

# EE2026: DIGITAL DESIGN

## Academic Year 2021-2022, Semester 2

### FPGA Design Project: Non-Entertainment Related Digital System

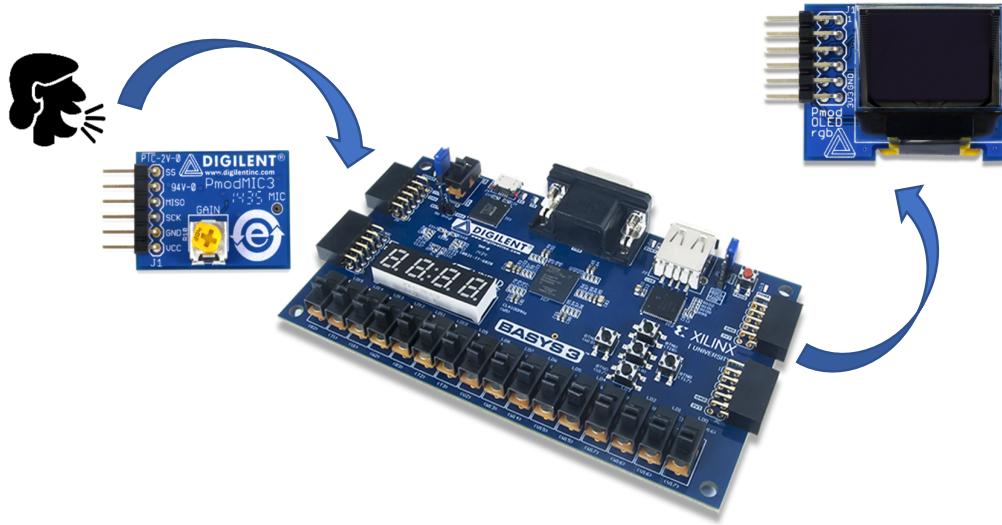
#### ABSTRACT

Using the fundamental technical skills obtained from the previous lab sessions, you will be creating a Non-Entertainment Related Digital System (NERDS). Two types of additional devices will be provided to you for the system:

- **MEMs microphone:** This analog-to-digital device will capture audio signals from the surroundings and provides the data to you in a digital format.
- **OLED RGB Display Screen:** Information can be displayed on this 96x64 pixels display screen with 16-bit colour resolution.

*You will need to replace the devices at your own cost if damaged due to negligence.  
Deadline extensions are not given for damaged components. Use them with upmost care!*

This manual introduces to you some concepts involved, and guides (NOT handheld!) you through getting a basic NERDS up and running. You will need to use your own logic, creativity, and resourcefulness to enhance the NERDS.



## 1. PROJECT OVERVIEW

NERDS is a pair work project that includes both **individual** and **teamwork** components.

Some milestones that you are requested to achieve for the NERDS will be detailed in **Section 4** of this lab manual. To make your system more functional, user-friendly, and unique, your team will also need to work on features of your own choosing, but which **MUST NOT** be leaning towards entertainment!

## 2. SCHEDULE AND PROJECT COMPONENTS

Week	Tasks	Student A	Student B	Tasks' Evaluation
7	<b>Teamwork</b> <b>Basic effects:</b> Audio signal on an OLED	Team Grading		<b>Evaluated at the start of week 8 lab session.</b> This will <b>not</b> be evaluated again in week 12, 13
8	<b>Student A (Task OTA)</b> <b>OLED task A:</b> Borders  <b>Student B (Task OTB)</b> <b>OLED task B:</b> Bars	Individual Grading	Individual Grading	<b>Evaluated in week 12, 13 as a complete project</b>
	<b>Teamwork (Task AVI)</b> <b>Audio volume indicator:</b> LEDs and segments	Team Grading		
9, 10	<b>Student A</b> <b>Personal Improvement feature(s)</b>  <b>Student B</b> <b>Personal Improvement feature(s)</b>	Individual Grading	Individual Grading	<i>* Note that there is a graded in-lab Verilog evaluation in week 9. It covers all Verilog programming up to the point of the actual evaluation</i>
	<b>Teamwork:</b> <ul style="list-style-type: none"><li>• <b>Team Improvement feature(s)</b></li><li>• <b>System Integration:</b> Combine all tasks and improvements into one bitstream</li></ul>	Team Grading		
	No lab session	-	-	Project archive and Report submission at the end of week 11 / Start of week 12
11	Project Assessment	Individual Q&A (Execution and understanding)	Individual Q&A (Execution and understanding)	The complete project ( <b>ONE bitstream only</b> ), supported by the report, is evaluated

### PLAGIARISM WARNING

This is a teamwork project. Discussions are encouraged. However, 'discussion' is not a valid excuse if your codes turn out to be uncomfortably similar. NUS and the EE2026 teaching team take plagiarism very seriously.

