# Experimental Project Overview

1. About the experimental project

In the MAS AT experimental project you will transfer all your learnings from the preceding CAS modules into practice – putting the "applied" into *Applied Technology*. You will take a technical project of your own from concept to reality, ending up with a working prototype. This covers the initial idea, electrical concept, mechanical design and software logic, ending up with testing of function and analysis of data. The goal is to give the participants a wide range of experiences in hardware and software, touching every aspect of R&D prototype development.

The development will start from the ground up, using pre-fabricated components as little as possible. The participants are encouraged to come up with their own project ideas and pursue them with continual support from APS. Example projects will be available for inspiration and as templates and can be adopted instead of an own idea with as little or as many modifications as desired. Large parts of the project can be completed independently, using freely available software running on the participants' computer and corresponding education resources. Some meetings will be scheduled for teaching and discussion and several hands-on sessions in the APS lab will be necessary for manufacturing and testing of the prototypes.

The topics include

- Conceptualization of an idea and planning the realization
- Electrical design of circuit and PCB using ECAD software
- Assembly and soldering of PCB
- Mechanical design of housing and other components using CAD software
- Manufacturing of mechanical components using a 3D printer
- Programming embedded software to run on a microcontroller
- Programming computer software to interface with the hardware and analyze data
- Testing of the desired functionality and measurement of the electrical workings
- Report of the process, the finished prototype and analysis of generated data
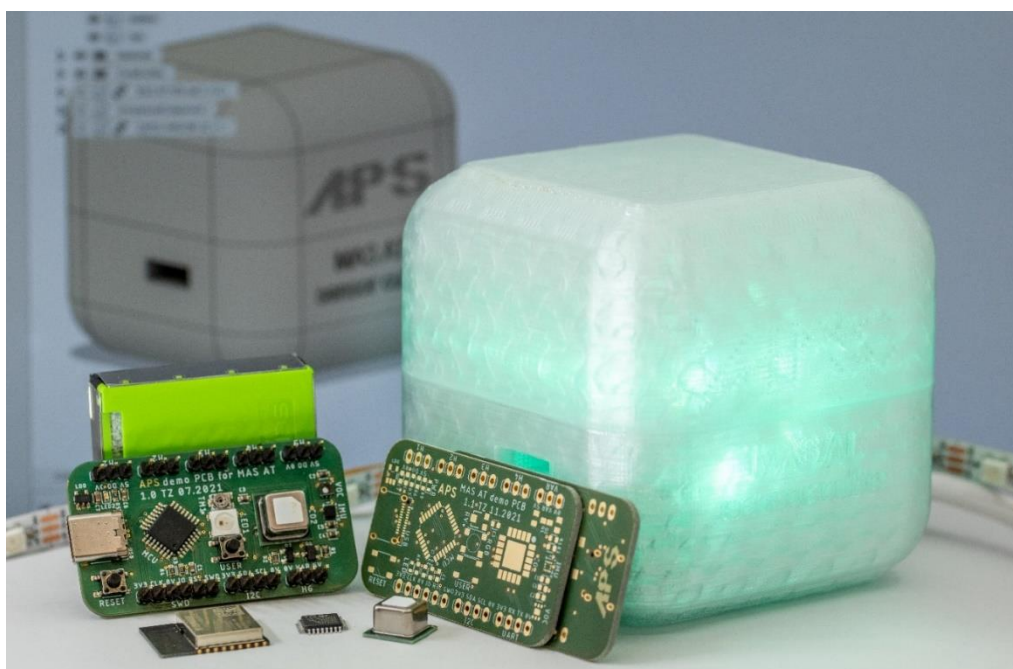
An example project, a 3D printed air quality sensor cube, showing low CO2 level by green light.
Next to it: components and circuit boards used (in 2021), the CAD drawing in the background.

2. Time plan

With 10 ECTS, the nominal time requirement of this course is 250 h. The schedule is based on a trimester (January - April), but the bulk of the work can be performed independently and most sessions can be attended remote, via Zoom. An in person kick-off day takes place at the beginning of January for an introduction and a first hands-on with example projects and components, with the remaining lectures and exercises done via zoom on a weekly basis.

