Project 5:

Learning the Development Environment

– The Final Step

EE/CSE 474 EMBEDDED MICROCOMPUTER SYSTEM

Contributor: Gaohong Liu, Shawn Xu, Yongqin Wang

Instructor: James Peckol, Daniel Predmore, Christy Truong, Jonathan Hamberg

University of Washington

Department of Electrical Engineering

A Special Appreciation to Dr. James K. Peckol for developing this challenging and extremely useful project and to Our TA Daniel Predmore, Christy Truong, Jonathan Hamberg for their useful help and patience!

" *Go get a beer!* "

—— **James K. Peckol**

Abstract

The final phase of development of a low cost, portable, medical monitoring system is completed in this lab. The following capabilities are added: the ability of modeling the control of blood pressure cuff with press button and switch; the ability of measuring patient's temperature, blood pressure, pulse rate, respiration rate and EKG modeled by signal generator generated signals; The ability to display detailed information on TFT console with special indication on critical data; The ability to support bidirectional communication with remote computer; The ability to respond to command from remote computer; The ability of displaying measurement, status and alarm information on a remote terminal.

Table of Contents

Abstract **2**

Introduction **3**

Discussion of the Lab **3**

Design Specification 3

Software Implementation 10

Presentation, discussion, and analysis of the results **18**

Analysis of errors needed to be resolved **46**

Analysis of errors encountered **46**

Test **47**

Test Plan 47

Test Specification 47

Test Cases 49

Summary and Conclusion **53**

Summary 53

Conclusion 54

Individual Contribution **54**

Appendices **54**

User Manual 54

Source Code 56

Extra Question 56

Reference **57**

1. Introduction

This project focused on the design and implementation of system architecture, data model, control flow, data exchange, operation control, intersystem and LAN communication for a low cost, portable, medical monitoring system. The system is designed to dynamically control and schedule tasks based on user instructions and other inputs; measure, compute and display six kinds of health related information (blood pressure, body temperature, pulse rate, respiration rate, EKG and battery status) and communicate with remote computer. We used UML - Use Cases, Functional Decomposition, Sequence Diagrams, State Charts, and Data and Control Flow Diagrams to design the architecture of the system, C language to implement functionality, Codeblock IDE to debug and finally loaded the final product to the Arduino ATMega2560 and Arduino Uno.

2. Discussion of the Lab

Design Specification

Overall Summary

The product developed is the medical monitor device that can support the measurement, display and communication of several modeled health data input. For this final stage of development, the goal is to design, implement and integrate three main functionality into the system.
First, the support for modeling health data with various analog/digital signals will be added. Peripheral system will be able to measure/compute/store/communicate these modeled signals. The ability of measuring respiration rate and EKG will be added. Also. peripheral system will be able to control the modeled blood pressure cuff using the signal sent from press button/switch circuit.
Second, a remote communication/control system will be integrated with LAN. The system will be able to send data and receive command from remote computer, interpret and respond to command and the remote computer terminal will be able to display measurement data, warning and alarm information.
Last, the TFT display will be able to display more detailed measurement information and display any critical information with special pattern.
To achieve the desired functionality of such system, the system is divided into three subsystems: Uno as Peripheral, ATMega2560 as System Control and PuTTY as remote system terminal. In addition, TFT display panel is used as user interface with the ability to interact with user using touch screen. System control subsystem is responsible for overall task management of the system, Peripheral subsystem is responsible for receiving, process , storing and communicating user health data, TFT display panel is responsible for displaying and PuTTY terminal is responsible for controlling and communicating with the system remotely .

Use Cases

This is the topology of our fully implemented application. Served as Control or Kernel, ATMega2560 has a dynamic control kernel which can control the status of whole system. Connecting with pulse generator, Uno works as a measurement tool and it receives data from outside world and send them to ATMega to processes and displays. As remote control system, PuTTY terminal connects with the rest of system through LAN, sends command to kernel to control system operation and display measurement data, warning information received.

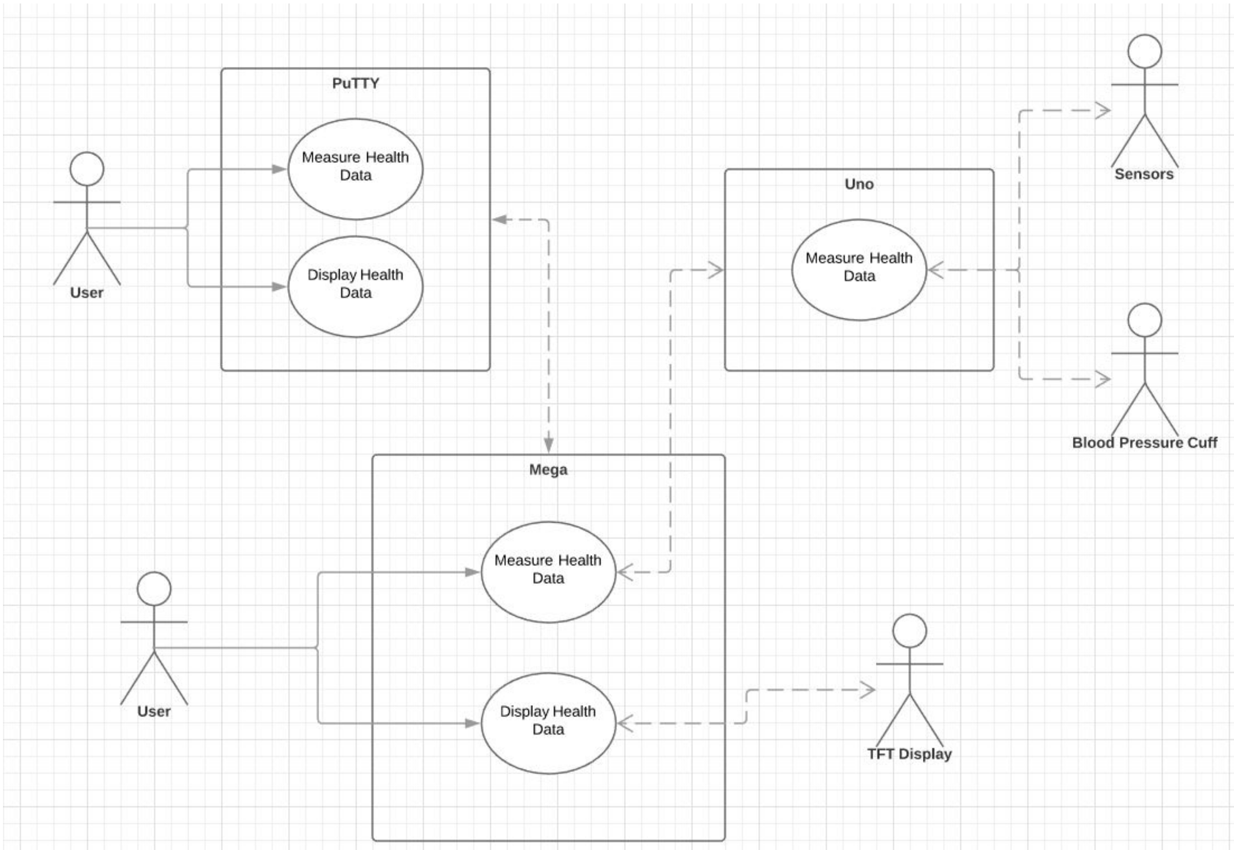

figure 1 Use Case Diagram

<u>Interface</u>

User to Computer Interface:

Input:

Task Selection Menu to selected measured task, Alarm Acknowledgement Button to acknowledge alarms

Output:

Measurement Item

Body Temperature, Blood Pressure, Pulse Rate, Respiration Rate, EKG

Annunciation

Blood Pressure, Body temperature, Pulse rate, Respiration Rate, Battery Status

System Process Interface:

Input

Measurement:

Blood Pressure, Body temperature, Pulse rate, Respiration Rate, EKG

Output

Display:

Blood Pressure, Body temperature, Pulse rate, Respiration Rate, EKG

Status:

Battery state

Alarms:

Temperature, Blood pressure, Pulse rate 20% out of range

Warning:

Temperature out of range, blood pressure out of range, pulse rate out of range

Side Effect:

After boot up, the display will show three text box options: Task Selection, Annunciation and Measurement. Tab on option will take user to corresponding sub menu.

In task selection, user will be able to selected which measurement task to perform, with selected task displayed in green text box and unselected tasks displayed in red text box.

In annunciation, user will be able to acknowledge warning by tabbing on "acknowledge" button. If the button is tab and warning are acknowledged, the tab will be in green text box, otherwise in red text box.

In measurement, user will be able to read all measurement data.

All the sub menu has a "back" button to return to top menu.

