Nu dat het project afgelopen is kan ik besluiten dat de functionaliteit van een programma op een oude computer vrij eenvoudig in een website toegepast kan worden. De enige meerkost die de website met zich meebrengt zijn de werkuren van de persoon die de website ontwikkeld. Na het aanschaffen van de wifimodule en de ontwikkeling van de website, brengt de website geen extra kosten teweeg.

Deze thesis toont aan dat veel oude machines aanstuurbaar zijn via een printplaat alsook via een website. Deze werkwijze zou kunnen toegepast worden op andere machines waar een nieuwere technologie gewenst is.

Contents

1	Introduction	8
1.1	Definition of terms	8
1.2	Old situation vs new situation	8
1.2.1	Old situation	8
1.2.2	New situation	9
1.3	Description of the chapters	9
2	Hardware	10
2.1	General	10
2.2	ESP8266	12
2.2.1	ESP 8266 specifications	12
2.2.2	Schematic	13
2.2.3	Arduino	14
2.2.4	Void setup	14
2.2.5	Void loop	15
3	Website	16
3.1	General	16
3.2	Web pages	17
3.2.1	Login page	17
3.3	Responsive web pages	23
3.3.1	Explanation responsive website	23
3.3.2	Example responsive routepro/drillpro page	25
3.4	Word wide web	28
3.4.1	General	28
3.4.2	Router	29
3.4.3	Fixed IP address	29
3.4.4	Port forwarding	30
3.4.5	DDNS	30
3.5	Composition of the website	32
3.5.1	HTML	32
3.5.2	CSS	32
3.5.3	JavaScript	32
3.6	Webserver ESP8266	33
3.6.1	Handling web pages	33
3.6.2	Receiving data	34
3.6.3	Receiving move buttons	35
3.6.4	Serial communication	36
3.6.5	Frame structure	38
3.6.6	Payload structure	38

4	Case	39
4.1	Process 3D printing	39
4.1.1	Tinkercad	39
4.1.2	Ultimaker cura	40
4.1.3	Anycubic I3 Mega	41
4.2	Top	42
4.3	Bottom	43
4.4	Emergency button	44
4.5	Slide switch	45
5	Cost	46
5.1	Total cost	46
5.2	Cost WiFi module	48
6	Conclusion	49
	Bibliography	49
	Appendices	50
A	Schematics PCB	51
B	Photographs case	58
C	Website source code	61
C.0.1	Welcome web page	61
C.0.2	Routepro/Drillpro web page	65
C.0.3	Configuration web page	71
C.0.4	Tools web page	76
C.0.5	Offset web page	85
C.0.6	Load web page	91
D	Arduino source code	96
D.0.1	Libraries and variables	96
D.0.2	Void setup	98
D.0.3	Void loop	98
D.0.4	Handle web pages	99
D.0.5	Read data	101
D.0.6	Serial communication	102
D.0.7	General functions	104

List of Figures

1.1	Old situation	8
1.2	New situation	9
2.1	Picture PCB	10
2.2	Foto ESP8266[3]	12
2.3	Pinout ESP8266	13
2.4	Screenshot of the arduino IDE	14
3.1	Web server schematic	16
3.2	Login page	17
3.3	Routeopro/Drillpro web page	18
3.4	Configuration web page	19
3.5	Load web page1	20
3.6	Load web page2	20
3.7	Offset web page	21
3.8	tools web page	22
3.9	Routeopro/drillpro page seen from computer	25
3.10	Routeopro/drillpro page seen from a tablet	26
3.11	Routeopro/drillpro page seen from a phone	27
3.12	Network diagram	28
3.13	Linksys WRT54G/GL/GS[12]	29
3.14	Fixed IP address	29
3.15	Port forwarding	30
3.16	No-IP.com	30
3.17	DDNS settings on router	31
3.18	Handling web pages flowchart	33
3.19	Receiving data flowchart	34
3.20	Movement buttons flowchart	35
3.21	Sending data flowchart	36
3.22	UART frame structure	38
3.23	UART payload structure	38
4.1	Screenshot tinkercad	39
4.2	Screenshot Ultimaker cura	40
4.3	Photo anycubic I3 mega	41
4.4	Case top	42
4.5	Case bottom	43
4.6	Emergency stop	44
4.7	Slide switch	45
B.1	Photograph case top	58
B.2	Photograph case bottom	59
B.5	Photograph case mounted	59
B.3	Photograph emergency stop	60
B.4	Photograph slide switch	60

Abbreviations

Abbreviation	explanation
WiFi	Wireless Fidelity
MS-DOS	MicroSoft Disk Operating System
CNC	Computer Numerical Control
USB	Universal Serial Bus
PCB	Printed Circuit Board
IC	Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
IO	Input/Output
bps	Bits Per Second
TxD	Transmit Data
RxD	receive data
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
WAN	Wide Area Network
LAN	Local Area Network
IP	Internet Protocol
DDNS	Dynamic Domain Name Server
MAC	Media Access Control
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
NAT	Network Address Translation
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SGML	Standard Generalized Markup Language
CRC	Cyclic Redundancy Check
3D	three-dimensional
STL	stereolithography
PLA	polylactide

Chapter 1

Introduction

1.1 Definition of terms

This final settlement was an exam. The comments made during the defence were not included. This bachelor's thesis was in collaboration with Elias Verstappe .We are two Belgian students from university of applied sciences Vives, campus Bruges Station. We study electronics, which is part of the applied sciences and technology field of study. Mr. Tom Cordemans is our mentor at the university of Bruges, Belgium . Our two mentors from the University of Valladolid are Dr. Jesús M. Hernández Mangas and Dr. Jesús Arias Álvarez.

1.2 Old situation vs new situation

1.2.1 Old situation

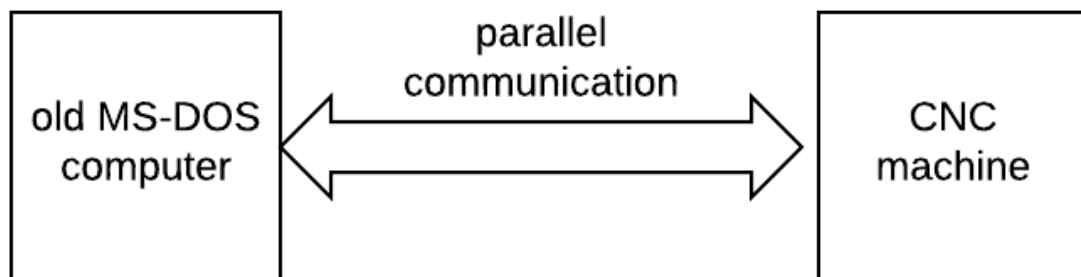


Figure 1.1: Old situation

The old situation is fairly simple, an old computer that communicates with the CNC machine via a parallel connection. This computer has two programs, 'drillpro' and 'routepro'. The 'drillpro' program is only able to drill holes in the PCB. The 'routepro' program can only mill traces. This old MS-DOS computer works quite slowly and needs to be replaced. That's why Elias Verstappe and I have worked out a new situation.

1.2.2 New situation

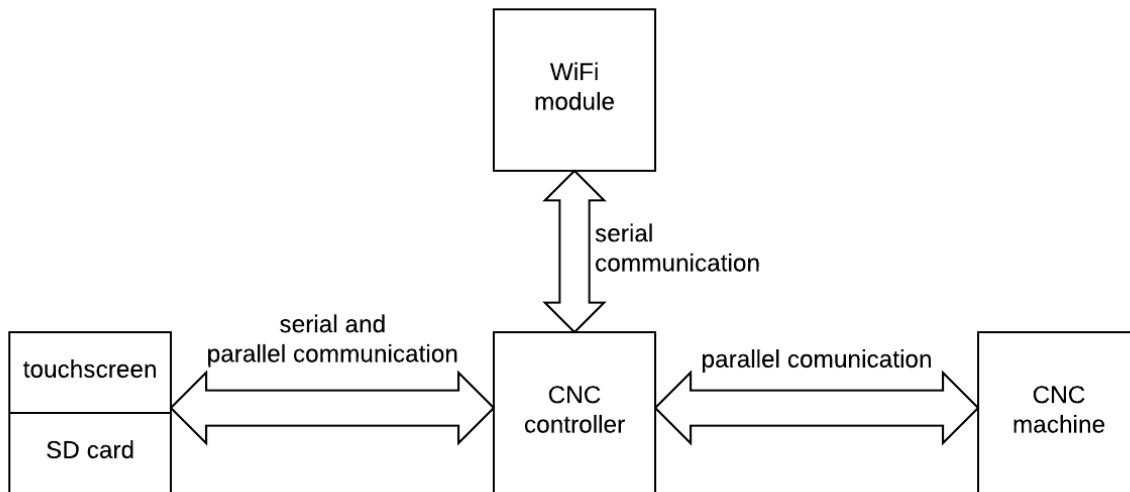


Figure 1.2: New situation

In the new situation, the old MS-DOS computer has been replaced by a printed circuit board with a microcontroller on it. This microcontroller replaces all the functionality of the MS-DOS computer and adds a number of extra features. Just as in the old situation, the communication between the CNC controller and the CNC machine happens via parallel communication. With the new situation, there are two possible ways to interact with the CNC machine. The first way is via a touch screen, this touch screen ends up on top of the circuit board. the touch screen also has an SD card slot, through this slot we can upload a file to the microcontroller, which will then control the CNC machine. The communication between the touch screen, SD card slot and the microcontroller is both serial and parallel. The second way to communicate with the CNC machine is via a WiFi module. This WiFi module will host a website, which has the same functionality as both programs on the old MS-DOS computer. In this study I will explain the WiFi module / website part of our thesis. A general explanation of the hardware follows in the next chapter. If you want a more detailed explanation about the printed circuit board or touch screen, I am happy to refer you to Elias his study on the hardware of our thesis.

1.3 Description of the chapters

- chapter 1 : general description of the project
- chapter 2 : description of the hardware
- chapter 3 : description of the website
- chapter 4 : description of the case
- chapter 5 : cost calculation
- chapter 6 : conclusion

Chapter 2

Hardware

2.1 General

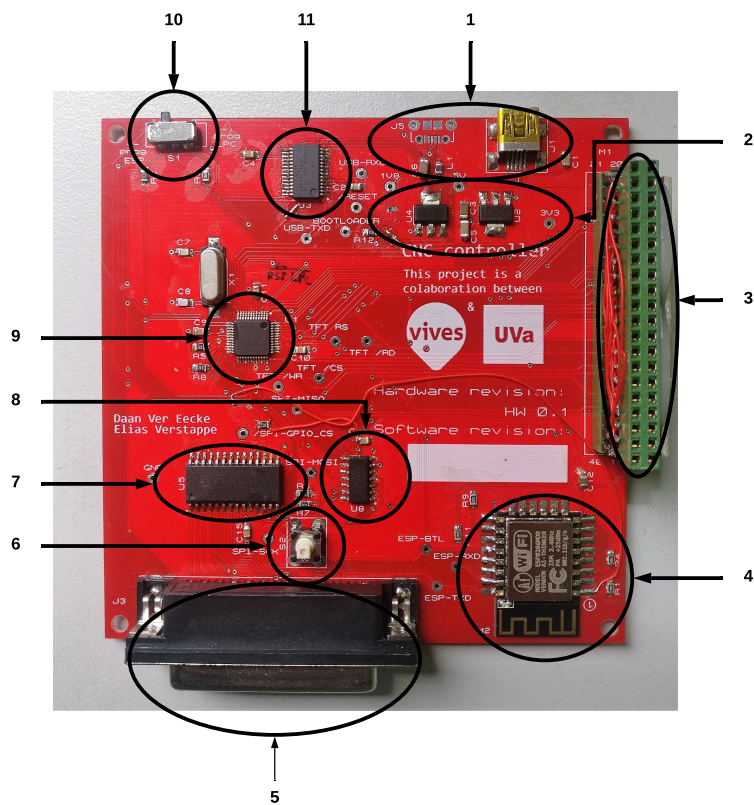


Figure 2.1: Picture PCB

This chapter briefly explains the hardware of this project and which components are used. This is explained on the next page using numbering on the photo above. The schematics of the PCB can be found in Appendix A at the back of this study. This appendix also contains the power dissipation analysis and heat dissipation analysis, if you want more information about this I am happy to refer you to Elias his study.

1: These are two USB connectors, on the right a mini-USB connector and on the left a micro-USB connector. The PCB is powered on the basis of these connectors. The LPC and the ESP8266[1][2] can be programmed through these connectors. As you can see the micro USB is not mounted, this is not necessary since the PCB can work without the micro USB

2: There are three different voltage levels on the PCB. The PCB is powered via USB, the voltage with which the PCB is powered is five volts. To achieve the other two voltage levels, there are two voltage regulators on the PCB. These two voltage regulators can be seen in circle number two on the photo above. The first converter converts 5 volts to 3.3 volts and the second converter converts 3.3 volts to 1.8 volts. These two components convert voltages and therefore these components can get quite hot, therefore these components are connected to a copper surface to better transfer the heat.

3: This is the 40 pin header. On this connector the touch screen is mounted. The touch screen is the interface with the user, on the touch screen the user can set a number of settings and upload a file via an SD card.

4: This component is the WiFi module, the ESP8266. This component is explained a little further below as this thesis is mainly about this ESP8266.

5: This is a 25 pin SUB-D connector, this connector is the interface between the microcontroller and the CNC machine.

6: This push button serves as an emergency stop. This is very important because if a dangerous situation should arise, it must be possible to stop the machine safely. This ensures that the machine cannot be damaged and people cannot be hurt.

7: The IC U5 is a 16 bit IO expander, this IC is controlled by the LPC2106 via SPI. Since all IO pins of the microcontroller were busy and a number of things still needed to be controlled, we needed an IC to make this possible. The 16 bit IO expander makes it possible to control an additional 16 IO pins via SPI.

8: There was a problem with SPI communication, the microcontroller operates on a voltage of 3V3 and the IO expander operates on a voltage of 5V, so we need a level shifter to make the SPI communication possible. This application uses an IC 74HCT04D as a level shifter, this is a NOT gate. By passing a SPI signal through two NOT gates, the signal is inverted twice and therefore no longer inverted, this IC converts the signal from 3V3 to 5 volts which makes it possible for the microcontroller to communicate with the IO expander via SPI.

9: The microcontroller is located halfway down the printed circuit board, this component will control all other components and is therefore one of the more important components. On the left above the microcontroller is the crystal (14.7456 MHz), this determines the clock frequency for the microcontroller, in our application the internal clock frequency is 58.9824 MHz.

10: At the top left of the PCB you will see a slide switch, through this switch the user can choose whether the LPC or the ESP8266 will be programmed. If the switch is to the left, the ESP8266 will be programmed, if the switch is to the right, the LPC will be programmed. The switch also serves to activate the communication between the LPC and the ESP8266. when the slide switch is slided to the left, the ESP8266 can communicate with the ESP8266. The LPC has two serial interfaces, 'Serial0' interface is used to program the LPC, the 'Serial1' interface is used to communicate with the ESP8266. The ESP8266 has only one serial interface so programming and communicating happens with the same serial interface.

11: This is a USB to UART converter. This component makes it possible to program the two microcontrollers via a computer.

2.2 ESP8266

2.2.1 ESP 8266 specifications

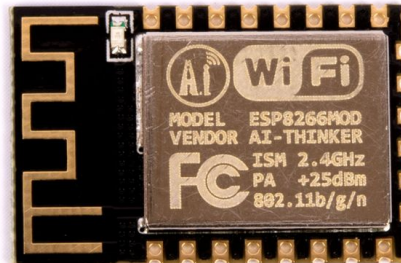


Figure 2.2: Foto ESP8266[3]

The figure above shows the ESP8266[2], which has the following specifications. On the figure above you can also see that the ESP8266[1][4] works within the 2,4 GHz band and uses the protocol 802.11 b/g/n.

Description	Parameters
Protocols	802.11 b/g/n (HT20)
Frequency range	2,4G - 2,5G
CPU	Tensilica L106 32-bit processor
Peripheral interface	UART/SDIO/SPI/I2C/...
Operating voltage	2,5V - 3,6V
WiFi mode	Station/SoftAP/SoftAP+Station
Security	WPA/WPA2
Encryption	WEP/TKIP/AES
Network Protocols	IPv4, TCP/UDP/HTTP

2.2.2 Schematic

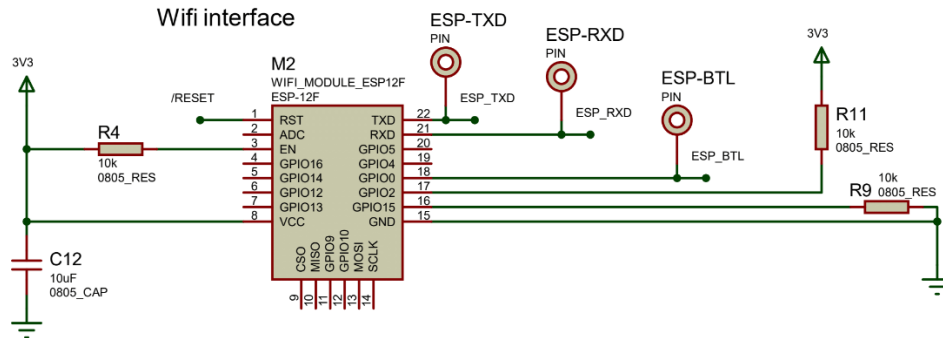


Figure 2.3: Pinout ESP8266

This section explains how the ESP8266 is connected and how the communication between the ESP8266 and the microcontroller works. The figure above shows how the ESP8266 is connected. The ESP8266 is supplied with a voltage of 3.3 volts. The reset pin of the ESP8266 is connected to the reset pin of the USB to UART converter. All components can be reset using this reset signal. The enable pin is connected to the 3V3 via his own 10 kilo ohm resistor. The enable pin must of course be connected to the 3V3 otherwise the esp8266 would not work. Pins 22 and 21 are responsible for communication with the microcontroller, pin 22 is the TxD and pin 21 is the RxD. Pins 18, 17 and 16 are explained a bit further in this section since these 3 pins are more important. To prevent us from getting floating voltages, the microcontroller's ground must of course be connected to the ground. All the pins that are not described here are not used. All these pins could not be connected without consequences.

GPIO15	GPIO0	GPIO2	mode	explanation
L	H	H	flash	boot from SPI flash
L	L	H	UART	program via UART
H	x	x	SDIO	boot from SD-card

The table above shows three different modes[5] for the ESP8266. In our application pin 16 is connected to the ground via a pulldown resistor, which means that the third mode in the table can never occur in our application. Pin 17 is connected via a pull up resistor to the VCC, this means that the signal 'ESP_BTL' determines in which mode the ESP8266 will be. If the signal 'ESP_BTL' is high, the ESP8266 will load the program that is in the SPI flash. To load a program via serial communication, the 'ESP_BTL' signal must be low.

2.2.3 Arduino

Arduino[6][7] is an open source computer platform intended to make microcontrollers simple. With Arduino it is possible to create devices and objects that respond to their environment through digital and analogue input signals. The Arduino can be programmed with the Arduino IDE. This software makes it easy to write your own code and upload it to an Arduino. Libraries are available for well-known tasks, such as connecting to Wi-Fi. It is also possible to download libraries, or write your own library to reduce repetitive code. All the Arduino source code can be found in Appendix D at the back of this study. This appendix is divided into different parts to keep it organized. Below you will see a screenshot of the arduino IDE.

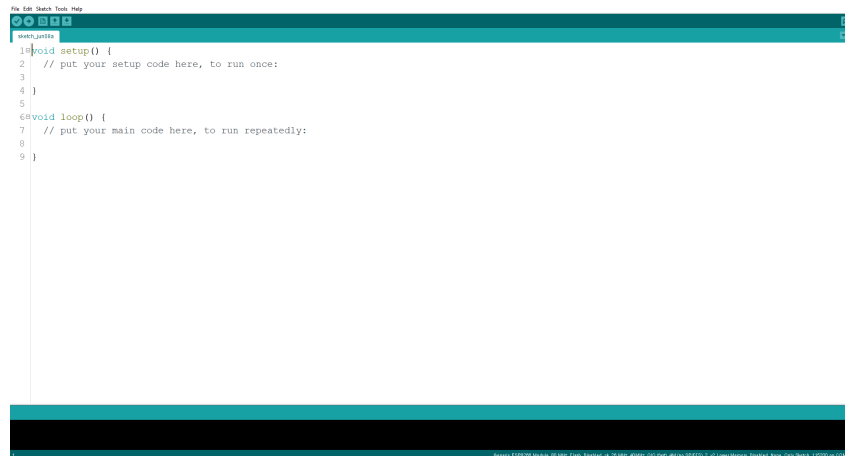


Figure 2.4: Screenshot of the arduino IDE

As you can see on the photo on the previous page, an empty arduino project contains two functions. An arduino project always contains these two functions. These two functions are explained in the following subsections

2.2.4 Void setup

The 'setup()' functions will only run once, after each powerup or reset of the Arduino board. This function is used to initialize variables, pin modes and to include libraries. Below you can see the setup function of this project.

```
1 void setup ( void ) {  
2   Serial.begin ( 115200 );  
3   WiFi.mode ( WIFI_STA );  
4   WiFi.begin ( ssid, password );  
5  
6   server.on("/Login", handleLogin);  
7   server.on ( "/", changeWebPage );  
8   server.on("/changeWebPage", changeWebPage);  
9   server.on("/handleData", handleData);  
10  server.on("/handleToolsData", handleToolsData);  
11  server.on("/handleButtonsPressed",handleButtonsPressed);  
12  
13  server.onNotFound ( handleNotFound );  
14  server.begin();  
15  Serial.println ( "HTTP server started" );  
16 }
```

In computer programming, when void is used as a function return type, it indicates that the function does not return a value. The setup function only initializes things so it makes sense that this function

should not return a value. The first thing that happens is initializing the serial communication between the ESP8266 and the LPC, this at a baud rate of 115200 bps. Then the WiFi is initialized with the correct ssid and password. After that a number of 'server.on()' functions follow, these functions process all http requests that arrive on the ESP8266. For example if the http request '/Login' is received, the 'server.on("/Login",handleLogin)' function will be triggered. This function will in turn call the 'handleLogin' function and this function will then process the http request. The 'server.on ()' function on line 13 is executed if there is a problem with the server. Then the setup function will boot the server and this is the last thing that happens in the setup function.

2.2.5 Void loop

After creating a setup() function, which initializes and sets the initial values, the loop() function is created. This function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Below you can see the loop function of this project.

```
1 void loop ( void ) {  
2     server.handleClient();  
3     checkSerialDataAvaibleToSend();  
4 }
```

The loop function is probably the simplest function of the entire program, this function only contains two lines of code. On the first line is the 'server.handleClient()' function, this function processes everything that has to do with the website, such as switching between web pages and adjusting values on the website. The second function in the loop function checks whether data has changed on the website. If the data is changed, this function will cause the data to be sent to the LPC2106.

