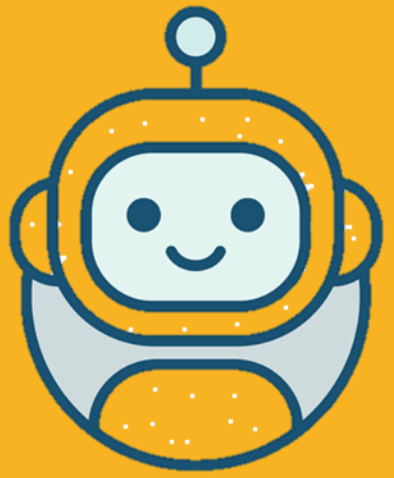
EASY ROBOTICS FORMATION FOR MAKERS

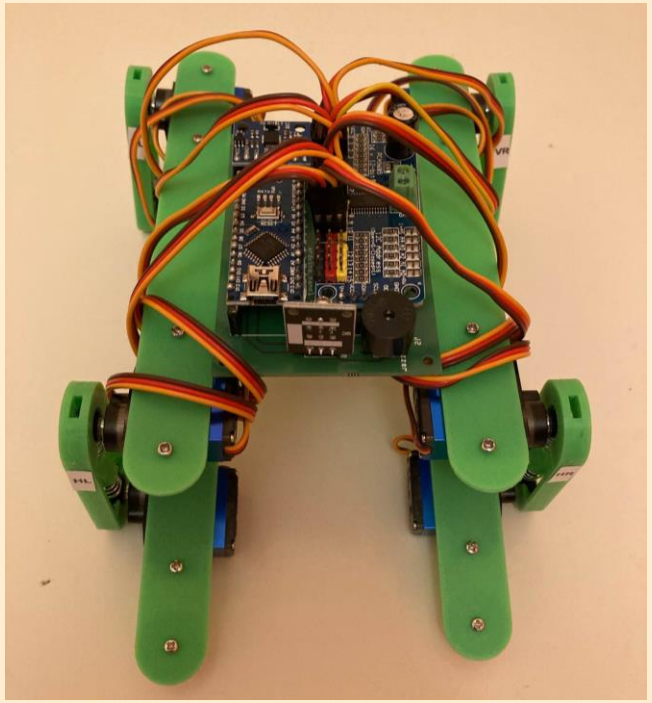
MINICAT WORKSHOP & CURRICULUM



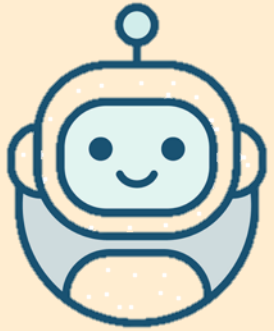
MINICAT WORKSHOP

Affordable robotics for makers

- Build your full robot from scratch !
- Workshop at Schools / MakerFabs

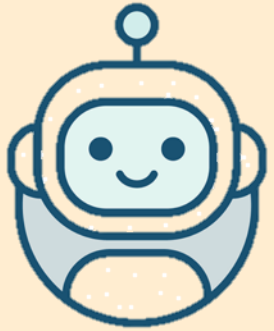


AGENDA



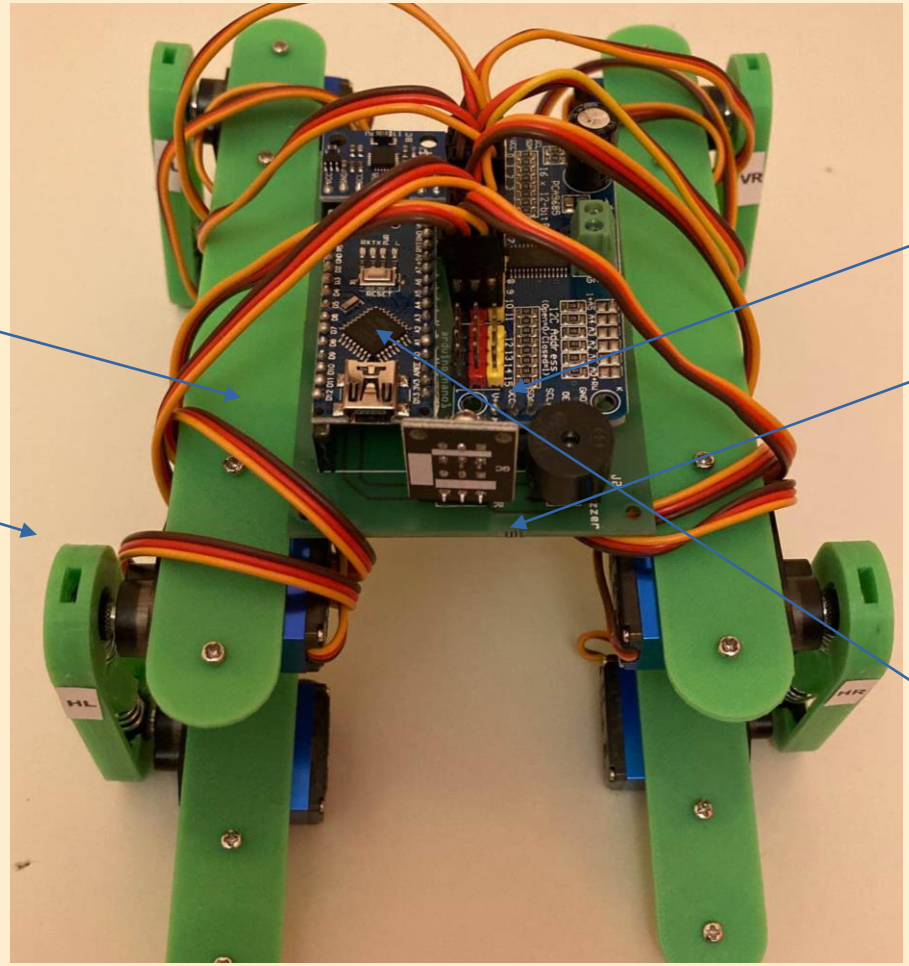
- Workshop Introduction
- 3D Parts models & Printing
- Hardware Modules & Test
- Final Software Integration

AGENDA



- **Workshop Introduction**
- 3D Parts models & Printing
- Hardware Modules & Test
- Final Software Integration

Workshop Overview
MiniCat Robot modules



1. 3D Printed parts

Body structure

4 legs

2. Electronics

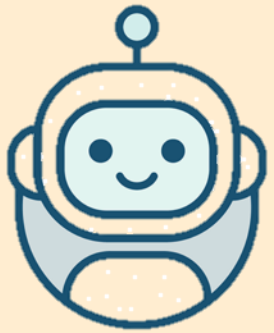
Electronics Modules on custom PCB (green)

3. Software

Running on Arduino Module

Workshop Overview

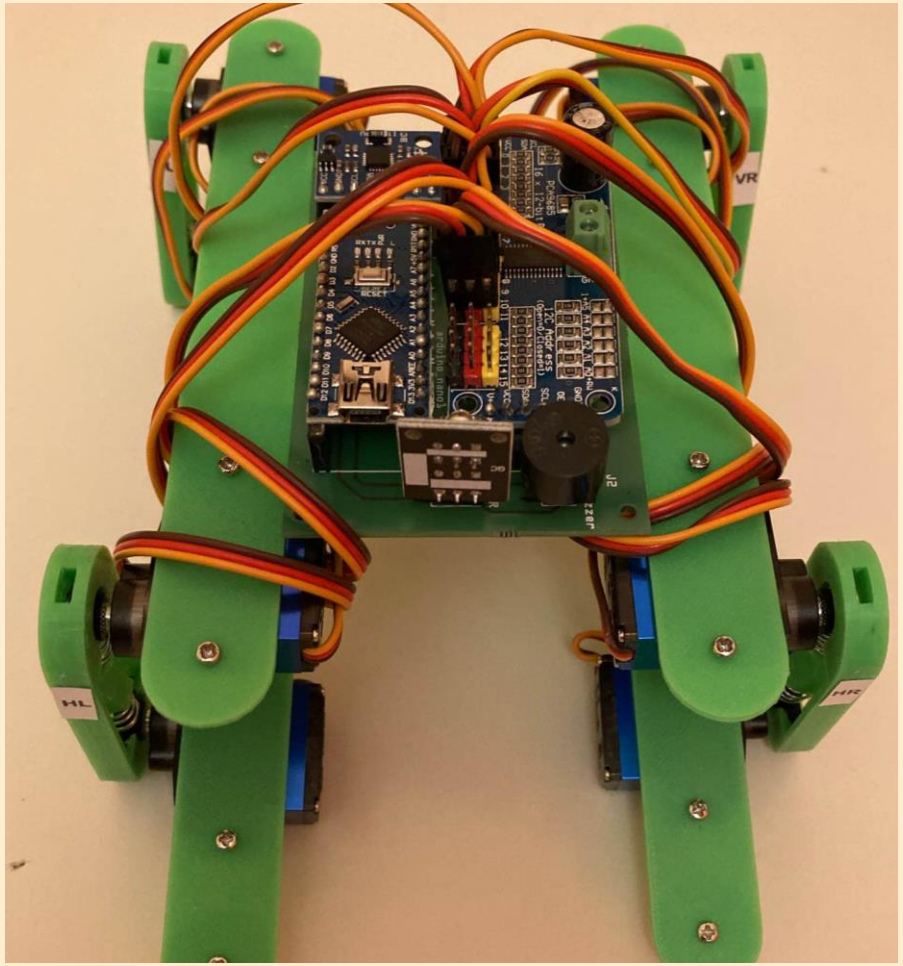
Maker tools installation check w/ participants



1. Tools for 3D Print

- Modelling
- **Fusion360** tool
- .stp files

- Slicing/Printing
- **Cura** tool
- .STL files



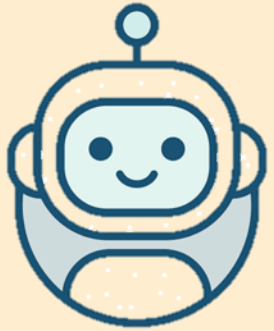
2. Tools for Electronics
- no in scope

3. Software

- Development Environment
- **Arduino IDE** tool

Workshop Overview

Toolset installation (to do before workshop)

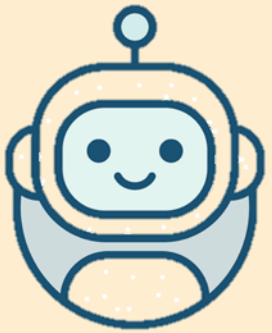


- **Open : https://roboticsformakers.com/index.php/maker-software-tools/**

- **Follow the instructions to install all the Maker tools**

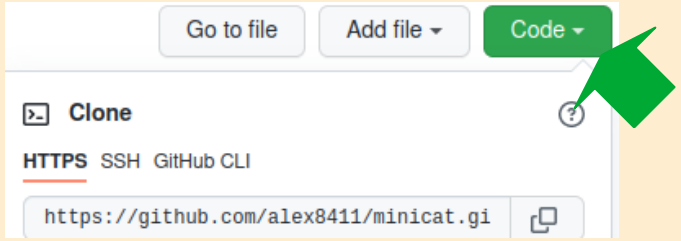
Workshop Overview

Software download



- Open : https://github.com/alex8411/minicat

- Click on “Code” then “Download ZIP”:



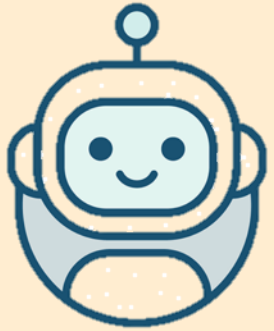
- Store the ZIP in your preferred Folder & extract it to get “minicat-main”

ToDo :

- Explore the folder especially :

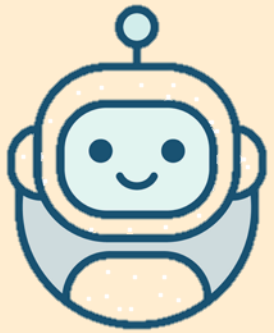
- **“3D_printing”**
  - 3D parts for modelling & 3D print
- **“Software” for MiniCat Modules**
  - **GIT** .ino files
  - Remark: Libraries to install separately

AGENDA

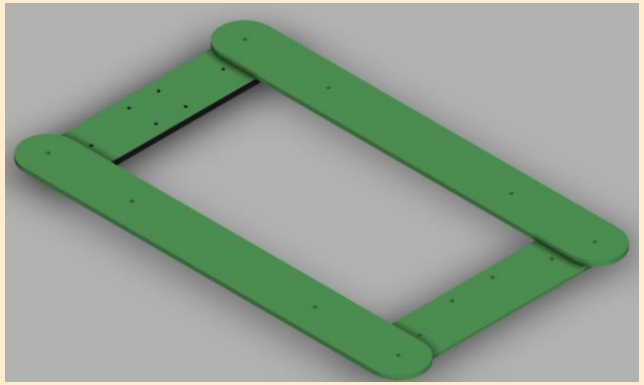


- Workshop Introduction
- **3D Parts models & Printing**
- Hardware Modules & Test
- Final Software Integration

3D parts models & printing



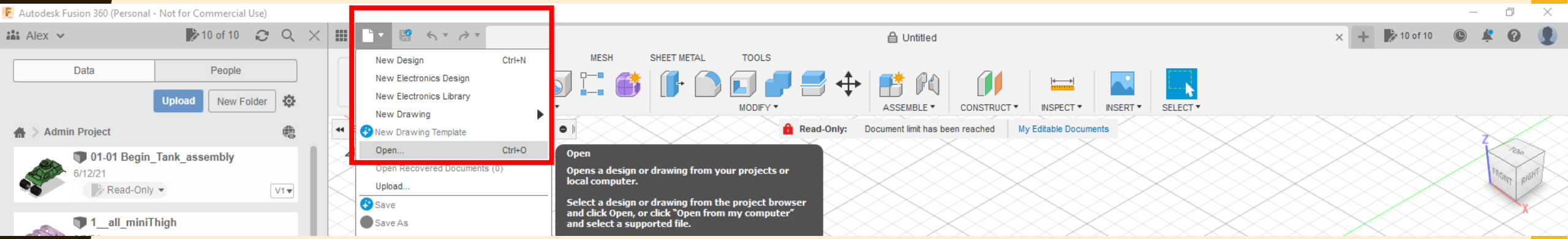
ToDo (.f3d assemby opening)



- Open **Fusion360**

- Click on “**File/Open/Open from my computer**”

- Navigate to “minicat-main/CAD/” & click “**MiniCat_full_assembly_v32.f3d**” (works only on Windows 10)