- Warning from the NUS Code of Student Conduct:  
"Any student found to have committed or aided and abetted the offence of plagiarism may be subject to disciplinary action" <https://www.nus.edu.sg/celc/programmes/plagiarism.html>
- Both the source and recipient of the project solutions (codes) or reports will be **unconditionally penalised** in such cases. Marks will also be adjusted and withheld from release if the codes are similar beyond a certain empirical threshold.
- Students will also be reported to BoD (Board of Discipline).

### 3. HARDWARE AND SOFTWARE RESOURCES

#### HARDWARE COMPONENTS:

- Each student signs out **1 PmodMIC3** and **1 Pmod OLEDrgb** from the Digital Electronics Lab from week 7 onwards
- Each student has already been assigned **1 Basys 3** Development Board at the beginning of the semester

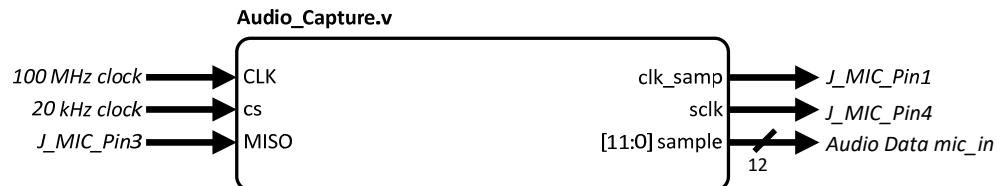
#### SOFTWARE FILES:

A project template (**SoundDisplay.xpr.zip**) can be downloaded from LumiNUS. The template consists of the following:

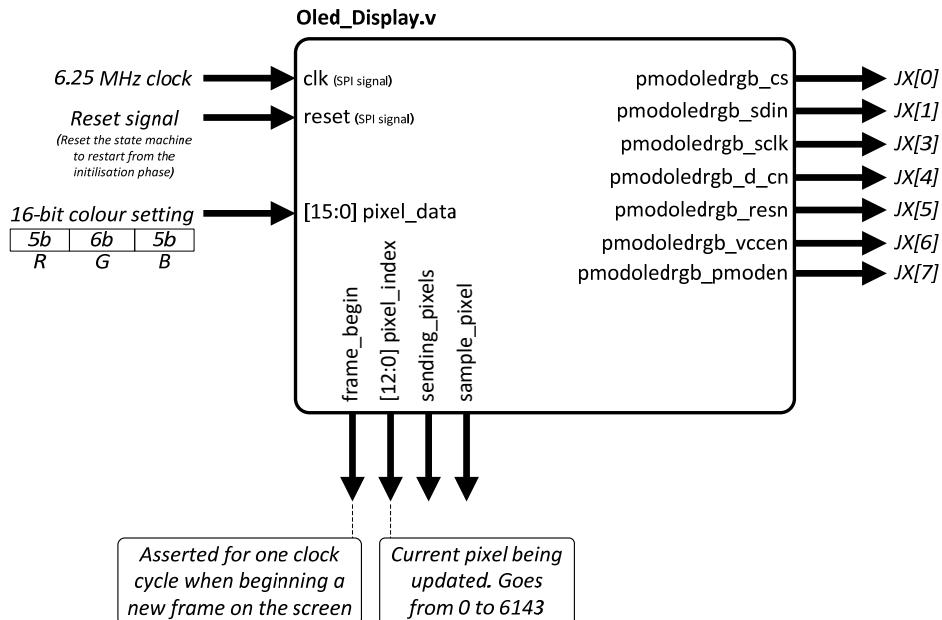
##### Design Sources:

**Top\_Student.v:** The top-module of the design. This will be your main module, or typically called the Top Level module, where you instantiate the sub-modules and make the necessary links between these sub modules. **You will need to modify this module, as well as create other design sources for use in this module.**

**Audio\_Capture.v:** An interface module between the microphone pmod device and your design on the FPGA. This module works with the *PmodMIC3* to convert the serial data input into a 12-bit parallel *mic\_in* sample data. The conversion needs a sampling clock and a serial clock. **You are NOT required to make changes to this module.**



**Oled\_Display.v:** An interface module between your design on the FPGA and the OLED display. This module works to send serial SPI data into the OLED display for initialization and drawing. **You are NOT required to make changes to this module.**



##### Constraint Sources:

**Basys3\_Master.xdc:** A master constraints file that defines the I/O constraints for the Basys 3 Development Board

#### MODULE WIKI:

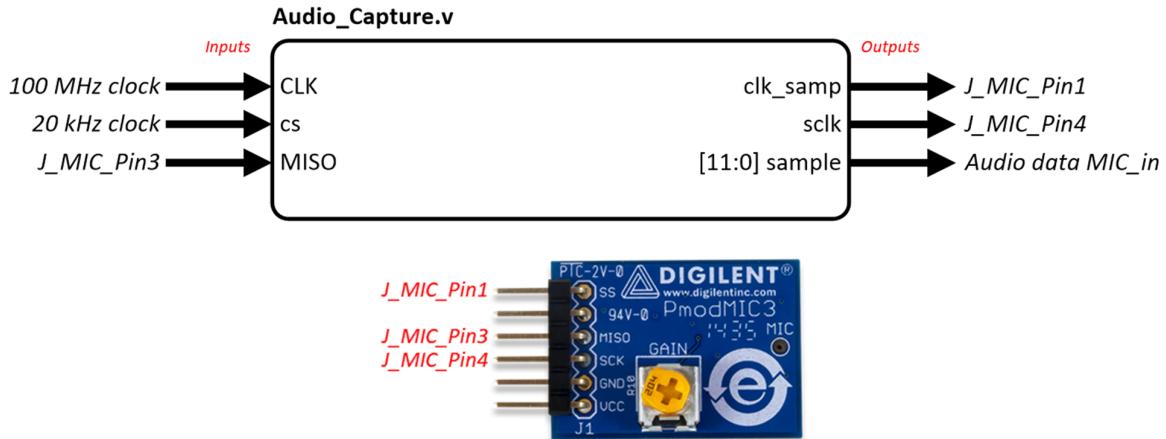
[tiny.cc/ee2026wiki](http://tiny.cc/ee2026wiki) : The project wiki includes a Verilog error troubleshooting guide, VHDL vs verilog code comparison and useful resources when you are working on additional features.

