Building Robots

from Mechatronic Components to Robotics

System integration

Tutorials

by

Erasmus+ project

GEMS - Graceful Equalising of Mechatronics Students

0 Access to information

Question 1: How to get information about a mechatronic components?

About: Mechatronic systems combine mechanical, electrical, and software elements, and accurate information is essential for design, integration, and safe operation. Official documentation, such as standards, datasheets, and verified repositories, provides precise specifications and usage instructions. These resources are also useful for finding example applications, recommended operating conditions, or code snippets for practical implementation.

When designing parts for production, it is important to consult manufacturing service documentation. For example, PCB manufacturers provide design rules and capabilities (minimum trace width, hole sizes, layer stackup, copper thickness) that must be followed to ensure manufacturability. Similarly, for 3D-printed parts, check the build plate size of the target printer.

Economic factors should also be considered when selecting components or services: the price, delivery cost, and customs duty or taxes can affect overall project feasibility. The cost of a part often depends on the quantity purchased (economy of scale). Sometimes parts are not available or are approaching end-of-life (EOL), so it is important to check availability and identify suitable substitutes if necessary.

To find official information, use search engines or reputable online component distributors (e.g., Digi-Key, Mouser, RS Components) to locate datasheets, standards, library references, or manufacturing guidelines. While AI tools or community forums can offer guidance, their information may be incomplete or unreliable.

Mechanical Components include parts like bolts, gears, shafts, and bearings. Reliable information comes from official standards (e.g., ISO, DIN, ANSI) or manufacturer datasheets.

Example: DIN 912 M3 Bolt

Use a search engine or supplier website to search for “DIN 912 M3 datasheet” or “DIN 912 M3 dimensions.” Find a document that provides information about head dimensions and length options.

Electrical Components include resistors, capacitors, transistors, sensors, motors, regulators, and controllers. Manufacturer datasheets provide voltage, current, pinout, thermal limits, and wiring diagrams.

Example: R1170H3318 3.3V LDO Regulator

Search using a search engine or an online distributor (e.g., Digi-Key, Mouser) for “R1170H3318 datasheet.” Check input/output voltage ranges, maximum current, dropout voltage, and recommended operating conditions. Also find the application circuit.

Software Components include functions, classes, data types, firmware, drivers, and libraries needed to control mechatronic systems. Official documentation and repositories (e.g., GitHub, vendor-maintained packages) provide verified code and usage examples.

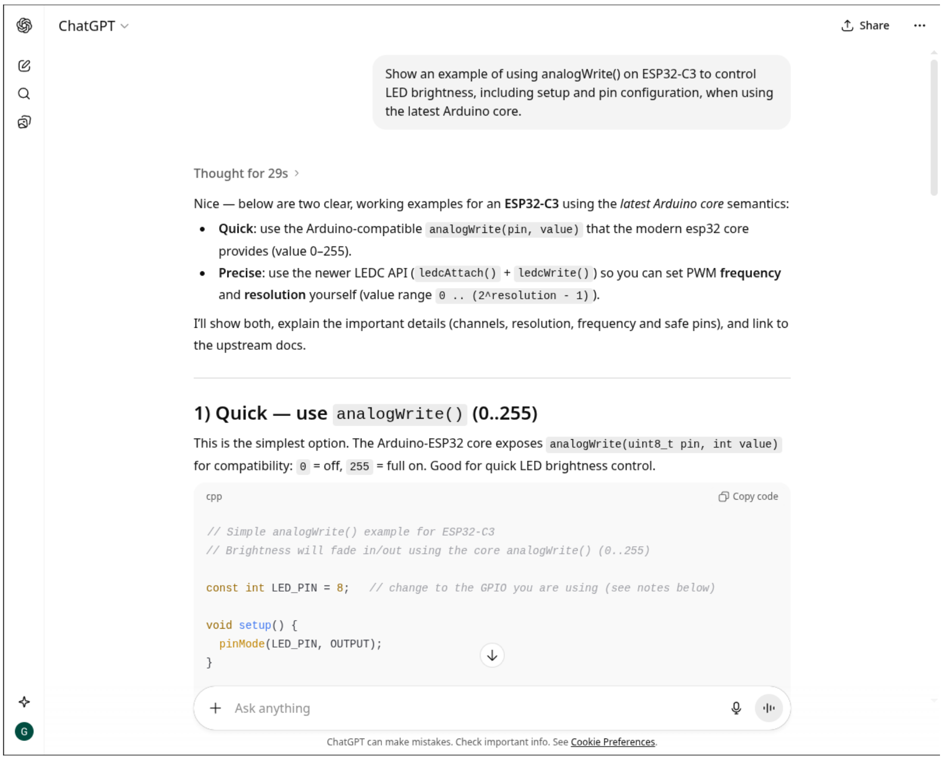
Example: PWM on ESP32-C3 Microcontroller

While official documentation provides references, tutorials or examples can be easier for practical implementation. Use a search engine to find trusted tutorials, e.g., search “ESP32-C3 PWM tutorial” or check Espressif example repositories on GitHub. Find an example program for the ESP32-C3 that sets PWM parameters (duty cycle and frequency) for a specific pin.

Question 2: How to use LLMs for Mechatronic system development?

Large Language Models (LLMs) are Artificial Intelligence (AI) tools that can provide guidance, explanations, and suggestions for mechatronic systems, programming, and CAD/EDA software. They can help with understanding concepts, finding documentation, or generating code snippets for tools like FreeCAD, KiCAD, or the Arduino IDE.

However, AI results can be incomplete or inaccurate, so it is important to cross-check outputs with official documentation, standards, datasheets, or verified tutorials. AI outputs should be used as guidance, not as a primary source of facts. Additionally, LLMs perform best when the input is clear and detailed enough for the model to understand the concept. Always include sufficient context and specifics in your query (e.g., model numbers, software version, desired outcome). Queries can be refined by providing additional prompts with more relevant details. Code generated by LLMs should never be copied blindly—it should be read, understood, and verified before use.



Examples:

- Mechanical Concept Explanation
  - Input: “Explain the difference between Bowden and direct drive extruders in FDM 3D printers, including advantages and limitations.”
- Programming Tool Assistance (Arduino IDE)
  - Input: “Show an example of using analogWrite() on ESP32-C3 to control LED brightness, including setup and pin configuration, when using the latest Arduino core.”
- CAD Software Workflow (FreeCAD)
  - Input: “Provide step-by-step instructions to create a spur gear using the FreeCAD Gear Workbench, with default settings and metric units.”
- PCB Design Assistance (KiCAD)

- Input: “Explain how to select track width in KiCAD for a signal and how to calculate which width to use.”

1 KiCAD

Question: How to create a PCB?

About: KiCAD is an open-source suite for electronic design automation (EDA) used to create schematics, printed circuit board (PCB) layouts, 3D visualizations, and fabrication files. KiCAD offers a fully featured environment that is free to use and continuously developed by its community, making it an accessible professional-grade option for PCB development.

Link: https://www.kicad.org/

Installation: KiCAD is available for Windows, Linux, and macOS. Install latest stable version from the official source (https://www.kicad.org/download/). Select your operating system, follow instructions and use recommended settings.

Custom parts library: KiCAD uses two types of libraries:

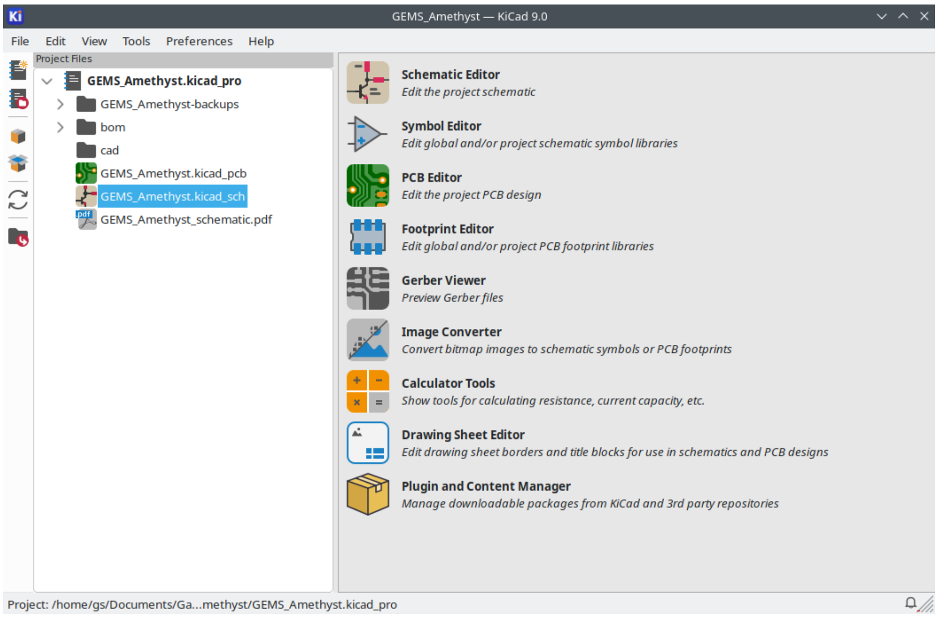
- Symbol libraries → schematic symbols (.kicad_sym file)
- Footprint libraries → PCB footprints (.pretty folder with .kicad_mod files and 3D_models folder with .step files)

To add the custom library start the program and:

1. Preferences → Configure Paths... → add (+) library **Name** and library **Path**
2. Preferences → Manage Symbol Libraries... → add (+) library **Nickname**, **Library Path**
3. Preferences → Manage Footprint Libraries... → add (+) library **Nickname**, **Library Path**

Example:

- Name: GEMS
- Path: /home/…/GEMS_Library
- Nickname: GEMS_Library
- Library Path: $GEMS/GEMS_Library.kicad_sym or $GEMS/GEMS_Library.pretty

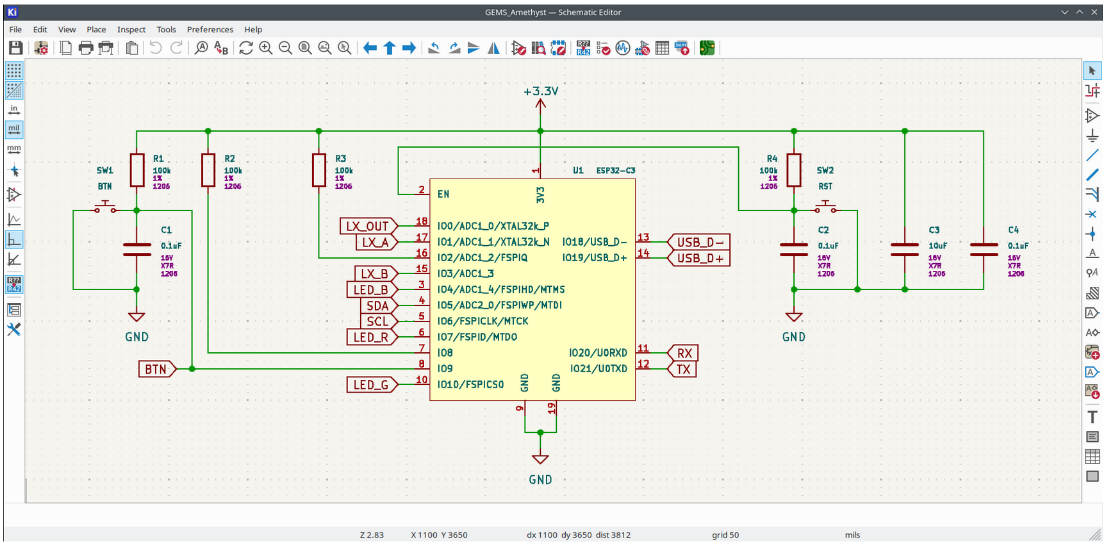


Create project

1. Start a New Project (File → New Project)

Create schematic

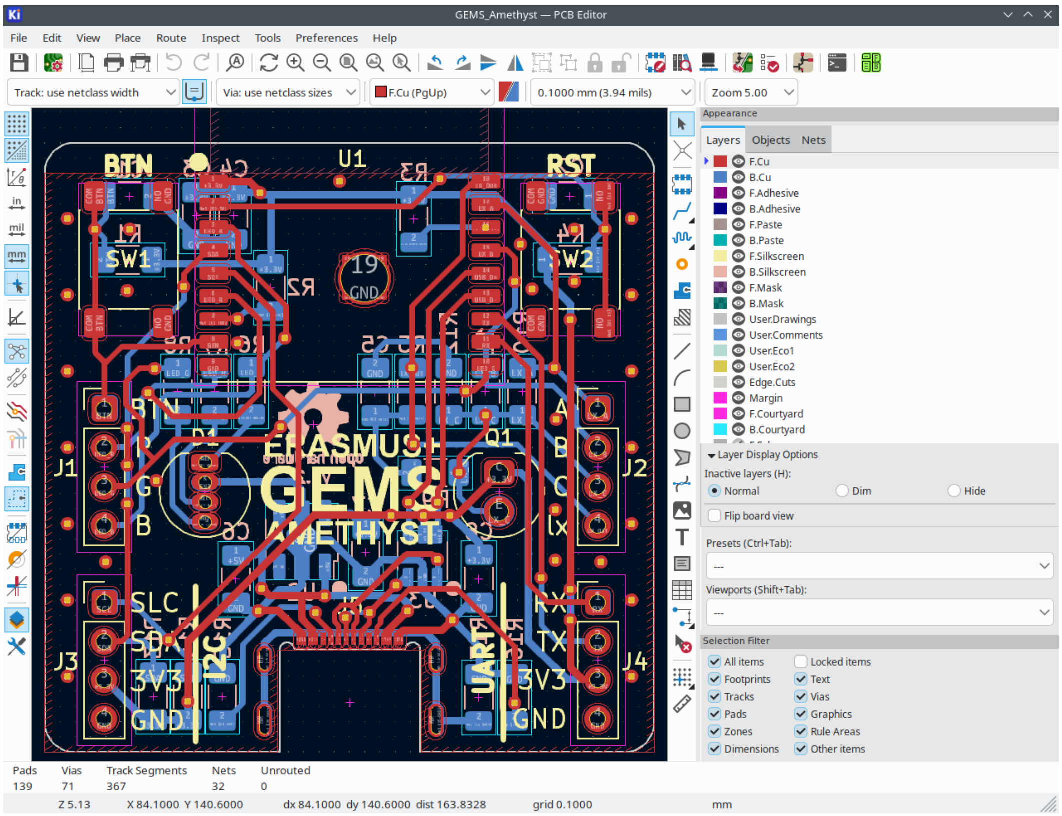
1. Open the Schematic Editor
   - In Project Manager → click Schematic Editor.
2. Add Components (Symbols)
   - Right toolbar → Place Symbols (hotkey A) → search and place parts on canvas.
3. Add Power and Ground Symbols
   - Right toolbar → Place Power Symbols → choose VCC, GND, or other rails.
4. Wire the Components
   - Right toolbar → Draw Wires (hotkey W) → click between pins to connect.
   - Use Place Net Labels (hotkey L) for cleaner connections.
5. Annotate the Schematic
   - Top toolbar → Annotate Schematic → assigns unique references (R1, C1, U1).
6. Assign Footprints
   - Top toolbar → Assign Footprints → link each symbol to a PCB footprint if not already assigned.
7. Run Electrical Rules Check (ERC)
   - Top toolbar → Perform Electrical Rules Check → review and fix warnings.



Create PCB

1. Open the PCB Editor
   - In the Schematic Editor → top toolbar → Update PCB from Schematic (green arrow PCB icon).

2. Define the Board Outline
   - Left toolbar → Add Graphic Line/Polygon on the Edge.Cuts layer → draw the board shape.
3. Arrange Footprints
   - Select footprints → press M (move), R (rotate), F (flip) → place inside outline.
4. Set Design Rules
   - Top menu → File → Board Setup → configure track width, via size, and clearance.
5. Route Traces
   - Right toolbar → Route Tracks (hotkey X) → follow ratsnest lines between pads.
6. Use V to add vias when routing multi-layer boards.
7. Add Copper Fills (Planes)
   - Right toolbar → Add Filled Zone (Shift+Ctrl+Z) → assign to GND net → draw over board.
   - Update Fill Zones → press B.
8. Run Design Rules Check (DRC)
   - Top toolbar → Inspect → Design Rules Checker → fix clearance and unconnected net errors.
9. Check in 3D Viewer
   - Top toolbar → View → 3D Viewer (hotkey Alt+3) → inspect layout and component fit.