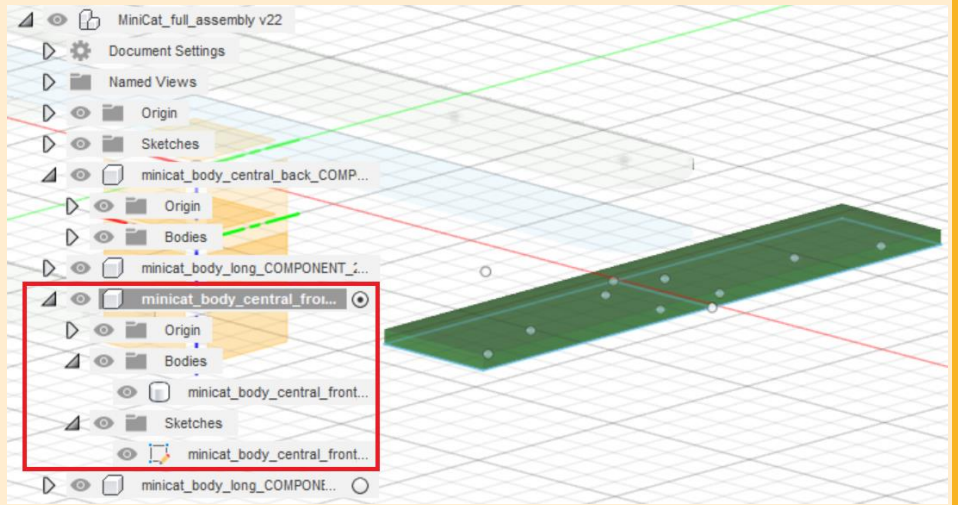
3D parts models & printing

ToDo (Sketch opening & modifying)

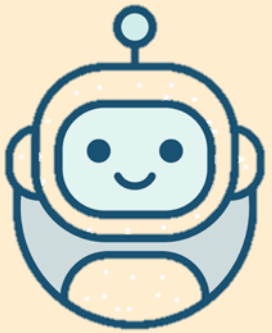
- Navigate to:
    - “minicat_body_central_front_COMPONENT”
        - “Sketches”
            - “minicat_body_central_front_SKETCH”

- Right click on: “minicat_body_central_front_SKETCH”
    - “Edit Sketch”

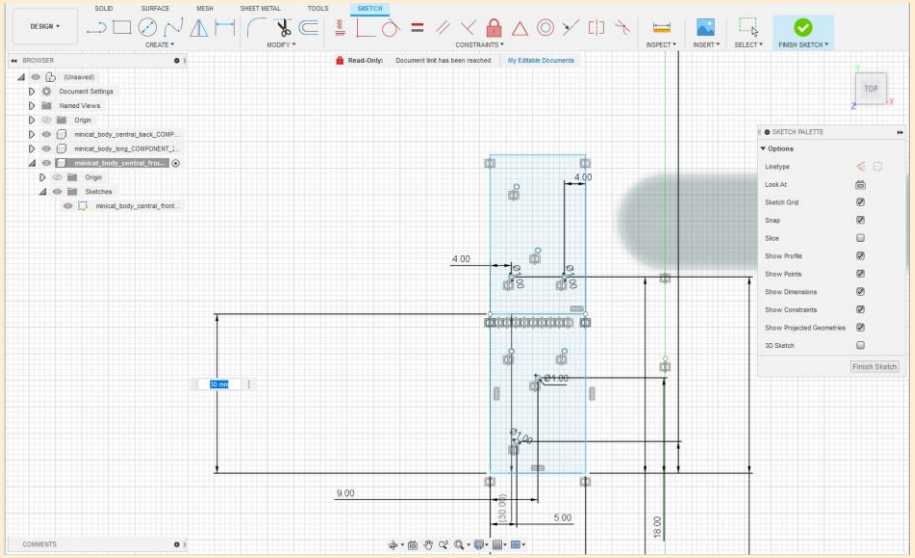
- Try to modify it

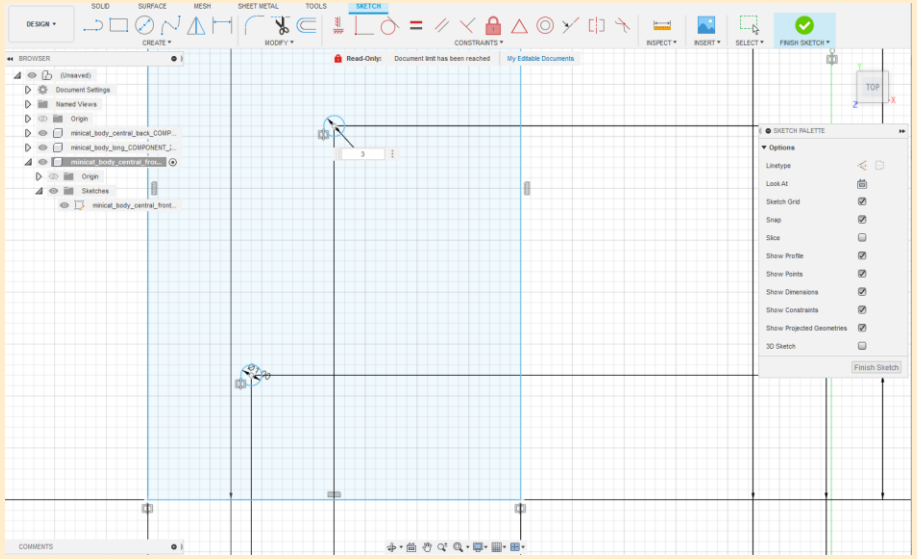


3D parts models & printing



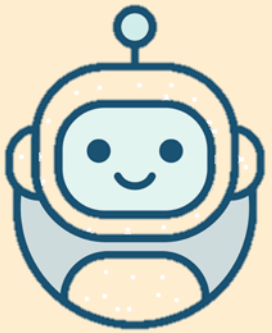
- **ToDo Editing the Sketch front part : rectangle and hole dimensions:**



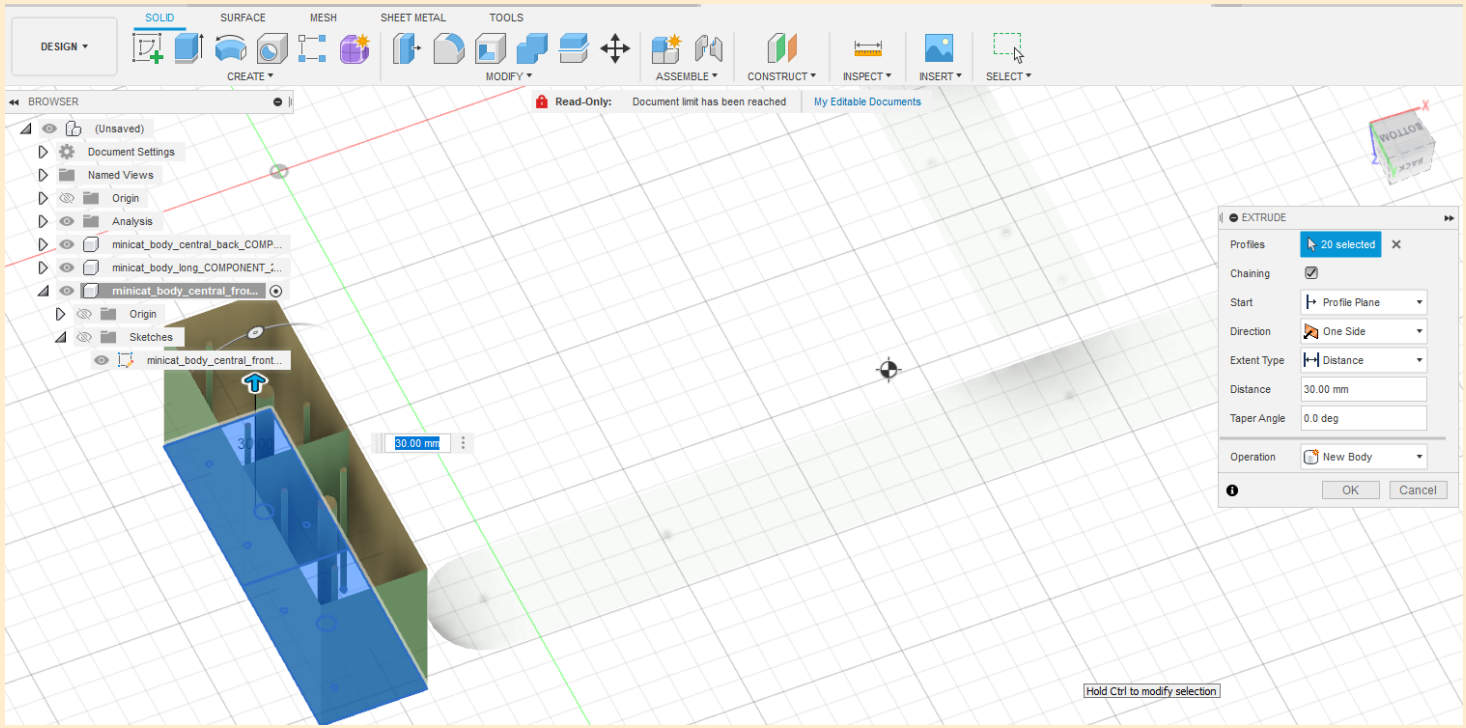


- click on “Finish Sketch” and observe the changes

3D parts models & printing

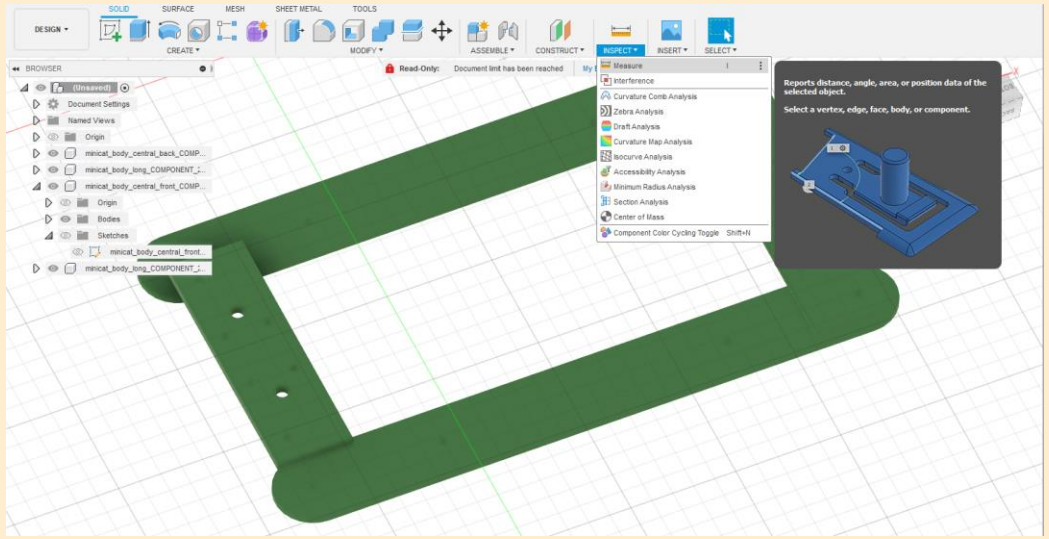


- **Changing the Extrusion dimension :**

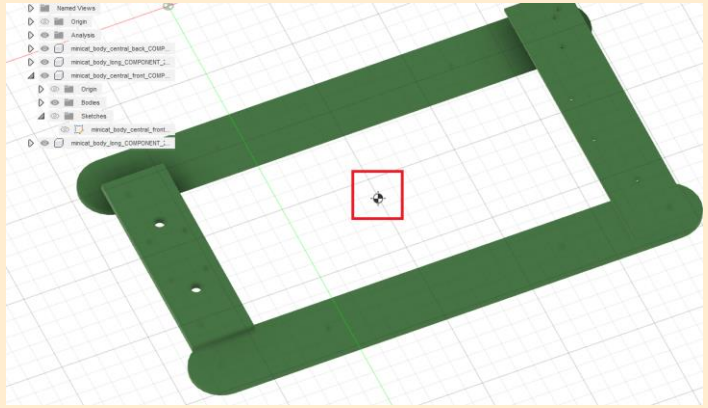


3D parts models & printing

- - measuring with “Inspect” a dimension :



- - and gravity center :

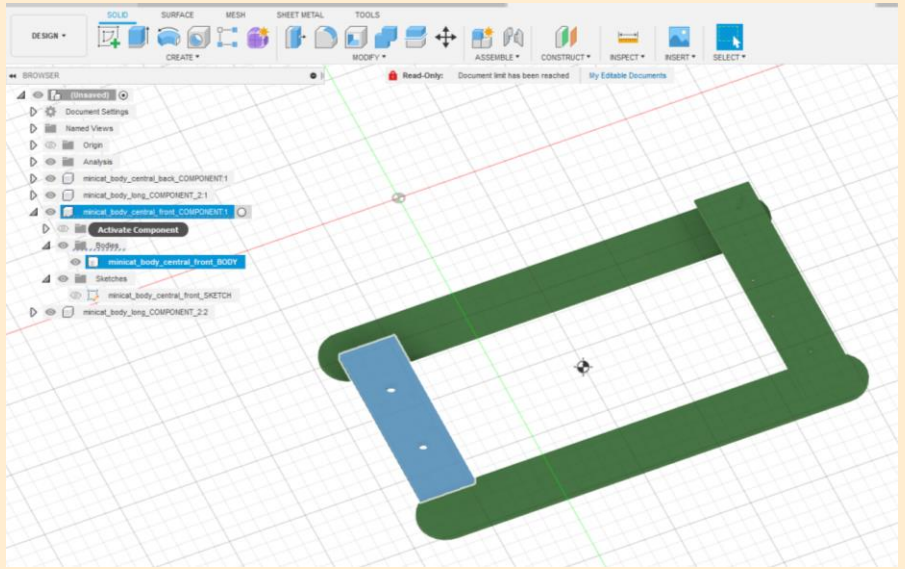


3D parts models & printing

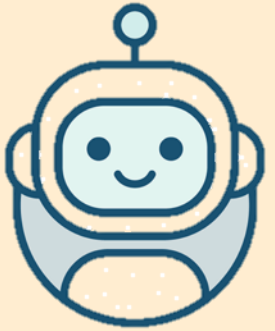
ToDo (.STL file generation)

- Navigate to: “**minicat_body_central_front_COMPONENT**”
  - “**Bodies**”
    - “**minicat_body_central_front_BODY**”

- “Activate” the “**minicat_body_central_front_COMPONENT**”



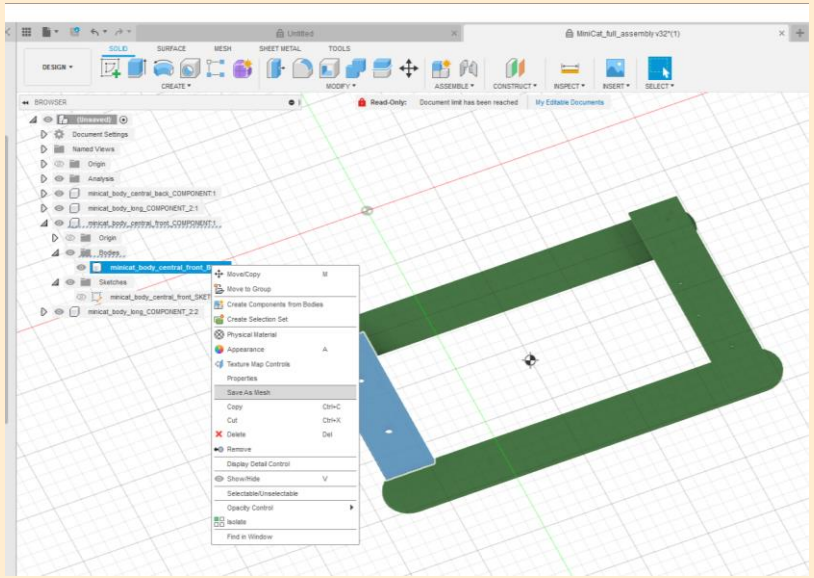
3D parts models & printing



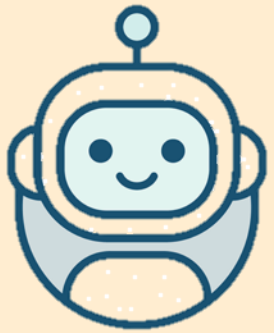
ToDo (.STL file generation)

