

# B31DG - Assignment 1

## Table of Contents

Revision History .....	1
Overview .....	1
Application Requirements .....	2
Control Inputs .....	2
Definition of Output Signal Timing .....	2
Calculation of Output Timing Parameters .....	3
Example Calculation – James Herd .....	3
Alternative DATA Output Behaviour.....	4
Development Environments .....	5
Deliverables.....	6
Appendix 1: Output Timing Diagram .....	7
Appendix 2: Hardware Configuration.....	8
Appendix 3: PDF Report.....	9

## Revision History

Revision	Date	Author	Notes
1.0	26/02/2024	Kieran O'Leary	Initial version.

## Overview

The objective of this assignment is to develop a bare-metal software application on an embedded system-on-chip (SoC). You will be required to develop a software application that will generate a pair of periodically repeating digital signals using an Espressif ESP32-based module.

The on/off state and nature of the periodic signals are to be controlled using two push buttons connected to the ESP32.

The application is to be first implemented using the Arduino IDE and then, to be reimplemented directly using the Espressif Internet of Things Development Framework (ESP-IDF).

An Integrated Development Environment (IDE) of your choice may be used with the ESP-IDF. Visual Studio Code (VS Code) with the ESP-IDF extension is a convenient and realistic development environment.

You will be required to create and use a Git repository (hosted on GitHub) for this assignment. One of deliverables for the assignment will be a tagged release of the Git repository.

# B31DG - Assignment 1

## Application Requirements

The application will produce two digital outputs from the ESP32, i.e. the DATA (Signal A) and SYNC (Signal B) signals.

The run-time operation of the system will be controlled using two push-button inputs to the ESP32, i.e. the OUTPUT\_ENABLE and OUTPUT\_SELECT signals.

At compile-time, a C pre-processor directive should be implemented to determine whether the application should be compiled with PRODUCTION timing, as specified below, or DEBUG timing, which slows down all timing parameters by a factor of 1000. This will permit visual debug of the application using LEDs alone. A conditional code-compile should be demonstrated.

## Control Inputs

The two push button inputs are momentary switches. Momentarily pressing either push button should result in a sustained state-change. The application should not require a push button to be continuously pressed. The application should always be aware of its operational state.

Input Signal	Description
OUTPUT_ENABLE	Connected to Push Button 1 (PB1), the OUTPUT_ENABLE input signal enables or disables the DATA (Signal A) output.
OUTPUT_SELECT	Connected to Push Button 2 (PB2), the OUTPUT_SELECT input signal controls whether the Normal or Alternative Waveform is output on the DATA (Signal A) output.

Table 1. Control Input Functionality

## Definition of Output Signal Timing

The timing diagram and table below describe the DATA and SYNC output signals in detail.

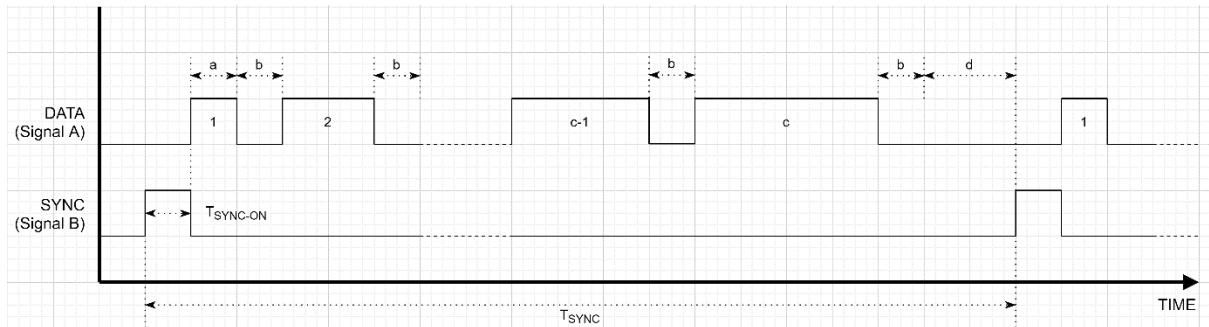


Figure 1. Output Timing Diagram (for Normal DATA Waveform)

