

TITLE AND DATE

Document name: *CMPE 311 Lab Report 2*

Document reference: *Hoover, Ian.cmpe311.fall25. project#2*

Date of publication: *11/11/25*

LEAD ENGINEER: *Ian Hoover, UMBC*

STAKEHOLDERS:

Prof Kidd, Instructor, University of Maryland Baltimore County, Baltimore, Maryland, USA

MD Safwan Zaman, TA, University of Maryland Baltimore County, Baltimore, Maryland, USA

HIGH-LEVEL DESCRIPTION: This project requires that an embedded system be built to allow two LEDs to blink independently. The blink intervals of both LEDs must be able to be independently set by the user. This project is implemented on the ELEGOO Uno R3, and input will be taken via the Arduino IDE serial port monitor. Software will be implemented in the Arduino IDE coding environment and compiled using gcc and submitted on Github.

DESCRIPTION: Given customer requirements, the technical requirements for the project/device will be derived. Additionally, testing requirements will be derived for the project to ensure that the device functions properly and all customer requirements are met. Once the technical and customer requirements have been derived, the device will be designed, built, and tested appropriately according to the above specifications.

REFERENCES: How to do Multitasking with the Arduino

The Arduino Handbook – Learn Microcontrollers for Embedded Systems

sec04b-engineeringProcess_pt1.pdf – Lecture on Engineering Process Part 1 of 2

sec04b-engineeringProcess_pt2.pdf – Lecture on Engineering Process Part 2 of 2

CMPE310 asynchronous Arduino lab

DEFINITIONS AND ABBREVIATIONS:

“The Customer-requirements” – The requirements defined by The Customer as satisfying the contract.

“The Requirements” – The System’s high-level technical requirements derived from the customer-requirements.

“Serial-monitor” – The serial port used by the Arduino IDE to communicate with The User.

Arduino – an Italian open-source hardware and software company; also refers to a development board created by the company
arduino.h – header for a library of convenience functions specific to the Arduino development platform
ELEGOO – A Chinese company that develops and markets 3D printers and accessories
IDE – Integrated Development Environment
gcc – front end for the GNU Compiler Collection
Github – A widely used distributed SVC (Software Version Control) system
LED – Light Emitting Diode

CONVENTIONS:

Must, shall or will – your design must satisfy the requirement
May – your design may satisfy the requirement but doesn't have to
Informative – the intent of the following description is to make the requirement more understandable
All customer requirements are started with "C.#".
All high-level requirements are started with "HL.#".
All testing/validation requirements are started with "T.#"

CUSTOMER REQUIREMENTS:

The customer requirements, C1 through C4, are identical to that of PROJECT-ASYNC.
Ref: kiddKid.cmpe311.fall25.project#1, CUSTOMER REQUIREMENTS section. See below.
C1. The User must be able to set the blink rate of two different LEDs.
C2. The User must be able to update the blink rate of each of the LEDs independently.
C3. The LED must blink at the set rate until The User tells the LED to blink at a different rate.
C4. The System must run upon an Arduino Uno R3 compatible development board.

HIGH-LEVEL TECHNICAL REQUIREMENTS:

HL.1 The System must use at least 2 LEDs
HL.2 The System must use a standard Arduino compatible development board (e.g. the provided ELEGOO Uno R3)
HL.3 The System must communicate with The User only via the Arduino IDE serial-monitor port
HL.4 Any use of the serial-monitor in HL.3 must be asynchronous and not affect the blinking of the LEDs.
HL.5 The User must be able to set the blink interval of the LEDs in msec
HL.6 The blink rate of each LED must be constant unless changed by The User

TESTING AND VALIDATION REQUIREMENTS:

- T.1 The system must be visually inspected to verify that exactly two LEDs are on the final product.
- T.2 The system must be inspected to ensure that the design was implemented on an ELEGOO Uno R3 and no other board.
- T.3 The system must be connected to appropriate hardware such as a laptop to ensure that it can be powered. Connect the device to multiple laptops to ensure it is compatible with various types of hardware.
- T.3.1 The system, once powered, must be compatible with the Arduino IDE coding environment. The environment must recognize the device once connected.
- T.4 Test the device with multiple inputs for each LED to ensure both LEDs blink, the LEDs are independent, and that the blink rate can be affected.
- T.4.1. Time the blink rates of each LED once input has been taken to ensure the rates are correct and are in msec.
- T.5 Visually verify that the LEDs are capable of blinking at two different rates.
- T.6 Measure the time between each blink of each LED to ensure that the blink rates do not change unless the user selects a new blink rate.

