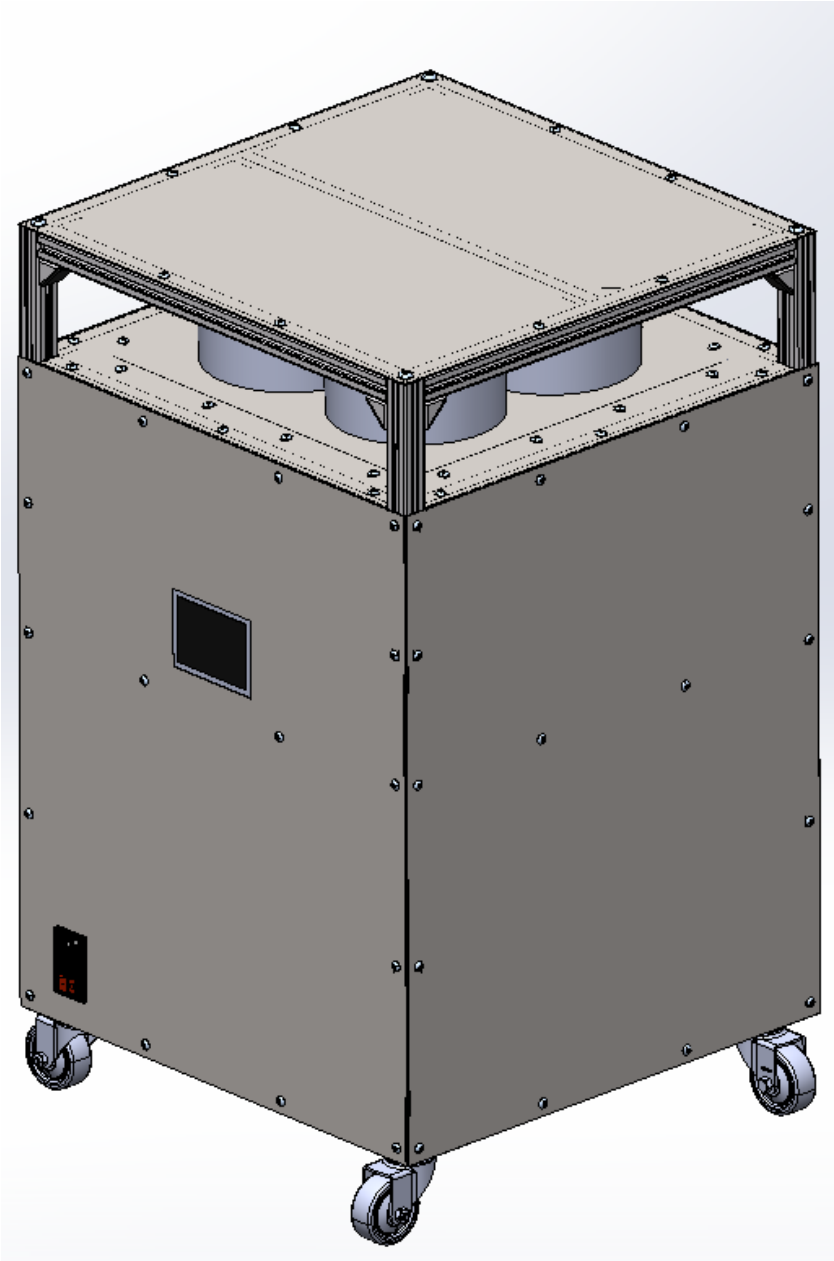
National Research Programme “Covid-19 consequences mitigation”
project VPP-COVID-2020/1-0004 “Integration of reliable technologies for protection against Covid-19 in healthcare and high risk areas”

Equipment for air and surface disinfection with ozone and UV radiation

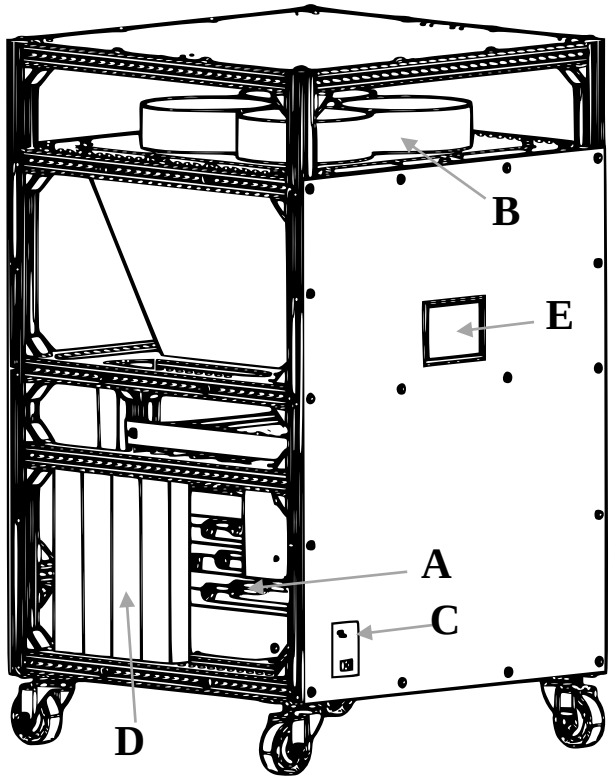
EXPLOITATION RULES



About the machine

The device is designed for disinfection of surfaces and air with UV radiation and ozone. The device uses 28 mercury lamps. They emit UV radiation (254nm) and ozone. The lamps are placed horizontally in a vertical shaft with mirrored walls. With the help of four fans, the room air enters the shaft, moves down along the UV radiation lamps, is disinfected and flows out at the bottom of the unit. UV radiation simultaneously irradiates and disinfects the floor under the unit. The construction of the device is designed so that the UV radiation emitted by the lamps cannot come into direct contact with the human eye. Only UV radiation reflected from the floor or structural elements of the equipment can get into the eyes. Preliminary tests have shown that the machine effectively destroys Escherichia coli, Staphylococcus aureus, bacteriophage phi6 and Semliki forest virus. Measurements of ozone concentration in the room during the operation of the equipment have not been performed yet. Equipment technology development level: TRL7.

Main technical parametres



UV radiation lamps (A):

- count - 28;
- filling- mercury;
- wavelength corresponding to the maximum radiation intensity - 254 nm;
- single lamps power- 11W

Fans (B):

- count- 4;
- productivity of one fan - 200m^3/h
- power of one fan 16W

Supply voltage: AC 200-240V (C)

Maximum power consumption:

- in the moment of switching on 450W
- work regime 250W

Automatic control system:

- ArduinoMega2560 microcontroller
- UV radiation lamp launchers (1 launcher to 4 lamps, 4x14W) (D)
- Touchscreen 2,8 inches for device controls (E)

Dimensions: length / width / height - 400mm / 400mm / 670mm

Weight: 25kg

Safety rules

Individual caution

Do not look at the appliance without goggles during operation of the UV lamps, as the reflected UV radiation from the floor and the components of the appliance can damage your eyesight.

Exploitation

It is forbidden to operate the device until the efficiency of its ozone production has been studied. In hospitals, ozone concentrations must not exceed 50 mg / m3.

During the operation of the equipment, observe the general requirements for 200-240V electrical installations.

The appliance is not intended for use in contact with water.

The machine must be positioned stably, so that all four wheels are supported against a hard surface.

Servicing

Repairs and replacement of lamps may only be performed by certified specialists.

When repairing the equipment and the UV lamps are switched on, it is mandatory to use appropriate goggles to protect the eyes from 254nm UV radiation and dense clothing to prevent skin burns.

UV lamps must be replaced every 10,000 operating hours.

Tranportation

When transporting the device, it must be taken into account that the UV lamps are a fragile load, and the device must be placed on a soft surface to prevent possible vibrations and shocks.

The machine must be secured with straps / cables to prevent the possibility of falling, no other objects / equipment must be on the machine.

When transporting the device, it must be taken into account that the touch screen must not be supported against a wall / other device / solid object to prevent the display from being damaged.

Storage

If the appliance is to be stored for a long time, make sure that it is not exposed to moisture and dust.

The machine must be placed stably (all four wheels are on a hard surface) to prevent them from falling.

Disposal

Used mercury lamps must not be disposed of with general waste. They must be taken to hazardous waste collection points.

User manual

Preparation for use

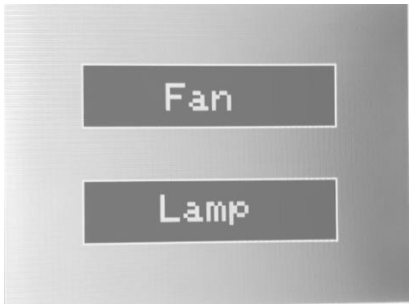
To ensure the operation of the device, it is necessary to connect it to the AC mains with a voltage of 200-240V.

To turn the machine on or off, press the on / off switch.

Controls

The device is controlled through the Arduino Mega 2560 microcontroller and touch screen. Each fan and launcher are controlled separately. This means that different operating modes of the device can be realized - it is possible to turn on launchers and fans in different combinations.

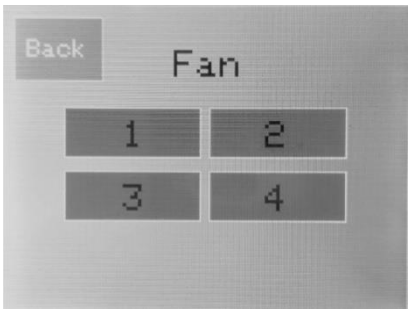
Home screen:



The home screen has to buttons:

1. Fan controls (Fan).
2. Lamp controls (Lamp).

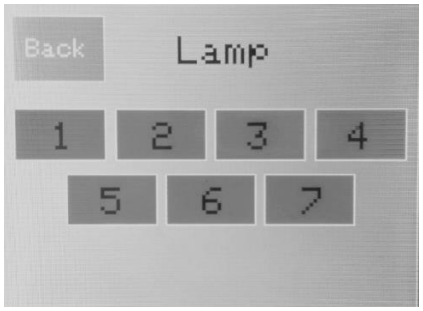
Fan controls:



The screen “Fan” has four buttons, each one controls a fan

If the fan is off, the button is red, when the fan is on, its color changes to green.

Lamp controls:

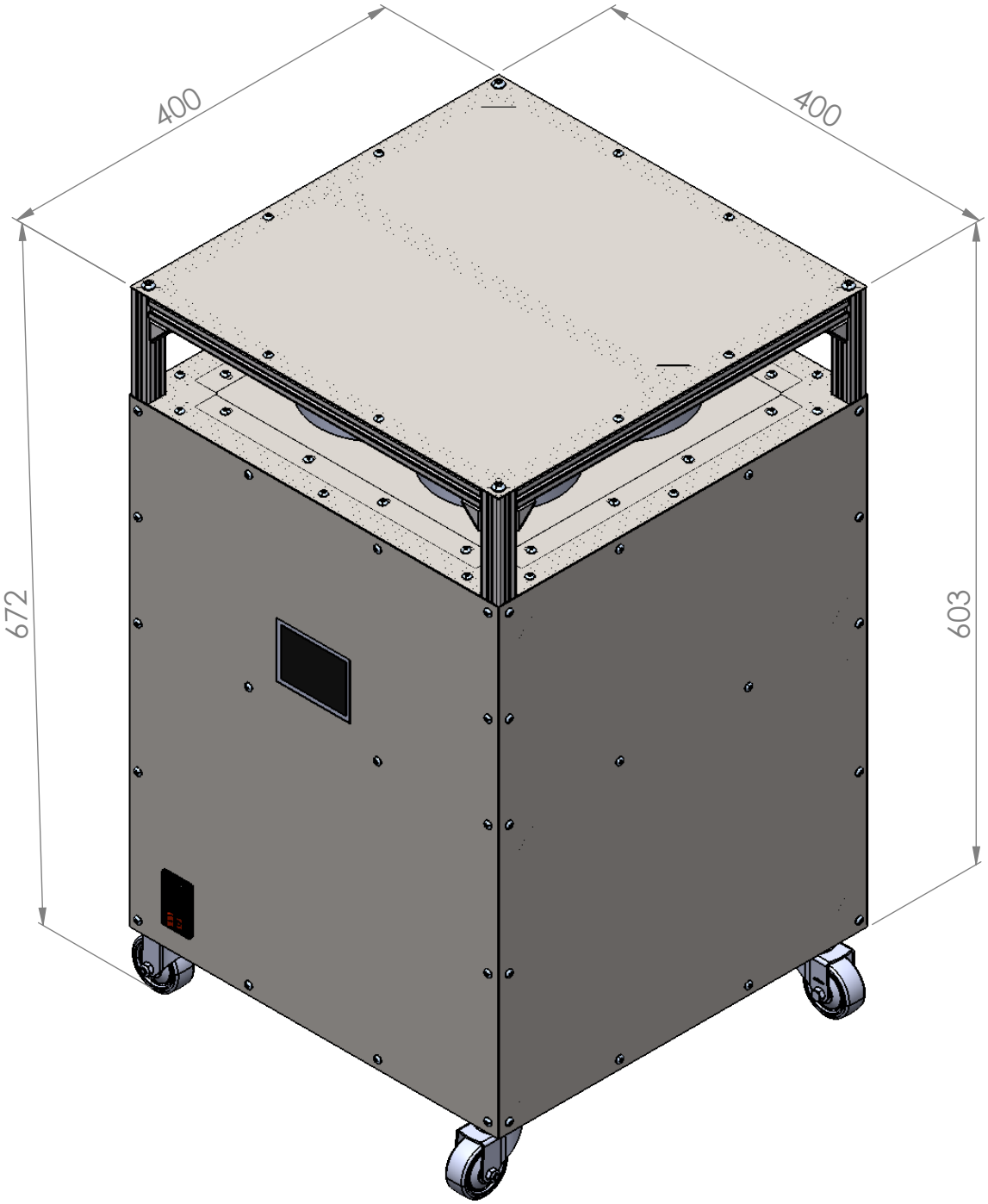


The screen “Lamp” has seven buttons, each one controls one launcher.