Chapter 3

Website

3.1 General

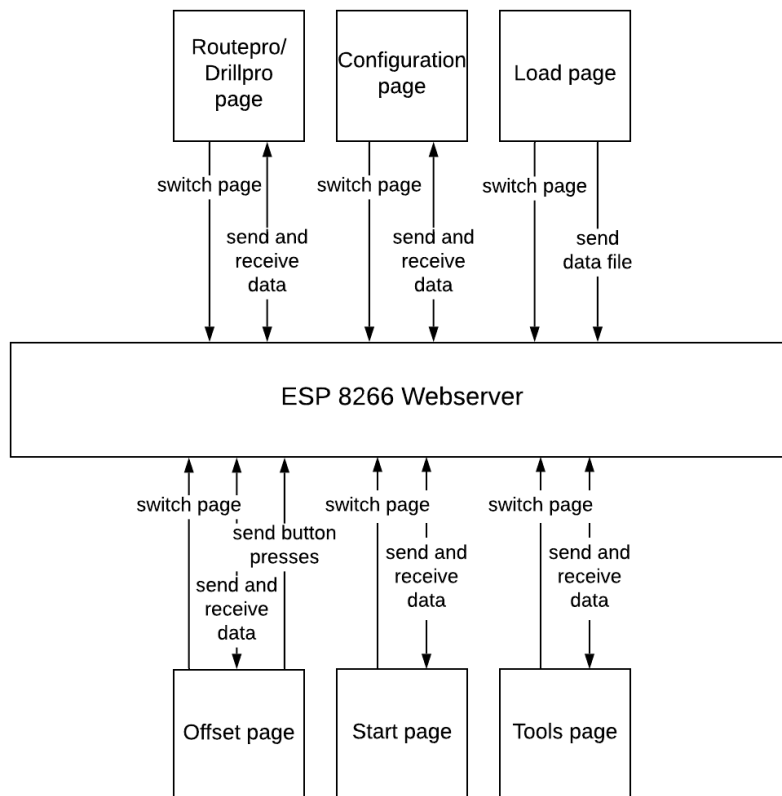


Figure 3.1: Web server schematic

The old MS-DOS computer has two programs. The program called ‘routepro’ can mill traces. The second program called ‘drillpro’ can only drill holes. The functionality of both programs are bundled together into a website. In total the website consists of 6 different web pages. On the figure above you can see every web page as well as the possible HTTP request between the web server and every web page. Every web page has an HTTP request that takes care of switching between web pages. If a user presses a button, an HTTP request will be sent to change the web page. The ‘Routepro/Drillpro’, ‘Configuration’, ‘Start’ and ‘Tools’ have as second type of HTTP requests the sending and receiving of data, this HTTP request is very important because they keep the web pages up to date with the most recent values for each variable. The ‘Load’ page has a second type of HTTP request and is responsible for sending a file from the website to the ESP8266. The ‘Offset’ page has 3 types of HTTP requests, one for changing web pages, the second for sending and receiving data. The ‘offset’ page has a number of buttons with which the offset of the machine can be adjusted. If one of these buttons is pressed, the web page will generate an HTTP request and send it to the ESP8266 so that the offset of the machine can be set.

3.2 Web pages

3.2.1 Login page

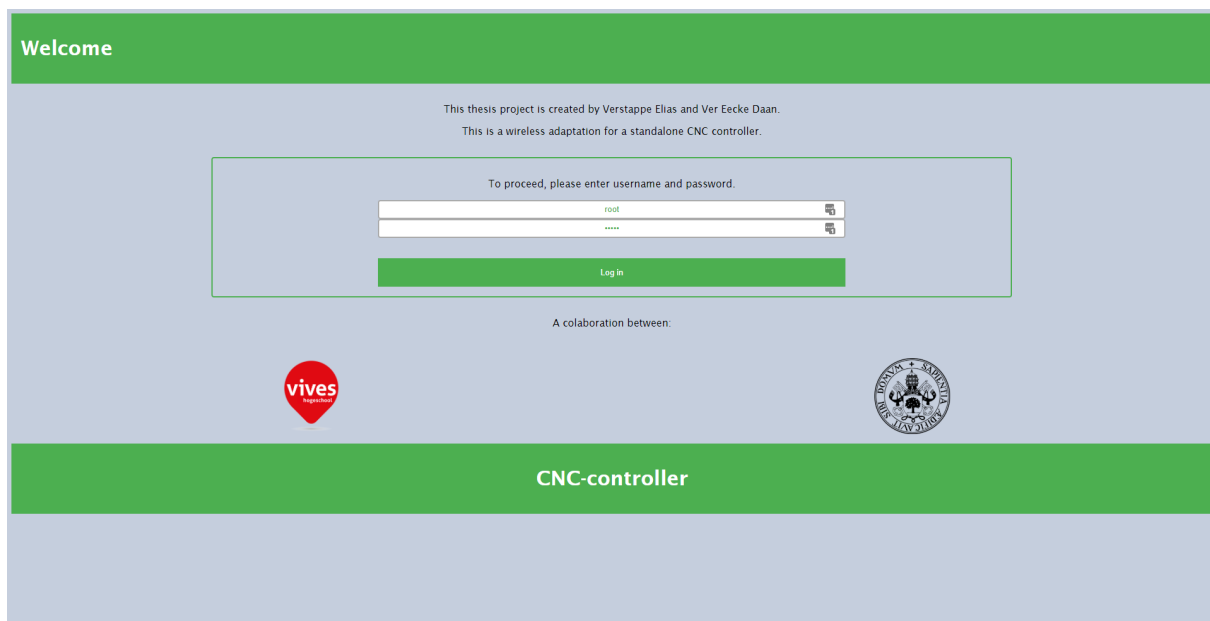


Figure 3.2: Login page

The screenshot above is the begin screen of the website. If a user surfs to ‘cnccontroller.ddns.net’, the user will see this screen. On this screen you will see a small description about the functionality of the website and who created the website. Below that you will see a login form, this ensures that not just anyone can use the website. Just like in many technical applications, the standard username and password are ‘root’ and ‘admin’. By clicking on the ‘login’ button with the correct username and password entered, the user is redirected to the ‘routepro/drillpro’ webpage. Under the login form you see two logos. On the left you see the logo of the University VIVES, on the right you see the logo of the University of Valladolid.

The credentials entered by the user are sent via HTTP to the ESP8266. The username and password are therefore sent in clear text, this is not safe and easily intercepted by hackers. A solution would be to use HTTPS[8]. This is an extension to the HTTP protocol with the aim of secure data exchange. Due to time constraints, it was not possible to add HTTPS to the project.

Routepro/Drillpro page

The screenshot shows the 'Routepro/drillpro' web page. It features a green header with the page title. On the left, there is a vertical navigation menu with buttons for 'Config', 'Tools', 'Offset', 'Load', and 'Save'. The main content area contains two tables. The first table is a configuration grid with columns for 'Offset', 'Clip', and 'Vectors'. The second table is a summary table with columns for 'Count' and values. A green footer bar contains the text 'CNC-controller for Bungard CNC-machine'.

	Offset:	Clip:		Vectors:	
X:	<input type="text" value="0"/>	<input type="text" value="325"/>	from	<input type="text" value="0"/>	
Y:	<input type="text" value="0"/>	<input type="text" value="278"/>	to	<input type="text" value="0"/>	
	Board:	Base:	0	< X <	0
Z:	<input type="text" value="1.5"/>	<input type="text" value="2.0"/>	0	< Y <	0

Count :	
Total	0
Done	0

CNC-controller for Bungard CNC-machine

Figure 3.3: Routepro/Drillpro web page

This is the screen on which the user ends up after logging in. This is a central web page where the most important parameters can be set. If the PCB is placed on the work surface with an offset, then the user can enter this offset on the basis of 'X-offset' and 'Y-offset'. 'X-clip' en 'Y-clip' restricts the range of valid coordinates. On the basis of 'vector' input boxes the user can choose to redo a certain part of a design. The thickness of the PCB can be adjusted with the 'Z-board' field. If the PCB is on a raised platform, the height can be entered in the 'Z-base' field. In the second table, a number of values are displayed related with the selected file. 'Total' shows the number of commands to be executed, 'Done' shows the number of executed commands and 'File' shows the file name. On the left side of the web page you will see five buttons, through which the user can navigate through the website. All values given are in millimeters

Configuration page

Configuration	
Load Save Back	
Coordinates: ...	Chord angle: 95
Correction: X: 0.9975 Y: 0.9975 Z: 0.8988	Stroke: Limit: 32.78 Free: 03.00
Spindle: Delay: 0.5s	
CNC-controller for Bungard CNC-machine	

Figure 3.4: Configuration web page

This is the configuration web page. On this web page you can also set a number of parameters. ‘stroke-free’ describes the height that the tool has when ‘flying’ over the board while moving to the next position. The height above the bed is determined on the basis of ‘stroke-limit’. Having the correct setting there is essential for the machine to calculate the correct height over the machine bed and the board. Circles and arcs are converted into a chain of vectors with adjustable ‘chord angle’. The correction values are fixed values and take into account mechanical wear of the machine. The spindle delay is also a fixed value and this value shows that the spindle will only start running half a second after starting the machine.

Load page

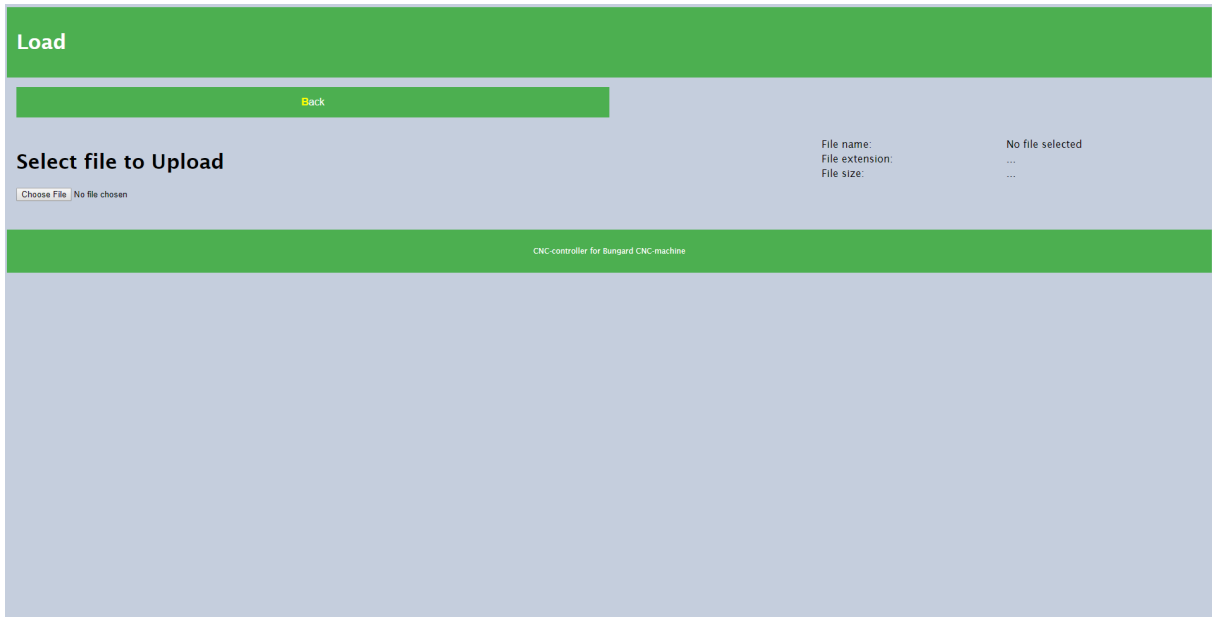


Figure 3.5: Load web page1

Via this page you can upload a file to the website which will forward the file to the WiFi module (ESP8266). If no file has been selected yet, the page will look like the screenshot above. Once a file is selected, the web page will look like the screenshot below. A start button and an emergency button will appear on the web page. The machine can be started via the start button, the machine can be stopped in the event of an emergency via the emergency stop. The file name, file extension and file size will also be displayed via JavaScript. The example below shows the file '123' with extension 'plt' which is 7 bytes in size.



Figure 3.6: Load web page2

Offset page

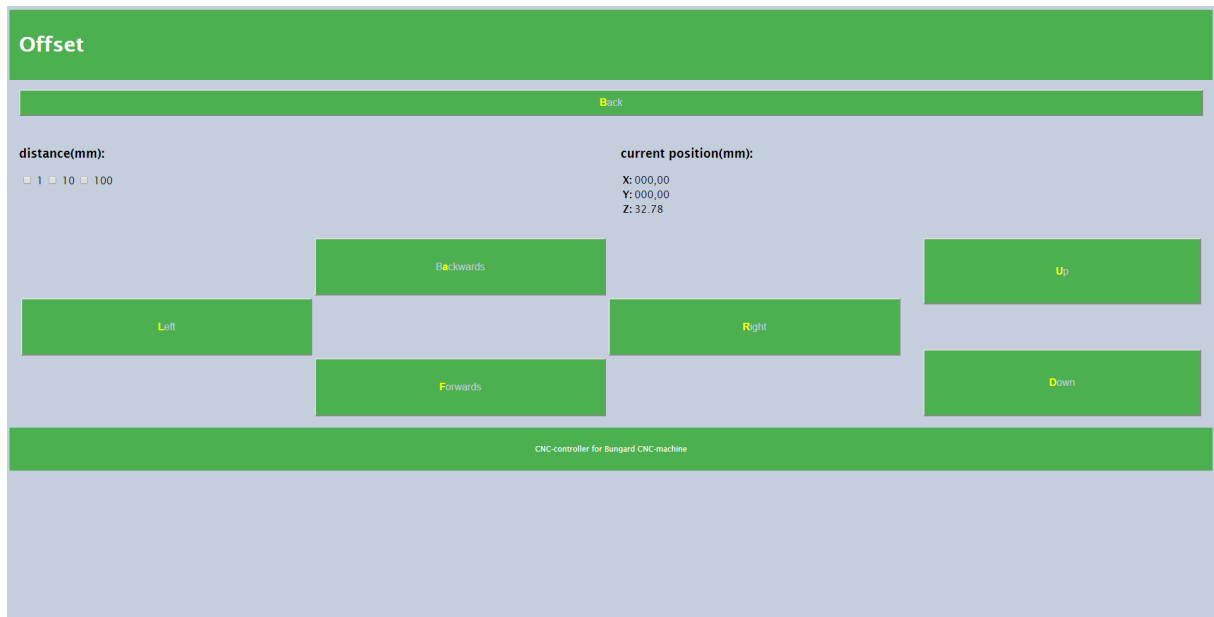


Figure 3.7: Offset web page

The CNC machine has a horizontal working plane that is determined by X and Y coordinates. The home position of the horizontal work plane is $X = 0$ and $Y = 0$. If a PCB is placed on the horizontal work surface, it may be that the home position of the work surface does not match the home position of the PCB. This problem can be solved with the offset page. The user can give an X and Y offset to the CNC machine so that these home positions nevertheless match. If the PCB is placed on a raised base, then the Z offset will not be correct either. The user can solve this by setting the correct Z offset in the offset page. If the user wishes to consult the offset page, a warning screen will first be displayed, because the CNC machine will move to its home position if the machine is not there. This is a safety procedure so that the user sees that the machine will move, to prevent accidents from happening. For each axis there are two buttons, for the Z-axis an 'up' and 'down' button, for the X-axis a 'forward' and a 'backward' button and for the Y-axis a 'left' and 'right' button. These buttons allow the user to set the offset for each axis, the machine will also move so that the user can check whether the entered offset is correct. The user can set the size of the desired step on the basis of a selection box. There are three possible steps: 1, 10 and 100. If the user wants to leave the offset page, a warning box will be displayed again, because the machine will move to its home position again, so this is the same safety procedure as before.

Tools page

PL	Depth mm	Iter. /n	X Feed mm/min	Z Feed mm/min	Spindle 1/min	Start Vect.	Used mm	Tab.
01	00.00	01	0050	0050	30000
02	00.00	01	0050	0050	30000
03	00.00	01	0050	0050	30000
04	00.00	01	0050	0050	30000
05	00.00	01	0050	0050	30000
06	00.00	01	0050	0050	30000
07	00.00	01	0050	0050	30000
08	00.00	01	0050	0050	30000
09	00.00	01	0050	0050	30000
10	00.00	01	0050	0050	30000
11	00.00	01	0050	0050	30000
12	00.00	01	0050	0050	30000
13	00.00	01	0050	0050	30000
14	00.00	01	0050	0050	30000
15	00.00	01	0050	0050	30000

Figure 3.8: tools web page

The tools page has a table showing all the tools that the machine has. A number of parameters are displayed per tool. The page has 15 different tools and each tool has its own name. The depth, iteration, X feed, Z feed, spindle, start, used and tabulation are all shown in the table. ‘Depth’ determines how deep the routing of the traces will be. When a hard material needs to be processed, iteration is to be used. Based on this setting, a certain operation is repeated a number of times, but each time the tool will mill a little deeper. ‘X-feed’ describes how fast the jobs are milled in mm / min, ‘Z-feed’ describes how fast the tool will move vertically. By changing the ‘Spindle’ value, the user can determine at what speed the tool will rotate.

3.3 Responsive web pages

3.3.1 Explanation responsive website

Responsive[9][10] web design is an approach to web design where the web developer strives for an optimal web experience for a wide range of devices (from desktop computer screens to mobile phones). The design of a responsive website scales with the dimensions of a screen, without compromising on readability of text or usability of the user interface. The viewport is the user's visible area of a web page. The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen. With the code below we set the viewport for a webpage.

```
1 <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Many web pages are based on a grid-view, which means that the page is divided into columns. A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window. First we must calculate the percentage for one column: $100\% / 12 \text{ columns} = 8.33\%$ Then we make one class in CSS for each of the 12 columns. On the picture you see the 12 classes.

```
1 .col-1 {width: 8.33%;}
2 .col-2 {width: 16.66%;}
3 .col-3 {width: 25%;}
4 .col-4 {width: 33.33%;}
5 .col-5 {width: 41.66%;}
6 .col-6 {width: 50%;}
7 .col-7 {width: 58.33%;}
8 .col-8 {width: 66.66%;}
9 .col-9 {width: 75%;}
10 .col-10 {width: 83.33%;}
11 .col-11 {width: 91.66%;}
12 .col-12 {width: 100%;}
```

Now we can make a row in the HTML code. In each row the numbers of columns should be equal to 12, as shown in the example below.

```
1 <div class="row">
2   <div class="col-3">...</div> <!-- 25% -->
3   <div class="col-9">...</div> <!-- 75% -->
4 </div>
```

The previous steps only makes the website responsive for a computer screen, if we want to make the website responsive for smaller devices we have to add other classes to specify the screen layout. Below these classes are showcased.

```
1 /* For mobile phones: */
2 [class*="col-"] {
3     width: 100%;
4 }
5
6 @media only screen and (min-width: 600px) {
7 /* For tablets: */
8     .col-s-1 {width: 8.33%;}
9     .col-s-2 {width: 16.66%;}
10    .col-s-3 {width: 25%;}
11    .col-s-4 {width: 33.33%;}
12    .col-s-5 {width: 41.66%;}
13    .col-s-6 {width: 50%;}
14    .col-s-7 {width: 58.33%;}
15    .col-s-8 {width: 66.66%;}
16    .col-s-9 {width: 75%;}
17    .col-s-10 {width: 83.33%;}
18    .col-s-11 {width: 91.66%;}
19    .col-s-12 {width: 100%;}
20 }
```

Now we also have to make a small adjustment to the HTML so that the site is also displayed correctly on a smaller screen. As you can see below, there are three columns within one row. When we visit this page from a computer, the three divs will be the same width since they are all three equally wide, namely four columns. However, when we visit this page via a mobile phone, the three divs are 12 columns wide, so the divs will be displayed below each other.

