

Alarm Clock

CSE 321 Project 2

Andrew Schick

Fall 2021

Table of Contents

Introduction.....	3
Specifications.....	3
Features.....	3
Applications.....	4
Block Diagram.....	4
Functionality Diagram.....	5
BOM.....	7
Schematic.....	8
Test Plan.....	9
Results.....	10
Recommendations for Improvement.....	10
BOM Index.....	11

Introduction

This document is the culmination of a four-week design process. This document outlines system specifications, design metrics, and test results of a user enabled alarm clock. The alarm clock utilizes both physical peripherals, a microcontroller, and software to set and display the clocks state. The alarm clock has dual functionality in that it can be used to count up to a target time or count down from a specified time. Once the target is reached the user will be alerted by LEDs.

Specifications

- Keypad inputs are debounced in the software to stop accidental triggers
- Port D, pins 4 through 7 are connected as inputs via jumpers to the columns of the keypad
- Port C, pins 8 through 11 are connected as outputs via jumpers to the rows of the keypad
- Power is alternated through the C pins to poll for keypad outputs from the columns
- Port A, pins 5 through 7 are connected via jumpers as outputs to grounded LEDs, each in series with a 1kOhm resistor.
- PA_5 blinks when a key is pressed, PA_6 and PA_7 are lit up when the target time is reached
- The LCD has four wires: ground is connected to ground, Vcc to 3.3V, SDA to the microcontrollers SDA pin, and SCL to the microcontrollers SCL pin
- The LCD can operate with up to 7V Vcc as per 1804 documentation
- Keypad constraints
 - 'D' is used to set the time
 - 'A' is used to begin the timer
 - 'B' is used to pause the timer
 - 'C' is used to change the counting direction
 - '0' through '9' are used as timer inputs

Features

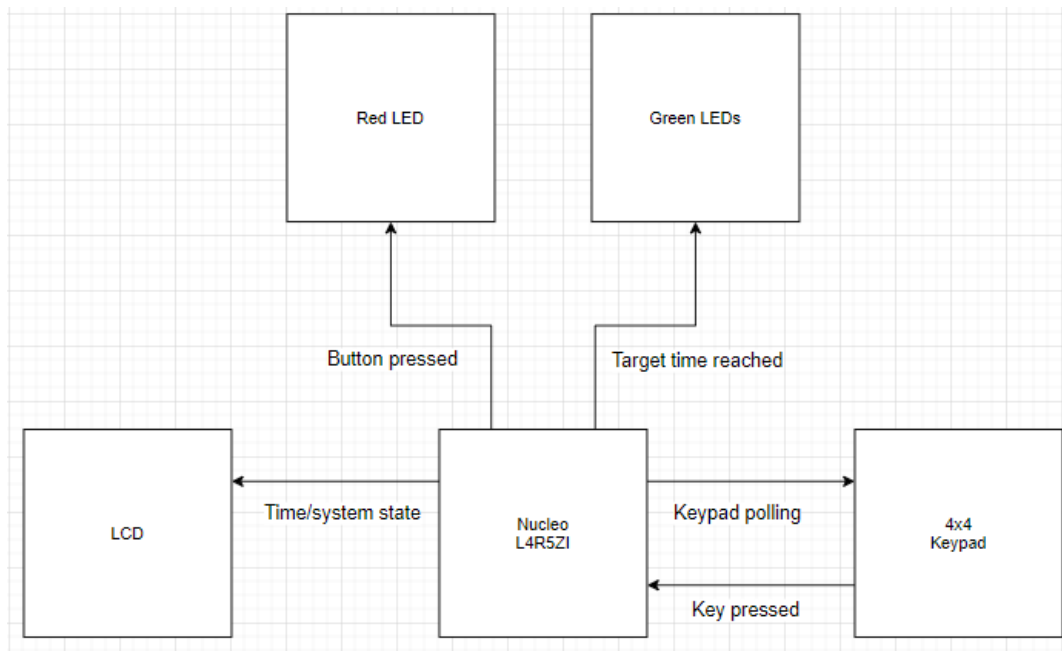
- Times can be set from 0:00 to 9:59 in terms of m: ss at any time, but cannot exceed 9:59
- Once a time is set, the user can use that saved time repeatedly until they change it. This even works when changing counting direction
- Dual counting functionality: count up or count down
- When switching between counting directions, a brief prompt will notify the user that the counting direction has been changed