- Right click on “minicat_body_central_front_BODY”

- “**Save as Mesh**” (=“als Netz speichern”)
/ and select Format “**STL (Binary)**” / “Save”



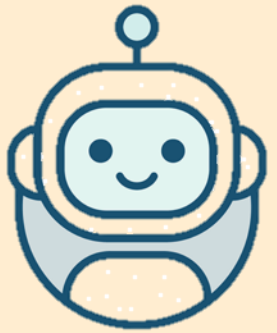
3D parts models & printing



ToDo (.STL file generation)

- Open CURA and Add a Non Networked Printer “**Creality Ender3 Pro**”
- “File/**Open file(s)” the .STL file you just generated** !
- “**Slice**” it !
- Check the “**Preview**” / click on “Play” button
- “**Save to Disk**” to save it as a 3D Printer Machine code **(.gcode**)

3D parts models & printing
.STL file, generate your own



ToDo (Opening Toes .STL, Slicing & Simulation)

- For the Workshop we will not use the previously generated Body STL file but use an existing .STL : "**minicat-main/3D_printing/STL/3__all_miniToe_rubber.stl**"

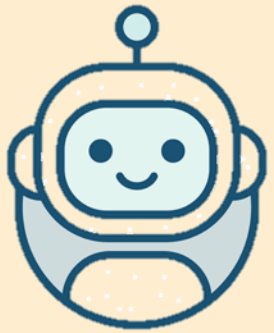
- Background :
  Printer is now optimized for TPU ("gummi") print, not PLA rigid plastic

- Open CURA

- Open "minicat-main/3D_printing/STL/3__all_miniToe_rubber.stl"

- **"Slice" and write the time**

3D parts models & printing
.STL file, generate your own



ToDo (3D printer config for TPU)

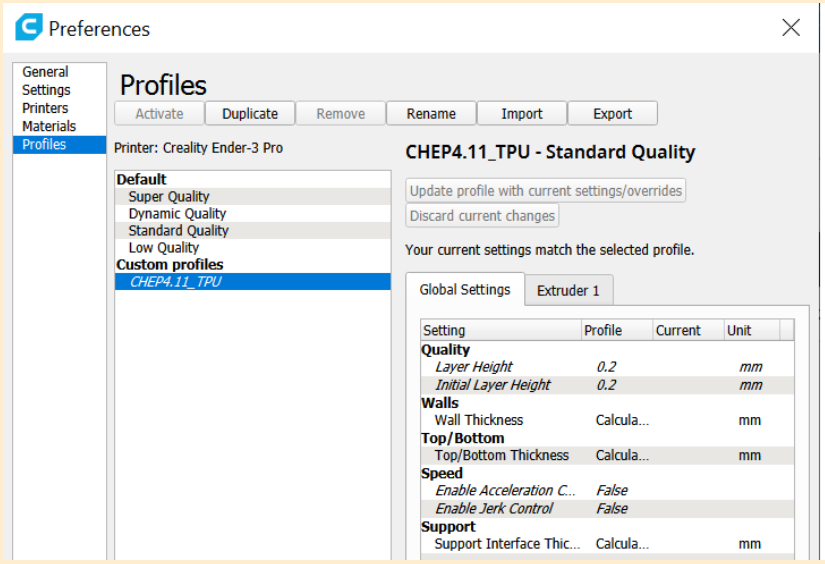
- Open Cura
- click on “Preferences/Configure Cura/Profiles

- ”Import” the content of “CHEP4.11_TPU” profile
  from “minicat-main/3D_printing/printing_profiles/ender3pro/tpu”

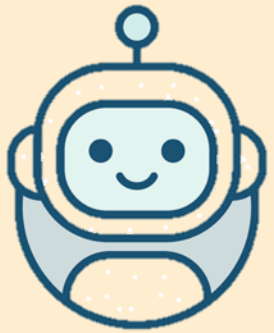
- ”Activate” the profile (work on on Mac, Windows10)

- Open your previously generated .STL file

- Slice it and check if the simulated printing time is different
  with the TPU profile vs “Default/Standard Quality” profile
- **“Save to Disk”** (**.gcode** machine code)



3D parts models & printing

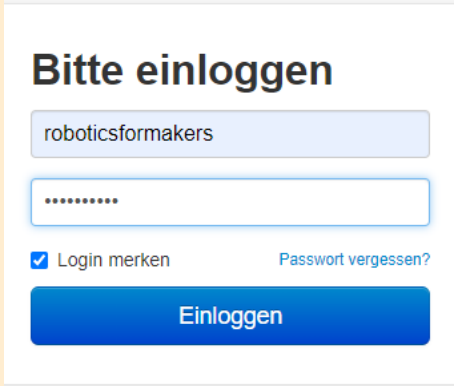


ToDo (Printing !)

3 ways to print a .gcode file :

1- SD Card:
- save generated .gcode and plugin in printer

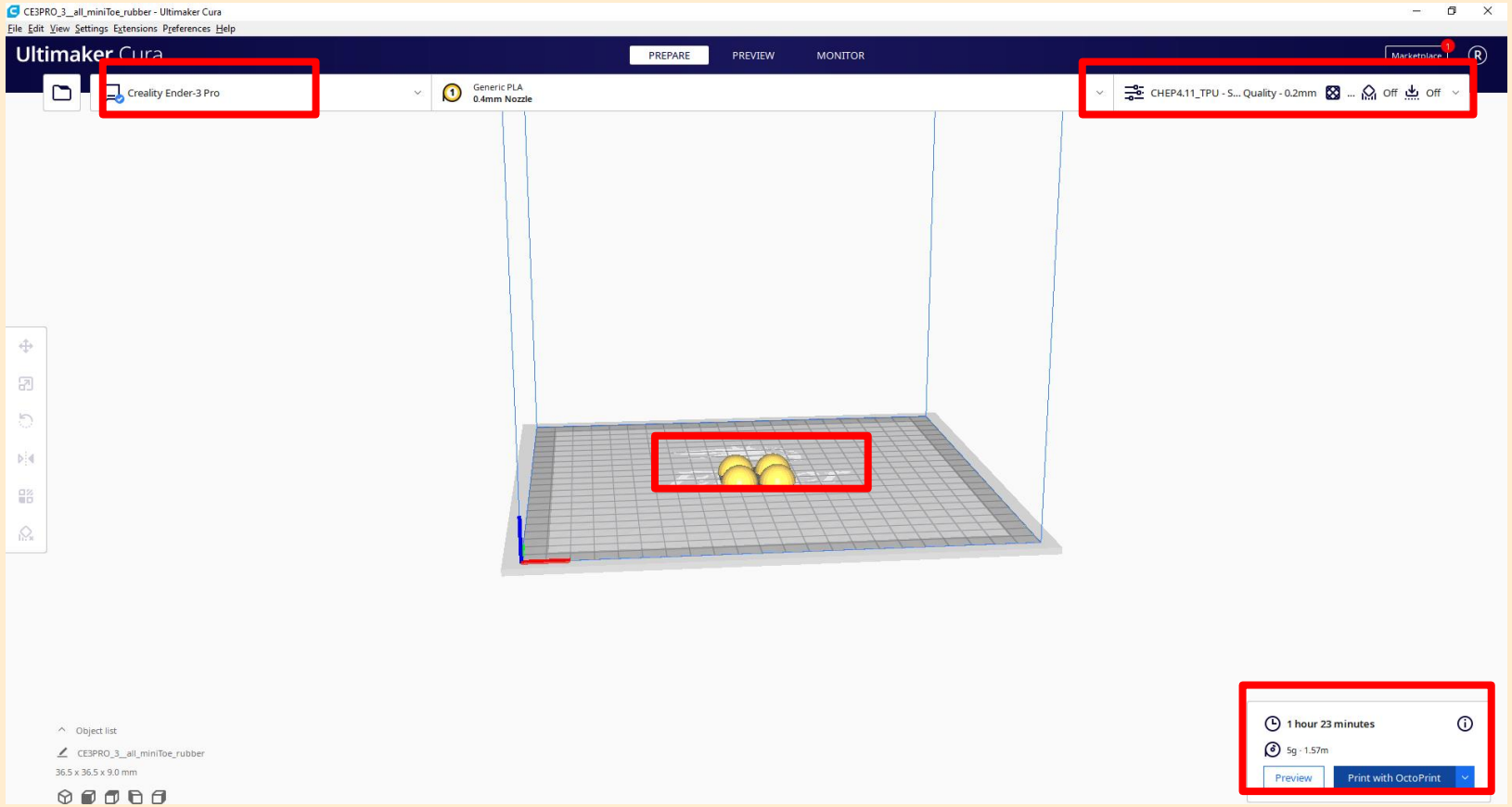
2- Octoprint Server on Raspi:
<u>- WLAN :</u>
connect to **HUAWEI** WLAN / Password: **2Groesser?**
<u>- Server :</u>
- open http://octopi.local/login in your browser
- login: **roboticsformakers** / Password: **2Groesser?**

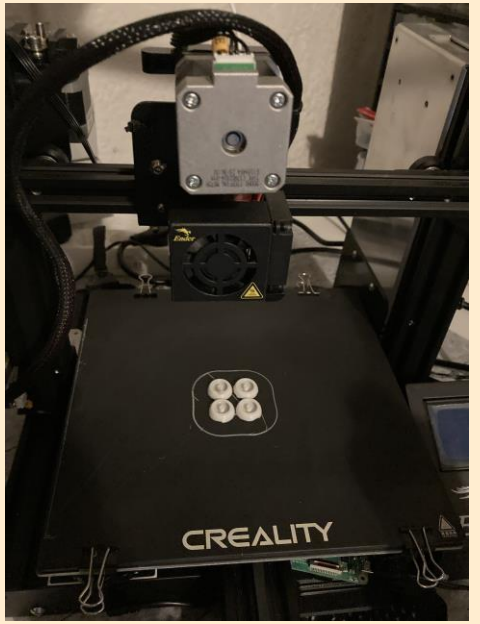


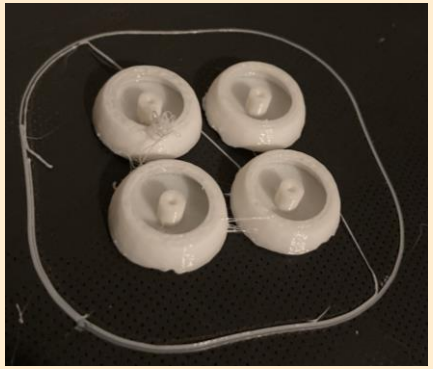
3- Octoprint Plugin for CURA: -> Focus today
- connect to HUAWEI WLAN / Password: 2Groesser?
- open CURA and install Octoprint Plugin
- start the “Preparation” and Print

3D printing

I loaded the TPU (flexible filament profile) in CURA and generated the .gcode for you :



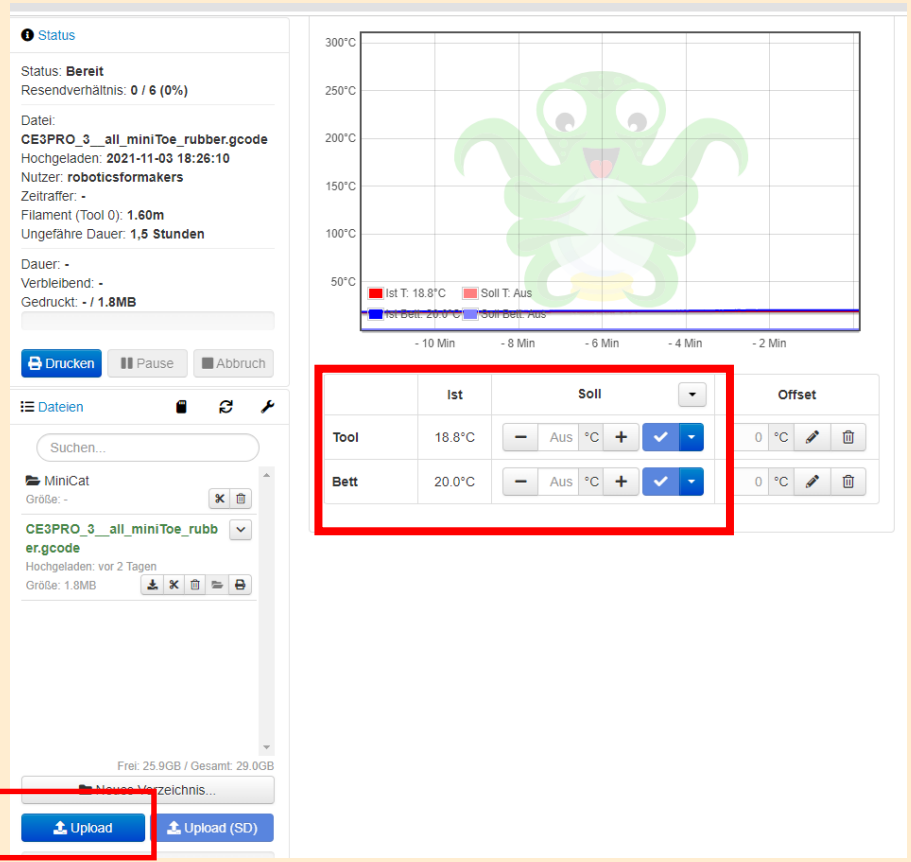


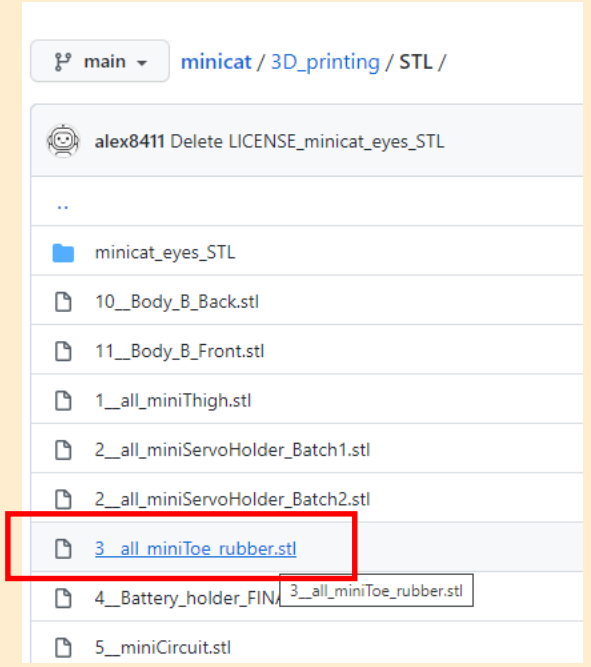


3D parts models & printing

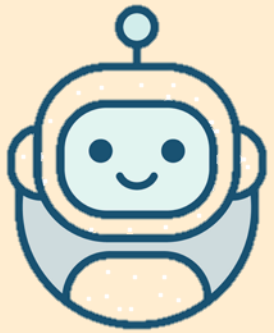
- Try to change some params : in “**Temperatur**” “Soll” for “**Tool**” and “**Bett**”
  - click on “**Upload**”