If the launcher is off, the button will turn red, when the launcher is on, it will turn green.

Back to home screen:

To return to the home screen from the "Fan" or "Lamp" screen, press the "Back" button in the upper left corner of the screen. After pressing the button, the home screen will return.



	NAME	SIGNATURE	DATE		
DRAWN	R. Rēvalds		24.11.2020		
CHK'D	V. Piziča		24.11.2020		

RTA
RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:
Equipment for air and surface disinfection with ozone and UV radiation

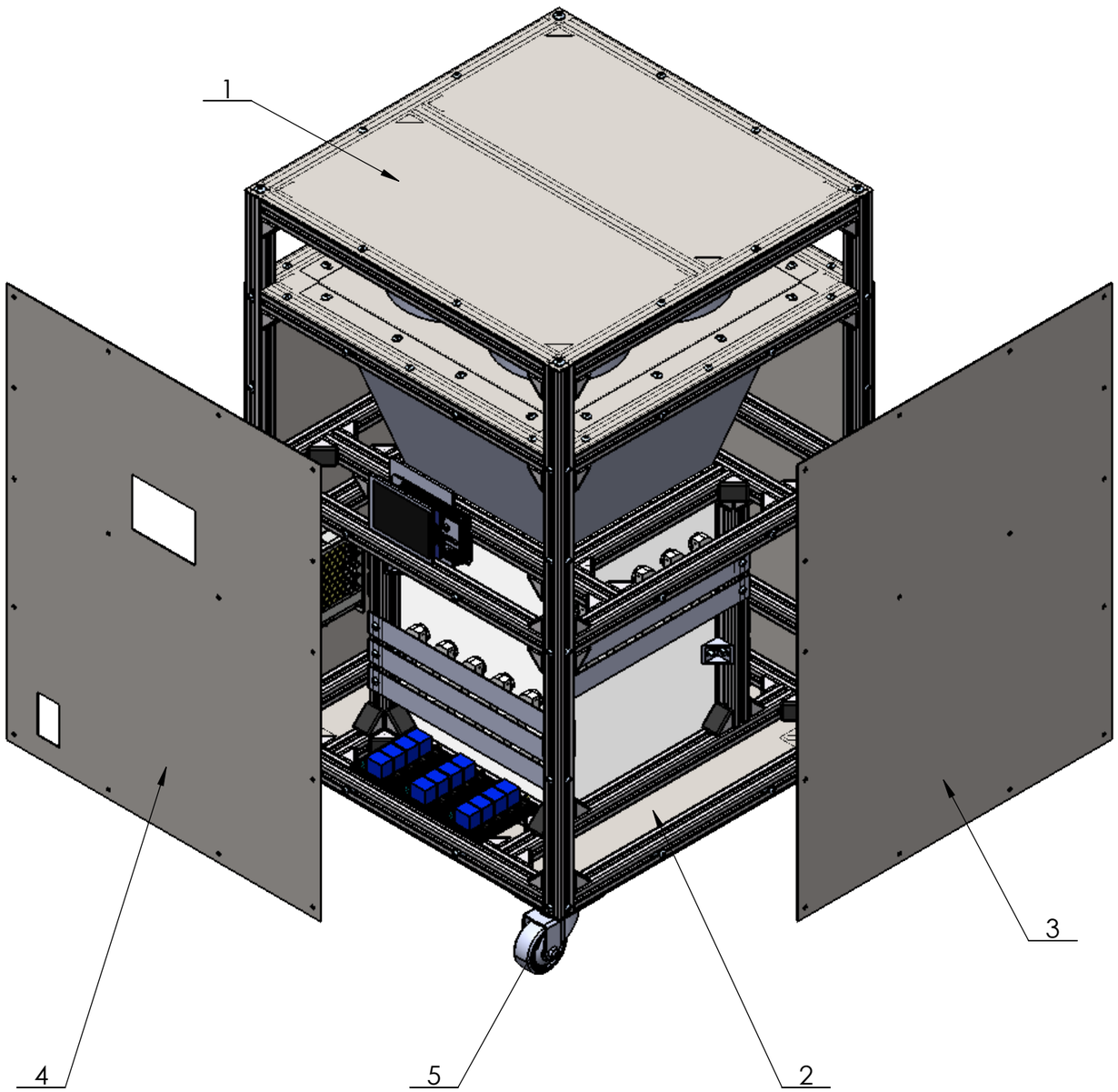
MATERIAL:

DWG NO.
C-19 UV 00.000.000.001

A4

SCALE:1:5

SHEET 1 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Top wall		1
2	Buttom wall		1
3	Side wall		3
4	Front wall		1
5	Cast roller		4

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE: Outer walls

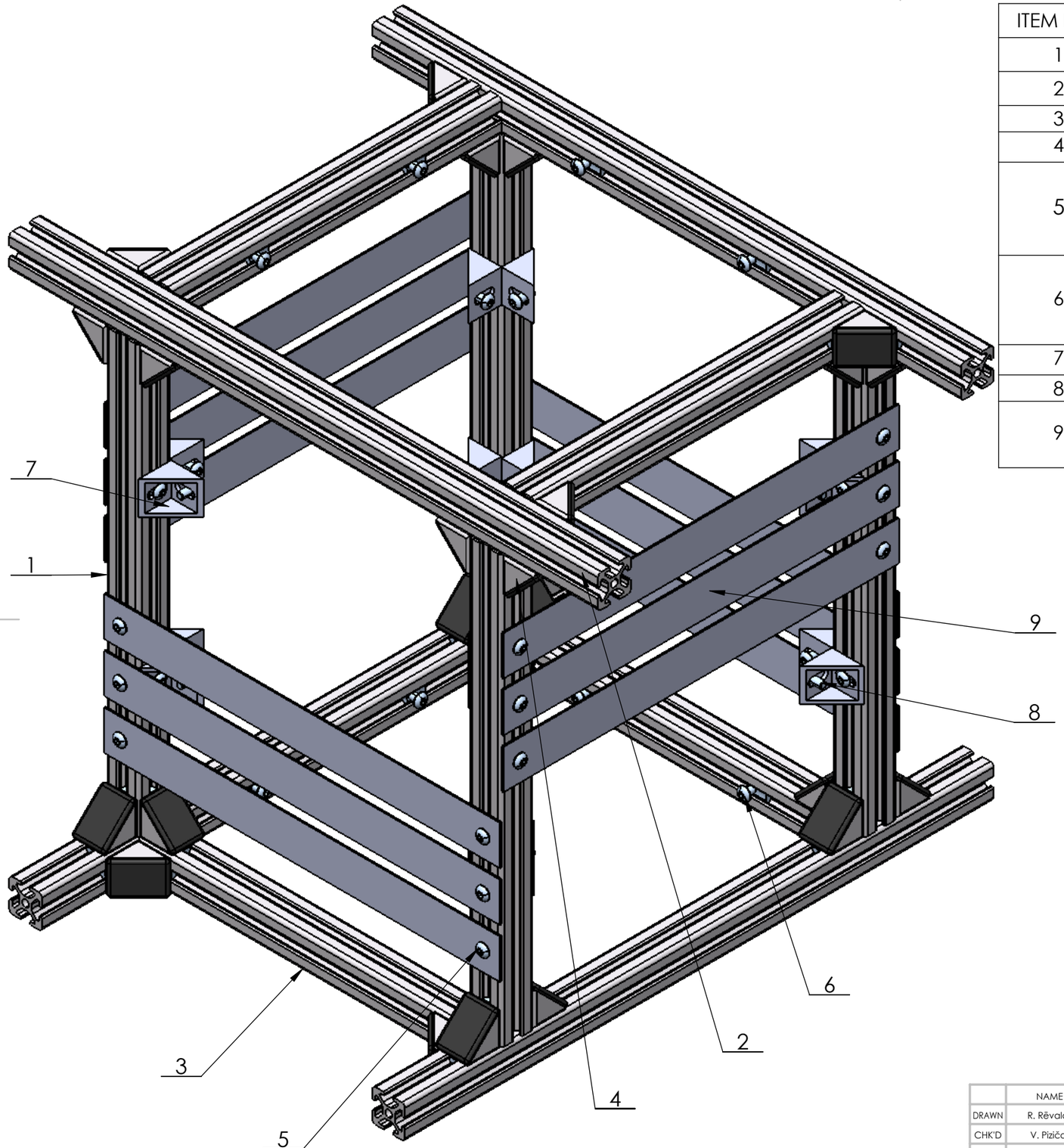
MATERIAL:

DWG NO. C-19 UV 00.000.000.002

A3

SCALE:1:5

SHEET 2 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Profile 20x20B slot 6	260 mm	4
2	Profile 20x20B slot 6	360 mm	4
3	Profile 20x20B slot 6	202 mm	4
4	Angle_Bracket Set	Angle Bracket Set 5 20x20	31
5	Bolt 10mm witk nut	Bolt M4 10mm with T nut B type Slot	48
6	Bolt 12mm with nut	Bolt M4 12mm with T nut B type Slot	16
7	Angle Bracket	Angle Bracket 20x20	16
8	Bolt 10mm	Bolt M4 10mm	16
9	Lamp wall		12

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE: Inner Frame

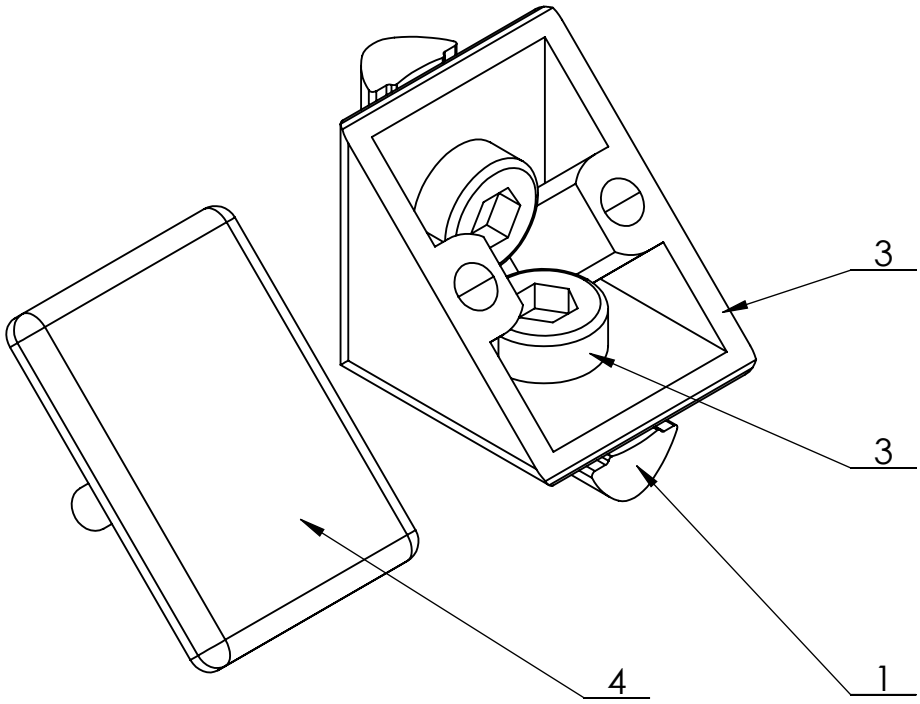
MATERIAL:

DWG NO. C-19 UV 00.000.000.003

A3

SCALE:1:2

SHEET 3 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	T-nut B-type slot 6 M4		2
2	Bolt M4 10mm		2
3	Angle Bracket		1
4	Angle Bracket Cap		1

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Angle Bracket Set

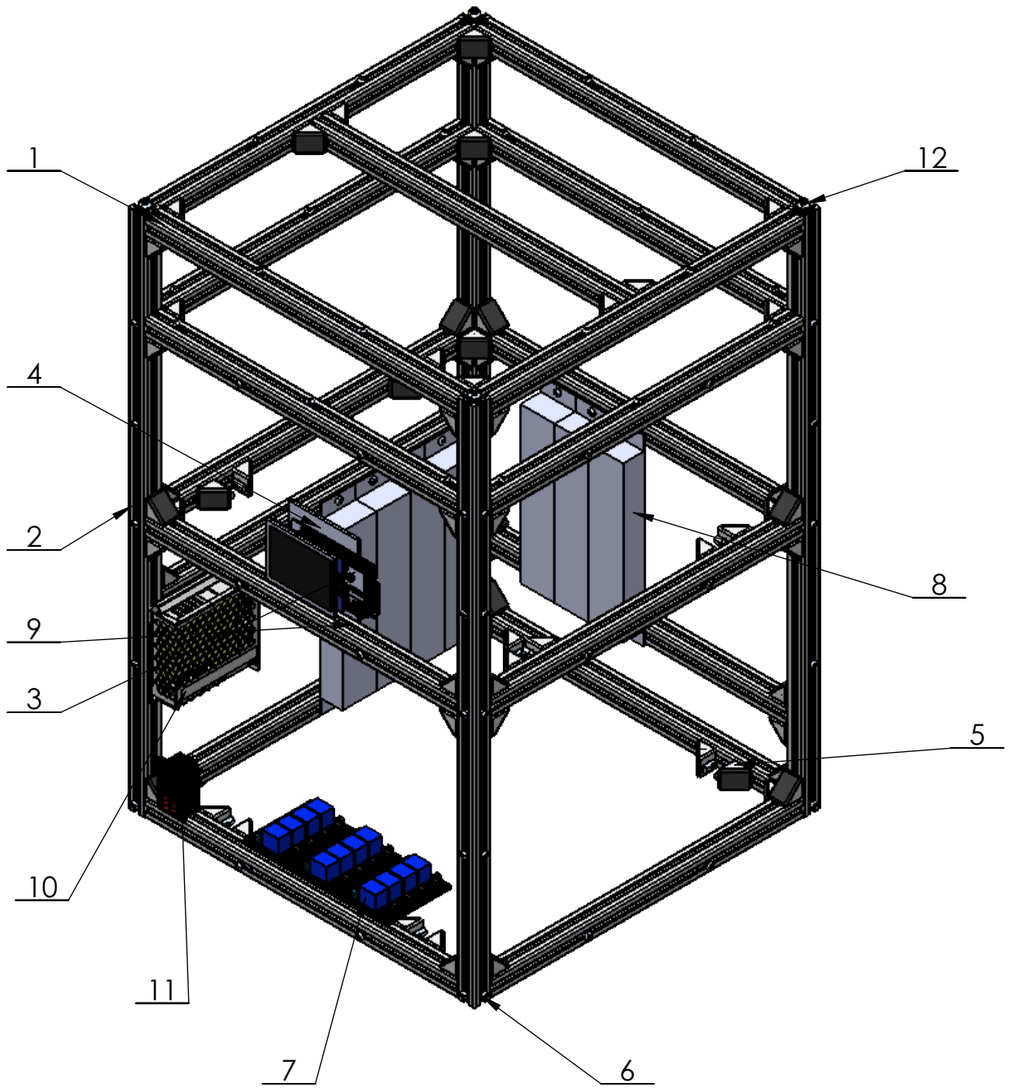
MATERIAL:

DWG NO. C-19 UV 00.000.000.004

A4

SCALE:2:1

SHEET 4 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Profile 20x20B slot 6	360 mm	19
2	Profile 20x20B slot 6	600 mm	4
3	Arduino_Touchscreen		1
4	Arduino Holder		1
5	Angle Bracket Set	Angle Bracket Set 5 20x20	115
6	Bolt 10mm witk nut	Bolt M4 10mm with T nut B type Slot	152
7	Relay Module	4 Chanel Relay Module	3
8	Ballast	Philips HF-E 3/414 TL5 II	7
9	Bolt 12mm	Bolt M4 10mm with T nut B type Slot	26
10	LRS-75 Mean Well		1
11	Power Socket	Power Socket with Fuse	1
12	Bolt 10mm	Bolt M6 10mm	8

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE: Outer frame

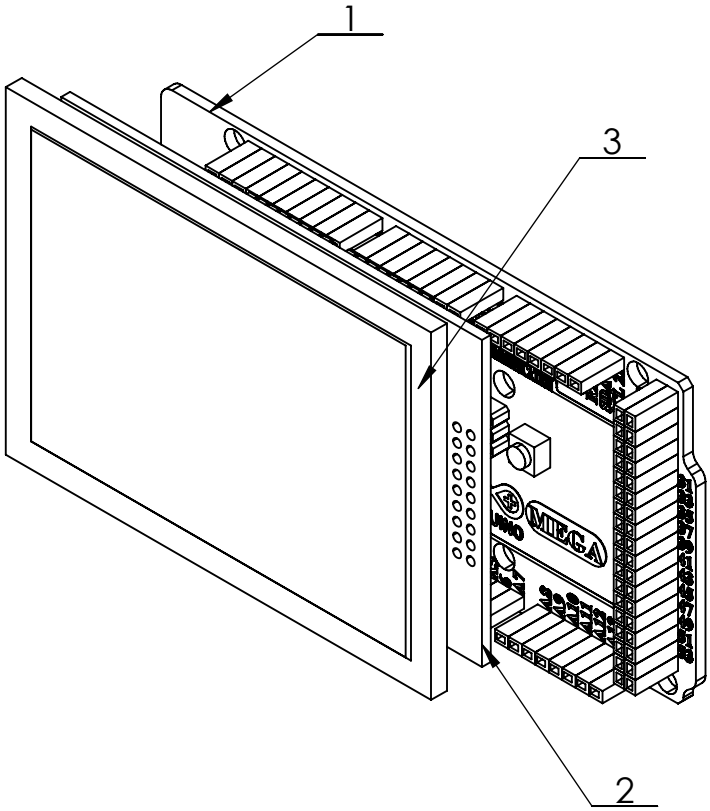
MATERIAL:

DWG NO. C-19 UV 00.000.000.005

A3

SCALE:1:5

SHEET 5 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	ARDUINO MEGA	Arduino Mega 2560	1
2	Touch Screen	Velleman VMA412 Touch Screen	1
3	Frame	Freme for Touch Screen	1

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

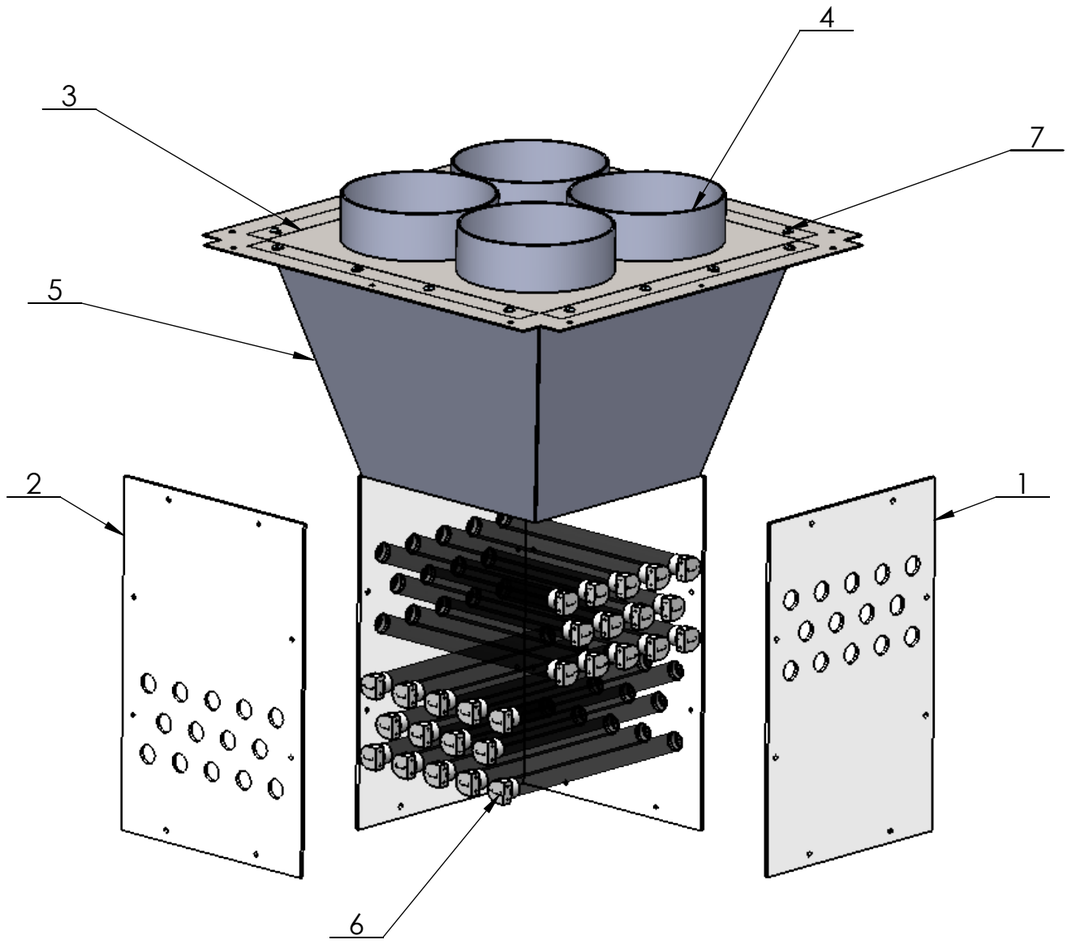
TITLE: Arduino_Touchscreen

MATERIAL:

DWG NO. CV-19 UV 00.000.000.006

A4

SCALE:1:1 SHEET 6 OF 20



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Inner wall_1	Inner wall	2
2	Inner wall_2		2
3	Inner top wall		1
4	Fan	WKA 125 TURBO	4
5	Truncated Square Pyramid		1
6	TUV LAMP		28
7	Bolt 10mm	Bolt M4 10mm	16

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

REZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE: Flow Chamber

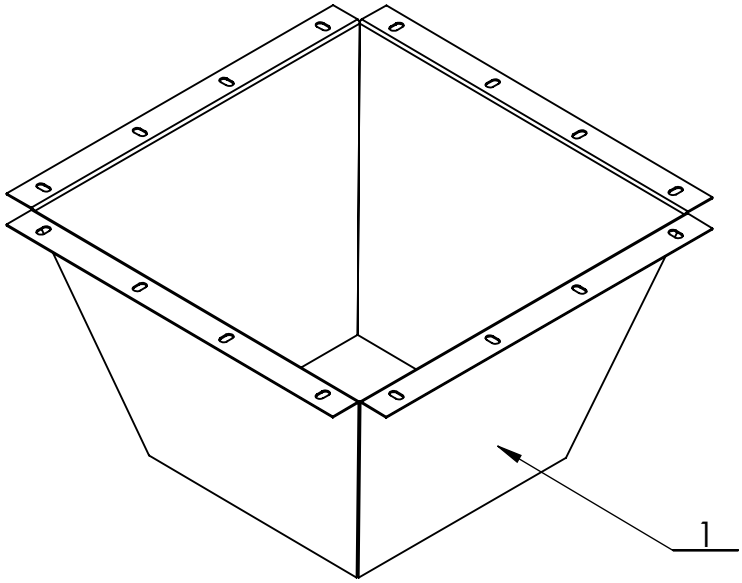
MATERIAL:

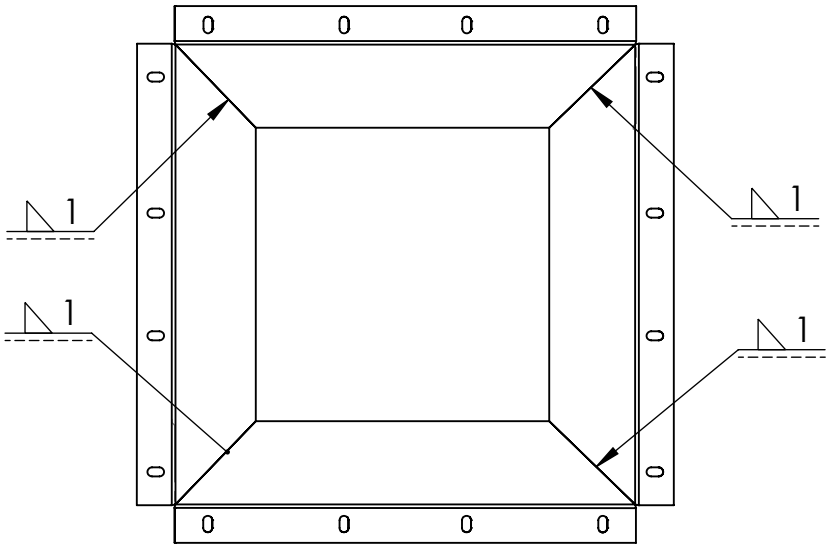
DWG NO. C-19 UV 00.000.000.007

A3

SCALE:1:5

SHEET 7 OF 20





ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Truncated Square Pyramid side		4

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Truncated Square Pyramid

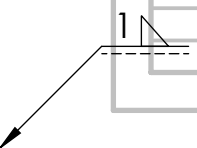
MATERIAL:

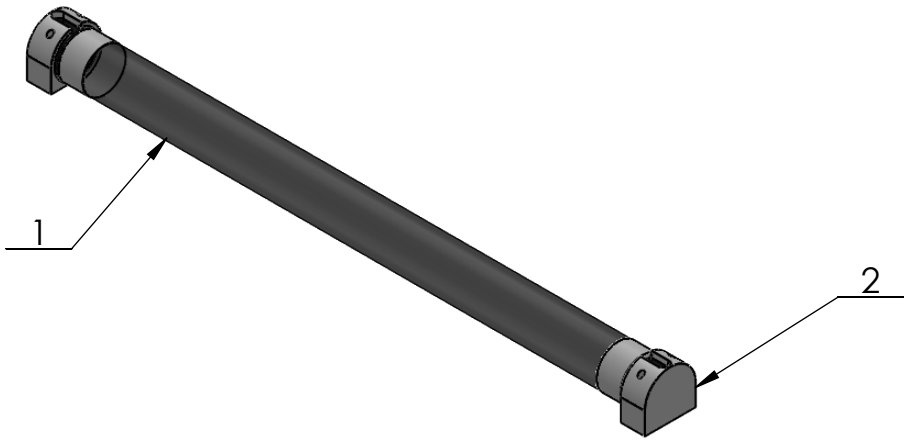
DWG NO. C-19 UV 00.000.000.008

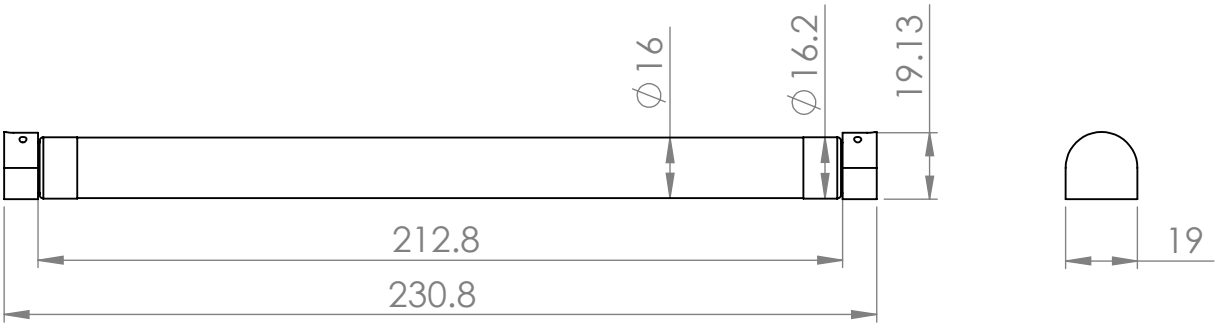
A4

SCALE:1:5

SHEET 8 OF 20







ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Lamp	TUV 11W FAM	1
2	T5 Socket	PUSH-FIT FOR T5 LAMPS	2

	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		24.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

TUV LAMP

MATERIAL:

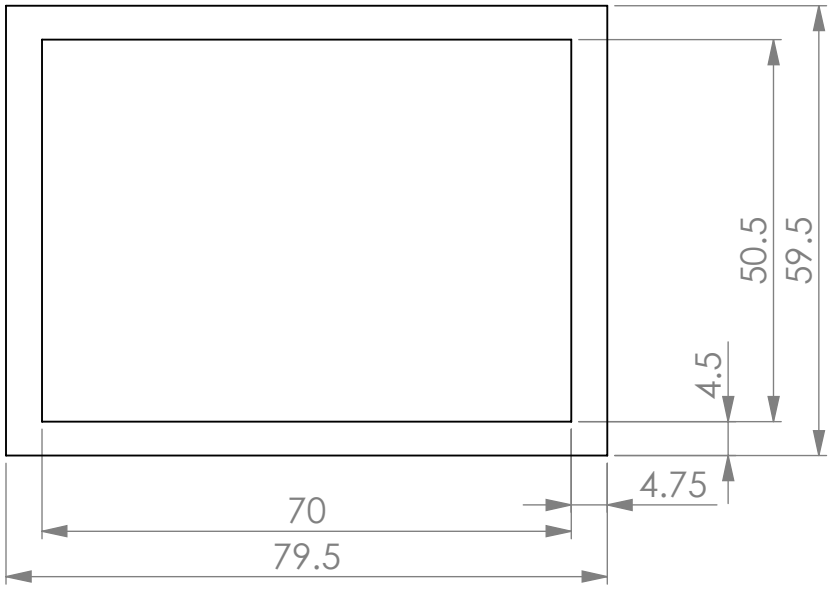
DWG NO. C-19 UV 00.000.000.009

A4

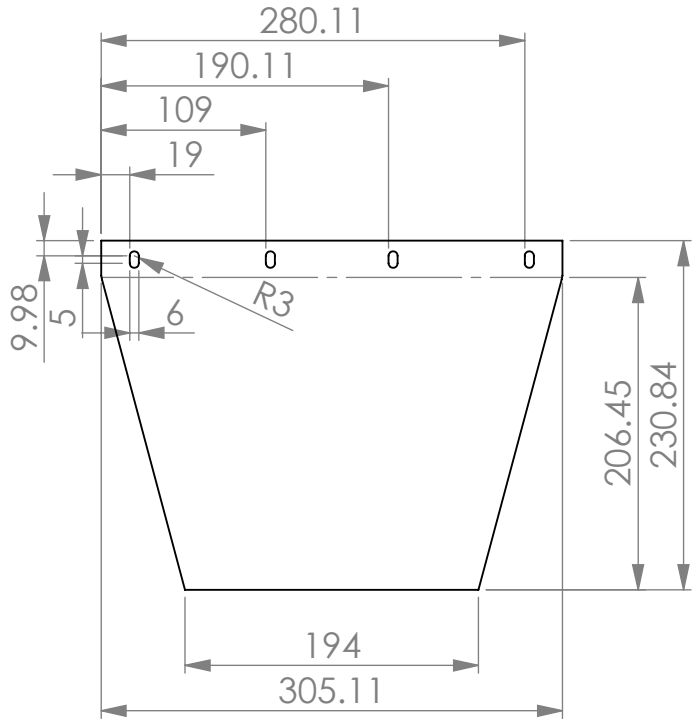
SCALE:1:2

SHEET 9 OF 20

	NAME	SIGNATURE	DATE			
DRAWN	R. Rēvalds		23.11.2020		RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA	TITLE: Lamp Wall
CHK'D	V. Piziča		24.11.2020			
					MATERIAL: AISI 304	DWG NO. C-19 UV 00.000.000.010 A4
						SCALE:1:2 SHEET 10 OF 20



	NAME	SIGNATURE	DATE			
DRAWN	R.Rēvalds		24.11.2020		RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA	TITLE: Frame
CHK'D	V. Piziča		24.11.2020			
					MATERIAL: Acrylic	DWG NO. C-19 UV 00.000.000.011 A4
						SCALE:1:1 SHEET 11 OF 20



	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		23.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

Truncated Square Pyramid Wall

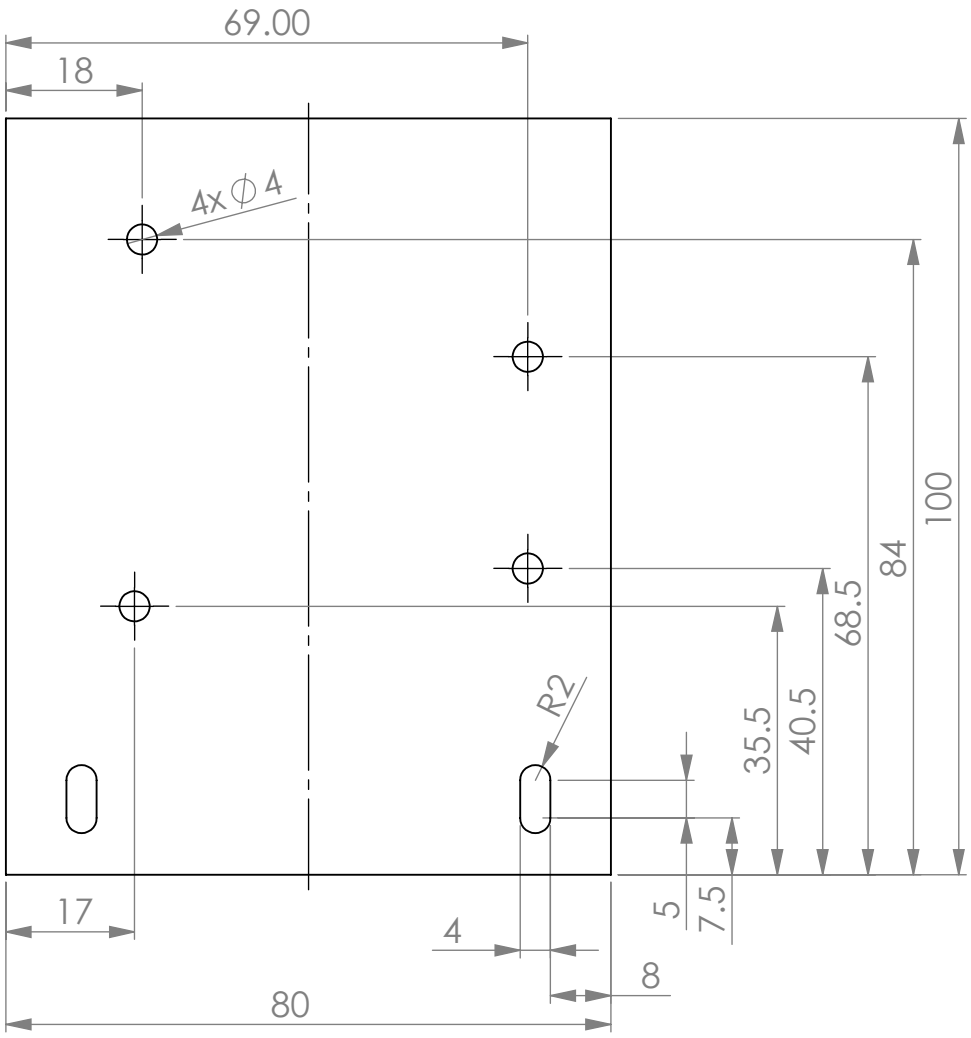
MATERIAL: AISI 304

DWG NO. C-19 UV 00.000.000.012

A4

SCALE:1:5

SHEET 12 OF 20



	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		23.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Arduino Holder

