SFWRENG 3K04: Software Development

Assignment 1 – Testing

Dr. Alan Wassyng

October 27, 2019

Adam Bujak – bujaka – 400113347

Eric Hillebrand – hillebre – 400143468

Harneet Singh – singhh76 – 400110275

Karan Gill – gillk20 – 400130973

Stephen Scott – scotts24 – 400139933

Table of Contents

[Hardware Testing 3](#_Toc23283557)

[Oscilloscope Testing 3](#_Toc23283558)

[AOO and VOO 3](#_Toc23283559)

[AAI and VVI 3](#_Toc23283560)

[Bonus 3](#_Toc23283561)

[LED Testing 4](#_Toc23283562)

[AOO and VOO 4](#_Toc23283563)

[AAI and VVI 4](#_Toc23283564)

[Bonus 4](#_Toc23283565)

[Simulink Simulation 4](#_Toc23283566)

[DCM Testing 5](#_Toc23283567)

[GUI 5](#_Toc23283568)

[Database 5](#_Toc23283569)

[Application 5](#_Toc23283570)

[Appendix 6](#_Toc23283571)

# Hardware Testing

1. Observing the waveform generated by the pacemaker using an oscilloscope
2. Blinking LEDs to represent the current state in the FSM
3. Simulating the model in Simulink

## Oscilloscope Testing

### AOO and VOO

* After connecting the oscilloscope with the pulse-generating output pins of the shield, the oscilloscope’s display showed a periodic pacing behaviour
* By observing the delay between waveforms, the beats per minute rate was confirmed to match the current BPM programmed in the model
* The pace waveforms were confirmed to match the requirements by examining both the amplitude and width with what were programmed in the model
  + The waves had an amplitude of 3000mV for 2ms
* Wave rectification was observed on the oscilloscope which confirms that the model was successfully implementing a net zero charge with each pace

### AAI and VVI

* The oscilloscope was connected to the input and output pins of the shield to allow for both pulse-generation and sensing by the hardware
* myRio software was used to control the pulse characteristics of the natural paces to be sensed
* myRio pulses were set at various BPM rates to determine the effect on the pace inhibition
  + When the natural rate was set to 30BPM, and the pacemaker to 60BPM, each second pacemaker pace was inhibited, so the pacemaker would send a pace in between two naturally detected pulses
  + When the natural rate was set higher than the pacemaker rate, the pacemaker would not send any paces because each pace was being inhibited by the natural pulses
* The pacemaker wave characteristics matched what was first confirmed in the AOO and VOO testing, e.g.:
  + Pulse Amplitude, Pulse Width, Wave Rectification

### Bonus

* For each programmed mode, when the push button was pressed while the shield was connected to the oscilloscope it was observed that a pace would be inhibited after the button was activated

## LED Testing

### AOO and VOO

* The FSM was programmed such that:
  + If in the atrium pacing state, turn the blue LED on
  + If in the ventricle pacing state, turn the red LED on
  + If in the initial state or charging states, turn any LEDs off
* By observing the LEDs, it was determined if the FSM was following the required behaviour, i.e.:
  + If the pacing mode was AOO, the LED would blink blue for a time defined by the atrium pulse width, at a rate defined by the BPM
  + If the pacing mode was VOO, the LED would blink red for a time defined by the ventricle pulse width, at a rate defined by the BPM

### AAI and VVI

* When a pace was to be sensed by the shield, the LED would not blink for the next beat, which signified that a pace had successfully been inhibited
* When a pace was sent, the LED blinked blue or red corresponding to an atrium or ventricle pace respectively

### Bonus

* If the onboard pushbutton was activated, it was noted that the LED would not blink blue or red for the next beat, thus signifying that a pace had been inhibited

## Simulink Simulation

* Running a simulation of the model in Simulink allowed us to confirm the transitions from charging the C22 to either the atrium or ventricle pacing state; however, it was not used in any sensing testing
* When a simulation was run, it was confirmed that the FSM would transition from C22 charging to atrium or ventricle pacing, corresponding to the current mode, at the rate defined by the programmed BPM
* This confirmed that the delays and transitions were satisfying the requirements

# DCM Testing

## GUI

* Verified all features of the GUI were working using print statements and assertions
  + All buttons perform the intended action
  + All callbacks are called properly and are executed
  + User input is passed back to the main application properly
  + All user errors produce correct error flag
    - Shown in appendix figures 1-5
* Verified that user input fields only accepted valid input by boundary testing the validation functions used for each input
  + As seen in appendix figures 6 and 7

## Database

* Verified user data is being stored in the database
* Verified user data can be retrieved and edited

## Application

* Verified user input from GUI is being passed correctly to database manager.

# Appendix

Figure 1.

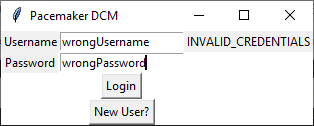


Figure 2.

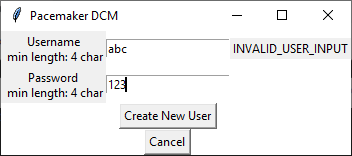


Figure 3.

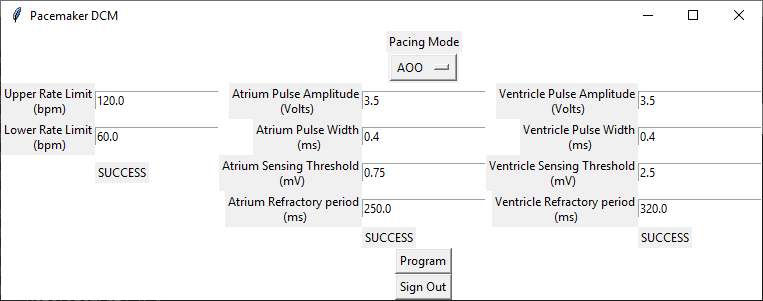


Figure 4.

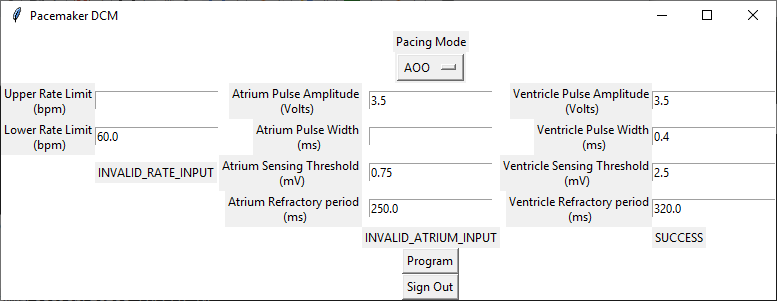


Figure 5.

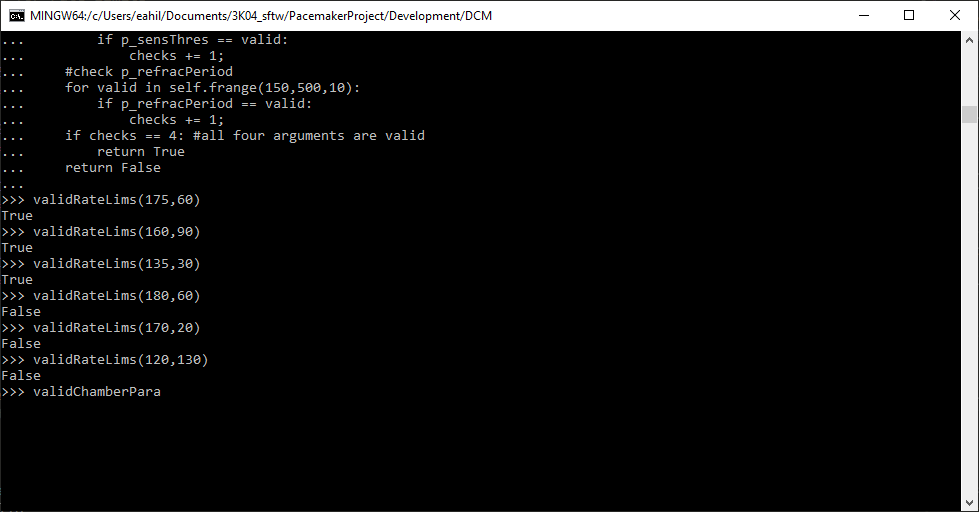


Figure 6.