## 4. TASKS AND REQUIREMENTS

### 4.1 BASIC EFFECTS: AUDIO SIGNAL ON AN OLED

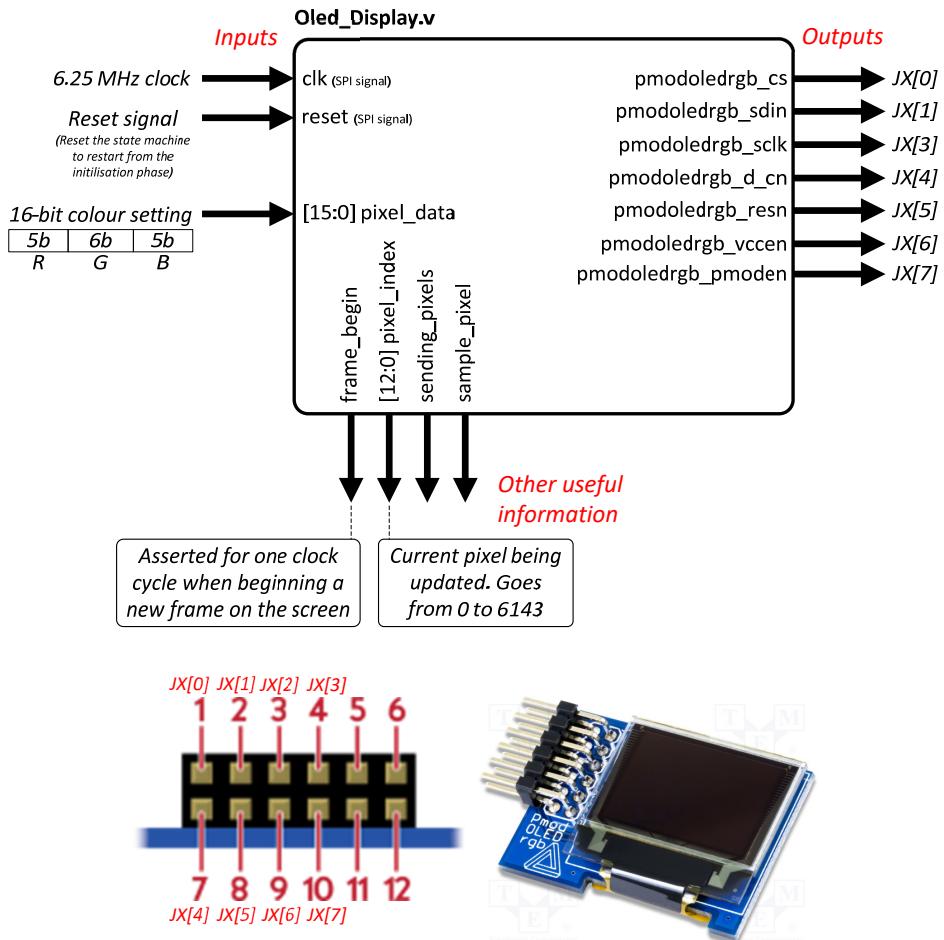
**OBJECTIVE:** To set up the NERDS, by interfacing the Basys 3 Development Board with the PmodMIC3 and the PmodOLED. Your system will capture and digitize the audio signal input from the microphone on the PmodMIC3. The data will then be processed by the FPGA, for display on the PmodOLED

Setting up the microphone by using `Audio_Capture.v`



- AC1. Instantiate the `Audio_Capture.v` module in `Top_Student.v`
- AC2. Create and provide the 100MHz and 20kHz (clk20k) clock signals to `Audio_Capture.v`  
The 20kHz clock is used as the sampling clock for capturing the audio signal. At every rising edge of this 20kHz clock signal, the audio signal from the microphone is read and stored in a 12-bit register named `sample`. This means that a 1 second audio signal will need to be represented by 20,000 discrete samples.
- AC3. Name the microphone data you receive from `Audio_Capture.v`, `mic_in`, and display this on 12 LEDs  
`mic_in` is the 12-bit data that is being read by the microphone. By displaying the microphone input on the LEDs, a quick visual observation can indicate if the microphone is properly connected.
- AC4. Write the codes that uses `sw[0]` to select between one of these two possibilities:
  - `sw[0]` is OFF: The 12-bit `mic_in` is represented on LEDs that updates at a clock speed of 20 kHz
  - `sw[0]` is ON: The 12-bit `mic_in` is represented on LEDs that updates at a clock speed of 10 Hz
- AC5. Connect the rest of the signals (`J_MIC_Pin1/3/4`) to the PmodMIC3 device accordingly.  
These are necessary input and output signals required for the microphone to function correctly.
- AC6. Attach the PmodMIC3 with FPGA. Update the constraints file according to the chosen Pmod Ports  
**Very important:** To avoid damaging the boards, make sure the GND and VCC pins on the Pmod and Basys are connected correspondingly
- AC7. Create and provide the 100MHz and 20kHz (clk20k) clock signals to `Audio_Capture.v`  
If the above steps are executed correctly, you would observe 12 LEDs flicker at a very high speed when `sw[0]` is OFF, and at a more observable speed when `sw[0]` is ON.