<u>Pseudo English description of algorithms, functions, or procedures</u>

Main function is the interface where users could choose the items they would like to measure and make system awake to work or sleep

```
main():
      If pressure on screen is in valid range:
            If user is in top menu:
                  Find the range users touch
            If user is in measurement selection menu:
                  If user choose return:
                        Return to top menu:
                  If choose to measure temperature:
                        Measure temperature
                  If choose to measure pulse:
                        Measure pulse
                  If choose to measure pressure:
                        Measure pressure
                  If choose to measure respiration rate:
                        Measure respiration rate
                  If choose to measure EKG:
                        Measure EKG
```

```
Else if:
     If user is in annunciation menu:
          Call dynamic scheduler
Else if:
     If user is in measurement menu:
          Call dynamic scheduler
```

Scheduler function is coupled with issue function where scheduler schedules task and send their information to issue. Then issue function issueses them.

```
Scheduler():
     Set task queue to be empty

     Check ekg capture flag and time constraint to add ekg
     capture task to task queue
     Check measure flag and time constraint to add measure
     task to task queue
     Check ekg process flag and time constraint to add ekg
     process task to task queue
     Check compute flag and time constraint to add compute
     task to task queue
     Check warning flag and time constraint to add warning
     task to task queue
     Check display flag and time constraint to add display
     task to task queue
     Check remote communication flag and time constraint
     to add remote communication task to task queue
     Check status flag and time constraint to add status
     task to task queue
     Check remote display flag and time constraint to add
     remote display task to task queue

     Execute each task in the task queue
```

To support the measurements of health data modeled by analog and digital signals, we implement the following functions in peripheral system:

```
setup():
     Setup pinMode
     Attachinterrupt cuff function
     Attachinterrupt pr_detect function

main()(loop):
     Debounce switch
     Debounce button

     Read inbyte
     Read function selection
     Read end byte
```

```
    If pulse rate selected
        attach pr_detect to interrupt
        Detach pr_detect from interrupt

    If temperature selected
        Measure temperature

    If blood pressure selected
        Measure blood pressure

    Write start byte
    Write measure data
    Write end byte

cuff():
    If switch is in up state
        pressure*1.1
        If pressure > 150
            Pressure = 150
    Else
        Pressure * 0.9
        If pressure < 50
            Pressure = 50

pr_detect():
    detect if we are measuring pulse rate
    if we are measuring the pulse rate
        set a new clock
        while clock != 1s
            start counting number of pulse
    pulse rate = count

respiration rate detect:
    detect if we are measuring respiration rate
    if we are measuring the respiration rate
        set a new clock
        while clock != 1s
            start counting number of pulse
    respiration rate = count

measure():
    If measure temperature
        Read signal from temperature pin
        Delay 5 seconds
    If measure blood pressure
        If cuff pressure is between 110 to 150
            Measure systolic pressure from analog port
```

```
          Set systolic flag
     If cuff pressure is between 50 to 80
          Measure diastolic pressure from analog
          port
          Set diastolic flag
```

Due to limit memory in Uno, we implemented EKG measurement in Mega with following functions:

```
ekg_capture():
     Set EkgCaptureFlag
     For (int i= 0; i < 256; i++) {
          Read from analog
          Offset by 2.5V
          Map the reading to -31 to 32 range
          Wait for sampling period
     }
     Set EkgProcessFlag

ekg _process():
     Index = max magnitude index found by provided optfft
     Ekg = index * sampling frequency / 256(buffer length)
     Write to ekg buffer
     Set EkgPricessFlag to false
```

To support remote communication, remote control and remote display capability, we implement the following functions:

```
Remote Communication Data:
     Temperature display data
     Systolic pressure display data
     Diastolic pressure display data
     Pulse rate display data
     Respiration rate display data
     Ekg display data
     Temperature warning
     Display warning
     Pulse rate warning
     Respiration rate warning

Command Management Data:
     Temperature data
     Systolic pressure data
     Diastolic pressure data
     Pulse rate data
     Respiration rate data
     Ekg data
     Temperature display data
     Systolic pressure display data
```

```
      Diastolic pressure display data
      Pulse rate display data
      Respiration rate display data
      Ekg display data
      Temperature warning
      Display warning
      Pulse rate warning
      Respiration rate warning

Remote Communication Data Initial Value:
      Temperature display data = 78;
      Systolic display data = 78;
      Diastolic display data = 67;
      Pulse rate display data = 67;
      Respiration display data = 45;
      Ekg display data = 56;

Command Management Data Initial Value:
      Temperature data = NULL;
      Systolic pressure data = NULL;
      Diastolic pressure data = NULL;
      Pulse rate data = NULL;
      Respiration rate data = NULL;
      Ekg data = NULL;
      Temperature display data = 78;
      Systolic display data = 78;
      Diastolic display data = 67;
      Pulse rate display data = 67;
      Respiration display data = 45;
      Ekg display data = 56;
      Temperature warning = FALSE;
      Display warning = FALSE;
      Pulse rate warning = FALSE;
      Respiration rate warning = FALSE;

command():
      Receive command;
            If number of command is larger than 2
                  Read command
            If command is i
                        Enable command
            Else if command is s
                        Enable system
            Else if command is p
                        Stop system
            Else if command is d
                        If display is off
```

```
                                    Turn on
                        Else
                  turn off
             else if command is m
                        update data
             else if command is w
                        update warning
             Else
                   give error

remoteCom():
     if update data
                 update all data
     if update warning
                 update all warning
     if not update data && not update warning
                 use old data

remoteDis():
     read temperature;
     print temperature;
     read systolic pressure
     print systolic pressure
     read diastolic pressure
     print diastolic pressure
     read pulse rate
     print pulse rate
     read respiration rate
     print respiration rate
     read ekg
     print ekg
     read warning
     if temp warning
          flash temp
     if systolic warning
          flash systolic
     if diastolic warning
          flash diastolic
     if pulse rate warning
          flash pulse rate
     if respiration rate warning
          flash respiration rate
```

To support the display of critical data, we made changes to display function and implement new function as follow:

```
display():
     If the measurement is in warning state
          Print nothing and skip that space
     Else
          Print out the information in format
     If acknowledge button is pressed, reset acknowledge
     flag

drawDisplay():
     Print out the formatted information without
     considering warning

flash():
     If temp measured
          If temp is in warning state
               Flash temp 2 second period
               Leave space for temp to display
     If blood pressure measured
          If blood pressure is in warning state
               Flash bp 1 second period
               Leave space for bp to display
     If pulse rate measured
          If pulse rate is in warning state
               Flash pr 0.5 second period
               Leave space for pr to display
```

<u>Execution Time</u>

For the first run, due to the cold start:

- Measure: 16µs
- Compute: 236µs
- WarningAlarm: 88µs
- Display: 718676µs
- Status: 12µs
- EKGCapture: 2.5ms
- EKGProcess: 17µs

For the rest run (almost all of rest run):

- Measure: 8µs
- Compute: 8µs
- WarningAlarm: 80µs
- Display: 8µs
- Status: 8µs
- EKGCapture: 2.5ms
- EKGProcess: 8µs

Annunciation will take 2 seconds to process.

However, due to the hierarchy of built-in cache, conflict miss or capacity miss, the time is varied. The probability to have such situation is pretty small.

Timing constraints

Due to the refresh rate of the screen and computation rate, the delay between each display should be larger than 10ms

EKG Measurement is hard real time task, we measure each sample at rate of 100ms to make capture possible at max input frequency of 3.5KHz.

Due to the scheduler algorithm, the remote control command can only be transmitted and updated to kernel when data is refreshed on putty.

Error handling

We try to give all data a flag to indicate their behavior. Out of Range or not, High, Low. The reason we have so many flags is we can simplify our logic to avoid losing significant logic in controlling Warning Alarm and Measure function.

In addition, our user interface has forced users to operate under our process procedure where they have to choose things to measure, and return to top menu and press annunciation to start, otherwise it the whole system will not awake to start working.

