
8. Mellékletek

1. sz. melléklet: *A motorok sebességének beállításáért felelős Arduino programkód:*

```
#include <AccelStepper.h>
#include <math.h>

AccelStepper stepper1(1, 2, 5); // (Type:driver, STEP, DIR)
AccelStepper stepper2(1, 3, 6);

String msg;
String x;
String y;
int LeftMotor;
int RightMotor;
double LeftMotor_good;
double RightMotor_good;

void setup() {
  Serial.begin(115200);
  Serial.setTimeout(10);

  //Maximum sebesség megadása
  stepper1.setMaxSpeed(100);
  stepper2.setMaxSpeed(100);
}

void loop() {

  szamolas();
  stepper1.setSpeed(LeftMotor_good);
  stepper2.setSpeed(RightMotor_good);

  stepper1.runSpeed();
  stepper2.runSpeed();
}
```

```
Serial.print(LeftMotor_good);  
Serial.print(" ");  
Serial.println(RightMotor_good);  
  
}  
  
void szamolas(){  
  
    if (Serial.available() > 0) {  
        msg = Serial.readStringUntil('\n');  
  
        if (msg.length() == 9 && msg.charAt(0) == '-') {  
  
            x = msg.substring(0, 5);  
  
            y = msg.substring(5, 9);  
  
        }  
  
        else if(msg.length() == 9) {  
  
            x = msg.substring(0, 4);  
  
            y = msg.substring(4, 9);  
  
        }  
  
        else if (msg.length() == 10){  
  
            x = msg.substring(0, 5);  
  
            y = msg.substring(5, 10);  
        }  
    }  
}
```

```
else {  
  
x = msg.substring(0, 4);  
  
y = msg.substring(4, 8);  
  
}  
LeftMotor = x.toInt();  
  
RightMotor = y.toInt();  
  
LeftMotor_good = (LeftMotor/100);  
RightMotor_good = (RightMotor/100);  
}  
}
```

2. sz. melléklet: Az ultrahangos szenzorok jelküldéséért felelős Arduino programkód:

```
#define trigPin1 3  
#define echoPin1 2  
#define trigPin2 4  
#define echoPin2 5  
#define trigPin3 7  
#define echoPin3 8  
  
long duration, distance,  
RightSensor, BackSensor, FrontSensor, LeftSensor;  
  
void setup()  
{  
Serial.begin (115200);  
pinMode(trigPin1, OUTPUT);
```

```
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);
}

void loop () {
  SonarSensor(trigPin1, echoPin1);
  RightSensor = distance;
  SonarSensor(trigPin2, echoPin2);
  LeftSensor = distance;
  SonarSensor(trigPin3, echoPin3);
  FrontSensor = distance;

  Serial.print(LeftSensor);
  Serial.print("-");
  Serial.print(FrontSensor);
  Serial.print("-");
  Serial.println(RightSensor);
}

void SonarSensor(int trigPin,int echoPin)
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1; //hangsebesség
}
```

3. sz. melléklet: A motorok sebességének kiszámolásáért felelős Python programkód:

```
#!/usr/bin/python3

import time
import math
import serial_send_arduino
import serial

#1 méteren 4.823-szor fordul körbe a kerék
#1 körbefordulás 200 lépés
#964.6 lépésem van 1 méteren, ezzel kell szorozni majd a vr és
vl változókat

def MotorInputs(vel, rot, direction, TankMode):

    vr = float
    vl = float
    vrinmm = float
    vlinmm = float
    MotorDistance = 0.31085
    Radius = 0.033
    RightMotor_raw = float
    LeftMotor_raw = float
    RightMotor_final = float
    LeftMotor_final = float
    N_of_steps = 964.6

    vr = (2*int(vel)+int(rot)*float(MotorDistance))/(2*float(Radius))
    vl = (2*int(vel)-int(rot)*float(MotorDistance))/(2*float(Radius))
    vrinmm = (float(vr)/1000)
```

```
vlinmm = (float(vl)/1000)

LeftMotor_raw = (float(vlinmm)*float(N_of_steps))
RightMotor_raw = (float(vrinmm)*float(N_of_steps))

LeftMotor_final = round(LeftMotor_raw , 2)
RightMotor_final = round(RightMotor_raw, 2)

LeftMotor_final_to_arduino = (float(LeftMotor_final)*100)
RightMotor_final_to_arduino = (float(RightMotor_final)*100)

LeftMotor_in_int = int(LeftMotor_final_to_arduino)
RightMotor_in_int = int(RightMotor_final_to_arduino)

print("Left side motor will move with %s steps/second" %
LeftMotor_final)
print("Right side motor will move with %s steps/second" %
RightMotor_final)

LeftMotorInString = str(LeftMotor_in_int)
RightMotorInString = str(RightMotor_in_int)

MotorValuesInString_together = (LeftMotorInString +
RightMotorInString)

if TankMode == True:
    if direction == 'Left':
        MotorValuesInString_together = ('-
'+LeftMotorInString+RightMotorInString)
    if direction == 'Right':
        MotorValuesInString_together = (LeftMotorInString+'-
'+RightMotorInString)
    print(MotorValuesInString_together)

serial_send_arduino.write_read(MotorValuesInString_together)
```

4. sz. melléklet: A robotporszívó orientációjának megfelelő beállításáért felelős Python programkód:

```
#!/usr/bin/python3

import serial
import time
import motor_speed_calculator

arduino_ultrasonic = serial.Serial('/dev/ttyACM1', 115200,
timeout=1)
arduino_ultrasonic.reset_input_buffer()
def callback():
    arduino_ultrasonic
    arduino_ultrasonic.reset_input_buffer()

def wall_hit_Left():
    motor_speed_calculator.MotorInputs(2,0,'Left', True)
    callback()

def wall_hit_Right():
    motor_speed_calculator.MotorInputs(2,0,'Right', True)
    callback()

def clear_way():
    motor_speed_calculator.MotorInputs(2,0, '', False)
    callback()

while True:
    if arduino_ultrasonic.in_waiting:
        data = []
        while len(data) != 3:
            packet = arduino_ultrasonic.readline()
```

```
print(packet.decode('utf-8'))
data = packet.decode('utf-8').split('-')

try:
    data[2] = data[2].split('\r')[0]
except:
    pass
print(data)
LeftSonic = int(data[0])
FrontSonic = int(data[1])
RightSonic = int(data[2])

if FrontSonic < 20 or RightSonic < 25:
    print('Akadály a jobb oldalon')
    wall_hit_Left()
elif FrontSonic < 20 or LeftSonic < 25:
    print('Akadály a bal oldalon')
    wall_hit_Right()
elif FrontSonic > 20 or RightSonic > 25 or LeftSonic > 25:
    print('Tiszta a terep')
    clear_way()
```