```
1 <div class="row">
2     <div class="col-4 col-s-12">
3         ...
4     </div>
5     <div class="col-4 col-s-12">
6         ...
7     </div>
8     <div class="col-4 col-s-12">
9         ...
10    </div>
11 </div>
```

3.3.2 Example responsive routepro/drillpro page

In this subsection we will see the routepro/drillpro page three times but each time with a different screen ratio. The first part will display the route pro / drillpro page on a computer, the second part via a tablet and the third part via a mobile phone.

Routepro/drillpro page on computer screen

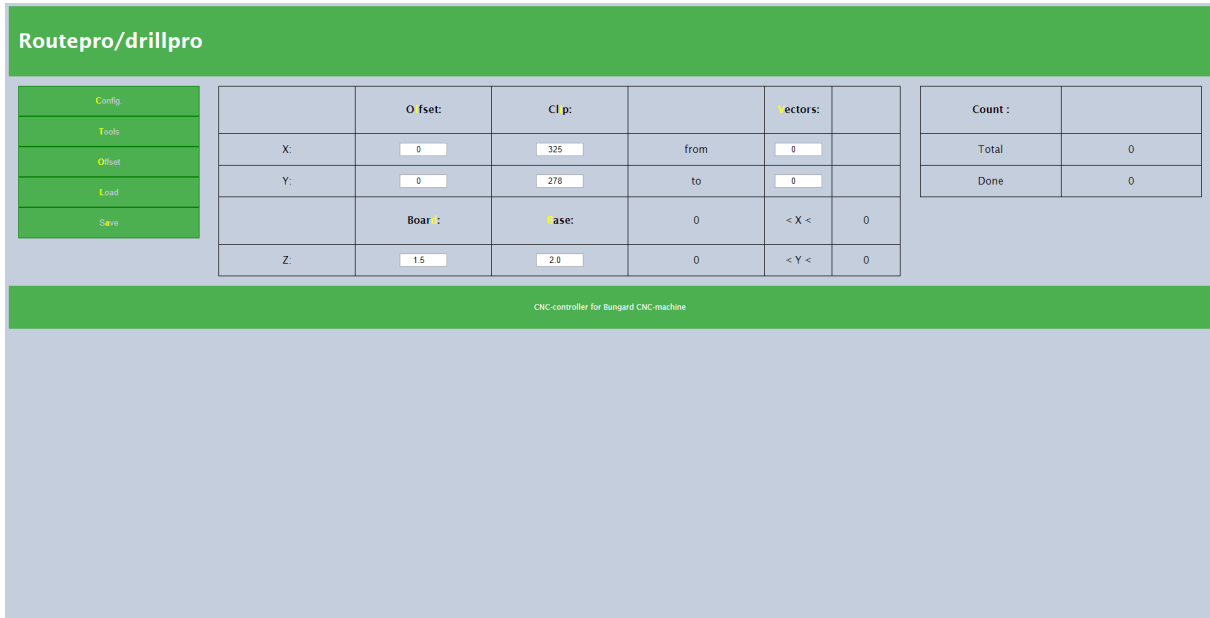


Figure 3.9: Routepro/drillpro page seen from computer

This is how the routepro/drillpro page will be displayed when a user surfs to this page on a computer. As you can see, the middle row is split into 12 columns. The buttons are two columns wide, the first table is 8 columns wide and the last table is 4 columns wide, so there are twelve columns in total.

Routepro/drillpro page on tablet screen



Figure 3.10: Routepro/drillpro page seen from a tablet

If a user surfs to this webpage from a tablet, the screen depends on how the user is holding his tablet. If the user holds his tablet horizontally, the layout of the web page will be almost the same as the layout seen from a computer. However, if the user holds his tablet vertically, the layout will be a bit different. As you can see, everything is still in the same place, but everything is somewhat narrower. We also see that there is much more free space at the bottom.

Routepro/drillpro page on phone a tablet

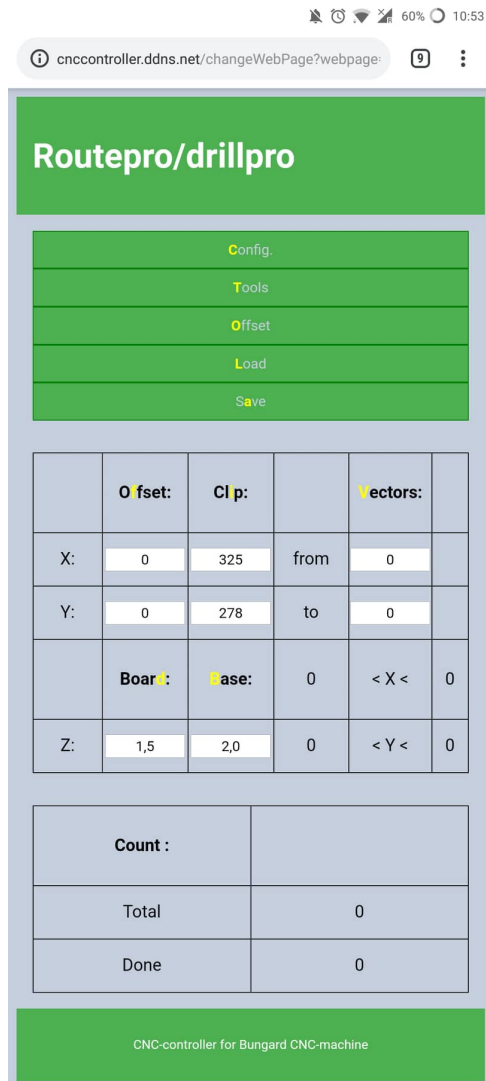


Figure 3.11: Routepro/drillpro page seen from a phone

As you can see, the layout seen from a mobile phone is completely different from previous layouts. For clarity this image is a 'scrolling screenshot', this to show the entire web page. If a user surfs to this page via his mobile phone, everything will be displayed up to and including the first table, by scrolling the user can view the rest of the web page. In this layout the three different columns of the previous layouts are displayed below each other. You will also see that every 'column' will now take up the entire width of the screen.

3.4 Word wide web

3.4.1 General

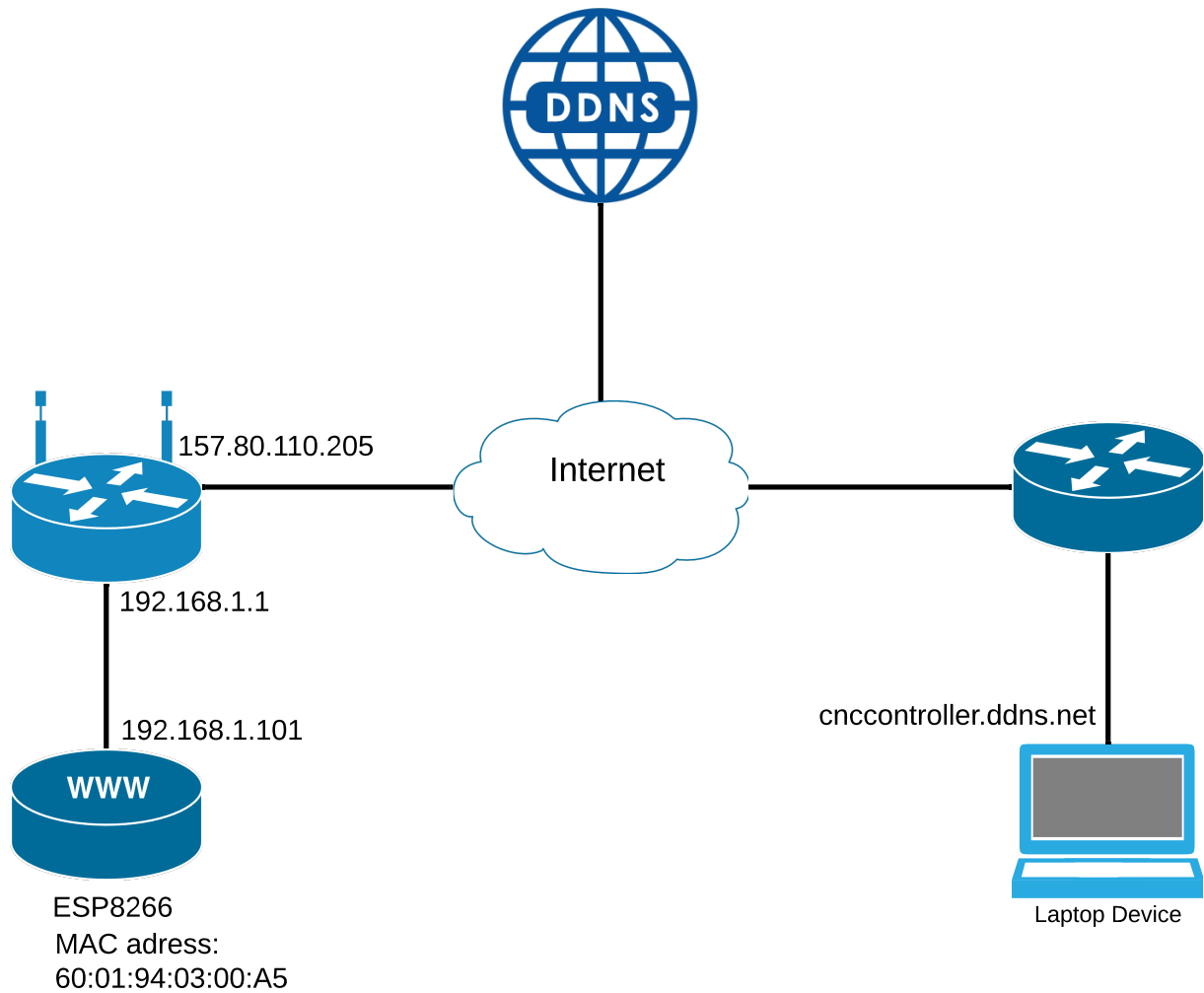


Figure 3.12: Network diagram

Above you see the network diagram. The WiFi module located on the PCB will connect to the router in the classroom. This router is connected to the internet. If a computer on an external network surfs to the URL 'cnccontroller.ddns.net', then using a DDNS[11] service this URL will be converted to the WAN IP address of the router in the classroom. The following subsections explain which settings must be made to make the website available on the internet.

3.4.2 Router



Figure 3.13: Linksys WRT54G/GL/GS[12]

The photo above shows the router that is located in the classroom, namely the Linksys WRT54G/GL/GS. To make the website available on the internet, three settings must be set in the router. The first thing to do is to set a fixed IP address for the ESP8266, this is done using the MAC address of the ESP8266. Then port forwarding is switched on and then DDNS is set. The following three sections explain these settings step by step.

3.4.3 Fixed IP address

The first step to get the website on the internet is to assign a fixed IP address to the ESP8266. To assign a fixed IP address to the ESP8266, we need the MAC address of the ESP8266. In the Arduino we need the WiFi library to connect the ESP8266 to a router, this library also has a function that returns the mac address of the ESP8266. The mac address of the ESP8266 on our printed circuit board is equal to 60:01:94:03:00:A5. Below is the arduino code that returns the mac address of the ESP8266

