



LED Control

Introduction

Microcontrollers are small, self-contained computers on a single integrated circuit. They were first introduced in the 1970s, and their popularity quickly grew due to their low cost, ease of use, and versatility. The Intel 8048 was one of the first microcontrollers to gain widespread adoption, and it was used in a variety of applications such as household appliances, automotive systems, and industrial control systems. Since then, micro-controllers have become an essential component in many electronic devices and systems, from toys and consumer electronics to medical devices and aerospace systems.

The Uno is based on the Atmel ATmega328P microcontroller and features a set of digital and analog input/output pins that can be used to connect to sensors, actuators, and other electronic components. The board also includes a USB interface for programming and communication with a computer, as well as a power regulator and other components to support the microcontroller. Since its introduction, the Arduino Uno has become a standard platform for hobbyists, educators, and professionals alike, and has been used in a wide range of applications, from robotics and automation to art and music projects. It features a set of digital and analog input/output pins, a USB interface for programming and communication, and a power regulator.

Common Instructions

During this peer learning program you will learn the basics about and work with Arduino, an open-source electronics platform.

This project is designed to provide you an immersive and hands-on learning experience that enables you to develop new skills and gain a deeper understanding of electronics and programming.

By collaborating closely with your peers to build a functional device you will get to understand how to use various electronic components and program the microcontroller while developing skills in problem-solving.

This project will serve as a general base on the essential concepts in Arduino. Your project should be written in the Arduino programming language - a variant of C++ with a specific syntax and libraries tailored for programming the Arduino boards.

The code must be written in the Arduino IDE.

Your code should have the expected output.

Submit your work to the assigned git repository by the specified deadline. You will need to solve this project with your team in a week, by the next Arduino peer learning session. You will then get to evaluate other teams on their projects and get your own project evaluated.

The electronics components needed to test the output of your code will be given to your team and they should be returned at the end of the program in the same condition you received them.

You can use a simulator to test your code's output.

Mandatory Part

You will be given the following components:

- A microcontroller (Arduino UNO)
- ON/OFF Switch
- 3 LEDs (Red, Green and yellow)
- 3 resistors 220 OHM
- A breadboard
- Jumper wires

A green LED has to be on until you press the switch button.

By pressing the button, the green LED has to turn off and the red and yellow lights should start blinking every second, at the same time.

Please use the electronics components safely:

- Power supply: Always use the correct voltage and current rated power supply for your Arduino board to avoid damaging the board or causing a fire.
- ESD protection: Electrostatic discharge (ESD) can damage electronic components, including the Arduino board. Avoid touching the pins and connectors of the board with your bare hands. Use an anti-static wrist strap or touch a grounded metal object before handling the board.
- Overheating: Do not overload the board with too much current, as this can cause the board to overheat and possibly catch fire. Use external power sources and avoid powering high current devices directly from the board.
- Short circuits: Avoid creating short circuits on the board by ensuring that there are no loose wires or connections touching each other. Use proper insulation and avoid exposing the board to moisture.
- Software safety: Always use reliable and tested code when programming the Arduino board. Do not run code that could potentially cause harm to people or property. Always use the appropriate safety precautions when testing and using your projects

Bonus Part

Now that you know how to make the red and yellow LEDs blink every second, your next tasks are:

- Make the red and yellow LEDs blink alternatively every two seconds
- Make the red LED blink every half second and the yellow LED every two seconds

Submission and Peer-Evaluation

Prepare a demonstration of the projects features, design, and performance.

Turn in your assignment in an open git repository of your choice. Only the code inside your repository will be evaluated during the defence. Be prepared to explain your code concisely and the methods you used to fine-tune the platform behavior.

All students on the group must be present at the evaluation.

The evaluation will take place on the next session, before receiving the next subject.

You will receive a badge on intra after successfully evaluating all the projects.