- Pausing at any time while the clock is running will prompt the user of a pause and stop the clock
- LCD will continuously reflect the state of the clock
- Whilst counting up the LCD will display the time passed and the target time
- Whilst counting down the LCD will display the time remaining
- When the target time is reached in either direction there will be a notification prompt and two green LEDs will light up and stay on until a new timer is started
- Whenever a button is pressed a red LED will briefly blink
- For the 10s seconds place, the user will be blocked from entering a number greater than five and notified of the error

Applications

- Track timer
 - Counts up from the start of the race and pauses at the end of the race
- Microwave timer
 - Counts down from the specified time and pauses if the door opens
- Game clock
 - Counts down per period and pauses when a timeout is signaled

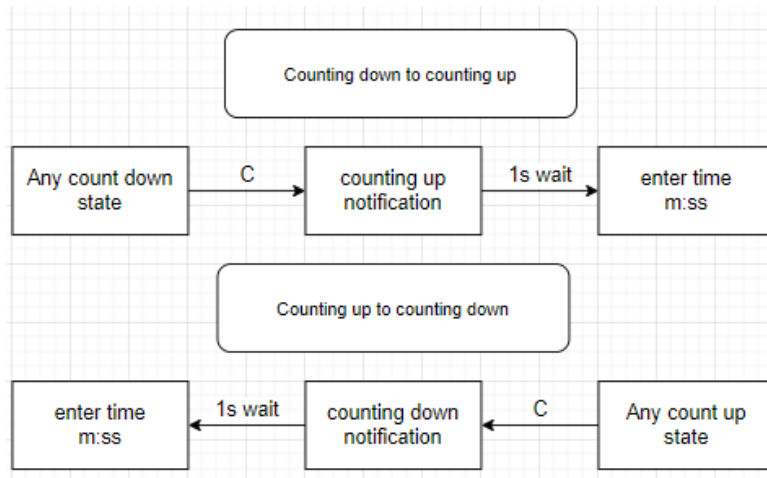
Block Diagram



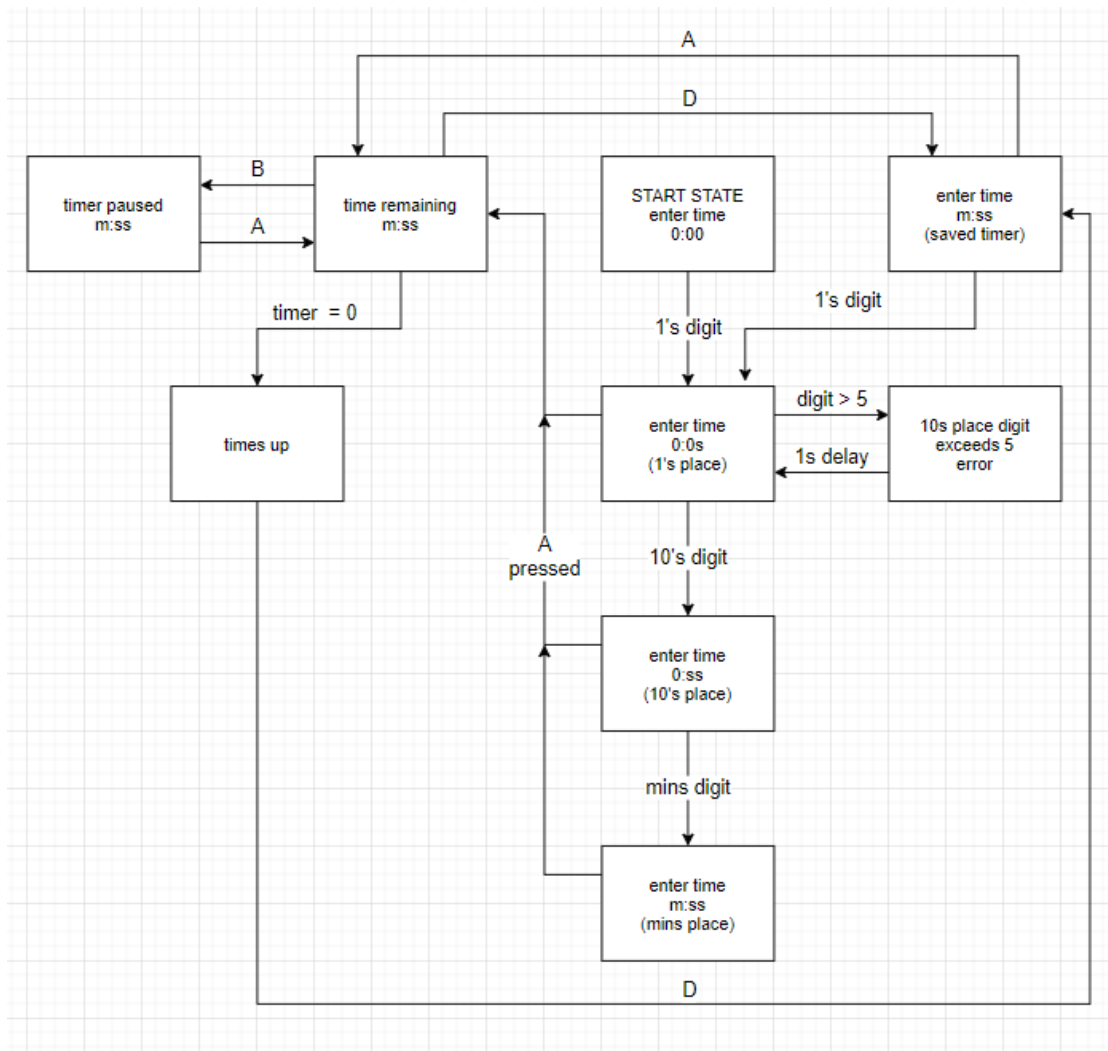
Functionality Diagram

The diagrams have been split between the counting directions to make functionality clearer.