5. sz. melléklet: A Python programkód amely azért felel, hogy az Arduinoval való kommunikáció létrejöjjön:

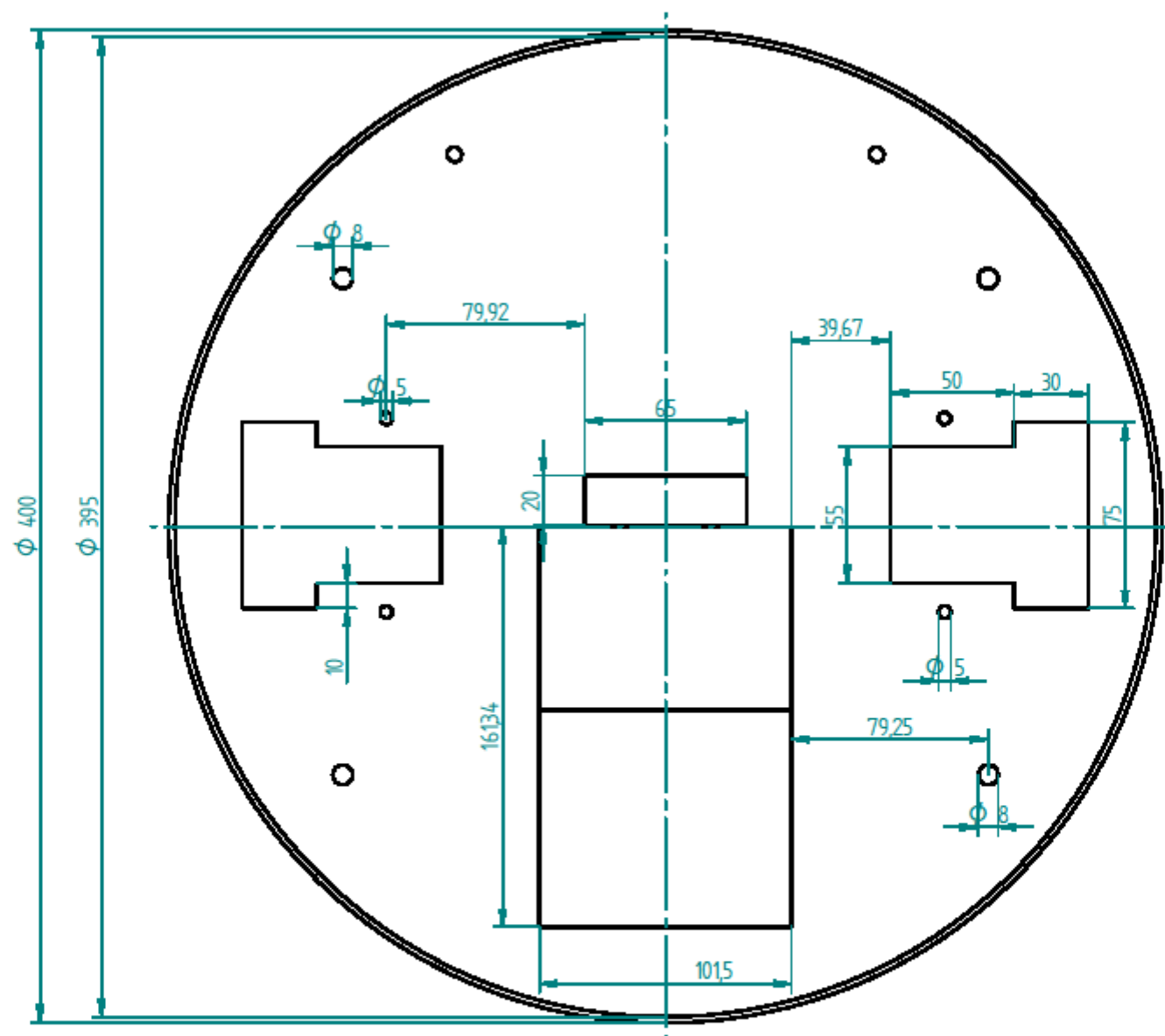
```
#!/usr/bin/python3

import serial
import time

arduino = serial.Serial('/dev/ttyACM0', 115200, timeout=1)
arduino.reset_input_buffer()

def write_read(x):
    arduino.write(bytes(x, 'utf-8'))
    time.sleep(1)
    data = arduino.readline().decode('utf-8').rstrip()
    return data
```