2.1. Kick-off day

- Introduction of people and project
- Lecture on electronics and components
- First hands-on with available components and tools
- In person attendance highly recommended, partial zoom and recording
- **5th of January**, 9:00 to 15:00

2.2. Foundations

- Introduction to technologies and software
- Attendance highly recommended, recordings available
- On Zoom, recorded
- 10 lecture sessions, 9:00 to 10:30

Number	Date	Topic
1	05.01.2024	Project introduction, electronics part 1 (kick-off)
2	12.01.2024	Electronics, part 2
3	19.01.2024	Circuit design
4	26.01.2024	PCB design & manufacturing
5	02.02.2024	CAD (mechanical design)
6	09.02.2024	3D printing (mechanical manufacturing)
7	16.02.2024	Embedded software (CircuitPython)
8	23.02.2024	Computer software (Python)
9	01.03.2024	Testing & measurement
10	08.03.2024	Documentation & reporting

2.3. Exercises

- Exercises to give participants first experiences with technology and software
- Attendance recommended to collect initial experiences and work on issues
- On Zoom, recorded
- 10 sessions, 10:30 to 12:00, after the lectures

Number	Date	Topic
1	05.01.2024	Project ideas discussion, part 1 (kick-off)
2	12.01.2024	Project ideas discussion, part 2
3	19.01.2024	Circuit design

4	26.01.2024	PCB design
5	02.02.2024	Circuit & PCB design, participants' projects
6	09.02.2024	CAD
7	16.02.2024	CAD, participants' projects
8	23.02.2024	Embedded software
9	01.03.2024	Computer software
10	08.03.2024	Software, participants' projects

2.4. Lab days

- Hands-on learning (first two sessions)
- Completion of participant hardware (third to fifth session)
- Attendance required
- Each participant needs one session (A or B) per topic for a total of 5 sessions
- At ETH, ETL building, in the APS lab D 11
- Signup in Moodle required, up to 10 students per date
- Fridays from 13:00 to 18:00, Saturdays from 10:00 to 15:00

Number	Weekday	Date	Topic
1A	Friday	02.02.2024	Electronics
1B	Saturday	03.02.2024	Electronics
2A	Friday	16.02.2024	3D printing
2B	Saturday	17.02.2024	3D printing
3A	Friday	01.03.2024	PCB soldering & testing
3B	Saturday	02.03.2024	PCB soldering & testing
4A	Friday	15.03.2024	System testing & Debugging
4B	Saturday	16.03.2024	System testing & Debugging
5A	Friday	05.04.2024	Measurement
5B	Saturday	06.04.2024	Measurement
If required	Friday	20.04.2024	Final problem solving

2.5. Independent learning and design work

- Topics and time estimates
  - Concept: 10 h
  - Schematic / PCB: 50 h
  - CAD: 50 h
  - 3D printing/slicing 10 h
  - Embedded programming: 50 h
  - Computer programming: 10 h
  - Report & Presentation: 10 h

Total time required for course: 250 h, of this 60 h presence or telepresence, of this 25 h in the lab.

3.   Deadlines & deliverables

- Concept
    - One page description of what you want to do
    - Description and drawings as text or .pdf on moodle
    - List of components you want to use, as survey on moodle
    - Due **Sunday, 21st of January**
- Schematic
    - Based on concept, preparation for PCB design
    - Screenshot on moodle, plus name of the schematic in Fusion360
    - Due **Sunday, 4th of February**
- PCB design
    - For fabrication (external, takes one week)
    - As .zip with fabrication data on moodle
    - Due **Sunday, 18th of February** (two weeks before your third lab day, PCB soldering & testing)
    - Ideally, no iterations
- CAD design
    - For printing (at APS, takes half a day per participant)
    - As .stl or .3mf on moodle
    - Due **Sunday, 3rd of March** (one week before your fourth lab day, system testing)
    - Iterations possible
- Presentation
    - Short presentation and demonstration of your project for other participants
    - **Friday, 3rd of May, 9:00 to 12:00** (four weeks after final regular lab day)
    - In person, remote participation possible
- Report
    - Describing what you wanted to do, what you did & what you learned
    - Due **Sunday, 5th of May**

4.   Example project: air quality cube

With COVID, indoor air quality has come into focus, as it impacts transmission risk, but also general health and wellbeing. CO_2 content of the air is often used as a quality indicator: high CO_2 values indicate a badly ventilated space, while low values show good ventilation. The CO_2 value is not only useful as a proxy for air quality, high CO_2 values directly degrade our wellbeing and mental performance. For this reason, an air quality measurement device is an interesting and useful project.

A complete example project in cube shape with a CO_2 sensor and controllable LEDs as display method is available, including electronics, enclosure and software. The example uses the LEDs to reflect the air quality based on CO_2 content, with a smooth fade from green - good via yellow - ok (ventilate) to red - bad (ventilate immediately). A vibration motor is used to give an alarm when *red* is reached.
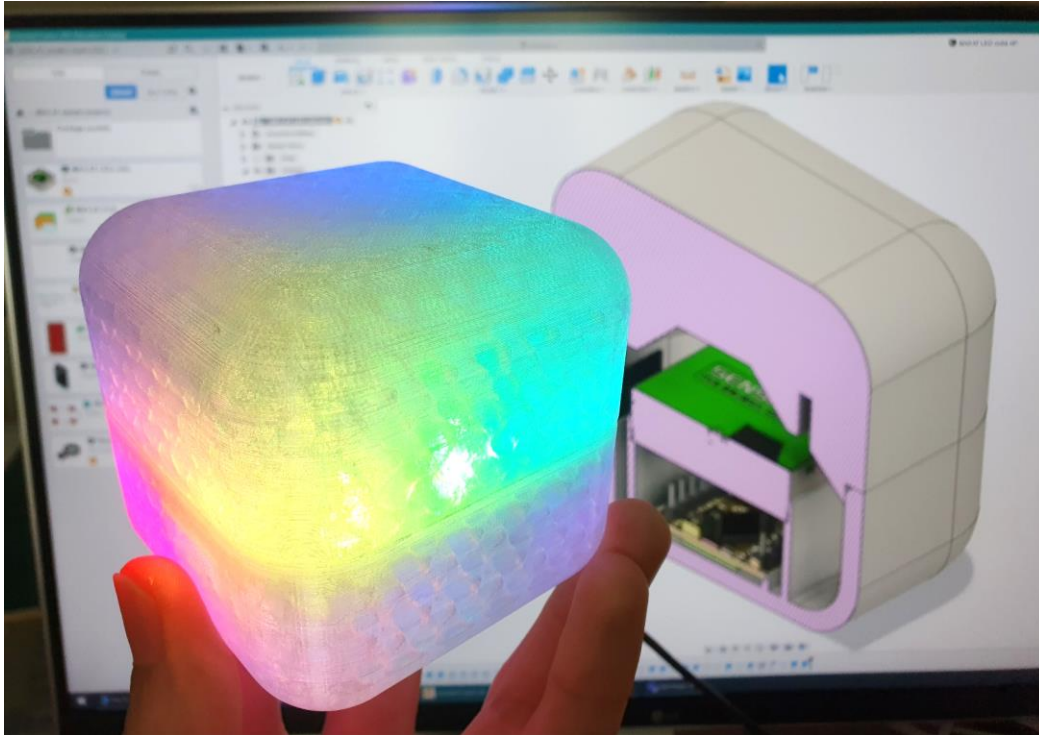
The base project consists of a microcontroller board with wifi, bluetooth and USB for power and connectivity, a Sensirion SCD40 CO_2 sensor and a strip of controllable color LEDs. All parts are placed on and connected by a custom PCB.

Each participant may choose to implement a close copy of this example project with minor modifications, or a major redesign of all aspects, or even a completely different project. Enclosure shape, sensor selection and display method / data output method can be chosen freely by the participants. A number of other features can be implemented with the example hardware or small component additions:

- Sending data to computer or internet (built in wifi, bluetooth or USB)
- Logging data for later analysis (built-in storage)
- Showing data of temperature and humidity on display (display)
- Showing data with a mechanical dial indicator (servo motor)
- LED control by computer or smartphone according to time of day, incoming emails or calendar status.
- Audible alarm tone or melody (buzzer or amplifier/speaker)
- Different modes, with change based on
    o Button presses (button)
    o Orientation of cube (accelerometer)
    o Tapping the cube (accelerometer)
- Addition of other air quality sensors
    o particle sensors
    o volatile organics sensor
    o NOx sensor
- Mobile use with a battery and charger

5. Available components:

Components printed fat are part of the basic example project, all others are optional add-ons. Many components on adafruit.com are also supported by CircuitPython and can be used as well.

- ESP32 microcontroller
    - On **Adafruit Qt Py** microcontroller board
    - Wifi & bluetooth
    - USB, SPI, I2C and many other peripheral connections
    - Digital & analog input and digital output
    - Touch sensing
- Sensors
    - **Sensirion SCD40** CO_2 sensor
    - Sensirion SGP40 volatile organics sensor
    - Sensirion SPS30 particle sensor
    - ST LSM303AGR 3-axis accelerometer and compass
    - ST LIS3DH 3-axis accelerometer
    - ST LPS22 pressure sensor
    - Vishay VCNL4040 light and distance sensor
    - ST lidar distance sensors
- **WS2812/SK6812** individually addressable RGB LEDs
- OLED display, monochrome
- LCD display, color
- Motors with continuous rotation and servos
- Sound generators like **vibration motors**, piezos, amp and speaker
- Battery and charger
- SD card slot
- Buttons, adjustment potentiometers, connectors etc.

5. Hardware and Software used

A large part of the project will be learning how to use different hardware and software tools, which are essential for realizing a complete prototype of a technical system. Care has been taken to select easy to use, but powerful and universally useful tools, which can be applied in several parts of the project, as well as afterwards. The availability of good documentation resources was also considered. All software is either commercial with a free license option or open source and should be installed on the participants computers (windows or macOS).

2.1. Fusion 360

Fusion 360 is a cloud-based CAD software suite, which has a good set of CAD features and a usable ECAD/PCB design workspace. Additionally, it offers CAM (computer aided manufacturing) with a multitude of different manufacturing techniques, such as CNC milling and 3D printing. This software will be used for all of the hardware design in this project.

With a paid or education license, advanced features such as simulation and generative design are also available, but for our project, the important aspect is the team collaboration function – all projects will be saved in the project workspace and can be viewed and edited as required. For CAD/ECAD, the personal version is sufficient and will not expire, so the learned skills can later be used for private projects.
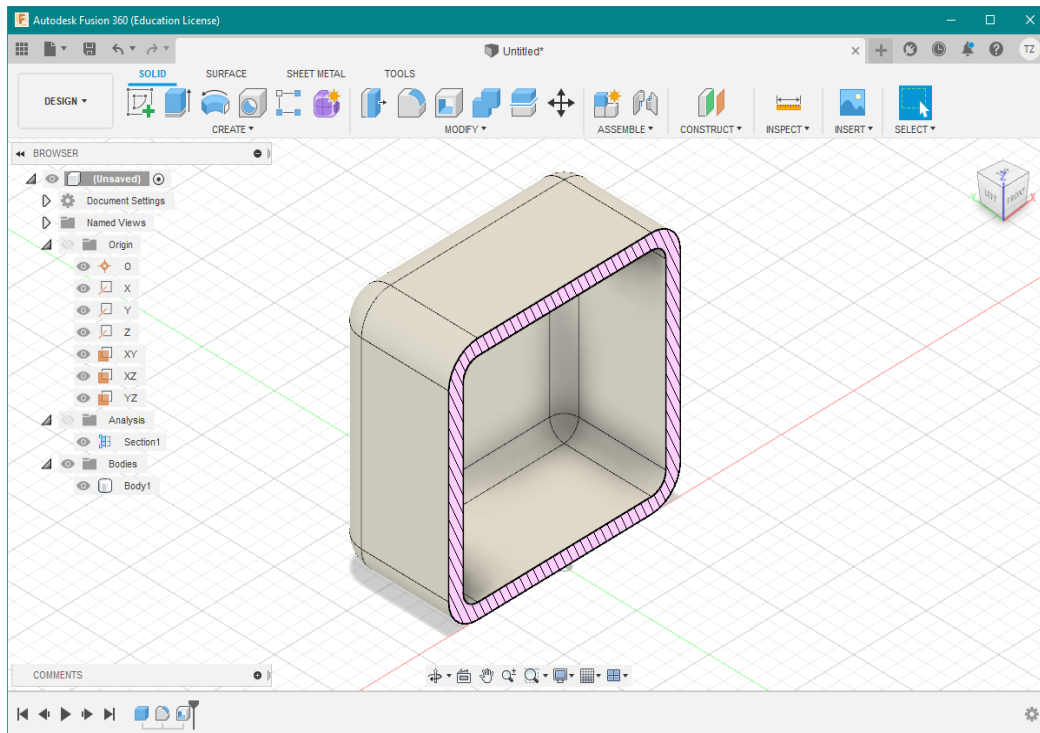
- [Education version](#)
  - Free while enrolled, registration and education verification required
  - During our course, we will use this for the team collaboration features

A large amount of training videos and exercises is available for Fusion 360, which give an overview of the program but also teach general mechanical and electrical design. The following resources are recommended to watch or keep in mind as a reference:

- [Getting started](#), videos 1-7 (highly recommended: 1-2)
  - Comprehensive introduction to the software
- [Getting started with modeling](#), videos 1-4 (highly recommended: 1-4)
  - Introduction to sketches and extrusions
- [Solid modeling basics](#), videos 1-16
  - Introduction of all tools for creating and modifying 3D bodies
- [Mechanical Assemblies](#), videos 1-7
  - Creating parts, assemblies and joints – more advanced 3D modelling
- [Additive manufacturing](#), videos 1-4, (highly recommended 1-4)
  - Quick repetition of the software and how to design for 3D printing
- [Electronics Fundamentals](#), videos 1-11
  - Basics of electrical circuits
- [ECAD](#), videos 1-5 (highly recommended: 1-5)
  - Introduction to PCB design and preparation for manufacturing

2.2. PrusaSlicer

While the preparation and translation of the 3D model (slicing) for the 3D printer can be done in Fusion, PrusaSlicer, the standard software of our printer, offers a better visualization and is easier to use. The slicing process takes only a few minutes, but it is a critical step for understanding the 3D printing process and getting the desired result. You do not necessarily have to get into the use of this software, but it is great for learning about the 3D printing process and visualizing and also controlling the created print.
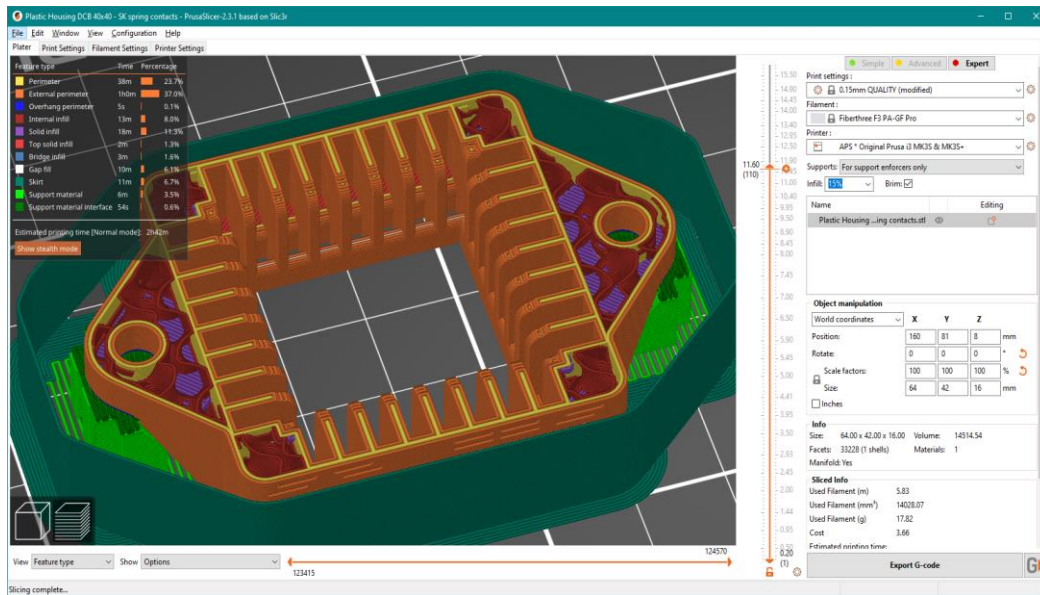
- [Download](#)
- [First print with PrusaSlicer](#)
  - Quick overview of the software and the most important parameters
- [Quick start guide](#)

2.3. Python (Mu IDE)

The microcontroller needs software programming to run all desired hardware functions, and if you want to receive and analyze data on a computer or smartphone, you need to write software for these too. We will use the interpreted language Python for both; for the microcontroller there is a special variant called CircuitPython, while the computer runs „normal" CPython (Python 3). Interpreted means no compilation is necessary, any command can be run immediately and the output will be presented. The editor Mu is well suited for beginners, and we will use it for both „normal" Python 3 on the computer and CircuitPython for embedded software. Any other IDE or text editor may be used instead if you have other preferences, for CircuitPython see [Recommended editors.](#)


- [Download](#)
- [Tutorials](#), especially:
  - ["Start here"](#)
  - For computer programming: ["Using Standard Python 3"](#)
  - For embedded programming: [Adafruit CircuitPython](#)
  - On circuitpython in general: [Adafruit CircuitPython](#)

2.4. Prusa i3 MK3S+ 3D printer

This open source 3D printer uses the common fused filament fabrication (FFF) technique, melting plastic delivered as a thin filament and depositing it where desired, creating arbitrary 3D shapes layer by layer. Most medium size (< 10 cm on each side) objects can be produced in a few hours. The build volume is 250 mm * 210 mm * 210 mm, the layer height 0.05 mm to 0.55 mm and the X/Y-resolution is 5 μm. Our standard material is PETG in black, white, red and transparent, other materials or colors can be discussed.
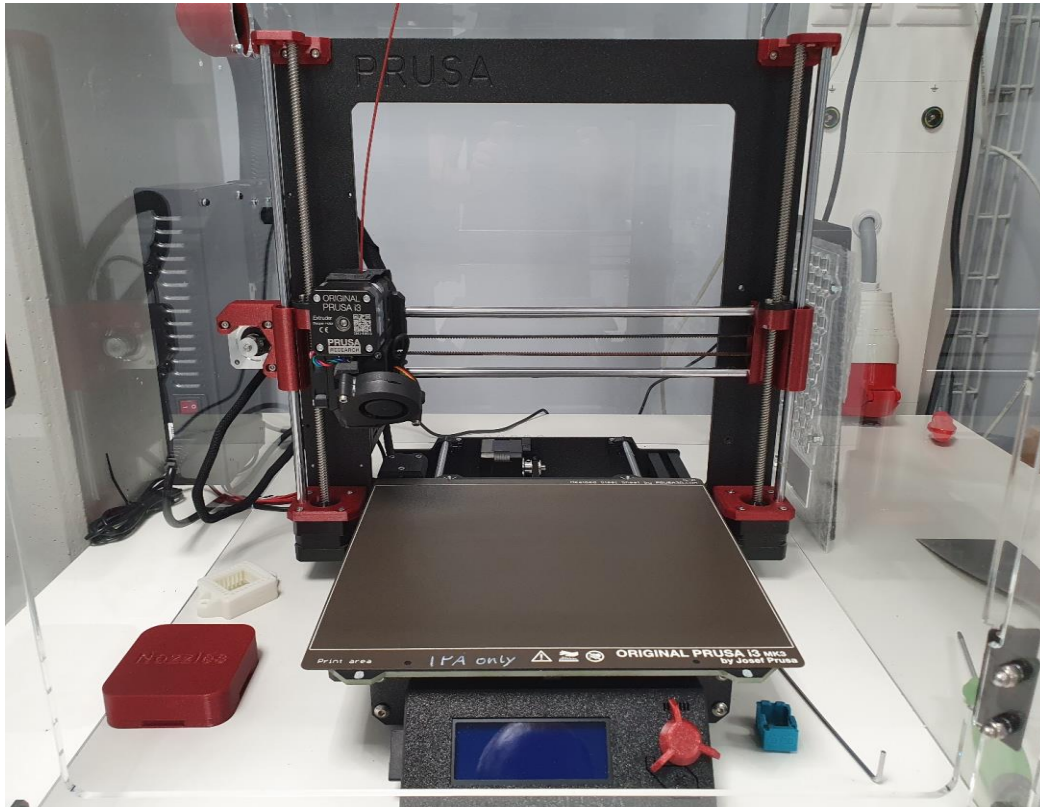
Prusa i3 MK3S+

2.5. Soldering equipment

We will use a stencil printer to apply soldering paste to the PCBs, place most components with the aid of a manual pick-and-place machine and then solder thr PCBs in a special oven. Any remaining components are then soldered by hand with a standard soldering iron, or with a hot air soldering tool.

2.6. Test & Measurement equipment

For testing the basic operation of the PCB, such as looking for short circuits and checking the supply voltage, a multimeter (measures constant voltage, current, resistance) is very useful. The correct function of the PCB and the developed software can then further be observed with an oscilloscope (measures voltage over time), which can show the control signals to LEDs or servos and bus communication with sensors.