- - choose “minicat-main\3D_printing\gcode\creality_ender3pro\ CE3PRO_3__all_miniToe_rubber.gcode



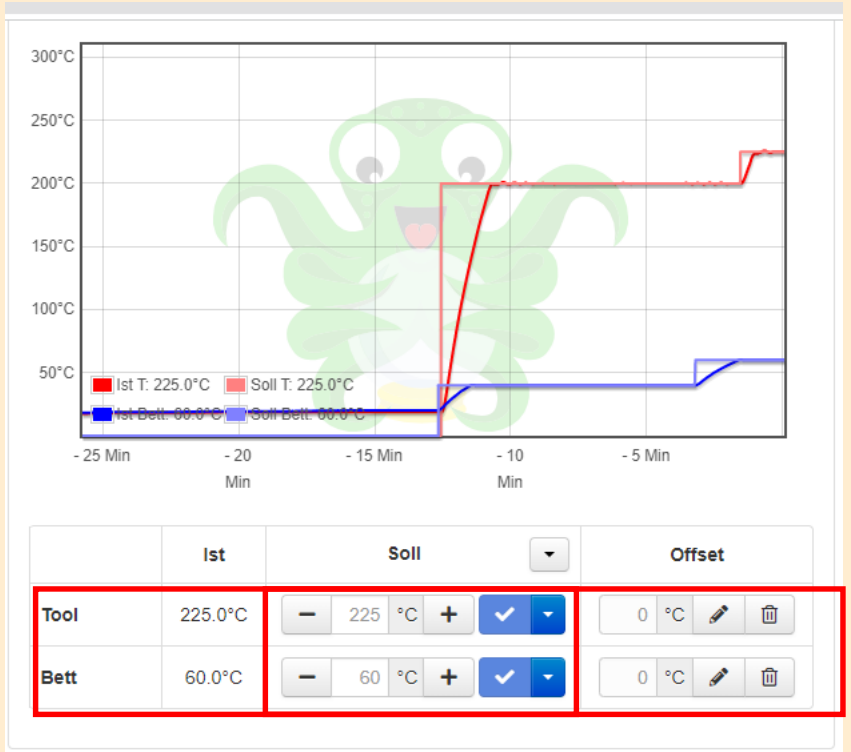


3D parts models & printing

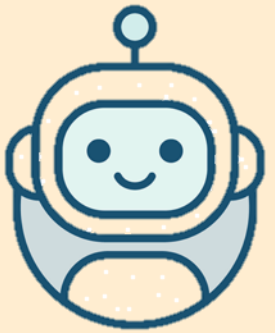


3D print w/ Octoprint :

- Preheat the Tool (Extruder) at 225°C
- Preheat the Bed at 60°C



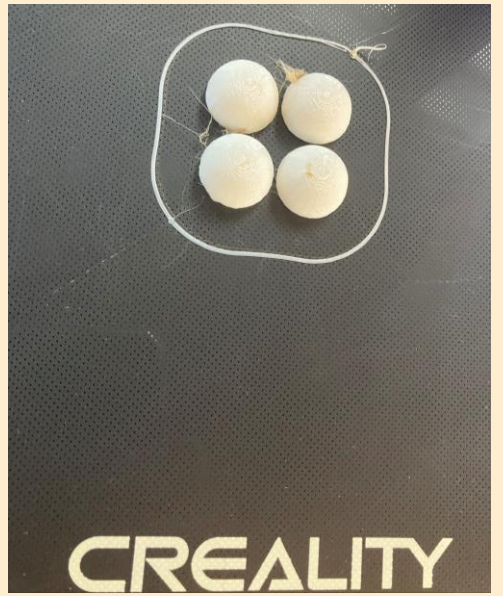
3D parts models & printing



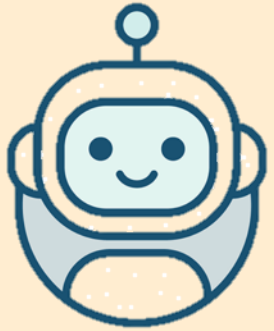
- **IMPORTANT**: Ask the teacher first !
- **only one participant can start the printing after another !**

End of the 2h

After the 2 hours break :



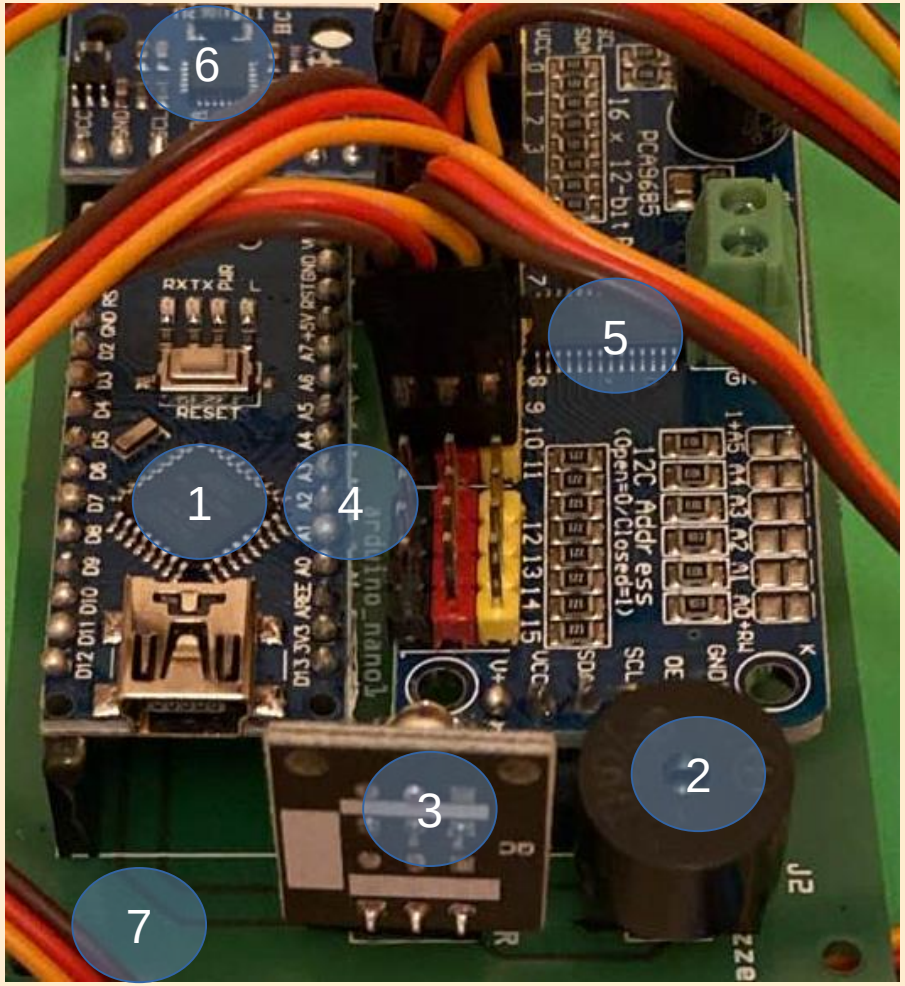
AGENDA



- Workshop Introduction
- 3D Parts models & Printing
- **Hardware Modules & Test**
- Final Software Integration

Hardware Modules & Test

Overview



Module 1: Arduino board
- “Brain” of the robot, runs Software

Module 2: Buzzer
- “Alive” signal generation

Module 3: Infrared Receiver & Remote
- Remote control

Module 4: I2C Bus
- comm. Betw. Arduino, Servomotors & Gyro boards

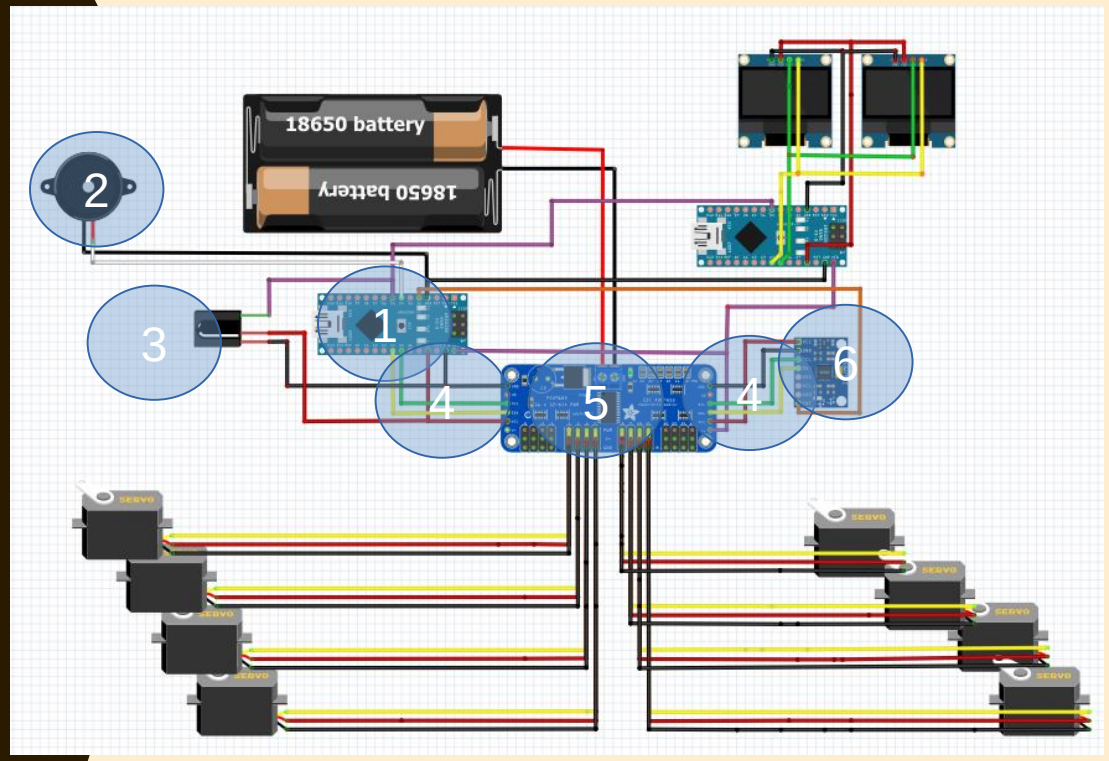
Module 5: Servomotors driver board
- drives the 8x Servomotors

Module 6: Gyroscope board
- roll/pitch/yaw meas. vs calibrated 0 for robot balance

Module 7: PCB w/ all Modules & Final Software

Hardware Modules & Test

Overview



Module 1: Arduino board

- “Brain” of the robot, runs Software

Module 2: Buzzer

- “Alive” signal generation

Module 3: Infrared Receiver & Remote

- Remote control

Module 4: I2C Bus

- comm. between Arduino, Servomotors & Gyro boards

Module 5: Servomotors driver board

- drives the 8x Servomotors

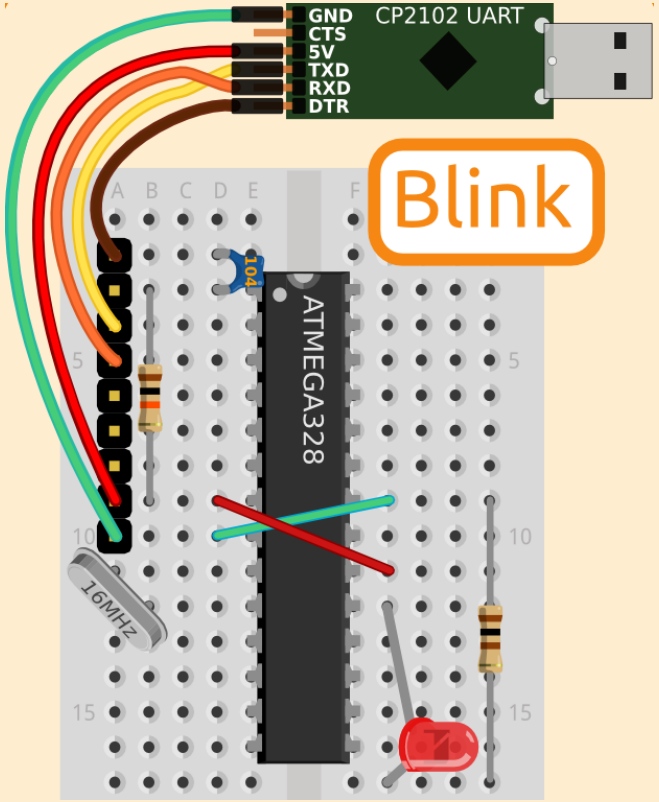
Module 6: Gyroscope board

- roll/pitch/yaw meas. vs calibrated 0 for robot balance

Module 7: PCB w/ all Modules & Final Software

Hardware Modules & Test

Module 1: Arduino Board, blink Led



Hardware Modules & Test

Module 1: Arduino Board

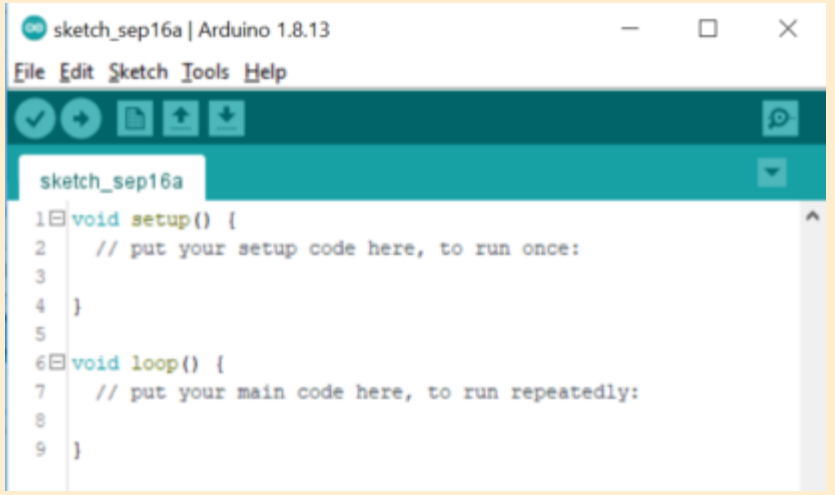
Your Laptop USB

Module 1: Arduino board & blink Led

- “Brain” of the robot, runs the Software

ToDo:

- Plugin the provided USB cable to your Laptop
- Start the Arduino IDE

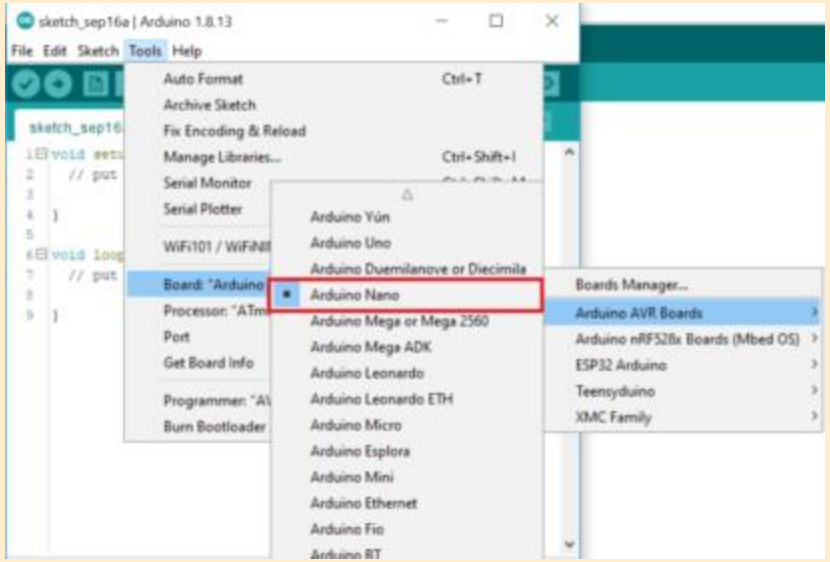


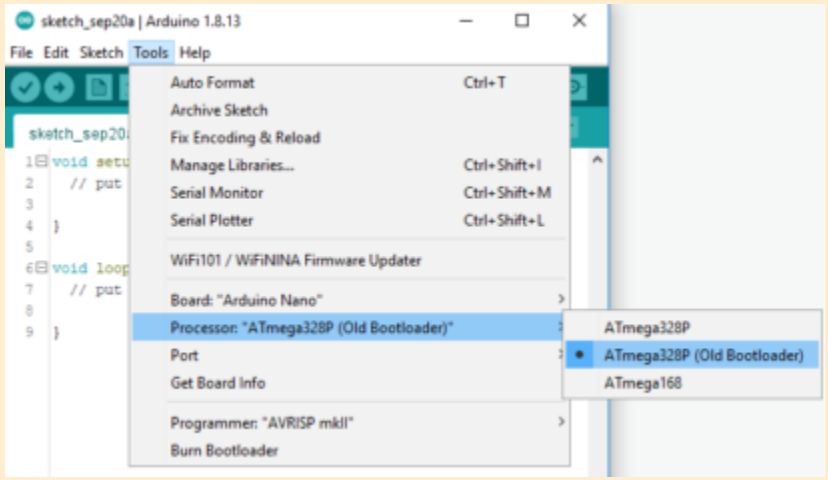
Hardware Modules & Test

Module 1: Arduino Board

Your Laptop USB

ToDo: - Connect Arduino Nano Board w/ PC over USB
- In the IDE, choose the Serial Port "**COM***"
- Choose the Board "**Arduino Nano**"
- Choose the Bootloader "**Old Bootloader**"





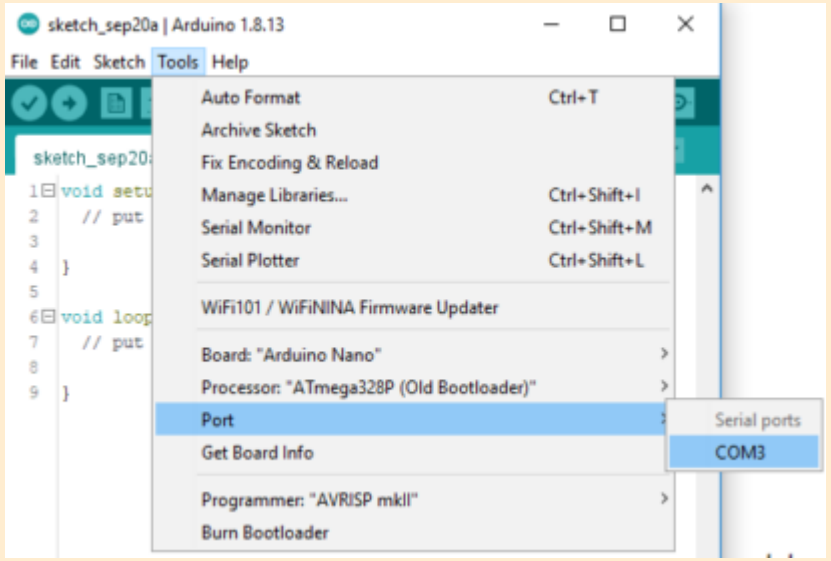
Hardware Modules & Test

Module 1: Arduino Board

Your Laptop USB

ToDo:

- In the IDE, choose the USB / Serial Port you are using :



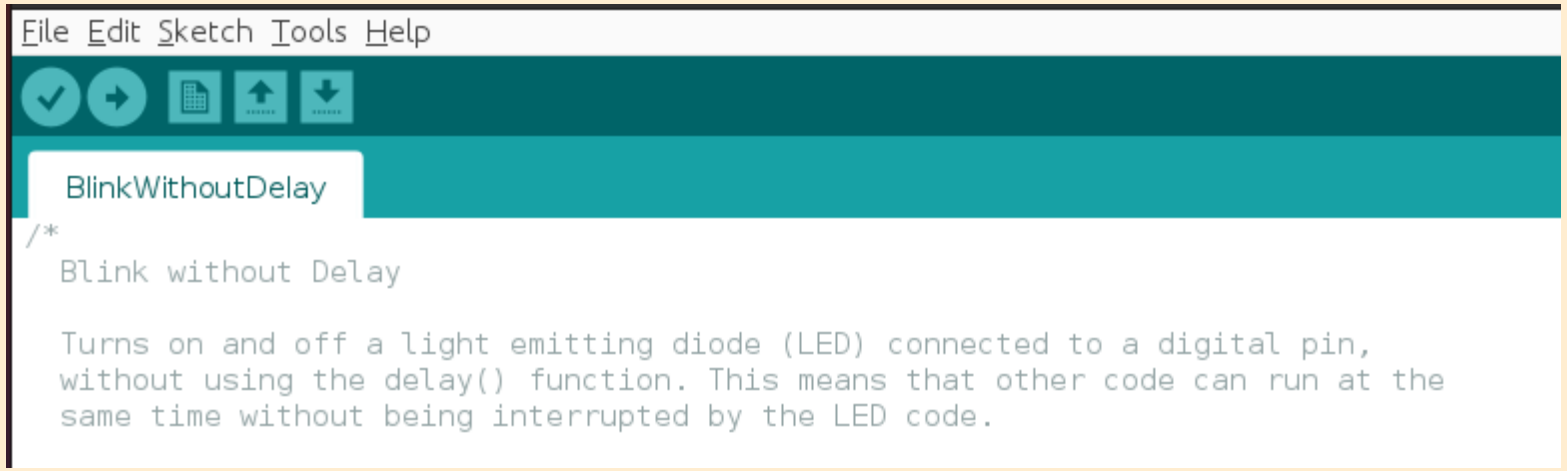
Hardware Modules & Test

Module 1: Arduino Board

Your Laptop
USB

ToDo:

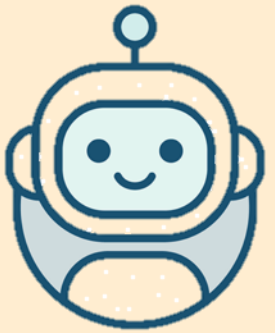
- Click on ”File/Examples/Digital/**Blink without delay**”



- For Windows 10, please install **USBtoSerial driver „CH340G**
this is part of MiniCat GIT, see Software folder (or „DirectLink“)
- MacOS and Ubuntu 20.04, no problems, no driver needed

Hardware Modules & Test

Module 1: Arduino Board & Led



BlinkWithoutDelay.ino

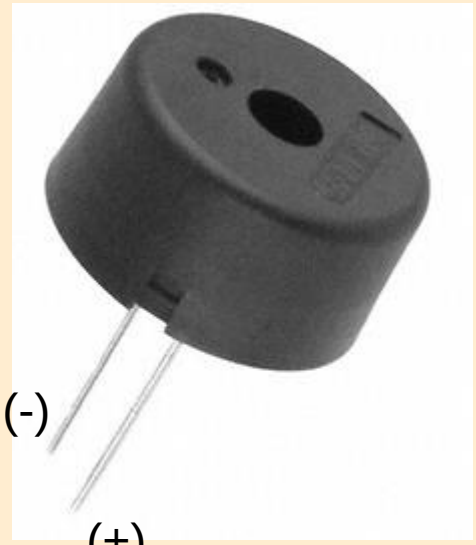
```
// constants won't change. Used here to set a pin number:
const int ledPin =  LED_BUILTIN;// the number of the LED pin

// Variables will change:
int ledState = LOW;             // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;        // will store last time LED was updated

// constants won't change:
const long interval = 5000;           // interval at which to blink (milliseconds)
```

ToDo (Software):

- Change the **blinking interval** to 5000 (default 1000=1s)
- Click on **“Upload”** to start the Software
- **Observe the result on the Arduino board**

Hardware Modules & Test

Module 2: Buzzer



Module 2: Buzzer

- “Alive” signal generation

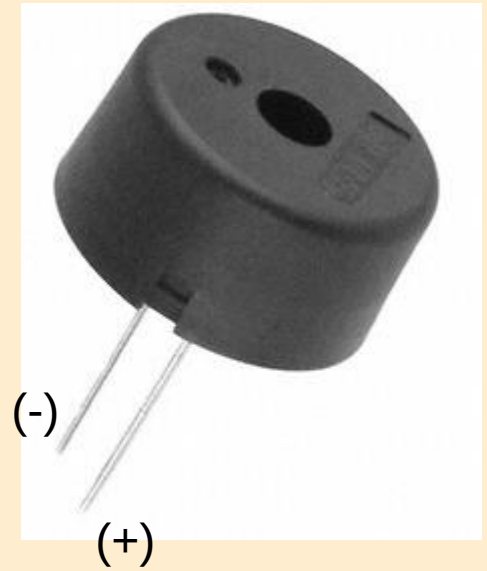
ToDo:

- Unplug the USB cable from your Laptop
- Plugin the **Arduino &** **Servodriver board** on the PCB
- Plugin the USB cable to your Laptop

- open the buzzer test software : double click on “minicat-main/Software/buzzer_test/minicat_buzzer_test.ino”

Hardware Modules & Test

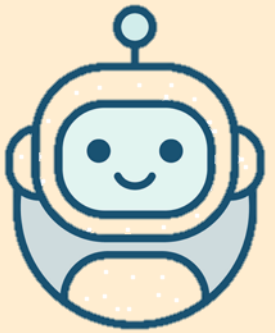
Module 2: Buzzer, modify the code adding:




```
int counter = 0;

void loop() {

  while (counter < 5)

  {

   byte melody[] = {8, 13, 10, 13, 8, 0, 5, 8, 3, 5, 8,    // definition of the tones list as a "melody"
                    8, 8, 32, 32, 8, 32, 32, 32, 32, 32, 8
                    //8,8,16,16,8,16,16,16,16,8
                   };
  playMelody(melody, sizeof(melody) / 2); // we play the melody

  }

}
```

Hardware Modules & Test

Module 2: Buzzer



minicat_buzzer_test.ino

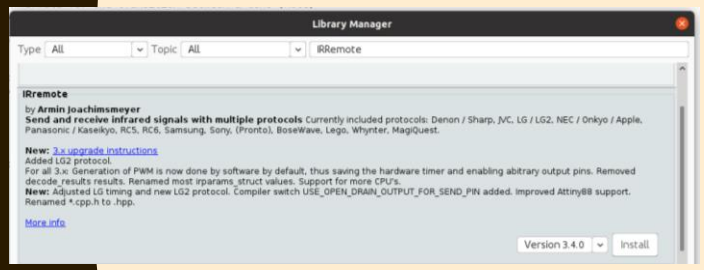
```
void loop() {


  byte melody[] = {8, 13, 10, 13, 8, 0, 5, 8, 3, 5, 8,   // definition of the tones list as a "melody"
                   8, 8, 32, 32, 8, 32, 32, 32, 32, 32, 8
                  };
```

ToDo (Software):

- Change the melody by changing the values / deleting some
- Click on **“Upload”** to start the Software
- **Observe the result on the Arduino board**

Hardware Modules & Test

Module 3: Infrared receiver & remote



Module 3: Infrared receiver & remote

- Remote Control

ToDo:

- Unplug the USB cable from your Laptop
- Plugin the Infrared receiver on the PCB
- Plugin the USB cable to your Laptop
- Check if the led is ON when you push the remote

- the **Software Library** "IRRemote" should be installed (Tools/Manage Libraries "IRremote")
- **open the IRRemote test software :** double click on "minicat-main/Software/ minicat_IRremote_test /minicat_IRremote_test.ino"

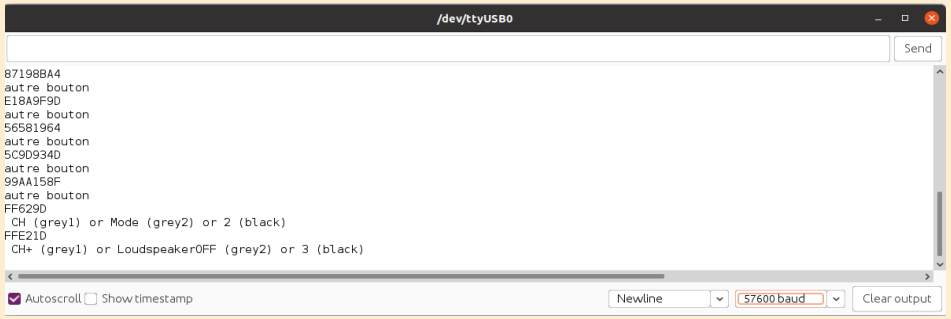
Hardware Modules & Test

Module 3: Infrared receiver & remote

ToDo:

- Open the Arduino **Serial Monitor** (Arduino IDE “Tools/Serial Monitor”)

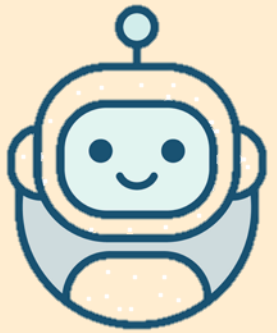
- in serial monitor, set the communication speed to **57600 baud**



- **Push on the buttons** on the Remote control and verify that they are correct in the Serial Monitor

Hardware Modules & Test

Module 3: Infrared receiver & remote



ToDo (Software):

- change a button description :

- push this button on remote

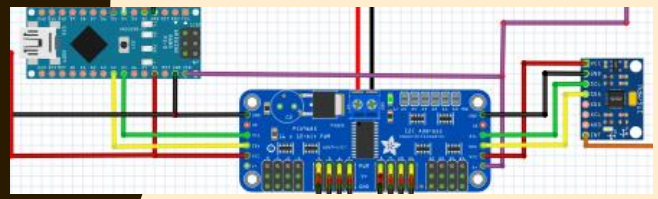
- verify that it is displayed in the serial monitor