Create Manufacturing files

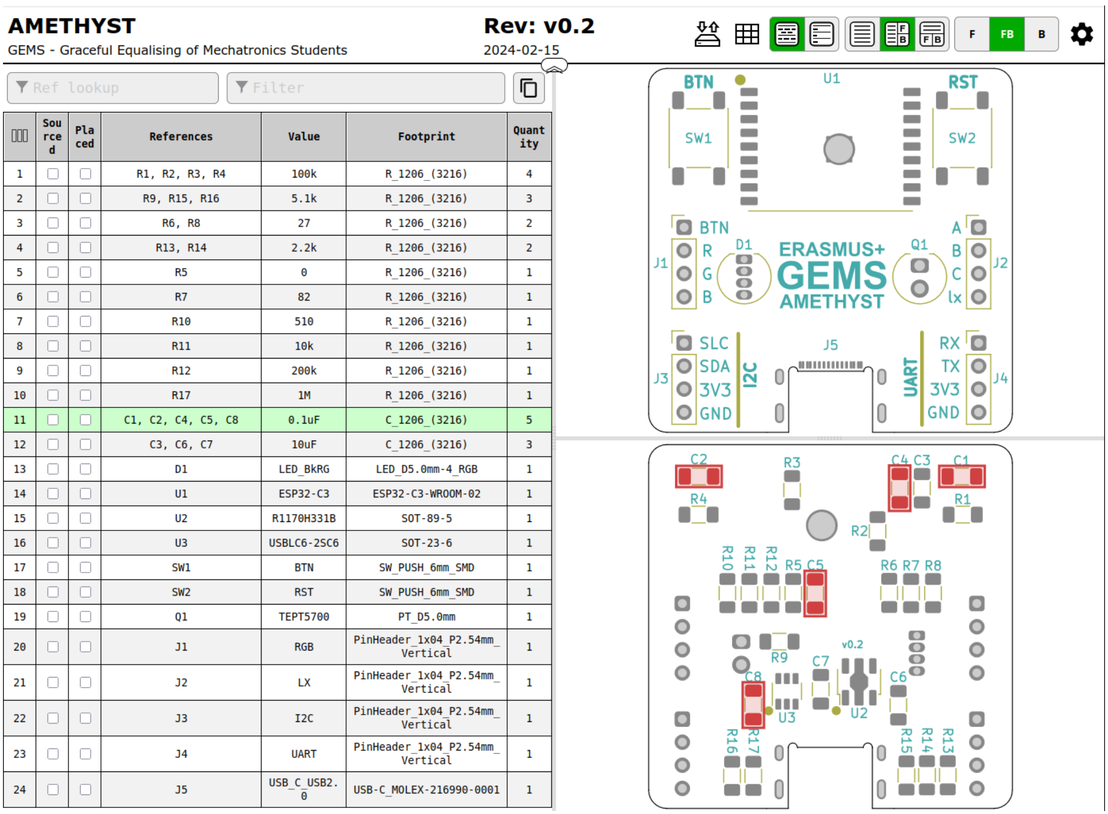
1. Open the PCB Editor
2. Check the Design
   - Top toolbar → Inspect → Design Rules Checker (DRC) → fix any errors.
   - 3D Viewer (Alt+3) to verify component placement.
3. Open the Plot Dialog
   - Top menu → File → Plot.
4. Select Gerber Output
   - Plot format → Gerber.
   - Layers → select all required layers:
     - F.Cu / B.Cu (copper layers)
     - F.Silkscreen / B.Silkscreen (silkscreen layers)
     - F.Mask / B.Mask (solder mask layers)
     - Edge.Cuts (board outline)
5. Generate Gerber Files
   - Click Plot → KiCAD creates individual Gerber files for each layer.
6. Generate Drill Files
   - In the same Plot dialog → click Generate Drill Files.
   - Choose drill format (Excellon) and output directory.
7. Verify Gerber Files
   - Open the Gerber Viewer
   - Top menu → File → Open Gerber Job File
   - Top menu → File → Open Excellon Drill File

Manufacturing

1. Combine all Gerber Job and Excellon Drill Files in a compressed .zip file.
2. Send files to a manufacturer. If the manufacturer has a website application for order submission the files can be uploaded there.
3. Based on the uploaded files the website application will automatically fill out some of the properties for the order.
4. Check the automatically selected properties and change them according to your specifications (e.g. PCB thickness, PCB color, Silkscreen …).
5. Check the expected result with the Gerber Viewer.
6. Specify the number of PCBs you want to be manufactured.
7. Double check the PCB settings, estimated price, build time and shipping.
8. Submit the order.

Documentation for assembly

1. Install plugin **Interactive Html Bom** via Plugin And Content Manager.
2. Open the PCB Editor
3. Generate Bill of material (Bom)
    1. Top toolbar → Interactive Html Bom
    2. Specify Directory where output file will be saved
    3. Modify settings if needed
    4. Generate BOM
    5. Navigate to specified directory and open the .html file with a browser
    6. Interactive website with the PCB and the list of the used components will open.
4. Use the this website when soldering to identify where a certain component must be soldered.



	Sourced	Placed	References	Value	Footprint	Quantity
1	☐	☐	R1, R2, R3, R4	100k	R_1206_(3216)	4
2	☐	☐	R9, R15, R16	5.1k	R_1206_(3216)	3
3	☐	☐	R6, R8	27	R_1206_(3216)	2
4	☐	☐	R13, R14	2.2k	R_1206_(3216)	2
5	☐	☐	R5	0	R_1206_(3216)	1
6	☐	☐	R7	82	R_1206_(3216)	1
7	☐	☐	R10	510	R_1206_(3216)	1
8	☐	☐	R11	10k	R_1206_(3216)	1
9	☐	☐	R12	200k	R_1206_(3216)	1
10	☐	☐	R17	1M	R_1206_(3216)	1
11	☐	☐	C1, C2, C4, C5, C8	0.1uF	C_1206_(3216)	5
12	☐	☐	C3, C6, C7	10uF	C_1206_(3216)	3
13	☐	☐	D1	LED_BkRG	LED_D5.0mm-4_RGB	1
14	☐	☐	U1	ESP32-C3	ESP32-C3-WROOM-02	1
15	☐	☐	U2	R1170H331B	SOT-89-5	1
16	☐	☐	U3	USBLC6-2SC6	SOT-23-6	1
17	☐	☐	SW1	BTN	SW_PUSH_6mm_SMD	1
18	☐	☐	SW2	RST	SW_PUSH_6mm_SMD	1
19	☐	☐	Q1	TEPT5700	PT_D5.0mm	1
20	☐	☐	J1	RGB	PinHeader_1x04_P2.54mm_Vertical	1
21	☐	☐	J2	LX	PinHeader_1x04_P2.54mm_Vertical	1
22	☐	☐	J3	I2C	PinHeader_1x04_P2.54mm_Vertical	1
23	☐	☐	J4	UART	PinHeader_1x04_P2.54mm_Vertical	1
24	☐	☐	J5	USB_C_USB2.0	USB-C_MOLEX-216990-0001	1

2 Hand soldering

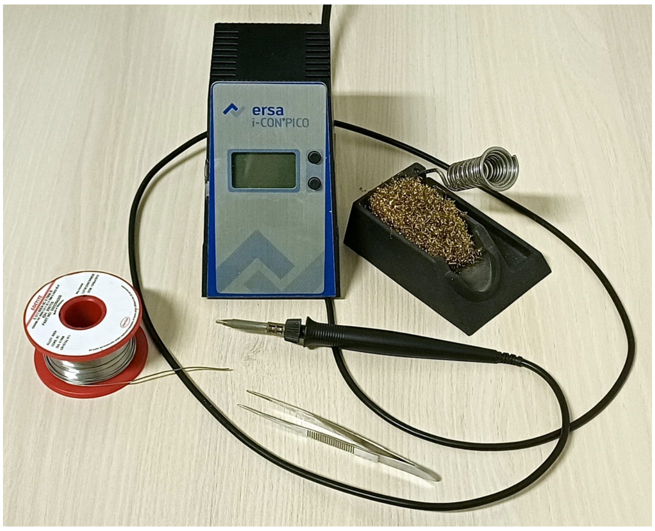
Question: How to hand solder electronic components on a PCB?

About: Hand soldering is the process of joining electronic components to a printed circuit board (PCB) using a soldering iron and solder. It is an important skill in electronics, used for prototyping, repairs, and small-scale production.

The process involves three key elements:

- **Heat** – supplied by a soldering iron to melt solder and form a reliable electrical and mechanical connection.
- **Solder** – a metal alloy (commonly tin/lead or lead-free alternatives) that melts and flows around the component lead and PCB pad.
- **Flux** – a chemical agent (often inside the solder or applied separately) that cleans the surfaces and helps the solder flow evenly.

Hand soldering requires practice, proper technique, and the right tools. A well-made solder joint ensures good electrical conductivity and mechanical stability of the component on the circuit.



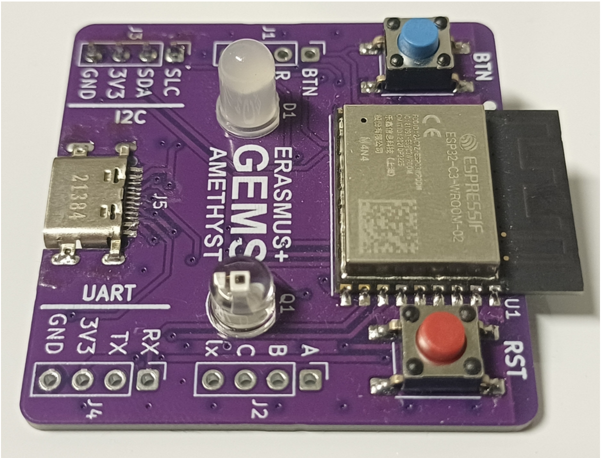
Essential tools

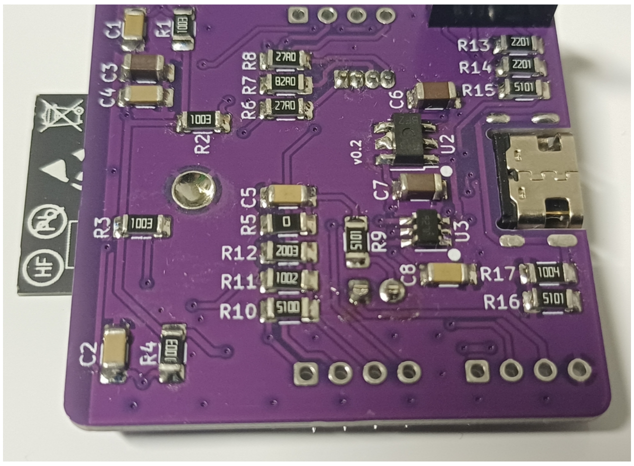
- Soldering iron
  - 15–60W
  - adjustable temperature is preferred
  - use a fine tip for small components
- Solder
  - 60/40 or 63/37 tin/lead for ease of flow (or lead-free for RoHS compliance).
  - diameter: 0.5–0.8 mm for Through-Hole Technology (THT), 0.3–0.5 mm for Surface-Mount Device (SMD).
- Soldering stand and damp sponge (for cleaning soldering iron tip)
- Tweezers (fine-point tweezers for SMD components)

- Diagonal cutters
- Flux (helps solder flow better and improves joint quality)
- Magnifier (for joint inspection)
- Desoldering pump or braid (for removing solder)

Procedure for Through-Hole Technology (THT) Components

1. Prepare the workspace and tools
    - Use a conical tip soldering iron (25–60 W, temperature ~350 °C) and solder 0.5–0.8 mm diameter with flux.
    - Have diagonal cutters ready.
    - Work on a heat-resistant surface in a well-lit, ventilated area.
2. Insert component
    - Work from smaller components to larger ones.
    - Insert the THT component leads through the correct PCB holes.
    - Ensure correct orientation (check polarity for LEDs, diodes, electrolytic capacitors).
    - Slightly bend the leads outward under the board to hold the component in place.
3. Apply heat and holder
    - Place the clean soldering iron tip on both the pad and the component lead to heat the joint.
    - After ~1 seconds, feed solder into the joint (not directly on the iron tip).
    - Allow solder to flow evenly around the lead and pad.
    - Remove solder first, then the iron.
    - Ensure component is supported until the solder solidifies.
    - Avoid overheating pads or leads (1–3 seconds per joint is usually enough).
4. Inspect the joint
    - Should be shiny, smooth, and cone-shaped.
    - Avoid cold joints (dull, cracked, or blob-like).
5. Trim excess leads
    - Use diagonal cutters to trim component leads just above the pad.
    - Ensure trimmed leads do not create shorts with other traces.
6. Cleaning
    - If using flux, clean the PCB with isopropyl alcohol to remove residue.

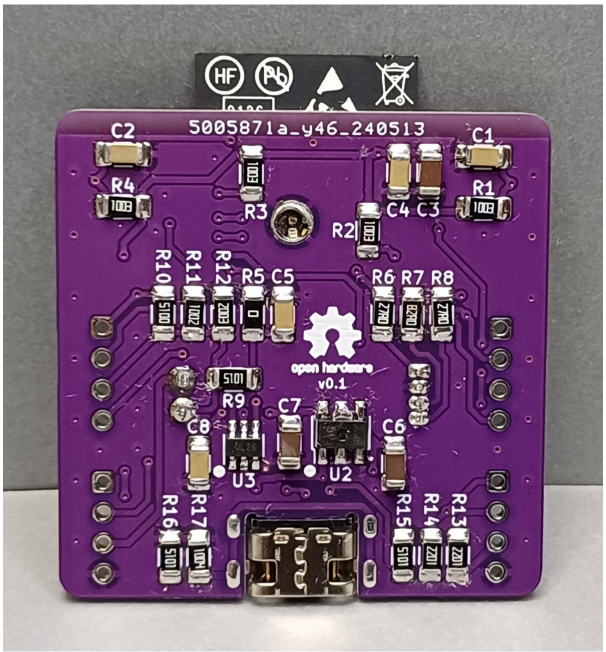


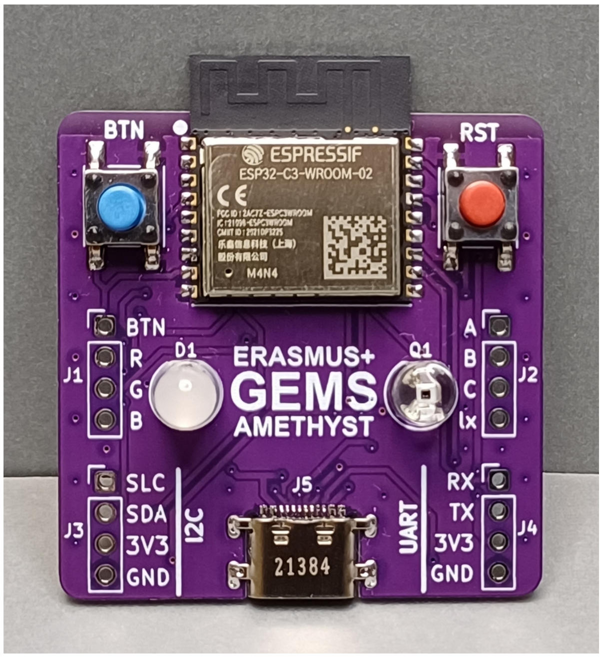


Procedure for Surface-Mount Device (SMD)

1. Prepare the workspace and tools
   - Use a fine-tip soldering iron (15–30 W, temperature ~350 °C).
   - Have tweezers, magnifier, solder and flux ready.
   - Work on a heat-resistant surface in a well-lit, ventilated area.
2. Select a component (Work from smallest components to largest.)
3. Apply solder to only one pad
   - Heat one pad with the soldering iron.
   - Apply a tiny bit of solder.
4. Apply flux
   - Apply a small amount of flux to the pads where the SMD component will sit.
   - Flux helps solder flow evenly and prevents bridging between pads.
5. Position the component
   - Use tweezers to place the SMD component on the pads.
   - Heat the pad with applied solder and its component lead simultaneously with the soldering iron.
   - Remove iron and release the component when solder solidifies.
   - Ensure correct orientation (check polarity for diodes, LEDs, ICs).
   - Double-check alignment with a magnifier.
   - Joint can be reheated to correct components orientation. Be careful to not overheat the joint.
6. Solder remaining pads
   - Verify the component is in the correct place before soldering the remaining pads.
   - Heat one pad at a time with the soldering iron.
   - Apply solder so it flows smoothly from the pad to the component lead.
   - Remove solder first, then the iron, to avoid cold joints.