```
1 Serial.print("MAC: ");  
2 Serial.println(WiFi.macAddress());
```

After we get the mac address we can set a fixed IP address for the ESP8266 in the router. Now the router will ensure that the ESP8266 will always get the same IP address within the network. Below you see how the setting of a fixed IP address is done on the router

Static Leases		
MAC Address	Host Name	IP Address
60:01:94:03:00:A5	ESP8266	192.168.1.101

[Add](#) [Remove](#)

Figure 3.14: Fixed IP address

3.4.4 Port forwarding

After setting the fixed IP address, the port forwarding is set. Port forwarding or port mapping is the forwarding of a TCP or UDP package in a NAT gateway to connect an internal network to an external network (for example the internet) by means of port numbers. This technique is used to enable communication between external hosts and other services within a LAN. Port forwarding makes it possible to make a connection to a specific computer within a private network (LAN) via an external computer. A router is the connection between a local network and the worldwide web. That is why a router has two network interfaces, an internal interface and an external interface. The external IP address of the router is in the classroom 157.88.110.205. A HTTP request is typical on port 80, so if we enable port forwarding for the router, we will enable it for port 80. On the photo below you can also see that we enable it for both TCP and UDP, this is to be sure that there will be no problems. To be specific, if an external HTTP request arrives at port 80 of the router, the router will forward this package to the ESP8266 at port 80. The photo below shows how the port forwarding is set up on the router.

Forwards

Application	Port from	Protocol	IP Address	Port to	Enable
ESP8266	80	Both ▾	192.168.1.101	80	<input checked="" type="checkbox"/>

Figure 3.15: Port forwarding

3.4.5 DDNS

DDNS stands for dynamic DNS or dynamic domain name system. A DDNS is a service that converts internet domain names to IP addresses. A DDNS service makes it possible to control your home computer from anywhere in the world via a domain name and not having to type in an IP address. Two things must happen to get a DDNS working, log in to a DDNS service and choose a domain name, set the router up for DDNS. I have chosen 'no-IP.com' as the DDNS service. Below you can see how I have configured the DDNS service on the no-IP website.

Hostname Information

Hostname:

cnccontroller.ddns.net

Hostname Type

- DNS Hostname (A) ⓘ
- DNS Hostname (Round Robin) ⓘ
- DNS Alias (CNAME) ⓘ
- Port 80 Redirect ⓘ
- Web Redirect ⓘ

IP Address ⓘ

157.88.110.205

Figure 3.16: No-IP.com

As you can see on the photo on the previous, the domain name is equal to 'cnccontroller.ddns.net'. If a user surfs to this domain name, the DDNS will convert this domain name to the external IP address of the router. The photo below shows what needs to be configured on the router to use one DDNS. First you have to select the correct DDNS service, then log in with your name and password and then enter the correct domain name.

Dynamic Domain Name System (DDNS)	
DDNS	
DDNS Service	No-IP.com ▼
User Name	daan.vereecke.dv@gmail.com
Password
Host Name	cnccontroller.ddns.net
Options	
Force Update Interval	10 (Default: 10 Days)

Figure 3.17: DDNS settings on router

After completing all these steps, the website is available on the world wide web. The website is accessible via the domain name 'cnccontroller.ddns.net'.

3.5 Composition of the website

The site is built on three different programming languages, namely HTML[13], CSS[14] and JavaScript[15]. HTML is responsible for the content of the website, CSS takes care of the layout of the website and JavaScript takes care of the functionality of the website.

3.5.1 HTML

HyperText Markup Language (abbreviated HTML) is an SGML-based markup language for the specification of documents, primarily intended for the worldwide web. It is the default markup language for web pages. HTML documents primarily contain the semantic structure and textual content of a web page. You can also use HTML for the layout of a web page, but nowadays CSS is preferred for the layout of a web page. HTML provides a number of very simple user interactions such as clicking on references, filling in text fields, clicking on images, An HTML page is made up of nested elements, which usually have a start label and end label. The most common tags are:

HTML : indicate beginning and end of the HTML document.

HEAD : indicate the header of the document that contains metadata such as the page title.

BODY : contains the concrete content of the document.

DIV: defines a logical section of the webpage.

3.5.2 CSS

Cascading Style Sheets (abbreviated to CSS) is responsible for the layout of a website. The term ‘cascading’ refers to the possibility of inheriting layout properties. There are three possible ways to link a CSS file to an HTML file, inline CSS, internal CSS and external CSS. During the initial period of designing the website, external CSS was chosen because this is the most comprehensive way of working. For every web page there was a separate HTML file as well as a separate CSS file. During the testing of the ESP8266 it turned out that it does not support external CSS files. The solution was to use internal CSS instead of external CSS. Internal CSS is literally in the HTML file between the following symbols: <style> and </style>.

3.5.3 JavaScript

JavaScript (JS for short) is a widely used scripting language to make web pages interactive and develop web applications. The script is transferred to the web browser by means of HTML and is executed in it. There were two options to make the website interactive, JavaScript or PHP. Javascript is used in our project since JS is a client side language. This means that the source code is processed by the user’s web browser instead of the web server. This ensures that the server is less stressed and has more power for other purposes. PHP is a server-side scripting language and this would put more strain on the server.

3.6 Webservice ESP8266

The source code for each web page can be found at the end of this study in Appendix C. This appendix is split into a number of parts, each web page has its own part.

3.6.1 Handling web pages

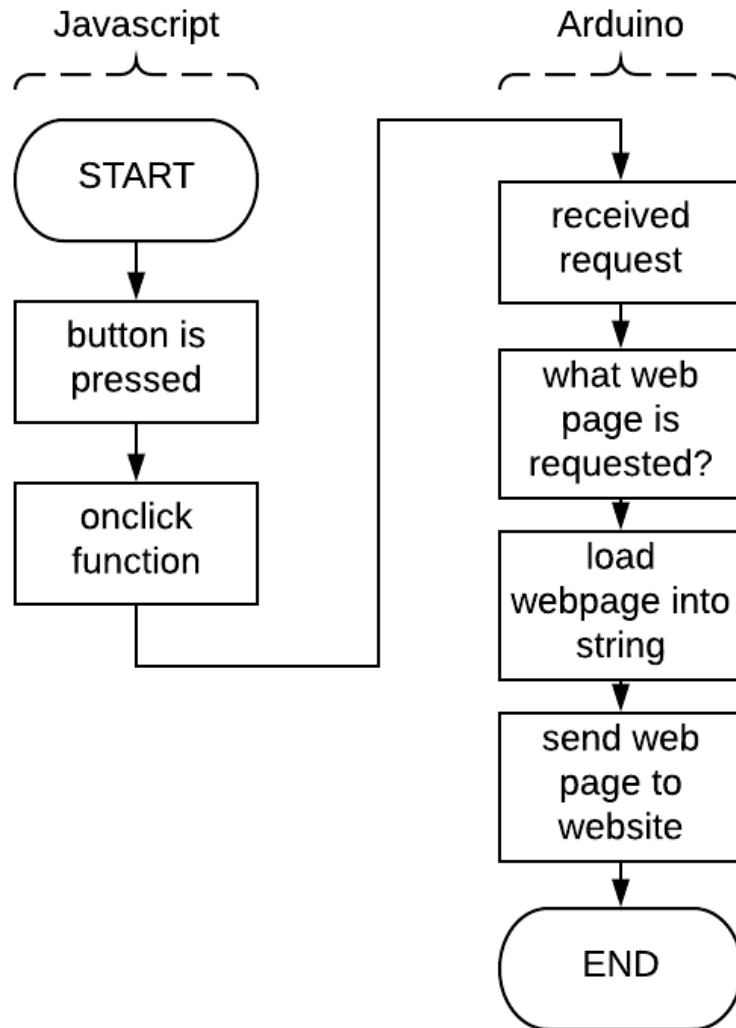


Figure 3.18: Handling web pages flowchart

The figure above contains the flowchart for switching web pages. Like all websites on the internet, the user can switch web pages using buttons. Every button has the JavaScript 'onclick' function, this function will generate an HTTP request to change web pages. This HTTP request will be received by the WiFi module. A switch case in the arduino code is used to find out which web page was requested. Once the requested web page is known, the web page is stored in a string, then this string is sent to the website with the 'server.send()' function. The website will receive this string and display it as a web page.

3.6.2 Receiving data

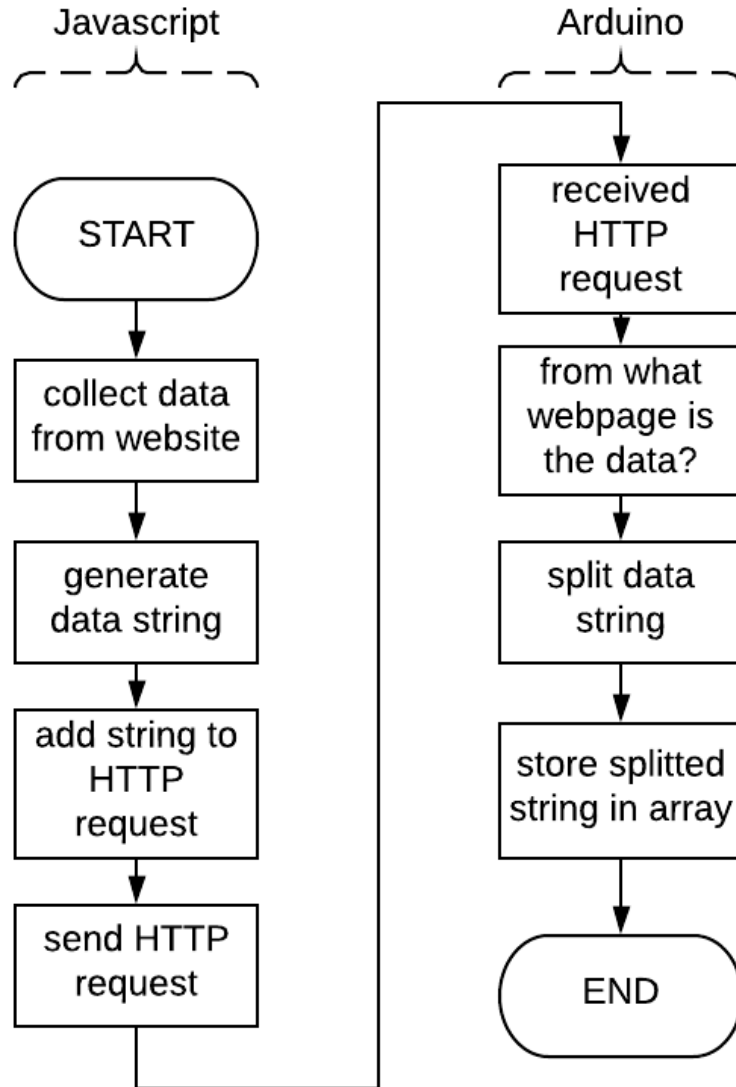


Figure 3.19: Receiving data flowchart

The figure above contains the flowchart for sending data from the website to the wifi module. Just like with previous flowcharts, a piece of JavaScript is needed here as well as arduino code. If the user changes from web page, The website will store the values and forward them to the ESP8266, following the following steps. First the webpage collects all data and stores it in an array, then a data string is generated. the data string is added to an HTTP request and then the HTTP request is sent to the esp8266. If the esp8266 receives an HTTP request, it will first identify the type of HTTP request, in this case an HTTP request that sends data. Then it is checked which web page has sent the data, in order to save the data correctly. First the data is temporarily stored in a string and then, based on a split character, the data is stored in an array. The arduino function that processes the data will also call the function that sends data serially to the microcontroller. This is because in this way the microcontroller remains well up to date with the changes on the website. The arduino function that transmits serial data to the microcontroller is explained in one of the following sections.

3.6.3 Receiving move buttons

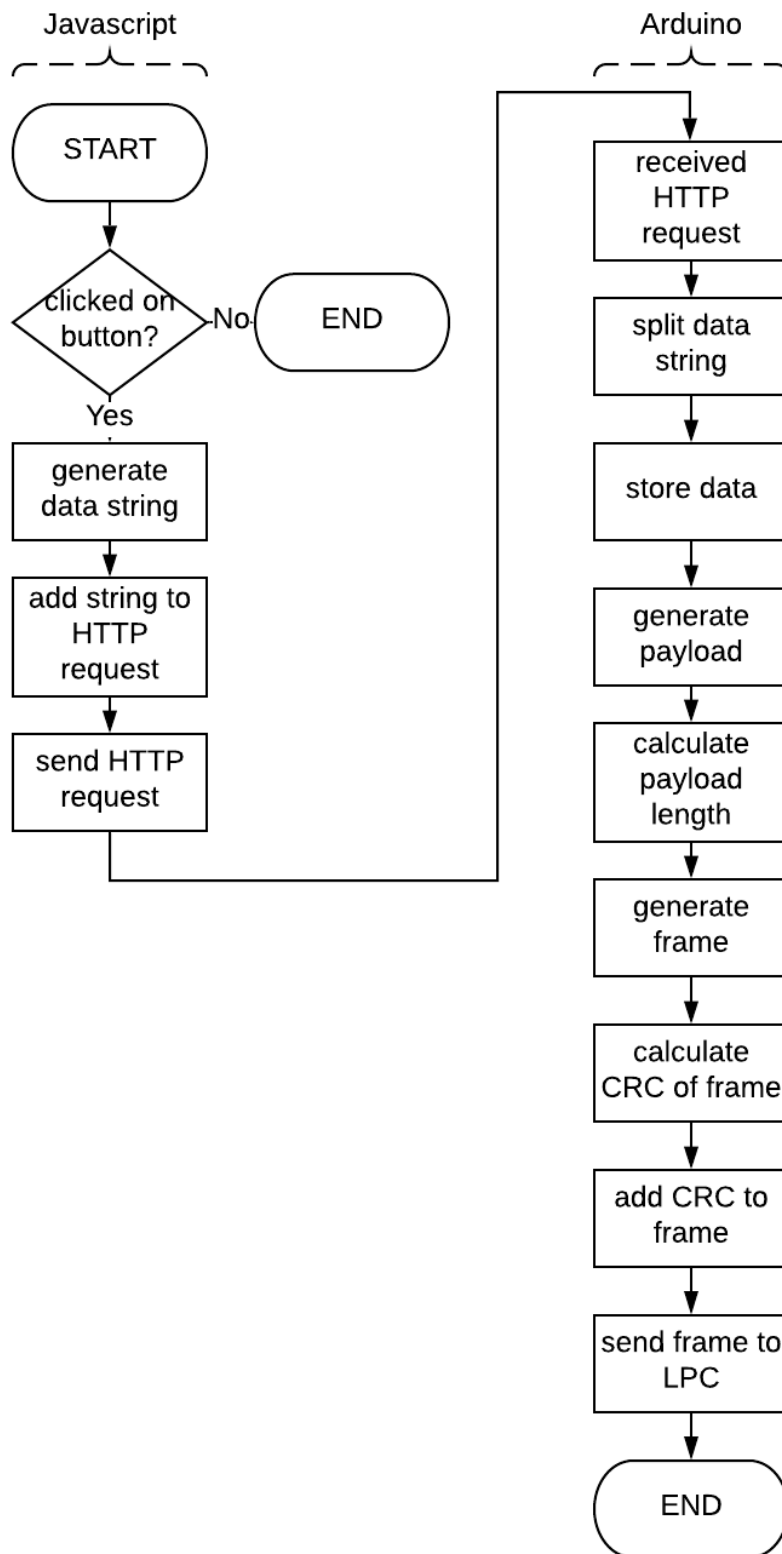


Figure 3.20: Movement buttons flowchart

The offset page has 6 buttons with which the user can set the offset of the CNC machine. The flow chart on the previous page shows how the offset of the machine can be controlled. If the user presses one of the six buttons, a data string will be generated. This data string contains the direction in which the cnc machine must move and the distance that has to be covered. This distance can be set on the website using a checkbox, the distance can be three values, namely 1,10 and 100. If no distance is checked and the user presses one of the six buttons, no data will be sent to the cnc machine and nothing will happen. After the data string is generated, an HTTP request is created and the data is added to this HTTP request, after which the HTTP request is sent to ESP8266. If the ESP8266 receives an HTTP request, the HTTP request is checked to see what type of HTTP request it is, in this case an HTTP request that contains data about the offset of the CNC machine. After the HTTP request has been checked, the data string is split into the different data values and stored in the ESP8266. Once the data has been saved, the data is immediately sent to the microcontroller via serial communication. First a payload is created and the length is calculated of this payload. A frame is then created. Once the frame is formatted, a CRC is calculated from the frame and added to the frame, then the frame is sent to the microcontroller, after which it will in turn process the data and control the CNC machine.

3.6.4 Serial communication

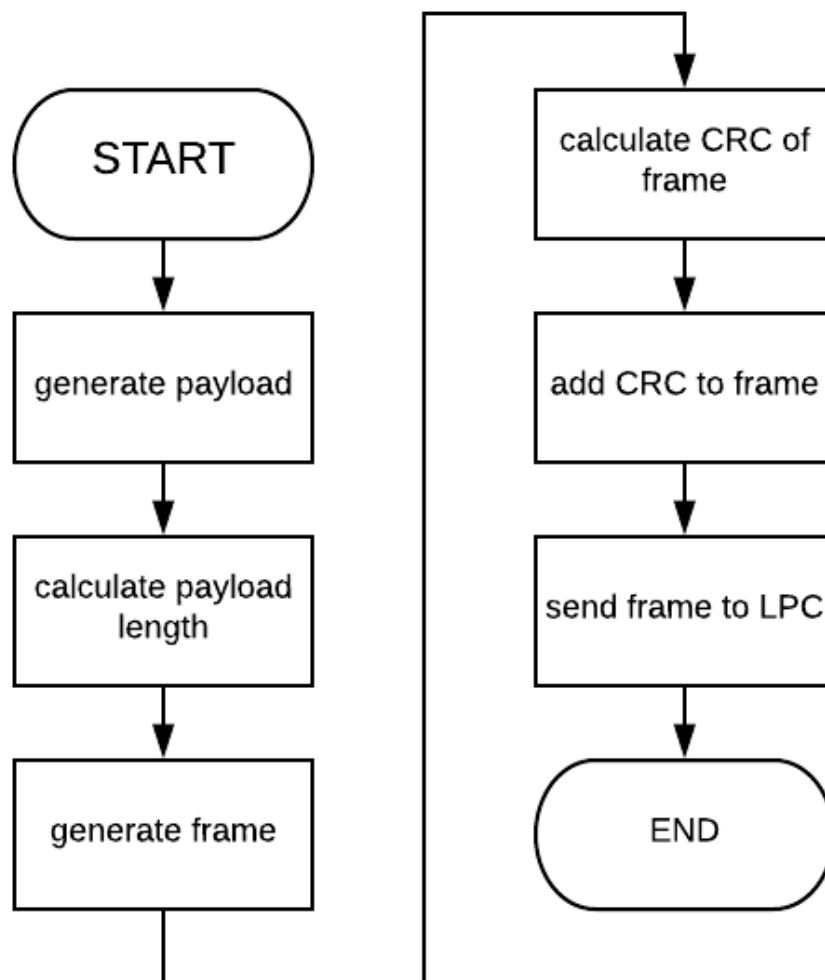


Figure 3.21: Sending data flowchart

If data updates occur on the website, the data must also be updated in the ESP8266 as well as in the microcontroller. Saving data in the ESP8266 is done on the basis of HTTP request, as already explained. Once the values are updated in the ESP8266, the values are sent to the microcontroller based on serial communication. The flowchart diagram above shows the method used for sending data via serial communication. This method is used for every website except for sending the file and some other exceptions. First a payload will be created that will contain all data from a certain web page, then the length of this payload will be calculated. Then the frame is set up that will be sent to the ESP8266. Once the frame is set up, a CRC is calculated on this frame. The CRC is added to the frame and sent to the microcontroller. The microcontroller will process the data and in some cases the microcontroller will control the CNC machine.

3.6.5 Frame structure

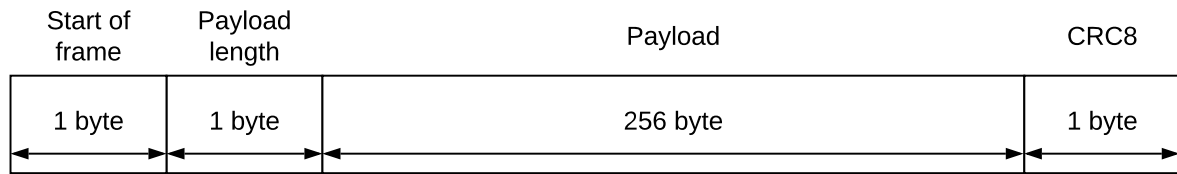


Figure 3.22: UART frame structure

The figure above shows the structure of a UART frame. The first byte is a start character, in this case the start character is a '\$'. On the basis of a start character the receiver can know when a frame starts, everything that the receiver receives that does not start with the start character is ignored. The second byte indicates the length of the payload, since this is indicated by a byte the maximum length of the frame can be 256, this because $2^8 = 256$. This is followed by the payload, which is a maximum of 256 bytes. The structure of the payload is explained in the following subsection. The last byte of a frame is a CRC. A CRC can be used to calculate whether the frame has been received correctly. CRC8 is used here, the '8' points to the fact that a CRC8 is 8 bits long, which is equal to 1 byte. 'CRC-8-CCITT' is used in this application, this CRC8 has a polynomial $x^8 + x^2 + x + 1$, which corresponds to '0x07' hexadecimal.

3.6.6 Payload structure



Figure 3.23: UART payload structure

The figure above shows the structure of the payload of a UART frame. The first byte indicates what the data is about, if this byte is equal to '0xC3' then the data contains information about a screen, if the first byte is equal to '0xC5' then the data contains information about a file. Since it was not possible to process a file via the website, this option is not further explained here. In case the data contains information about a screen, the second byte will indicate which screen the data is about. The table below shows which value corresponds to which screen.

screen	value
configuration screen	1
routepro/drillpro screen	2
tools screen	3
load screen	4
offset screen	5

The effective payload, namely the data, starts from the third byte. This data can be up to 254 bytes in length.

Chapter 4

Case

4.1 Process 3D printing

Designing and creating a 3D housing consists of various steps that must be completed. In this section these steps are explained one by one. Photos of the case can be found in Appendix B at the back of this study.

4.1.1 Tinkercad

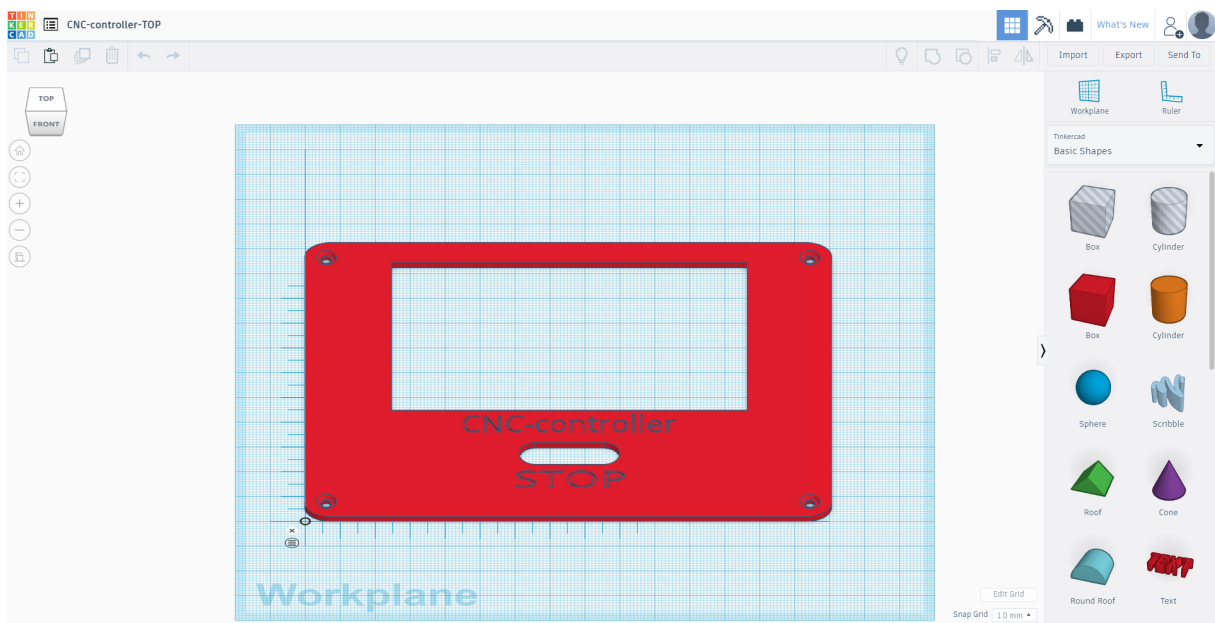


Figure 4.1: Screenshot tinkercad

The first step is to design a 3D case. The figure above shows a screenshot of the tinkercad[16] program. Tinkercad is a free and user friendly web application made by Autodesk[17]. On the basis of a number of 3D objects that are freely adaptable, it is fairly easy to make a simple housing. Our housing consists of a number of different parts, the screenshot above shows the design of the top of the housing. After the design is ready we can download the design as an 'STL' file. stereolithography[18] (abbreviated 'STL') is a file format native to the stereolithography CAD software created by 3D Systems. STL describes the geometric surface of a 3D object.

4.1.2 Ultimaker cura

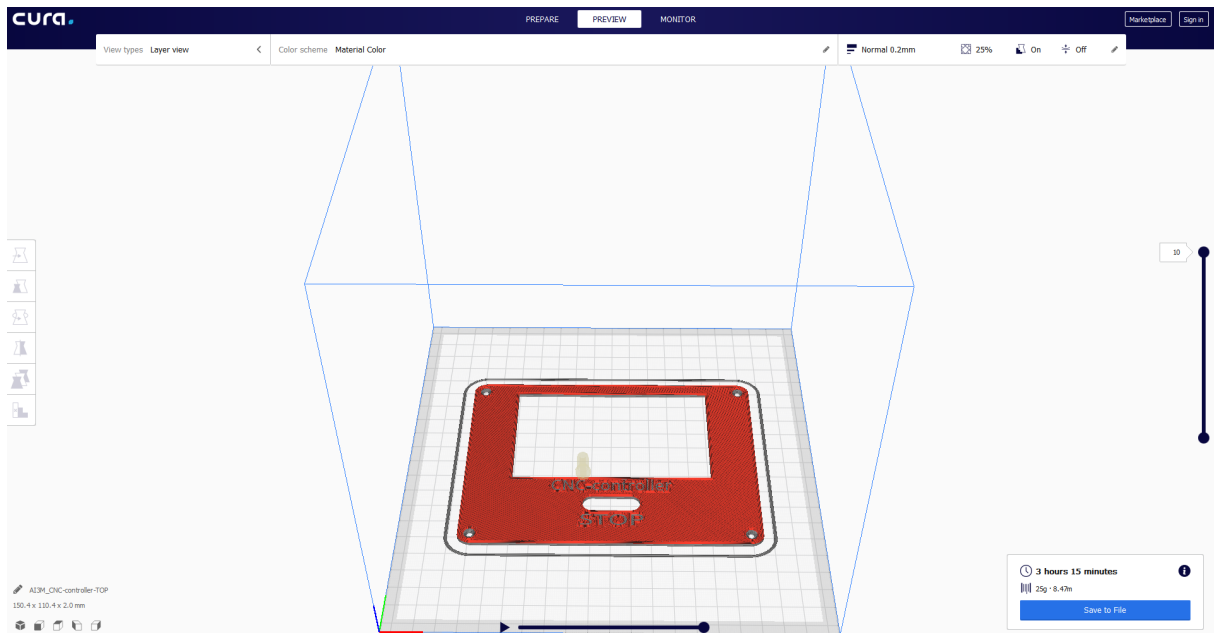


Figure 4.2: Screenshot Ultimaker cura