Software Implementation

Functional Decomposition Diagram for the System Control

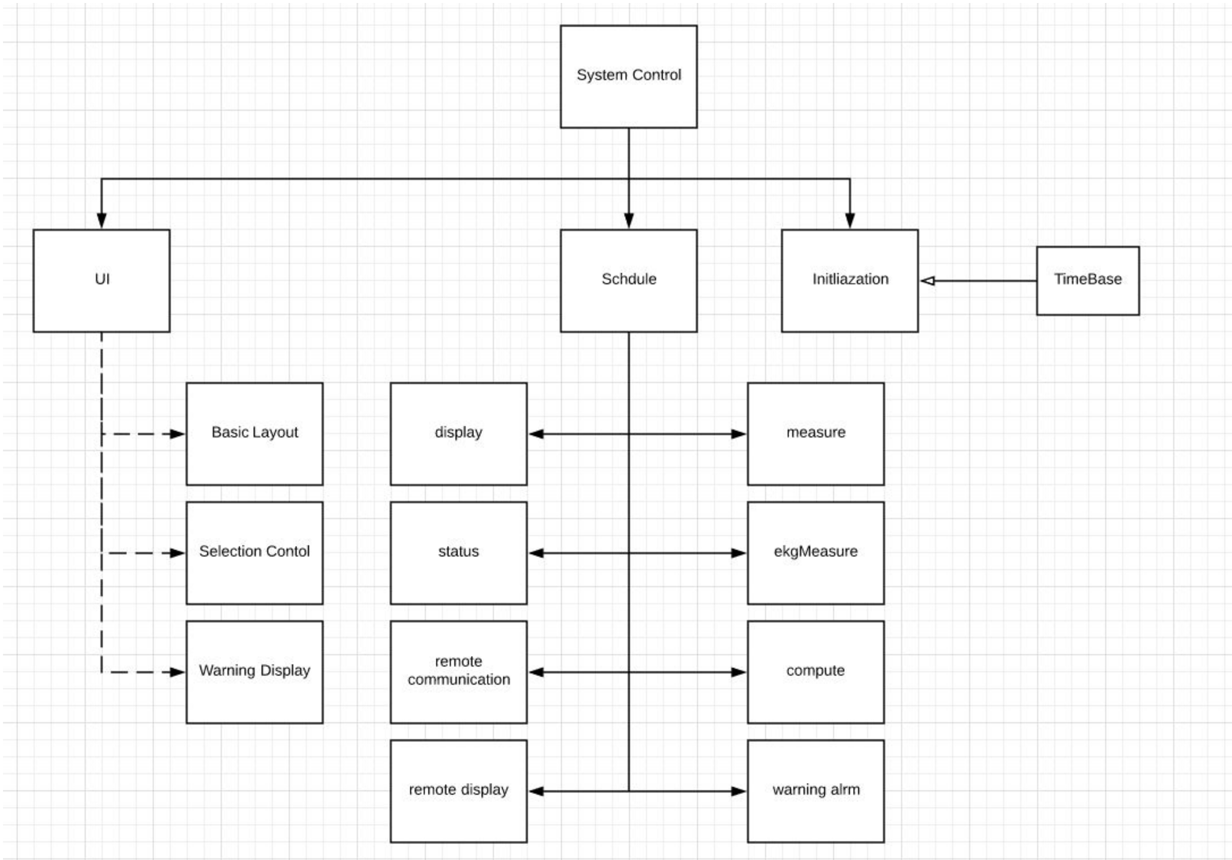

Figure 2 System Control Functional Decomposition Diagram

System Control Task/Class Diagrams

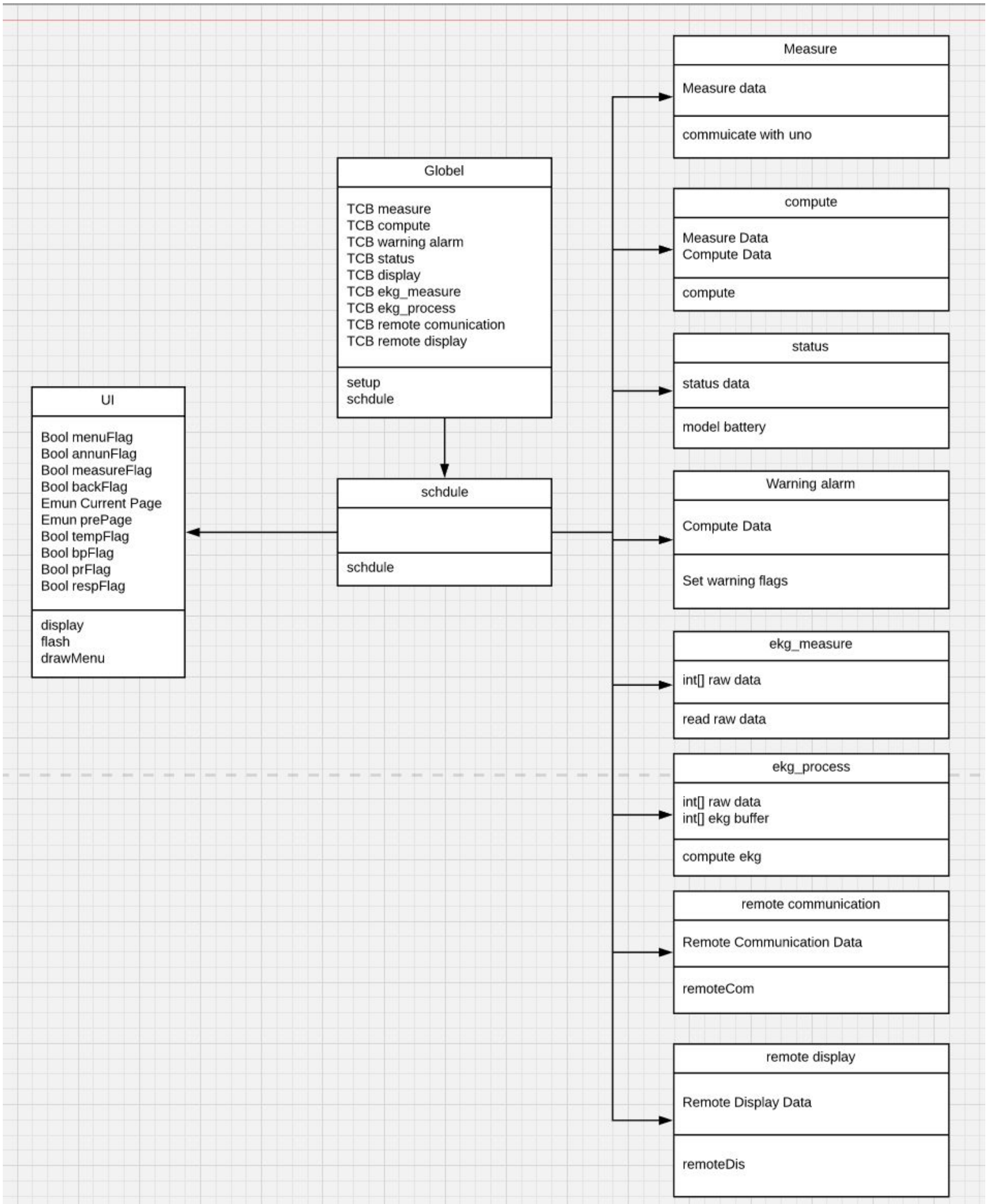

Figure 3 System Control task/class diagrams

System Control Activity Diagrams

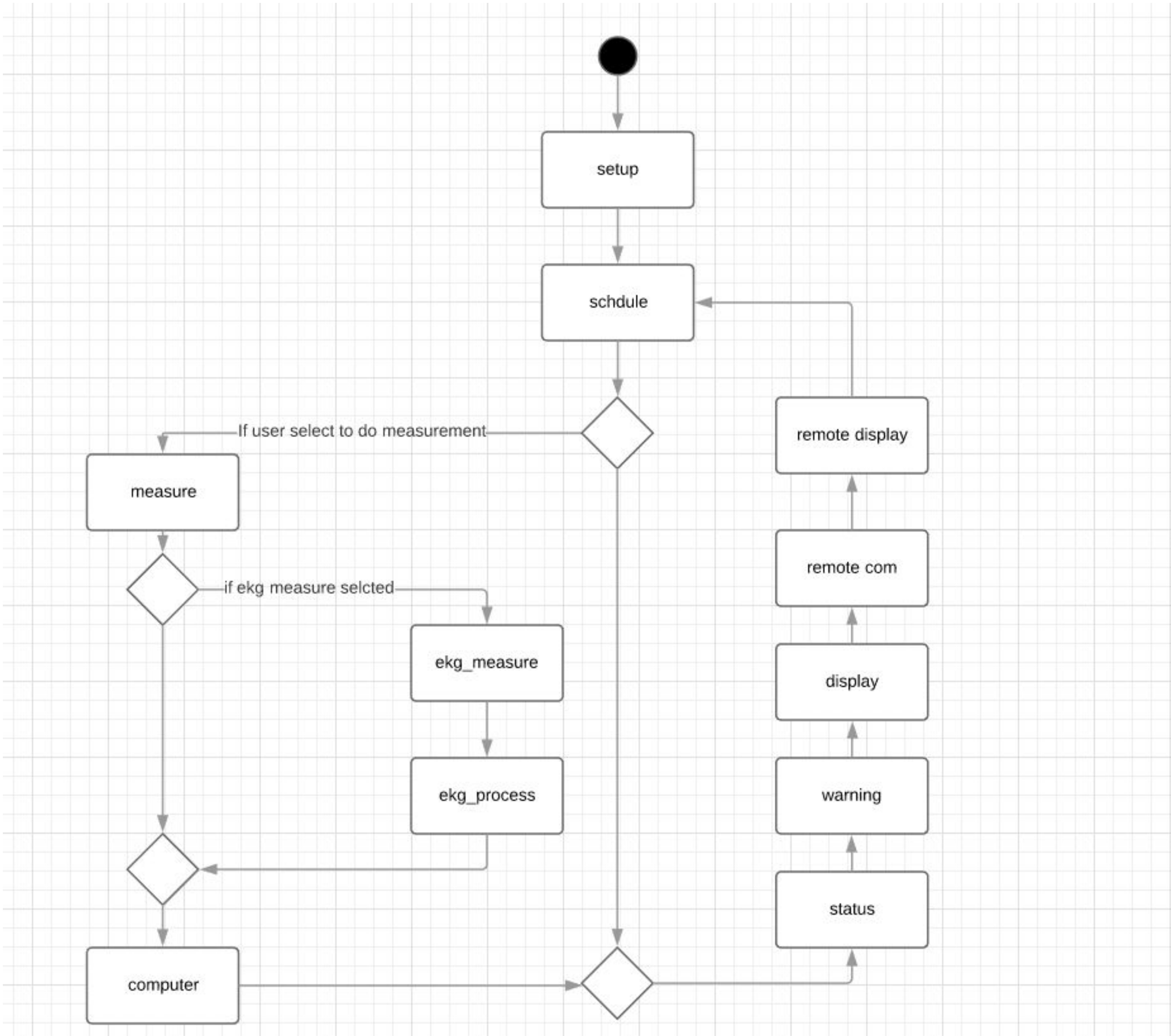

Figure 4 System Control Communication Task Diagrams

Functional Decomposition Diagram for the Peripheral Subsystem

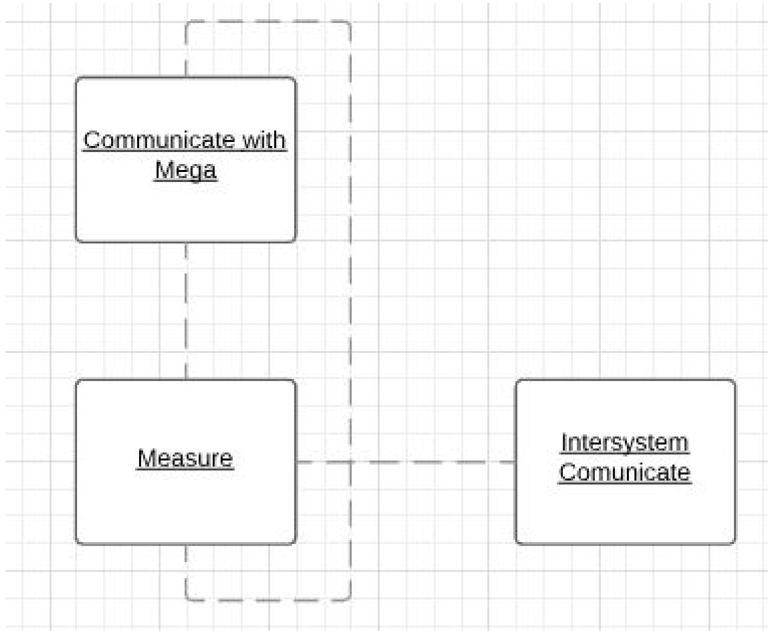