7. Inspect and correct
   - Check each joint under a magnifier: should be smooth, shiny, and slightly concave.
   - Bridges (solder connecting two pads unintentionally) can be fixed using desoldering braid to wick excess solder.
8. Clean
   - Remove flux residue with isopropyl alcohol.
   - This prevents long-term corrosion and improves appearance.





Orientation of electronic components

- Non-polarized components
  - resistors, ceramic capacitors, inductors, jumpers
  - For non-polarized parts, orientation is mainly for readability and aesthetics.
- Polarized Components
  - Electrolytic capacitors (Cathode is marked with “–” stripe or longer lead is Anode)
  - Diode (Cathode is marked with a stripe on diode body)
  - LED (Cathode is marked with a flat edge or longer lead is Anode).
  - Check both component markings and PCB silkscreen symbols.
- Integrated Circuits (ICs)
  - ICs (DIP, SOIC, QFP, QFN, BGA) have Pin 1 indicators (A dot, notch, bevel, or chamfer marks Pin 1.)
  - PCB footprint usually has a dot, notch, or “1” marker.
  - Match the component’s Pin 1 mark with the PCB footprint’s Pin 1 marker.
- Transistors

    - Orientation depends on package (TO-92, SOT-23, etc.).
    - Look for flat side, chamfer, or pin order (E-B-C, G-D-S).
    - PCB silkscreen usually outlines the correct shape, however it is good to check documentation.
- Connectors
    - Many connectors are shaped to prevent incorrect insertion.
    - Silkscreen often shows Pin 1 position with a square pad or marker.

General advice

- Clean soldering iron tip and use flux.
- Do not overheat the joint. The joint can be remelted after it cools down if correction is needed.
- When soldering on GND plane or larger heat masses it takes longer for joint parts to heat up to the correct soldering temperature.
- Electrolytic capacitors, diodes, LEDs, transistors and ICs must be aligned correctly. Check polarity of the components before and after soldering the first pin. A component with more than one soldered pin is much harder to remove.
- If there is any doubt about orientation of the component check the PCB's schematic and the component’s datasheet to confirm orientation before soldering.

3 FreeCAD

Question: How to design a part with CAD?

About: FreeCAD is a free, open-source parametric 3D CAD software designed for mechanical engineering, product design, and manufacturing. Its parametric modeling approach allows users to define geometry using constraints and dimensions, making designs easy to modify and adapt.

The software is organized into workbenches for specific tasks:

- **Sketcher** for creating constrained 2D profiles.
- **Part Design** for building complex 3D parts from sketches.
- **Part** for solid modeling and Boolean operations.
- **Assembly** for combining parts into mechanical systems.
- **TechDraw** for generating 2D technical drawings.
- **FEM** for finite element analysis and structural simulation.

FreeCAD supports standard CAD formats (STEP, STL, etc.) and can be extended through add-ons and Python scripting, making it a flexible engineering tool.

Link: https://www.freecad.org/

Installation: FreeCAD is available for Windows, Linux, and macOS. Install latest stable version from the official source (https://www.freecad.org/downloads.php). Select your operating system, follow instructions and use recommended settings. After installation on first program startup select language, unit system, navigation style (Gesture) and theme. Later this can be changed in Edit → Preferences → General or in bottom right corner of the program window.

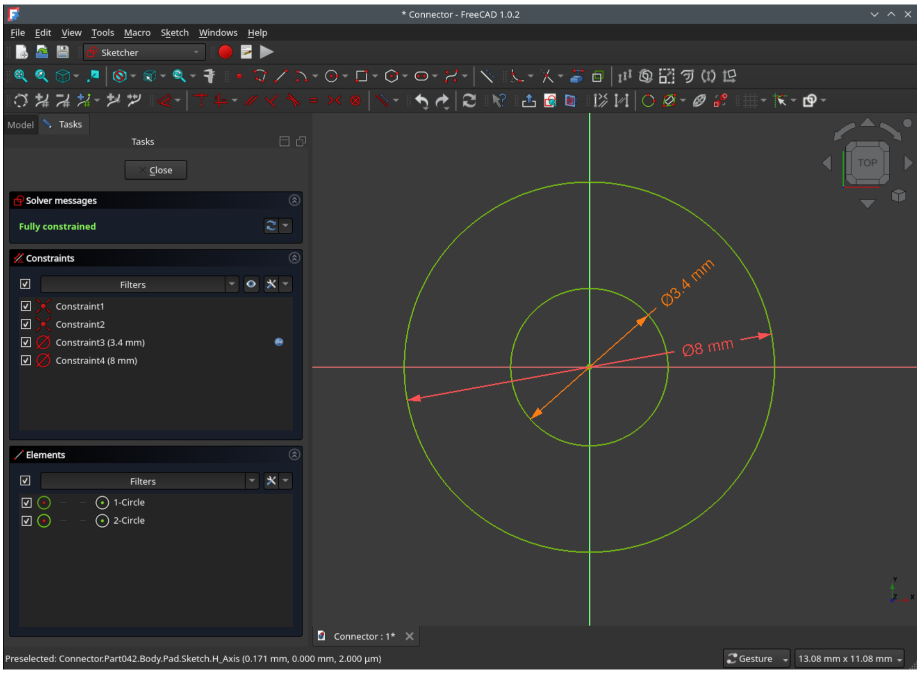
Add-ons: To add add-on start the program and:

1. Navigate to Tools → Add-on Manager
2. Search for add-on (freecad.gears workbench, Fastners Workbench)
3. Select and install add-on
4. new workbench is now available to select in the top toolbar

Create a simple part

1. Create a new Document
   - Open FreeCAD
   - Create new empty document (top toolbar → white paper icon)
   - In the workbench selector (top toolbar → drop-down list) choose Part Design.
2. Create a new Part
   - Click Create part (top toolbar → yellow block icon).
   - A Part will appear in the Model tree (left).
   - The Part is active when name font is bold.
   - Name of the part can be changed by modifying Label in Data tab (bottom left).

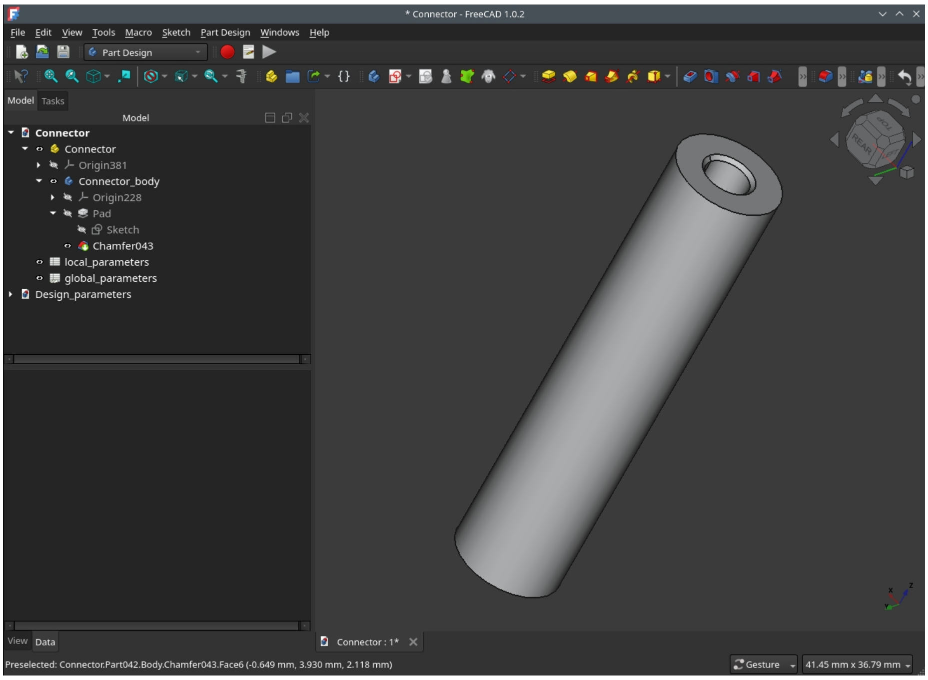
3. Create a new Body
   - Click Create body (top toolbar → blue block icon).
   - A Body will appear in the Model tree under active Part.
4. Create a Sketch
   - The Body has to be active (name with bold font).
   - Click Create sketch (top toolbar → red shapes on paper icon).
   - Select a plane (XY plane recommended).
   - If something goes wrong with the body in future steps (indicated by red alert sign in the Model tree) it is usually linked to a broken sketch Attachment Support. It can be corrected in the sketch's Data tab.
   - The Sketcher workbench opens and view changes to 2D.



5. Draw the Sketch
   - In the Sketcher toolbar, select Create circle by center.
   - Place a circle roughly in the center.
   - Select circle center point and origin point and apply Constraint coincident (top toolbar)
   - Select circle parameter and apply Dimension (top toolbar). Specify Diameter 8mm or Radius.
   - Create another smaller circle with diameter 3.4mm.
6. Close Sketch and Pad (Extrude)
   - The Sketch should be fully constrained. This is indicated by change of lines color or Solver message in Tasks tab (left).
   - Click Close to exit Sketcher.

- In Part Design, select Pad (top toolbar → yellow block above red shape icon).
- Enter length 36mm in Pad parameters in Tasks pad.
- Click OK.
- Solid is created.

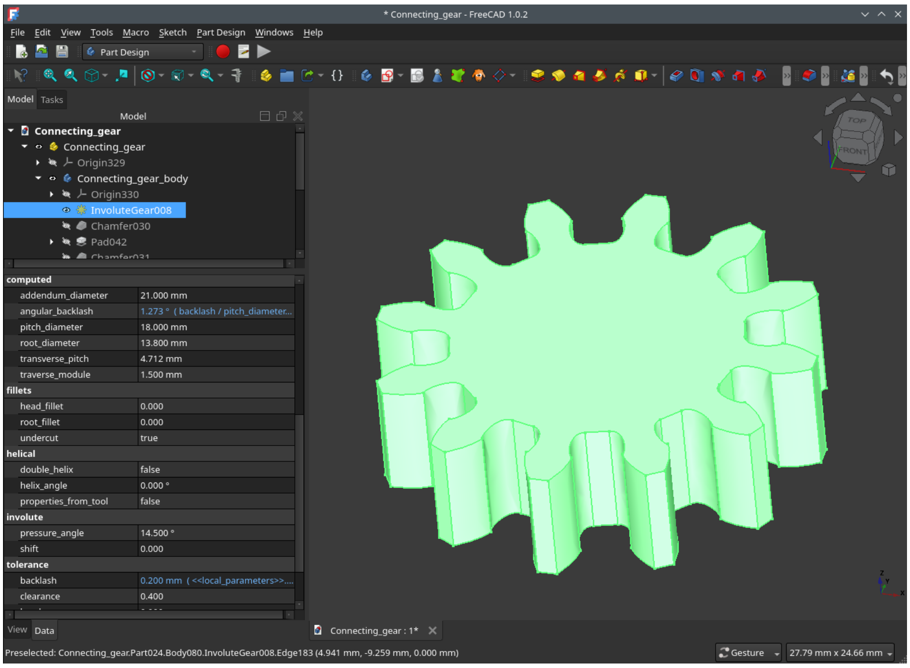
7. Add Chamfer
   - Click Chamfer (top toolbar → one of blue red icons).
   - Select all edges and specify Size 0.3mm.
   - Click OK.
   - Solid is modified.
8. If dimensions need changes:
   - Expand Body → Pad → Sketch in the Model tree.
   - Double-click a Sketch to edit constraints.
   - The 3D model updates automatically.
9. Save the document and export the Part
   - Go to File → Save As and name the document.
   - Select the part in the Model tree.
   - Go to File → Export and save as .stl file.



Create a part with special workbench

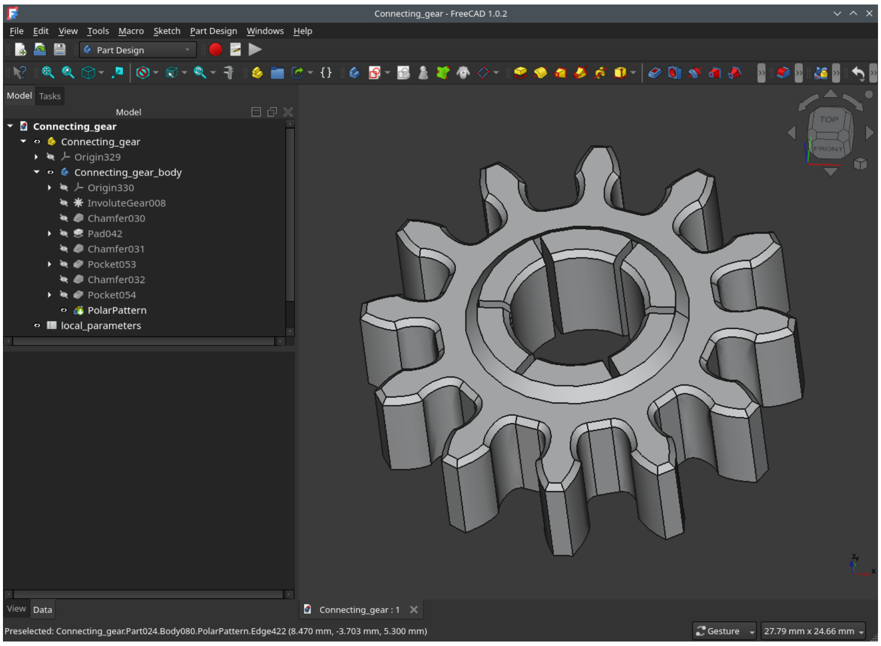
1. Open FreeCAD and create new document.
2. In the workbench selector choose Part Design.

3. Create a new Part.
4. Create a new Body.
5. In the workbench selector choose Gear.
6. Create a new gear.
    - Click Involute Gear (top toolbar → first yellow gear icon).
    - A gear with default parameters will appear in the Model tree and in 3D view.
    - Select the gear in the Model tree.
    - Set Attachment Support in Model tab → Data tab to Body's XY plane.
    - Change Map Mode from Deactivated to XY on plane (automatically suggested by bold font).



7. Modify the gear parameters.
    - Select the gear in the Model tree.
    - Parameters can be modified in in Model tab → Data tab. Set parameters:
        - accuracy → numpoints 6
        - base → height 2.4mm
        - base → module 1.5mm
        - base → num_teeth 36
        - fillets → undercut true
        - involute → pressure_angle 14.5°
        - tolerance → backlash 0.2mm
        - tolerance → clearance 0.4mm
8. Create a hole.

- In the workbench selector choose Part Design.
- In 3D view select the gear's top surface.
- Click Create sketch.
- In the sketch Create a circle by center. Select first point to be the sketch's origin point and define circle's diameter 6.5mm (you can use Dimension tool).
- The sketch should be selected in Model tree.
- Click Pocket (top toolbar → blue block with red pocket on top icon). The Pocket tool is a more general tool than Hole tool.
- Set Type in Tasks tab to Through all (or appropriate length to get through hole).



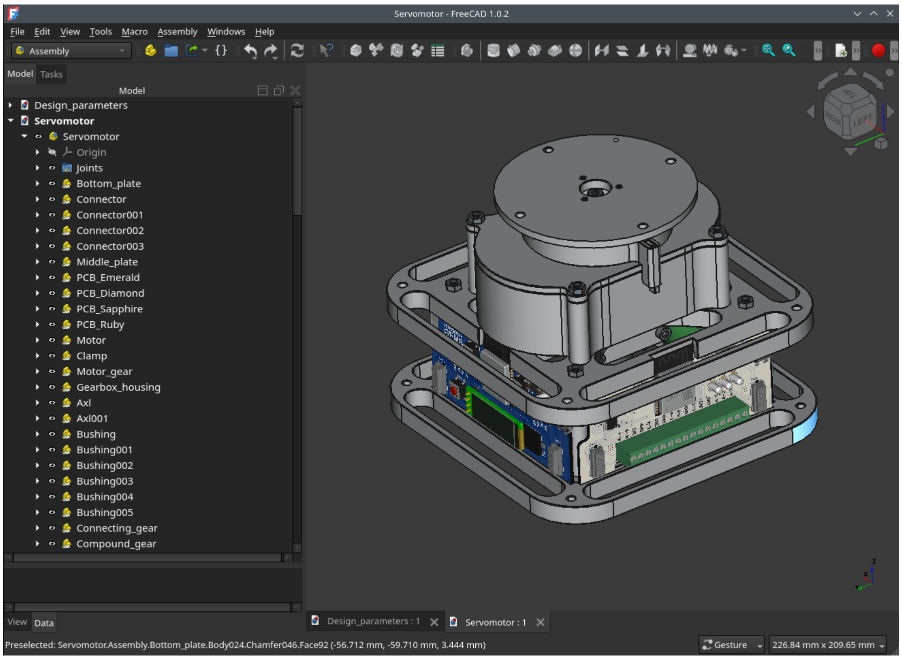
9. Create a chamfer.
   - Click Chamfer.
   - Set Size in Tasks tab → Chamfer parameters to 0.3mm. If Size is too large and complex shape is selected the program may encounter problems.
   - If program encounters problems errors will be displayed in red in the Report view at the bottom. The Report view can be closed at any time. It is automatically reopened when new errors needs to be displayed.
   - Click Select.
   - Select gear's top and bottom surfaces. (This may take some time for the program to process.)
   - Click Preview and inspect the gear's geometry for errors.
   - Click OK if no errors, otherwise reduce chamfer size.
10. Inspect the body's modification sequence.