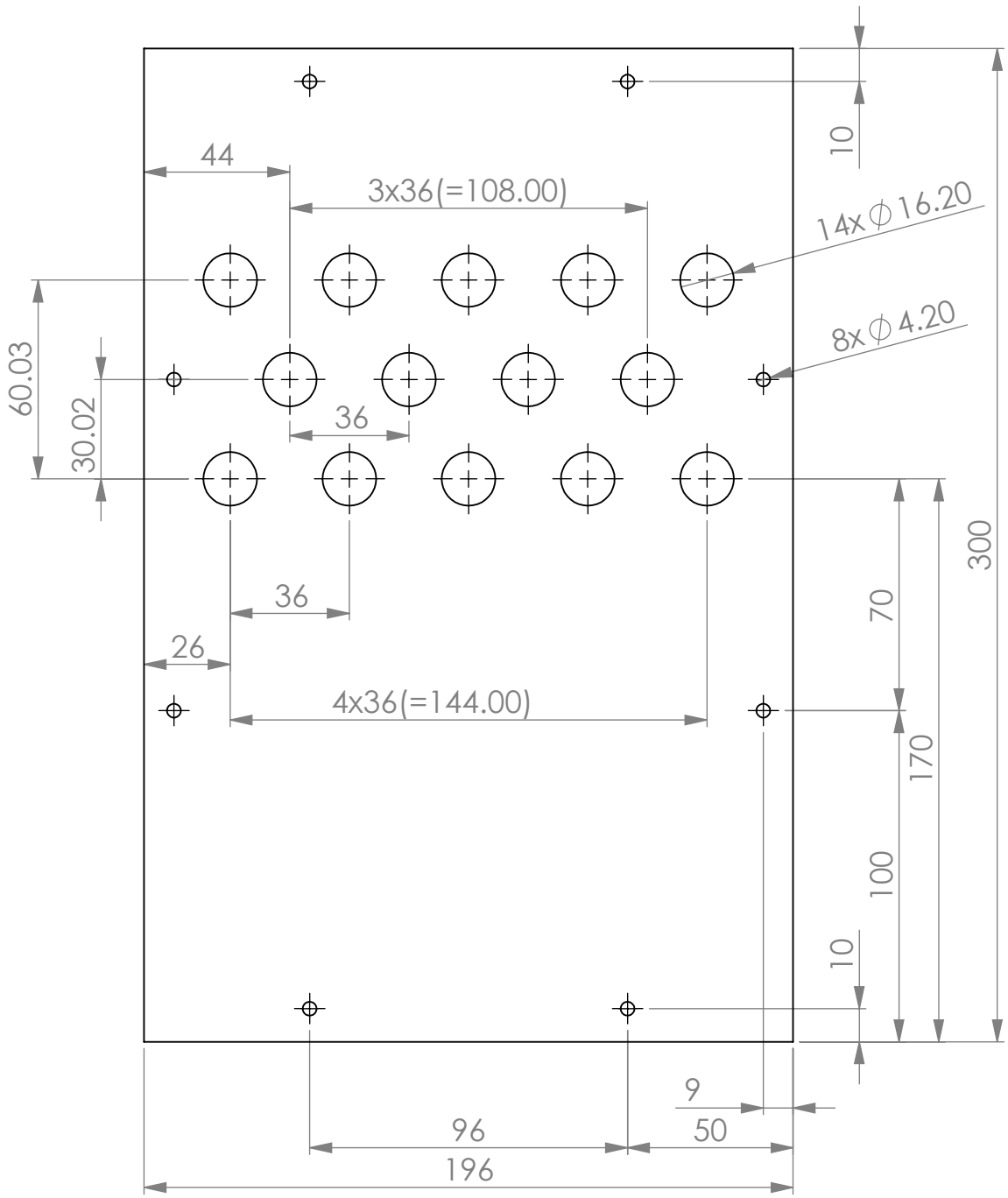
MATERIAL: Acrylic

DWG NO. C-19 UV 00.000.000.013

A4

SCALE:1:1

SHEET 13 OF 20



	NAME	SIGNATURE	DATE		
DRAWN	R. Rēvalds		23.11.2020		
CHK'D	V. Piziča		24.11.2020		

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Inner wall_1

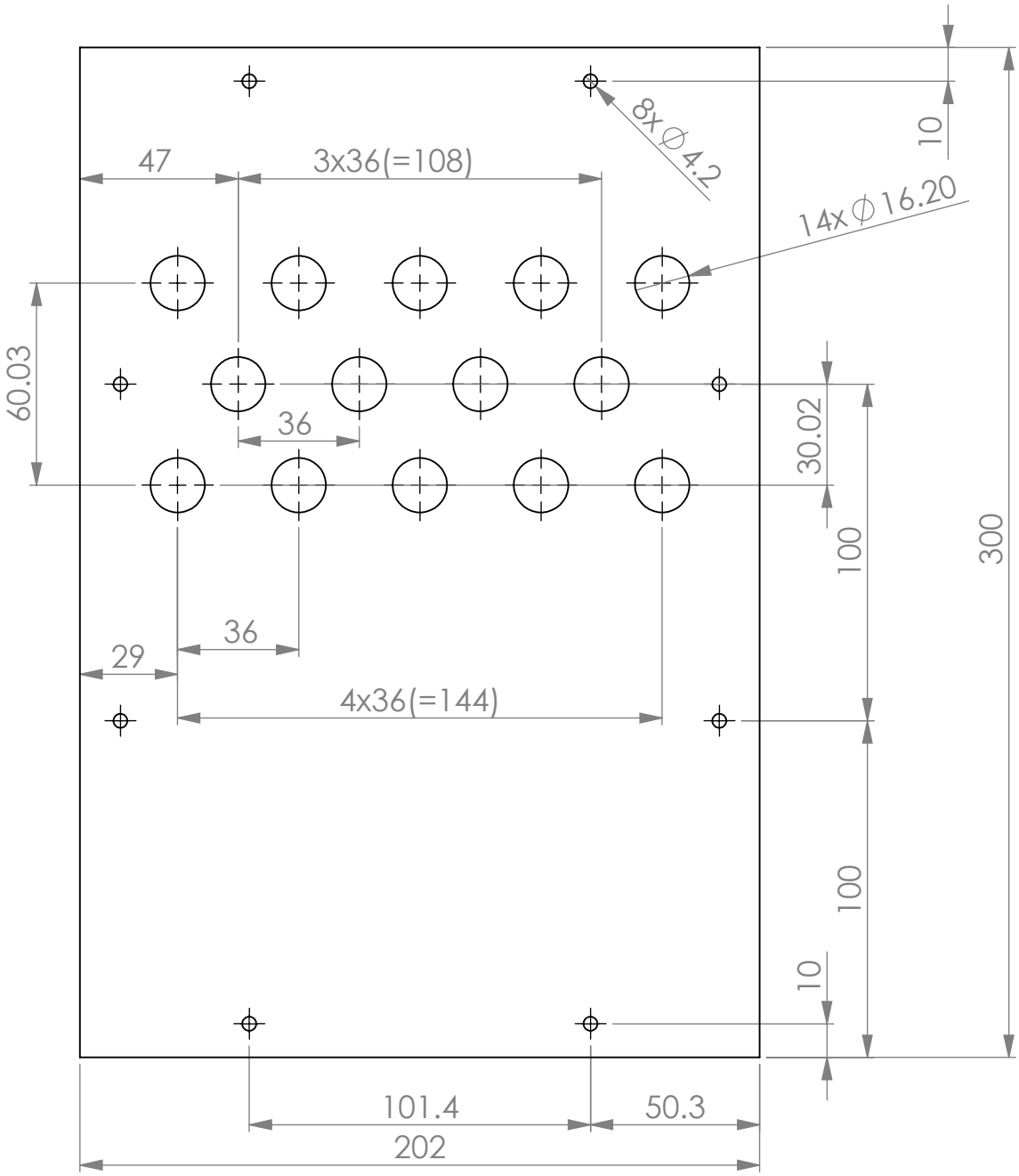
MATERIAL: Acrylic

DWG NO. C-19 UV 00.000.000.014

A4

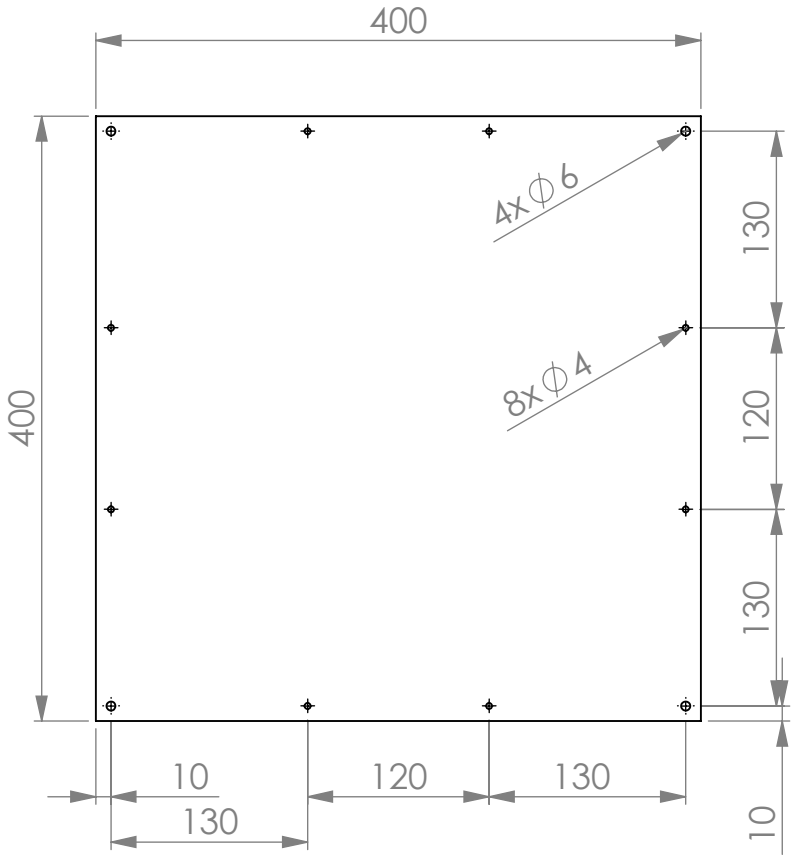
SCALE:1:2

SHEET 14 OF 20

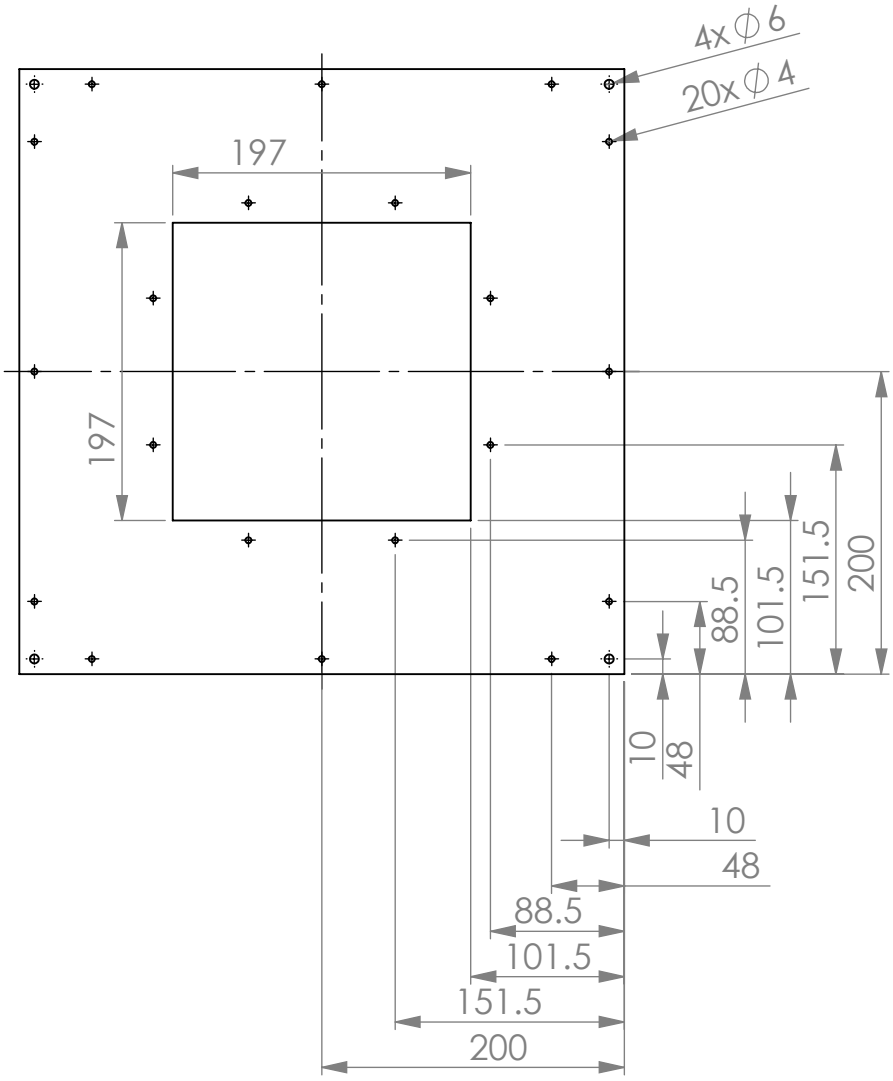


	NAME	SIGNATURE	DATE			
DRAWN	R. Rēvalds		23.11.2020		TITLE:	Inner wall_2
CHK'D	V. Piziča		24.11.2020			
					MATERIAL: Acrylic	DWG NO. C-19 UV 00.000.000.015
						A4
					SCALE:1:2	SHEET 15 OF 20