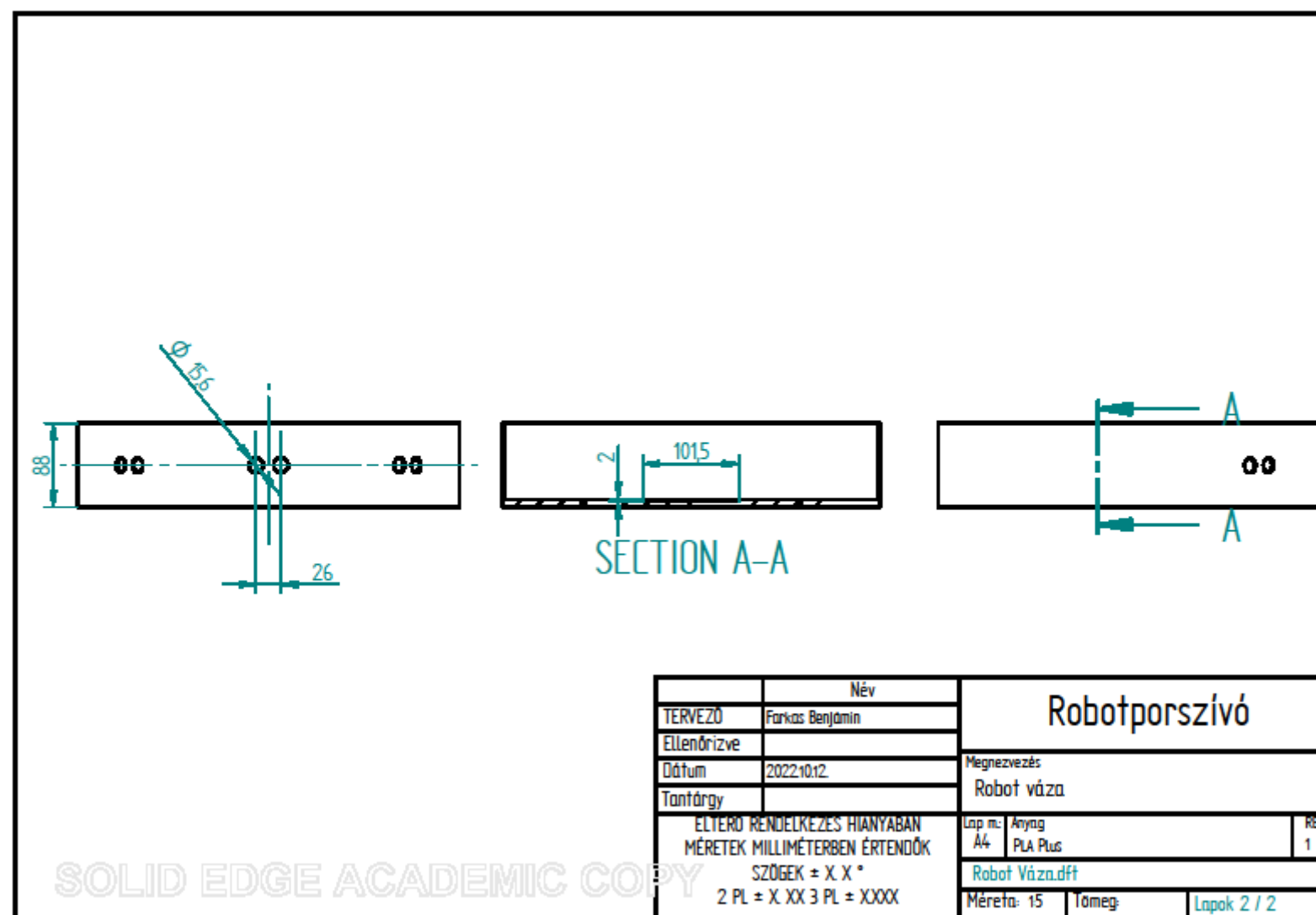
6. sz. melléklet: A robot vázának műszaki rajza felülnézetből:



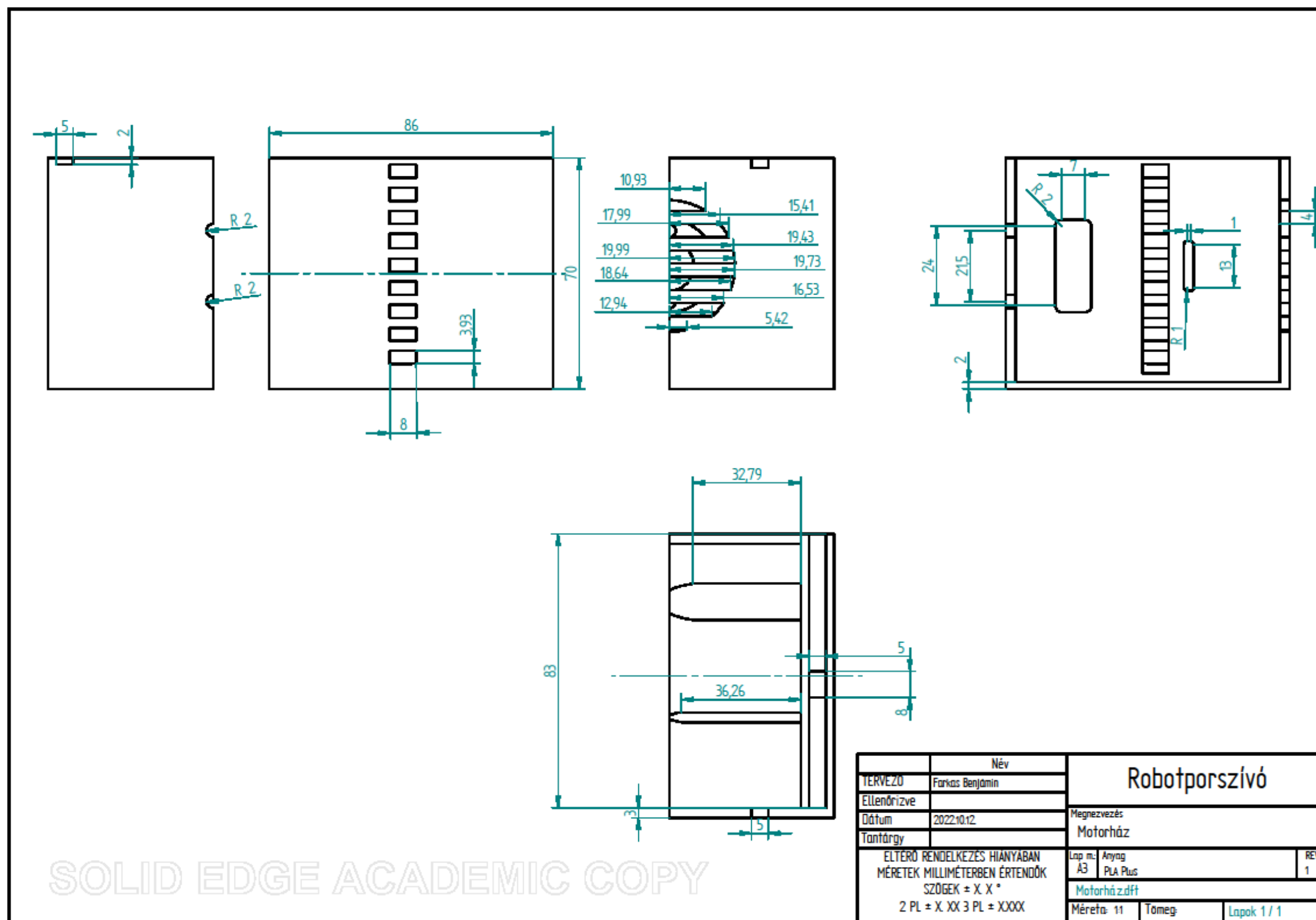
SOLID EDGE ACADEMIC COPY

Név		Robotporszívó	
TERVEZŐ	Farkas Benjámín		
Ellenőrizve		Megnevezés	
Dátum	2022.10.12.	Robot váza	
Tantárgy		Lap m.: Anyag	
ELTÉRŐ RENDELKEZÉS HIÁNYÁBAN MÉRETEK MILLIMÉTERBEN ÉRTENDŐK SZÖGEK ± X ° 2 PL ± X XX 3 PL ± XXXX		A3	PLA Plus
		Robot Váza.dft	
		Méret: 12	Tömeg: Lapok 1 / 2
		REV 1	