- Select operation of interest in the Model tree and toggle display of its result with space key (or press eye icon in front oft its name). The body can also be made transparent by setting Transparency percent in Model → View tab.

11. Save document and export the part as .stl.

Create an assembly

1. Create all parts of the assembly. This can be done in the same document or several documents. Parts in the same document are displayed unless hidden (toggle display).
2. In FreeCAD open all documents containing relevant parts.



3. Create a new Assembly
   1. In the workbench selector choose Assembly.
   2. An Assembly will appear in the Model tree.
4. Insert Components.
   1. Assembly should be selected in the Model tree.
   2. Click Insert Component (top toolbar → yellow and blue blocks with green arrow icon)
   3. Click on the parts you want to insert.
   4. Confirm that the first inserted part is grounded (fixed in place).
   5. To increase the number of inserted parts left click on the part and to decrease it right click.
   6. When parts to be inserted are selected click OK.
   7. Inserted parts are shown in the Model tree and 3D view.
5. Change individual part appearance.

1. Select a part in the Model view.
2. In Model tab → View tab set Override material true and set Shape Material (Diffuse color, Transparency, ...)

6. Place parts in their positions.
    1. This can be done by manually dragging and rotating the part or by creating joints (top toolbar)
    2. Manually position the part.
        1. Right click on the part in the Model tree and select Transform operation.
        2. In Task tab set Translation Increment and Rotation Increment (at any time).
        3. Move and rotate the part by dragging coordinate system that is displayed at the part’s center.
        4. Click OK to confirm the position transformation.
    3. Create assembly joints
        1. Assembly should be active.
        2. Select appropriate joint type tool (top toolbar → yellow and blue parts with red arrow icon). e.g. select Create a Fixed joint.
        3. In 3D view select feature from the first part and then feature from the second part.
        4. In the Task tab → Create Joint you can set Offset, Rotation and change direction of the join.
        5. Click OK to confirm creation of the joint.
        6. Save the document. (Creating assembly with joints is prone to errors.)

Manually positioning the parts forces the user to better simulate assembly procedure and check the designed tolerances. Positioning the parts with joints is faster and more precise, but the assembly can easily be broken by deleting or modifying previous joints.

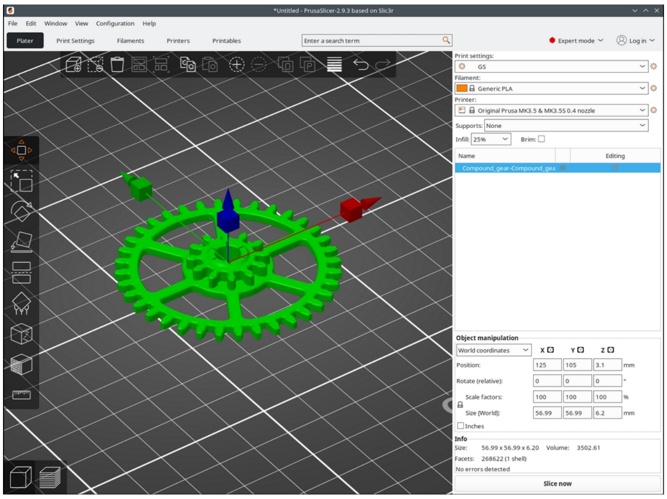
4 Prusa Slicer

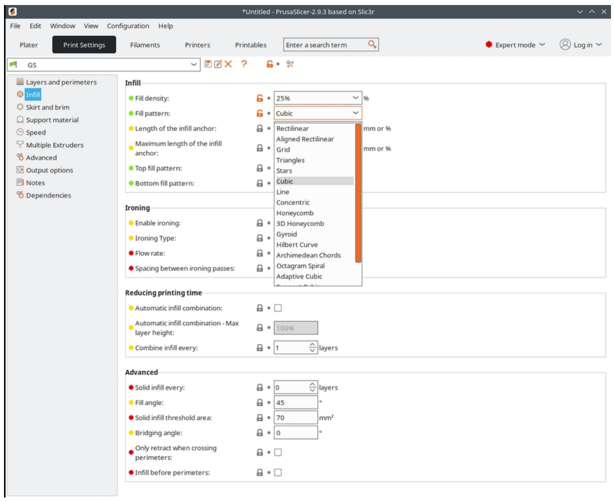
Question: How to prepare a part for FDM 3D printing?

About: PrusaSlicer is an open-source slicing software developed by Prusa Research, designed to prepare 3D models for 3D printing. It converts digital designs into G-code instructions that control 3D printers, handling tasks such as slicing models into layers, generating supports, and optimizing print settings. PrusaSlicer comes with default profiles tuned for many printers and materials, while also allowing users to create and customize their own. The program is continuously improved by both Prusa Research and the open-source community, making it a flexible and powerful tool not only for Prusa printers but also for a wide variety of other machines.

Link: https://www.prusa3d.com/

Installation: PrusaSlicer is available for Windows, Linux, and macOS. Install the latest stable version from the official source (https://www.prusa3d.com/page/prusaslicer_424/). Select your operating system, follow the installation instructions, and use the recommended settings. On the first launch, you will be prompted to configure your printer — for example, choose Original Prusa i3 MK3S. You will also be asked to select default filament profiles, such as PLA. These presets provide reliable starting parameters for most prints, but can be customized later as needed.





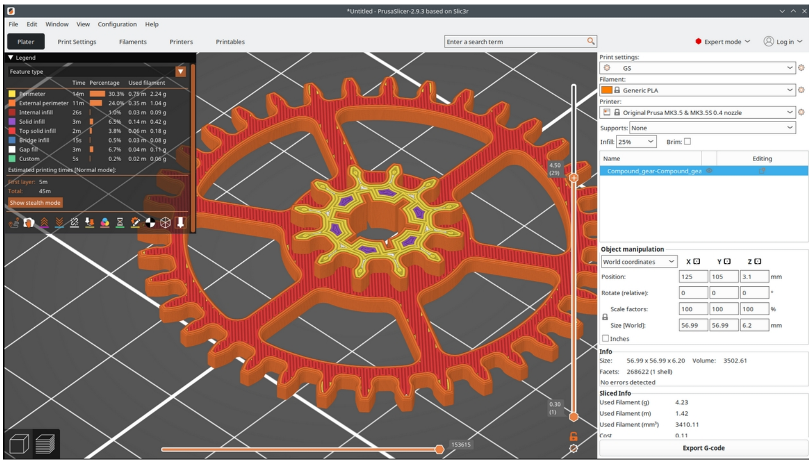
Slice a Model

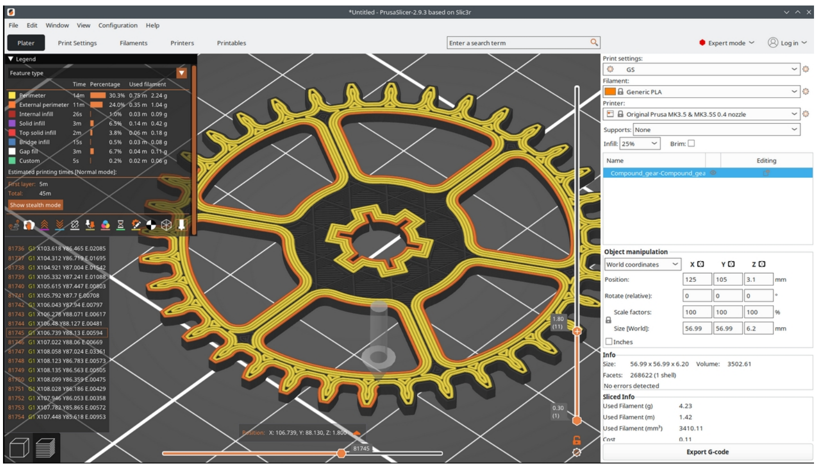
1. Open PrusaSlicer
   - Start PrusaSlicer after installation.
   - On first launch, ensure a printer profile is set up (e.g. Original Prusa i3 MK3S).
2. Import a 3D Model
   - Click Add (top toolbar → Cube with plus icon) or press Ctrl+I.
   - Browse and select a .stl or .3mf file.
   - The model appears on the virtual print bed in the 3D view.
3. Position the Model

  - Use mouse controls to rotate and zoom the view. Left click and drag to rotate, Right click and drag to pan.
  - Select the part.
  - Move model: select Move tool (left toolbar) and drag the part
  - Place model face on the print bed: select Place on face tool (left toolbar) and select face of the part that will be printed first.
  - Rotate model: select Rotate tool (left toolbar), select the rotation handle and drag it.
4. Select Printer Profile
   - In the Printer drop-down (upper right), select Original Prusa i3 MK3S.
   - For further adjustments, expand the Printers tab (upper left):
     - Check in Extruder 1 that Nozzle diameter is set correctly (default 0.4mm).
5. Select Filament Material
   - In the Filament drop-down (upper right), select Generic PLA.
   - For further adjustments, expand the Filaments tab (upper left):
     - Check in Filament that diameter is set correctly (default 1.75mm) and that the print temperatures are appropriate for the material you are using, according to the filament manufacturer's recommendations.
6. Set Print Settings
   - In the Print Settings drop-down (upper right), set 0.15mm QUALITY.
   - Supports: None
   - Infill: 30%
   - Brim: unchecked
   - For further adjustments, expand the Print Settings tab (upper left):
     - Layers and Perimeters
       - Layer height: 0.15 mm
       - Perimeters: 4
       - Solid layers: 4
       - Seam position: Aligned
       - Optionally use Plater tab (upper left) → Seam painting (left toolbar) to define preferred seam position
     - Infill
       - Fill density: 33%
       - Fill pastern: Cubic
     - Skirt and brim:
       - Loops: 10 (0 to disable)
       - Distance from brim/object: 6mm
       - Skirt height: 1 layer
7. Slice the Model

- Expand the Plater tab (upper left):
- Click Slice Now (bottom right).
- PrusaSlicer generates a toolpath preview with estimated print time and filament usage.
- Move vertical slider to inspect individual layers.
- Move horizontal slider to inspect toolpath on the individual layer.
- Inspect for Overhangs – check the model for faces angled more than 45° from vertical (or less than 45° relative to the build plate).
  - If possible, reorient the part on the build plate to reduce or eliminate overhangs.
  - If reorientation is not sufficient, enable supports in the slicer.
  - Supports improve printability but require post-processing and may result in a rougher surface finish where they contact the model.
- Click 3D editor view (bottom left) to modify part placement (needs re-slicing).

8. Export G-code
   - Click Export G-code (bottom right) and save the file (can save directly to SD card).
   - Copy file to the SD card for MK3S 3D printer.
   - Insert the SD card into the 3D printer.

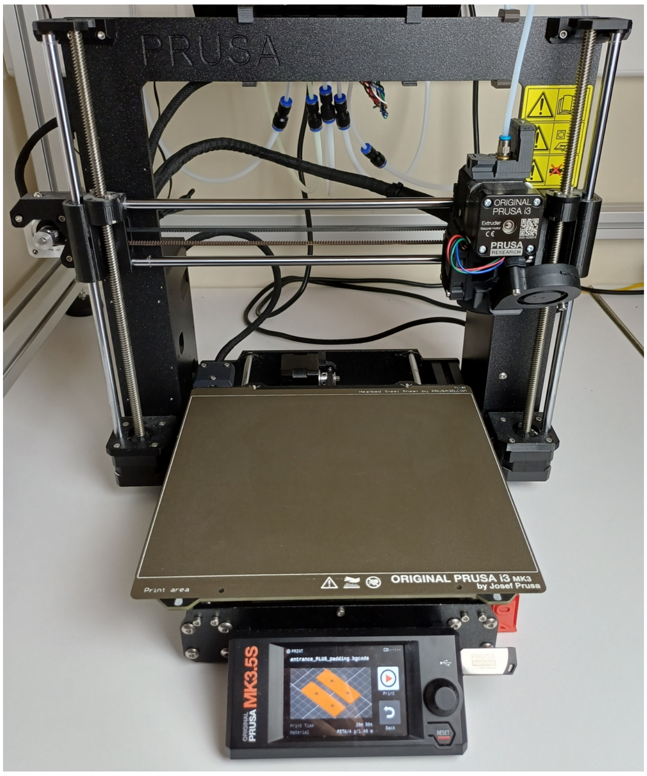


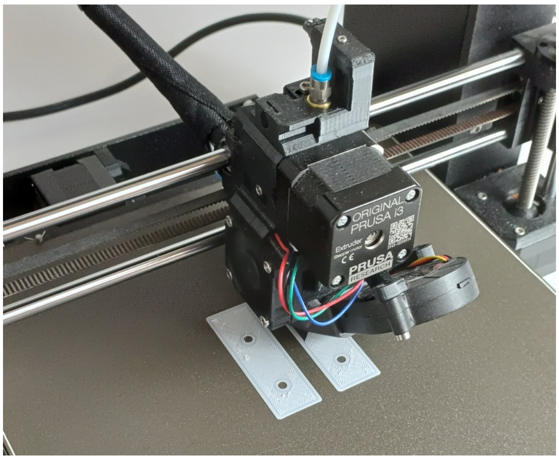


5 FDM 3D printer

Question: How to use FDM 3D printer?

About: 3D Printers are machines that create physical objects from digital models by depositing material layer by layer. The most common consumer technology is Fused Deposition Modeling (FDM), where a heated nozzle extrudes thermoplastic filament onto a build plate. A typical printer consists of a rigid frame, motion system (rails, belts, or rods), extruder and hotend, heated build plate, cooling fans, and a control interface. Printers differ in construction style — for example, Cartesian and CoreXY designs — as well as in extruder configuration, with direct drive and Bowden systems offering different performance characteristics. Build areas vary by model, and machines may use a moving bed in the Y-axis or in the Z-axis depending on design. 3D printers follow G-code instructions generated by slicing software, and they support a wide range of materials such as PLA, PETG, and ABS. Modern machines are continuously refined by both manufacturers and open-source communities, making them accessible, professional-grade tools for prototyping and production.





Materials

- **PLA** (Polylactic Acid) is the most widely used filament in FDM 3D printing. It is derived from renewable resources like corn starch or sugarcane, making it biodegradable and easy to print. PLA melts at relatively low temperatures, requires little to no heated bed. While not as heat- or impact-resistant as some other plastics, it is excellent for prototypes, decorative models, and general-purpose printing.

  Nozzle temperature: 190–220 °C

  Bed temperature: 0–60 °C

- **PETG** (Polyethylene Terephthalate Glycol) is a durable and versatile filament commonly used for functional parts. It combines the strength and temperature resistance of ABS with the ease of printing of PLA. PETG is less brittle than PLA, resists impact and moisture, and offers good layer adhesion. It requires a heated bed and benefits from controlled cooling, making it well-suited for mechanical components, enclosures, and outdoor applications.

  Nozzle temperature: 220–250 °C

  Bed temperature: 70–90 °C

- **ABS** (Acrylonitrile Butadiene Styrene) is a strong, heat-resistant filament often used in industrial and engineering applications. It can withstand higher operating temperatures than PLA or PETG, but it is more challenging to print due to warping and the release of fumes during extrusion. ABS requires a heated bed, controlled chamber temperature, and good ventilation. When printed successfully, it produces durable, long-lasting parts suitable for mechanical and functional use.