After receiving the 'STL' file from tinkercad, the cura program is used to process this file, this program is made by ultimaker[19]. Via this program, a number of important settings can be adjusted, such as the material used, printing speed, The figure above is a screenshot of the cura program. Via this program we can also determine the cost price per part, this is important in the next chapter where the total price of the entire project is calculated. When the 'slice' button is pressed, the G code[20] is generated. This code contains all the commands that the 3D printer must execute. At the bottom right of the screenshot you can see the time it takes the 3D printer to print the 3D object. The table below shows the time that the 3D printer needs per object. It also displays the weight of the object and length of the material used.

3D object	weight grams	length meters	time hours
case top	25	8,47	3h15
case bottom	98	32,81	10h22
case slide switch	1	0,23	0h05
case emergency stop	5	1,60	0h22
	total weight	total length	total time
	129	43,11	14h04

4.1.3 Anycubic I3 Mega

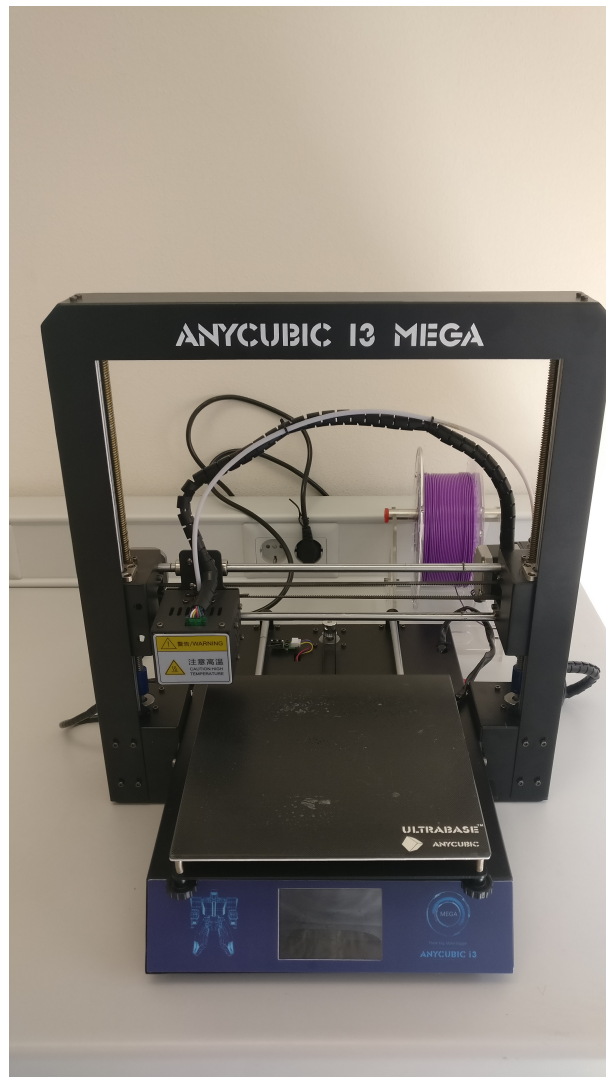


Figure 4.3: Photo anycubic I3 mega

The photo above shows the 3D printer used for printing for the case, this is the anycubic I3 mega[**3D printer**]. This 3D printer has a working surface of 210 x 210 x 205 mm. This 3D printer has a layer resolution of 0.05 - 0.3 mm. The recommended print speed is 60mm/s. Various parameters can be set via a touchscreen such as print bed temperature, extruder temperature, For our case all parameters were set to the standard values. the G code can be loaded via a sd card and the 3D printer is be started via the touchscreen screen. Progress is displayed on the touch screen during printing

4.2 Top



Figure 4.4: Case top

This is the top of the case that will ensure that the case is completely sealed. The large hole is provided for the screen. This hole is slightly smaller than the screen of the TFT, because this way there is certainly no gap along the edge the screen, so the chance of dust in the case is smaller. The hole below the big hole is provided for the emergency stop. The four holes on the corners of the top are provided for mounting the top to the bottom of the case. These four holes are also conical so that the screw is completely recessed in the top, this makes the surface nicer and even.

4.3 Bottom

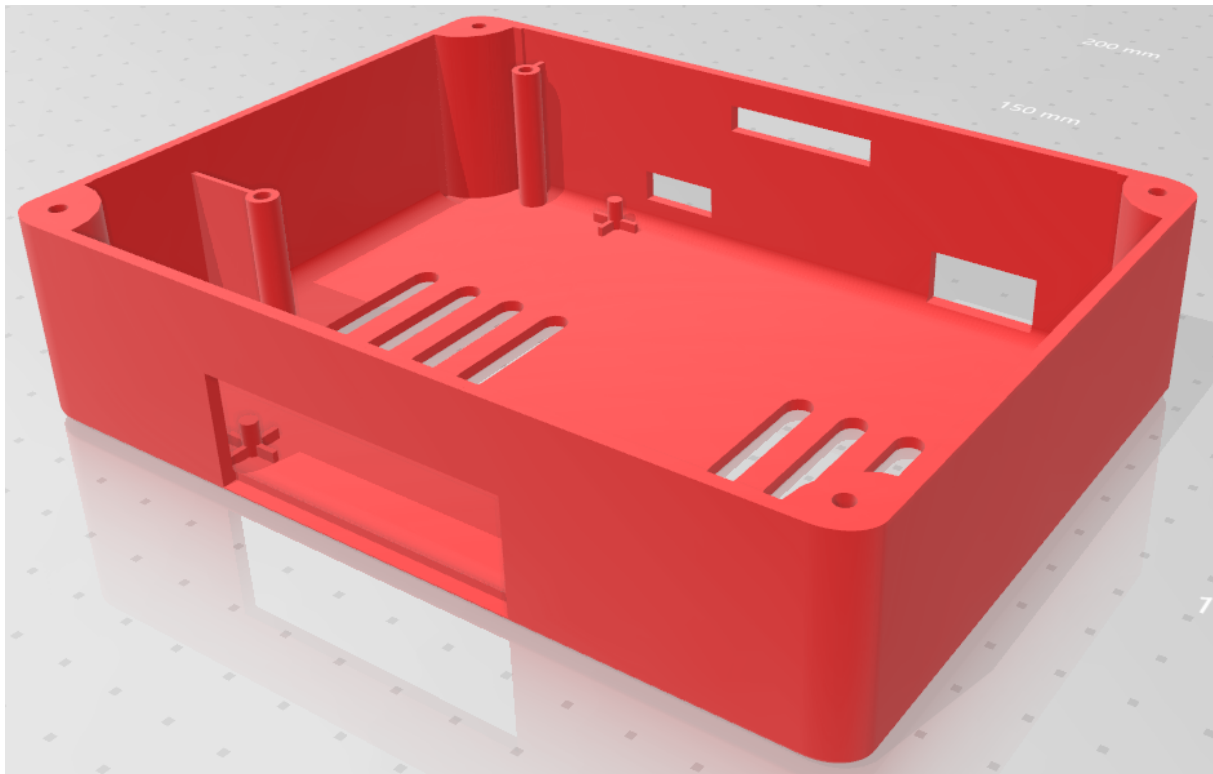


Figure 4.5: Case bottom

This is the second big part of the case, in this part the PCB will be mounted. As you can see on the photos above, 4 pins are provided in the bottom, these are perfectly positioned so that the PCB fits nicely in the housing and everything is nicely aligned. These 4 pins ensure that the PCB cannot move, once the top is mounted on the bottom, then the TFT screen is just against the bottom of the top, so the PCB can not move. The housing also has two higher tubular pillars, which serve to mount the TFT shield. On one side the shield is attached to the PCB with a 40 pins header and on the other side it is mounted on these pillars. The bottom also has a number of holes in the bottom, this is to let the heat out of the case. The holes in the sides each have a specific task. The largest hole is located at the front of the case and is provided for the 25 SUB-D connector. The other three holes are on the other side of the case. The long narrow hole that is located the highest is used to place the SD card in the TFT shield. The left hole on the left photo is provided to be able to access the switch which determines which component can be programmed. The right hole on the left photo serves to pass the USB cable that will serve as power supply.

4.4 Emergency button

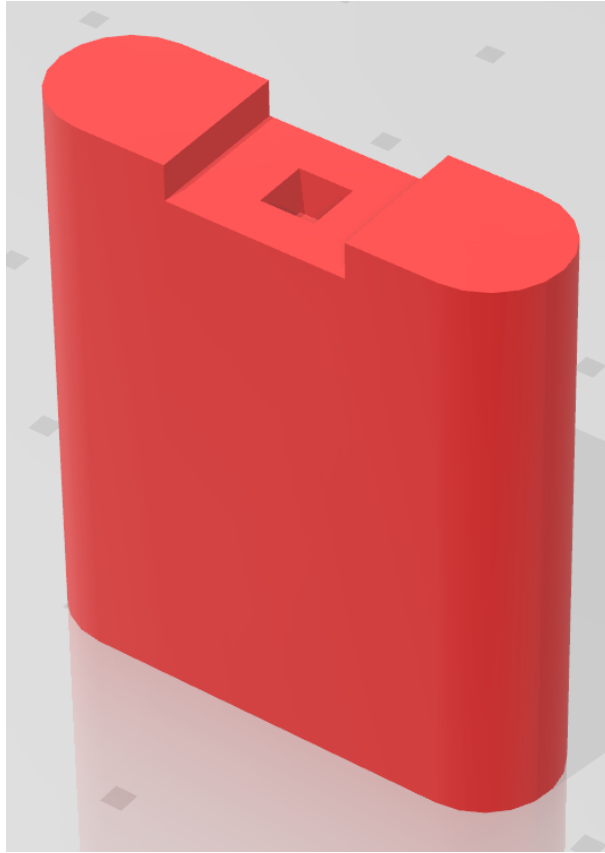


Figure 4.6: Emergency stop

On the figure above you can see the emergency stop. The emergency stop is fairly long, this because the emergency stop has to fit through the whole case and must fit on the push button. On the photo the emergency button is showed upside down, this is because the 3D printer can not print it otherwise. On the top you see a small cubic hole, this whole just fits the push button, this ensures that the button stays in place.

4.5 Slide switch

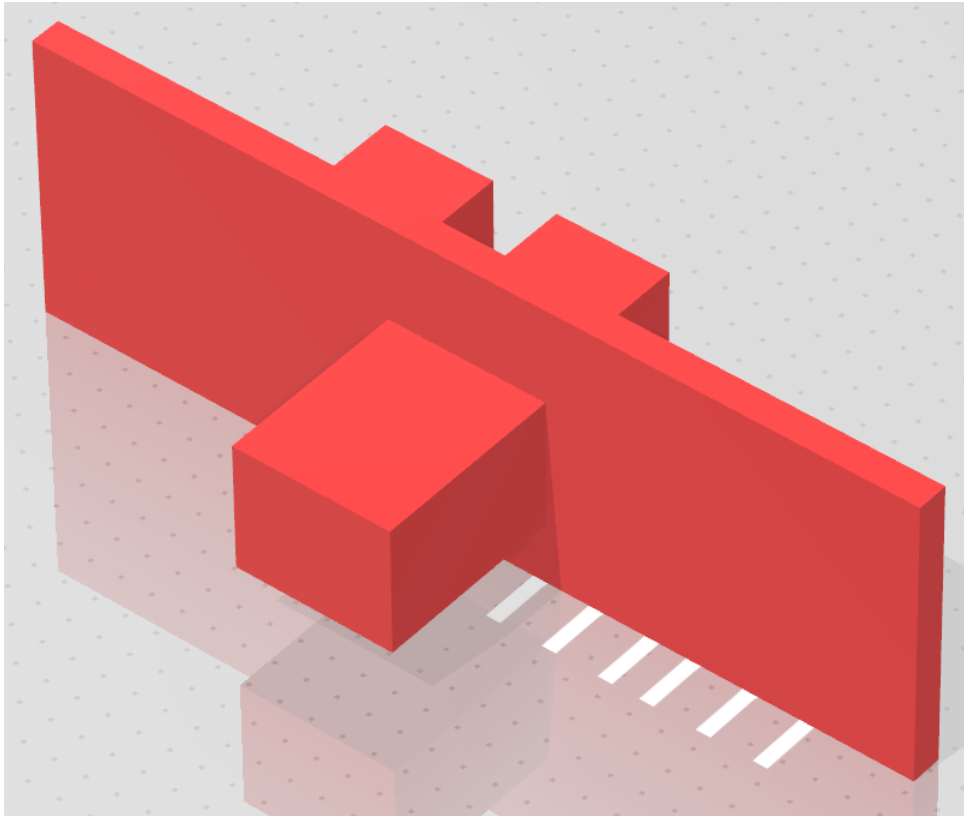


Figure 4.7: Slide switch

On the figure above you can see the slide switch. If the PCB is mounted in the case, it is difficult to change the position of the slide switch as this is a small button. One part of this piece passes over the sliding part of the slide switch. The other side comes through the hole and makes it easier to use the slide switch.

Chapter 5

Cost

5.1 Total cost

stock code	description	price/unit €	units	price €
2308433	TFT screen shield	19,74	1	19,74
	WiFi module	2,23	1	2,23
2773208RL	capacitors 100nF	0,04	8	0,32
298879	capacitors 22uF	0,03	3	0,09
2310647	capacitors 33pF	0,02	2	0,04
2776900RL	capacitor 10uF	0,42	1	0,42
2692284	resistors 10k ohm	0,01	5	0,05
2688819	resistors 1k ohm	0,01	3	0,03
2123191	resistors 3k3 ohm	0,01	2	0,02
2484004	resistor 560 ohm	0,14	1	0,14
1832283	LPC2106FBD48/01,15	10,33	1	10,33
1605581	TC2117-3.3VDBTR	0,48	1	0,48
1146032RL	FT232RL-REEL	2,31	1	2,31
2724158	MC33375ST-1.8T3G	0,25	1	0,25
1332093	MCP23S17-E/SO	0,83	1	0,83
1085299	74HCT04D	0,09	1	0,09
2308433	BCS-120-L-D-TE	3,25	1	3,25
2426381	67503-1020	0,44	1	0,44
1084703	5504F1-25S-02-03-F1	0,37	1	0,37
2668482	10118194-0001LF	0,16	1	0,16
...	PCB	1,76	1	1,76
2435145	JS202011AQN	0,24	1	0,24
176433	BUTTON	0,16	1	0,16
2832032RL	LFXTAL030358	0,17	1	0,17
total price components				43,94

On the table above you can see all components and their cost price. The component on top is the most expensive component, the TFT screen has a price of €19,74. The second most expensive component is the LPC and has a cost of €10,33. The cost price for 3 PCBs is €5,28, which means a price of €1,76 per PCB. The prices of all components added up is equal to €43,94.

The table below shows the price per part of the case as well as the total price for the case. The case is made from polylactide (PLA), this is a renewable plant-based material and is used as an alternative to the traditional plastic. The purchase price for PLA is €20 per kilogram. Based on this weight/price ratio, the cost price can be calculated per part of the case. €20 per kilogram corresponds to 2 cents per gram, the top of the case weighs 25 grams, this means that the price of the top is half a euro. The total price for the case is €2,58.

description	weight/unit gram	price/unit €	units	price €
case top	25	0,50	1	0,50
case bottom	98	1,96	1	1,96
case slide switch	1	0,02	1	0,02
case emergency stop	5	0,10	1	0,10
total price case				2,58

To calculate the total price of the project, the hours of work of Elias and I must also be calculated. In Belgium, a starter with a bachelor's degree without work experience can expect a gross salary of €2.463 [21] per month. On average an employee works 38 hours a week in Belgium, the gross hourly wage per hour is therefore equal to $\frac{€2.463}{(38 * 4)}$, this corresponds to a gross salary of €16,20 per hour. In the table below you can see the hours of work of Elias and I together. In total the hours of work correspond to a cost of €19.164,60.

person	time hour	price/hour €	price €
Elias Verstappe	617	16,20	9.995,40
Daan Ver Eecke	566	16,20	9.169,20
total price hours			19.164,60

The total cost of the entire project can be calculated by adding the cost of the components to the cost of the case and the cost of our hours of work.

price components	€43,94
price case	€2,58
price hours	€19.164,60
total price	€19.211,1

The total cost of the project is €19.211,1. The hours worked represent the largest amount, namely €19.164,60.

5.2 Cost WiFi module

Since this thesis text is about the WiFi part of this project, the cost price for this part is also calculated. To control the CNC machine via the website, the PCB must be fully assembled, which means that the cost price for hardware components remains the same. The cost price of the hours of work is also calculated in this section. Since only I (Daan Ver Eecke) worked on the website, only my hours are taken into account. In total I worked 530 hours on everything that has to do with the website. 530 hours multiplied by €16.20 per working hour is equal to €8 586.

price components	€2,23
price hours	€9.169,20
total price	€9171.43

When calculating the price of the WiFi part, it is clear that the hours of work form the largest cost. The total price for the WiFi part is €9171.43.

Chapter 6

Conclusion

The assignment was to control the CNC machine on the basis of a self-designed control unit. The CNC machine must also be capable of being controlled via a website.

I think we have been able to approach the objectives of the project quite well. The CNC controller can set parameters for the CNC machine, provide an offset to the CNC machine and also read and process a file. Designing the website has also been fairly successful, various values can also be set and the offset of the CNC machine can also be modified via the website. A goal was to upload a file via the website, this was unsuccessful since it was quite complex. The communication between the ESP8266 and LPC2106 is also not bi-directional, only the ESP8266 can send data to the microcontroller. If the microcontroller would send data to the WiFi module, this would make the website slower and this is not desired. Due to time constraints I could not add these features to the website without unwanted side effects.

If we still had time to further develop the project, then a number of things could certainly be programmed. HTTPS could be implemented in the website, this would make the website a lot safer. We could also make the communication between the two microcontrollers bidirectional, this would ensure that the website as well as the CNC controller both display the latest values. We could also program that the website can upload a file to the CNC controller to control the CNC machine. If we could add all these features to the project, the project would be completely finished.

Bibliography

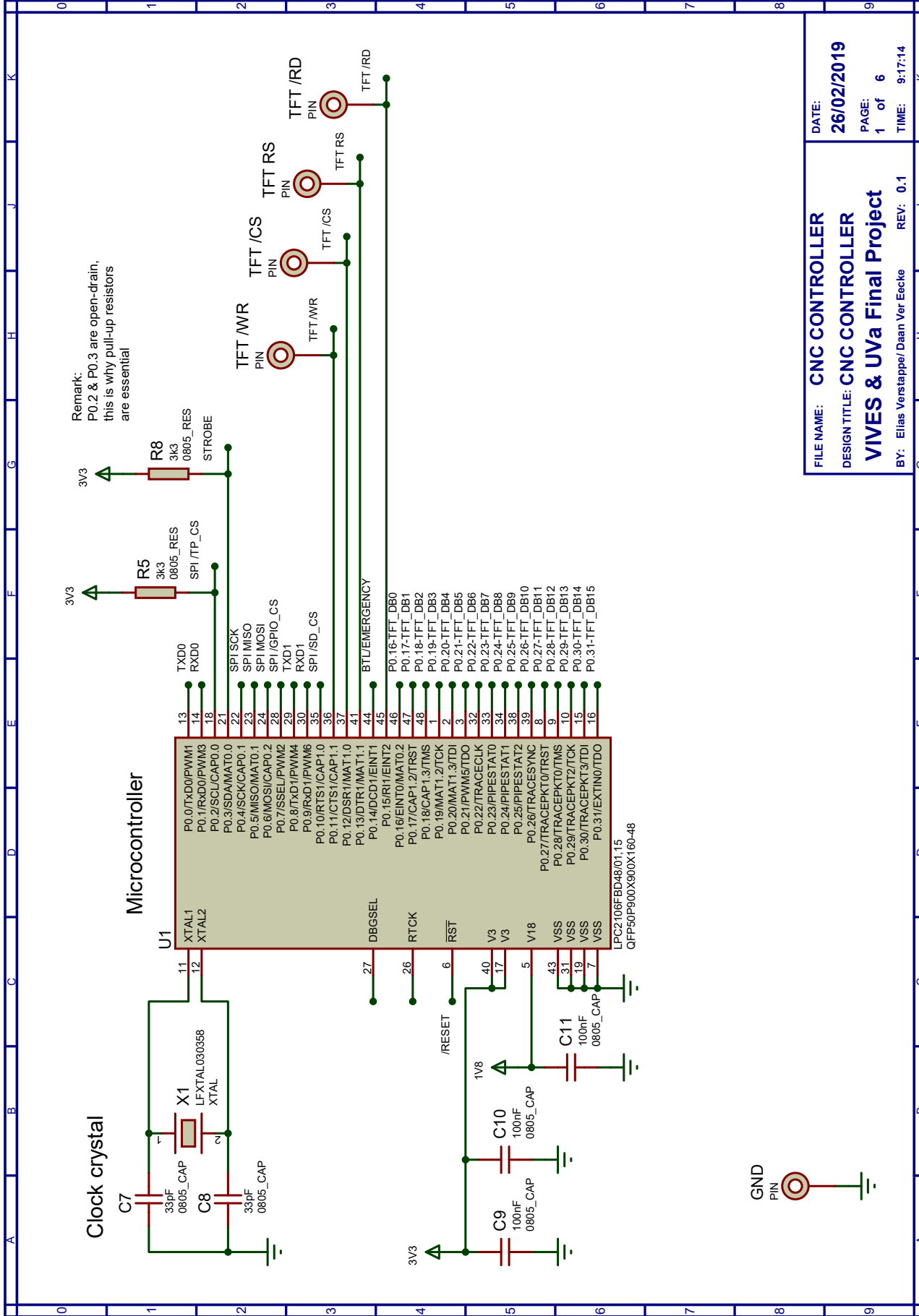
- [1] Wikipedia. *ESP8266 wikipedia*. May 6, 2019. URL: <https://en.wikipedia.org/wiki/ESP8266>.
- [2] ESP8266 arduino core. *ESP8266 arduino core*. May 6, 2019. URL: <https://arduino-esp8266.readthedocs.io/en/latest/index.html>.
- [3] www.kiwi-electronics.nl. *ESP8266*. May 28, 2019. URL: <https://www.kiwi-electronics.nl/image/cache/data/products/netwerk/wireless/KW-1847-02-1000x667.jpg>.
- [4] Espressif Systems. *ESP8266 datasheet*. June 8, 2019. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [5] Forward Computing and Control Pty. *ESP8266*. June 11, 2019. URL: <https://www.forward.com.au/pfod/ESP8266/GPIOpins/index.html>.
- [6] arduino. *arduino*. June 8, 2019. URL: <https://www.arduino.cc/>.
- [7] wikipedia. *arduino wikipedia*. June 8, 2019. URL: [https://nl.wikipedia.org/wiki/Arduino_\(computerplatform\)](https://nl.wikipedia.org/wiki/Arduino_(computerplatform)).
- [8] wikipedia. *HTTPS*. June 6, 2019. URL: https://nl.wikipedia.org/wiki/HyperText_Transfer_Protocol_Secure.
- [9] w3schools.com. *responsive webpages*. URL: https://www.w3schools.com/css/css_rwd_intro.asp.
- [10] Wikipedia. *responsive webpages*. Jan. 30, 2018. URL: https://nl.wikipedia.org/wiki/Responsive_webdesign.
- [11] Wikipedia. *DynamisDNS*. Apr. 24, 2019. URL: https://en.wikipedia.org/wiki/Dynamic_DNS.
- [12] ugc.kn3.net. *linksys router*. May 27, 2019. URL: [https://ugc.kn3.net/i/760x/http://www.appinformatica.com/imf/routers-router-linksys-ap-54mbpswrt54gl\(-linux\)-1g.jpg](https://ugc.kn3.net/i/760x/http://www.appinformatica.com/imf/routers-router-linksys-ap-54mbpswrt54gl(-linux)-1g.jpg).
- [13] Wikipedia. *HyperText Markup Language*. Apr. 17, 2019. URL: https://nl.wikipedia.org/wiki/HyperText_Markup_Language.
- [14] Wikipedia. *Cascading Style Sheets*. June 17, 2018. URL: https://nl.wikipedia.org/wiki/Cascading_Style_Sheets.
- [15] Wikipedia. *JavaScript*. May 6, 2019. URL: <https://nl.wikipedia.org/wiki/JavaScript>.
- [16] Autodesk. *tinkercad*. June 6, 2019. URL: <https://www.tinkercad.com/>.
- [17] Autodesk. *Autodesk*. June 6, 2019. URL: <https://www.autodesk.com/>.
- [18] wikipedia. *STL explanation*. June 6, 2019. URL: [https://en.wikipedia.org/wiki/STL_\(file_format\)](https://en.wikipedia.org/wiki/STL_(file_format)).
- [19] ultimaker. *cura_ultimaker*. June 6, 2019. URL: <https://ultimaker.com/>.
- [20] wikipedia. *G_code*. June 6, 2019. URL: <https://nl.wikipedia.org/wiki/G-code>.
- [21] www.vacature.com. *cost hours of work*. June 4, 2019. URL: <https://www.vacature.com/nl-be/carriere/salaris/hoeveel-verdien-je-in-de-sector-elektronica-en-technologische-industrie>.
- [22] anycubic. *anycubic I3 mega*. June 6, 2019. URL: <http://www.anycubic3d.com/products/show/1048.html>.

Appendix A

Schematics PCB

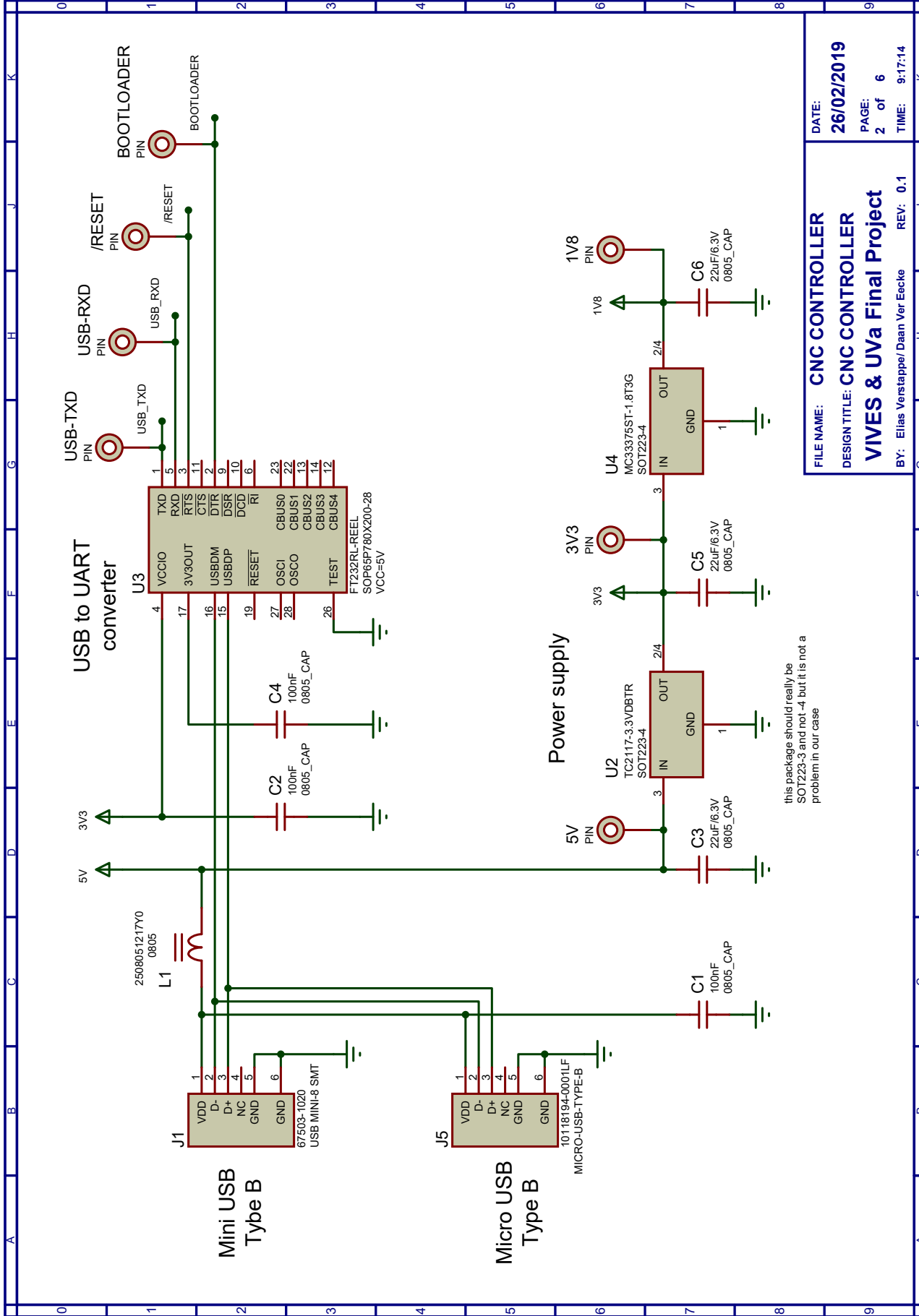
This appendix contains the schematics for the PCB.

- page 1 : microcontroller
- page 2 : UART converter, USB connections and voltage regulators
- page 3 : 40 pin TFT header, 25 pin SUB-D connector, the 16 bit IO expander and the NOT port
- page 4 : WiFi module, emergency stop and slide switch
- page 5 : power dissipation analysis
- page 6 : heat dissipation analysis



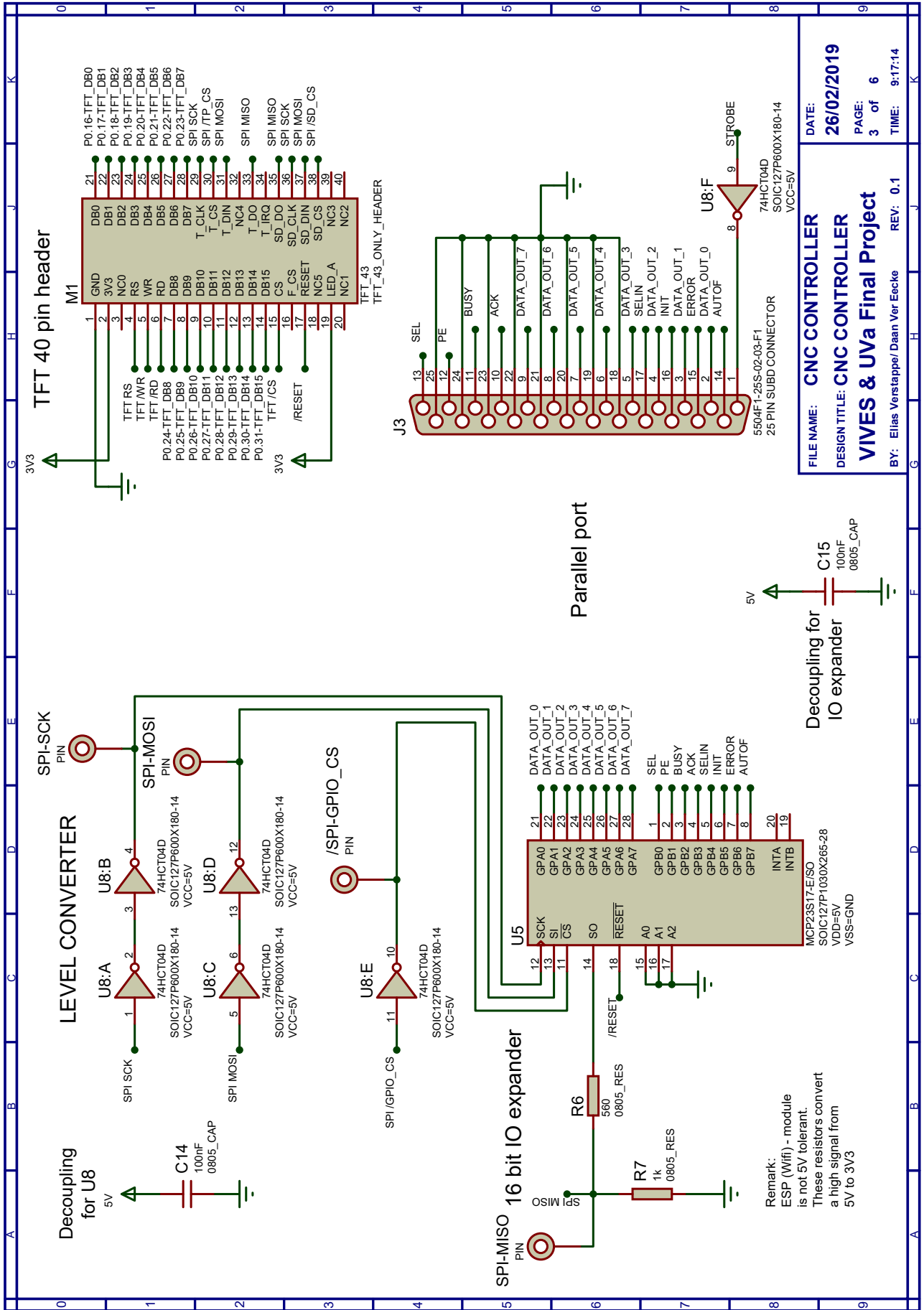
Remark:
P0.2 & P0.3 are open-drain,
this is why pull-up resistors
are essential

FILE NAME: **CNC CONTROLLER**
 DESIGN TITLE: **CNC CONTROLLER**
VIVES & Uva Final Project
 BY: Elias Verstappe/ Daan Ver Eecke REV: 0.1
 DATE: 26/02/2019
 PAGE: 1 of 6
 TIME: 9:17:14



FILE NAME: **CNC CONTROLLER**
 DESIGN TITLE: **CNC CONTROLLER**
VIVES & Uva Final Project
 BY: Elias Verstappe/ Daan Ver Eecke REV: 0.1
 DATE: **26/02/2019**
 PAGE: **2** of **6**
 TIME: **9:17:14**

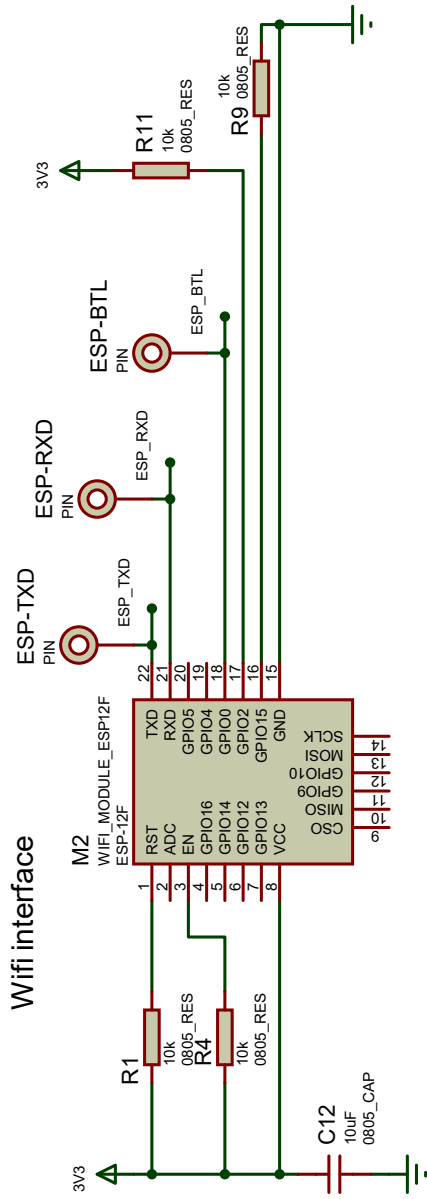
this package should really be SOT223-3 and not -4 but it is not a problem in our case



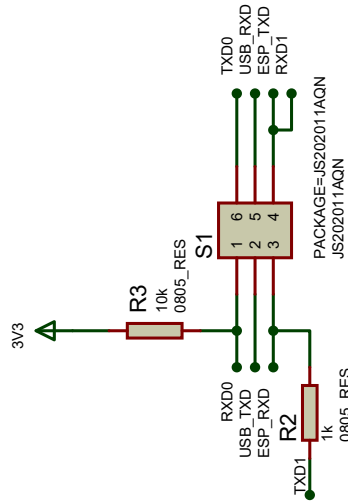
FILE NAME: CNC CONTROLLER
DESIGN TITLE: CNC CONTROLLER
VIVES & Uva Final Project
BY: Elias Verstappel/ Daan Ver Eecke
REV: 0.1

DATE: 26/02/2019
PAGE: 3 of 6
TIME: 9:17:14

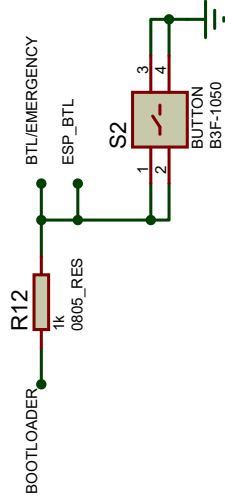
Wifi interface



UART SWITCH



BOOTLOADER



FILE NAME: CNC CONTROLLER
DESIGN TITLE: CNC CONTROLLER
VIVES & UVa Final Project
DATE: 26/02/2019
PAGE: 4 of 6
TIME: 9:17:14
BY: Elias Verstappel/ Daan Ver Eecke
REV: 0.1