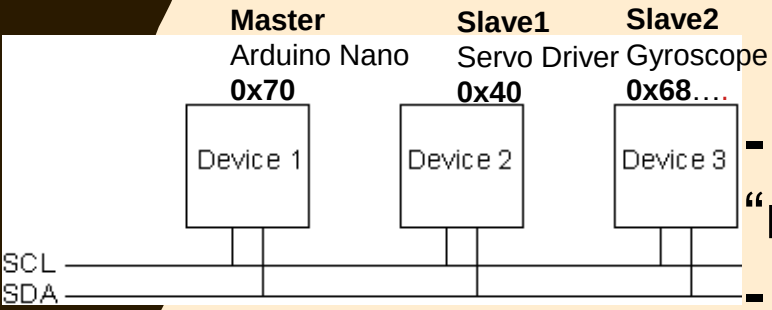
minicat_IRremote_test §

```
}

void translateIR() {//fonction pour associer une action à chaque bouton

  switch(result.value){ //switch permet de lister tous les cas possibles et
    //de leur associer une action

    //on va donc lister tous les codes et retourner leur nom sur la télécommande

    //AV: FOR MINICAT1, "Carmp3" RemoteControl
    //AV: almost all original MiniCat RemoteControl codes are working except for the "Carm

    //AV: original MiniCat SW :
    case 0xFF18E7: Serial.println(" 2 (grey1) or 2 (grey2) or UP (black)"); break;
    //case 0x3D9AE3F7: Serial.println(" HAUT"); break;

    case 0xFF10EF: Serial.println(" 4 (grey1) or 4 (grey2) or LEFT (black) or Mode (grey2)
    //case 0x8C22657B: Serial.println(" GAUCHE");    break;

    case 0xFF38C7: Serial.println(" 5 (grey1) or 5 (grey2) or OK (black)");    break;
    //case 0x488F3CBB: Serial.println(" -OK-");    break;

    case 0xFF5AA5: Serial.println(" 6 (grey1) or 6 (grey2) or RIGHT (black)");   break;
    //case 0x449E79F: Serial.println(" DROITE");   break;

    case 0xFF4AB5: Serial.println(" 8 (grey1) or 8 (grey2) or DOWN (black)"); break;
    //case 0x1BC0157B: Serial.println(" BAS"); break;

    case 0xFFA25D: Serial.println(" CH- (grey1) or ON (grey2) or 1 (black)");    break;
    //case 0xE318261B: Serial.println(" 1");    break;

    case 0xFF629D: Serial.println(" CH (grey1) or Mode (grey2) or 2 (black)");    break;
    //case 0x511DBB: Serial.println(" 2");    break;

    case 0xFFE21D: Serial.println(" CH+ (grey1) or LoudspeakerOFF (grey2) or 3 (black)");
    //case 0xEE886D7F: Serial.println(" 3");    break;
```

Hardware Modules & Test

Module 4: I2C Bus connections





Module 4: I2C Bus connections / devices check

- comm. between Arduino & Servomotors & Gyro boards

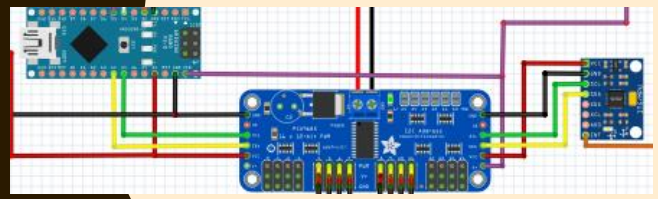
ToDo:

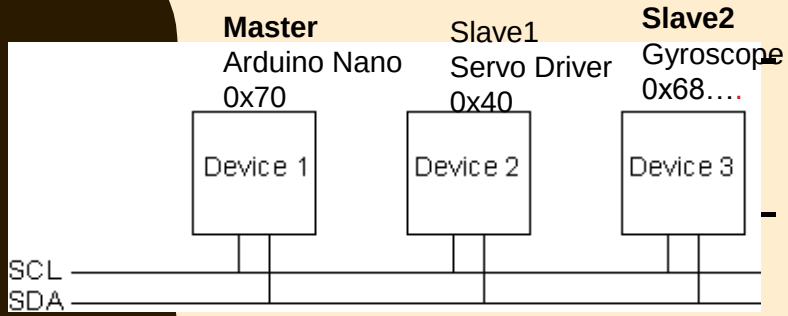
- Unplug the USB cable from your Laptop
- Plugin the Gyro and Servodrivers boards on PCB
- Plugin the USB cable to your Laptop
- Put the 2 batteries in place and turn ON on holder (MPU6050 need enough power from Servodrivers board)
- I2C library (“Wire”) is installed by default (Arduino IDE)

- open the i2c communication test software:
“minicat-main/Software/minicat_i2c_test/minicat_i2c_test.ino”
- 3x I2C devices should appear (serial monitor)

Hardware Modules & Test

Module 4: I2C Bus connections





ToDo:

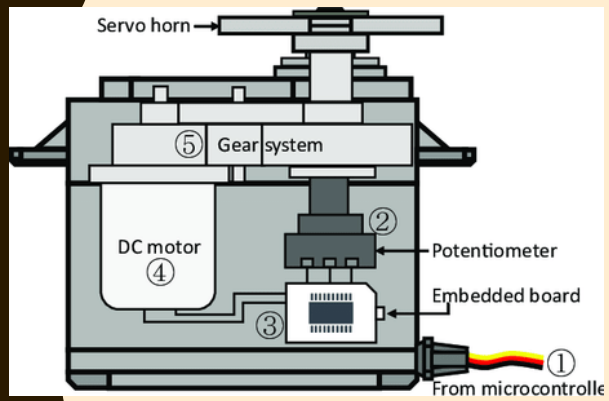
- Open the Arduino **Serial Monitor**
  (Arduino IDE “Tools/Serial Monitor”)

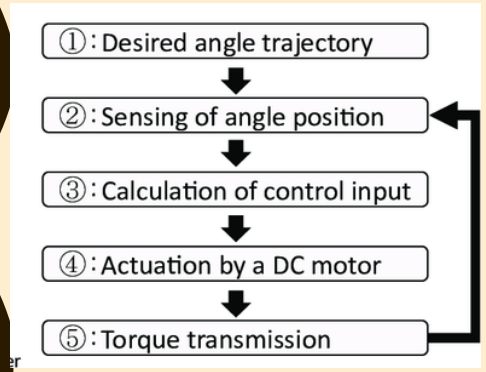
- Set the communication speed to **57600 baud**

- **Verify that the 3 I2C Devices are present:**
  - I2C device found at address 0x40 ! (Servos Driver)
  - I2C device found at address 0x68 ! (Gyro)
  - I2C device found at address 0x70 ! (Arduino Nano)

Hardware Modules & Test

Module 5: Servomotors driver board





Module 5: Servomotors driver board

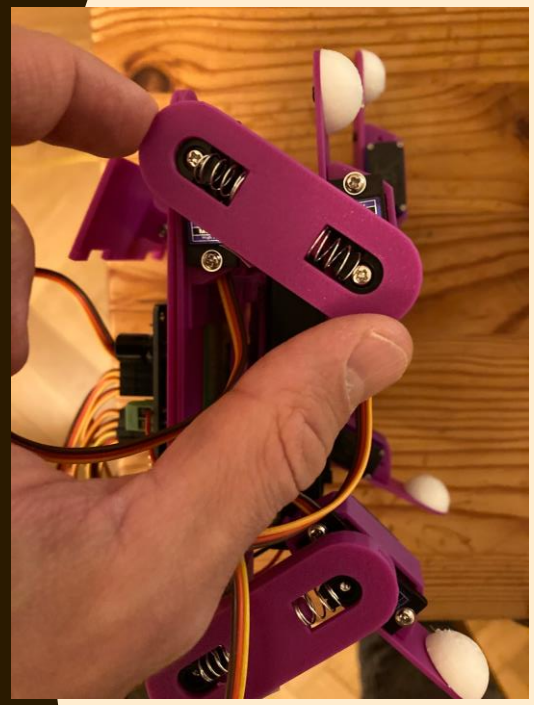
- drives the 8 servos, here we **test w/ 1 servomotor**
- based on Arduino inputs on I2C
- and Servodriver board 8 x PWM control signals outputs
- **we use first 5V supply from Arduino (no batteries)**

ToDo:

- Unplug the USB cable from your Laptop
- The 8x Servomotors cables are already plugged

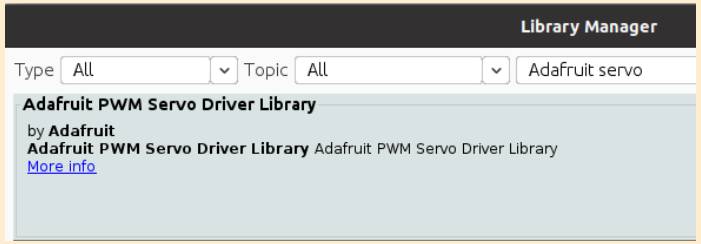
Hardware Modules & Test

Module 5: Servomotors driver board



ToDo:

- Plugin the USB cable to your Laptop
- Start the Arduino IDE

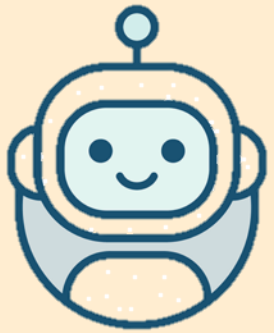


- the Software Library
"Adafruit PWM servo driver library"
should be installed
(Tools/Manage Libraries "Adafruit PWM Servo Driver Library")

- open the minicat_one_servo_test software : double click on
"minicat-main/Software/minicat_one_servo_test/minicat_one_servo_test.ino"

Hardware Modules & Test

Module 5: Servomotors driver board



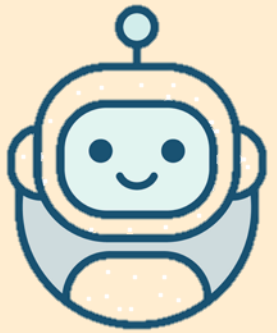
ToDo (Software):

change servo n.
and delay :

```
minicat_one_servo_test §
  //pwm.setPWM(SERVO, 0, SERVOMIN+range/2);//l'impulsion est large de SERVOMIN + range
                                          //=>position milieu du servomoteur
  //delay(2000);

  //AV: Commenting out below lines and changing the rotation angle to avoid too much r

  //Contrôle du servomoteur par angle (voir fonction setToAngle)
  //Serial.println("position 45 degrés ");
  //setToAngle(45);

  //Serial.println("position 20 degrés");
  //setToAngle(20);


  //AV : ToDo : MODIFY here to test the Servo Angle position,

  /*************************************************************/

  //AV: ToDo : MODIFY THE 1st POSITION OF THE SERVO :

  Serial.println("position 10 degrés");
  setToAngle(10);



  //AV: ToDo : MODIFY THE DELAY (in Miliseconds) BETWEEN 2 POSITIONS AS YOU WANT :

  delay(1000);


  //AV: ToDo : MODIFY THE 2d POSITION OF THE SERVO :

  Serial.println("position 30 degrés");
  setToAngle(30);

 /*************************************************************/

}
```

Hardware Modules & Test

Module 5: Servomotors driver board



ToDo (Software):

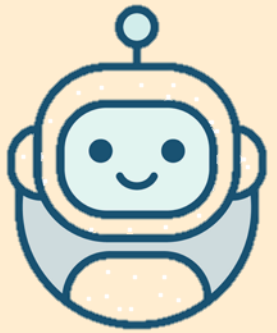
Change the servo number to test all servos one by one :

minicat_one_servo_test §

```
//on inclut les librairies nécéssaires


#include <Wire.h>
#include <Adafruit_PWMServoDriver.h> //librairie spécifique à notre servo driver

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(); //déclaration du driver pwm

/*############################################################*/

//AV: ToDo : Modify the Servo number

#define SERVO 4 //pin où est branché le servomoteur // AV : ToDo: Modify here to test all the Servos (4 to 11)

/*############################################################*/
```

Hardware Modules & Test

Module 5**bis**: Servomotors driver board



Module 5bis: Servomotors driver board

- we use now the 2x4,2V Lithium batteries supply

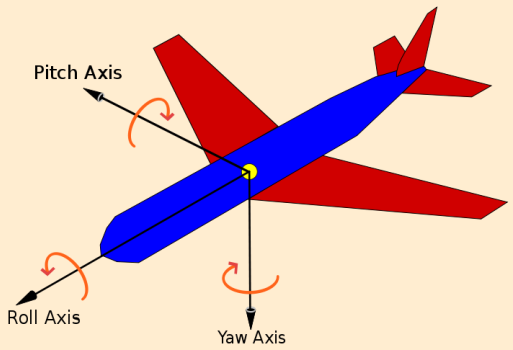
(your teacher will install them, please ask him)

ToDo:

- Unplug the USB cable from your Laptop
- Ask your teacher to install the 2 x Batteries in the robot
- Put the batteries holder button to “ON” under the robot
- Check if the Servomotor is moving (Front Right Top)

Hardware Modules & Test

Module 6: Gyroscope board



Module 6: Gyroscope board

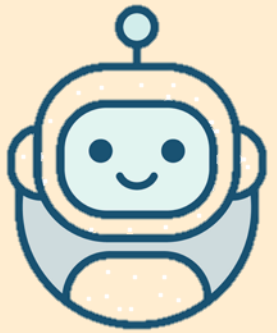
- measuring roll/pitch/yaw
- compared with calibrated “zero”
- compensate the legs height / balance

ToDo:

- Unplug the USB cable from your Laptop
- Plug the Gyro module in the PCB
- Plugin the USB cable to your Laptop
- Start the Arduino IDE’
- Get the Arduino Libraries “I2Cdev and MPU6050”
- Start the 1x Gyro test software “minicat_i2c_test.ino”

Hardware Modules & Test

Module 6: Gyroscope board



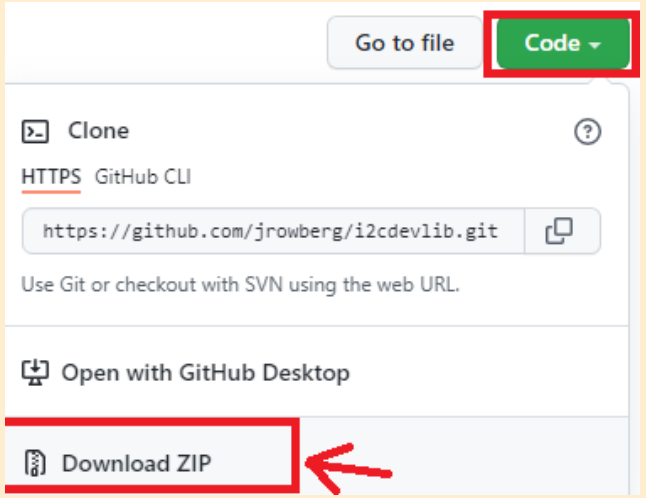
The Gyroscope module need the Libraries “**I2Cdev**” and “**MPU6050**” for I2C communication to Gyroscope MPU6050 sensor :

– Open : https://github.com/jrowberg/i2cdevlib

– Click on “Code” and then “**Download ZIP**”

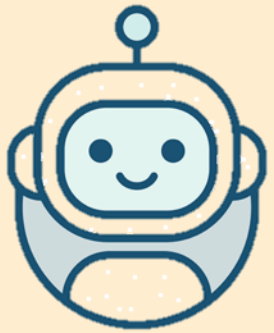
– the Downloaded .zip is normally stored into your “C:\Users*UserName*\Downloads” folder

