Electrical, Electronic & Computer Engineering
School of Engineering & Physical Sciences

B31DG: Embedded Software
Assignment 1.

1. Problem

In this assignment, you will create the waveform in Fig. 1 using an ESP32 system.

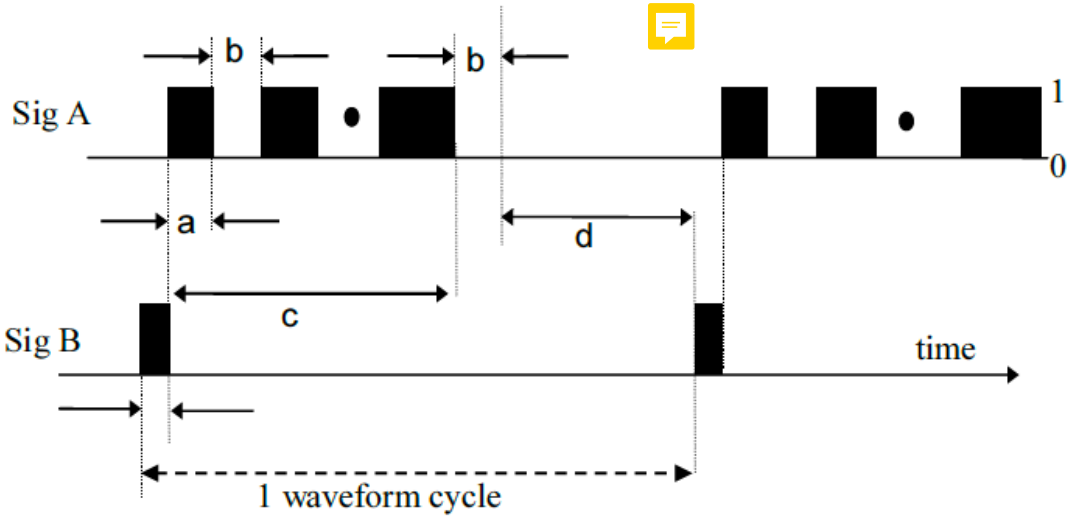


Figure 1

The waveform consists of a repeated set of cycles. Sig B should be 50µS.
For Grid labs. the signal from Sig B can be used to trigger an oscilloscope display.

There are two switch inputs to the system.

Switch Input	Purpose
Switch 1	0 = enable stream of pulses 1 = disable stream of pulses
Switch 2	0 = run as normal mode (above waveform) 1 = run with a modified waveform cycle

Note that changes to either switch should only take effect at the next waveform cycle.

The parameters for the above normal waveform are:

Parameter	Description
a	width of 1st pulse 2nd pulse = **a** + 50µS 3rd pulse = **a** + 100µS 4th pulse = **a** + 150µS etc
b	width of space between pulses (ALL spaces are the same)
c	Number of pulses in a block
d	space between pulse blocks

The possible system modes are as follows

Mode	Switch input 2 = '0'	Switch input 2 = '1'
1	Normal	Remove 3 pulses from pulse block (i.e. **c**-3 pulses in block) until switch 2 set back to 0.
2	Normal	Generate inverted form of complete Sig A waveform (from the largest pulse to the shortest) until switch 2 set back to 0
3	Normal	insert an extra 3 pulses into pulse block (i.e. **c**+3 pulses in block) until switch set back to 0
4	Normal	Half **d** and **b** time until switch set back to 0

2. Calculation of parameters

Each student will have a set of (a, b, c, d, and mode) parameters. To calculate your set of parameters use the following information.

Alphabet is numbered -

a=1, b=2, c=3, … m=13
n=13, o=12, p=11, … z=1

Using the first 5 letters of your surname (repeat last letter if less than 5) the parameters are calculated as follows :

Parameter	Example (James Herd)
a = first letter * 100µS	a = h * 100µS = 8 * 100µS = 800µS
b = second letter * 100µS	b = e * 100µS = 5 * 100µS = 500µS
c = third letter + 4	c = r + 4 = 9 + 4 = 13
d = fourth letter * 500µS	d = d * 500µS = 4 * 500µS = 2.0mS
mode = remainder (fifth letter/4) +1	mode = rem(d / 4) +1 = rem(4 / 4) +1 =1

Therefore 'James Herd' would be given the parameter set 800µS, 500µS, 13, 2.0mS and mode 1. The "rem" function is the integer division REMAINDER function, i.e. the remainder when the numerator is divided by the denominator. E.g. rem (10/3) = 1.