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA



	NAME	SIGNATURE	DATE			
DRAWN	R. Rēvalds		23.11.2020		RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA	TITLE: Top Wall
CHK'D	V. Piziča		24.11.2020			
					MATERIAL: AISI 304	DWG NO. C-19 UV 00.000.000.016 A4
						SCALE:1:5 SHEET 16 OF 20



	NAME	SIGNATURE	DATE		
DRAWN	R. Rēvalds		23.11.2020		
CHK'D	V. Piziča		24.11.2020		

RTA
RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Buttom Wall

MATERIAL:

AISI 304

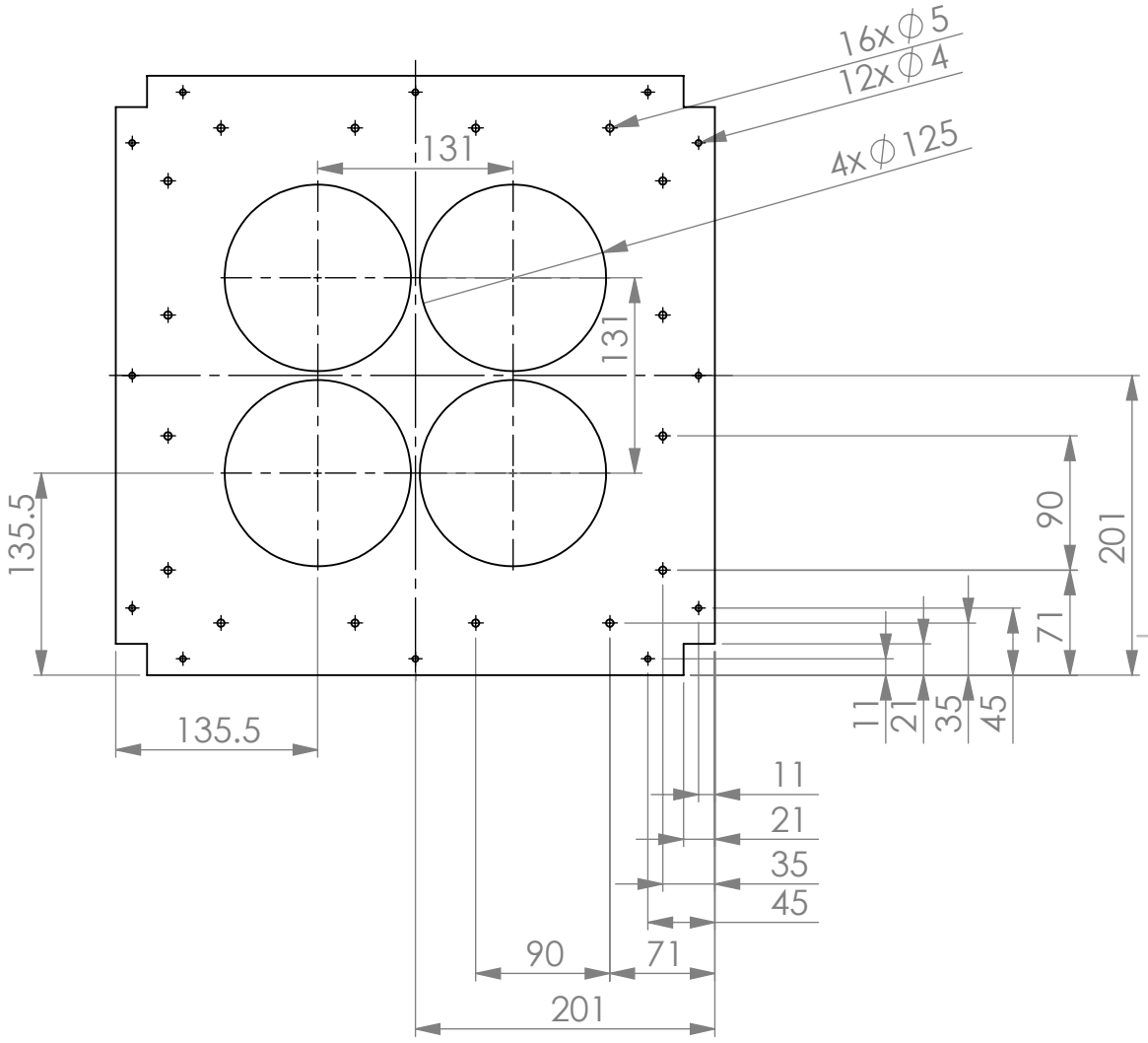
DWG NO.

C-19 UV 00.000.000.017

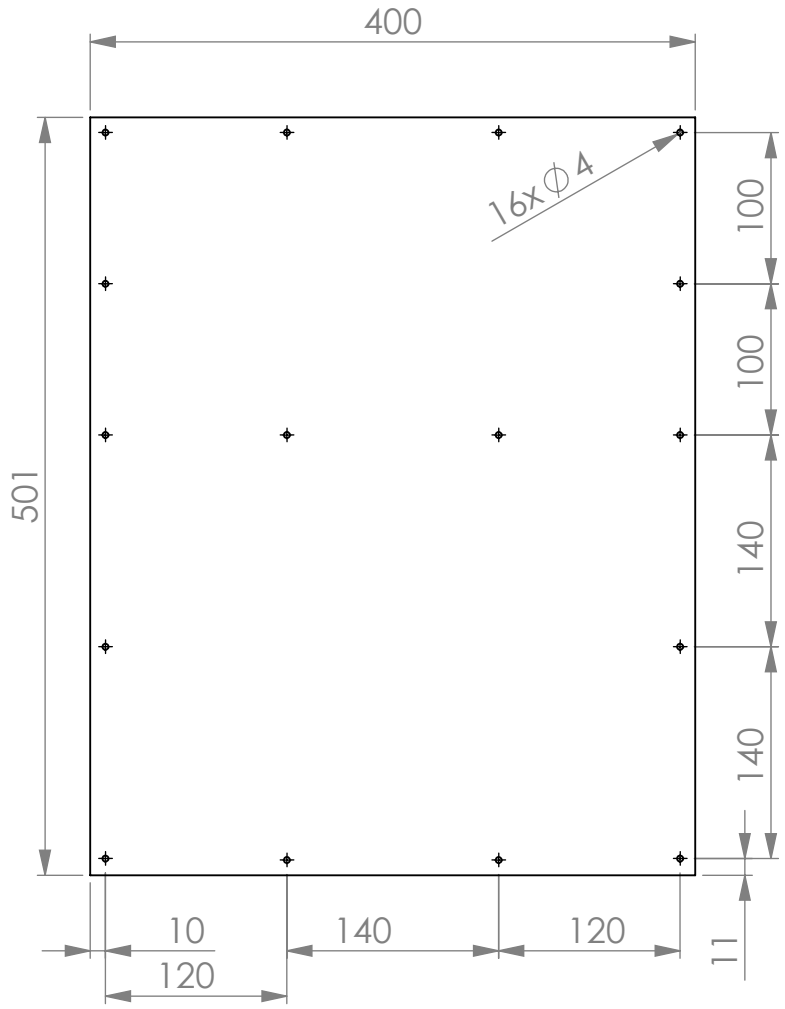
A4

SCALE:1:5

SHEET 17 OF 20



	NAME	SIGNATURE	DATE			
DRAWN	R. Rēvalds		23.11.2020	RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA	TITLE: Inner top wall	
CHK'D	V. Piziča		24.11.2020			
				MATERIAL: AISI 304	DWG NO. C-19 UV 00.000.000.018	A4
					SCALE:1:5	SHEET 18 OF 20



	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		23.11.2020
CHK'D	V. Piziča		24.11.2020

RTA
RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Side Wall

MATERIAL:

AISI 304

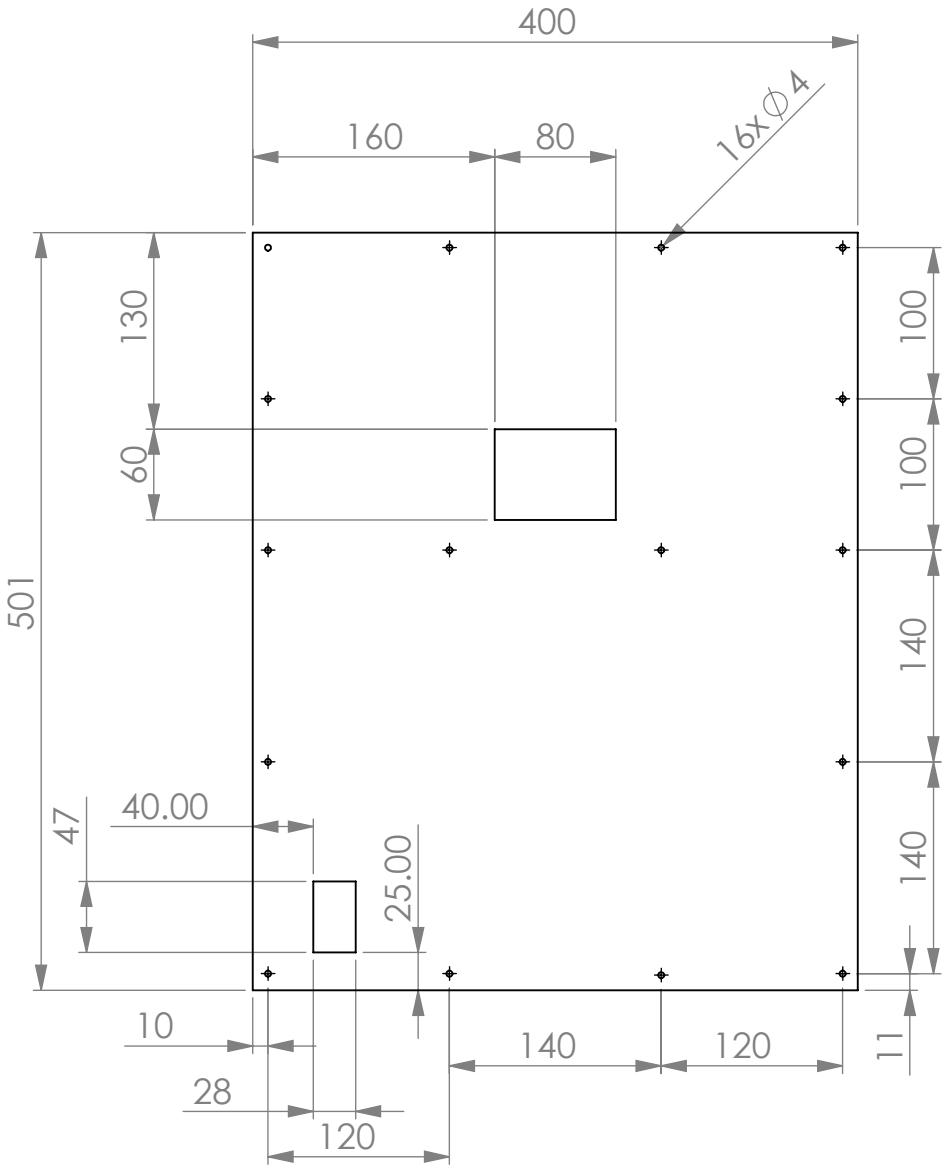
DWG NO.