Figure 5 Peripheral Subsystem Functional Decomposition Diagram

Peripheral Subsystem Task/Class Diagrams

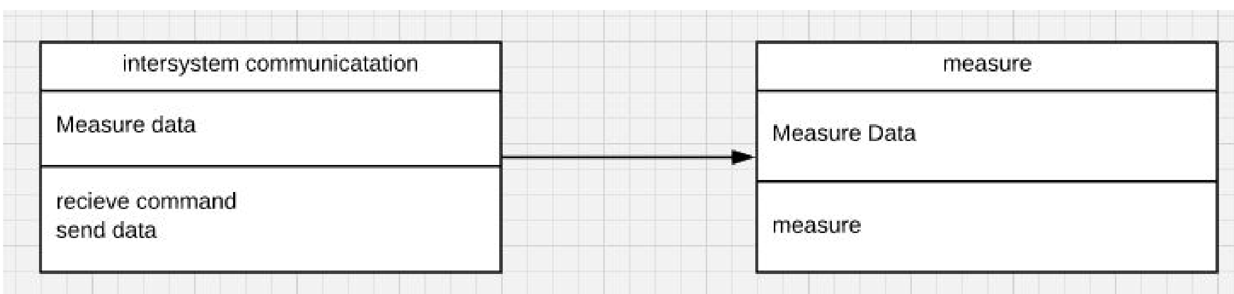

Figure 6 Peripheral Subsystem task/class Diagrams

Peripheral Subsystem Activity Diagrams

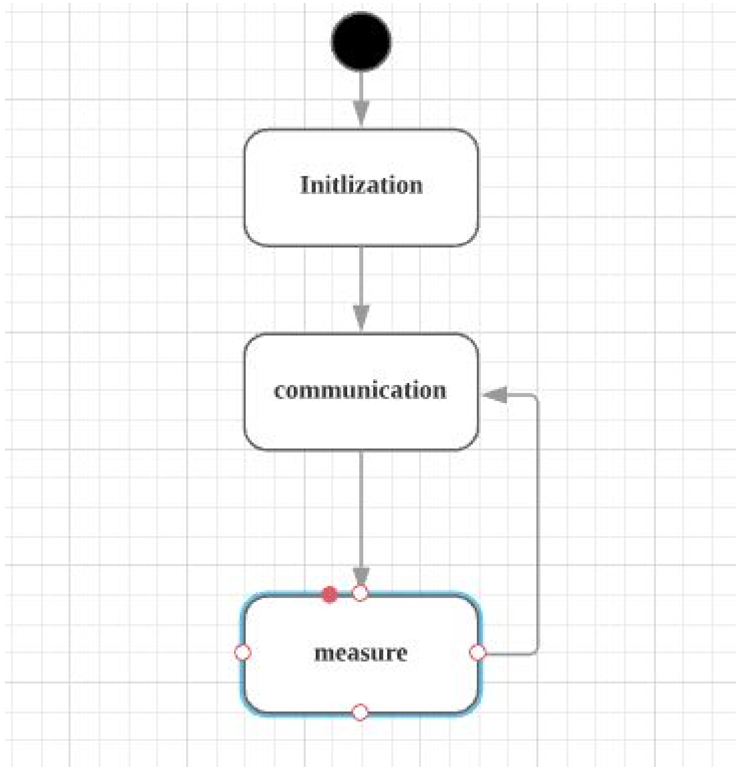

Figure 7 Peripheral Subsystem Activity Diagrams

Startup Sequence Diagram:

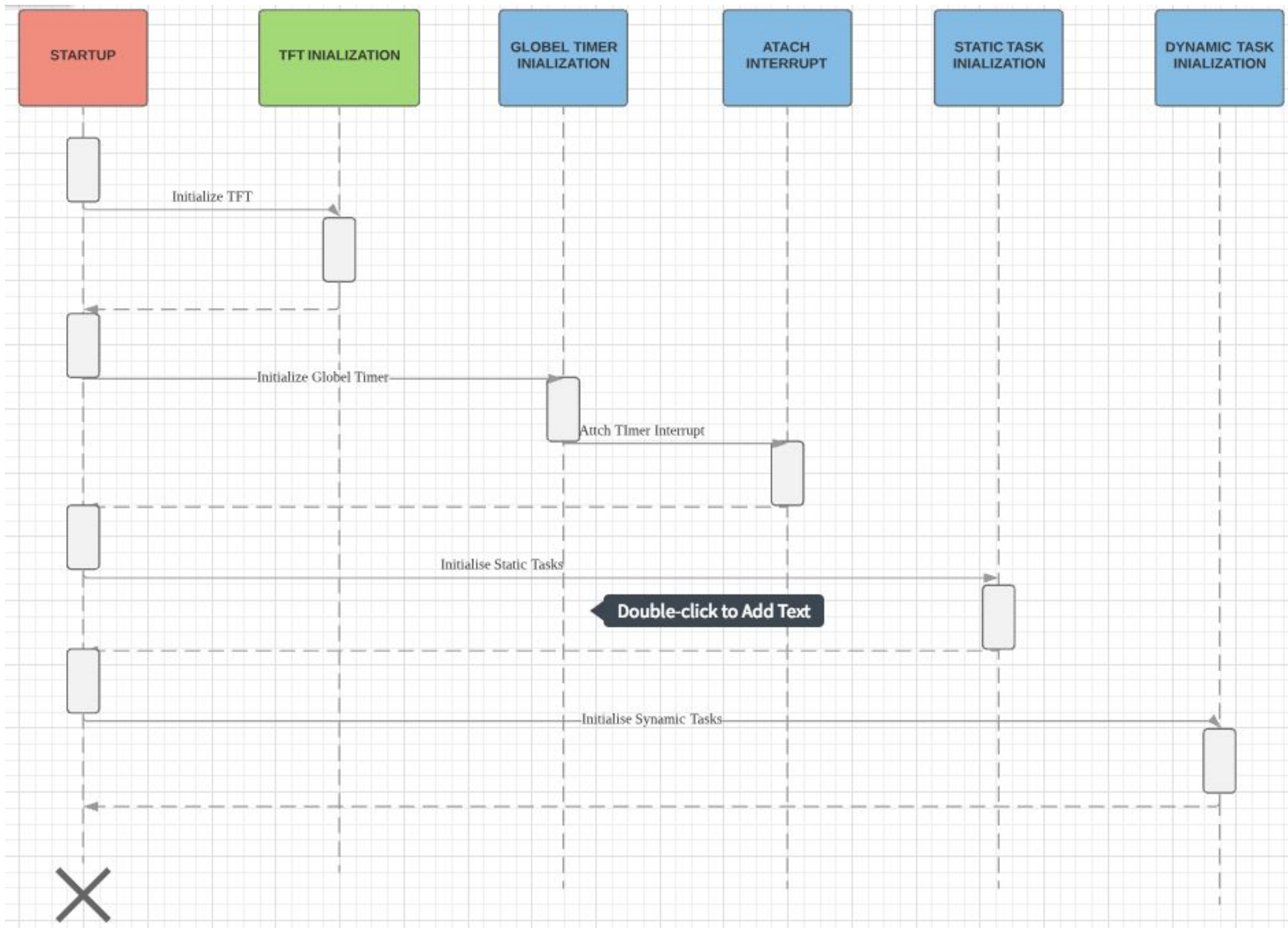

Figure 8 Startup Sequence Diagram

Static Tasks are given the same priorities since all static taks is based on TCBs and will execute in the same sequence every time.