~~~~~  
**POWER DISSIPATION**  
 ~~~~~

+	-----+		
	5V		3V3
+	-----+		
	40uA (level-converter)		15mA (USB to UART controller)
	2mA (I/O expander)		200mA (TFT)
	420.2mA (3V3)		170mA (WIFI module)
	130uA (3V3 regulator internal)		35mA (1V8)
			200uA (1V8 regulator internal)
+	-----+		
	TOTAL @ 5V = 422.3mA		TOTAL @ 3V3 = 420.2mA
+	-----+		
	POWER @ 5V = 2.12W		POWER @ 3V3 = 1.39W
+	-----+		
			TOTAL @ 1V8 = 35mA
+	-----+		
			POWER @ 1V8 = 0.063W
+	-----+		

FILE NAME:	CNC CONTROLLER	DATE:	26/02/2019
DESIGN TITLE:	CNC CONTROLLER	PAGE:	5 of 6
VIVES & UVa Final Project		TIME:	9:17:14
BY:	Elias Verstappe/Daan Ver Eecke	REV:	0.1


~~~~~  
**HEAT DISSIPATION**  
 ~~~~~

component	IC code	max temperatur (°C)	conditions
microcontroller	LPC2016	85	IDD(45mA)@60MHz
USB to UART	FT232RL	85	/
16 bit I/O expander	MCP23S17	125	/
level converter (NOT port)	74HCT04D	70	above 70°C Ptot drops 8mw/K
wifi interface	ESP-12F	125	operating temperature
3V3 voltage regulator	MC33375ST-1.8T3G	245°C/W	/
1V8 voltage regulator	TC2117-3.3VDBTR	59°C/W	/

~~~~~  
**1V8 voltage regulator**  
 ~~~~~

$$\begin{aligned}
 P &= U * I = 35\text{mA} * (3V3-1V8) = 0.0525W \Rightarrow T = 245^\circ\text{C/W} * 0.0525W + T_a \\
 &= 12.86^\circ\text{C} + 25^\circ\text{C} \\
 &= 37.86^\circ\text{C}
 \end{aligned}$$

~~~~~  
**3V3 voltage regulator**  
 ~~~~~

$$\begin{aligned}
 P &= U * I = 420.2\text{mA} * (5V-3V3) = 0.72W \Rightarrow T = 59^\circ\text{C/W} * 0.72W + T_a \\
 &= 42.15^\circ\text{C} + 25^\circ\text{C} \\
 &= 67.15^\circ\text{C}
 \end{aligned}$$

FILE NAME: CNC CONTROLLER	DATE: 26/02/2019
DESIGN TITLE: CNC CONTROLLER	PAGE: 6 of 6
VIVES & UVa Final Project	TIME: 9:17:14
BY: Elias Verstappe/Daan Ver Eecke	REV: 0.1

Appendix B

Photographs case

This appendix contains photos of every part of the case. The explanation of how these components are made can be found in chapter four.

- page 1 : photograph case top
- page 2 : photograph case bottom and photograph case mounted
- page 3 : photograph emergency stop and photograph slide switch



Figure B.1: Photograph case top



Figure B.2: Photograph case bottom



Figure B.5: Photograph case mounted



Figure B.3: Photograph emergency stop

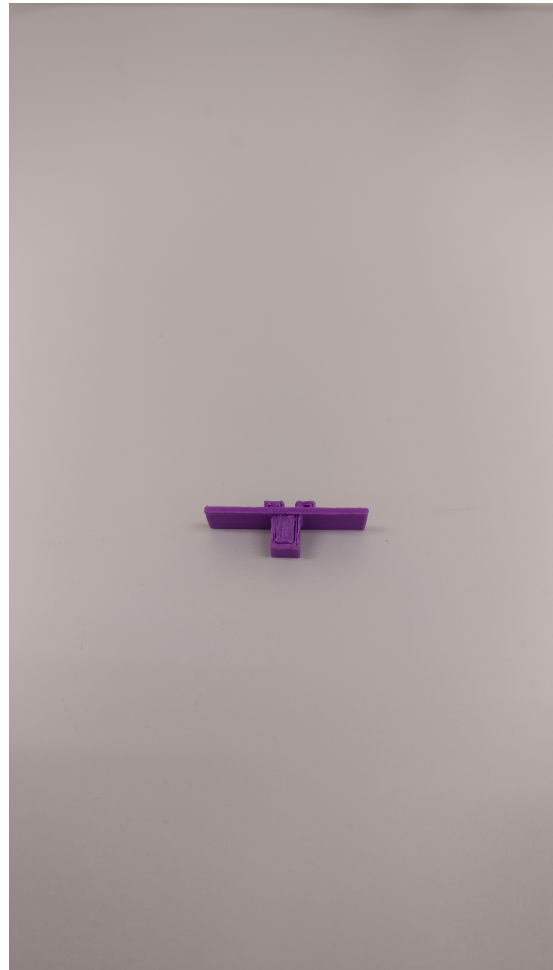


Figure B.4: Photograph slide switch

Appendix C

Website source code

C.0.1 Welcome web page

```
1 const char WELCOME_page[] PROGMEM = R"=====(
2 <html>
3 <head>
4 <meta name="author" content="Daan Ver Eecke">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0"
6 >
7 <title>welcome</title>
8 <style>
9 * {
10     box-sizing: border-box;
11 }
12 .row::after {
13     content: "";
14     clear: both;
15     display: block;
16 }
17 [class*="col-"] {
18     float: left;
19     padding: 15px;
20 }
21 html {
22     font-family: "Lucida Sans", sans-serif;
23 }
24
25 .col-1 {width: 8.33%;}
26 .col-2 {width: 16.66%;}
27 .col-3 {width: 25%;}
28 .col-4 {width: 33.33%;}
29 .col-5 {width: 41.66%;}
30 .col-6 {width: 50%;}
31 .col-7 {width: 58.33%;}
32 .col-8 {width: 66.66%;}
33 .col-9 {width: 75%;}
34 .col-10 {width: 83.33%;}
35 .col-11 {width: 91.66%;}
36 .col-12 {width: 100%;}
37
38 @media only screen and (max-width: 768px) {
39     [class*="col-"] {
40         width: 100%;
```

```

40     }
41   }
42   .header {
43     background-color: #4CAF50;
44     color: #ffffff;
45     padding: 15px;
46   }
47   .footer {
48     background-color: #4CAF50;
49     color: #ffffff;
50     padding: 15px;
51     text-align: center;
52   }
53   body{
54     background-color: #c5cedd;
55   }
56   button {
57     display: inline-block;
58     width: 60%;
59     background-color: #4CAF50;
60     border: none;
61     color: white;
62     padding: 15px 32px;
63     text-align: center;
64     text-decoration: none;
65   }
66   button:hover {
67     background-color:#00802b;
68     color: #c5cedd;
69   }
70   .form{
71     text-align: center;
72     border:2px solid #4CAF50;
73     border-radius:4px;
74   }
75   input[type=text] {
76     width: 60%;
77     border:2px solid #aaa;
78     border-radius:4px;
79     height:30px;
80     display: inline-block;
81     text-align: center;
82     background-color: white;
83     color: #4CAF50;
84     transition:.3s
85   }
86   input[type=text]:focus{
87     border-color: #4CAF50;
88     border-shadow: 0 0 8px 0 #4CAF50;
89   }
90   input[type=password] {
91     width: 60%;
92     border:2px solid #aaa;
93     border-radius:4px;
94     height:30px;
95     display: inline-block;
96     text-align: center;

```



```

97     background-color: white;
98     color: #4CAF50;
99     }
100    input[type=password]:focus{
101        border-color: #4CAF50;
102        border-shadow: 0 0 8px 0 #4CAF50;
103    }
104    #infoText {
105        display: inline-block;
106    }
107    p {
108        text-align: center;
109    }
110    .photo {
111        display: block;
112        margin-left: auto;
113        margin-right: auto;
114    }
115    </style>
116    </head>
117    <body>
118        <div class="header">
119            <h1>Welcome</h1>
120        </div>
121        <div class="row">
122            <div class="col-12">
123                <p>This thesis project is created by Verstappe Elias and Ver
124                Eecke Daan.</p>
125                <p>This is a wireless adaptation for a standalone CNC controller
126                .</p>
127            </div>
128            <div class="row">
129                <div class="col-2">
130                <div class="col-8 form">
131                    <p>To proceed, please enter username and password.</p>
132                    <input type="text" id="username" placeholder="Username"><br />
133                    <input type="password" id="password" placeholder="Password"><br
134                />
135                    <p id="infoText"></p><br />
136                    <button onClick="login()">Log in</button>
137                </div>
138                <div class="col-2">
139                </div>
140            </div>
141            <div class="row">
142                <div class="col-12">
143                    <p id="colaboration">A colaboration between:</p>
144                </div>
145            </div>
146            <div class="row">
147                <div class="col-6">
148                    

```

```

150     
151     </div>
152 </div>
153 <div class="footer">
154     <h1>CNC-controller</h1>
155 </div>
156 <script>
157     function login() {
158         var user = document.getElementById("username").value;
159         var pass = document.getElementById("password").value;
160         var answer = "";
161         var xhttp = new XMLHttpRequest();
162         var url = "Login?credentials" + "=" + user + "!" + pass;
163         xhttp.onreadystatechange = function() {
164             if (this.readyState == 4 && this.status == 200) {
165                 var answer = this.responseText;
166                 handleResponse(answer);
167             }
168         }
169         xhttp.open("GET", url , true);
170         xhttp.send();
171     }
172
173     function handleResponse(answer) {
174         if(answer == "correct"){
175             document.getElementById("infoText").innerHTML = "Username and
password correct!";
176             document.getElementById("infoText").style.color = "green";
177             document.location.href = "/changeWebPage?webpage=Routepro/
Drillpro";
178         }else{
179             document.getElementById("infoText").innerHTML = "Username or
password wrong!";
180             document.getElementById("infoText").style.color = "red";
181         }
182     }
183
184     document.onkeypress = function(event) {
185         var x = event.charCode;
186         switch(x){
187             case 13:
188                 login();
189                 break;
190         }
191     };
192 </script>
193 </body>
194 </html>
195 )=====";

```

C.0.2 Routepro/Drillpro web page

```
1 const char ROUTEPRO_page[] PROGMEM = R"=====(
2   <html>
3     <head>
4       <meta name="author" content="Daan Ver Eecke">
5       <meta name="viewport" content="width=device-width, initial-
6 scale=1.0">
7       <title>routepro</title>
8       <style>
9         * {
10          box-sizing: border-box;
11        }
12        .row::after {
13          content: "";
14          clear: both;
15          display: table;
16        }
17        [class*="col-"] {
18          float: left;
19          padding: 15px;
20        }
21        html {
22          font-family: "Lucida Sans", sans-serif;
23        }
24        [class*="col-"] {
25          width: 100%;
26        }
27        @media only screen and (min-width: 600px) {
28          .col-s-1 {width: 8.33%;}
29          .col-s-2 {width: 16.66%;}
30          .col-s-3 {width: 25%;}
31          .col-s-4 {width: 33.33%;}
32          .col-s-5 {width: 41.66%;}
33          .col-s-6 {width: 50%;}
34          .col-s-7 {width: 58.33%;}
35          .col-s-8 {width: 66.66%;}
36          .col-s-9 {width: 75%;}
37          .col-s-10 {width: 83.33%;}
38          .col-s-11 {width: 91.66%;}
39          .col-s-12 {width: 100%;}
40        }
41        @media only screen and (min-width: 768px) {
42          .col-1 {width: 8.33%;}
43          .col-2 {width: 16.66%;}
44          .col-3 {width: 25%;}
45          .col-4 {width: 33.33%;}
46          .col-5 {width: 41.66%;}
47          .col-6 {width: 50%;}
48          .col-7 {width: 58.33%;}
49          .col-8 {width: 66.66%;}
50          .col-9 {width: 75%;}
51          .col-10 {width: 83.33%;}
52          .col-11 {width: 91.66%;}
53          .col-12 {width: 100%;}
54        }
55        body {
```

```

55         background-color: #c5cedd;
56     }
57     button {
58         background-color: #4CAF50;
59         border: 1px solid green;
60         color: #c5cedd;
61         text-align: center;
62         text-decoration: none;
63         display: inline-block;
64         display: block;
65         width: 100%;
66         height: 5%;
67     }
68     button:hover {
69         background-color:#c5cedd; /* Green */
70         color: #4CAF50;
71     }
72     b {
73         color: yellow;
74     }
75     .table1 {
76         border: 1px solid black;
77         border-collapse: collapse;
78         text-align: center;
79         height: 50px;
80         width: 100%;
81     }
82     .table1 th {
83         border: 1px solid black;
84         height: 75px;
85         width: 20%;
86     }
87     .table1 td {
88         border: 1px solid black;
89         height: 50px;
90     }
91     #tableheader {
92         width: 20%;
93     }
94     #small {
95         width: 10%;
96     }
97     .table2 {
98         border: 1px solid black;
99         border-collapse: collapse;
100        text-align: center;
101        height: 50px;
102        width: 100%;
103    }
104    .table2 th {
105        border: 1px solid black;
106        height: 75px;
107        width: 50%;
108    }
109    .table2 td {
110        border: 1px solid black;
111        height: 50px;

```

```

112     }
113     input {
114         align-self: center;
115         text-align: center;
116         width: 75px;
117     }
118     .header {
119         background-color: #4CAF50;
120         color: #ffffff;
121         padding: 15px;
122     }
123     .footer {
124         background-color: #4CAF50;
125         color: #ffffff;
126         text-align: center;
127         font-size: 12px;
128         padding: 15px;
129     }
130 </style>
131 </head>
132 <body>
133     <div class="header">
134         <h1>Routepro/drillpro</h1>
135     </div>
136     <div class="row">
137         <div class="col-2 col-s-12">
138             <button id="config_button" onclick="saveOrIgnoreChanges
('Configuration')"><b>C</b>onfig.</button>
139             <button id="tools_button" onclick="saveOrIgnoreChanges(
'Tools')"><b>T</b>ools</button>
140             <button id="offest_button" onclick="saveOrIgnoreChanges(
'Offset')"><b>O</b>ffset</button>
141             <button id="load_button" onclick="saveOrIgnoreChanges(
'Load')"><b>L</b>oad</button>
142             <button id="save_button" onclick="receivingRouteproData
())">S<b>a</b>ve</button>
143         </div>
144         <div class="col-7 col-s-12">
145             <table class="table1">
146                 <tr>
147                     <th> </th>
148                     <th>O<b>f</b>fset:</th>
149                     <th>Cl<b>i</b>p:</th>
150                     <th> </th>
151                     <th id="small"><b>V</b>ectors:</th>
152                     <th id="tableheader"> </th>
153                 </tr>
154                 <tr>
155                     <td >X: </td>
156                     <td ><input type="number" id="X_offset" value="
0"></td>
157                     <td ><input type="number" id="X_clip" value="
325"></td>
158                     <td id="small">from</td>
159                     <td><input type="number" id="X_vectors" value="
0"></td>
160                     <td> </td>

```

```

161         </tr>
162         <tr>
163             <td>Y: </td>
164             <td ><input type="number" id="Y_offset" value="
0"></td>
165             <td ><input type="number" id="Y_clip" value="
278"></td>
166             <td id="small">to</td>
167             <td ><input type="number" id="Y_vectors" value=
"0"></td>
168             <td ></td>
169         </tr>
170         <tr>
171             <th> </th>
172             <th>Boar<b>d</b>:</th>
173             <th><b>B</b>ase:</th>
174             <td><span id="X_min_value" >0</td>
175             <td id="small">&lt; X &lt;>
176             <td><span id="X_max_value" >0</td>
177         </tr>
178         <tr>
179             <td>Z: </td>
180             <td ><input type="number" id="Z_board" value="
1.5"></td>
181             <td ><input type="number" id="Z_base" value="
2.0"></td>
182             <td><span id="Y_min_value" >0</td>
183             <td id="small">&lt; Y &lt;>
184             <td><span id="Y_max_value" >0</td>
185         </tr>
186     </table>
187 </div>
188 <div class="col-3 col-s-12">
189     <table class="table2">
190         <tr>
191             <th>Count :</th>
192             <th> </th>
193         </tr>
194         <tr>
195             <td>Total</td>
196             <td><span id="Total" >0</td>
197         </tr>
198         <tr>
199             <td>Done</td>
200             <td><span id="Done" >0</td>
201         </tr>
202     </table>
203 </div>
204 </div>
205 <div class="footer">
206     <p>CNC-controller for Bungard CNC-machine</p>
207 </div>
208 <script src="https://ajax.googleapis.com/ajax/libs/jquery
/3.3.1/jquery.min.js"></script>
209 <script>
210     var saved = false;
211

```

```

212         function saveOrIgnoreChanges(webpage){
213             if (saved == false){
214                 var r = confirm("Save changes?");
215                 if (r == true){
216                     receivingRouteproData();
217                 }
218             }
219             document.location.href = "/changeWebPage?webpage=" +
webpage;
220         }
221
222         function receivingRouteproData(){
223             saved = true;
224             var index = [$("#X_offset").val(),$("#X_clip").val(),$(
"#X_vectors").val(),
225                 $("#Y_offset").val(),$("#Y_clip").val(),$("#Y_vectors")
.val(),$("#Z_board").val(),
226                 $("#Z_base").val(),$("#X_min_value").text(),$("#
X_max_value").text(),$("#Y_min_value").text(),$("#Y_max_value").text(),
227                 $("#Total").text(),$("#Done").text(),$("#File_Name").
text()];
228             var url = "handleData?RouteproData=";
229             for (let i = 0; i < index.length; i++){
230                 if(i == 0){
231                     var temp = index[i];
232                 }else{
233                     var temp = "!" + index[i];
234                 }
235                 url += temp;
236             }
237             var xhttp = new XMLHttpRequest();
238             xhttp.open("GET", url);
239             xhttp.onload = function() {
240                 var data = JSON.parse(xhttp.responseText);
241                 console.log(data);
242             }
243             xhttp.send();
244         }
245
246         document.onkeypress = function(event) {
247             var x = event.charCode;
248             switch(x){
249                 case 99:
250                     document.location.href = "/changeWebPage?
webpage=Configuration";
251                     break;
252                 case 116:
253                     document.location.href = "/changeWebPage?
webpage=Tools";
254                     break;
255                 case 111:
256                     document.location.href = "/changeWebPage?
webpage=Offset";
257                     break;
258                 case 108:
259                     document.location.href = "/changeWebPage?
webpage=Load";

```

```
260         break;
261     case 97:
262         receivingRouteproData();
263         break;
264     case 102:
265         document.getElementById("X_offset").focus();
266         break;
267     case 105:
268         document.getElementById("X_clip").focus();
269         break;
270     case 118:
271         document.getElementById("X_vectors").focus();
272         break;
273     case 100:
274         document.getElementById("Z_board").focus();
275         break;
276     case 98:
277         document.getElementById("Z_base").focus();
278         break;
279     }
280 };
281 </script>
282 </body>
283 </html>
284 )=====";
```


C.0.3 Configuration web page

```
1 const char CONFIG_page[] PROGMEM = R"=====(
2 <html>
3   <head>
4     <meta name="author" content="Daan Ver Eecke">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0"
6   >
7     <title>config.</title>
8     <style>
9       * {
10        box-sizing: border-box;
11      }
12      .row::after {
13        content: "";
14        clear: both;
15        display: table;
16      }
17      [class*="col-"] {
18        float: left;
19        padding: 15px;
20      }
21      html {
22        font-family: "Lucida Sans", sans-serif;
23      }
24      [class*="col-"] {
25        width: 100%;
26      }
27      @media only screen and (min-width: 600px) {
28        .col-s-1 {width: 8.33%;}
29        .col-s-2 {width: 16.66%;}
30        .col-s-3 {width: 25%;}
31        .col-s-4 {width: 33.33%;}
32        .col-s-5 {width: 41.66%;}
33        .col-s-6 {width: 50%;}
34        .col-s-7 {width: 58.33%;}
35        .col-s-8 {width: 66.66%;}
36        .col-s-9 {width: 75%;}
37        .col-s-10 {width: 83.33%;}
38        .col-s-11 {width: 91.66%;}
39        .col-s-12 {width: 100%;}
40      }
41      @media only screen and (min-width: 768px) {
42        .col-1 {width: 8.33%;}
43        .col-2 {width: 16.66%;}
44        .col-3 {width: 25%;}
45        .col-4 {width: 33.33%;}
46        .col-5 {width: 41.66%;}
47        .col-6 {width: 50%;}
48        .col-7 {width: 58.33%;}
49        .col-8 {width: 66.66%;}
50        .col-9 {width: 75%;}
51        .col-10 {width: 83.33%;}
52        .col-11 {width: 91.66%;}
53        .col-12 {width: 100%;}
54      }
55      body{
```

```

55     background-color: #c5cedd;
56 }
57 .button-group button {
58     background-color: #4CAF50;
59     border: none;
60     color: white;
61     padding: 15px 32px;
62     text-align: center;
63     text-decoration: none;
64     display: inline-block;
65     font-size: 16px;
66     cursor: pointer;
67     float: left;
68 }
69 button {
70     width: 33%;
71 }
72 .button-group button:hover {
73     background-color:#00802b;
74     color: #c5cedd;
75 }
76 #apply_changes {
77     background-color: #4CAF50;
78     border: none;
79     color: white;
80     padding: 15px 32px;
81     text-align: center;
82     text-decoration: none;
83 }
84 #apply_changes:hover {
85     background-color:#00802b;
86     color: #c5cedd;
87 }
88 b {
89     color: yellow;
90 }
91 .header {
92     background-color: #4CAF50;
93     color: #ffffff;
94     padding: 5px;
95 }
96 .footer {
97     background-color: #4CAF50;
98     color: #ffffff;
99     text-align: center;
100    font-size: 12px;
101    padding: 15px;
102 }
103 table {
104     border: 1px solid black;
105     width: 100%
106 }
107 th {
108     width: 50%;
109     text-align: center;
110 }
111 td {

```

```

112     width: 50%;
113     text-align: center;
114 }
115 #corr_values {
116     vertical-align: top;
117 }
118 input[type=number] {
119     border:2px solid #aaa;
120     border-radius:4px;
121     height:30px;
122     display: inline-block;
123     text-align: center;
124     background-color: white;
125     color: #4CAF50;
126     transition:.3s
127 }
128 input[type=number]:focus{
129     border-color: #4CAF50;
130     border-shadow: 0 0 8px 0 #4CAF50;
131 }
132 </style>
133 </head>
134 <body>
135     <div class="header">
136         <h1>Configuration</h1>
137     </div>
138     <div class="row">
139         <div class="col-12 col-s-12 button-group">
140             <button id="load_button" onclick="window.location.href='/
changeWebPage?webpage=Load'"><b>L</b>oad</button>
141             <button id="save_button" onclick="receivingConfigurationData()"><
b>S</b>ave</button>
142             <button id="back_button" onclick="saveOrIgnoreChanges()"><b>B</b>
ack</button>
143         </div>
144     </div>
145     <div class="row">
146         <div class="col-12 col-s-12">
147             <table>
148                 <tr>
149                     <th><h2>Coordinates:</h2></th>
150                     <th><h2><b>C</b>hord angle:</h2></th>
151                 </tr>
152                 <tr>
153                     <td><p id="coordinates">...</p></td>
154                     <td><input type="number" id="chord_angle" value="05"></td>
155                 </tr>
156             </table>
157         </div>
158     </div>
159     <div class="row">
160         <div class="col-12 col-s-12">
161             <table>
162                 <tr>
163                     <th><h2>Correction:</h2></th>
164                     <th><h2>S<b>t</b>roke:</h2></th>
165                 </tr>

```

```

166         <tr>
167             <td id="corr_values">
168                 <p id="X_correct" >X: 0.9975</p>
169                 <p id="Y_correct" >Y: 0.9975</p>
170                 <p id="Z_correct" >Z: 0.8988</p>
171                 <h2>Spindle:</h2>
172                 <p>Delay: 0.5s</p>
173             </td>
174             <td>
175                 <h3>Limit:</h3>
176                 <input type="number" id="limit" value="32.78">
177                 <h3>Free:</h3>
178                 <input type="number" id="free" value="03.00">
179             </td>
180         </tr>
181     </table>
182 </div>
183 </div>
184 <div class="footer">
185     <p>CNC-controller for Bungard CNC-machine</p>
186 </div>
187 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/
jquery.min.js"></script>
188 <script>
189     var saved = false;
190
191     setInterval(function(){
192         checkRequestedWebpage();
193     }, 5000);
194
195     function saveOrIgnoreChanges(){
196         if (saved == false){
197             var r = confirm("Save changes?");
198             if (r == true){
199                 receivingConfigurationData();
200             }
201         }
202         document.location.href = "/changeWebPage?webpage=Back";
203     }
204
205     function receivingConfigurationData(){
206         saved = true;
207         var index = [$("#coordinates").text(),$("#chord_angle").val(),
208             $("#limit").val(),$("#free").val()];
209         var url = "handleData?ConfigurationData=";
210         for (let i = 0; i < index.length; i++){
211             if(i == 0){
212                 var temp = index[i]
213             }else{
214                 var temp = "!" + index[i];
215             }
216             url += temp;
217         }
218         var xhttp = new XMLHttpRequest();
219         xhttp.open("GET", url);
220         xhttp.onload = function() {
221             var data = JSON.parse(xhttp.responseText);

```

