SHARD

Structural Health and Rupture Detection

CSE Team Members:

Julian Herrera, jherrera2020@my.fit.edu

Matthew Manley, mmanley2018@my.fit.edu

Aero/Mechanical Team Members:

Matthew Meesit, mmeesit2019@my.fit.edu

John Bruce, jbruce2020@my.fit.edu

Paul Awad, pawad2021@my.fit.edu

CSE Faculty Advisor:

Dr. Silaghi, msilaghi@fit.edu

Advisor Meeting Dates:

10/25/2022

Client:

Dr. Willard, FIT Aeronautics Professor and NASA Engineer

Client Meeting Dates:

10/7/2022

10/14/2022

10/21/2022

10/28/2022

Table of Contents

1 Current Milestone Progress

Task	Completion %	Julian	Matthew	To do
1. Increase understanding of how to draw the tiles from the hardware system (algorithm).	100%	50%	50%	none
2. Experiment with multiplexer and raspberry pi connection with each other and with the laptop.	90%	50%	50%	We still don't have a multiplexer to use
3. Create the basic structure of GUI	100%	80%	20%	none

2 Discussion - Tasks

2.1 Task 1: Increase understanding of how to draw the tiles from the hardware system (algorithm)

In order to understand how we analyze the tiles from the hardware system, we first have to understand the hardware system. For this we had to design the data acquisition system (DAQ) for our project. This is essentially the interface between hardware and software; the flow of data from sensors to computer.

In the overall physical system/structure, each tile has 8 sensors (tentatively, subject to change). 3 vibration sensors per lattice layer (2 layers per tile) and 2 sensors for the RDCS (reaction damage control system). We decided to have the sensors connect to multiplexers. 1 group of 8 sensors distinguishes 1 tile on the multiplexer, and each multiplexer can have 16 inputs (2 tiles). These multiplexers will be connected to different analog pin numbers on microcontrollers. The microcontroller processes the data and sends it to the computer.
In the software application, the user must designate which physical tile corresponds to which virtual tile by use of a universally unique identifier (UUID). The UUID is calculated based upon the configuration (i.e. which position in the multiplexer, which multiplexer connected to, and which microcontroller). Once a virtual tile has been correlated with a physical tile, the application can then know where data for a particular UUID will be stored and directed to.
An obstacle encountered with this task was getting the necessary information for the shield tiles. At some times it was discussed that we would have only 3 sensors per tile, and other times we would have 8.

2.2 Task 2: Experiment with multiplexer and raspberry pi connection with each other and with the laptop

We were unable to get a multiplexer to experiment with, however we did get a raspberry pi from the library. As an extension to this task, we also experimented with an arduino because we are still unsure of which microcontroller to use. As part of this task, we made some experimental programs to test the capabilities of the microcontrollers. For the arduino, Julian created a program that sent commands to the arduino via usb serial port in order for the arduino to turn its internal led on and off.

The Raspberry Pi microprocessor was also retrieved from the library and experimented with. Matthew wrote multiple programs for future testing of the Raspberry Pi. One program simply signals the Raspberry Pi to blink an LED light. The second program is more complicated and involved connecting the Raspberry Pi to a vibration sensor and reading the input. Sample programs were also conceived to test the multiplexer.

2.3 Task 3: Create the basic structure of GUI.

The purpose of this task was to create a framework for the GUI to get started before we start implementing all of the functionality. We wanted to be able to see how the GUI would look like and discuss with our client how the GUI should look before we start implementing the more labor intensive graphics and simulation implementations. We have a working application that has menus and menu items, as well as a basic welcome screen.

3 Discussion - Contributions

3.1 Julian Herrera

Organized the initial commit and code base for the project on github, along with some of the code necessary to use wxWidgets and OpenGL together in an application. Created arduino and c++ program that interface with each other over a usb serial connection in order to experiment with an arduino microcontroller before we start dealing with sensors. Designed our team's (not just CSE) data acquisition system (DAQ).

