ESSA - Learning Contract

Michael Cooke - 13244785, Raul Lyons - 24565184, Jay Tinsley - 24567707

August 2024

Contents

1 Project Outline **2**
- 1.1 Flowchart . . . 2
  - 1.1.1 UML Code . . . 2
- 1.2 Description . . . 3
  - 1.2.1 Aim . . . 3
  - 1.2.2 Why this project . . . 4

2 Specifications **4**
- 2.1 Input Sensors . . . 4
  - 2.1.1 Decibel Sensor . . . 4
  - 2.1.2 Potentiometer . . . 4
- 2.2 Arduino . . . 4
- 2.3 I2C Communication . . . 5
- 2.4 FPGA Processing . . . 5
- 2.5 VGA Driver . . . 5
- 2.6 Screen Format . . . 5

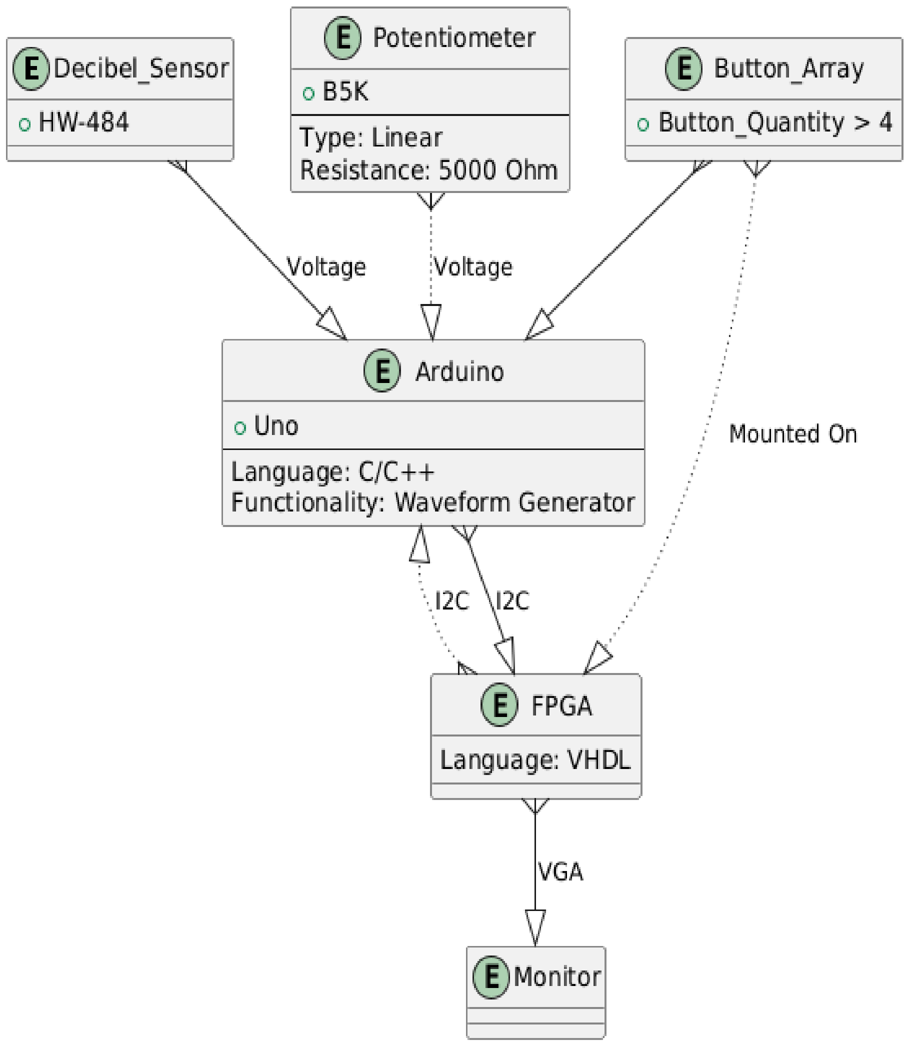
3 Timeline **6**
- 3.1 Task Allocation . . . 6