C-19 UV 00.000.000.019

A4

SCALE:1:5

SHEET 19 OF 20



	NAME	SIGNATURE	DATE
DRAWN	R. Rēvalds		23.11.2020
CHK'D	V. Piziča		24.11.2020

RTA RĒZEKNES TEHNOLOĢIJU AKADĒMIJA

TITLE:

Front wall

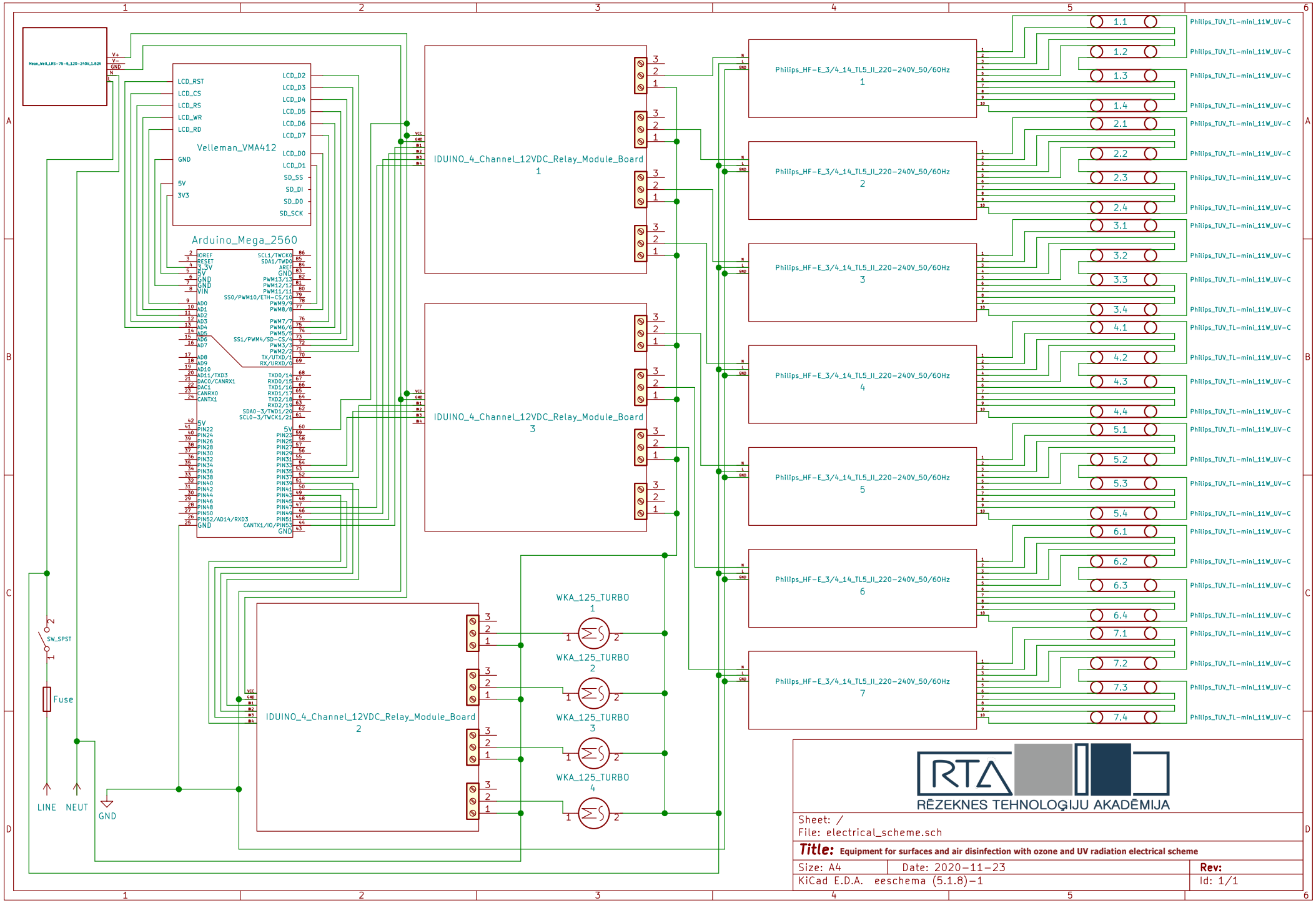
MATERIAL: AISI 304

DWG NO. C-19 UV 00.000.000.020

A4

SCALE:1:5

SHEET 20 OF 20


Mean_Well_LRS-75-5_120-240V_1.52A
V+
V-
GND
N
L
Velleman_VMA412
LCD_RST
LCD_CS
LCD_RS
LCD_WR
LCD_RD
GND
5V
3V3
LCD_D2
LCD_D3
LCD_D4
LCD_D5
LCD_D6
LCD_D7
LCD_D0
LCD_D1
SD_SS
SD_DI
SD_D0
SD_SCK
Arduino_Mega_2560
IDUINO_4_Channel_12VDC_Relay_Module_Board
1
IDUINO_4_Channel_12VDC_Relay_Module_Board
3
IDUINO_4_Channel_12VDC_Relay_Module_Board
2
WKA_125_TURBO
1
WKA_125_TURBO
2
WKA_125_TURBO
3
WKA_125_TURBO
4
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
1
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
2
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
3
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
4
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
5
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
6
Philips_HF-E_3/4_14_TL5_II_220-240V_50/60Hz
7
Philips_TUV_TL-mini_11W_UV-C
SW_SPST
Fuse
LINE
NEUT
GND
RĒZEKNES TEHNOLOĢIJU AKADĒMIJA
Sheet: /
File: electrical_scheme.sch
Title: Equipment for surfaces and air disinfection with ozone and UV radiation electrical scheme
Size: A4
Date: 2020-11-23
Rev:
KiCad E.D.A. eeschema (5.1.8)-1
Id: 1/1

```
/*
 Program for equipment for surfaces and air disinfection with ozone and UV radiation
 11.2020.
*/

#include <Adafruit_TFTLCD.h>
#include <Adafruit_GFX.h>
#include <TouchScreen.h>
#include <UTFTGLUE.h>
UTFTGLUE myGLCD(0x9341, A2, A1, A3, A4, A0);

//Screen pin
#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

#define YP A2
#define XM A3
#define YM 8
#define XP 9

#define TS_MINX 129
#define TS_MINY 179
#define TS_MAXX 907
#define TS_MAXY 849

//Colours
#define Black           0x0000      /*   0,   0,   0 */
#define Navy            0x000F      /*   0,   0, 128 */
#define DarkGreen       0x03E0      /*   0, 128,   0 */
#define DarkCyan        0x03EF      /*   0, 128, 128 */
#define Maroon          0x7800      /* 128,   0,   0 */
```

```
#define Purple          0x780F      /* 128,   0, 128 */
#define Olive           0x7BE0      /* 128, 128,   0 */
#define LightGrey       0xC618      /* 192, 192, 192 */
#define DarkGrey        0x7BEF      /* 128, 128, 128 */
#define Blue            0x001F      /*   0,   0, 255 */
#define Green           0x07E0      /*   0, 255,   0 */
#define Cyan            0x07FF      /*   0, 255, 255 */
#define Red             0xF800      /* 255,   0,   0 */
#define Magenta         0xF81F      /* 255,   0, 255 */
#define Yellow          0xFFE0      /* 255, 255,   0 */
#define White           0xFFFF      /* 255, 255, 255 */
#define Orange          0xFD20      /* 255, 165,   0 */
#define GreenYellow     0xAFE5      /* 173, 255,  47 */
#define Pink            0xF81F
#define Fons            0xAF0F

#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

boolean buttonEnabled = true;

//Fan pin
int Fan1Pin = 41;
int Fan2Pin = 39;
int Fan3Pin = 45;
int Fan4Pin = 43;

//Ballast pin
int Pal1Pin = 53;
int Pal2Pin = 51;
int Pal3Pin = 49;
int Pal4Pin = 47;
```

```
int Pal5Pin = 37;
int Pal6Pin = 35;
int Pal7Pin = 33;

char currentPage;
boolean Fan1 = 0;
boolean Fan1_state = 0;

boolean Fan2 = 0;
boolean Fan2_state = 0;

boolean Fan3 = 0;
boolean Fan3_state = 0;

boolean Fan4 = 0;
boolean Fan4_state = 0;

boolean FanV = 0;
boolean FanV_state = 0;

boolean Pal1 = 0;
boolean Pal1_state = 0;

boolean Pal2 = 0;
boolean Pal2_state = 0;

boolean Pal3 = 0;
boolean Pal3_state = 0;

boolean Pal4 = 0;
boolean Pal4_state = 0;

boolean Pal5 = 0;
boolean Pal5_state = 0;
```

```
boolean Pal6 = 0;
boolean Pal6_state = 0;

boolean Pal7 = 0;
boolean Pal7_state = 0;

boolean PalV = 0;
boolean PalV_state = 0;

//Touchscreen pressure points
#define MINPRESSURE 10
#define MAXPRESSURE 1000

void HomeScreen()
{
tft.fillScreen(DarkGrey);

//Draw white frame
tft.drawRect(0,0,319,240,White);

//Fan
tft.fillRect(60,50, 200, 50, Navy);
tft.drawRect(60,50, 200, 50,White);

tft.setCursor(125,65);
tft.setTextColor(LightGrey);
tft.setTextSize(3);
tft.print("Fan");

//Lamp
tft.fillRect(60,140, 200, 50, Navy);
tft.drawRect(60,140, 200, 50,White);
```

```
tft.setCursor(120,152);
tft.setTextColor(LightGrey);
tft.setTextSize(3);
tft.print("Lamp");
}
void Ventilatori()
{
tft.fillScreen(LightGrey);

tft.drawRect(0,0,319,240,White);

tft.setCursor(135,35);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("Fan");

tft.fillRect(10,10, 70, 50, DarkGrey);
tft.drawRect(10,10, 70, 50,White);

tft.setCursor(20,27);
tft.setTextColor(White);
tft.setTextSize(2);
tft.print("Back");
}
void Fan1_recG()
{
tft.fillRect(49,80, 108, 40, Green);
tft.drawRect(49,80, 108, 40,White);

tft.setCursor(95,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("1");
}
```

```
void Fan1_recR()
{
tft.fillRect(49,80, 108, 40, Red);
tft.drawRect(49,80, 108, 40,White);

tft.setCursor(95,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("1");
}

void Fan2_recG()
{
tft.fillRect(163,80, 108, 40, Green);
tft.drawRect(163,80, 108, 40,White);

tft.setCursor(207,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("2");
}

void Fan2_recR()
{
tft.fillRect(163,80, 108, 40, Red);
tft.drawRect(163,80, 108, 40,White);

tft.setCursor(207,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("2");
}
```

```
void Fan3_recG()
{
tft.fillRect(49,130, 108, 40,Green);
tft.drawRect(49,130, 108, 40,White);

tft.setCursor(95,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("3");
}

void Fan3_recR()
{
tft.fillRect(49,130, 108, 40,Red);
tft.drawRect(49,130, 108, 40,White);

tft.setCursor(95,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("3");
}

void Fan4_recG()
{
tft.fillRect(163,130, 108, 40, Green);
tft.drawRect(163,130, 108, 40,White);

tft.setCursor(207,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("4");
}
```

```
void Fan4_recR()
{
tft.fillRect(163,130, 108, 40, Red);
tft.drawRect(163,130, 108, 40,White);

tft.setCursor(207,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("4");
}

void FanV_recG()
{
tft.fillRect(84,180, 152, 40, Green);
tft.drawRect(84,180, 152, 40,White);

tft.setCursor(135,190);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("All");
}

void FanV_recR()
{
tft.fillRect(84,180, 152, 40, Red);
tft.drawRect(84,180, 152, 40,White);

tft.setCursor(135,190);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("All");
}
```

```
void Lampas()
{
tft.fillScreen(LightGrey);

tft.setCursor(135,25);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("Lamp");

tft.drawRect(0,0,319,240,White);

tft.fillRect(10,10, 70, 50, DarkGrey);
tft.drawRect(10,10, 70, 50,White);

tft.setCursor(20,27);
tft.setTextColor(White);
tft.setTextSize(2);
tft.print("Back");

}

void Pal1_recG()
{
tft.fillRect(10,80, 70, 40, Green);
tft.drawRect(10,80, 70, 40,White);

tft.setCursor(20,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print(" 1");
}
```

```
void Pal1_recR()
{
tft.fillRect(10,80, 70, 40, Red);
tft.drawRect(10,80, 70, 40,White);

tft.setCursor(20,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print(" 1");
}

void Pal2_recG()
{
tft.fillRect(87,80, 70, 40, Green);
tft.drawRect(87,80, 70, 40,White);

tft.setCursor(115,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("2");
}

void Pal2_recR()
{
tft.fillRect(87,80, 70, 40, Red);
tft.drawRect(87,80, 70, 40,White);

tft.setCursor(115,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("2");
}
```

```
void Pal3_recG()
{
tft.fillRect(163,80, 70, 40, Green);
tft.drawRect(163,80, 70, 40,White);

tft.setCursor(188,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("3");
}

void Pal3_recR()
{
tft.fillRect(163,80, 70, 40, Red);
tft.drawRect(163,80, 70, 40,White);

tft.setCursor(188,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("3");
}

void Pal4_recG()
{
tft.fillRect(240,80, 70, 40, Green);
tft.drawRect(240,80, 70, 40,White);

tft.setCursor(247,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print(" 4");
}
```

```
void Pal4_recR()
{
tft.fillRect(240,80, 70, 40, Red);
tft.drawRect(240,80, 70, 40,White);

tft.setCursor(247,90);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print(" 4");
}

void Pal5_recG()
{
tft.fillRect(49,130, 70, 40, Green);
tft.drawRect(49,130, 70, 40,White);

tft.setCursor(75,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("5");
}

void Pal5_recR()
{
tft.fillRect(49,130, 70, 40, Red);
tft.drawRect(49,130, 70, 40,White);

tft.setCursor(75,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("5");
}
```

```
void Pal6_recG()
{