  Nozzle temperature: 230–260 °C

  Bed temperature: 90–110 °C

Build plate surfaces

- **Smooth PEI Sheet** is a popular print surface made from Polyetherimide. It offers strong adhesion for materials like PLA, PETG, and ABS when heated, while allowing parts to release easily once the bed cools. Smooth PEI surfaces produce a glossy bottom finish and are durable with proper care. Regular cleaning with isopropyl alcohol helps maintain adhesion.
- T**extured Powder-Coated PEI Sheet** provides excellent adhesion and durability, especially for materials like PETG. Its textured surface hides imperfections and gives the bottom layer a matte, textured look. Parts usually release on their own after the bed cools, making removal easy.
- **Glass Bed** is a flat, rigid surface often used with adhesives such as hairspray or glue stick to improve print adhesion. Glass provides a perfectly flat build area and produces a smooth, glossy bottom layer finish. While it is inexpensive and long-lasting, adhesion can be less reliable compared to PEI, especially for large or warp-prone materials.

After printing

Post-Processing in FDM 3D printing refers to the steps taken after a part is printed to improve its appearance, accuracy, or functionality. Simple post-processing includes removing support structures, trimming away stringing or blobs with a knife or flush cutters, and lightly sanding surfaces to smooth layer lines. For PLA and PETG, sanding and painting are common; for ABS, acetone vapor smoothing can also be used.

3D Print process:

1. Prepare the print in slicer
   - Slice the model and export G-code to SD card.
2. Prepare printing material
   - Make sure the filament you will use has the same properties as filament profile in slicer.

- Only use dry filament. (Filament can be dried but it can take several hours or even a day or two.)
- Load filament spool in the designated place on 3D printer.
- Make sure the nozzle has the correct diameter, same as used in slicer.

3. Prepare the build plate
    - Clean the surface before each print:
    - PEI sheets: wipe with isopropyl alcohol.
    - Glass plates: wash with warm water and dish soap and use glue/hairspray.
4. Select print file
    - Insert SD card into printer slot.
    - On the printer interface, navigate to the SD card menu.
    - Select the desired G-code file for printing.
    - Start the print.
    - Follow potential instructions on the printer's display.
5. Startup procedure
    - Nozzle and bed will heat up.
    - The printer will calibrate for bed surface height.
    - Filament will be extruded on the side of the build plate.
6. Printing process
    - Printer's nozzle will move to the area where part was placed in the slicer and extrude the filament while moving following the G-code in the print file.
    - Monitor the first layer: it should be smooth and slightly squished for proper adhesion. First layer adhesion is the most crucial step in the process. If the extruded filament detaches from the build plate the printing process should be canceled and build plate cleaned before trying again.
    - After each printed layer nozzle moves one layer higher from the build plate.
    - After all layers are printed the printer moves into the end position.
7. After printing
    - Wait for the bed to cool before removing the part. Parts will deform when hot.
    - Remove supports carefully, if present.
    - Clean the print bed for the next print.

6 Robot assembly

Question: How to assemble GEMS mobile robot?

About: A GEMS robot can be assembled by joining two GEMS servomotors. The GEMS servomotor consists of four PCB boards (GEMS modules: Sapphire, Ruby, Diamond and Emerald), DC motor with feedback sensors, a high reduction ratio gearbox and an output shaft with a bearing. The servomotor has a battery power supply managed by the Emerald module and can act autonomously.

GEMS mobile robot is a differential drive mobile robot and can be created by coaxially connecting 2 servomotors. On each output shaft a 3D printed wheel is mounted and to enable the robot move a smaller supporting wheel is added.

GEMS articulated robot can also be created by connecting 2 servomotors so that their axis are offset and perpendicular to each other. Output shafts of two servomotors are linked using a 3D printed L shaped arm.

6.1 GEMS Servomotor

6.1.1 Tools

- FDM 3D printer for PLA with 0.4mm nozzle
- 2.5mm Allen key for M3 bolt
- 5.5mm Wrench for M3 nut
- 3mm drill bit (with handheld bit holder)
- utility knife
- sand paper (min. P400)
- hammer
- small vise
- straight serrated tip tweezers
- flush cutter
- Soldering iron
- Hot glue gun
- Hot air gun or lighter (to heat up heat-shrink tube)
- 3D printed fixture (for flat cable with connectors)

6.1.2 Parts

3D printed parts

- bottom plate
- middle plate
- output shaft with a disc
- motor gear

- connecting gear
- **5x** compound gear
- output shaft with a gear
- gearbox housing
- gearbox top cover
- bearing ring
- **4x** plate connector
- clamp
- bearing cage
- **3x** wedge

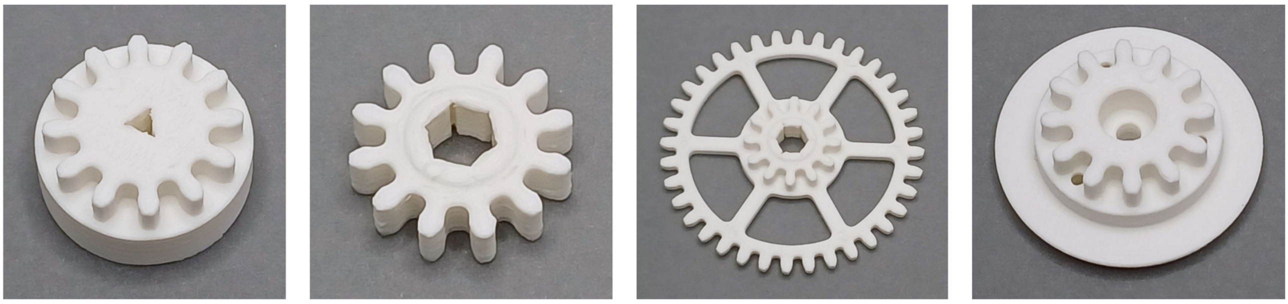
Manufacturing: Parts can be printed with PLA material on any FDM 3D printer with 0.4mm nozzle. Print settings that produce best quality prints on a particular 3D printer should be used. Post-processing may still be required to get the assembly pieces to match.

As **an example**, settings used on Ultimaker S5 are:

- slicer program: Ultimaker Cura
- profile: Normal engineering
- material: PLA
- filament diameter: 2.85mm
- nozzle size: 0.4mm
- nozzle temperature: 200 C
- bed temperature: 60 C
- layer height: 0.15mm
- infill density: 33
- infill pattern: Cubic subdivision
- Z seam alignment: random
- support: none
- adhesion: none
- bed surface: glass with thin layer of PVA glue (glue stick)

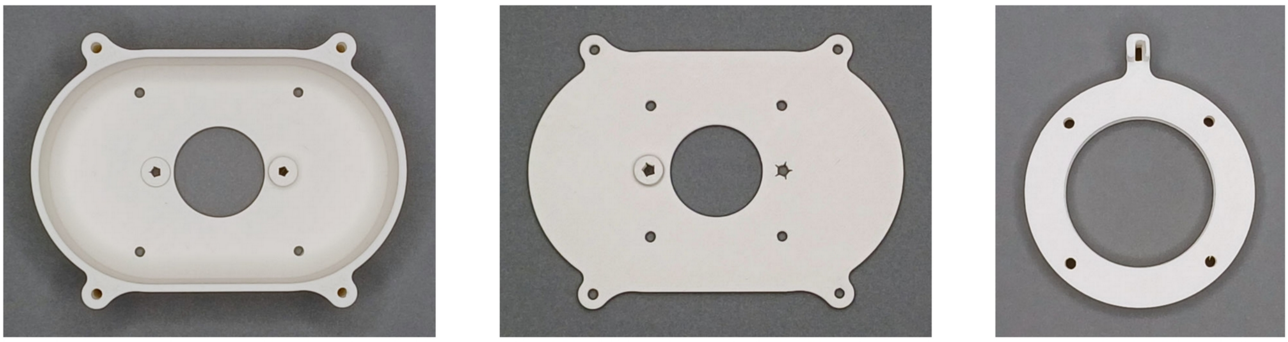
(a) (b) (c)

Fig. 6.1: Plates. (a) Bottom plate. (b) Middle plate. (c) Output shaft with disc (Top plate).



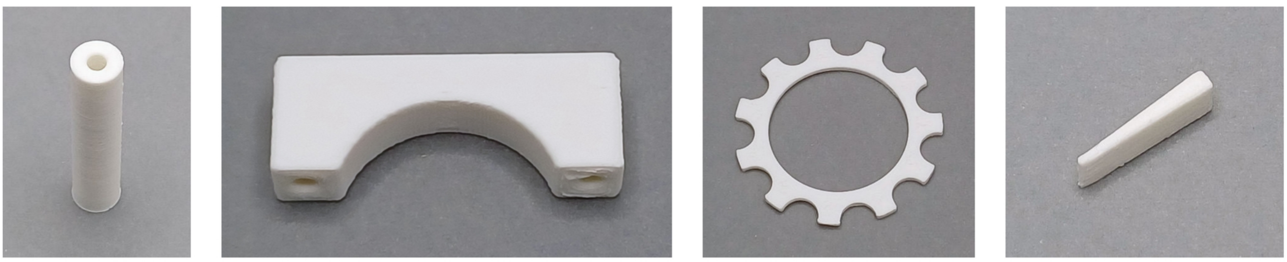
(a) (b) (c) (d)

Fig. 6.2: Gears. (a) Motor gear. (b) Connecting gear. (c) Compound gear. (d) Output shaft with gear.



(a) (b) (c)

Fig. 6.3: Housings. (a) Gearbox housing. (b) Gearbox top cover. (c) Bearing ring (Bearing housing).

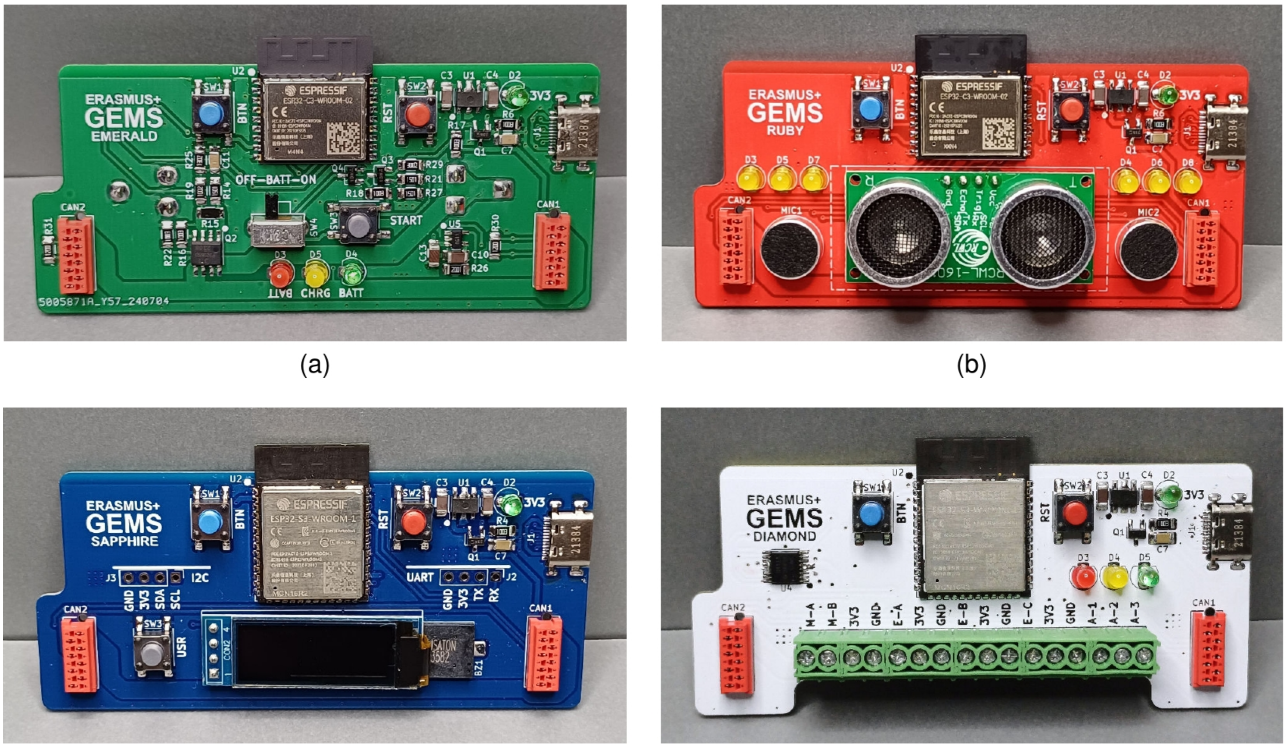


(a) (b) (c) (d)

Fig. 6.4: Smaller parts. (a) Connector. (b) Clamp. (c) Bearing cage. (d) Wedge.

GEMS modules

- Emerald
- Ruby
- Sapphire
- Diamond



(a) (b) (c) (d)

Fig. 6.5: GEMS modules. (a) Emerald. (b) Ruby. (c) Sapphire. (d) Diamond.

Standard mechanical parts

- DC motor ϕ24 x 45mm (Velleman MOT2N)
- **3x** magnets ϕ3 x 1mm (Eclipse Magnetics N835)
- **11x** bearing ball ϕ6mm (AISI 52100)
- **6x** bushing ϕ6.35mm x 6.35mm with ϕ3.05mm hole (Keystone electronics 1454A, Round spacer #4 brass 1/4")
- **2x** machinist dowel pin ϕ3mm x 30mm (DIN 6325)
- **3x** machinist dowel pin ϕ3mm x 16mm (DIN 6325)
- **4x** bolt M3 x 50mm (DIN 912)
- **6x** bolt M3 x 40mm (DIN 912)
- **5x** bolt M3 x 16mm (DIN 912)

- **4x** bolt M3 x 10mm (DIN 912)
- **20x** nut M3 (DIN 934)

Standard electrical parts

- **2x** insulated wire 100mm (AWG 20-21)
- 1nF ceramic capacitor TH
- **8x** connector (TE Connectivity 215083-8 MOW 08P 28 AWG [Micro-MaTch MALE-ON-WIRE CONNECTOR])
- **3x** hall sensor (DRV5013AGQLPG)
- **4x** 8-line flat cable 45mm
- **3x** 3-line flat cable 110mm
- heat-shrink tube 2:1 ϕ1.5mm x 15mm

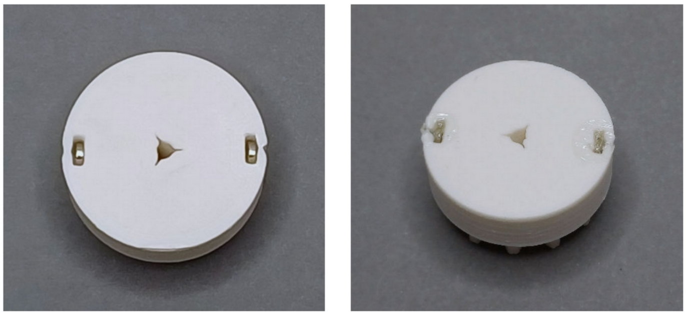
Additional material

- petroleum jelly (vaseline)
- hot glue
- solder wire

6.1.3 Servomotor assembly process

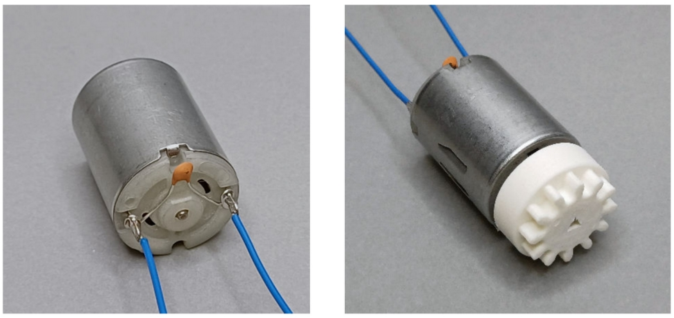
Motor

1. Put 2 magnets in motor gear and glue them in place with small amount of hot glue. (When inserted the magnets must have the same magnetic global orientation.) Remove access glue at the bottom of the motor gear.
2. Solder 1nF ceramic capacitor between motor terminals.
3. Cut 2 wires (length 100mm, AWG 20-21).
4. Remove 5mm of insulation from both ends of the 2 wires.
5. Solder the wires to each of the motor terminals.
6. Press the motor gear on to the motor shaft to create a tight fit. Use 2 small peaces of paper to create offset between the motor housing and the motor gear. Remove paper peaces. Motor gear should not touch the motor housing while rotating.
7. Check all 18 holes (ϕ3.2mm) in the middle plate and 2 in the clamp. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill bit (can be done by hand).
8. Check hall sensor cutout for 3D printing defects. If the sensor does not fit, clean the cutout with a small file or other appropriate tool.
9. Place the clamp next to the central hole of the middle plate.
10. Use 2 bolts (length 40mm) and nuts to position the clamp on the middle plate. Leave place for the motor.
11. Place the motor in the central hole create a transition fit by equally tightening the 2 bolts. After the motor will be more precisely positioned bolts will be tightened further to prevent motor movement in relation to middle plate.