4 Materials **6**

5 Risks **8**

1 Project Outline

1.1 Flowchart



1.1.1 UML Code

```
entity Decibel_Sensor {
  +HW-484
}

entity Potentiometer {
  +B5K
```

```
  --
  Type: Linear
  Resistance: 5000 Ohm
}

entity Arduino {
  +Uno
  --
  Language: C/C++
  Functionality: Waveform Generator
}

entity FPGA {
  Language: VHDL
}

entity Button_Array {
  +Button_Quantity > 4
}

entity Monitor {}

Decibel_Sensor }--|> Arduino : "Voltage"
Potentiometer }-[dotted]-|> Arduino : "Voltage"
Arduino }--|> FPGA: "I2C"
FPGA }-[dotted]-|> Arduino: "I2C"
Button_Array }-[dotted]-|> FPGA: "Mounted On"
Button_Array }--|> Arduino: ""
FPGA }--|> Monitor: "VGA"
```

1.2 Description

1.2.1 Aim

The purpose of the project is display the output of a sensor to a screen via VGA. This will be achieved by ingesting the direct input into an Arduino, then using an I2C bus to communicate with the FPGA. After the input information is on the FPGA some processing will be done, then the output will be transformed to a VGA output and will be displayed to a monitor.

During development simulated wave-forms will be generated on the Arduino until input devices can be tested. This functionality will be toggled in the final product via a control signal (button press).

The complexity of this project can be extended over the course of the subject through:

- Addition of a second input device
- Increasing the complexity of the signal processing completed on the FPGA
- Attaching the control signal directly onto the FPGA

1.2.2 Why this project

This project can be used as a prototype for any kind of application requiring real-time sensor data visualisation. The main application that was considered was for electrocardiogram (ECG) monitoring, but it can also be used to measure data from position sensors, audio signals, thermometers, etc.

This project allows for the opportunity to develop and demonstrate skills including:

- Embedded systems designing
- FPGA programming
- Serial communication between IC's
- VGA signal generation

2 Specifications

2.1 Input Sensors

These sensors will be used as the input for our circuit system, This will provide real time data that we can test our system with. End goal will be to use the Decibel sensor to provide a visual representation of the Sound in the room.

2.1.1 Decibel Sensor

The decibel sensor will be hooked into the Arduino and provide real time numerical analysis of the audio levels in the room. This value can be attained by the arduino software through the AnalogRead Function.

2.1.2 Potentiometer

A potentiometer will also be used to provide changing variable data in a simpler more controllable fashion for testing purposes.

2.2 Arduino

The arduino will be used to interface with the input sensors to then be related through I2C towards the FPGA board. With aims to also interface with the arduino board through I2C from the FPGA.

The arduino uses ATmega328P as its main processor, and also contains 2KB SRAM, 32KB FLASH, 1KB EEPROM of memory.

2.3 I2C Communication

The inter-integrated circuit (I2C) will be used to transmit the sensor data from the Arduino to the FPGA. For Arduino, the clock-speed for the I2C generally ranges from 100 KHz to 400 KHz. If necessary, we may also try to implement bi-directional communication if data from the FPGA needs to be sent back to the Arduino.

I2C is synchronous serial protocol that uses two signals, one data and one clock, to communicate between one master to multiple slave devices. The primary downside of is that it is limited to 5Mb/s, but it should be sufficient for our purposes.

2.4 FPGA Processing

The FPGA board will be used to process the Waveform from the Arduino board into the VGA format as well as create the layout for the screen.

It is expected that most, if not all programming will be written in VHDL/verilog instead of the visual editor that was used in IDE.

2.5 VGA Driver

The FPGA board will be programmed to output a series of voltages to be transmitted through the board's VGA port. We aim to process this at 30hz for a reasonable viewing experience.

The VGA protocol consists of an active region and a Blanking region. There are 2 values for when to shift to a new line and a new frame, these values are Hsync and Vsync respectively.

For each pixel horizontally in the active region voltages are passed for the R, G and B values. During this period the Hsync is high. Hsync continues High for another 16 clock cycles (Front porch) and then shifts to Low for 96 clock cycles (Sync pulse), Hsync then shifts back to high for 48 clock cycles (Back porch). The shifting from High to low is what tells the screen to move to a new line.

A similar approach is used for the shift to a new frame, the Vsync Front porch is 10 clock cycles and then shifts to Low for 2 clock cycles for its Sync pulse, Hsync then shifts back to high for 33 clock cycles for its Back porch.

2.6 Screen Format

We aim to display on a screen with aims of 30Hz or higher frequency.