Count directional transition

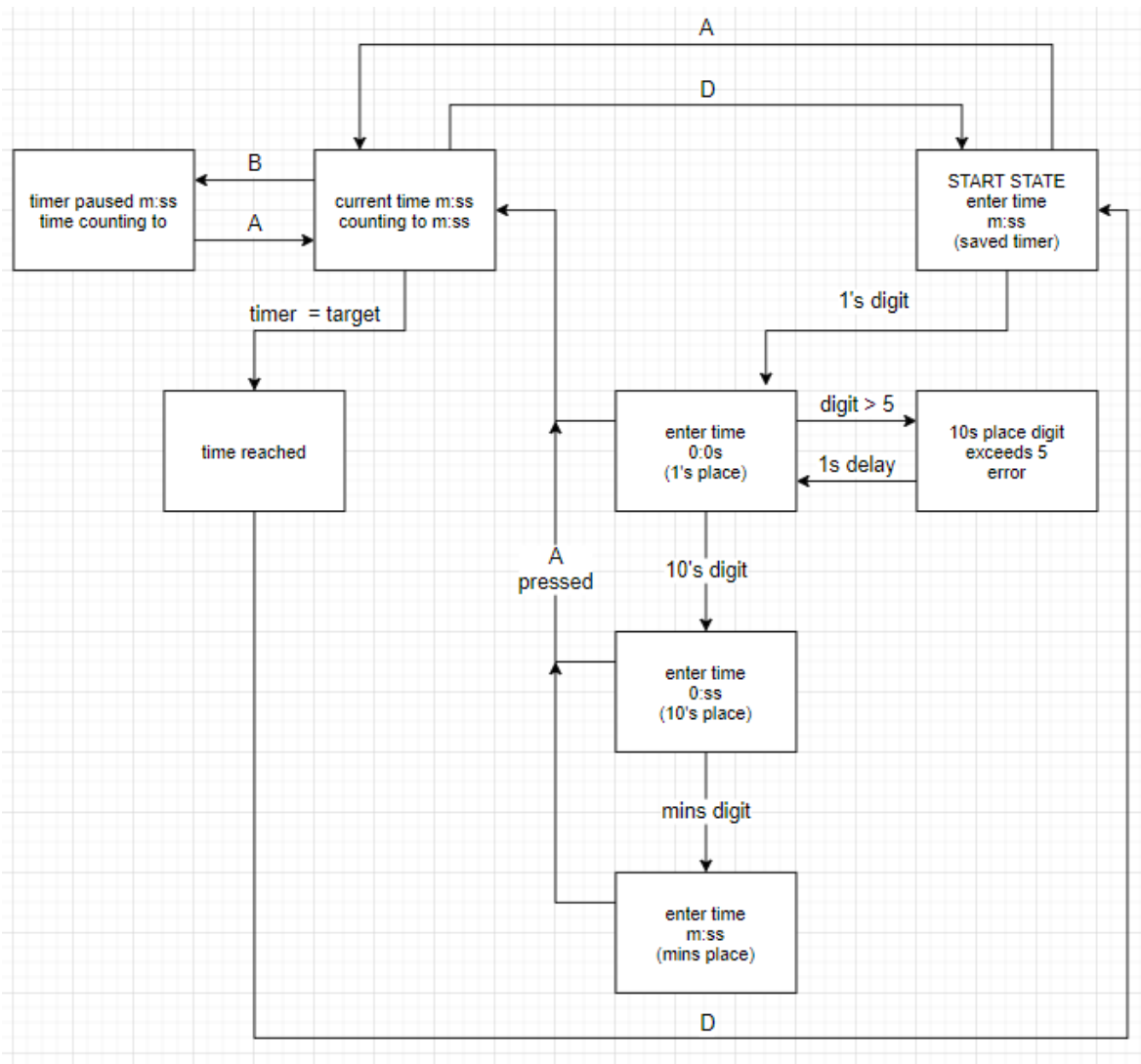


Count down



Note that the system will always boot up as a countdown timer that is why the start state has a special state. However, when the system switches to count up and back to count down, without being rebooted, the start state will change to the (saved timer) state for every swap afterwards.