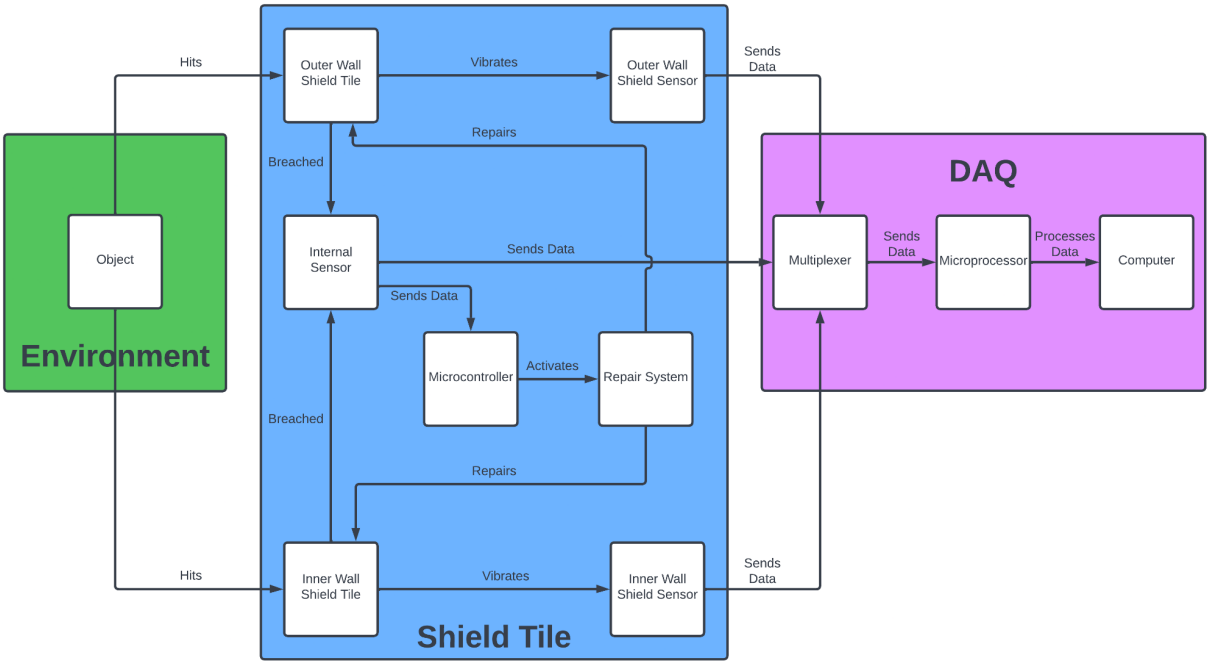

Figure 1: Flow of data to the DAQ system

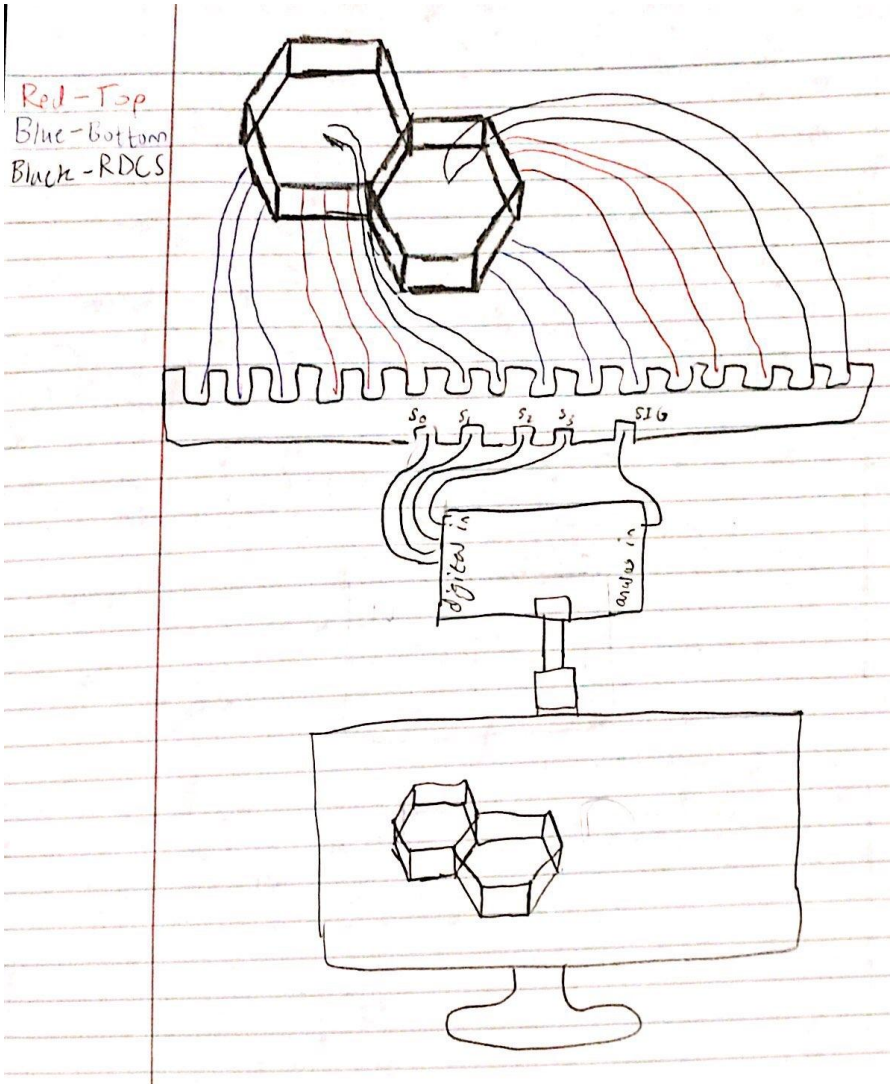

Figure 2: Mockup of the DAQ system

3.2 Matthew Manley

Wrote sample programs for the Raspberry Pi microprocessor, including a program that takes input from a vibration sensor. Performed research into the Repair Damage Control System (RDCS) and the various methods to repair a puncture when a collision occurs. Different types of foam sealants were researched and experimented on. Also, sample test programs for the multiplexer were outlined for future testing.

```python
#Vibration Sensor Raspberry Pi
import RPi.GPIO as GPIO
import time

# GPIO SETUP
channel = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.IN)


def callback(channel):
    if GPIO.input(channel):
        print("Movement Detected!")
    else:
        print("No Movement Detected!")


GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)  # let us know when the pin goes HIGH or LOW
GPIO.add_event_callback(channel, callback)  # assign function to GPIO PIN, Run function on change

# infinite loop
```

```python
#Basic RPi program to blink LED

import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library
from time import sleep # Import the sleep function from the time module

GPIO.setmode(GPIO.BOARD) # Use physical pin numbering
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW) # Set pin 8 to be an output pin and set

while True: # Run forever
 GPIO.output(8, GPIO.HIGH) # Turn on
 sleep(1) # Sleep for 1 second
 GPIO.output(8, GPIO.LOW) # Turn off
 sleep(1) # Sleep for 1 second
```

4 Next Milestone Plan

Task	Julian	Matthew
Finish simple GUI	30%	70%
Implement 3D rendering capabilities	70%	30%
Experiment hardware to software interfaces	50%	50%

5 Discussion - Planned Tasks

5.1 Task 1: Finish simple GUI

The purpose of this task is to finish off the overlay GUI. We need to create the taskbar icons, a loading icon, and a desktop icon. We also need to fully implement the welcome screen. It should have buttons wherein the user can click to direct them to the model editing panels or to the about page.

