

Introduction to GitHub

A BASIC OVERVIEW

Overview

- Introduction to GitHub
- Usage of GitHub
- GitHub Collaboration features
- Best practices to use GitHub
- Documentation on GitHub
- Benefits for students using GitHub

What is GitHub?

- GitHub is a cloud-based hosting service for Git
- Stores Git repositories online
- Enables collaboration and sharing
- Provides backup and access from anywhere
- Popular in open-source and industry

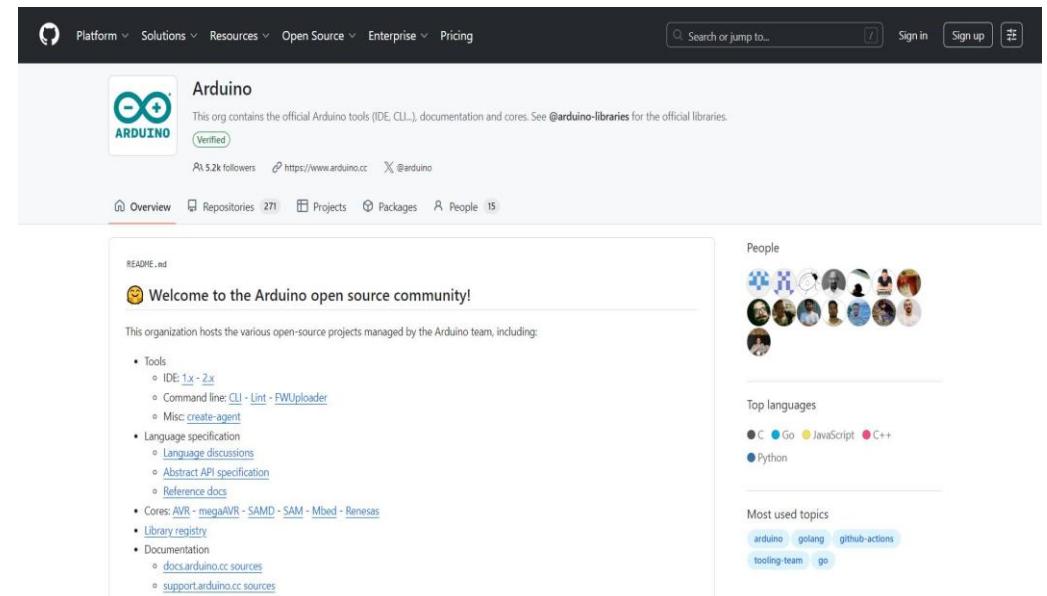


Fig: Arduino GitHub Repository

When to Use GitHub?

- Team projects
- Open-source contributions
- Project backup
- Code review
- Resume and portfolio building

GitHub Collaboration Features

- Pull Requests – propose changes
- Issues – track bugs and tasks
- Forks – personal copy of a project
- Actions – automation and CI/CD

The screenshot shows the GitHub interface for the repository `arduino / arduino-examples`. At the top, there are two search bars: one for pull requests (`is:pr is:open`) showing 14 open and 56 closed, and another for issues (`is:issue state:open`) showing 11 open and 9 closed. Below these are two lists of items:

- Pull Requests:**
 - Hoist functions in ArduinoISP sketch (topic: code, type: enhancement)
 - Add Nicla Sense ME to compilation checks
 - Add Portenta X8 to compilation checks (topic: infrastructure, type: enhancement)
 - Remove magic numbers from Tone() examples (topic: code, type: enhancement)
 - Change hardcoded Pin 13 to LED_BUILTIN
 - Fixes #33 (type: imperfection)
- Issues:**
 - [ACELLO4] Include the Portenta X8 and Nicla Sense ME as part of the CI workflow (topic: code, type: enhancement)
 - tonePitchFollower schematic doesn't match text description (topic: code, type: imperfection)
 - 08.Strings > StringLength has several problems
 - the content of the example file ReadASCIIString.ino feels odd to me.
 - 11.ArduinoISP/ArduinoISP.ino: with 'void pulse(int pin, int times)', it typically pulses `times + 1`, which
 - Problems with the Debounce tutorial

At the bottom, there are repository statistics: main branch (10 branches, 90 tags), a code search bar, a code dropdown menu, and an about link.