<u>Schedule State Chart:</u>

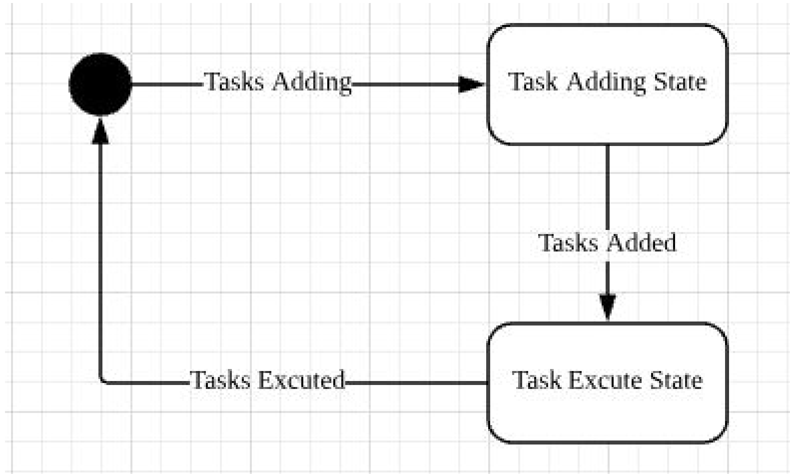

Figure 9 Schedule State Chart

Display Front Panel Diagram:

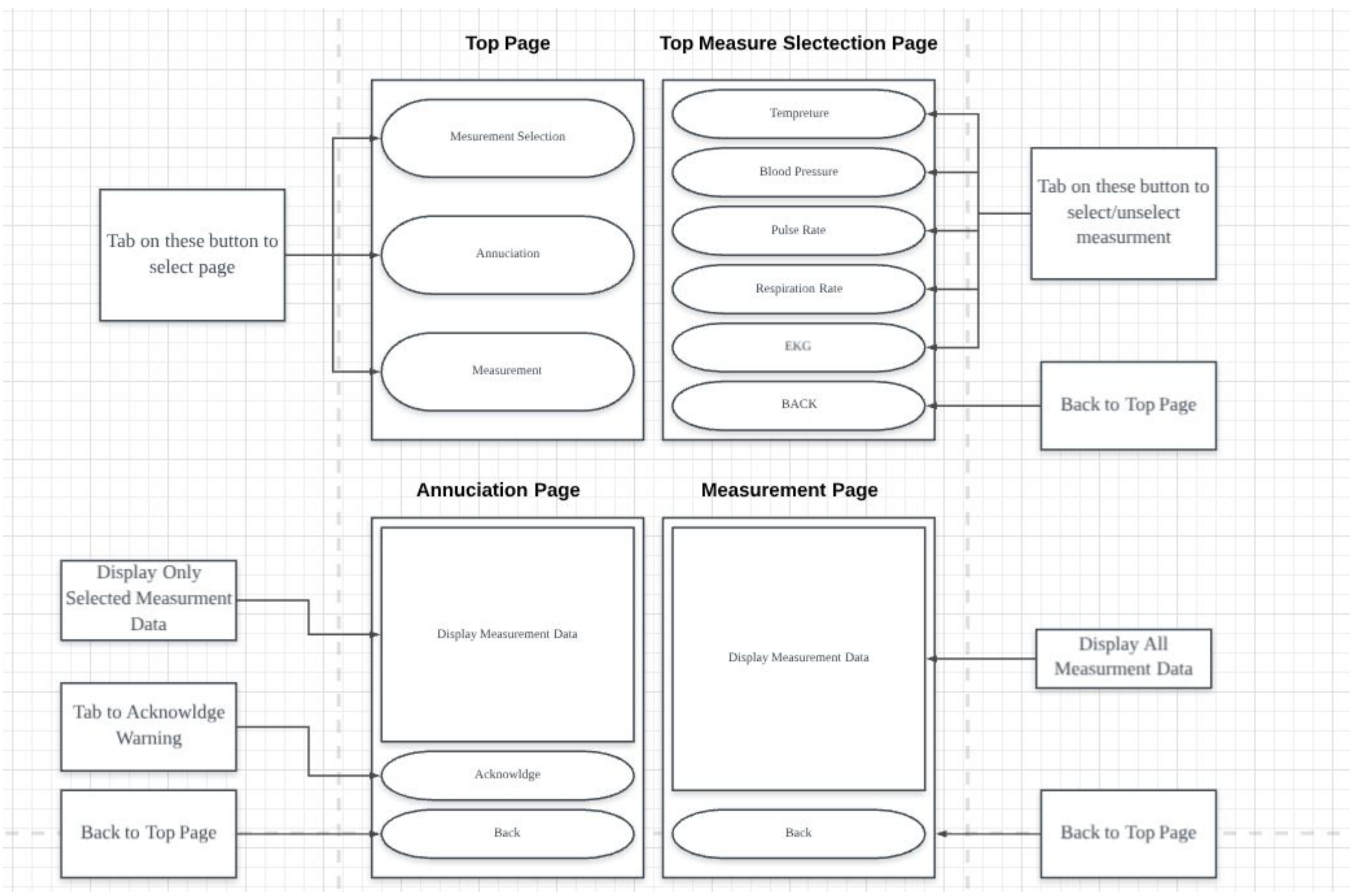

Figure 10 Display Layout

Remote Web Page Layout:

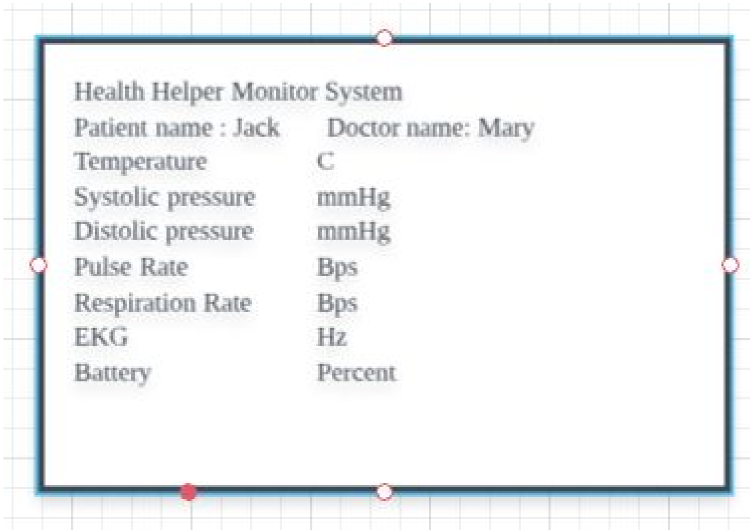

Figure 11 Remote Web page Layout

3. Presentation, discussion, and analysis of the results

Overview:
The final product has critical checkpoints which need to be confirmed to ensure the correctness of the implementation. First, temperature, blood pressure, and respiration measurements are switch from software modeling to analog input modeling. We need to ensure the correctness of each measurement. Also, for blood pressure measurements, we need to ensure the modeled blood pressure cuff is working as expected. Second, we need to ensure the display function flash the warning measurement in correct rate. Third, we need to ensure the EKG measurement works as expected. At last, we need to ensure the PuTTY remote control and communication works as expected. The rest of rules from previous lab remains in effect. Since there is no alternation made on those parts of implementation, we ensure the correctness by inspection and the result will not be included in this lab report. Please refer "EE474 Lab3 report" for more information.

Temperature Measurement:
The temperature measurement is now measured from analog signal and should be map to human body temperature range.

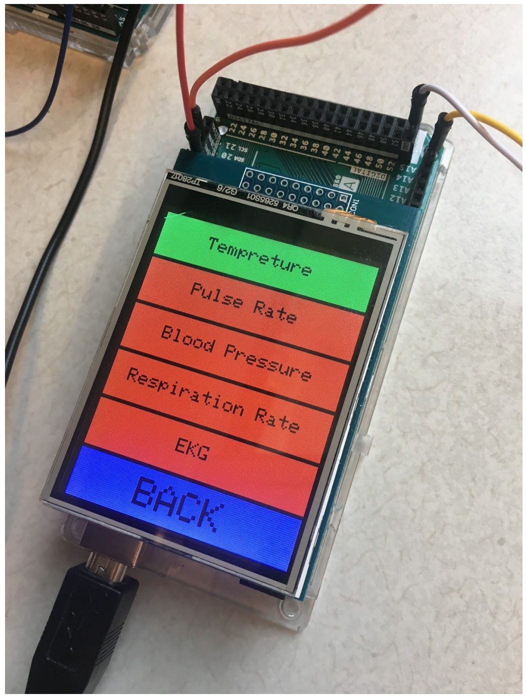

Figure 12 Select To Measure Temperature

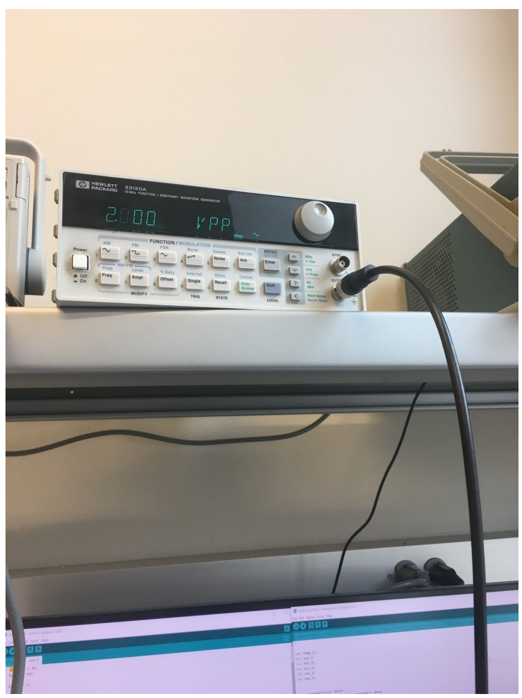

Figure 13 Temperature Signal Generated by Signal Generator

Figure 14 Temperature Reading with Correct scaling

As figure 12 to 14 demonstrated, temperature measurement is selected on mega, got schedule correctly, read input from signal generator, scale correctly and displayed on the TFT panel with correct waring color.

