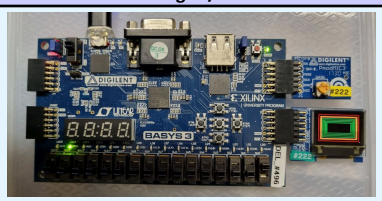
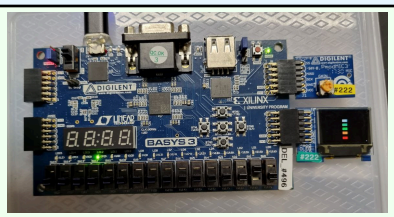
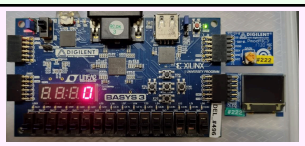
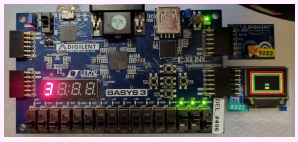
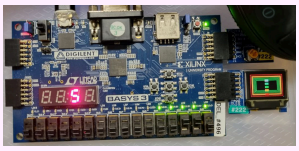
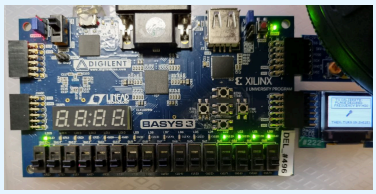
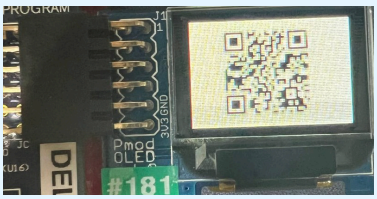
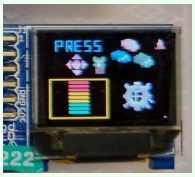
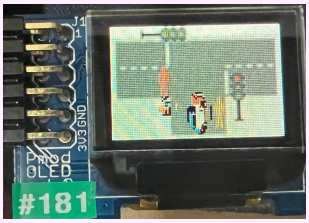
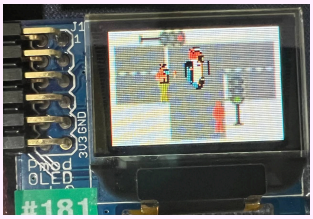
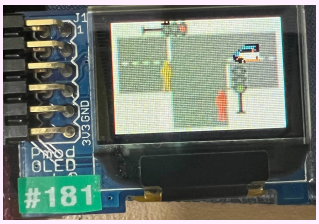
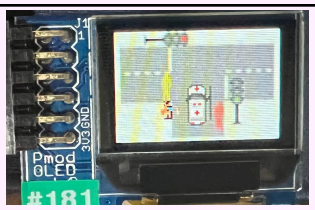
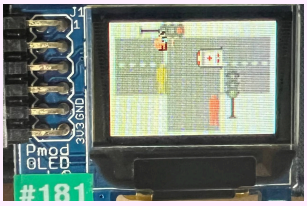
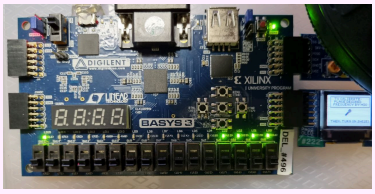
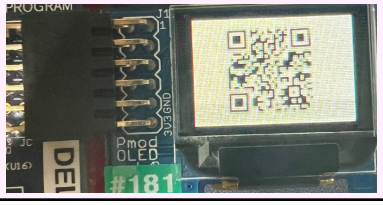
Friday P.M. S1_13
Student A: Thomas Joseph Lee Alba A0238909N
Student B: Joshua Goh Min Rui A0239779A