5.2 Task 2: Implement 3D rendering capabilities

The motivation for this task is similar to that of the “hello world” demos conducted in the first milestone. We must understand how the application and wxWidgets works with OpenGL 3D rendering so that we are able to understand the tasks that need to be done in the future when it comes to implementing the graphics alongside the GUI (such as when there is a mouse event in the wxGLCanvas). We will do this by creating a simple 3D shape, which is the OpenGL part. The wxWidgets part is when the user clicks on “New”, which is the interface between wxWidgets and OpenGL.

5.3 Task 3: Experiment hardware to software interfaces

The motivation for this task is built off of previous tasks in regard to experimenting with microcontrollers. Once our team acquires the sensors they need for their part of the project, we will be using those sensors to make simple demos. We need to do this in order to develop an understanding of what technical information the sensors will be transmitting (e.g. it may be that we need to do some math to extract the data received from the microcontroller).

6 Client Meetings

- October 7, 2022 - Discussed with Dr. Willard about the sensors and the Data Acquisition (DAQ) system
- October 14, 2022 - Discussed with Dr. Willard about the sensors and the Data Acquisition (DAQ) system
- October 21, 2022 - Discussed with Dr. Willard about the sensors and the Data Acquisition (DAQ) system
- October 28, 2022 - Discussed with Dr. Willard about the sensors and the Data Acquisition (DAQ) system

7 Client Feedback

- Main concern with how the sensors are going to work and be produced.
- Wants to understand how the Data Acquisition (DAQ) system will be implemented with the other subsystems

8 Faculty Meetings

- October 25, 2022 - Met with Dr. Silaghi to discuss our progress and show him the GUI framework

9 Faculty Feedback

- Task 1: None
- Task 2: None
- Task 3: Confirmed that the GUI is good so far.