Parameter	Description
a	Duration of the first pulse on-time ( $T_{\text{ON}(1)}$ ) in a DATA waveform cycle.
b	Duration of each pulse off-time in a DATA waveform cycle.
c	Number of pulses in a DATA waveform cycle.
d	Idle time between the end of the final pulse off-time and the start of the next SYNC pulse.
$T_{\text{SYNC-ON}}$	The SYNC pulse on-time duration. This is specified as <b>50us</b> (with PRODUCTION timing). The SYNC pulse period, i.e. $T_{\text{SYNC}}$ is dependent on the other parameters.

Table 2. Output Timing Parameter Descriptions (for Normal DATA Waveform)

# B31DG - Assignment 1

## Calculation of Output Timing Parameters

The first four letters of the developer's surname are used to calculate the Output Timing Parameters. Each of the first four letters in the surname is mapped to a number. If the surname consists of less than four letters, the final letter is repeated until four letters are available for mapping. The alphanumeric mapping and parameter definitions are shown in the tables below.

Letter	Letter	Numerical Mapping
a	z	1
b	y	2
c	x	3
d	w	4
e	v	5
f	u	6
g	t	7
h	s	8
i	r	9
j	q	10
k	p	11
l	o	12
m	n	13

Table 3. Alphanumeric Mapping

Parameter	Definition
a	First Letter Numerical Mapping x 100us
b	Second Letter Numerical Mapping x 100us
c	Third Letter Numerical Mapping + 4
d	Fourth Letter Numerical Mapping x 500us

Table 4. Output Timing Parameter Definitions (for Normal Waveform)

The on-time duration of the first pulse is defined by the parameter **a**. Subsequent pulses have an on-time that is determined by the following formula:

$$T_{ON(n)} = a + ((n-1) \times 50\mu s), \text{ where } 2 \leq n$$

## Example Calculation – James Herd

To illustrate how the Output Timing Parameters are calculated, an example is provided. The developer's name is assumed to be **James Herd**.

Parameter	Numerical Mapping (for James Herd)	Calculation (for James Herd)
a	“H” maps to 8	$8 \times 100\mu s = 800\mu s$
b	“e” maps to 5	$5 \times 100\mu s = 500\mu s$
c	“r” maps to 9	$9 + 4 = 13$
d	“d” maps to 4	$4 \times 500\mu s = 2000\mu s = 2\text{ms}$

Table 5. Output Timing Parameter Calculations (for James Herd)

# B31DG - Assignment 1

## Alternative DATA Output Behaviour

The Push Button 2 (PB2) input allows a user to change the behaviour of the DATA output signal. The OUTPUT\_SELECT signal, connected to PB2, allows the user to toggle between the Normal and Alternative DATA output behaviour.

There are four possible Alternative DATA output behaviours, only one of which will be implemented in the application. The four possible behaviours are described below.

Option	Description
1	Remove the final 3 pulses from each data waveform cycle (i.e. c-3 pulses in a data waveform cycle) until the Output Select push button is pressed again.
2	Generate a reversed form of the data waveform (from the largest pulse to the shortest) until the Output Select push button is pressed again.
3	Insert an extra 3 pulses into each data waveform cycle (i.e. c+3 pulses in a data waveform cycle) until the Output Select push button is pressed again.
4	Half the <b>b</b> and <b>d</b> time intervals until the Output Select push button is pressed again.

Table 6. Definition of Possible Alternative DATA Output Behaviours

In a similar manner to the calculations for the Output Timing Parameters, the Alternative DATA Output behaviour is dependent on the developer's surname. The behaviour is selected using the fifth letter of the developer's surname. If the surname consists of less than five letters, the final letter is repeated until a fifth letter is available.