Fig: GitHub collaboration Features

GitHub Best Practices

- Commit frequently
- Write clear commit messages
- Pull before pushing
- Use branches for features
- Keep repositories organized

Creating Documentation Using GitHub

- Commonly used to host project documentation
- Documentation lives inside the repository
- Written in simple text formats (Markdown)
- Easy to update and collaborate on

The screenshot shows the GitHub organization page for the Arduino open source community. At the top, there's a header with a smiley face icon and the text "Welcome to the Arduino open source community!". Below it, a message states: "This organization hosts the various open-source projects managed by the Arduino team, including:". A large list of links follows, categorized into sections: Tools (IDE: 1.x - 2.x, Command line: CLI - Lint - FWUploader, Misc: create-agent), Language specification (Language discussions, Abstract API specification, Reference docs), Cores (AVR - megaAVR - SAMD - SAM - Mbed - Renesas), Library registry, Documentation (docs.arduino.cc sources, support.arduino.cc sources, Built-in examples), and GitHub Actions (arduino-lint-action - compile-sketches - report-size-deltas). At the bottom, a note says: "Maintaining these projects and handling community contributions is a hard job. Please support us by [buying original Arduino products](#)". Below that is a section titled "How you can contribute" with a list of items: Triage open issues, Submit fixes and implementations, Test open pull requests, Help others contribute, and Write documentation.

- Tools
 - IDE: [1.x](#) - [2.x](#)
 - Command line: [CLI](#) - [Lint](#) - [FWUploader](#)
 - Misc: [create-agent](#)
- Language specification
 - [Language discussions](#)
 - [Abstract API specification](#)
 - [Reference docs](#)
- Cores: [AVR](#) - [megaAVR](#) - [SAMD](#) - [SAM](#) - [Mbed](#) - [Renesas](#)
- [Library registry](#)
- Documentation
 - [docs.arduino.cc sources](#)
 - [support.arduino.cc sources](#)
 - [Built-in examples](#)
- GitHub Actions: [arduino-lint-action](#) - [compile-sketches](#) - [report-size-deltas](#)

Maintaining these projects and handling community contributions is a hard job. Please support us by [buying original Arduino products](#).

💡 How you can contribute

- **Triage** open issues: try to reproduce issues reported by other users and confirm whether you can experience them as well, or ask users for more details if needed. Spot duplicates. Improve descriptions. Help users who ask for support.
- **Submit** fixes and implementations: pick an open issue or feature request that you think you can implement yourself, and submit a pull request with an implementation.
- **Test** open pull requests: try to run the proposed modifications and report your success or failure. Testing on real hardware takes time and any help in this will speed up our responsiveness in merging contributions.
- **Help others** contribute by reviewing their code and suggesting good ways to implement fixes and features.
- **Write documentation** and improve the existing content.

Fig: Documentation in GitHub

What is Markdown?

- Lightweight formatting language
- Easy to read and write
- Uses simple symbols (#, *, -)
- Automatically rendered by GitHub

Common Documentation Files on GitHub

- README.md
 - Main project description
 - Explains what the project does
 - Shows how to install and use it
- CONTRIBUTING.md
 - Rules for contributing
- LICENSE
 - Legal usage terms
- docs/ folder
 - Detailed documentation files

GitHub Pages (Documentation Websites)

- GitHub can host documentation as a website
- Uses Markdown + static site generators
- Common tools: Jekyll, MkDocs
- Free hosting for public repositories

GitHub Benefits for Students

- Industry-relevant skill
- Teamwork experience
- Organized project history
- Easy project submission
- Public portfolio for internships/jobs

Thank You

Any Questions?