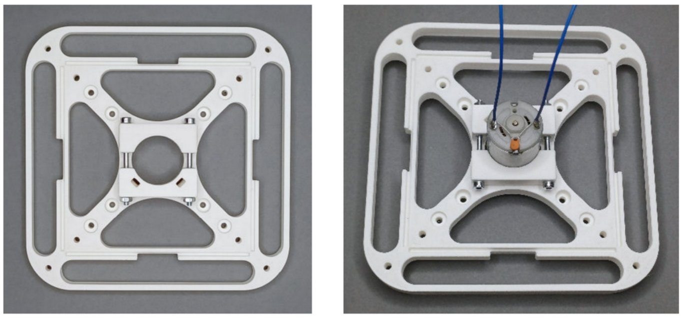
(a) (b)

Fig. 6.6: Motor gear with magnets assembly.



(a) (b)

Fig. 6.7: Motor connections assembly.



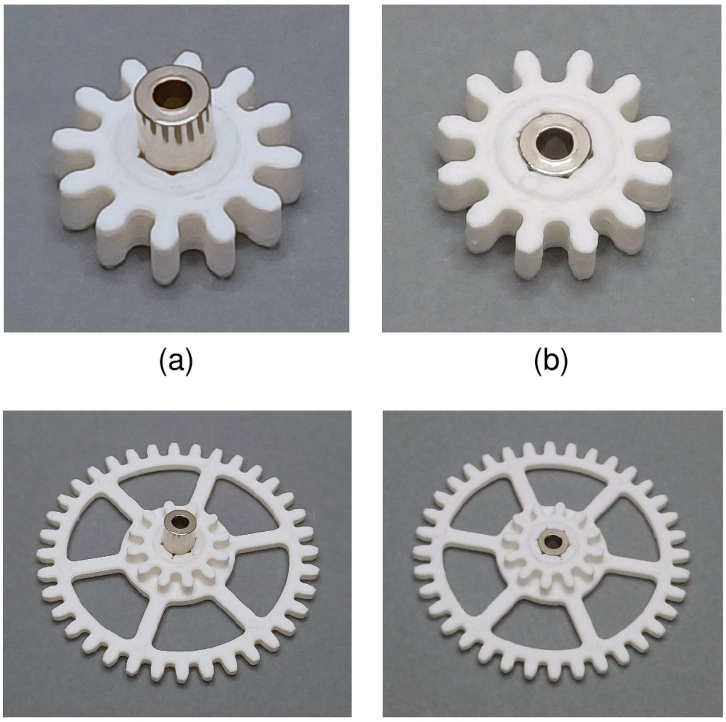
(a) (b)

Fig. 6.8: Motor on middle plate assembly.

Gearbox

1. Check all gear teeth for 3D printing defects and precisely remove them with a utility knife or fine sanding paper. Try to achieve proper tooth geometry.
2. Press/force metal bushings in the central hole of all 5 compound gears and 1 connecting gear. Tight fit should be created. Use a hammer or a small press. Also use 2 flat parallel metal surfaces to assure gear and bushing coaxiality.
3. Check that bottom of the gear surface is flush with bushing side surface.

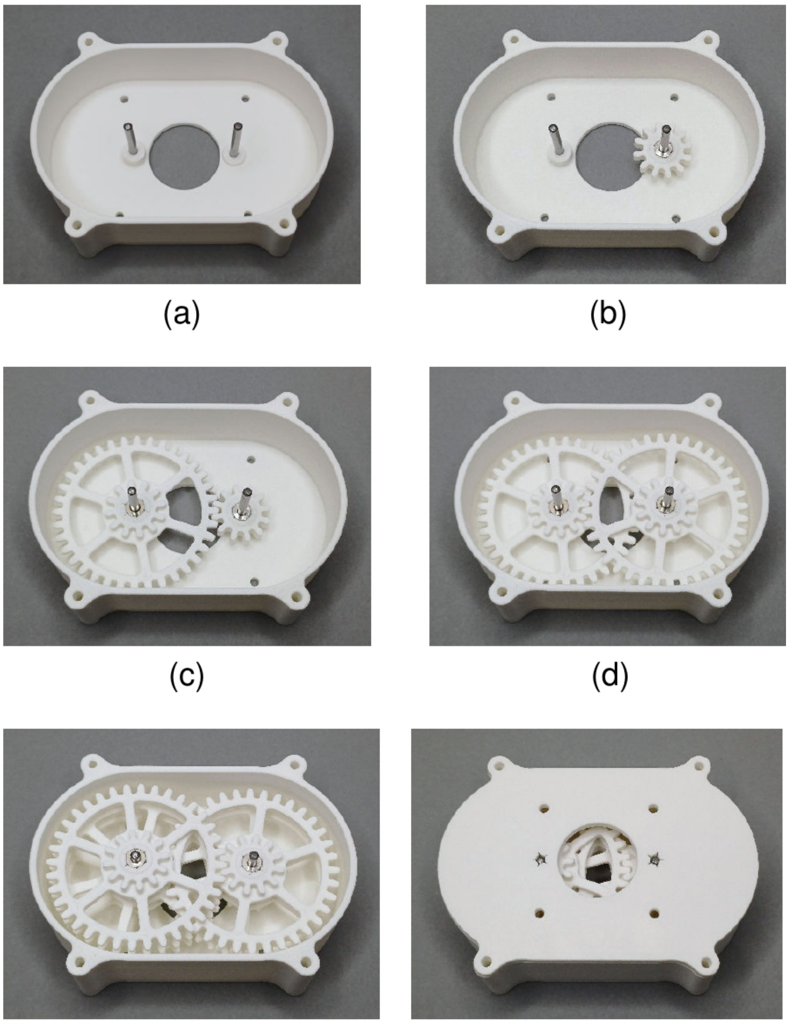
4. Force 2 machinist dowel pins (ϕ3mm and length 30mm) in the holes next to the center hole of the gearbox housing. Use a hammer or a small press. Tight fit should be created. The bottom pin surface should be flush with the the gearbox housing bottom surface.
5. Check all remaining 8 holes (ϕ3.2mm) in the gearbox housing. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill bit (can be done by hand).
6. Lubricate both machinist dowel pins with petroleum jelly.
7. Lubricate the connecting gear bushing with petroleum jelly and place it on the dowel pin with smallest offset feature at the bottom of the gearbox housing.
8. Adjust the motor position to create best possible mesh between the motor gear and the connecting gear.
9. Rotate the gears and check for jamming that could be caused by a 3D printing defect. The defect should be removed.
10. Lubricate one compound gear and place it on the empty dowel pin so it meshes with the connecting gear. Check for jamming and contact with the motor gear.
11. Gear teeth can also be lubricated with a small amount of petroleum jelly.
12. Lubricate other compound gears and alternately place them on the dowel pins while checking for jamming. After 3 compound gears are added, the gearbox is no longer back-drivable and care should be taken not to break the assembly.
13. Order of compound gears can be changed to optimize the gearbox performance. Well-matched gears should be placed closer to the motor.



(a) (b)

(c) (d)

Fig. 6.9: Gears bushings assembly.



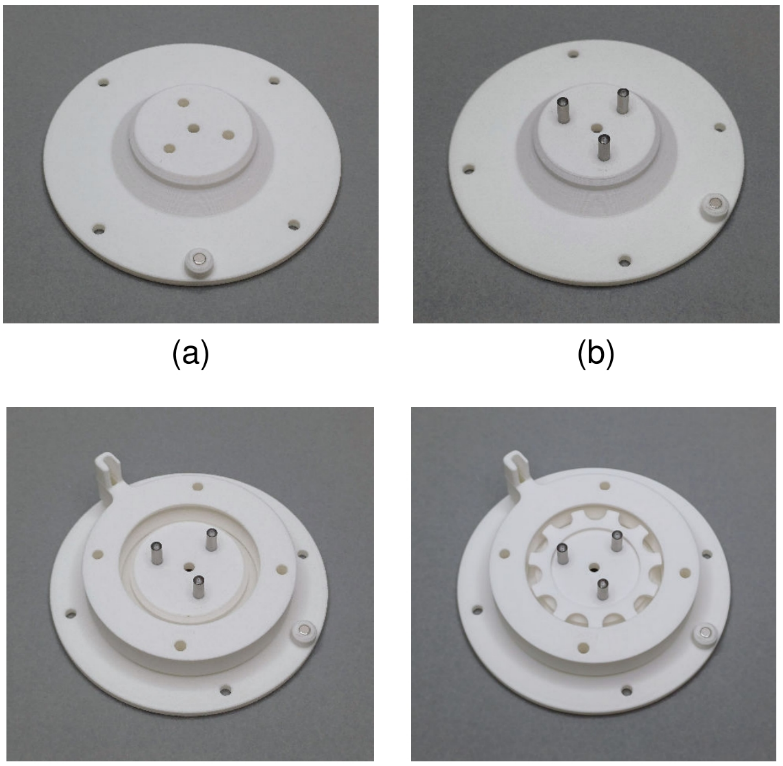
(a) (b) (c) (d) (e) (f)

Fig. 6.10: Gearbox assembly.

Bearing

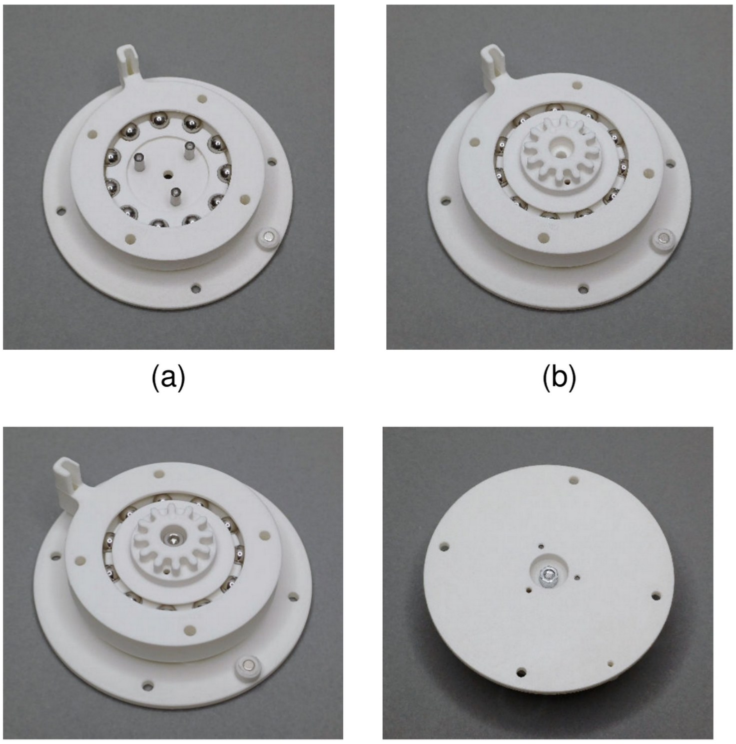
1. Check all parts of the bearing (output shaft with a gear, output shaft with a disc, bearing ring) for 3D printing defects. Remove the defects from bearing ball rolling surfaces (45° angled surface). Try to achieve smooth surface with fine sand paper (min. P400).
2. Check hall sensor cutout for 3D printing defects. If the sensor does not fit, clean the cutout with a small file or other appropriate tool.
3. Check the set of 3 holes (ϕ3mm) near the center of the output shaft with a gear. The holes are spaced 120° apart and have smaller diameter at one end. If the machinist dowel pin (ϕ3mm and length 16mm) does not fit, enlarge the hole with a 3mm drill (can be done by hand). Do not enlarge the smaller diameter end of the hole.
4. Check all other holes (ϕ3.2mm) in the bearing parts and the gearbox top cover. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill (can be done by hand).

5. Place hot glue in the magnet hole of the output shaft with a disc and then insert the magnet. The magnet surface should be flush with the top surface of the magnet holding feature. Remove excess glue if necessary.
6. Insert 3 machinist dowel pins (ϕ3mm and length 16mm) in 3 holes near the center and spaced 120° apart.
7. Position the output shaft with a disc flat on the table (pins facing up).
8. Position the bearing ring centrally on top of the output shaft with a disc so that it protrudes inside the bearing ring central hole.
9. Place the bearing cage inside the bearing ring central hole and on top of the output shaft with a disc.
10. Arrange 11 bearing balls (ϕ6mm) in spaces around the cage.
11. Position the output shaft with a gear on top of the pins while gear facing up. The holes should be aligned with the pins.
12. Carefully press the output shaft parts together until all bearing balls are in contact with both output parts. The balls must remain in their positions in the cage.
13. Connect the output shaft parts with central bolt and 2 nuts. The bolt should be tightened just enough to eliminate shaft wiggle but allow for its free rotation. First nut is then locked in place with the second nut.
14. The bearing can be lubricated by adding petroleum jelly on the bearing balls and rotating the shaft to spread the lubricant.



(a) (b) (c) (d)

Fig. 6.11: First part of bearing assembly.



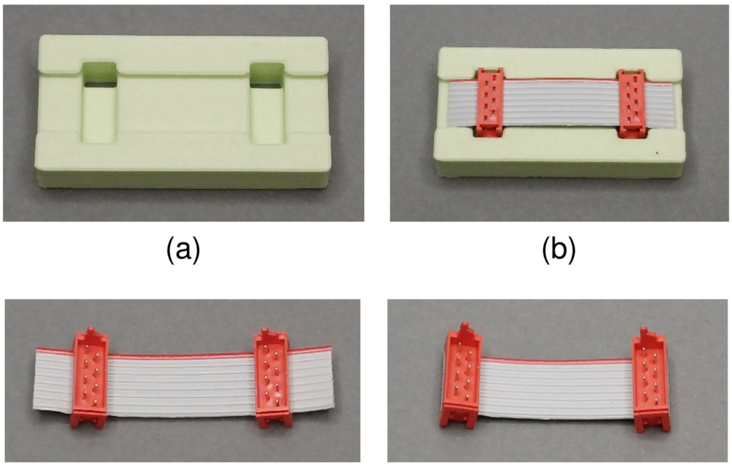
(a) (b)

(c) (d)

Fig. 6.12: Second part of bearing assembly.

4x Flat cable with connectors

1. Cut 8-line flat cable to length (45mm). Top line should be red.
2. Position 2 connectors on the flat cable 23mm apart. Place the cable with connectors in the provided 3D printed fixture.
3. Press the connectors until they are locked. It is best to use a vise with flat jaws. Connectors will make click sound when locked.
4. Cut the excess cable.



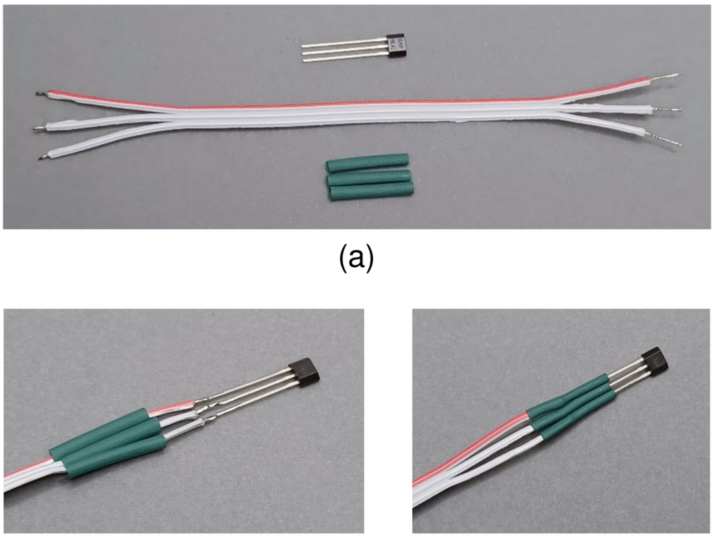
(a) (b)

(c) (d)

Fig. 6.13: Flat cable assembly process.

3x Hall sensor

1. Cut 3-line flat cable to length (110mm). Top line should be red. Split lines at each end (25mm) and remove insulation at one end 5mm and on the other 3mm.
2. Cut appropriate diameter heat-shrink tube to length (15mm).
3. Place heat-shrink tubes on lines on the side with less removed insulation.
4. Solder the red line to Vcc pin of the hall sensor (at the end of the pin).
5. Solder the middle line to GND pin of the hall sensor (at the end of the pin).
6. Solder the last line to OUT pin of the hall sensor (at the end of the pin).
7. Position all heat-shrink tubes so that the middle of each tube is approx. at the solder joint.
8. Apply heat to heat-shrink tubes.