The appropriate behaviour to use for a given surname is determined by the following formula:

$$\text{Option Number} = (\text{Fifth Letter Numerical Mapping} \% 4) + 1$$

To illustrate how the Alternative DATA Output behaviour is selected, an example is provided. The developer's name is assumed to be **James Herd**. As the surname only has four letters, the final letter is duplicated for the purposes of numerical mapping.

Numerical Mapping (for James Herd)	Alternative Behaviour Selection (for James Herd)
"d" maps to 4	$(4 \% 4) + 1 = 1$ , therefore Option 1 is selected

Table 7. Alternative DATA Output Behaviour Selection (for James Herd)

# B31DG - Assignment 1

## Development Environments

The above application must be implemented in two different software development environments:

1. The Arduino IDE
  - Support for the ESP32-based hardware must be added through the Arduino Board Manager.
2. Any IDE that allows you to directly utilise the **Espressif Internet of Things Development Framework (ESP-IDF)**.
  - Visual Studio Code (VS Code) with the ESP-IDF Extension installed is an appropriate solution.

# B31DG - Assignment 1

## Deliverables

The following deliverables must be submitted for this assignment:

1. A zip file, generated by creating a tagged release, of your git repository within GitHub.
  1. The git repository must include your source code for both development environments, i.e. an Arduino .ino file as well as a collection of C source and header files.
  2. The repository should be well-structured.
    - i. src, bin, docs, and tools folders may be expected (though not required).
    - ii. A .gitignore file should be considered to exclude build files.
    - iii. A readme.md file should be considered to explain the repository structure.
2. A short video showing your hardware setup running your application.
3. A PDF report that includes the following:
  1. Calculation of your application parameters.
  2. Oscilloscope Screen Captures:
    - i. Show both DATA and SYNC digital output signals.
    - ii. Oscilloscope to be triggered on Sync signal.
    - iii. Horizontal timebase selected to show two Sync Pulses in image.
  3. Images of Hardware Circuit including ESP32 PCB, Push Buttons and LEDs.
  4. Flowchart showing application control flow.



# B31DG - Assignment 1

## Appendix 2: Hardware Configuration

The assignment can be completed using either of the available ESP32-based development boards:

1. ESP32 NodeMCU Module
2. ESP32-C3-DevKitC-02

A simplified circuit diagram is shown below. The circuit is to be built using the provided ESP32-based development board and discrete components, i.e. push buttons, LEDs, and resistors.

The two pull-down resistors connected to the push buttons can be implemented using external resistors or resistors internal to the ESP32, if the GPIO pins on the ESP32 are appropriately configured.

An appropriate resistance value for each of the four resistors should be determined and stated in the assignment deliverables.

Note that the push buttons are connected to a 3.3V supply rail (rather than a 5V supply rail) as the operating voltage of the ESP32 is 3.3V.