```

222     console.log(data);
223     }
224     xhttp.send();
225     }
226
227     $(document).ready(function(){
228     $('input:checkbox').click(function() {
229     $('input:checkbox').not(this).prop('checked', false);
230     });
231     });
232
233     document.onkeypress = function(event) {
234     var x = event.charCode;
235     switch(x){
236     case 108:
237     document.location.href = "/changeWebPage?webpage=Load";
238     break;
239     case 98:
240     document.location.href = "/changeWebPage?webpage=Back";
241     break;
242     case 115:
243     receivingConfigurationData();
244     break;
245     case 99:
246     document.getElementById("chord_angle").focus();
247     break;
248     case 116:
249     document.getElementById("limit").focus();
250     break;
251     }
252     };
253     </script>
254     </body>
255     </html>
256 )=====";

```

C.0.4 Tools web page

```
1 const char TOOLS_page[] PROGMEM = R"=====(
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="author" content="Daan Ver Eecke">
6 <title>tools</title>
7 <style>
8 * {
9     box-sizing: border-box;
10 }
11 .row::after {
12     content: "";
13     clear: both;
14     display: table;
15 }
16 [class*="col-"] {
17     float: left;
18     padding: 15px;
19 }
20 html {
21     font-family: "Lucida Sans", sans-serif;
22 }
23 [class*="col-"] {
24     width: 100%;
25 }
26 @media only screen and (min-width: 600px) {
27     .col-s-1 {width: 8.33%;}
28     .col-s-2 {width: 16.66%;}
29     .col-s-3 {width: 25%;}
30     .col-s-4 {width: 33.33%;}
31     .col-s-5 {width: 41.66%;}
32     .col-s-6 {width: 50%;}
33     .col-s-7 {width: 58.33%;}
34     .col-s-8 {width: 66.66%;}
35     .col-s-9 {width: 75%;}
36     .col-s-10 {width: 83.33%;}
37     .col-s-11 {width: 91.66%;}
38     .col-s-12 {width: 100%;}
39 }
40 @media only screen and (min-width: 768px) {
41     .col-1 {width: 8.33%;}
42     .col-2 {width: 16.66%;}
43     .col-3 {width: 25%;}
44     .col-4 {width: 33.33%;}
45     .col-5 {width: 41.66%;}
46     .col-6 {width: 50%;}
47     .col-7 {width: 58.33%;}
48     .col-8 {width: 66.66%;}
49     .col-9 {width: 75%;}
50     .col-10 {width: 83.33%;}
51     .col-11 {width: 91.66%;}
52     .col-12 {width: 100%;}
53 }
54 body {
55     background-color: #c5cedd;
```

```

56     }
57     .button-group button {
58         background-color: #4CAF50;
59         border: none;
60         color: white;
61         padding: 15px 32px;
62         text-align: center;
63         text-decoration: none;
64         display: inline-block;
65         font-size: 16px;
66         cursor: pointer;
67         float: left;
68     }
69     button {
70         width: 33%;
71     }
72     .button-group button:hover {
73         background-color:#00802b;
74         color: #c5cedd;
75     }
76     b {
77         color: yellow;
78     }
79     .header {
80         background-color: #4CAF50;
81         color: #ffffff;
82         padding: 15px;
83     }
84     .footer {
85         background-color: #4CAF50;
86         color: #ffffff;
87         text-align: center;
88         font-size: 12px;
89         padding: 15px;
90     }
91     table {
92         width: 100%;
93     }
94     th {
95         height: 30px;
96         text-align: center;
97     }
98     td {
99         height: 30px;
100        text-align: center;
101        vertical-align: center;
102    }
103    input[type=number] {
104        width: 100%;
105        border:2px solid #aaa;
106        border-radius:4px;
107        height:30px;
108        display: inline-block;
109        text-align: center;
110        background-color: white;
111        color: #4CAF50;
112        transition:.3s

```

```

113     }
114     input[type=number]:focus{
115         border-color: #4CAF50;
116         border-shadow: 0 0 8px 0 #4CAF50;
117     }
118     </style>
119 </head>
120 <body>
121     <div class="header">
122         <h1>Tool Data</h1>
123     </div>
124     <div class="row button-group">
125         <div class="col-12 col-s-12">
126             <button id="load_button" onclick="window.location.href='/
changeWebPage?webpage=Load'"><b>L</b>oad</button>
127             <button id="save_button" onclick="receivingToolsData()"><b>S</b>
ave</button>
128             <button id="back_button" onclick="saveOrIgnoreChanges()"><b>B</b>
ack</button>
129         </div>
130     </div>
131     <div class="row">
132         <div class="col-12 col-s-12" overflow="auto">
133             <table id="tableData">
134                 <tr>
135                     <th>Pl.</th>
136                     <th><b>D</b>epth</th>
137                     <th><b>I</b>ter.</th>
138                     <th><b>X</b> Feed</th>
139                     <th><b>Z</b> Feed</th>
140                     <th>S<b>p</b>indle</th>
141                     <th>Start</th>
142                     <th>Used</th>
143                     <th>Tab.</th>
144                 </tr>
145                 <tr>
146                     <td> </td>
147                     <td>mm</td>
148                     <td>/n</td>
149                     <td>mm/min</td>
150                     <td>mm/min</td>
151                     <td>1/min</td>
152                     <td>Vect.</td>
153                     <td>mm</td>
154                     <td> </td>
155                 </tr>
156                 <tr>
157                     <td>01</td>
158                     <td><input type="number" id="depth_1" value="00.00"></td>
159                     <td><input type="number" id="itr_1" value="01"></td>
160                     <td><input type="number" id="X_feed_1" value="0050"></td>
161                     <td><input type="number" id="Z_feed_1" value="0050"></td>
162                     <td><input type="number" id="spindle_1" value="30000"></td>
163                     <td id="start_1">...</td>
164                     <td id="used_1">...</td>
165                     <td id="tab_1">...</td>
166                 </tr>

```



```

167 <tr>
168 <td>02</td>
169 <td><input type="number" id="depth_2" value="00.00"></td>
170 <td><input type="number" id="itr_2" value="01"></td>
171 <td><input type="number" id="X_feed_2" value="0050"></td>
172 <td><input type="number" id="Z_feed_2" value="0050"></td>
173 <td><input type="number" id="spindle_2" value="30000"></td>
174 <td id="start_2">...</td>
175 <td id="used_2">...</td>
176 <td id="tab_2">...</td>
177 </tr>
178 <tr>
179 <td>03</td>
180 <td><input type="number" id="depth_3" value="00.00"></td>
181 <td><input type="number" id="itr_3" value="01"></td>
182 <td><input type="number" id="X_feed_3" value="0050"></td>
183 <td><input type="number" id="Z_feed_3" value="0050"></td>
184 <td><input type="number" id="spindle_3" value="30000"></td>
185 <td id="start_3">...</td>
186 <td id="used_3">...</td>
187 <td id="tab_3">...</td>
188 </tr>
189 <tr>
190 <td>04</td>
191 <td><input type="number" id="depth_4" value="00.00"></td>
192 <td><input type="number" id="itr_4" value="01"></td>
193 <td><input type="number" id="X_feed_4" value="0050"></td>
194 <td><input type="number" id="Z_feed_4" value="0050"></td>
195 <td><input type="number" id="spindle_4" value="30000"></td>
196 <td id="start_4">...</td>
197 <td id="used_4">...</td>
198 <td id="tab_4">...</td>
199 </tr>
200 <tr>
201 <td>05</td>
202 <td><input type="number" id="depth_5" value="00.00"></td>
203 <td><input type="number" id="itr_5" value="01"></td>
204 <td><input type="number" id="X_feed_5" value="0050"></td>
205 <td><input type="number" id="Z_feed_5" value="0050"></td>
206 <td><input type="number" id="spindle_5" value="30000"></td>
207 <td id="start_5">...</td>
208 <td id="used_5">...</td>
209 <td id="tab_5">...</td>
210 </tr>
211 <tr>
212 <td>06</td>
213 <td><input type="number" id="depth_6" value="00.00"></td>
214 <td><input type="number" id="itr_6" value="01"></td>
215 <td><input type="number" id="X_feed_6" value="0050"></td>
216 <td><input type="number" id="Z_feed_6" value="0050"></td>
217 <td><input type="number" id="spindle_6" value="30000"></td>
218 <td id="start_6">...</td>
219 <td id="used_6">...</td>
220 <td id="tab_6">...</td>
221 </tr>
222 <tr>
223 <td>07</td>

```

```

224     <td><input type="number" id="depth_7" value="00.00"></td>
225     <td><input type="number" id="itr_7" value="01"></td>
226     <td><input type="number" id="X_feed_7" value="0050"></td>
227     <td><input type="number" id="Z_feed_7" value="0050"></td>
228     <td><input type="number" id="spindle_7" value="30000"></td>
229     <td id="start_7">...</td>
230     <td id="used_7">...</td>
231     <td id="tab_7">...</td>
232 </tr>
233 <tr>
234     <td>08</td>
235     <td><input type="number" id="depth_8" value="00.00"></td>
236     <td><input type="number" id="itr_8" value="01"></td>
237     <td><input type="number" id="X_feed_8" value="0050"></td>
238     <td><input type="number" id="Z_feed_8" value="0050"></td>
239     <td><input type="number" id="spindle_8" value="30000"></td>
240     <td id="start_8">...</td>
241     <td id="used_8">...</td>
242     <td id="tab_8">...</td>
243 </tr>
244     <td>09</td>
245     <td><input type="number" id="depth_9" value="00.00"></td>
246     <td><input type="number" id="itr_9" value="01"></td>
247     <td><input type="number" id="X_feed_9" value="0050"></td>
248     <td><input type="number" id="Z_feed_9" value="0050"></td>
249     <td><input type="number" id="spindle_9" value="30000"></td>
250     <td id="start_9">...</td>
251     <td id="used_9">...</td>
252     <td id="tab_9">...</td>
253 </tr>
254 <tr>
255     <td>10</td>
256     <td><input type="number" id="depth_10" value="00.00"></td>
257     <td><input type="number" id="itr_10" value="01"></td>
258     <td><input type="number" id="X_feed_10" value="0050"></td>
259     <td><input type="number" id="Z_feed_10" value="0050"></td>
260     <td><input type="number" id="spindle_10" value="30000"></td>
261     <td id="start_10">...</td>
262     <td id="used_10">...</td>
263     <td id="tab_10">...</td>
264 </tr>
265 <tr>
266     <td>11</td>
267     <td><input type="number" id="depth_11" value="00.00"></td>
268     <td><input type="number" id="itr_11" value="01"></td>
269     <td><input type="number" id="X_feed_11" value="0050"></td>
270     <td><input type="number" id="Z_feed_11" value="0050"></td>
271     <td><input type="number" id="spindle_11" value="30000"></td>
272     <td id="start_11">...</td>
273     <td id="used_11">...</td>
274     <td id="tab_11">...</td>
275 </tr>
276 <tr>
277     <td>12</td>
278     <td><input type="number" id="depth_12" value="00.00"></td>
279     <td><input type="number" id="itr_12" value="01"></td>
280     <td><input type="number" id="X_feed_12" value="0050"></td>

```

```

281         <td><input type="number" id="Z_feed_12" value="0050"></td>
282         <td><input type="number" id="spindle_12" value="30000"></td>
283         <td id="start_12">...</td>
284         <td id="used_12">...</td>
285         <td id="tab_12">...</td>
286     </tr>
287     <tr>
288         <td>13</td>
289         <td><input type="number" id="depth_13" value="00.00"></td>
290         <td><input type="number" id="itr_13" value="01"></td>
291         <td><input type="number" id="X_feed_13" value="0050"></td>
292         <td><input type="number" id="Z_feed_13" value="0050"></td>
293         <td><input type="number" id="spindle_13" value="30000"></td>
294         <td id="start_13">...</td>
295         <td id="used_13">...</td>
296         <td id="tab_13">...</td>
297     </tr>
298     <tr>
299         <td>14</td>
300         <td><input type="number" id="depth_14" value="00.00"></td>
301         <td><input type="number" id="itr_14" value="01"></td>
302         <td><input type="number" id="X_feed_14" value="0050"></td>
303         <td><input type="number" id="Z_feed_14" value="0050"></td>
304         <td><input type="number" id="spindle_14" value="30000"></td>
305         <td id="start_14">...</td>
306         <td id="used_14">...</td>
307         <td id="tab_14">...</td>
308     </tr>
309     <tr>
310         <td>15</td>
311         <td><input type="number" id="depth_15" value="00.00"></td>
312         <td><input type="number" id="itr_15" value="01"></td>
313         <td><input type="number" id="X_feed_15" value="0050"></td>
314         <td><input type="number" id="Z_feed_15" value="0050"></td>
315         <td><input type="number" id="spindle_15" value="30000"></td>
316         <td id="start_15">...</td>
317         <td id="used_15">...</td>
318         <td id="tab_15">...</td>
319     </tr>
320 </table>
321 </div>
322 </div>
323 <div class="footer">
324     <p>CNC-controller for Bungard CNC-machine</p>
325 </div>
326 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/
jquery.min.js"></script>
327 <script>
328     var saved = false;
329
330     function saveOrIgnoreChanges(){
331         if (saved == false){
332             var r = confirm("Save changes?");
333             if (r == true){
334                 receivingToolsData();
335             }
336         }

```

```

337     document.location.href = "/changeWebPage?webpage=Back";
338 }
339
340 function receivingToolsData(){
341     saved = true;
342     var index = [$("#depth_1").val(),$("#depth_2").val(),
343                 $("#depth_3").val(),$("#depth_4").val(),
344                 $("#depth_5").val(),$("#depth_6").val(),
345                 $("#depth_7").val(),$("#depth_8").val(),
346                 $("#depth_9").val(),$("#depth_10").val(),
347                 $("#depth_11").val(),$("#depth_12").val(),
348                 $("#depth_13").val(),$("#depth_14").val(),
349                 $("#depth_15").val(),
350                 $("#itr_1").val(),$("#itr_2").val(),
351                 $("#itr_3").val(),$("#itr_4").val(),
352                 $("#itr_5").val(),$("#itr_6").val(),
353                 $("#itr_7").val(),$("#itr_8").val(),
354                 $("#itr_9").val(),$("#itr_10").val(),
355                 $("#itr_11").val(),$("#itr_12").val(),
356                 $("#itr_13").val(),$("#itr_14").val(),
357                 $("#itr_15").val(),
358                 $("#X_feed_1").val(),$("#X_feed_2").val(),
359                 $("#X_feed_3").val(),$("#X_feed_4").val(),
360                 $("#X_feed_5").val(),$("#X_feed_6").val(),
361                 $("#X_feed_7").val(),$("#X_feed_8").val(),
362                 $("#X_feed_9").val(),$("#X_feed_10").val(),
363                 $("#X_feed_11").val(),$("#X_feed_12").val(),
364                 $("#X_feed_13").val(),$("#X_feed_14").val(),
365                 $("#X_feed_15").val(),
366                 $("#Z_feed_1").val(),$("#Z_feed_2").val(),
367                 $("#Z_feed_3").val(),$("#Z_feed_4").val(),
368                 $("#Z_feed_5").val(),$("#Z_feed_6").val(),
369                 $("#Z_feed_7").val(),$("#Z_feed_8").val(),
370                 $("#Z_feed_9").val(),$("#Z_feed_10").val(),
371                 $("#Z_feed_11").val(),$("#Z_feed_12").val(),
372                 $("#Z_feed_13").val(),$("#Z_feed_14").val(),
373                 $("#Z_feed_15").val(),
374                 $("#spindle_1").val(),$("#spindle_2").val(),
375                 $("#spindle_3").val(),$("#spindle_4").val(),
376                 $("#spindle_5").val(),$("#spindle_6").val(),
377                 $("#spindle_7").val(),$("#spindle_8").val(),
378                 $("#spindle_9").val(),$("#spindle_10").val(),
379                 $("#spindle_11").val(),$("#spindle_12").val(),
380                 $("#spindle_13").val(),$("#spindle_14").val(),
381                 $("#spindle_15").val(),
382                 $("#start_1").text(),$("#start_2").text(),
383                 $("#start_3").text(),$("#start_4").text(),
384                 $("#start_5").text(),$("#start_6").text(),
385                 $("#start_7").text(),$("#start_8").text(),
386                 $("#start_9").text(),$("#start_10").text(),
387                 $("#start_11").text(),$("#start_12").text(),
388                 $("#start_13").text(),$("#start_14").text(),
389                 $("#start_15").text(),
390                 $("#used_1").text(),$("#used_2").text(),
391                 $("#used_3").text(),$("#used_4").text(),
392                 $("#used_5").text(),$("#used_6").text(),
393                 $("#used_7").text(),$("#used_8").text(),

```

```

394         $("#used_9").text(),$("#used_10").text(),
395         $("#used_11").text(),$("#used_12").text(),
396         $("#used_13").text(),$("#used_14").text(),
397         $("#used_15").text(),
398         $("#tab_1").text(),$("#tab_2").text(),
399         $("#tab_3").text(),$("#tab_4").text(),
400         $("#tab_5").text(),$("#tab_6").text(),
401         $("#tab_7").text(),$("#tab_8").text(),
402         $("#tab_9").text(),$("#tab_10").text(),
403         $("#tab_11").text(),$("#tab_12").text(),
404         $("#tab_13").text(),$("#tab_14").text(),
405         $("#tab_15").text()
406     ];
407     var url = "handleToolsData?ToolsData=";
408     for (let i = 0; i < index.length; i++){
409         if(i == 0){
410             var temp = index[i]
411         }else{
412             var temp = "!" + index[i];
413         }
414         url += temp;
415     }
416     var xhttp= new XMLHttpRequest();
417     xhttp.open("GET", url);
418     xhttp.onload = function() {
419         var data = JSON.parse(xhttp.responseText);
420         console.log(data);
421     }
422     xhttp.send();
423 }
424
425 document.onkeypress = function(event) {
426     var x = event.charCode;
427     switch(x){
428         case 108:
429             document.location.href = "/changeWebPage?webpage=Load";
430             break;
431         case 98:
432             document.location.href = "/changeWebPage?webpage=Back";
433             break;
434         case 115:
435             receivingToolsData();
436             break;
437         case 100:
438             document.getElementById("depth_1").focus();
439             break;
440         case 105:
441             document.getElementById("itr_1").focus();
442             break;
443         case 120:
444             document.getElementById("X_feed_1").focus();
445             break;
446         case 122:
447             document.getElementById("Z_feed_1").focus();
448             break;
449         case 112:
450             document.getElementById("spindle_1").focus();

```

```
451         break;
452     }
453 };
454 </script>
455 </body>
456 </html>
457 )=====";
```

C.0.5 Offset web page

```
1 const char OFFSET_page[] PROGMEM = R"=====(
2 <html>
3   <head>
4     <meta name="author" content="Daan Ver Eecke">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0"
6   >
7     <title>offset</title>
8     <style>
9       * {
10        box-sizing: border-box;
11      }
12      .row::after {
13        content: "";
14        clear: both;
15        display: table;
16      }
17      [class*="col-"] {
18        float: left;
19        padding: 15px;
20      }
21      html {
22        font-family: "Lucida Sans", sans-serif;
23      }
24      [class*="col-"] {
25        width: 100%;
26      }
27      @media only screen and (min-width: 600px) {
28        .col-s-1 {width: 8.33%;}
29        .col-s-2 {width: 16.66%;}
30        .col-s-3 {width: 25%;}
31        .col-s-4 {width: 33.33%;}
32        .col-s-5 {width: 41.66%;}
33        .col-s-6 {width: 50%;}
34        .col-s-7 {width: 58.33%;}
35        .col-s-8 {width: 66.66%;}
36        .col-s-9 {width: 75%;}
37        .col-s-10 {width: 83.33%;}
38        .col-s-11 {width: 91.66%;}
39        .col-s-12 {width: 100%;}
40      }
41      @media only screen and (min-width: 768px) {
42        .col-1 {width: 8.33%;}
43        .col-2 {width: 16.66%;}
44        .col-3 {width: 25%;}
45        .col-4 {width: 33.33%;}
46        .col-5 {width: 41.66%;}
47        .col-6 {width: 50%;}
48        .col-7 {width: 58.33%;}
49        .col-8 {width: 66.66%;}
50        .col-9 {width: 75%;}
51        .col-10 {width: 83.33%;}
52        .col-11 {width: 91.66%;}
53        .col-12 {width: 100%;}
54      }
55      body {
```

```

55     background-color: #c5cedd;
56 }
57 button {
58     width: 100%;
59     background-color: #4CAF50;
60     color: #c5cedd;
61     font-size: 16px;
62     padding: 10px 24px;
63     -webkit-transition-duration: 0.4s; /* Safari */
64     transition-duration: 0.4s;
65 }
66 button:hover {
67     background-color:#c5cedd; /* Green */
68     color: #4CAF50;
69 }
70 .header {
71     background-color: #4CAF50;
72     color: #ffffff;
73     padding: 15px;
74 }
75 .footer {
76     background-color: #4CAF50;
77     color: #ffffff;
78     text-align: center;
79     font-size: 12px;
80     padding: 15px;
81 }
82 .container {
83     display: block;
84     position: relative;
85     padding-left: 35px;
86     margin-bottom: 12px;
87     cursor: pointer;
88     font-size: 22px;
89     -webkit-user-select: none;
90     -moz-user-select: none;
91     -ms-user-select: none;
92     user-select: none;
93 }
94 .container input {
95     position: absolute;
96     opacity: 0;
97     cursor: pointer;
98     height: 0;
99     width: 0;
100 }
101 .checkmark {
102     position: absolute;
103     top: 0;
104     left: 0;
105     height: 25px;
106     width: 25px;
107     background-color: #eee;
108 }
109 .container:hover input ~ .checkmark {
110     background-color: #ccc;
111 }

```



```

112     .container input:checked ~ .checkmark {
113         background-color: #4CAF50;
114     }
115     .checkmark:after {
116         content: "";
117         position: absolute;
118         display: none;
119     }
120     .container input:checked ~ .checkmark:after {
121         display: block;
122     }
123     .container .checkmark:after {
124         left: 9px;
125         top: 5px;
126         width: 5px;
127         height: 10px;
128         border: solid white;
129         border-width: 0 3px 3px 0;
130         -webkit-transform: rotate(45deg);
131         -ms-transform: rotate(45deg);
132         transform: rotate(45deg);
133     }
134     b {
135         color: yellow !important;
136     }
137     #button_table {
138         width: 100%;
139         height: 30%;
140     }
141     #button_td {
142         width: 33%;
143     }
144     #up_down_table {
145         vertical-align: center;
146         width: 100%;
147         height: 30%;
148     }
149     #up_down_td {
150         width: 100%;
151     }
152     #movement_button {
153         height: 100%;
154     }
155 </style>
156 </head>
157 <body>
158     <div class="header">
159         <h1>Offset</h1>
160     </div>
161     <div class="row">
162         <div class="col-12 col-s-12">
163             <button id="back_button" onclick="window.location.href='/
changeWebPage?webpage=Back '"><b>B</b>ack</button>
164         </div>
165     <div>
166     <div class="row">
167         <div class="col-6 col-s-12">

```