3. Task 1

Write a C/C++ program to implement the above problem. Show it running on the ESP32 system (this will depend on the nature of your lab: Grid, or simulation, see notes below). The code should be clean, good layout and well commented.

4. Task 2

At the test session, you will be asked to

- Run your code and show it meets the specification
- Show the commit/revision history of your code repository and include a screenshot in your report

5. Task 3

- Submit the following paperwork at the test time
    - fully documented listing of your program.
        - In clear readable form.
        - You need to submit the source files.
    - Nassi Shneiderman diagram of your code.
        - Use the tool at http://structorizer.fisch.lu/ or any other suitable program.
        - You will have to mount this on your own computer as it is not available on HW computers

6. Notes

- Use your own computers to be able to work home.
- **Students doing Labs in Grid:** You can demonstrate the code running on an ESP32 board using the oscilloscope. Timings should be within 5%. You can check your code at home by increasing times (multiply by 100 or 1000) to visualise SigA and SigB using LEDs.
- **Students doing simulation Labs:** You can demonstrate the code running on a simulated board (using Proteus on an Arduino board). Using the simulator, you can check your code by increasing times (multiply by 100 or 1000) to visualise SigA and SigB using the LEDs.
- Work as an individual.
- Use any of the I/O pins of the ESP32
- **Documented listing (very important) and Nassi-Shneiderman diagram** MUST be submitted in Week 5 on Canvas
- Testing will be during lab session on week 6,
- **Use a source code control system (e.g., Github).** To be shown on test day.
- Contributes 15% towards the 50% continuous assessment mark.

Electrical, Electronic & Computer Engineering
School of Engineering & Physical Sciences

Appendix: First use of ESP32 system

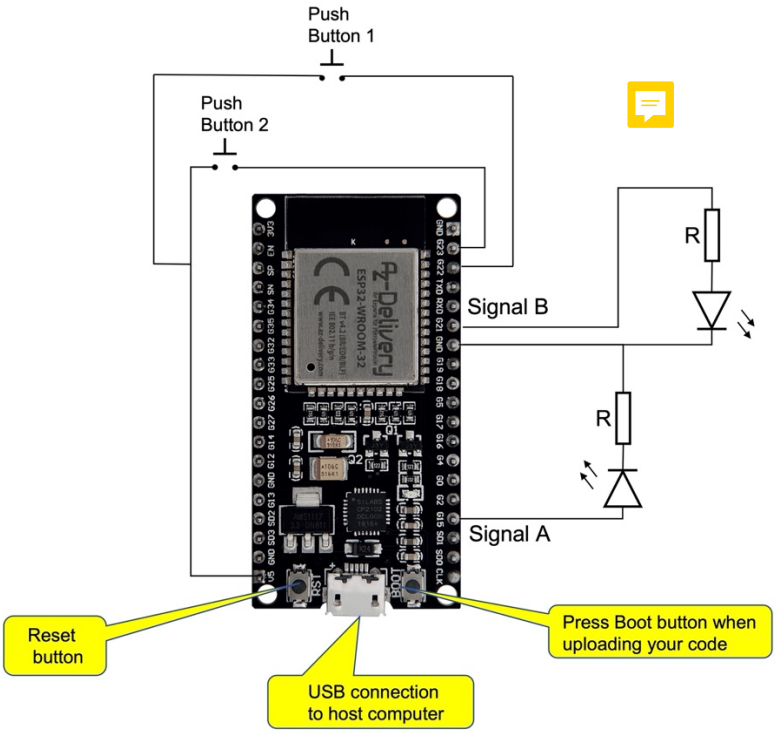
Step 1: Detecting ESP32 by your computer

-Install and run Arduino IDE (https://www.arduino.cc/en/software),
-Select the ESP32 board (DOIT ESP32 DEVKIT V1),
-Connect the ESP32 board and check that it is detected in Port.

See this tutorial to help with this step: https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-mac-and-linux-instructions/

Step 2: Build the electronic circuit

-Build the circuit using the provided equipment (ESP32, 2 switches, 2 LEDs and an oscilloscope to visualise Signals A and B)
-Ask a teaching assistant to check the circuit is correct



Step 3: Design and implement the code in Arduino IDE (see available examples)

The manual for the ESP32 board is on Canvas (the ESP32 pinout).