Count up

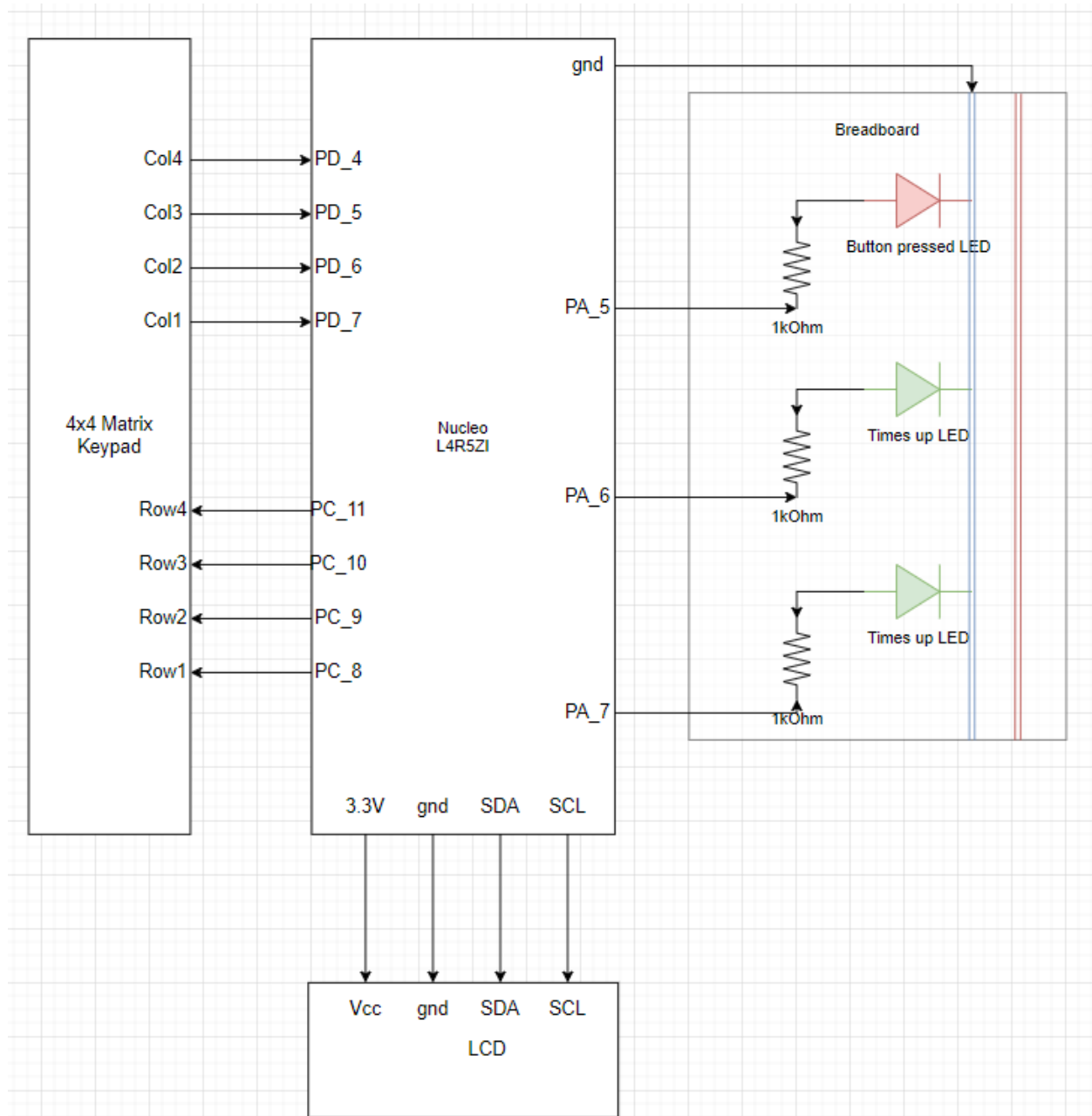


BOM

See BOM Index for digital links.

- At least 15 male to male jumper wires: keypad (8), LCD (4), LEDs (3)
- 3 LEDs, preferably 2 green and 1 red
- LCD screen, Mouser model 1804
- 3, 1kOhm resistors
- Breadboard
- Nucleo L4R5ZI microcontroller
- 4x4 Matrix style keypad
- Micro USB cable

Schematic



Test Plan

Testing will be done manually through keypad inputs. This test plan tests the LCD screens state after each keypad input. LED behavior can be tested manually by making sure the red LED blinks whenever a key is pressed and the Green LEDs only turn on when the target time is reached. Whenever the present state jumps in terms of time that just indicates that while testing, wait until the LCD reads the present state time. Lines highlighted in red indicate unintended behavior occurred during testing.

Present State	Keypad Input	Expected/Next State	Test case
Boot up, enter time	1	Enter time: m:s1	Adds to 1's
Enter time: m:s1	6	Number exceeds 5	Overflow exception thrown
Enter time: m:s1	5	Enter time: m:51	Add to 10's
Enter time: m:51	A	Time remaining: 0:51	M is 0 placeholder
Enter time: 0:29	B	Paused at: 0:29	Paused stops timer
Paused at: 0:29	B	Paused at: 0:29	Double pause does nothing
Paused at: 0:29	D	Enter time: m:51	Change time while paused
Enter time: m:51	A	Time remaining: 0:51	Time is saved
Time remaining: 0:09	D	Enter time: m:51	Reset while counting down
Enter time: m:51	2	Enter time: m:s2	Test green LEDs
Enter time: m:s2	A	Times up!	Test green LEDs
Times up!	D	Enter time: m:s2	Enter new timer
Enter time: m:s2	4	Enter time: m:s4	Enter new timer
Enter time: m:s4	0	Enter time: m:04	Enter new timer
Enter time: m:04	9	Enter time: 9:04	Enter max min
Enter time: 9:04	9	Enter time: 9:04	No overflow
Enter time: 9:04	B	Enter time: 9:04	Pause while setting
Enter time: 9:04	A	Time remaining: 9:04	Mins decreases