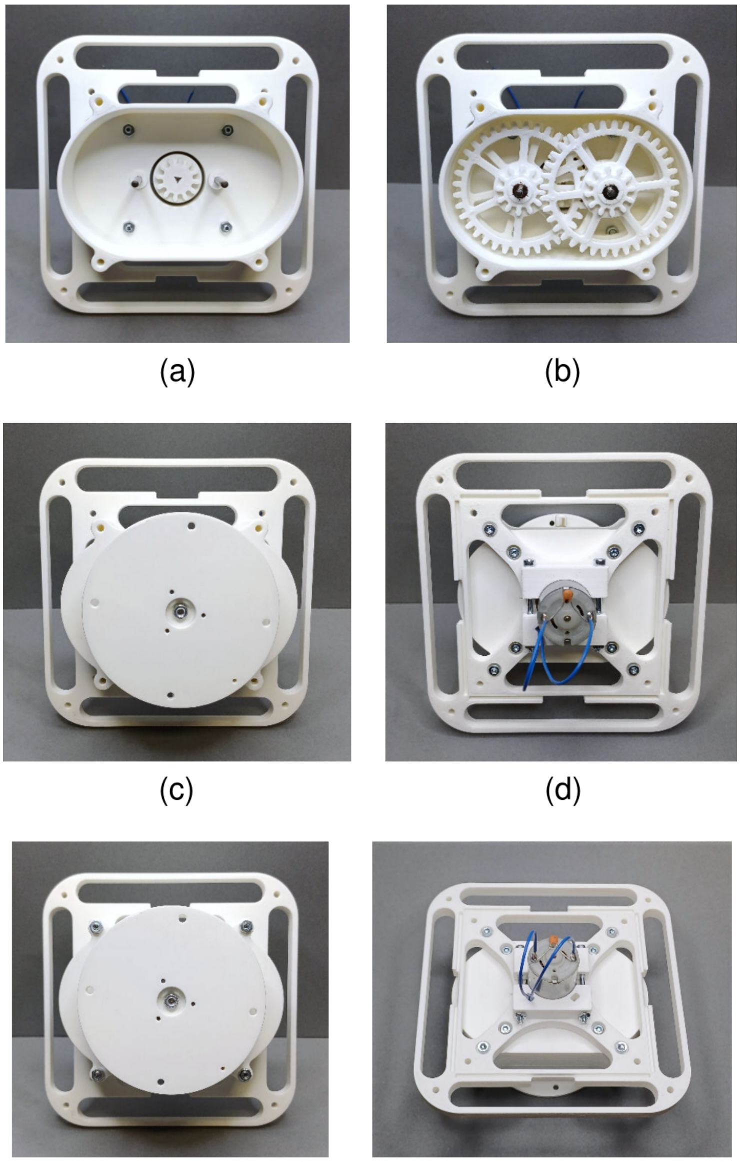


(a) (b) (c)

Fig. 6.14: Hall sensor assembly process.

Final assembly

1. Remove all gears from the gearbox housing. Preserve the best order of gears.
2. Use 4 bolts (length 10mm) and nuts to connect gearbox housing with the middle plate. Use holes closest to the center. Position the motor gear in the center of the gearbox housing central hole.
3. Place the connecting gear on the dowel pin with smallest offset feature at the bottom of the gearbox housing.
4. Adjust the motor position to create best possible mesh between the motor gear and the connecting gear while preventing contact with the first compound gear.
5. Place all other gears in the gearbox housing.
6. Connect the bearing assembly with the gearbox top cover using 4 bolts (length 16mm) and nuts.



(a) (b)

(c) (d)

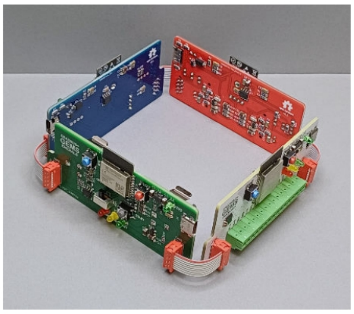
(e) (f)

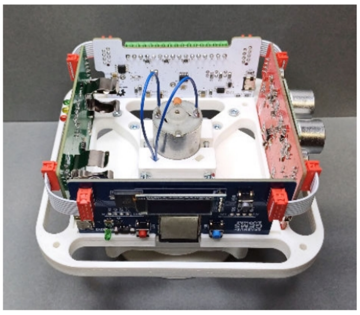
Fig. 6.15: First part of final assembly process.

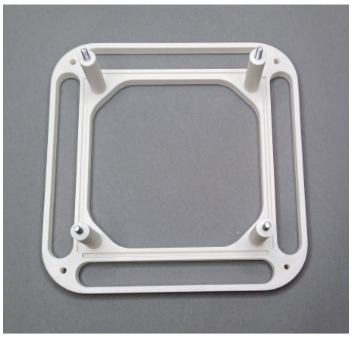
7. Place the bearing assembly with the gearbox top on top of the gearbox. Rotate the output shaft if needed to create a mesh between last compound gear and gear on the output shaft. The gearbox is not back-drivable.
8. Connect the bearing assembly with the gearbox top cover and the gearbox assembly with middle plate by inserting 4 bolts (length 40mm) in the remaining holes of the middle plate closest to the center and pushing them through the gearbox holes. Place the nuts and tighten the bolts.
9. Place the assembly flat on a table so that the middle plate is facing up.
10. Connect all 4 GEMS modules with flat cables in a loop.
11. Place the GEMS module loop in the groove of the middle plate. Diamond module should be placed on the side with the output shaft hall sensor hole. Care should be taken not to

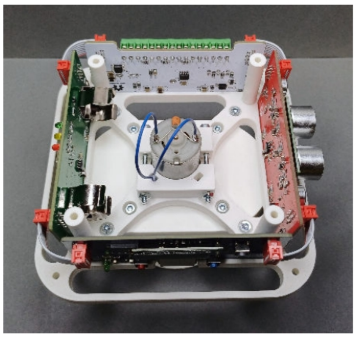
damage PCB components (protruding ESP32 PCB antenna (black), resistors and capacitors near the module edge).

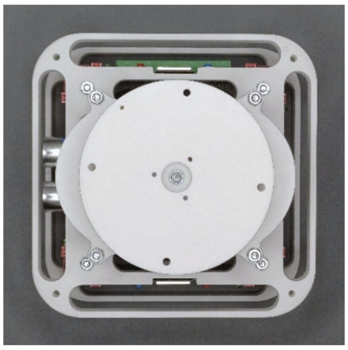
12. Check all 8 holes (ϕ3.2mm) of the bottom plate. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill (can be done by hand).
13. Check holes (ϕ3.2mm) of 4 plate connectors (length 40mm). If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill (can be done by hand).
14. Separately position the bottom plate bottom surface up.
15. Insert 4 bolts (length 50mm) in innermost holes of the bottom plate. Bolts should hang down through the plate.
16. Bring the bottom plate with bolts closer to the assembly aligning the bolts with the holes in the middle plate.


(a)


(b)


(c)

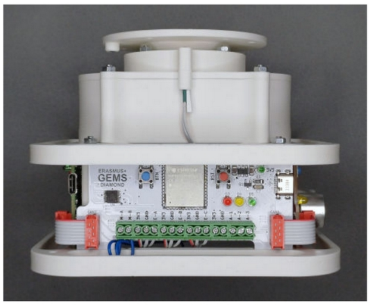

(d)

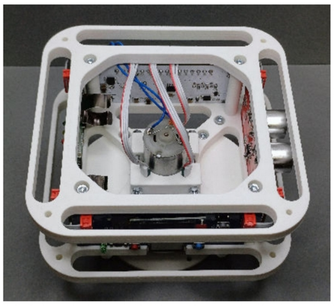

(e)

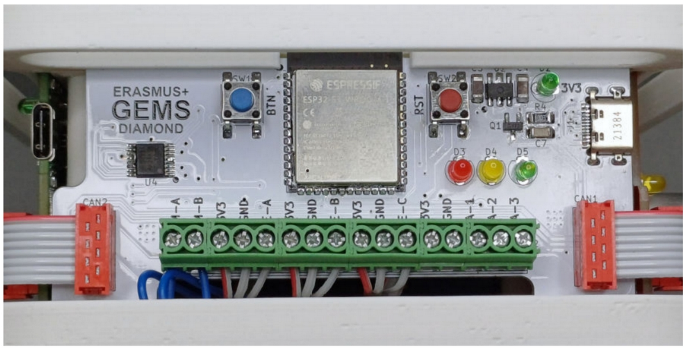

(f)

Fig. 6.16: Second part of final assembly process.

17. Place plate connectors on the bolts and then insert the bolts in the middle plate holes.
18. Carefully correct the position of the GEMS module loop so that all modules sit in the groove of the middle and bottom plate.
19. When the GEMS module loop is in place fasten the bolts and nuts.
20. Insert 2 hall sensors in the cutouts in the middle plate near the motor. The sensors should be facing (tapered end) towards the motor and pushed to the bottom of the gearbox housing. The sensor is fixed in place by a small wedge.
21. Insert the last hall sensor in the cutout in the bearing ring. (If the sensor can't be inserted, the bearing assembly should be removed and then new attempt to insert the sensor can be made.)
22. Connect motor and sensors wires to the GEMS Diamond module.


(a)


(b)



(c)

Fig. 6.17: Third part of final assembly process.

6.2 Mobile robot assembly process

6.2.1 Tools

- 2.5mm Allen key for M3 bolt
- 5.5mm Wrench for M3 nut

6.2.2 Parts

2x GEMS servomotor

3D printed parts

- **2x** mobile robot wheel
- mobile robot support

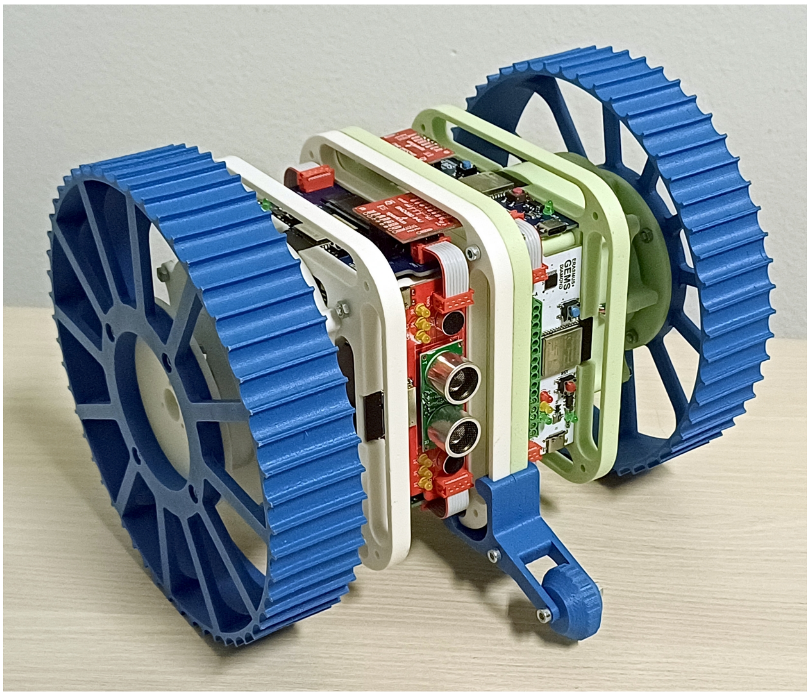
Standard mechanical parts

- **8x** bolt M3 x 10mm (DIN 912)
- **4x** bolt M3 x 20mm (DIN 912)
- **2x** bolt M3 x 25mm (DIN 912)
- **14x** nut M3 (DIN 934)

6.2.3 Assembly process

1. Place the 2 servomotors on the work-surface with output shaft pointing upwards.
2. Check the holes (ϕ3.2mm) of the output shaft disc of both servomotors and the holes (ϕ3.2mm) of the mobile robot wheels. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill (can be done by hand).
3. Use 4 bolts (length 10mm) and nuts to connect each wheel to the servomotor's output disc. Optionally use insulation tape on one side of the wrench head to hold a nut in place while positioning it.
4. Insert a battery in the holder on the Emerald module off each servomotor.
5. Place the servomotor that will be on the left side of the robot so that the wheel is flat on the work-surface and the Sapphire module is facing to the left.
6. Place the servomotor that will be on the right side of the so that the wheel is facing away from the work-surface and the Sapphire module is also facing to the left. Both Sapphire modules are next to each other. On the opposite side there are both Emerald modules. On the other two sides Ruby and Diamond module are next to each other so that Ruby on the top servomotor is facing forward.
7. Use 4 bolts (length 20mm) and nuts to connect the two servomotor's bottom plates. The Allen key can optionally be inserted through the hole of the middle plate for better access to the bolt head.
8. Attach one part of the mobile robot support on the bottom plate of one servomotor so that it is pointing backwards.
9. Place the small wheel of the mobile robot support on the attached part of the mobile robot support. Make sure it can spin.
10. Attach the second part of the mobile robot support so that the small wheel is now enclosed.

11. Use 2 bolts (length 25mm) and nuts to permanently connect the two parts of mobile robot support.
12. Place the assembled robot on the worksurface so that both large wheels and the small support wheel are at the bottom and the Sapphire modules are both at the top. Robot is now ready to be programmed by programming each module with the corresponding program.



6.3 Articulated robot assembly process

6.3.1 Tools

- 2.5mm Allen key for M3 bolt
- 5.5mm Wrench for M3 nut

6.3.2 Parts

2x GEMS servomotor

3D printed parts

- articulated robot arm

Standard mechanical parts

- **4x** bolt M3 x 10mm (DIN 912)
- **4x** nut M3 (DIN 934)

6.3.3 Assembly process

1. Check the holes (ϕ3.2mm) of the output shaft disc of both servomotors and the holes (ϕ3.2mm) of the articulated robot arm. If M3 bolt does not fit through a hole, enlarge the hole with a 3mm drill (can be done by hand).

2. Place one of the servomotors on the work-surface with output shaft pointing upwards. This servomotor will be the robot's base.
3. Place the articulated robot arm on the servomotor's output disc so that the longer section is pointing upwards.
4. Use 2 bolts (length 10mm) and nuts to connect the articulated robot arm to the servomotor's output disc. Optionally use insulation tape on one side of the wrench head to hold a nut in place while positioning it.
5. Place the second servomotor next to the other part of the articulated robot arm so that the servomotor is on top of the first servomotor.
6. Use 2 bolts (length 10mm) and nuts to connect the articulated robot arm to the second servomotor's output disc.
7. Insert a battery in the holder on the Emerald module off each servomotor.
8. Robot is now ready to be programmed by programming each module with the corresponding program.

7 Arduino IDE

Question: How to program GEMS mobile robot?

About:

Arduino IDE is an open-source development environment for programming and uploading code to microcontrollers. When used with ESP32-C3 and ESP32-S3 devices from Espressif Systems, it provides a powerful platform for building automation and wireless projects. The ESP32-C3 (RISC-V core) and ESP32-S3 (dual-core Xtensa LX7) are Wi-Fi and Bluetooth LE–enabled microcontrollers that combine efficiency, power, and flexibility. The Arduino language, based on C/C++, integrates with the ESP32 Arduino core, allowing access to advanced features such as multitasking, power management, USB communication, and wireless networking.

Arduino IDE in combination with ESP32 microcontroller was selected for GEMS servomotor and mobile robot because of its simple setup and use that allows for rapid prototyping without sacrificing performance or control.