Feature Name	Feature Description	Images / Photos
Student A: Thomas OLED Basic Task A	Under the Audio Volume Indicator in the menu (Bottom Left) SW[0] = 1 : Turns on OLED Basic Task A BTNU = 1: additional borders will appear	
Student B: Joshua OLED Basic Task B	Under the Audio Volume Indicator in the menu (Bottom Left) SW[1]=1 : Turns on OLED Basic Task B BTND = 1: additional boxes will appear	
Team: Audio Volume Indicator Basic Task	Enter the Audio Volume Indicator in menu (Bottom Left) SW[3] = 1: Turn on Basic Team Task's Audio Volume Indicator. 1) OLED would display level 0 to level 5 based on real time volume levels. 2) Anode and 7 Segment display would also display numbers from 0 to 5, indicating the current level mic is detecting 3) LED[4:0] would also light up depending on the real time volume level detected as well	  
Student A: Thomas Audio Frequency Detector	Enter the Settings from the menu (Bottom Right) 1) SW[15] = 1 : Turns on calibration mode and calibrates frequency based on audio frequency being played at the mic (Under Team Task) 2) SW[14] = 1: OLED displays QR code which links to a website that can play audio at different frequencies 3) LED[4:0] displays the volume of the audio detected, similar to the behaviour of LED[4:0] in basic task for team, with no LD lighting up with no audible sound, and all LDs lighting up if it detects high volume audio 4) LED[15] lights up when the frequency of the detected audio **passes the calibrated threshold** of 1.5kHz (by default). Frequency is measured via finding the sample data's **zero-crossing rate**	 
Student B: Joshua Menu Display Animation and Design	BTNR||BTNL||BTNU||BTND = 1 : Cycles right/left/up/down between the menu items BTNC = 1: Enter current setting that is selected SW[7] = 1 : return back to the main menu 1) At the start of the program, the OLED screen will display the menu settings where users can **toggle** between different functions of the system 2) When "cursor" is hovering over the current setting, that section of the screen will be **animated** 3) This is done by creating a background image using a python script to create a case basis for every index. After which, for every frame, by taking the **relative position** of the x-axis position and y-axis position, the image can be calibrated to move according to its position on the OLED	   

	4) By using this method, I was able to save memory space, as for each animation, I did not need to read in a new memfile but rather just tweak the pixels accordingly or adjust the x or y position of the frame	
Team: Traffic Simulation During Non-Emergency	Enter the Traffic Simulation from the menu (Top Right) BTNU = 1: Simulates pedestrian button being pressed. Thus traffic lights will change, allowing pedestrians to cross and cars at minor road to move Mic Audio Detection (of above 3500 threshold): Simulates car waiting at minor road, thus traffic lights would change color as well The OLED screen displays a **traffic simulation** at a T junction road with the horizontal road being a major road while the vertical road represents a minor road. This idea was inspired by Prof Chua when she gave an example of this during her lecture. The concept is the major road will continue to let traffic flow unless: 1) BTNU(Pedestrian at top) is pressed 2) The mic detects an audiowave (Car waiting at minor road) of digital threshold of 3500 from distance 1 cm away Once the minor road turns green, a countdown will appear on the 7 segment display to show how long a pedestrian has to cross the road, a mini Among Us figure simulates pedestrian crossing Minor road traffic would be green for 8 seconds, yellow for 2 seconds then finally red, allowing major road to turn green again. This signals the end of the cycle After 1 cycle of traffic change, the push buttons and mic intensity detection would wait 5 seconds before allowing traffic to change again	  
Team: Traffic Simulation During Emergency	Enter the Traffic Simulation from the menu (Top Right) 1) When mic captures an audio wave of above 1500 Hz (or calibrated threshold frequency) for 1-2 seconds, it **signals** that there is **an ambulance** at the minor road 2) Traffic light immediately changes to let the ambulance go. An ambulance turning right would then appear once there is a green light for the bottom lane 3) This function overwrites any traffic rules and thus will be activated immediately after hearing the frequency above 1500 Hz (should minor road be red)	 
Team: Calibration of Emergency Vehicle Frequency	Enter the Settings from the menu (Bottom Right) 1) SW[15] = 1: turns on **calibration mode** for the frequency detector, to adjust the frequency threshold detected (both for the settings and the traffic simulation) 2) Play audio with the desired frequency onto the Mic, and ensure it is being picked up by the Mic with sufficient volume (at least LED[2:0] lights up) 3) SW[15] = 0 to finish the calibration. Audio frequency picked up would be calibrated as the threshold frequency demarking when LED[15] should light up, and what constitutes an emergency vehicle detected in the traffic simulation. Replaces the initial default 1.5kHz frequency set	 

Feedback: It was fun especially the project component! We learnt a lot of new things about Verilog that we never knew.