## Setting up the OLED by using Oled\_Display.v



- OD1.** Instantiate the **Oled\_Display.v** module in **Top\_Student.v**
- OD2.** Create and provide a 6.25MHz clock signal named **clk6p25m**, to **Oled\_Display.v**
- OD3.** Create a 16-bit signal named **oled\_data** and initialize it with a value of **16'h07E0**  
The **oled\_data** defines the colour of each pixel. The RGB OLED screen has 16-bit colour resolution, this is represented through 5 bits for the red colour component, 6 bits for the green colour component, and 5 bits for the blue colour component. Connect **oled\_data** to the **pixel\_data** input of **Oled\_Display.v**
- OD4.** Connect the **JX[0 : 7]** signals to the PmodOLED device accordingly
- OD5.** Attach the OLED display to the Basys 3 and update the constraints file accordingly
- OD6.** Optionally connect the **reset** port of **Oled\_Display.v** to any pushbutton, or alternatively give the **reset** port a constant value of 0
- OD7.** Generate the bitstream and download to the FPGA  
Verify that the background colour of the screen is green. Why is it green?  
Optionally, verify that the screen resets when you press the pushbutton indicated in step OD6.

## System integration for the basic effects

- BE1.** Connect the MSBs of `mic_in` to part of the 16-bit colour setting given to the OLED through `oled_data`. The 16-bit colour setting consists of 5 bits for the red colour component, 6 bits for the green colour component, and 5 bits for the blue component. Taking the **first rightmost numerical digit** of the matriculation card of **both members of the team**, add them to produce a `sum_for_the_team`. Then, the 5 (or 6 for green) most significant bits of `mic_in` must be connected to the RGB components of `pixel_data` as tabulated below:

<code>sum_for_the_team</code>	<b>R</b>	<b>G</b>	<b>B</b>
0, 1, 2			✓
3, 4, 5,		✓	
6, 7, 8, 9		✓	✓
10, 11, 12	✓		
13, 14, 15	✓		✓
16, 17, 18	✓	✓	

A tick (✓) indicates that the 5 (or 6 for green) most significant bits of `mic_in` are connected to that respective RGB colour component of `pixel_data`

- BE2. Generate the bitstream and download to the FPGA**

- Observe a single colour with varying intensities, depending on the audio signal
- Using a tone generator application, supply a sine wave to the microphone. You will observe a cyclic repetitive pattern on the OLED screen

- BE3. Write the set of codes to allow the user to press the right or left pushbutton to move a single dash character across the 7 segment displays**

- Right pushbutton should move the dash character to the right, and stop at AN0
- Left pushbutton should move the dash character to the left, and stop at AN3

No matter how long the user press and hold the pushbutton, the dash can only move once per press. Ideally, the pressing of the pushbutton must be accurate and responsive

- BE4. Demonstrate the above to the lab instructors at the beginning of your lab session in Week 8**

**Note:** After this section 4.1. has been evaluated in week 8, it will not be evaluated again during the complete project assessment in week 12, 13. You may remove components you no longer need after the evaluation in week 8

## OLED COORDINATES SYSTEM:

Work on creating an **x** and **y** coordinate system derived from `pixel_index` to make the pixel drawings on the OLED screen less cumbersome (*Hint: You may employ the modulus % and / operators*)

### 4.2A [STUDENT A – OTA] OLED TASK A – BORDERS

**DESCRIPTION:** To create three borders (Red, Orange, Green) on the OLED



- OTA1.** Create a green border near the four corners of the OLED screen  
The thickness of the border should be 1 pixel

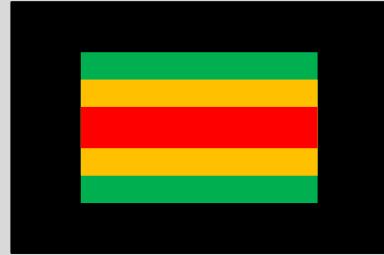
- OTA2.** Create an orange border close to and within the green border  
The thickness of the border should be 1 pixel

- OTA3.** Create a red border close to and within the orange border  
The thickness of the border should be 3 pixels

- OTA4.** The user must have an option to show / hide **ALL** the borders simultaneously when needed  
This show / hide feature only needs to work when demonstrating the OLED TASK A

### 4.2B [STUDENT B – OTB] OLED TASK B – BARS

**DESCRIPTION:** To create three bars (Red, Orange, Green) on the OLED



- OTB1.** Create a green bar reasonably centred on the OLED  
The bar should be 60 by 38 pixels

- OTB2.** Create an orange bar reasonably centred on the OLED, on top of the green bars  
The bar should be 60 by 24 pixels