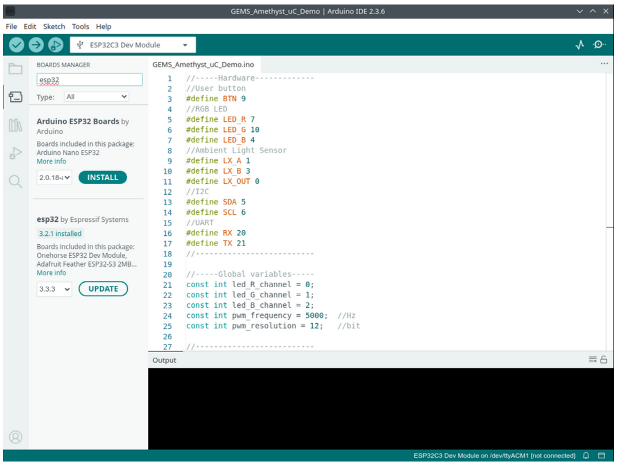
Link: https://www.arduino.cc/

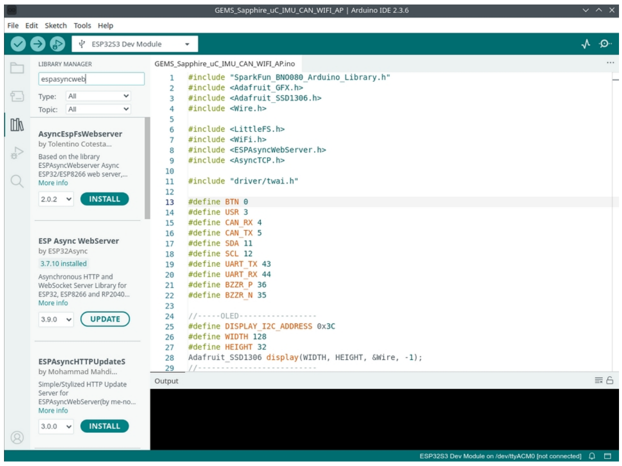
Installation: The Arduino IDE is available for Windows, Linux, and macOS. Download the latest version from https://www.arduino.cc/en/software and follow the installation steps for your operating system. Once installed, add support for the ESP32 boards/microcontrollers:

1. Open File → Preferences (or Arduino → Preferences on macOS).
2. In the Additional Boards Manager URLs field, paste:
   - https://espressif.github.io/arduino-esp32/package_esp32_index.json
3. Go to Tools → Board → Boards Manager, search for ESP32, and click Install on esp32 by Espressif Systems.

Configuration: Configure the specific board you are using to enable USB serial and Wi-Fi communication.

1. Tools → Board menu → esp32 (select ESP32C3 Dev Module or ESP32S3 Dev Module).
2. Tools → USB CDC on Boot: Enabled
   - This allows the ESP32-C3/S3 to use its built-in USB interface for serial communication with your PC. Make sure to select the correct Port under Tools → Port — that appears automatically once USB CDC is enabled.
   - Tip: If USB serial communication between micro-controller and PC doesn't work first check if this setting was reset.
3. Tools → CPU Frequency:
   - ESP32-C3: 160 MHz
   - ESP32-S3: 240 MHz





Libraries: To install Arduino Libraries in Arduino IDE use build-in Library Manager.

1. Go to Sketch → Include Library → Manage Libraries….
2. In the Library Manager window, search for the library you want by name or keyword.
3. Check the version you need (latest stable version is recommended).
4. Click Install.
5. Once installed, the library will be available under Sketch → Include Library in your projects.

7.1 Arduino program

To create a new program (sketch) in Arduino IDE:

1. Go to File → New Sketch
2. Go to File → Save (The new program is saved as .ino file with specified name in a folder with the same name. **Tip:** Program and folder name must remain the same for Arduino IDE to compile the program.)

The Arduino program **structure**:

- include **libraries**
  ```
  #include <WiFi.h>
  ```
- define **constants**
  ```
  #define BTN 0
  ```
- declare **global variables**
  ```
  float motor_reference = 0.0;
  ```
- define **classes**
  ```
  class PID { ... }
  ```
- **setup()** function runs once at the start of the program
  ```
  void setup() { ... }
  ```

- **loop()** function runs repeatedly in a loop for the lifetime of the program.
  `void loop() {…}`
- define **functions**
  `float get_time_change(unsigned long t_1, unsigned long t_0) {…}`

Short list of common Arduino core **functions**:

- Configure a digital pin as input, output, or input with pull-up
  `pinMode(pin, mode);`
- Set a digital pin HIGH or LOW
  `digitalWrite(pin, value);`
- Read the state of a digital pin (HIGH or LOW)
  `int x = digitalRead(pin);`
- Read the value from an analog pin
  `int y = analogRead(pin);`
- Output PWM signal on a pin
  `analogWrite(pin, value);`
- Pauses the program for a specified number of milliseconds
  `delay(ms);`
- Returns the number of milliseconds since the program started
  `unsigned long t = millis();`
- Initializes serial communication at a specified baud rate
  `Serial.begin(baud);`
- Sends data to the serial monitor
  `Serial.println("text");`

For **official documentation** of all core functions and syntax visit Arduino Reference web page https://docs.arduino.cc/language-reference/. Use of LLMs can also be of great help when trying to interpret an example program or write your own. However, AI generated should be used as an example and not blindly copied. Ask AI to explain how the program works and what a specific function does. Also check the language references if generated code doesn't work as expected.

Tip: If your program doesn't compile, carefully read the error messages shown at the bottom of the Arduino IDE. They usually indicate the line number where the problem occurs (or near it). Common mistakes include:

- Missing a semicolon (;) at the end of a line
- Typing errors, such as a misspelled function or variable name

Verify/compile new code incrementally to discover and correct mistakes as soon as possible.

7.2 Programming the microcontroller

To check if the Arduino IDE was successfully installed it is always useful to first program the microcontroller with a simple program that sends the string "Hello world!" over USB serial connection.

1. Open or create the **Hello world** program.

   ```
   void setup() {
     Serial.begin(115200);
   }
   void loop() {
     delay(3000);
     Serial.println("Hello world!");
   }
   ```

2. Select the correct target microcontroller / board.
   - Go to Tools → Board menu → esp32 (select ESP32C3 Dev Module or ESP32S3 Dev Module)
3. Compile the program.
   - Go to Sketch → Verify/Compile or click button with check mark in top left corner
   - If compilation process ends with errors displayed in the Output tab at the bottom of the window, correct the errors in the program and recompile.
4. Upload the program.
   - Connect the target board.
     - Select the board in the box next to the button with arrow mark in top left corner.
     - The board can also be selected by selecting the correct port in Tools → Port.
   - Double-check if board configuration is correct.
   - Go to Sketch → Upload or click button with arrow mark in top left corner
   - Upload process should end successfully at 100% without errors in the output tab.
     - **Tip:** If error occurs during upload, try to set the board in Programming Mode:
       1. hold boot button BTN
       2. press and release reset button RST
       3. release boot button BTN
       4. restart Upload process
5. Press reset button RST to restart the program.
6. In Arduino IDE open Serial Monitor to display the incoming message form the board.
   - Go to Tools → Serial Monitor
   - Click magnifying glass icon in the top left corner.
7. Serial Monitor tab will open at the bottom and every 3 seconds “Hello world!” will be printed out in a new line.

7.3 GEMS examples

To find useful examples of programs for GEMS boards explore the project’s gitHub page: https://github.com/GemsErasmusPlus. Example programs can be used as a starting point for creating your own program or source of information on how to use the GEMS hardware.

Tip: Don't use a code if you don't know what it does. You can always use AI in combination with online documentation to help you understand a new piece of code line by line before you upload it to the microcontroller.

7.4 Web server

Both ESP32C3 and ESP32S3 microcontroller can be used as a web server that can host web page related files like .html, .css, .js and also images. However, for the web page to be displayed in a browser, the device with the browser (e.g. smartphone, PC) must be able to to connect to the microcontroller running the server. This can simply be done in two ways: (1) all devices are connected to the same router or (2) microcontroller acts as an Access Point (AP) and other devices can directly connect with it.

In the case of GEMS servomotor Sapphire board is intended as communication hub and will act as a AP and webserver. In this way we don't need a router and both communicating devices can be battery powered. This is useful if we want to use the GEMS servomotor or robot outside.

Access Point (AP):

1. Include library.

```
#include <WiFi.h>
```

2. Define AP parameters as global constants.

```
const char *ssid = "GEMS_MR_1";
const char *password = "equalequal";
IPAddress local_ip(192, 168, 3, 1);
IPAddress gateway(192, 168, 3, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress start_ip(192, 168, 3, 2);
IPAddress end_ip(192, 168, 3, 100);
```

3. Start AP in setup function.

```
WiFi.mode(WIFI_AP);
WiFi.softAP(ssid, password, 1, 0, 10);
WiFi.softAPConfig(local_ip, gateway, subnet);
```

When you run the code on the microcontroller you will be with your smartphone able to see the new wireless network and can also connect with it by using the password. Your device will be given an ip adress within the set range.

File server:

1. Install and include libraries

```
#include <LittleFS.h>
#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>
```

2. Define server port and server

```
const uint16_t serverPort = 80;
```

```
AsyncWebServer server(serverPort);
```

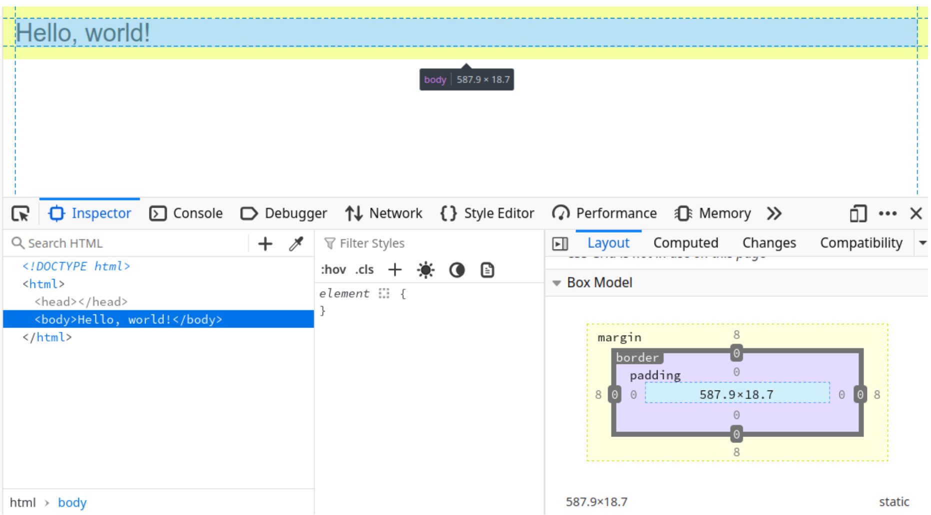
3. In setup function
    1. Mount file system.
        ```
        LittleFS.begin()
        ```
    2. Define server root directory and file.
        ```
        server.serveStatic("/", LittleFS,
        "/www/").setDefaultFile("index.html").setFilter(ON_AP_FILTER);
        ```
    3. Start server.
        ```
        server.begin();
        ```

Server files:

1. Install the arduino-littlefs-upload from gitHub.

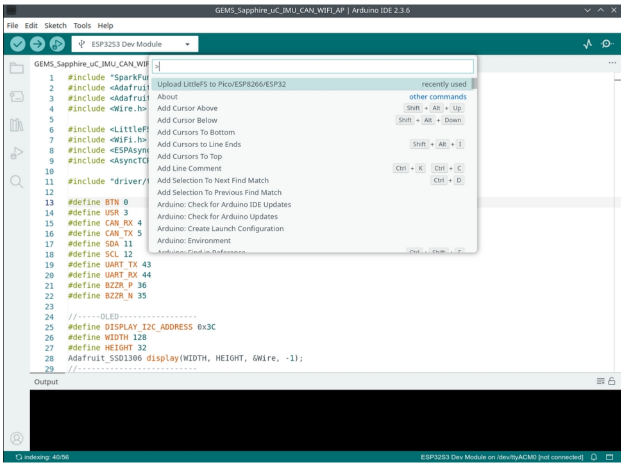
    link: https://github.com/earlephilhower/arduino-littlefs-upload

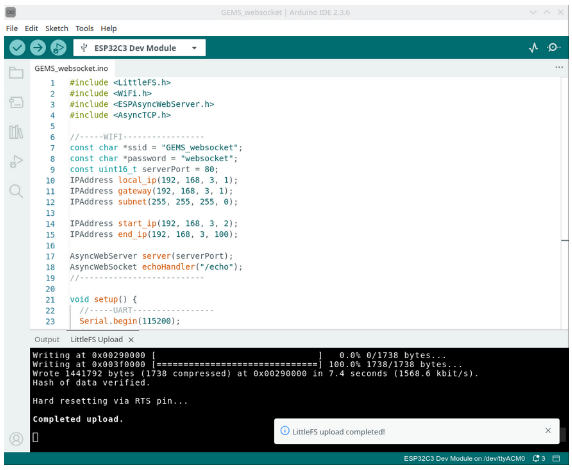
    Follow simple installation process described in README.
2. Create folders and files in sketch folder.
    1. create folder “data”
    2. in “data” folder create folder “www”
    3. in “www” folder create file “index.html” for first test or place website files.
        ```
        <!DOCTYPE html>
        <html>
          <body>
            Hello, world!
          </body>
        </html>
        ```



3. Upload files to the microcontroller.
    1. Check that files you want server to host are in the sketch subfolder named data.
    2. **Important:** Close all Serial Monitor windows.
    3. In Arduino IDE press keys [Ctrl] + [Shift] + [P]
    4. Find and select "Upload LittleFS to Pico/ESP8266/ESP32" from the menu that opened at the top of the window. If upload doesn't start restart Arduino IDE and try to upload again.

When you run the microcontroller connect your smartphone to the AP, open browser and navigate to the web page by writing address 192.168.3.1 into the browser address bar.





7.5 Web sockets

Because in our applications we want to have responsive control over the servomotor we need fast communication between user interface and microcontrollers. This can be achieved by creating a persistent web socket connection between a user interface on a website opened on a smartphone and a microcontroller with an asynchronous web server. Incoming messages are handled by handler functions that can also respond with an outgoing message.

Tip: For testing a Firefox browser extension **WebSocket Weasel** can be used.

- link: https://github.com/mhgolkar/Weasel

Websocket server:

1. Define websocket object and endpoint address.

    ```
    AsyncWebSocket echoHandler("/echo");
    ```

2. In setup function.
    1. Attach event handler function.

        ```
        echoHandler.onEvent(onEchoEvent);
        ```

    2. Add socket to the server.

        ```
        server.addHandler(&echoHandler);
        ```

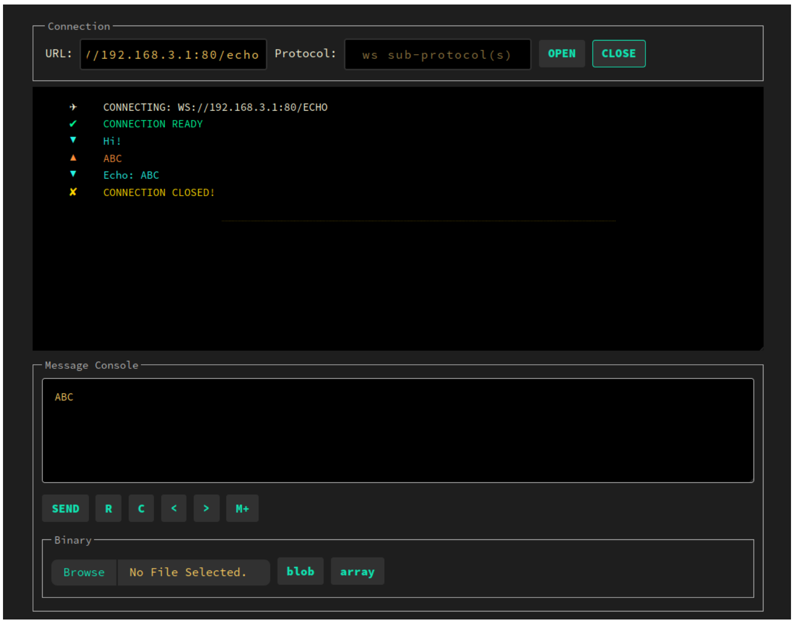
3. Create event handler function.

```
void onEchoEvent(AsyncWebSocket *server, AsyncWebSocketClient *client,
                 AwsEventType type, void *arg, uint8_t *data, size_t
len) {
  switch (type) {
    case WS_EVT_CONNECT:
        client->text("Hi!");
      break;
    case WS_EVT_DISCONNECT:
        client->text("Bye!");
      break;
    case WS_EVT_DATA:
      String msg = "";
      for (size_t i = 0; i < len; i++) {
        msg += (char)data[i];
      }
      client->text("Echo: " + msg);
      break;
  }
}
```

When you run the microcontroller with updated program connect your smartphone to the AP, open browser extension **WebSocket Weasel** and open websocket connection.

- URL: ws://192.168.3.1:80/echo

If connection is opened successfully you should see message “Hi!” from the server. Then try to send some text from the Message Console and server should echo the text back.



ACKNOWLEDGMENT

This work was co-funded by the European Union Erasmus+ programme, project: 2022-1-SI01-KA220-HED-000087727.

Co-funded by the European Union