Time remaining: 8:59	C	Counting up	Change direction
Enter time: 9:04	A	Curr time: 0:00 Counts to: 9:04	Saved time carries over
Curr time: 1:20 Counts to: 9:04	B	Paused at: 1:20	Pause stops the timer
Paused at: 1:20	B	Paused at: 1:20	Double pause does nothing
Paused at: 1:20	D	Enter time: 9:04	Enter new time from pause
Enter time: 9:04	B	Enter time: 9:04	Pause doesn't
Enter time: 9:04	0	Enter time: m:s0	Times reached immediately
Enter time: m:s0	A	Time reached!	Times reached immediately
Time reached!	D	Enter time: m:s0	Set the max time
Enter time: m:s0	9	Enter time: m:s9	Set the max time
Enter time: m:s9	9	Number exceeds 5	Overflow exception thrown
Number exceeds 5	5	Enter time: m:59	Set the max time
Enter time: m:59	9	Enter time: 9:59	Set the max time

Enter time: 9:59	A	Curr time: 0:00 Counts to: 9:59	Counts to max time
Curr time: 0:00 Counts to: 9:59	C	Counting down	Switch back to count down
Enter time: 9:59	A	Time remaining: 9:59	Saved time carries back
Time remaining: 8:59	B	Time paused: 8:59	Check pause not altered
Time paused: 8:59	D	Enter time: 9:59	Check setting not altered
Enter time: m:59	6	Enter time: m:s6	Check setting not altered

Results

The test plan above was used to test the system and only the red lines above had unintended behavior. The results were flawless for the necessary implementation part in terms of LCD display, ignoring unnecessary inputs, throwing exceptions, count down time, and LED behavior. For the Expansion functionality there was only one unintended behavior which was when the user tries to pause while setting the time the prompt will change. This however doesn't affect the actual state of the timer and only the LCD display which meant the testing could go on and this is a non-critical bug. This behavior is also not an ideal use i.e. a user should never be trying to pause while entering a time.