DESIGN:

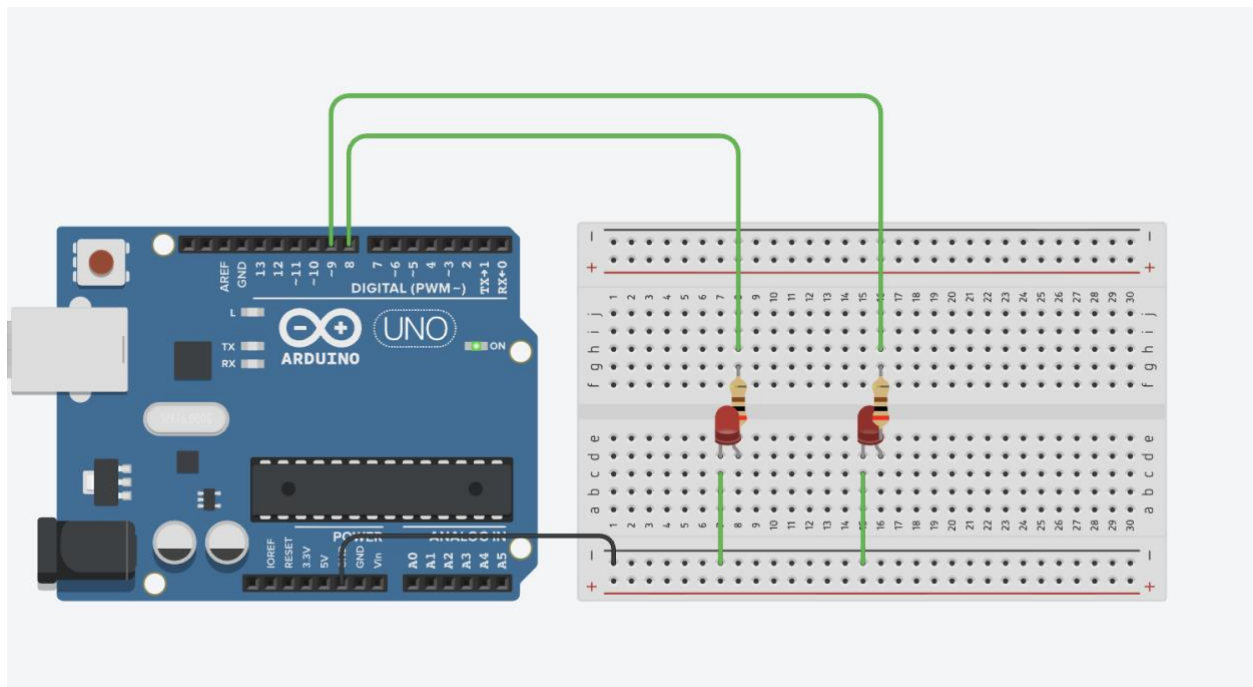


Figure 1: Schematic of the circuit used to implement the asynchronous blinking of two LEDs at user-input intervals.

The software used to implement the circuit above in Figure 1 is the Arduino IDE coding environment. The code is written in an Arduino dialect of C++. The equipment needed to implement this design is as follows: One ELEGOO Arduino R3, 2 LEDs, 2 200 ohm resistors, a breadboard, the Arduino IDE coding environment (to communicate with the user), a laptop, and a USB cable.

TESTING:

Serial Port I/O	Notes
What LED (1-2)? <i>1</i>	
What interval? (msec) <i>500</i>	LED 1 starts blinking at a rate of 500 msec (twice every second).
What LED (1-2)? <i>2</i>	
What interval? (msec) <i>1000</i>	LED 2 starts blinking at a rate of 1000 msec (once a second).
What LED (1-2)?	Waiting for the next LED and interval pair

Table 1: Dialog from testing the project implementation

Testing the implementation of the project design results in the dialog and observations seen above in Table 1. The project functions as intended and meets all customer requirements. Testing can also be seen below in the attached video. The testing requirements, T1-T6 are all met by visual inspection.

END OF DOCUMENT