ARDUINO

Syllabus:

1. Introduction to Arduino

Brief history of the Arduino; open-source electronics prototyping.

2. Basic ideas

Basic ideas of Arduino, Familiarize the Arduino board, Setting up the arduino board. Installation of IDE in PC/ laptop for Arduino programming(Sketch)

3. Arduino Programming

(a) Program structure: data types, variables and constants, operators, control statements, loops, functions, string.

(b) Interfacing: serial communication, digital and analog input/output, getting input from sensors (e.g., temperature sensor, ultrasonic sensor etc)

ARDUINO

References:

1. https://www.arduino.cc/en/Tutorial/HomePage
2. Arduino Cookbook, Michael Margolis, O’Reilly Media (2011)
3. Getting Started with Arduino, Massimo Banzi, O’Reilly Media (2009)

Software:

Arduino IDE (Integrated Development Environment)

Hardware:

1. Arduino UNO
2. Basic electronic components

TERMINOLOGIES

A. Open-Source Software (OSS): Type of computer software in which source code is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software to anyone and for any purpose. Examples: Linux, Android, Firefox etc.

B. Open-Source hardware (OSH): Physical artifacts of technology designed and offered by the open-design movement. Information about the hardware is easily discerned so that others can make it – coupling it closely to the maker movement. Hardware design (i.e., mechanical drawings, schematics, bills of material, PCB layout data, HDL source code and integrated circuit layout data), in addition to the software that drives the hardware, are all released under free/libre terms. Examples: RepRap (3D printing), Arduino etc.

open source initiative®

open source hardware

TERMINOLOGIES

C. Microcontroller: An integrated circuit (IC) device used for controlling other portions of an electronic system, usually via a microprocessor unit (MPU), memory, and some peripherals. These devices are optimized for embedded applications that require both processing functionality and agile, responsive interaction with digital, analog, or electromechanical components. A typical microcontroller includes a processor (CPU), memory and input/output (I/O) peripherals on a single chip. MCUs are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices. They are essentially simple miniature personal computers (PCs) designed to control small features of a larger component, without a complex front-end operating system (OS).

https://internetofthingsagenda.techtarget.com/definition/microcontroller

What is Arduino?

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

https://en.wikipedia.org/wiki/Arduino#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators.

What is Arduino?

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the "Arduino language". In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

‘This little board has made it possible for people to do things they wouldn’t have done otherwise’
- David A. Mellis

https://en.wikipedia.org/wiki/Arduino#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators.

Arduino: History

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $50. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas. Casey Reas is known for co-creating, with Ben Fry, the Processing development platform. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller.

https://en.wikipedia.org/wiki/Arduino#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators.

Arduino: History

In 2005, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, extended Wiring by adding support for the cheaper ATmega8 microcontroller. The new project, forked from Wiring, was called Arduino.

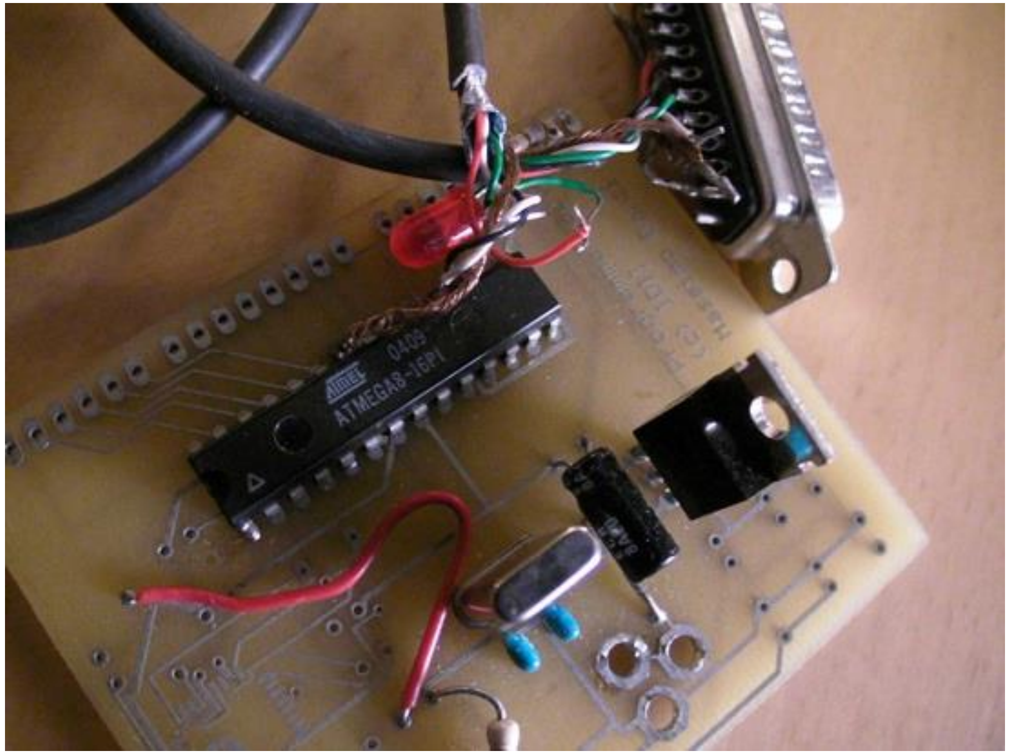
The initial Arduino core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis.

Following the completion of the platform, lighter and less expensive versions were distributed in the open-source community. It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced and in 2013 that 700,000 official boards were in users' hands.

https://en.wikipedia.org/wiki/Arduino#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators.

ARDUINO: History

David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi



https://www.circuitstoday.com/story-and-history-of-development-of-arduino

https://en.wikipedia.org/wiki/Arduino#:~:text=The%20Arduino%20project%20began%20in,environment%20using%20sensors%20and%20actuators.

Why Arduino?

- ❑ **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- ❑ **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

https://www.arduino.cc/en/Guide/Introduction

Why Arduino?

- ❑ **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.
- ❑ **Cross-platform -** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- ❑ **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.

https://www.arduino.cc/en/Guide/Introduction