<u>Blood Pressure Measurement:</u>
The blood pressure measurement is now modeled by analog signal. To determine whether to perform systolic or diastolic measurement, the modeled blood pressure cuff should be inflated or deflated to certain range. In this project, the blood pressure cuff is modeled by circuit with a switch and a button. If the switch is at up state, press the button will increase the pressure by 10%. If the switch is at down state, press the button will decrease by 10%.

After connect the signal generator to the correct pin, the switch is turned to up state.

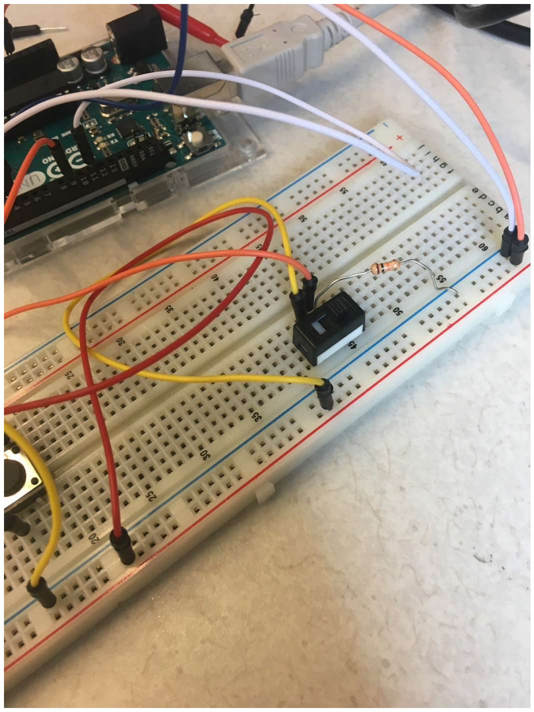

Figure 15 Switch is at Up State

After the button was pressed few time to get the pressure into the range of systolic pressure reading, the system read in the signal and scaled it correctly as Figure 12 demonstrated.

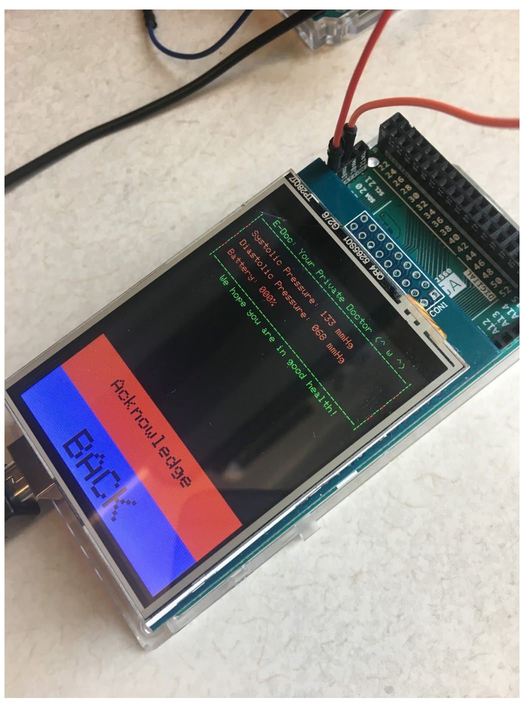

Figure 16 Systolic Pressure Reading Updated

Then, the switch is turn to down state for diastolic pressure measurement.

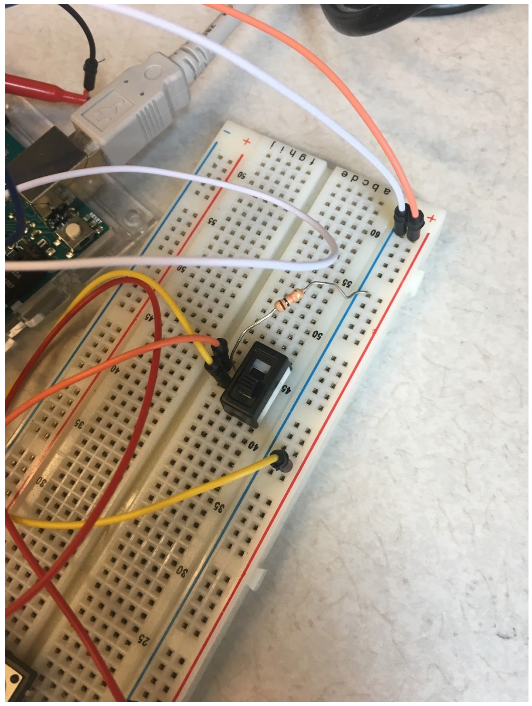

Figure 17 Switch is at Down State

After the button was pressed few time to get the pressure into the range of diastolic pressure reading, the system read in the signal and scaled it correctly as figure 14 demonstrated.

Figure 18 Systolic Pressure Reading Updated

Respiration Measurements:
The respiration rate measurement is now measured from analog signal and should be map to human respiration rate range. The input frequency is corresponding with the output frequency as demonstrated by Figure 19 to 22.

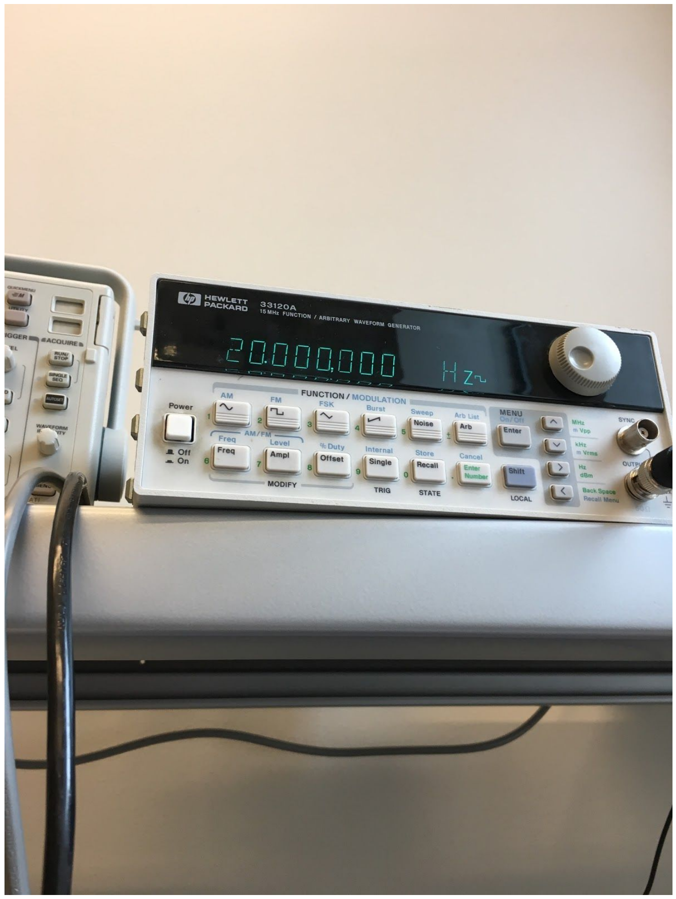

Figure 19 Set Input Frequency to 20Hz

Figure 20 Respiration Rate Matches with Input

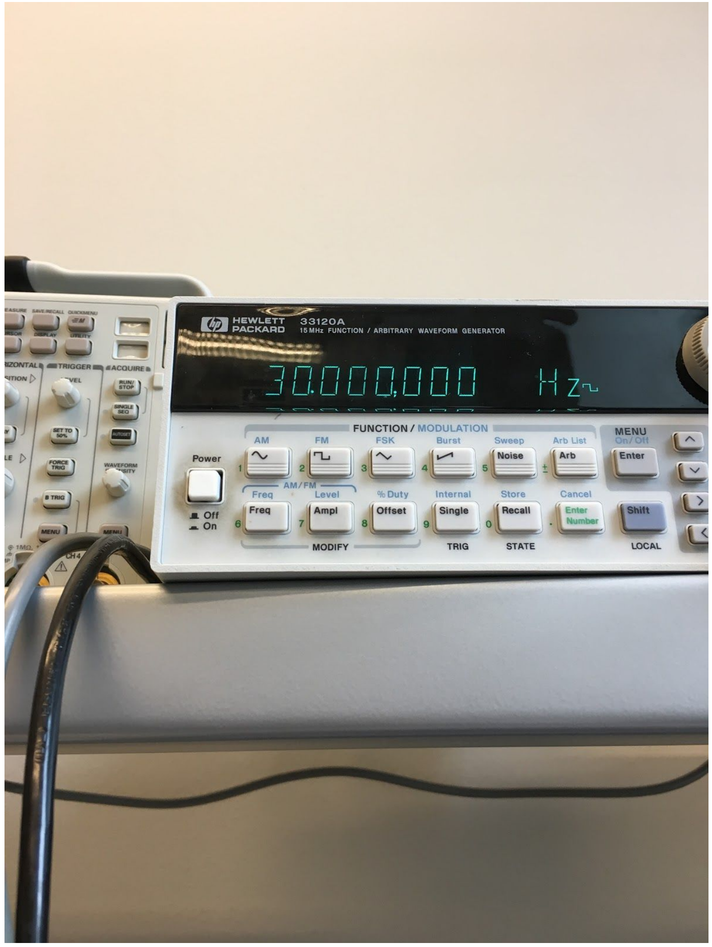

Figure 21 Set Input Frequency to 30Hz

Figure 22 Respiration Rate Matches with Input

<u>Flash On Warning:</u>
Figure 23 to 28 demonstrated that the temperature is flashed at 1 second period. The temperature display went from on to off in one second and back on at the beginning of the next second.