Recommendations for Improvement

- Issue: If the user clicks 'b' while the mode is count up and a new time is currently being entered the prompt will be changed to the paused prompt. Note, this error is not referenced in the functionality diagrams as it is unintended behavior.
 - Solution: ignore the users request to pause, similar to the countdown functionality on unnecessary pauses.
- Issue: Altered an unnecessary mutable global variable within the interrupt functions. The variable 'digit' was altered whenever a digit keypad input was pressed and the user was in the setting time state. Have a lot of shared state within interrupt handlers can be dangerous.
 - Solution: Convert the digit from string form in the 'key' variable to its integer form and store it in the 'digit' variable.
- Issue: global variables, 'digit' and 'time_entered', don't need to be global. Main.cpp uses too many mutable global variables which can be dangerous, hard to debug, and create excessive code.
 - Solution: 'digit' and 'time_entered' can be local variables within main while passing them as arguments to the helper functions.

- Issue: efficiency/performance, when the timer is running, the code is constantly calculating the minutes place, 10s seconds place, and seconds place. This is very inefficient because the minutes place will normally only change every 60 seconds yet it is still calculated every iteration. Similarly, with the 10s seconds place, it will normally only change every 10 seconds, but it is still calculated in every iteration.
 - Solution: Ticker can be attached to periodically update each digit's place.

BOM Index

Male to male Jumper wires

https://www.amazon.com/Premium-Breadboard-Jumper-100-Pack-Hellotronics/dp/B07GJLH7V1/ref=sr_1_1_sspa?keywords=jumper+wires+male+to+male&qid=1636100151&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyNjYyQVZCU0pTUVITJmVuY3J5cHRlZElkPUEwMTIyNDI2NVZRVetHNU1QNUxOJmVuY3J5cHRlZEFkSWQ9QTAyNTg0ODhSTDNZSjFGMzUyODgmd2lkZ2V0TmFtZT1zcF9hdGYmYWN0aW9uPWNsaWNrUmVkaXJlY3QmZG9Ob3RMb2dDbGljaz10cnVl

LEDs

https://www.amazon.com/DiCUNO-450pcs-Colors-Emitting-Assorted/dp/B073QMYKDM/ref=sr_1_1?keywords=breadboard+leds&qid=1636100377&sr=8-1

Breadboard

https://www.amazon.com/4Pcs-MCIGICM-Point-Solderless-Breadboard/dp/B08115P2T4/ref=sr_1_1_sspa?keywords=breadboard&qid=1636100586&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEwNVMwWlI3R1NOMEo5JmVuY3J5cHRlZElkPUEwMzY3OTIwRFFHR1EyQ0cyMTZKJmVuY3J5cHRlZEFkSWQ9QTA0MzE5MjMyNzZaRktQVUE0Mkc2JndpZGdldE5hbWU9c3BfYXRmJmFjdGljbGlja1JlZGlyZW50JmRvTm90TG9nQ2xpY2s9dHJlZQ==

1 kOhm resistors

https://www.amazon.com/EDGELEC-Resistor-Tolerance-Multiple-Resistance/dp/B07QG1V4YL/ref=sr_1_6?crid=1LSULI7Z94P18&keywords=1kohm+resistor&qid=1636100611&sprefix=1+kohm+%2Caps%2C183&sr=8-6

Nucleo L4R5ZI

<https://www.st.com/en/evaluation-tools/nucleo-l4r5zi.html>

1804 LCD

<https://www.mouser.com/ProductDetail/713-104020111>

Micro USB cable

https://www.amazon.com/Android-JSAUX-Charger-Braided-Compatible/dp/B07H91JTCD/ref=sr_1_2_sspa?keywords=micro+usb+cable&qid=1636106508&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUE0UIVLWFVYOU5DNkUmZW5jcnlwdGVkSWQ9QTA1MDI2NDkzMjc1Qk9MMzI2UUNGJmVuY3J5cHRlZEFkSWQ9QTA2ODA1NjAxMUUpON1JQUkZYSUxEJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZW50JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==