3 Timeline

Week	Task
1	Subject introduction, group establishment
2	Creation of project scaffold, establish git and communication channels
3	Complete **Assessment 1** - Learning Contract, acquire all required materials
4	Assign tasks to individual members and research theoretical implementations of I2C and VGA protocols
5	Implement POC for Arduino Signal Generation, I2C communication and VGA driver
6	Prep for and completion of **Assessment 2** - Project Update
7	Connect input sensor/s to the Arduino, Establish 2-way IC2 communication, Draw shape via VGA
8	Integration part 1, Test signal to I2C to VGA to monitor
9	Toggle signal, FPGA processing, generalised VGA driver
10	Integration part 2, Toggled signal to I2C, FPGA processing, VGA to monitor
11	Testing / Buffer week / Opportunity for adding extensions
12	**Assessment 3** - Project Video and Report

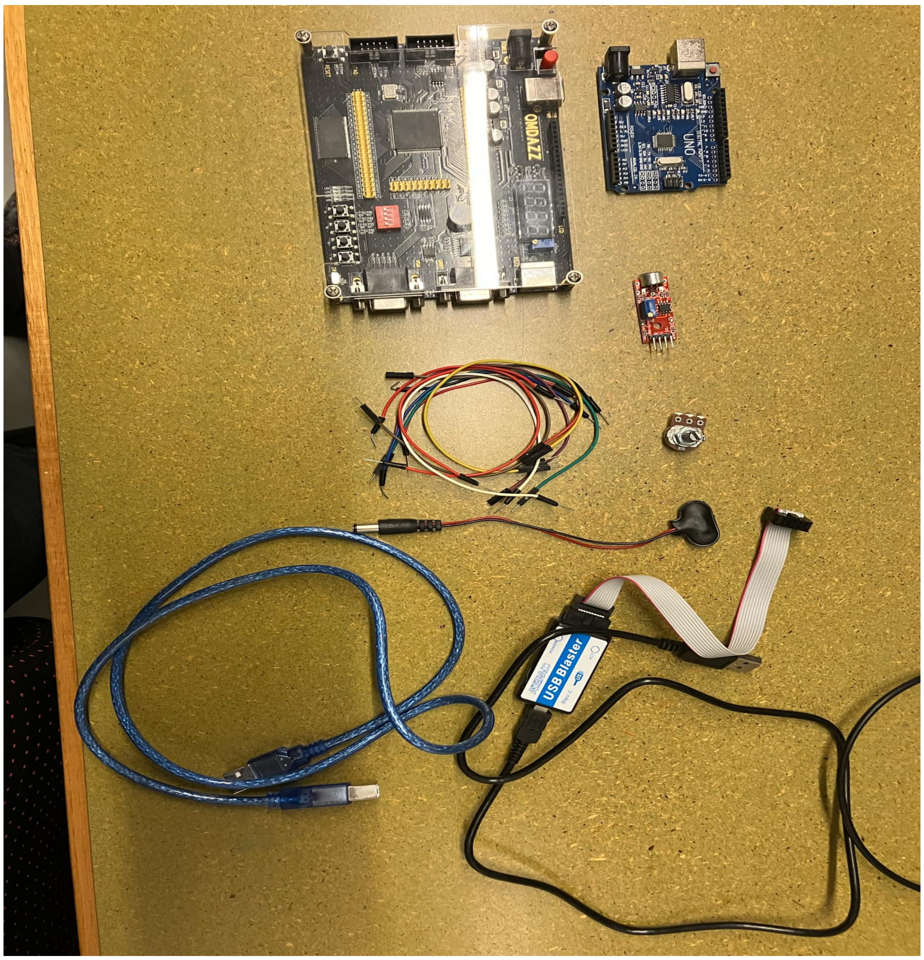
3.1 Task Allocation

Team Member	Expected Tasks
Raul Lyons	I2C Communication
Jay Tinsley	VGA Communication
Michael Cooke	Collecting Sensor readings, mapping screens and layouts

4 Materials

1. FPGA Kit
2. Computer w/ Intel Quartus Prime Lite
3. Ardunio Uno w/ Power
4. Primary Input sensor
5. Secondary Input sensor
6. Control Signal - button
7. Console cable
8. VGA cable

9. Monitor (that accepts VGA)
10. Wires
11. Breadboard



5 Risks

Risk Identified	Risk Mitigation
Data loss between devices	Timing protocols, synchronise FPGA clock with Arduino
Software data loss	Use Git for source control of local issues, with Github for sharing and remove storage
Debugging and testing	Work on and test each component before adding the next
Breaking hardware	Take extra caution when handling hardware. Have backup parts for uncommon materials that cannot be sourced on short notice (within 3 days)
Project incomplete due to time management issues	Keep project on track using timeline. Prioritize MVP tasks before moving onto extension tasks.