```

168     <h3>distance(mm): </h3>
169     <table>
170         <tr>
171             <td><input type="checkbox" name="skill" id="check1" class="
check" value="1"> 1</td>
172             <td><input type="checkbox" name="skill" id="check2" class="
check" value="10"> 10</td>
173             <td><input type="checkbox" name="skill" id="check3" class="
check" value="100"> 100</td>
174         </tr>
175     </table>
176 </div>
177 <div class="col-6 col-s-12">
178     <h3>current position(mm): </h3>
179     <table>
180         <tr>
181             <th>X: </th>
182             <td><p id="X_position">000,00</p></td>
183         </tr>
184         <tr>
185             <th>Y: </th>
186             <td><p id="Y_position">000,00</p></td>
187         </tr>
188         <tr>
189             <th>Z: </th>
190             <td><p id="Z_position">32.78</p></td>
191         </tr>
192     </table>
193 </div>
194 </div>
195 <div class="row">
196     <div class="col-9 col-s-12">
197         <table id="button_table">
198             <tr>
199                 <td id="button_td"> </td>
200                 <td id="button_td"><button id="movement_button" onclick="
receiveButtonsPressed('Back')">B<b>a</b>ckwards</button></td>
201                 <td id="button_td"> </td>
202             </tr>
203             <tr>
204                 <td id="button_td"><button id="movement_button" onclick="
receiveButtonsPressed('Left')"><b>L</b>eft</button></td>
205                 <td id="button_td"> </td>
206                 <td id="button_td"><button id="movement_button" onclick="
receiveButtonsPressed('Right')"><b>R</b>ight</button></td>
207             </tr>
208             <tr>
209                 <td id="button_td"> </td>
210                 <td id="button_td"><button id="movement_button" onclick="
receiveButtonsPressed('Forward')"><b>F</b>orwards</button></td>
211                 <td id="button_td"> </td>
212             </tr>
213         </table>
214     </div>
215     <div class="col-3 col-s-12">
216         <table id="up_down_table">
217             <tr>

```

```

218         <td id="up_down_td"><button id="movement_button" onclick="
receiveButtonsPressed('Up')"><b>U</b>p</button></td>
219     </tr>
220     <tr>
221         <td id="up_down_td"></td>
222     </tr>
223     <tr>
224         <td id="up_down_td"><button id="movement_button" onclick="
receiveButtonsPressed('Down')"><b>D</b>own</button></td>
225     </tr>
226 </table>
227 </div>
228 </div>
229 <div class="footer">
230     <p>CNC-controller for Bungard CNC-machine</p>
231 </div>
232 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.
min.js"></script>
233 <script>
234     function homing_axes(btn){
235         console.log("homing_axes");
236         alert("Be carefull, machine will move");
237         if(btn == "Ok"){
238             document.location.href = "/changeWebPage?webpage=HomingAxes";
239         }
240     }
241
242     function readCheckbox(){
243         if (document.getElementById("check1").checked){
244             var checkbox = "1";
245         } else if (document.getElementById("check2").checked) {
246             checkbox = "10";
247         } else if (document.getElementById("check3").checked){
248             checkbox = "100";
249         } else {
250             checkbox = "";
251         }
252         return checkbox;
253     }
254
255     function resetCheckbox(){
256         document.getElementById("check1").checked = false;
257         document.getElementById("check2").checked = false;
258         document.getElementById("check3").checked = false;
259         document.getElementById("check4").checked = false;
260     }
261
262     function setCheckbox(checkbox){
263         resetCheckbox();
264         if (checkbox == "1"){
265             document.getElementById("check1").checked = true;
266         } else if (checkbox == "10") {
267             document.getElementById("check2").checked = true;
268         } else if (checkbox == "100"){
269             document.getElementById("check3").checked = true;
270         } else {
271             document.getElementById("check1").checked = false;

```

```

272     document.getElementById("check2").checked = false;
273     document.getElementById("check3").checked = false;
274 }
275 }
276
277 $(document).ready(function(){
278     $('input:checkbox').click(function() {
279         $('input:checkbox').not(this).prop('checked', false);
280     });
281 });
282
283 function receiveButtonsPressed(buttonName){
284     var check = readCheckbox();
285     var xhttp = new XMLHttpRequest();
286     if (check != ""){
287         var url = "handleButtonsPressed?MovementAxes" + "=" + check + "!"
+ buttonName;
288         xhttp.onreadystatechange = function() {
289             if (this.readyState == 4 && this.status == 200) {
290                 }
291             };
292             xhttp.open("GET", url , true);
293             xhttp.send();
294         }
295     }
296
297     document.onkeypress = function(event) {
298         var x = event.charCode;
299         switch(x){
300             case 98:
301                 homing_axes('Ok');
302                 break;
303             case 117:
304                 receiveButtonsPressed('Up');
305                 break;
306             case 100:
307                 receiveButtonsPressed('Down');
308                 break;
309             case 108:
310                 receiveButtonsPressed('Left');
311                 break;
312             case 114:
313                 receiveButtonsPressed('Right');
314                 break;
315             case 102:
316                 receiveButtonsPressed('Forwards');
317                 break;
318             case 97:
319                 receiveButtonsPressed('Backwards');
320                 break;
321         }
322     };
323 </script>
324 </body>
325 </html>
326 )=====";

```

C.0.6 Load web page

```
1 const char LOAD_page[] PROGMEM = R"=====(
2 <html>
3   <head>
4     <meta name="author" content="Daan Ver Eecke">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0"
6   >
7     <title>load</title>
8     <style>
9       * {
10        box-sizing: border-box;
11      }
12      .row::after {
13        content: "";
14        clear: both;
15        display: table;
16      }
17      [class*="col-"] {
18        float: left;
19        padding: 15px;
20      }
21      html {
22        font-family: "Lucida Sans", sans-serif;
23      }
24      /* For mobile phones: */
25      [class*="col-"] {
26        width: 100%;
27      }
28      @media only screen and (min-width: 600px) {
29        .col-s-1 {width: 8.33%;}
30        .col-s-2 {width: 16.66%;}
31        .col-s-3 {width: 25%;}
32        .col-s-4 {width: 33.33%;}
33        .col-s-5 {width: 41.66%;}
34        .col-s-6 {width: 50%;}
35        .col-s-7 {width: 58.33%;}
36        .col-s-8 {width: 66.66%;}
37        .col-s-9 {width: 75%;}
38        .col-s-10 {width: 83.33%;}
39        .col-s-11 {width: 91.66%;}
40        .col-s-12 {width: 100%;}
41      }
42      @media only screen and (min-width: 768px) {
43        .col-1 {width: 8.33%;}
44        .col-2 {width: 16.66%;}
45        .col-3 {width: 25%;}
46        .col-4 {width: 33.33%;}
47        .col-5 {width: 41.66%;}
48        .col-6 {width: 50%;}
49        .col-7 {width: 58.33%;}
50        .col-8 {width: 66.66%;}
51        .col-9 {width: 75%;}
52        .col-10 {width: 83.33%;}
53        .col-11 {width: 91.66%;}
54        .col-12 {width: 100%;}
55      }
56    </style>
57  </head>
58  <body>
59    <div class="row">
60      <div class="col-1">1</div>
61      <div class="col-1">1</div>
62      <div class="col-1">1</div>
63      <div class="col-1">1</div>
64      <div class="col-1">1</div>
65      <div class="col-1">1</div>
66      <div class="col-1">1</div>
67      <div class="col-1">1</div>
68      <div class="col-1">1</div>
69      <div class="col-1">1</div>
70      <div class="col-1">1</div>
71      <div class="col-1">1</div>
72      <div class="col-1">1</div>
73    </div>
74  </body>
75 </html>
76 )=====";
```

```

55     body {
56         background-color: #c5cedd;
57     }
58     .button-group button {
59         background-color: #4CAF50;
60         border: none;
61         color: white;
62         padding: 15px 32px;
63         text-align: center;
64         text-decoration: none;
65         display: inline-block;
66         font-size: 16px;
67         cursor: pointer;
68         float: left;
69     }
70     button {
71         width: 50%;
72     }
73     .button-group button:hover {
74         background-color:#00802b;
75         color: #c5cedd;
76     }
77     button:disabled {
78         display:none;
79     }
80     b {
81         color: yellow;
82     }
83     .header {
84         background-color: #4CAF50;
85         color: #ffffff;
86         padding: 15px;
87     }
88     .footer {
89         background-color: #4CAF50;
90         color: #ffffff;
91         text-align: center;
92         font-size: 12px;
93         padding: 15px;
94     }
95     #emergency_stop {
96         background-color: #ff0000;
97         border: 1px solid green;
98         color: #c5cedd;
99         text-align: center;
100        text-decoration: none;
101        display: inline-block;
102        display: block;
103        width: 100%;
104        height: 5%;
105    }
106    #emergency_stop:disabled {
107        display:none;
108    }
109    #emergency_stop:hover {
110        background-color:#c5cedd;
111        color: #ff0000;

```

```

112     }
113     </style>
114 </head>
115 <body>
116     <div class="header">
117         <h1>Load</h1>
118     </div>
119     <div class="row">
120         <div class="col-12 col-s-12 button-group">
121             <button id="back_button" onclick="window.location.href='/
changeWebPage?webpage=Back'"><b>B</b>ack</button>
122             <button id="start_button" onclick="send_file()" disabled><b>S</b>
tart</button>
123         </div>
124     </div>
125     <div class="row">
126         <div class="col-8 col-s-12">
127             <h1>Select file to Upload</h1>
128             <input id="file" type="file" accept=".plt, .cnd"/>
129         </div>
130         <div class="col-4 col-s-12">
131             <table style="width:100%">
132                 <tr>
133                     <td>File name: </td>
134                     <td id="file_name" >No file selected</td>
135                 </tr>
136                 <tr>
137                     <td>File extension: </td>
138                     <td id="file_extension">...</td>
139                 </tr>
140                 <tr>
141                     <td>File size: </td>
142                     <td id="file_size">...</td>
143                 </tr>
144             </table>
145         </div>
146     </div>
147     <div class="row">
148         <div class="col-12 col-s-12">
149             <button id="emergency_stop" disabled><b>E</b>MERGENCY BUTTON</
button>
150         </div>
151     </div>
152     <div class="footer">
153         <p>CNC-controller for Bungard CNC-machine</p>
154     </div>
155     <script>
156         const input = document.querySelector('input[type="file"]')
157         input.addEventListener('change', function (e) {
158             const reader = new FileReader()
159             reader.onload = function (){
160                 var file;
161                 file = reader.result;
162                 file_selected();
163             }
164             reader.readAsText(input.files[0])
165         }, false)

```

```

166
167     function file_selected(){
168         var name_path = document.getElementById('file').files[0].name.
split(".");
169         document.getElementById("start_button").disabled = false;
170         document.getElementById("emergency_stop").disabled = false;
171         document.getElementById("file_name").innerHTML = name_path[0];
172         document.getElementById("file_extension").innerHTML = name_path
[1];
173         document.getElementById("file_size").innerHTML = document.
getElementById('file').files[0].size;
174     }
175
176     var numberOfHTTPRequestsdone = 0;
177
178     function send_file(){
179         var file_size = file.length;
180         var f_size = file_size
181         var numberOfHTTPRequest = 1;
182         for (let i = 0; 1000 < file_size;i++){
183             numberOfHTTPRequest ++;
184             file_size -= 1000;
185         }
186         console.log(file_size);
187         console.log(numberOfHTTPRequest);
188
189         requests = new Array(numberOfHTTPRequest);
190         for (let i = 0; i < numberOfHTTPRequest; i++){
191             for (let i = 0; i < 3; i++){
192                 }
193             numberOfHTTPRequestsdone++;
194             var sub_string = file.slice(i*1000,((i+1)*1000)-1);
195             var sub_string = sub_string.replace(/;/g,"!");
196             if (i == 0){
197                 var url = "handleFileUpload?file" + i + "=" + f_size + "!" +
sub_string;
198             }else{
199                 var url = "handleFileUpload?file" + i + "=" + sub_string;
200             }
201             requests[i] = new XMLHttpRequest();
202         }
203     }
204
205     document.onkeypress = function(event) {
206         var x = event.charCode;
207         switch(x){
208             case 98:
209                 document.location.href = "/changeWebPage?webpage=Back";
210                 break;
211             case 115:
212                 document.location.href = "/changeWebPage?webpage=Start";
213                 break;
214             case 101:
215                 alert("EMERGENCY STOP!");
216                 break;
217         }
218     };

```



```
219     </script>
220     </body>
221 </html>
222 )=====";
```


Appendix D

Arduino source code

D.0.1 Libraries and variables

```
1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WebServer.h>
4 #include <ESP8266mDNS.h>
5 #include <ESP8266WiFiMulti.h>
6 #include <FS.h>
7 #include <SoftwareSerial.h>
8
9 //webpages
10 #include "z_routepro_drillpro.h"
11 #include "z_config.h"
12 #include "z_load.h"
13 #include "z_tools.h"
14 #include "z_offset.h"
15 #include "z_welcome.h"
16
17 ESP8266WiFiMulti wifiMulti;
18 ESP8266WebServer server (80);
19
20 #define INTEG(N) (payload[N]|(payload[N+1]<<8)|(payload[N+2]<<16)|
21 payload[N+3]<<24))
22 #define I2B(i, N) send_payload[N]=i & 0xFF;send_payload[N+1]=(i>>8) & 0
23 xFF;send_payload[N+2]=(i>>16) & 0xFF;send_payload[N+3]=(i>>24) & 0xFF;
24 char send_payload[253];
25 char payload[253];
26
27 #define STASSID "Wifi_Lab_Proyectos"
28 #define STAPSK "R3B00TY0URM1ND"
29 //#define STASSID "MIWIFI_2G_Y3mu"
30 //#define STAPSK "tUPMfUKv"
31 const char* ssid = STASSID;
32 const char* password = STAPSK;
33
34 String username = "root";
35 String pass = "admin";
36 bool login = false;
37
38 char web_page;
39
40 char requested_web_page = '\0';
```

```

39
40 bool RDdataReady = false;
41 bool CdataReady = false;
42 bool TdataReady = false;
43
44 String routepro_drillpro_variables[] = {"0","325","0","0","278","0","15","2
    ","0","0","","0","0","0","No file selected"};
45 int routepro_drillpro_arrayLength = 15;
46
47 String configuration_variables[] = {"...","05","32,78","03,00"};
48 int configuration_arrayLength = 4;
49
50 String offset_variables[] = {"000,00","000,00","32,78"};
51 String movement_distance = "";
52 String movement_direction = "";
53 int offset_arrayLength = 3;
54
55 int tool_arrayLengths = 15;
56 String tool_depths[] = {"00,00","00,00","00,00","00,00","00,00","00,00","
    00,00","00,00","00,00","00,00","00,00","00,00","00,00","00,00","00,00"};
57 String tool_iterations[] = {"01","01","01","01","01","01","01","01","01","
    01","01","01","01","01","01"};
58 String tool_X_feeds[] = {"0050","0050","0050","0050","0050","0050","0050","
    0050","0050","0050","0050","0050","0050","0050","0050"};
59 String tool_Z_feeds[] = {"0050","0050","0050","0050","0050","0050","0050","
    0050","0050","0050","0050","0050","0050","0050","0050"};
60 String tool_spindles[] = {"30000","30000","30000","30000","30000","30000","
    30000","30000","30000","30000","30000","30000","30000","30000","30000"};
61 String tool_starts[] = {"...","...","...","...","...","...","...","...","
    ...","...","...","...","...","...","..."};
62 String tool_useds[] = {"...","...","...","...","...","...","...","...","
    ...","...","...","...","...","...","..."};
63 String tool_tabs[] = {"...","...","...","...","...","...","...","...","
    ...","...","...","...","...","...","..."};

```

D.0.2 Void setup

```
1 void setup ( void ) {
2   Serial.begin ( 115200 );
3   WiFi.mode ( WIFI_STA );
4   WiFi.begin ( ssid, password );
5
6   server.on("/Login", handleLogin);
7   server.on ( "/", changeWebPage );
8   server.on("/changeWebPage", changeWebPage);
9   server.on("/handleData", handleData);
10  server.on("/handleToolsData", handleToolsData);
11  server.on("/handleButtonsPressed",handleButtonsPressed);
12
13  server.onNotFound ( handleNotFound );
14  server.begin();
15  Serial.println ( "HTTP server started" );
16 }
```

D.0.3 Void loop

```
1 void loop ( void ) {
2   server.handleClient();
3   checkSerialDataAvaibleToSend();
4 }
```

D.0.4 Handle web pages

```
1 void handleLogin(){
2   String arguments = server.arg("credentials");
3   String receivedUser = getValueFromString(arguments, '!',0);
4   String receivedPass = getValueFromString(arguments, '!',1);
5   String payload = "";
6   if(receivedUser == username && receivedPass == pass){
7     payload = "correct";
8     login = true;
9   }else{
10    payload = "wrong";
11    login = false;
12  }
13  server.send(200, "text/plain", payload);
14 }
15
16 void changeWebPage(){
17   String s = "";
18   if (login == true){
19     String command = server.arg("webpage");
20     char selector = command.charAt(0);
21     switch(selector){
22       case 'R':
23         web_page = 'R';
24         s = ROUTEPRO_page;
25         break;
26       case 'C':
27         web_page = 'C';
28         s = CONFIG_page;
29         break;
30       case 'T':
31         web_page = 'T';
32         s = TOOLS_page;
33         break;
34       case 'L':
35         web_page = 'L';
36         s = LOAD_page;
37         break;
38       case 'B':
39         web_page = 'R';
40         s = ROUTEPRO_page;
41         break;
42       case 'O':
43         web_page = 'O';
44         s = OFFSET_page;
45         break;
46     }
47   }else{
48     web_page = 'W';
49     s = WELCOME_page;
50   }
51   server.send(200, "text/html", s);
52 }
53
54 void handleNotFound() {
55   String message = "File Not Found\n\n";
```

```
56 message += "URI: ";
57 message += server.uri();
58 message += "\nMethod: ";
59 message += ( server.method() == HTTP_GET ) ? "GET" : "POST";
60 message += "\nArguments: ";
61 message += server.args();
62 message += "\n";
63 for ( uint8_t i = 0; i < server.args(); i++ ) {
64     message += " " + server.argName ( i ) + ": " + server.arg ( i ) + "\n";
65 }
66 server.send ( 404, "text/plain", message );
67 }
```

D.0.5 Read data

```
1 void handleData(){
2   switch(web_page){
3     case 'R':
4       readData(routepro_drillpro_variables , routepro_drillpro_arrayLength ,
5         "RouteproData");
6       RDdataReady = true;
7       break;
8     case 'C':
9       readData(configuration_variables , configuration_arrayLength , "
10      ConfigurationData");
11       CdataReady = true;
12       break;
13   }
14 }
15
16 void readData (String dataArray[], int arrayLength, String webpage){
17   String webPageData = server.arg(webpage);
18   for(int i = 0; i < arrayLength; i++){
19     dataArray[i] = getValueFromString(webPageData, '!', i);
20   }
21 }
22
23 void handleToolsData(){
24   String Toolsdata = server.arg("ToolsData");
25   for(int i = 0; i < tool_arrayLengths; i++){
26     tool_depths[i] = getValueFromString(Toolsdata, '!', i);
27     tool_iterations[i] = getValueFromString(Toolsdata, '!', i+15);
28     tool_X_feeds[i] = getValueFromString(Toolsdata, '!', i+30);
29     tool_Z_feeds[i] = getValueFromString(Toolsdata, '!', i+45);
30     tool_spindles[i] = getValueFromString(Toolsdata, '!', i+60);
31     tool_starts[i] = getValueFromString(Toolsdata, '!', i+75);
32     tool_usesd[i] = getValueFromString(Toolsdata, '!', i+90);
33     tool_tabs[i] = getValueFromString(Toolsdata, '!', i+105);
34   }
35   TdataReady = true;
36 }
37
38 void handleButtonsPressed(){
39   String movement = server.arg("MovementAxes");
40   movement_distance = getValueFromString(movement, '!', 0);
41   movement_direction= getValueFromString(movement, '!', 1);
42   serialSendButtons();
43 }
```


D.0.6 Serial communication

```
1 void checkSerialDataAvaibleToSend(){
2   if (RDdataReady == true){
3     makeRouteproDrillproPayload();
4     RDdataReady = false;
5   } else if (CdataReady == true){
6     makeConfigurationPayload();
7     CdataReady = false;
8   } else if (TdataReady == true){
9     makeToolPayload();
10    TdataReady = false;
11  }
12 }
13
14 void makeRouteproDrillproPayload(){
15   int arrayLength = routepro_drillpro_arrayLength;
16   int values[arrayLength-1];
17   for(int i = 0; i < arrayLength-1; i++){
18     if(i == 6 || i == 7){
19       values[i] = routepro_drillpro_variables[i].toFloat()*10;
20     }else{
21       values[i] = routepro_drillpro_variables[i].toInt();
22     }
23   }
24   int numberOfInts = sizeof(values)/sizeof(values[0]);
25   send_payload[0] = 0xC3;
26   send_payload[1] = 2;
27   for(int i = 0; i < numberOfInts; i++){
28     I2B(values[i],2+i*4);
29   }
30   constructFrame(send_payload,58);
31 }
32
33 void makeConfigurationPayload(){
34   int arrayLength = configuration_arrayLength;
35   int values[arrayLength];
36   for(int i = 0; i < arrayLength; i++){
37     values[i] = configuration_variables[i+1].toInt();
38   }
39   int numberOfInts = sizeof(values)/sizeof(values[0]);
40   send_payload[0] = 0xC3;
41   send_payload[1] = 1;
42   for(int i = 0; i < numberOfInts; i++){
43     I2B(values[i],2+i*4);
44   }
45   constructFrame(send_payload,14);
46 }
47
48 void makeToolPayload(){
49   send_payload[0] = 0xC3;
50   send_payload[1] = 3;
51   for(int i = 0; i < 5; i++){
52     send_payload[2+i*24] = i+1;
53     I2B(tool_depths[i].toInt(),6+i*24);
54     I2B(tool_iterations[i].toInt(),10+i*24);
55     I2B(tool_X_feeds[i].toInt(),14+i*24);
```

```

56     I2B(tool_Z_feeds[i].toInt(),18+i*24);
57     I2B(tool_spindles[i].toInt(),22+i*24);
58 }
59 constructFrame(send_payload,122);
60 }
61
62 void serialSendButtons(){
63     char distance = determineDistance(movement_distance);
64     char dir = movement_direction.charAt(0);
65     char payload[4];
66     payload[0] = 0xC3;
67     payload[1] = 6;
68     payload[2] = dir;
69     payload[3] = distance;
70     constructFrame(payload,4);
71 }
72
73 char determineDistance(String distance){
74     char dis = 0;
75     if(distance == "1"){
76         dis = 1;
77     } else if (distance == "10"){
78         dis = 10;
79     }else{
80         dis = 100;
81     }
82     return dis;
83 }
84
85 void constructFrame(char payload[], int payload_length){
86     int Size = payload_length;
87     char payloadSize = char(Size);
88     char frame[Size+3];
89     frame[0] = 0x24;
90     frame[1] = payloadSize;
91     for (int i = 0; i < Size; i++){
92         frame[i+2] = payload[i];
93     }
94     char polynomial = 0b10101010;
95     frame[Size+2] = CRC8(frame, polynomial, Size+1);
96     serialSendFrame(frame, payload_length+3);
97     frame[0] = 0;
98 }
99
100 void serialSendFrame(char frame[], int frame_length){
101     for(int i = 0; i < frame_length; i++){
102         Serial.write(frame[i]);
103     }
104 }

```

D.0.7 General functions

```
1 char CRC8(char *s, char polinomio, int len) {
2   unsigned char crc=0, datum;
3   int i, j;
4   for(i=0;i<len;i++) {
5     datum = s[i];
6     for(j=0;j<8;j++) {
7       if(((crc^datum)&0x01)==0) crc >>=1;
8       else {
9         crc ^= polinomio;
10        crc >>= 1;
11        crc |= 0x80;
12      }
13      datum >>=1;
14    }
15  }
16  return crc;
17 }
18
19 String getValueFromString(String data, char separator, int index){
20   int found = 0;
21   int strIndex[] = {0, -1};
22   int maxIndex = data.length()-1;
23   for(int i=0; i<=maxIndex && found<=index; i++){
24     if(data.charAt(i)==separator || i==maxIndex){
25       found++;
26       strIndex[0] = strIndex[1]+1;
27       strIndex[1] = (i == maxIndex) ? i+1 : i;
28     }
29   }
30   return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
31 }
```