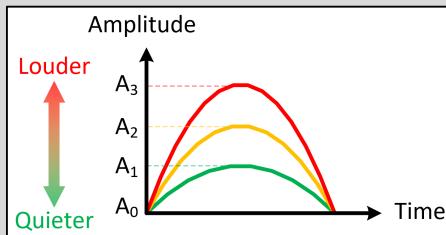
- OTB3.** Create a red bar reasonably centred on the OLED, on top of the orange bars  
The bar should be 60 by 10 pixels

- OTB4.** The user must have an option to show / hide the **ORANGE** bar only when needed  
This show / hide feature only needs to work when demonstrating the OLED TASK B

**Note:** For 1-member group, only TASK 2A OR TASK 2B should be done. Marks are not awarded for both

#### 4.2C [TEAM – AVI] AUDIO VOLUME INDICATOR

**DESCRIPTION:** An audio volume indicator displays a representation of the audio intensity in audio equipment. It helps people to observe the changes in the audio signal visually, such as by extracting the amplitude data from a waveform:



**AVI1.** Find the peak intensity value within regular time intervals

**AVI2A.** Create 6 different ranges for the volume levels and represent them on LD0 to LD4

**Marks for this LED part are awarded to student A**

Level 0: No clearly audible sound (No LEDs)

Level 1: Very low volume (LD0)

Level 2: Low volume (LD0 to LD1)

Level 3: Medium volume (LD0 to LD2)

Level 4: High volume (LD0 to LD3)

Level 5: Very high volume (LD0 to LD4)

**AVI2B.** For each range, a value between 0 to 5 must be shown on any of the anode of the 7-segment displays

**Marks for 7-segment displays part are awarded to student B**

**Note:** For 1-member group, only AVI2A **OR** AVI2B should be done. Marks are not awarded for both

**AVI3.** The following must be seen on the OLED display depending on the volume level:

Level 0: Completely black screen (No borders and no bars)

Level 1, 2: Green border of task OTA1 and green bar of task OTB1

Level 3, 4: Contents of Level 1, 2, with orange border of task OTA2 and orange bar of task OTB2

Level 5: Contents of Level 1, 2, 3, 4, with red border of task OTA3 and red bar of task OTB3

#### GUIDELINES:

**To test if your peak intensity (volume) algorithm works well, the following must be observed:**

If a sinusoidal test signal of constant amplitude is used as an input audio signal, an approximately constant volume reading should be produced. If your volume reading is not quite stable, you would need to improve upon your peak intensity algorithm

#### 4.3 SYSTEM INTEGRATION AND IMPROVEMENT FEATURE(S)

As part of the project requirements, you are required to implement improvement feature(s) to add value to the Non-Entertainment Related Digital System, in order to distinguish your unique system from the rest. The system should be made more interactive, interesting, functional, and user-friendly.

These additional improvement(s) are open-ended but **MUST NOT** be leaning towards entertainment! Evaluation consists of:

- (1) **Functionality:** Feature(s) should provide useful function(s) in the context of NERDS
- (2) **Complexity:** Complexity in implementation of feature.
- (3) **Quality:** Whether feature(s) or user experience(s) is designed well or thorough.
- (4) **Creativity:** Creativity or uniqueness in feature(s) or implementation.

**If you need to connect other external devices (Keyboard, mouse, etc) to the Basys 3 development board, please seek the approval of your lab assistant.**

**If you are connecting one Basys 3 board to other Basys 3 boards, you need to provide a schematic of the PMOD wire connections to seek for the approval of your lab assistant.**

## 5. PROJECT SUBMISSION

There are two items (ITEM A, and ITEM B) to submit to LumiNUS:

Official Lab Session	ITEM A	ITEM B
	Project Archive	Report (Quick Start / User Guide)
Monday	<b>Deadline:</b> End of Week 11 <b>Sunday, 3<sup>rd</sup> April 2022, 12:00 P.M.</b>	<b>Deadline:</b> Start of Week 12 <b>Wednesday, 06<sup>th</sup> April 2022, 12:00 P.M.</b>
Tuesday		
Wednesday		
Thursday		
Friday		

Improvements are evaluated based on your own original **Verilog** program codes. The grading metrics consist of quality, complexity, functionality, and creativity, with the requirement that the features are not leaning towards entertainment.