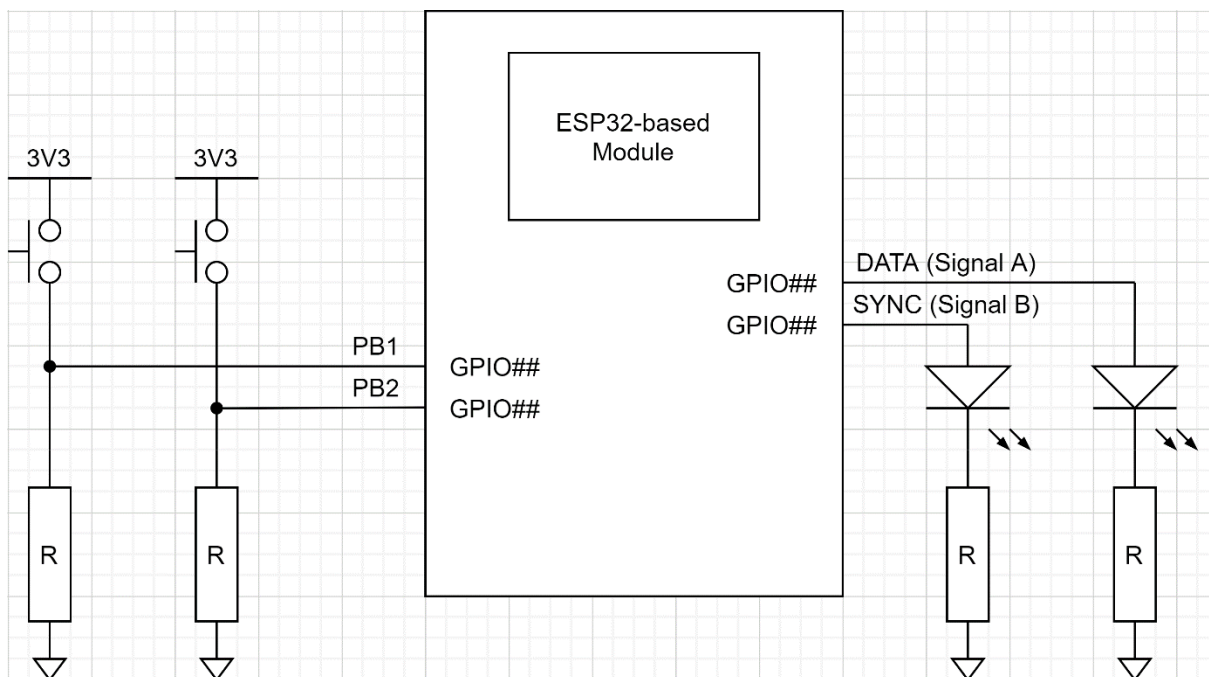


Figure 3. Simplified Circuit Diagram



# B31DG - Assignment 1

## Appendix 3: PDF Report

The deliverables for the assignment include a PDF report.

The PDF document is required to meet the following standard:

1. The PDF file should be named according to the following syntax:  
**B31DG - Assignment 1 - <Student Number> - <Student Name>**
2. The first page of the document should have a **Contact Details** section.
  - This should include your Name, Student Number and Email Address.
3. The first page of the document should contain a **Revision History** table.
  - Even if an engineering document only has a single revision, it is good practice to have a Revision History table.
  - The Revision History table should include Date, Author Name, Version Number and Notes columns.
    - The Notes entry for each document revision should include a brief description of the purpose of the revision and any changes from the previous version, e.g. "Initial draft", "Updated formatting", etc.
4. The document should include a **Table of Contents**.
  - It is good practice for an engineering document to include a Table of Contents.
  - One can be auto-generated from Headings in Microsoft Word.
    - Do not manually generate the Table of Contents. This is time-consuming and will inevitably become outdated.
5. Each section should begin on a new page and include an identifying Heading.
  - **Title Case** should be used for all Headings.
6. All figures should have a **Caption**, including a number.
  - Again, this is best practice and conveys professionalism.
  - Captions and numbers can be auto-generated in Microsoft Word.
    - Do not manually generate the caption numbers. This is time-consuming and will inevitably become outdated.
    - An example caption is *Figure 2. Output Timing Diagram*
7. The **Footer** of each page should contain the **Page Number**.
  - The **Date** and/or **Document Revision** may also be included, if desired.
  - Do not manually generate these parameters. This is time-consuming and they will inevitably become outdated.
    - Use Microsoft Word (or other editor of your choice) to auto-generate them.
8. Spelling and grammar must be checked prior to submission.
  - Use the tools available within Microsoft Word (or other editor of your choice) to assist you with this.
    - These tools are particularly helpful if you find this task to be difficult due to a language barrier, dyslexia, etc.
  - Failure to check for spelling and grammatical errors can be interpreted as a lack of attention to detail and respect for the reader.
  - UK English rather than US English must be utilised.