Figure 23 Temperature is On at 16:52:21

Figure 24 Temperature is off at 16:52:21

Figure 25 Temperature is off at 16:52:22

Figure 22 to 24 demonstrated that the pulse rate is flashed at 2 second period. The pulse rate display stay on for one second stay off at the next second.

Figure 26 Pulse Rate is on at 16:52:21

Figure 27 Pulse Rate is off at 16:52:22

Figure 28 Pulse Rate is on at 16:52:23

Figure 29 to 32 demonstrated that the Blood Pressure is flashed at 0.5 second period. The Blood Pressure display went from on to off to on to off in one second.

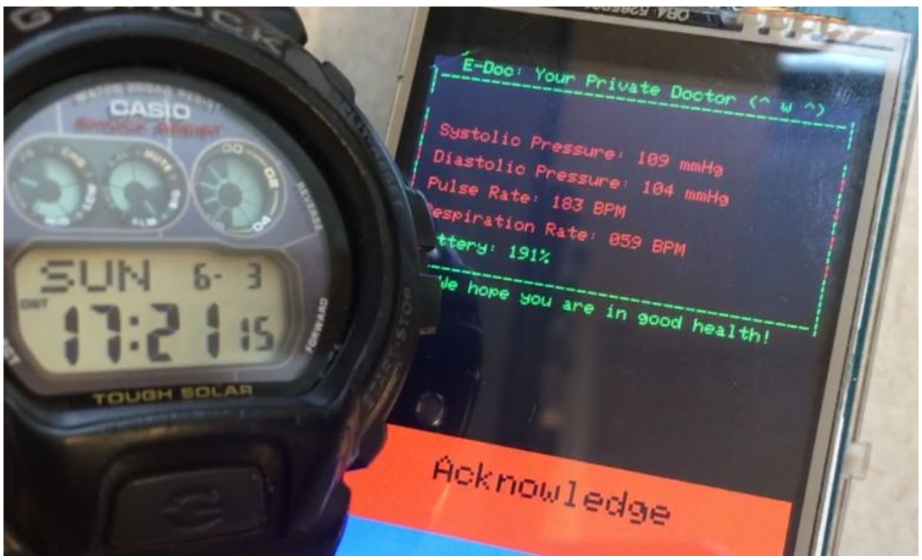

Figure 29 Systolic Pressure is on at 17:21:15

Figure 30 Systolic Pressure is on at 17:21:15

Figure 31 Systolic Pressure is on at 17:21:15

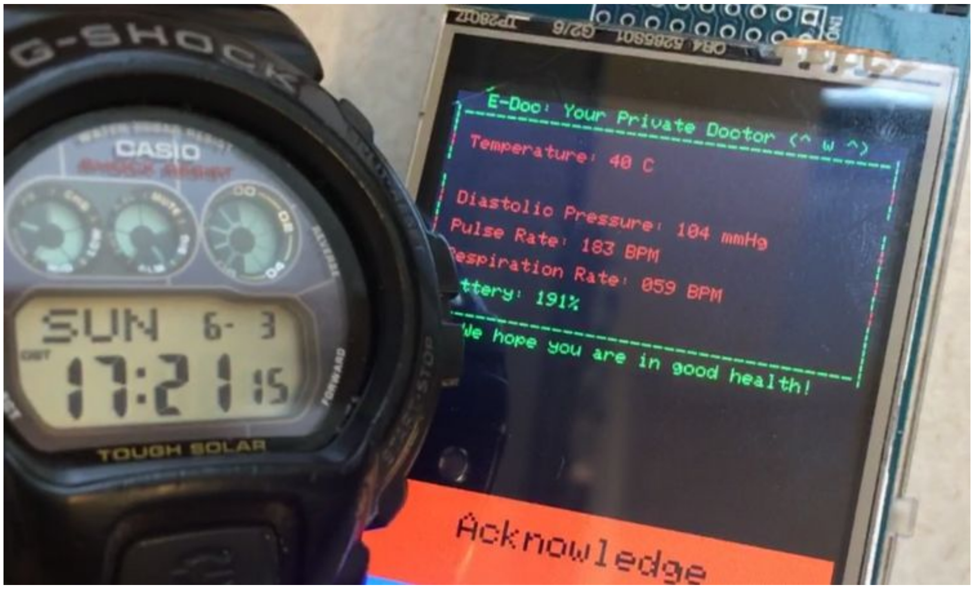

Figure 32 Systolic Pressure is on at 17:21:15

<u>EKG Measurement:</u>

For the input frequency range from 35Hz to 3.75kHz, the system is able to measure, store and process the input signal and produce EKG data corresponding to the input frequency. Figure 33 to 36 demonstrates that the EKG reading is corresponding to the input frequency.

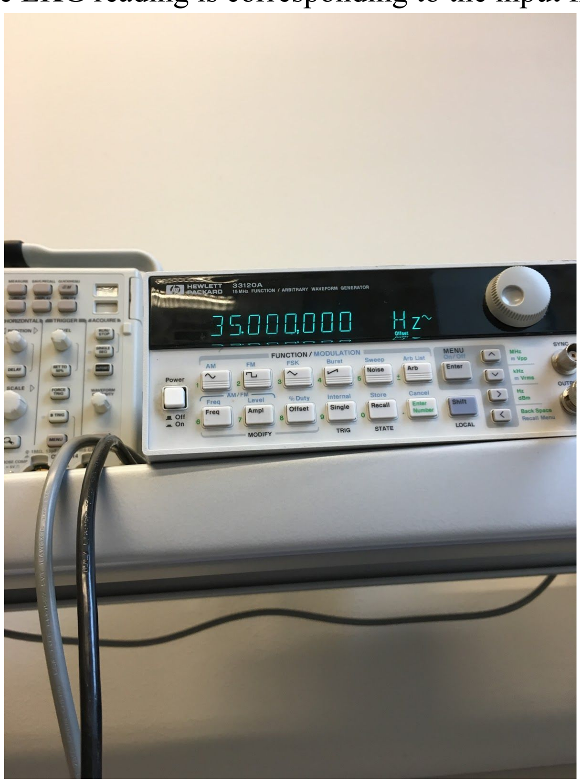

Figure 33 Input Frequency is Set to 35 Hz

Figure 34 EKG Reading is 23 Hz

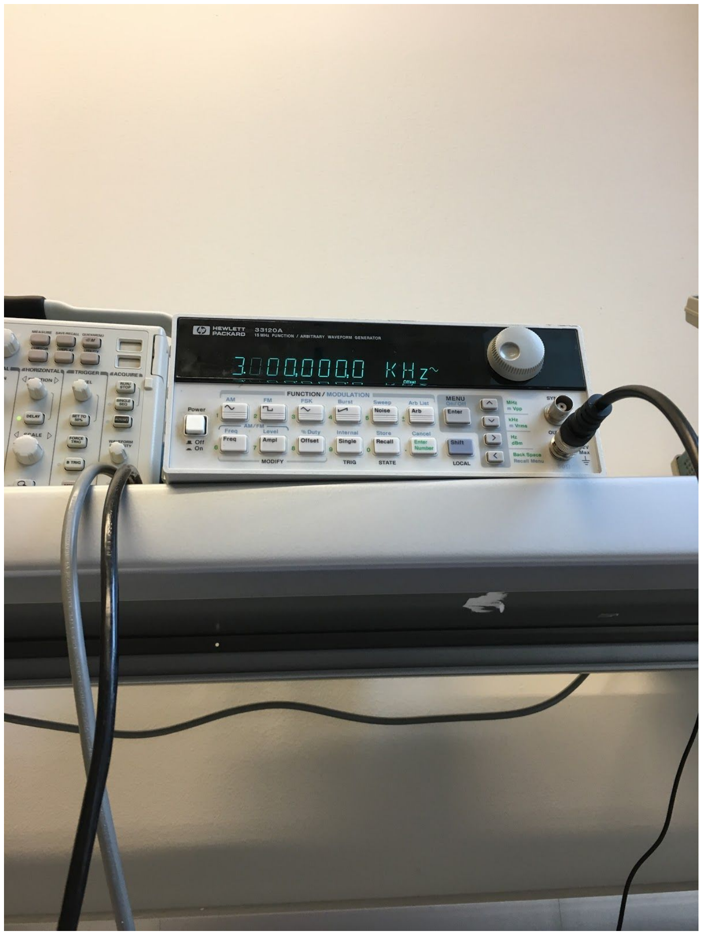

Figure 35 Input Frequency is Set to 3000 Hz

Figure 36 EKG Reading is 2783 Hz

<u>Remote Command:</u>
Figure 37 demonstrates that enter "I" command on PuTTY terminal initializes the network communications between your system and the browser page.