– **Unzip** the “i2cdevlib-master.zip” in your “C:\Users*UserName*\Downloads” folder to get a “i2cdevlib-master” folder

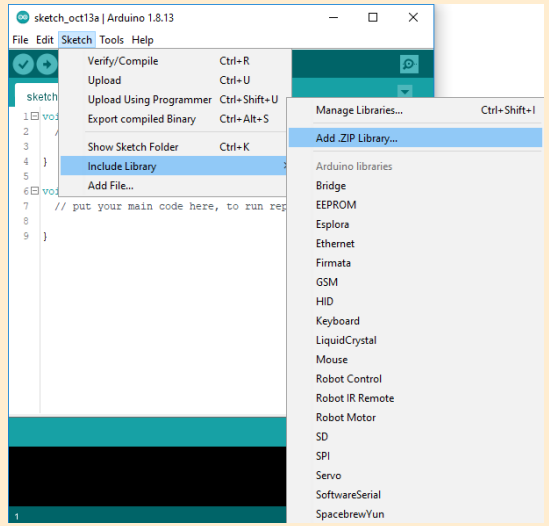


Hardware Modules & Test

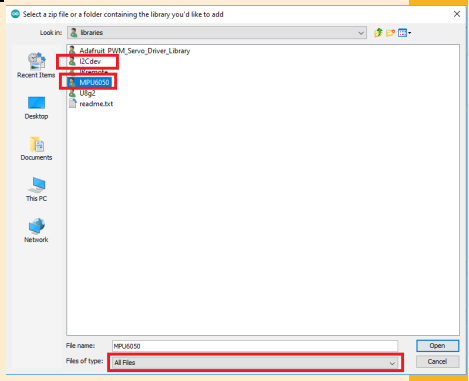
Module 6: Gyroscope board



1- Now to add the 2 libs we need,
Click on “**Sketch/Include Library/Add .ZIP Library**”

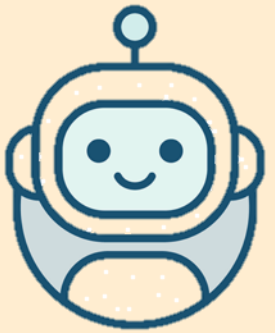


2- Change the “Files of type” to “**All Files**”
(as we already unzipped) :
3- Navigate to
“C:\Users*UserName*\Documents\Arduino\libraries\i2cdevlib-master\i2cdevlib-master**Arduino**\

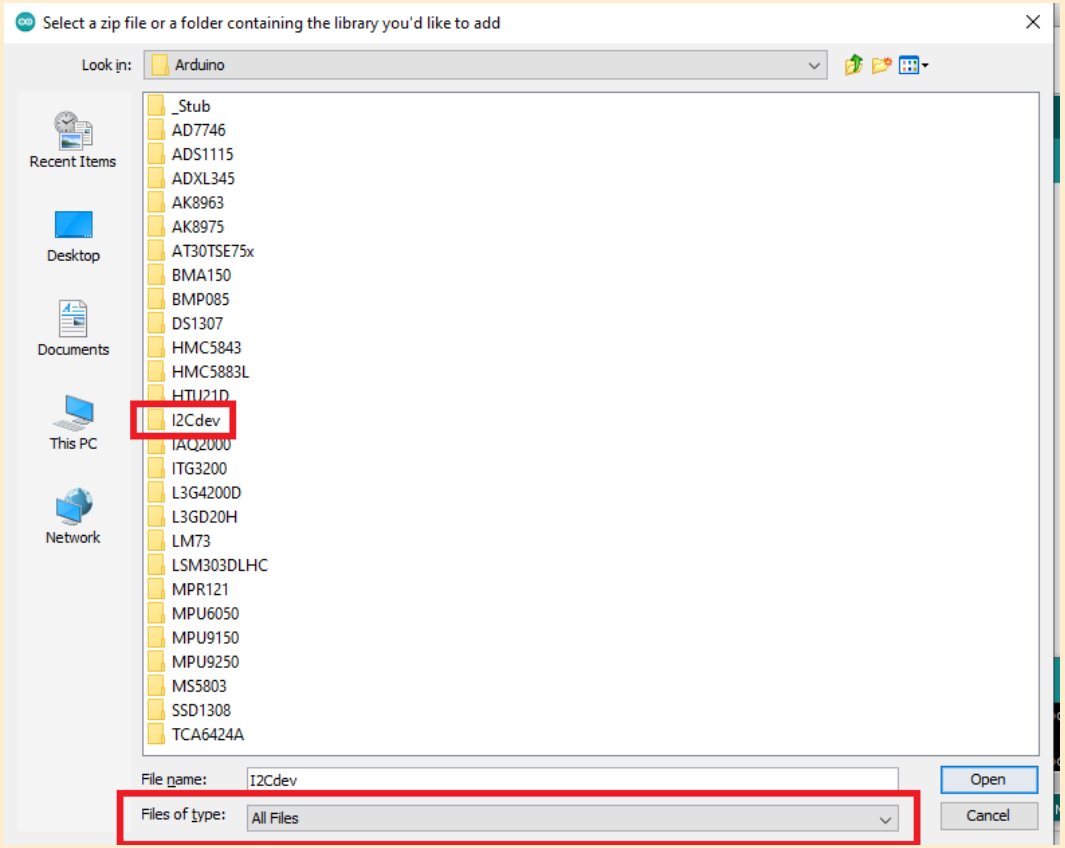


Hardware Modules & Test

Module 6: Gyroscope board

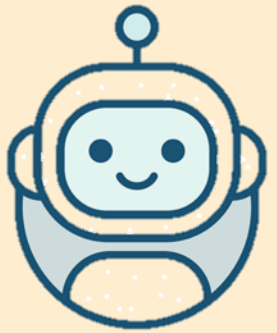


4- Then Single Click on the “**I2Cdev**” Folder and then “**Open**” :

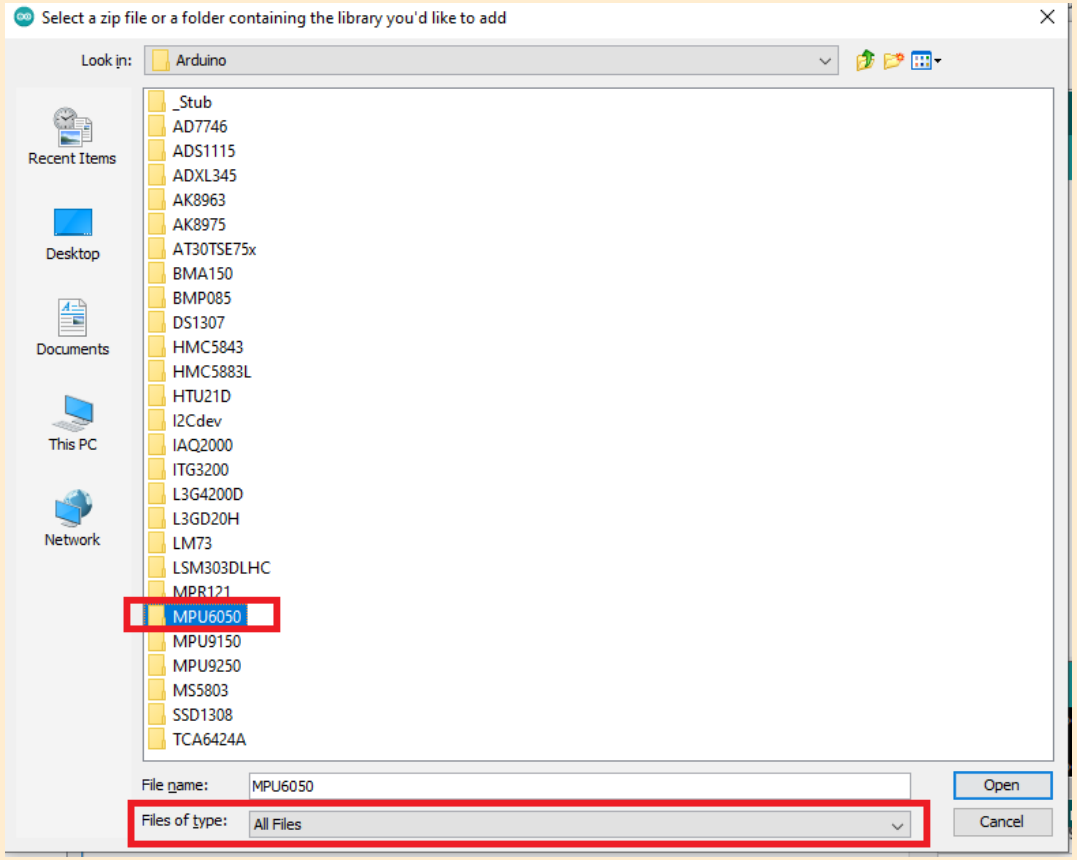


Hardware Modules & Test

Module 6: Gyroscope board

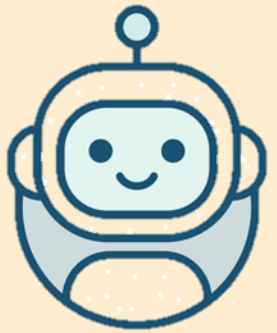


5- **Repeat** the steps above for adding the "**MPU6050**" lib :



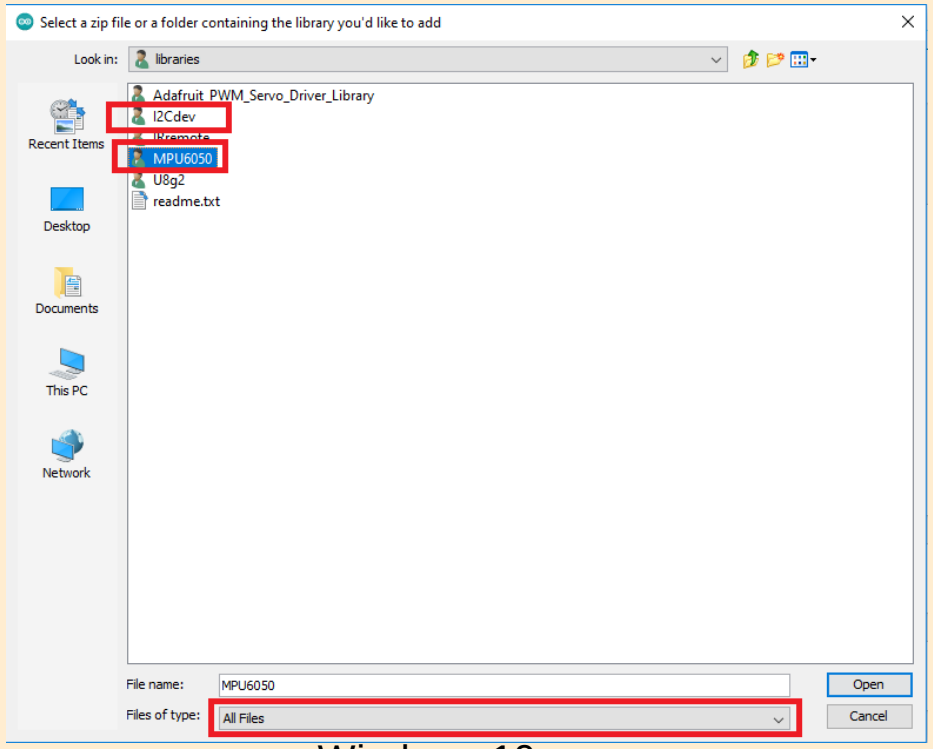
Hardware Modules & Test

Module 6: Gyroscope board

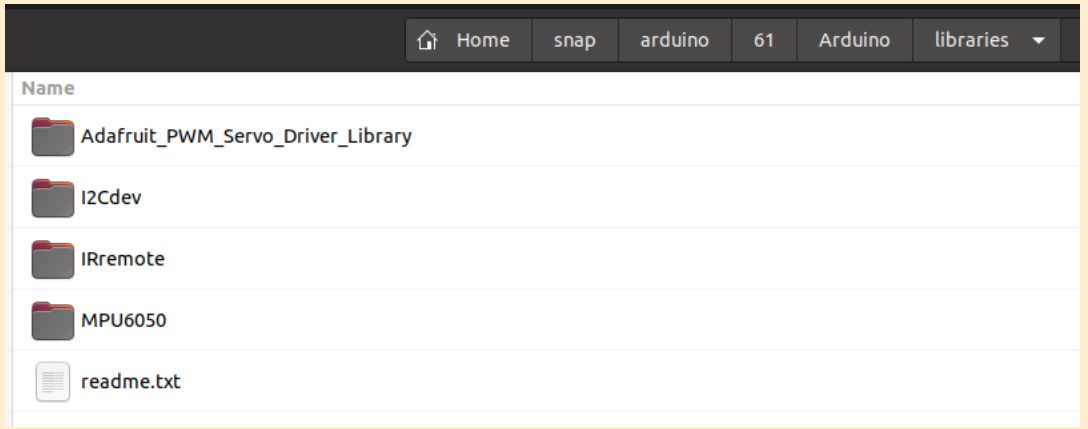


6- RESULT:

The 2 Libs should now appear in your “C:\Users*UserName*\Documents**Arduino\libraries** folder:



Windows 10



Ubuntu 20.04

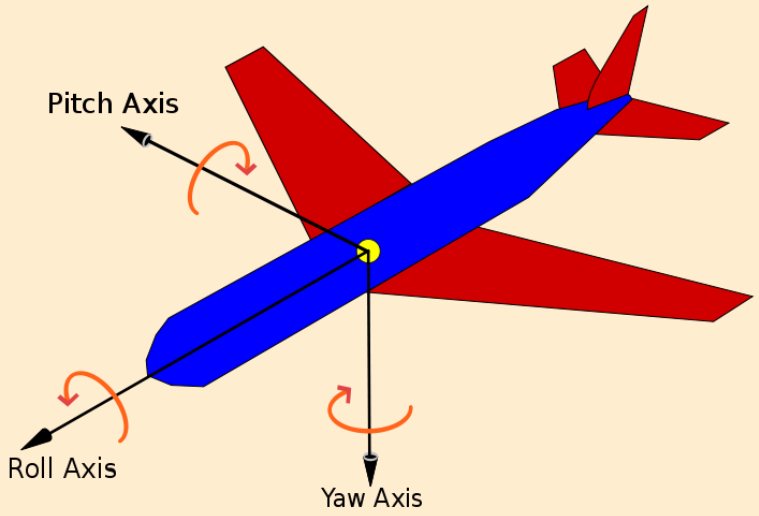
Hardware Modules & Test

Module 6: Gyroscope board

ToDo:

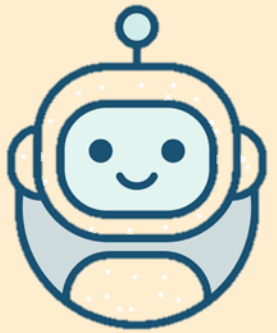
- Open the Serial Monitor

- Check the “**ypr**” = **y**aw **p**itch **r**oll values when moving MiniCat :
  (click on “Autoscroll” to stop)