### 5.1 ITEM A: PROJECT ARCHIVE SUBMISSION

- Only **ONE** Vivado project archive (.zip) per team. **The archive should not exceed 100 MB in size.** (Excess of 100 MB in size is only allowed with prior approval from the lab assistant, and only if Xilinx IP cores have been used)
- Ensure that your single bitstream has been successfully generated and tested on your Basys 3 development board **BEFORE** archiving your Vivado workspace for LumiNUS upload. Download your LumiNUS archive after uploading. Unzip it / Extract all, and check if you can run your bitstream correctly
- The naming for the project and report submission will be provided closer to the submission deadline.

### 5.2 ITEM B: REPORT (QUICK START / USER GUIDE) SUBMISSION

- Include the following in your report (**One A4 size double-sided sheet at most for the report**):
  - ❖ Name and matriculation number of both students
    - **Indicate clearly** who is Student A and who is Student B
  - ❖ Official lab session (Examples: Monday P.M., Wednesday P.M., Thursday A.M.) and group I.D.
    - Your project group I.D. will be present in the document "*EE2026 Project Assessment Schedule.pdf*", and which will be uploaded at the end of week 11
  - ❖ Quick start / User guide which consists of:
    - Description of the features that you have designed and successfully implemented
    - Instructions on how these features can be operated by the user  
(Use the template provided on the next page)
    - Instructions may alternatively be described through flowcharts
  - ❖ Images must be in colour
  - ❖ Feedbacks. Possible topics include:
    - What did you like most / least about the project?
    - How would you suggest the overall project assignment be improved?
    - Any other constructive feedbacks / suggestions are welcome
- Note that feedbacks, whether positive or negative, **DO NOT** have any effects on your grades 😊
- ❖ References:
  - Include references to open source codes
  - Plagiarism penalties apply for open source codes that are not properly referenced
- The naming for the project and report submission will be provided closer to the submission deadline.

## TEMPLATE FOR THE QUICK START / USER GUIDE

Ensure that your quick start / user guide has the following 3 columns (There is no need to include the instructions found on the second row):

Brief Feature Name	Feature Description	Images / Photos
Features not indicated here will not be evaluated  Claim the feature marks for: - Student A <b>OR</b> - Student B <b>OR</b> - Team marks  Your team is responsible in <b>deciding</b> who will claim the marks	Indicate how to use the feature, and the expected effect that would be observed.  The instructions should be clear and should also indicate the input devices used. For example: - SW0, SW[12:8], PBC, Character "X" on keyboard, Left mouse button	Take pictures of your Basys 3 / OLED screen to support your feature description  Images that properly support your feature description will help you explain better
<b>Student A: Jane</b> OLED Task A Borders + AVI2A	<b>SW[0] = 0</b> : Borders shown <b>SW[0] = 1</b> : Borders hidden ...	
<b>Student B: Tom</b> OLED Task B Bars+ AVI2B	<b>PBL:</b> Show bars ...	
<b>Student A: Jane</b> "Name of improvement 1"	...	...
<b>Student B: Tom</b> "Name of improvement 2"	...	
<b>Student B: Tom</b> "Name of improvement 3"	...	...
<b>Team</b> "Name of improvement 4"	...	...
<b>Team</b> "Name of improvement 5"	...	
<b>Team</b> "Name of improvement 6"	...	

- HIGHLIGHT THE BACKGROUND COLOUR OF TASKS WHOSE MARKS ARE CLAIMED BY **STUDENT A IN PASTEL / LIGHT BLUE**
- HIGHLIGHT THE BACKGROUND COLOUR OF TASKS WHOSE MARKS ARE CLAIMED BY **STUDENT B IN PASTEL / LIGHT GREEN**
- HIGHLIGHT THE BACKGROUND COLOUR OF TASKS WHOSE MARKS ARE CLAIMED BY **THE TEAM IN PASTEL / LIGHT PINK**

**Each distinct improvement can only be claimed once, either for the individual component, OR for the team component**

Bring **TWO** printed copies of your report in **COLOUR**, when you come for your assessment  
Remember to limit your report (User guide) to 1 A4 sheet