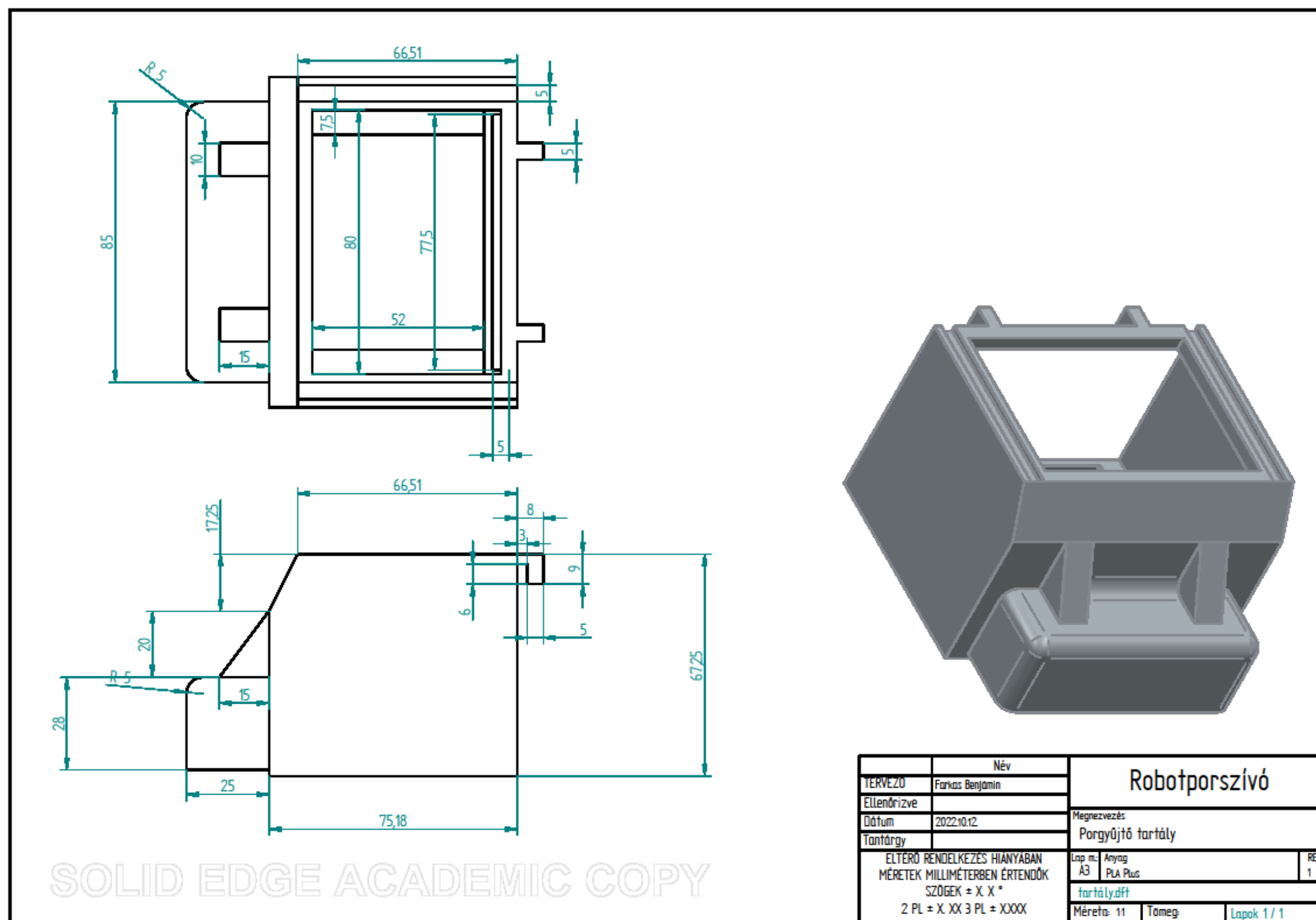
7. sz. melléklet: A robot vázának műszaki rajza előlnézetben:



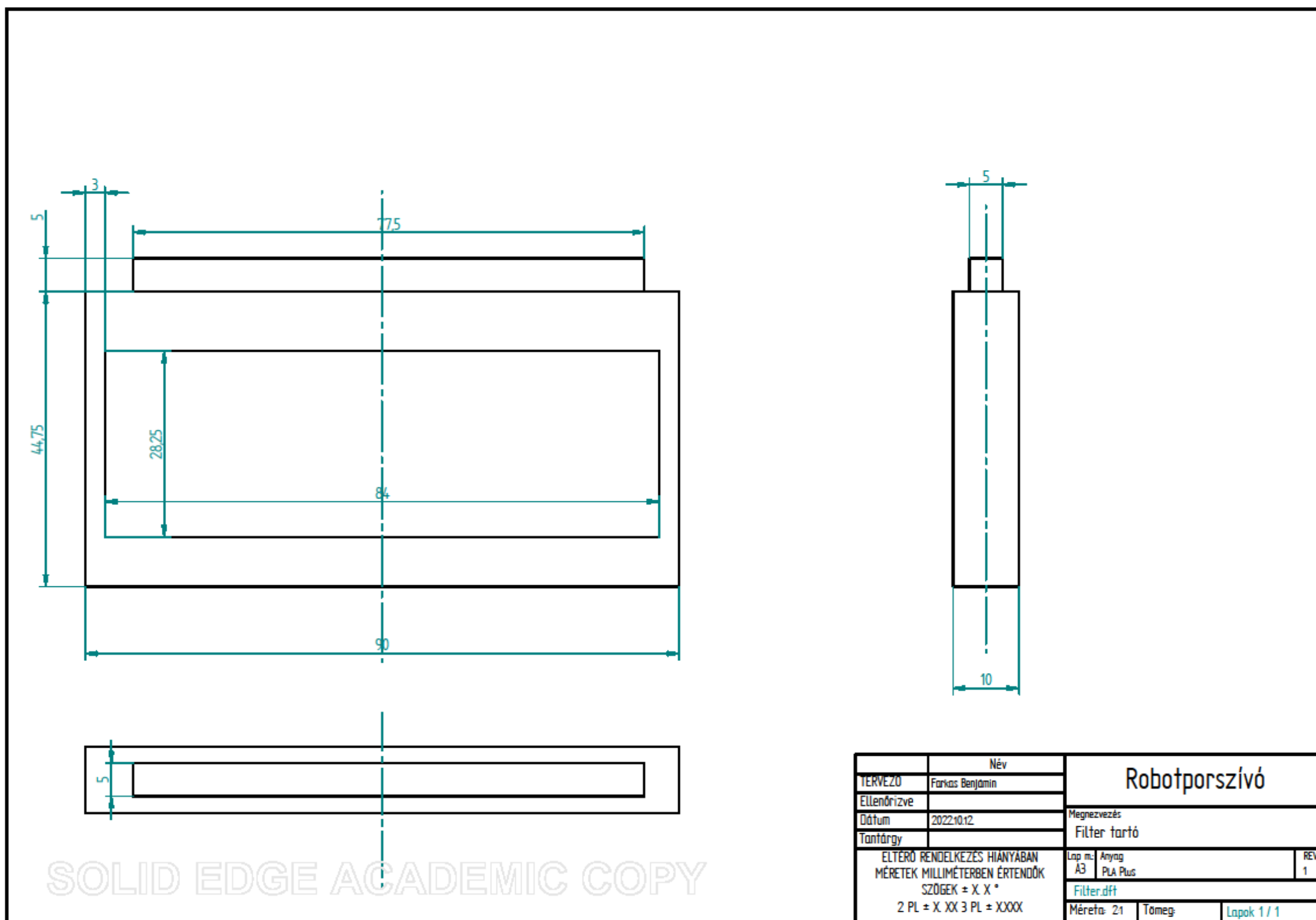
8. sz. melléklet: A szívmotor házának műszaki rajza:



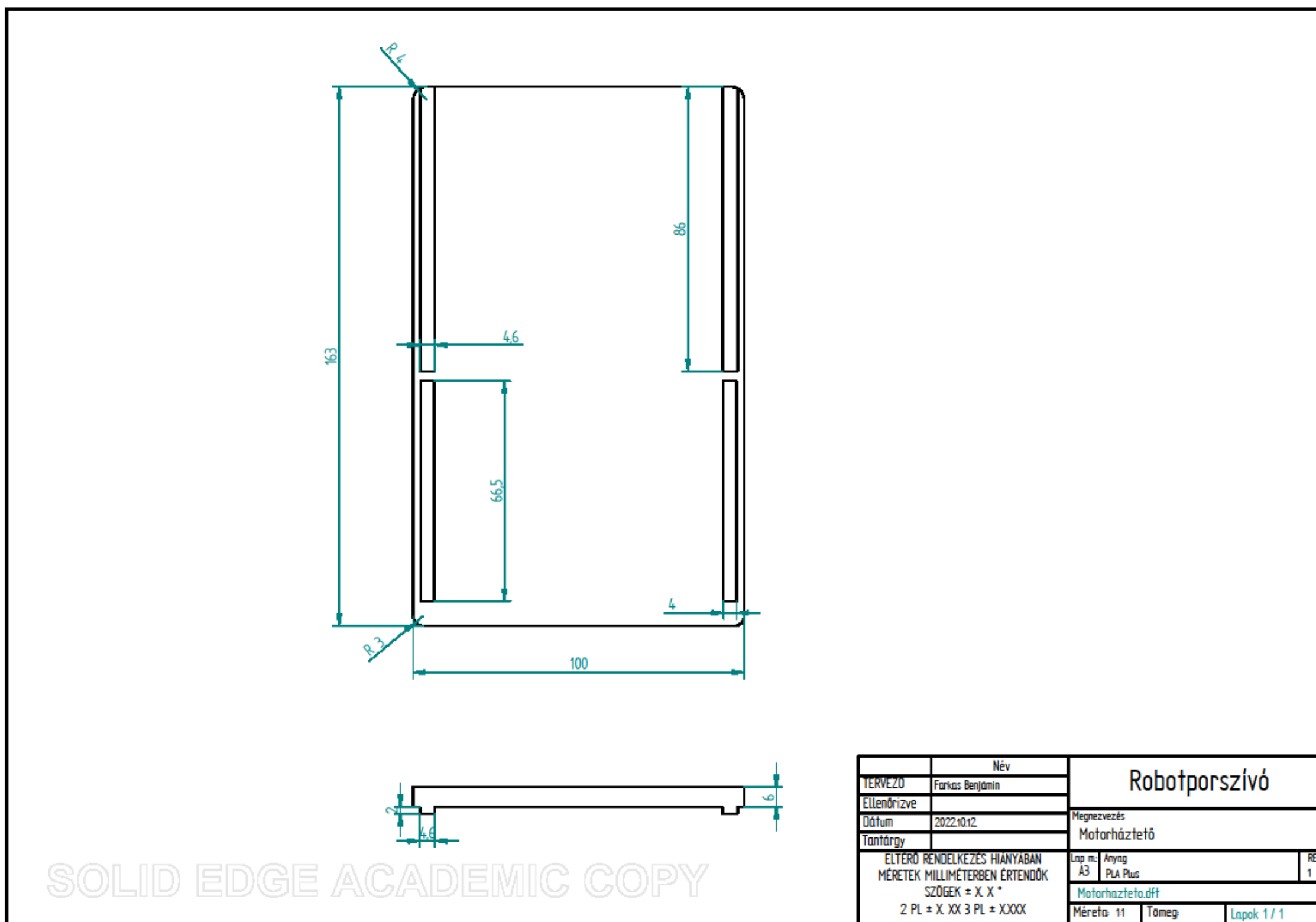
9. sz. melléklet: A porgyűjtő tartály műszaki rajza:



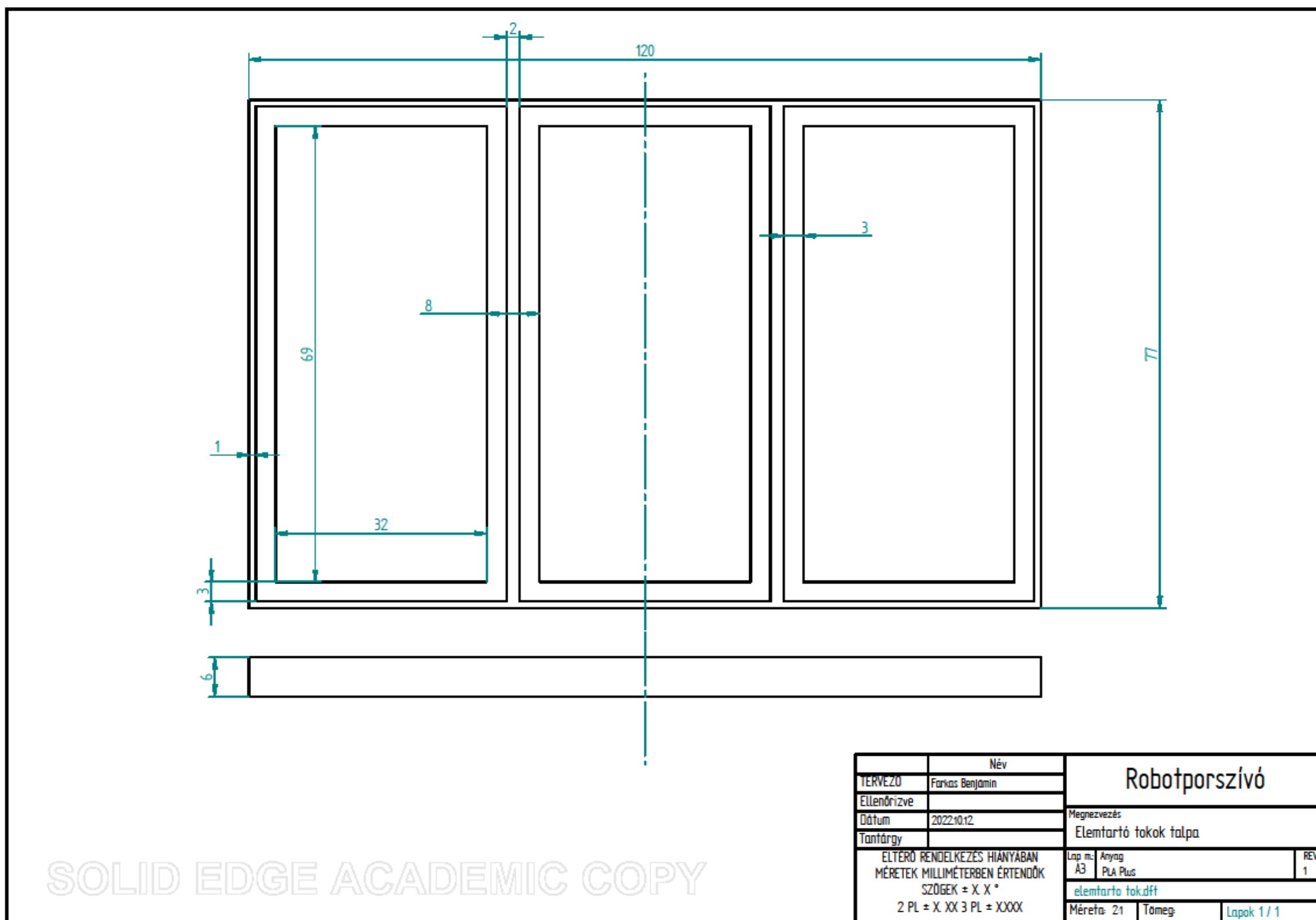
10. sz. melléklet: A porgyűjtő tartályban elhelyezkedő szűrőszivacs tartójának műszaki rajza:



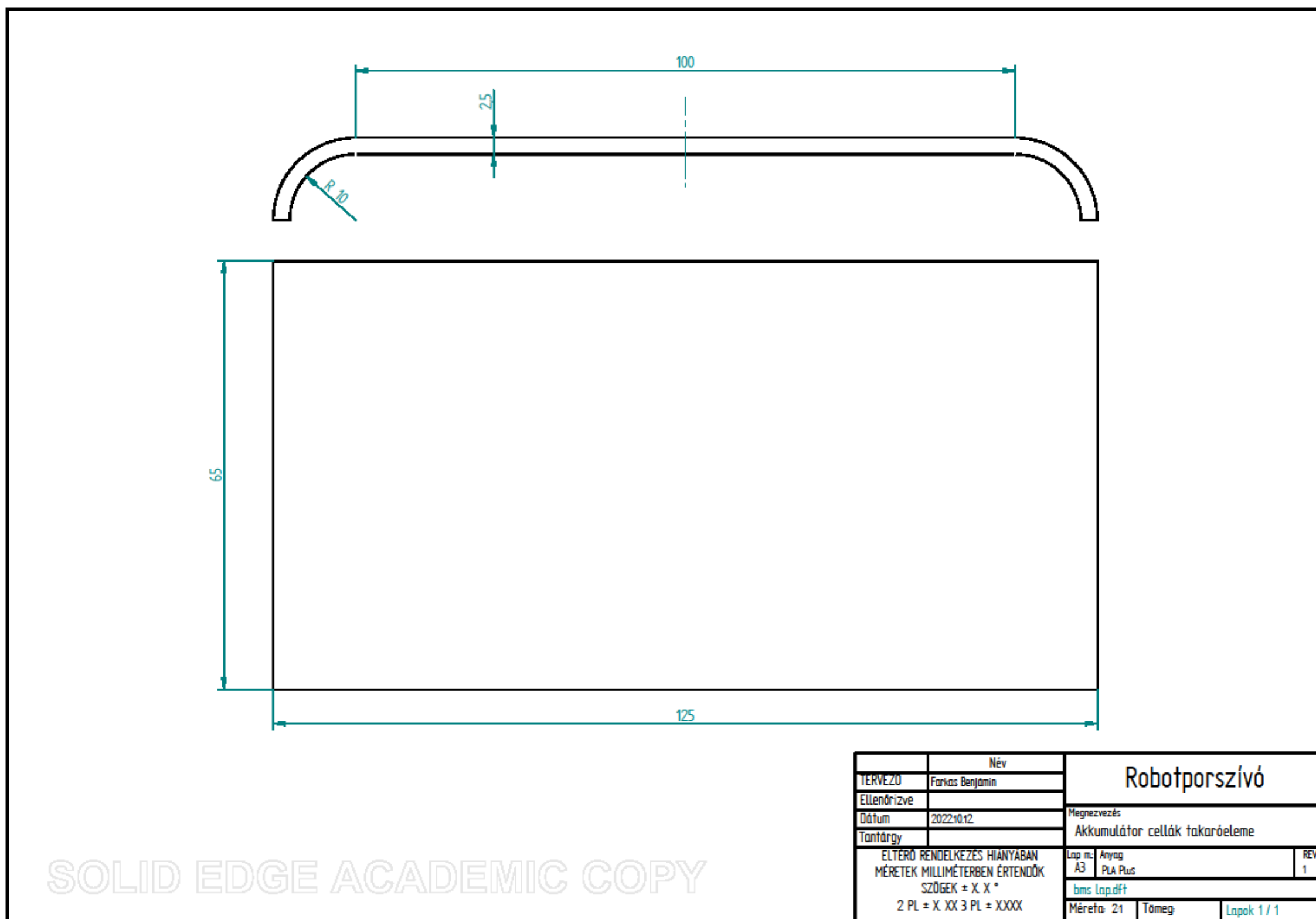
11. sz. melléklet: A porgyűjtő tartály, valamint a motorház tetejeként szolgáló elem műszaki rajza:



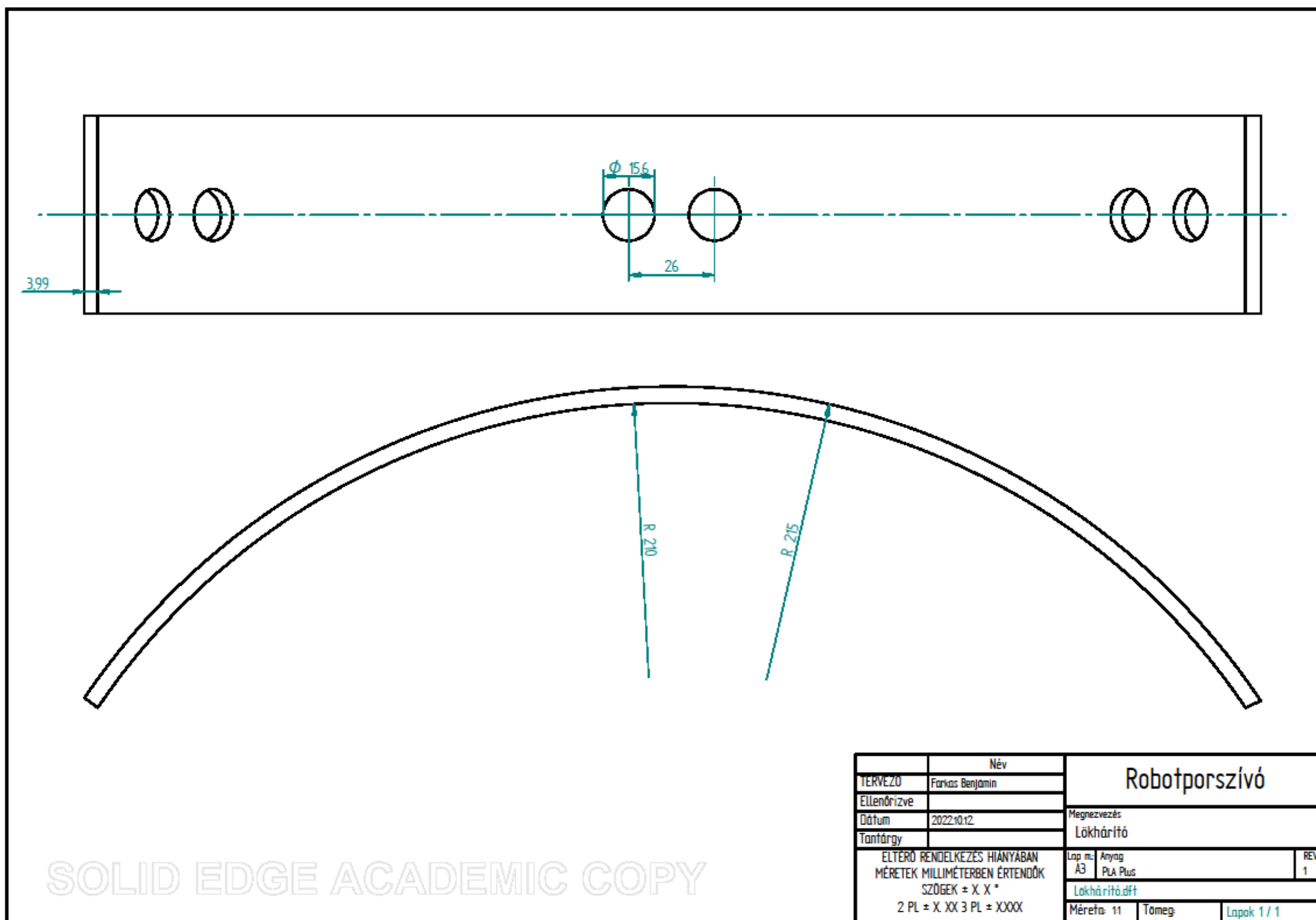
12. sz. melléklet: Az akkumulátortartó tokok talpjaként szolgáló elem műszaki rajza:



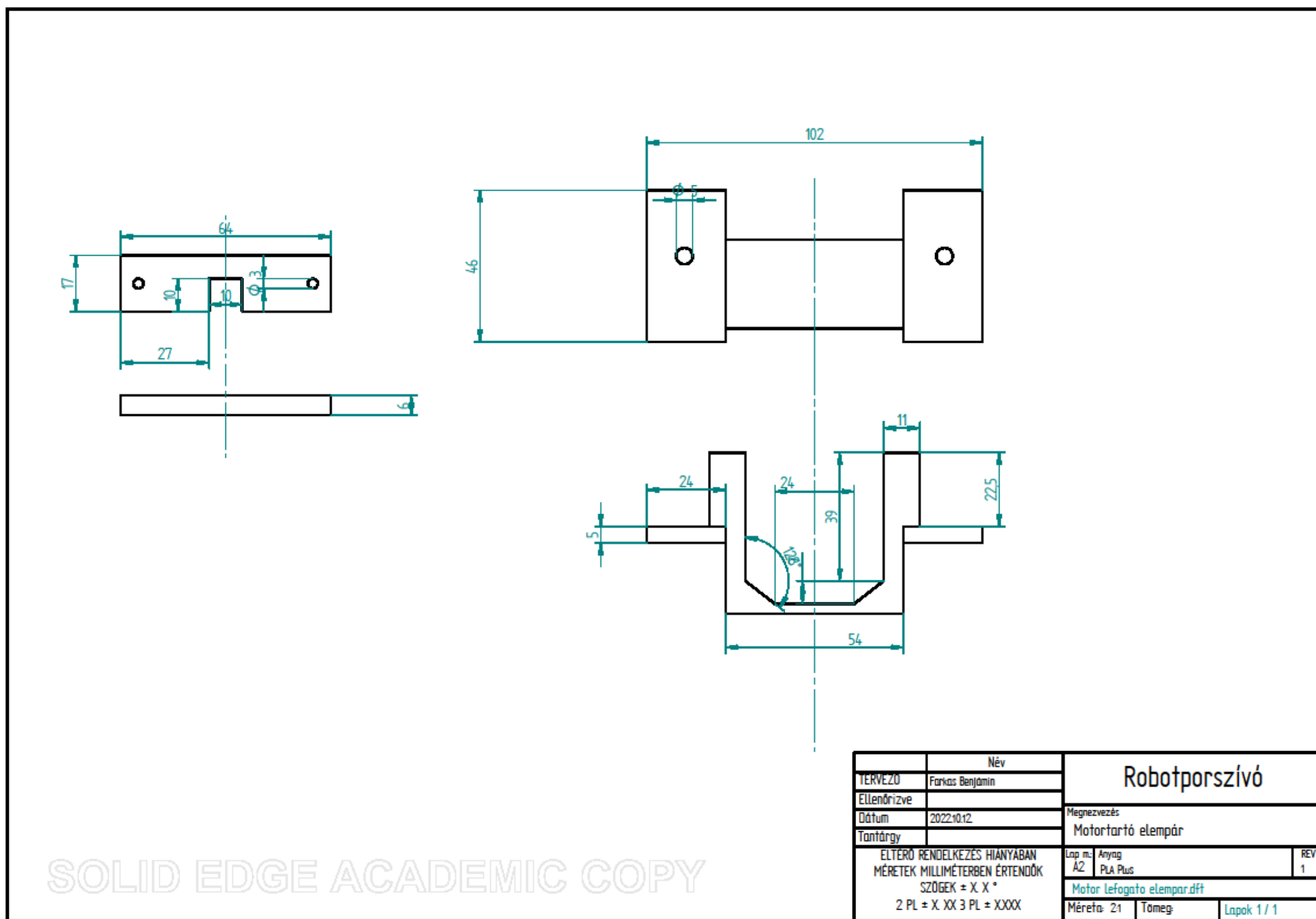
13. sz. melléklet: Az akkumulátor cellák takaróelemének műszaki rajza:



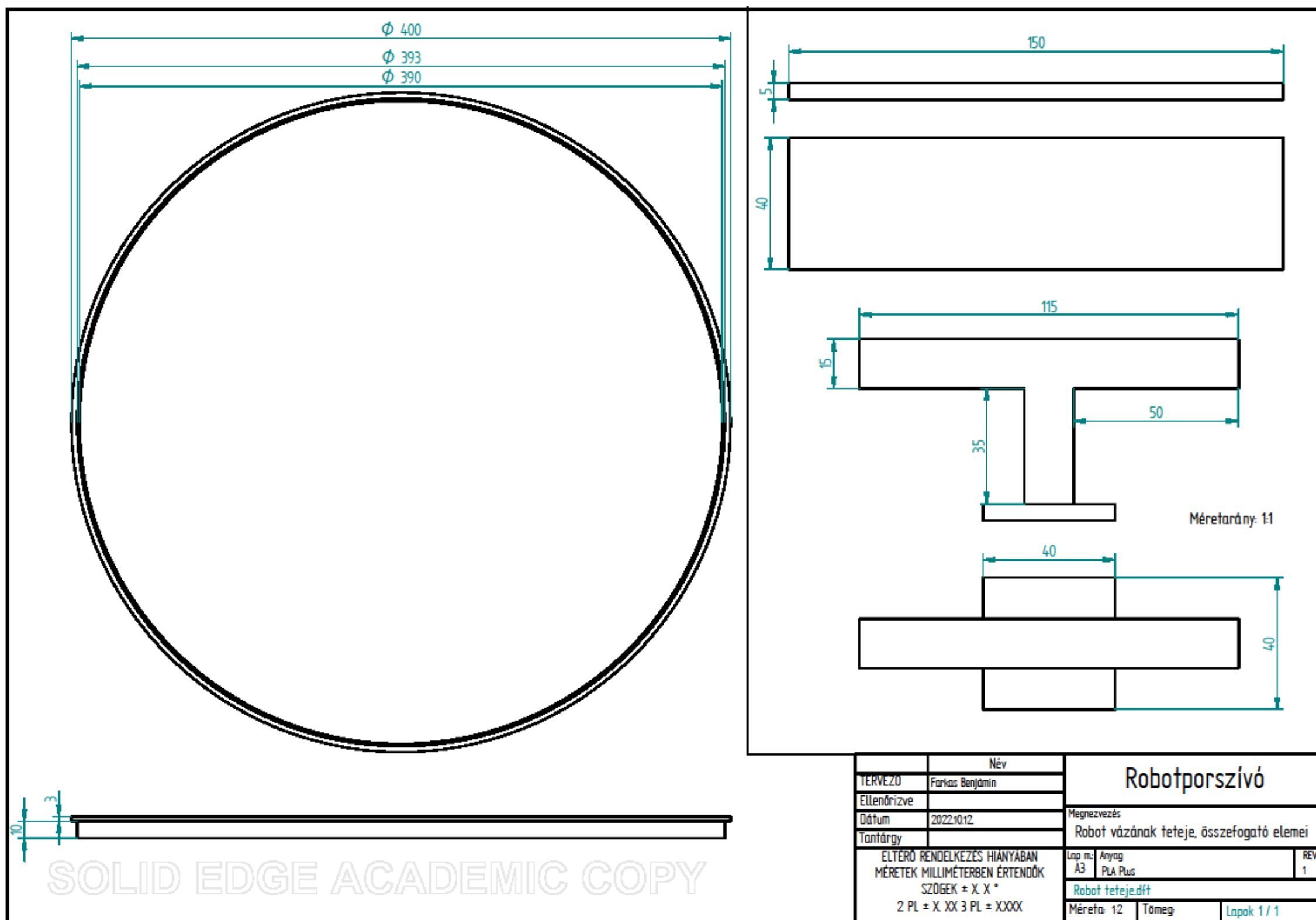
14. sz. melléklet: A lökhárító műszaki rajza:



15. sz. melléklet: A léptetőmotorok házaként szolgáló elempárok műszaki rajza:



16. sz. melléklet: A robot tetejének műszaki rajza:



17. sz. melléklet: A léptetőmotorok házaként szolgáló elempárok műszaki rajza:

