Cheat Sheet
Embedded Systems

Tutorials

01 **SparkFun**
Online store and community for hardware projects, esp. for beginners with tutorials + a lot of parts one can use for fast prototyping

02 **adafruit**
Online store and DIY electronics community to discover projects by tech enthusiasts + many tutorials to get started & learn

Technologies

RISC-V 01
Open source meets processors: Design, share and use innovative Instruction Set Architechtures.

IoT 02
Connected smart devices: Get started with ESP32 (Wifi) or Raspberry Pi Zero W (Wifi & BLE) and nRF5340 (BLE, Zigbee, Thread)

Get started

01 **Hackaday**
Collaborative hardware development community, great to get inspiration about what is possible

02 **GitHub/GitLab**
Code sharing and code backup platform + repository of many (standard) libraries

Automation 01
Establish efficient workflow with processes, devices and rules

Get inspired

03 **Stack Overflow**
Largest online community for programmers to learn & share their knowledge

Tip **The Power of the Community**
Similar cases have most probably been done by someone already -> always look for existing solutions and use popular parts in case you need quick support

Get Hands-on

Prototyping 02
For more information about Hardware & Software Prototyping check out our previous Cheat Sheet

Software

Tip **Simulation Tools**
Use Spice, QEMU or MatLab for modelling your system

Development Environment

Tip **Debugging**
Build debugging information into your code, e.g. printing memory consumption or profiling code with timers

PlatformIO Tip
Choose your favourite IDE (Integrated Development Environment) or editor but we recommend using the PlatformIO core (CLI tool) at any time

01 **Git**
Cross-platform tool for version control (collaborative way of saving & correcting code versions)
Commonly used platforms are GitHub and GitLab

Code Quality

C / C++ 01
Most commonly used coding languages for embedded systems

MicroPython 02
Program microcontrollers with the ease of use of the Python language– but needs powerful microcontrollers

Rust 03
Multi-paradigm programming language focused on safety - syntactically similar to C++ but better memory safety and high performance

Languages

Hardware

01 **KiCad**
Open source software suite for EDA (Electronic Design Automation) and PCB (Printed Circuit Board) layout, increasingly popular

Design & Analysis

01 **PCBWay**
Chinese full feature custom PCB (Printed Circuit Boards) prototype service - from 1-10.000 pieces

02 **Aisler**
Upload your design files and get boards, parts, stencils or whole prototypes made in Germany - upscaling possible

Manufacturing

03 **adafruit Feather**
Complete line of development boards that are both standalone and stackable

01 **Raspberry Pi**
Open source, easy to use microcomputer platform, more powerful that microcontrollers, can run full operating systems - cheap hardware and big community

02 **Multimeter, Logic Analyzer, Oscilloscope**
Tools for electronic hardware debugging by measuring the output of ports

Protoyping Hardware

MOTIUS
WE R&D.