Hardware Modules & Test

Module 7a: All modules + calib



Module 7a: All Modules + calibration

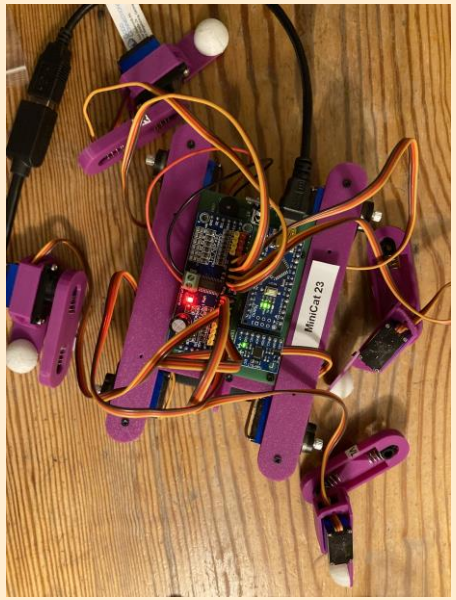
- **servos check + gyro calibration**
- **not in focus of this workshop (time consuming)**
- servo need to be placed at their zero (middle) angle
- gyro need to be calibrated to know its zero angles (rpy)

Todo:

- Make sure all modules plugged on the PCB
- **Put the battery holder button on “ON” first**
- **Plugin the USB cable to your Laptop**

Hardware Modules & Test

Module 7a: All modules + calib



Todo (Software): calib done, not in focus of workshop

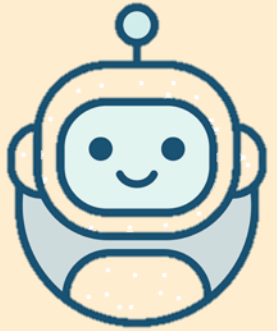
- disconnect all legs of MiniCat, no servo blocked

- make sure MiniCat is on a flat surface

- start the "**minicat_servos_and_gyro_calibration.ino**"
  - Open the **Serial Monitor**
  - "**Upload**" the Software
  - The Gyroscope will measure the MiniCat chassis roll/pitch/yaw angles offsets compared to the MPU6050 chip default zero angles, the **offsets will be shown ´' in the serial monitor**

Hardware Modules & Test

Module 7a: All modules + calib



Background :

servos calib:
- we need to put all servos at their central zero position before mounting the legs of MiniCat

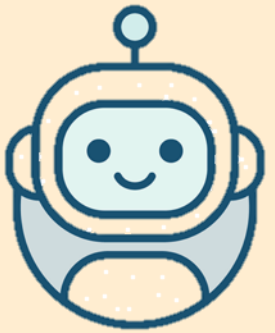
gyroscope (MPU6050 chip) calib:
- we need to measure the offset of the gyro yaw pitch roll angles compared to the default zero angles in the Gyro board firmware (MiniCat should be put on a flat surface)

Remark :

the gyro angles offset values will be stored in EEPROM
(non volatile memory of Arduino Nano)
so that this Calibration is needed only once on a particular Arduino

Hardware Modules & Test

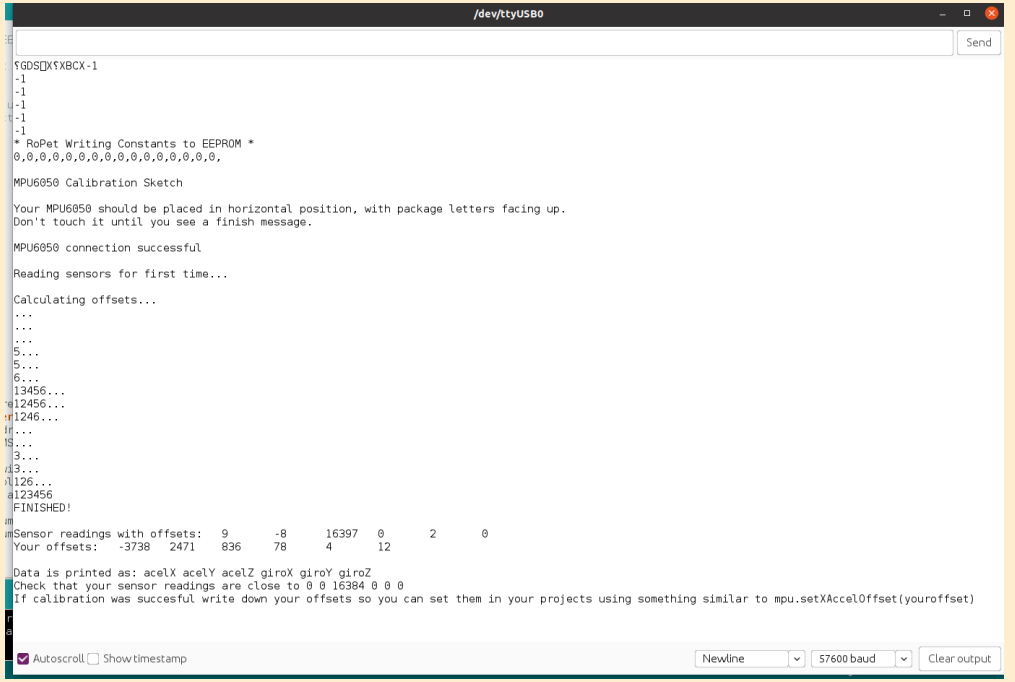
Module 7a: All modules + calib



Todo (Software):

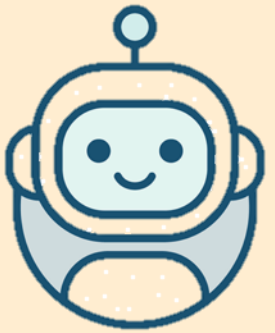
- the gyro and servos calibration process will start

- if everything is fine, you get this output in the serial monitor



Hardware Modules & Test

Module 7a: All modules + calib



Todo (Software):

- keep the serial monitor open and
type "c" in the command line then "**Enter**"
This will place the servos at their central "zero" position

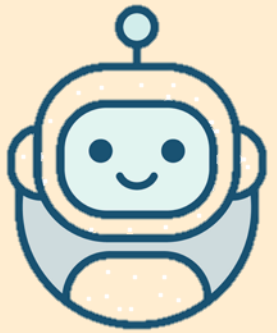
- The Servo angles will be shown w/ zeros in the serial monitor (only 8 to 15 are our 8 servos)

8,	**9,**	**10,**	**11,**	**12,**	**13,**	**14,**	**15**
0,	0,	0,	0,	0,	0,	0,	0,

- Keep the battery holder button on "ON"
- Do not remove the USB cable from your Laptop yet !
- you will only insert the tighs & feet parts in the servos axis
see how on next slide, do not turn the servo axis manually !
(as you just set the servos to their zero angle pos.)

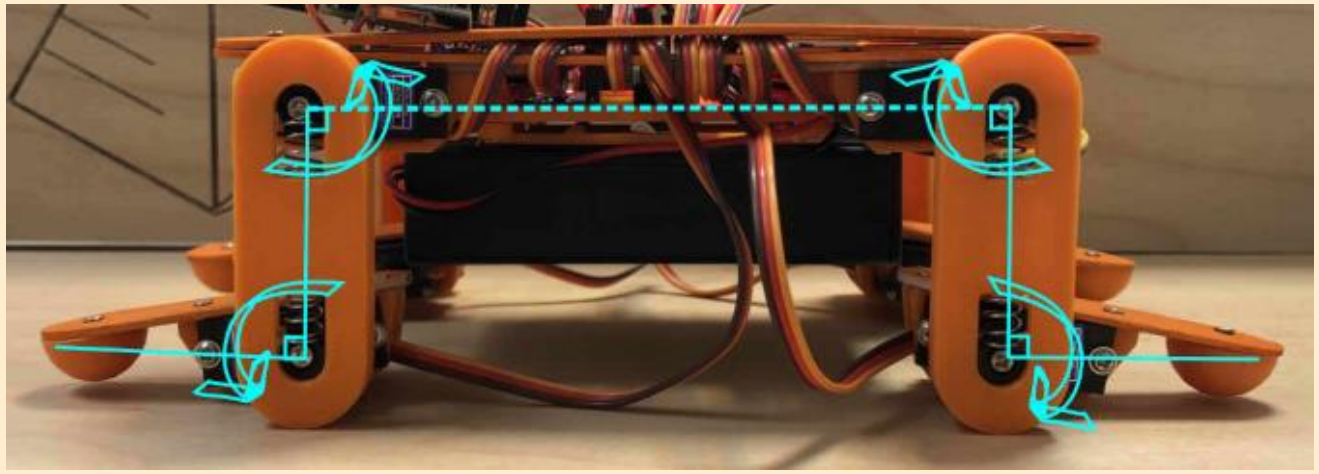
Hardware Modules & Test

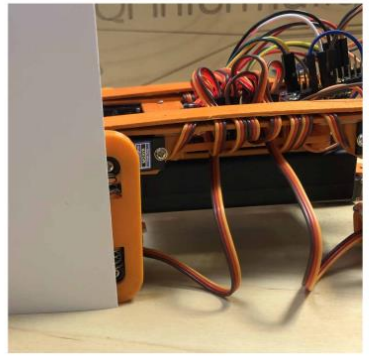
Module 7a: All modules + calib

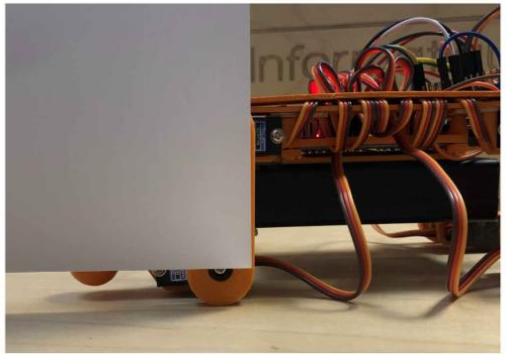


Todo (Software):

- Now that you have set the servos to their zero angle you can mount them as below
(Hint: **do not move the servo axis anymore!**
If so, start again
“minicat_servos_and_gyro_calibration.ino”) :

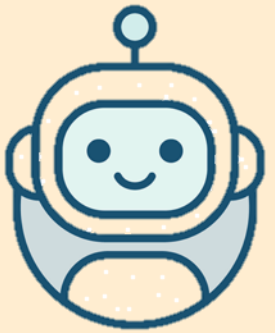






Hardware Modules & Test

Module 7a: All modules + calib



Optional / Todo (Software): Servos angle Fine tuning :

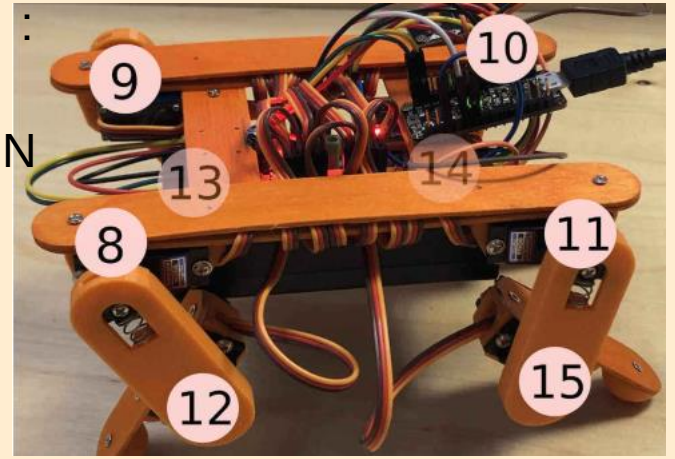
- If you can not achieve 100% the above, you can still **fine tune the servo angles by software by running the below in the Serial Monitor:**

c "servo link index" "correction angle

For example, if you want to change the angle of link 12 of +5 deg you can run : ***c12 5***

Servos indexes reminder :

FRONT



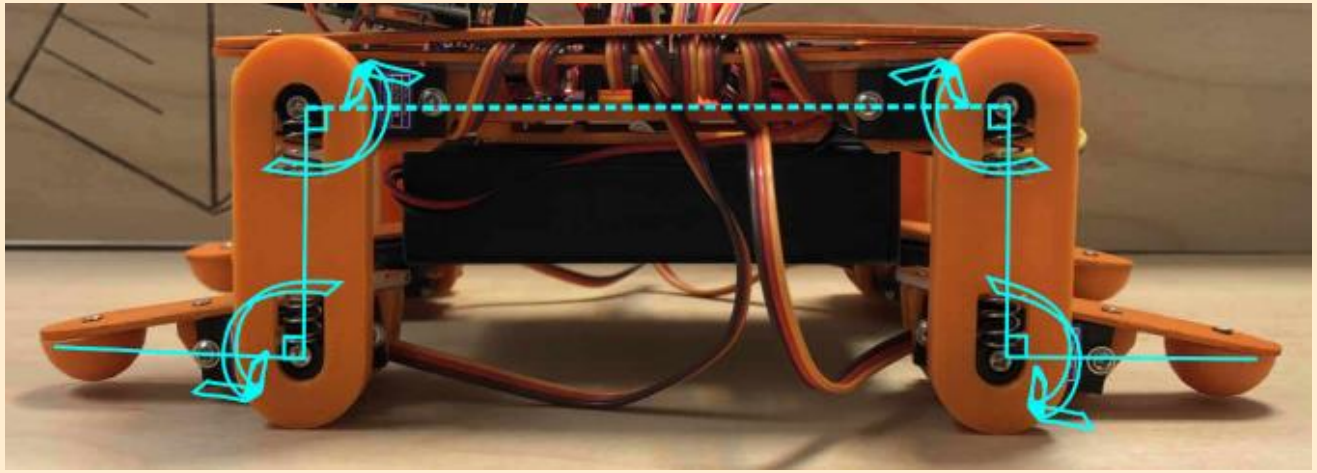
Remark : each tooth of the servo axis is 360deg/20teeth so 18deg so the fine tuning makes sense for +/- 9deg, if more needed just remove the servo arm from the servo axis and place it on a next tooth!

Hardware Modules & Test

Module 7a: All modules + calib

Todo (Software):

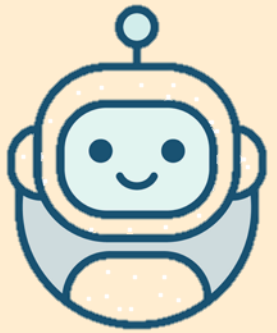
- put the servo screws in the servo axis to fix the positions :



- Once you have achieved the above, **type “s”** in the serial monitor (this will save your servos angles calibration in Arduino EEPROM (non volatile))

Hardware Modules & Test

Module 7a: All modules + calib



Todo (Software):

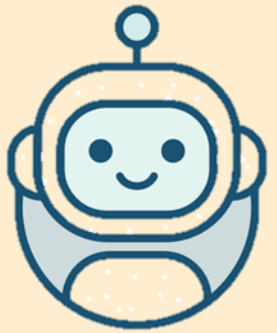
- to test your servos calibration, try different positions :

- type “**d**” in monitor to go in default position (“stop”), expected :

- type “**pbalance**” to go in “stand” position, expected :

Hardware Modules & Test

Module 7b: All modules on PCB



Module 7b: All Modules on PCB and Play !

ToDo:

- Unplug the USB cable from your Laptop
- Make sure all modules plugged on the PCB
- **Put the battery holder button on “ON” first**
- **Plugin the USB cable to your Laptop**

- **open the minicat_basic software :** double click on “minicat-main/Software/minicat_basic/minicat_basic.ino

- Use the remote control to play !