tft.fillRect(125,130, 70, 40, Green);
tft.drawRect(125,130, 70, 40,White);

tft.setCursor(153,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("6");
}

void Pal6_recR()
{
tft.fillRect(125,130, 70, 40, Red);
tft.drawRect(125,130, 70, 40,White);

tft.setCursor(153,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("6");
}

void Pal7_recG()
{
tft.fillRect(201,130, 70, 40, Green);
tft.drawRect(201,130, 70, 40,White);

tft.setCursor(220,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("7");
}
```

```
void Pal7_recR()
{
tft.fillRect(201,130, 70, 40, Red);
tft.drawRect(201,130, 70, 40,White);

tft.setCursor(230,140);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("7");
}

void PalV_recG()
{
tft.fillRect(84,180, 152, 40, Green);
tft.drawRect(84,180, 152, 40,White);

tft.setCursor(136,190);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("All");
}

void PalV_recR()
{
tft.fillRect(84,180, 152, 40, Red);
tft.drawRect(84,180, 152, 40,White);

tft.setCursor(136,190);
tft.setTextColor(Black);
tft.setTextSize(3);
tft.print("All");
}
```

```
void setup()
{
  Serial.begin(9600);
  Serial.print("Starting...");

  tft.reset();
  tft.begin(0x9341);
  tft.setRotation(1);

  HomeScreen();
  currentPage = '0';

  pinMode(Fan1Pin, OUTPUT);
  pinMode(Fan2Pin, OUTPUT);
  pinMode(Fan3Pin, OUTPUT);
  pinMode(Fan4Pin, OUTPUT);
  pinMode(Pal1Pin, OUTPUT);
  pinMode(Pal2Pin, OUTPUT);
  pinMode(Pal3Pin, OUTPUT);
  pinMode(Pal4Pin, OUTPUT);
  pinMode(Pal5Pin, OUTPUT);
  pinMode(Pal6Pin, OUTPUT);
  pinMode(Pal7Pin, OUTPUT);

}
```

```
void loop()
{
  TSPoint p = ts.getPoint();  //Get touch point

  if (p.z > ts.pressureThreshhold)
  {

   Serial.print("X = "); Serial.print(p.x);
   Serial.print("\tY = "); Serial.print(p.y);
   Serial.print("\n");

   p.x = map(p.x, TS_MAXX, TS_MINX, 0, 320);
   p.y = map(p.y, TS_MAXY, TS_MINY, 0, 240);

  }
   boolean buttonEnabled = true;

   if (currentPage == '0')
   {

    if(p.x>60 && p.x<260 && p.y>50 && p.y<100 && buttonEnabled)
    {
    buttonEnabled = false;



    pinMode(XM, OUTPUT);
    pinMode(YP, OUTPUT);

    currentPage = '1';
    Ventilatori();
    }
```

```
if(p.x>60 && p.x<260 && p.y>140 && p.y<190 && buttonEnabled)
      {
      buttonEnabled = false;

      pinMode(XM, OUTPUT);
      pinMode(YP, OUTPUT);

      currentPage = '2';
      Lampas();
      }
 }

 //Fan
 if(currentPage == '1')
 {
  //Fan 1
  if(p.x>49 && p.x<157 && p.y>80 && p.y<120 && buttonEnabled)
  {
  buttonEnabled = false;

  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  Fan1 = 1;
  }
  else
  {
   Fan1 = 0;
  }
      Serial.println(Fan1);
      Serial.println("dg" );
      Serial.println(Fan1_state);
```

```
//Fan 2
if(p.x>163 && p.x<271 && p.y>80 && p.y<120 && buttonEnabled)
{
buttonEnabled = false;

pinMode(XM, OUTPUT);
pinMode(YP, OUTPUT);

Fan2 = 1;
}
else
{
 Fan2 = 0;
}

    //Fan3
    if(p.x>49 && p.x<157 && p.y>130 && p.y<170 && buttonEnabled)
    {
    buttonEnabled = false;

    pinMode(XM, OUTPUT);
    pinMode(YP, OUTPUT);

    Fan3 = 1;
    }
    else
    {
     Fan3 = 0;
    }
```

```
                //Fan 4
                if(p.x>163 && p.x<271 && p.y>130 && p.y<170 && buttonEnabled)
                {
                buttonEnabled = false;

                pinMode(XM, OUTPUT);
                pinMode(YP, OUTPUT);

                Fan4 = 1;
                }
                else
                {
                 Fan4 = 0;
                }
if(Fan1 == 1)
{
 Fan1_state =! Fan1_state;
 delay(500);
}
  if(Fan2 ==1)
  {
   Fan2_state =! Fan2_state;
   delay(500);
  }
    if(Fan3 ==1)
    {
     Fan3_state =! Fan3_state;
     delay(500);
    }
      if(Fan4 ==1)
      {
       Fan4_state =! Fan4_state;
       delay(500);
      }
```

```
if(Fan1_state == 1)
{
 digitalWrite(Fan1Pin, HIGH);
 Fan1_recG();
}
else
{
 digitalWrite(Fan1Pin, LOW);
 Fan1_recR();
}
  if(Fan2_state == 1)
  {
   digitalWrite(Fan2Pin, HIGH);
   Fan2_recG();
  }
  else
  {
   digitalWrite(Fan2Pin, LOW);
   Fan2_recR();
  }
    if(Fan3_state == 1)
    {
     digitalWrite(Fan3Pin, HIGH);
     Fan3_recG();
    }
    else
    {
    digitalWrite(Fan3Pin, LOW);
    Fan3_recR();
    }
      if(Fan4_state == 1)
      {
       digitalWrite(Fan4Pin, HIGH);
       Fan4_recG(); }
```

```
      else
      {
       digitalWrite(Fan4Pin, LOW);
       Fan4_recR();
      }

  //Back
  if(p.x>10 && p.x<80 && p.y>10 && p.y<60 && buttonEnabled)
  {
  buttonEnabled = false;

  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  currentPage = '0';
  HomeScreen();
  }
}
//Lamp
if(currentPage == '2')
{
    //Ballast 1
    if(p.x>10 && p.x<80 && p.y>80 && p.y<120 && buttonEnabled)
    {
    buttonEnabled = false;

    pinMode(XM, OUTPUT);
    pinMode(YP, OUTPUT);
    Pal1 = 1;
    }
    else
    {
     Pal1 = 0;
    }
```

```
//Ballast 2
if(p.x>87 && p.x<157 && p.y>80 && p.y<120 && buttonEnabled)
{
buttonEnabled = false;

pinMode(XM, OUTPUT);
pinMode(YP, OUTPUT);
Pal2 = 1;
}
else
{
 Pal2 = 0;
}
   //Ballast 3
   if(p.x>163 && p.x<233 && p.y>80 && p.y<120 && buttonEnabled)
   {
   buttonEnabled = false;

   pinMode(XM, OUTPUT);
   pinMode(YP, OUTPUT);

   Pal3 = 1;
   }
   else
   {
    Pal3 = 0;
   }
      //Ballast 4
      if(p.x>240 && p.x<310 && p.y>80 && p.y<120 && buttonEnabled)
      {
      buttonEnabled = false;
      pinMode(XM, OUTPUT);
      pinMode(YP, OUTPUT);
      Pal4 = 1; }
```

```
else
{
 Pal4 = 0;
}
   //Ballast 5
   if(p.x>49 && p.x<119 && p.y>130 && p.y<170 && buttonEnabled)
   {
   buttonEnabled = false;

   pinMode(XM, OUTPUT);
   pinMode(YP, OUTPUT);

   Pal5 = 1;
   }
   else
   {
    Pal5 = 0;
   }
      //Ballast 6
      if(p.x>125 && p.x<195 && p.y>130 && p.y<170 && buttonEnabled)
      {
      buttonEnabled = false;

      pinMode(XM, OUTPUT);
      pinMode(YP, OUTPUT);

      Pal6 = 1;
      }
      else
      {
       Pal6 = 0;
      }
```

```
                    //Ballast 7
                    if(p.x>201 && p.x<271 && p.y>130 && p.y<170 && buttonEnabled)
                    {
                    buttonEnabled = false;

                    pinMode(XM, OUTPUT);
                    pinMode(YP, OUTPUT);

                    Pal7 = 1;
                    }
                    else
                    {
                     Pal7 = 0;
                    }
if(Pal1 == 1)
{
 Pal1_state =! Pal1_state;
 delay(500);
}
   if(Pal2 == 1)
   {
    Pal2_state =! Pal2_state;
    delay(500);
   }
      if(Pal3 == 1)
      {
       Pal3_state =! Pal3_state;
       delay(500);
      }
         if(Pal4 == 1)
         {
          Pal4_state =! Pal4_state;
          delay(500);
         }
```

```
            if(Pal5 == 1)
            {
             Pal5_state =! Pal5_state;
             delay(500);
            }
               if(Pal6 == 1)
               {
                Pal6_state =! Pal6_state;
                delay(500);
               }
                  if(Pal7 == 1)
                  {
                   Pal7_state =! Pal7_state;
                   delay(500);
                  }
if(Pal1_state == 1)
{
 digitalWrite(Pal1Pin, HIGH);
 Pal1_recG();
}
else
{
 digitalWrite(Pal1Pin, LOW);
 Pal1_recR();
}
   if(Pal2_state == 1)
   {
    digitalWrite(Pal2Pin, HIGH);
    Pal2_recG();
   }
   else
   {
    digitalWrite(Pal2Pin, LOW);
    Pal2_recR(); }
```

```
 if(Pal3_state == 1)
{
 digitalWrite(Pal3Pin, HIGH);
 Pal3_recG();
}
else
{
 digitalWrite(Pal3Pin, LOW);
 Pal3_recR();
}
    if(Pal4_state == 1)
    {
     digitalWrite(Pal4Pin, HIGH);
    Pal4_recG();
    }
    else
    {
     digitalWrite(Pal4Pin, LOW);
     Pal4_recR();
    }
        if(Pal5_state == 1)
        {
         digitalWrite(Pal5Pin, HIGH);
         Pal5_recG();
        }
        else
        {
         digitalWrite(Pal5Pin, LOW);
         Pal5_recR();
        }
            if(Pal6_state == 1)
            {
             digitalWrite(Pal6Pin, HIGH);
             Pal6_recG(); }
```

```
                    else
                   {
                    digitalWrite(Pal6Pin, LOW);
                    Pal6_recR();
                   }
                      if(Pal7_state == 1)
                      {
                       digitalWrite(Pal7Pin, HIGH);
                       Pal7_recG();
                      }
                      else
                      {
                       digitalWrite(Pal7Pin, LOW);
                       Pal7_recR();
                      }
        //Back
        if(p.x>10 && p.x<80 && p.y>10 && p.y<60 && buttonEnabled)
        {
        buttonEnabled = false;



        pinMode(XM, OUTPUT);
        pinMode(YP, OUTPUT);

        currentPage = '0';
        HomeScreen();
        }
   }
   delay(10);
}
```