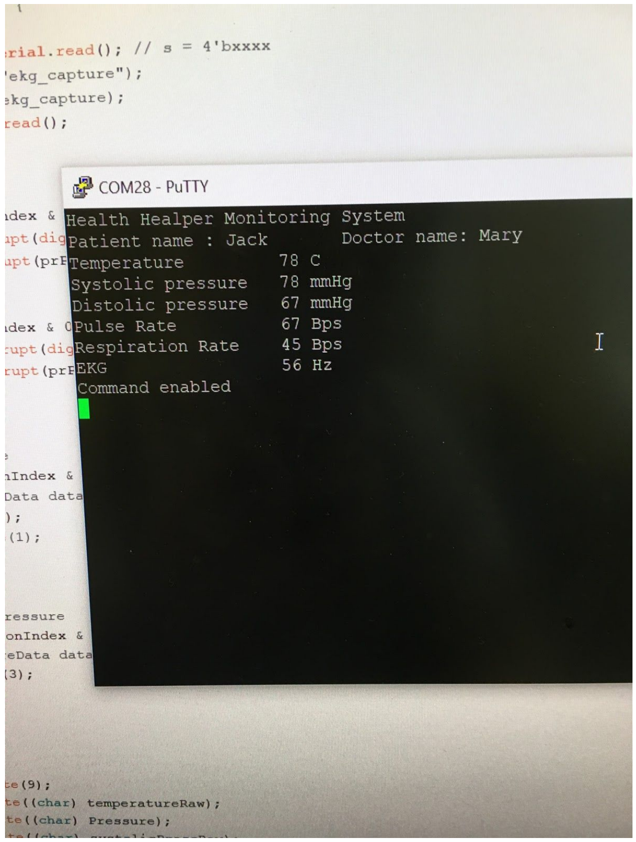

Figure 37 Remote Computer System Initiation Prompt

Figure 38 demonstrates that the E error response is given is given for incorrect commands or non-existent commands on PuTTY terminal.

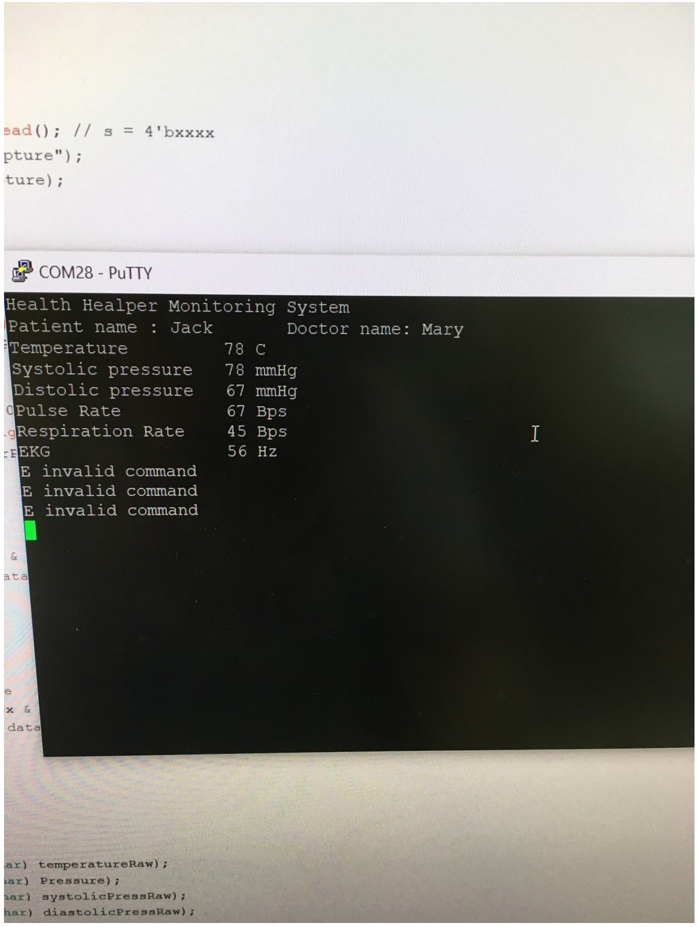

Figure 38 Remote Computer System Invalid Command Prompt

Figure 39 demonstrates that enter "S" command indicates START mode. The command shall start the embedded tasks by directing the hardware to initiate the measurement tasks.

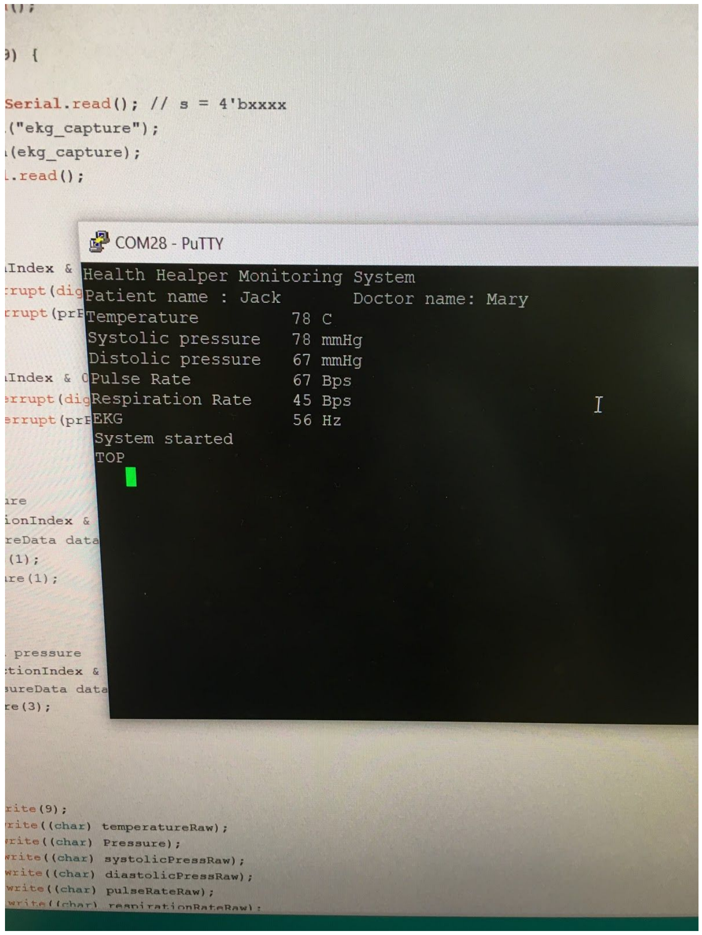

Figure 39 Remote Computer System Task Started Prompt

Figure 40, 41 demonstrates that enter “P” command indicates STOP mode. This command shall stop the embedded tasks by terminating any running measurement tasks.

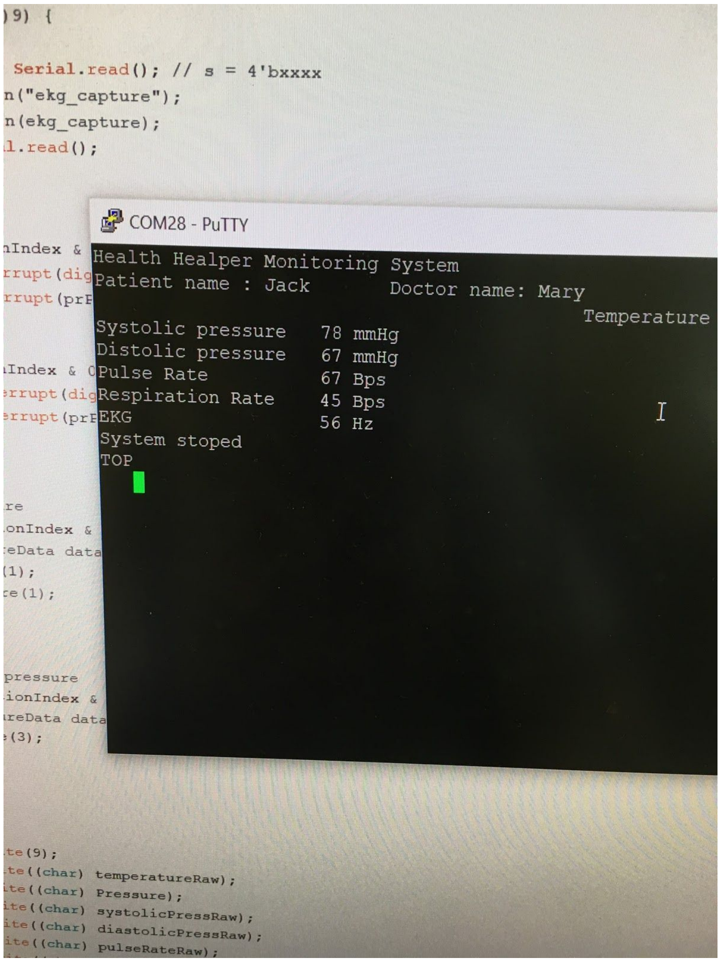

Figure 40 Remote Computer System Task Stopped Prompt

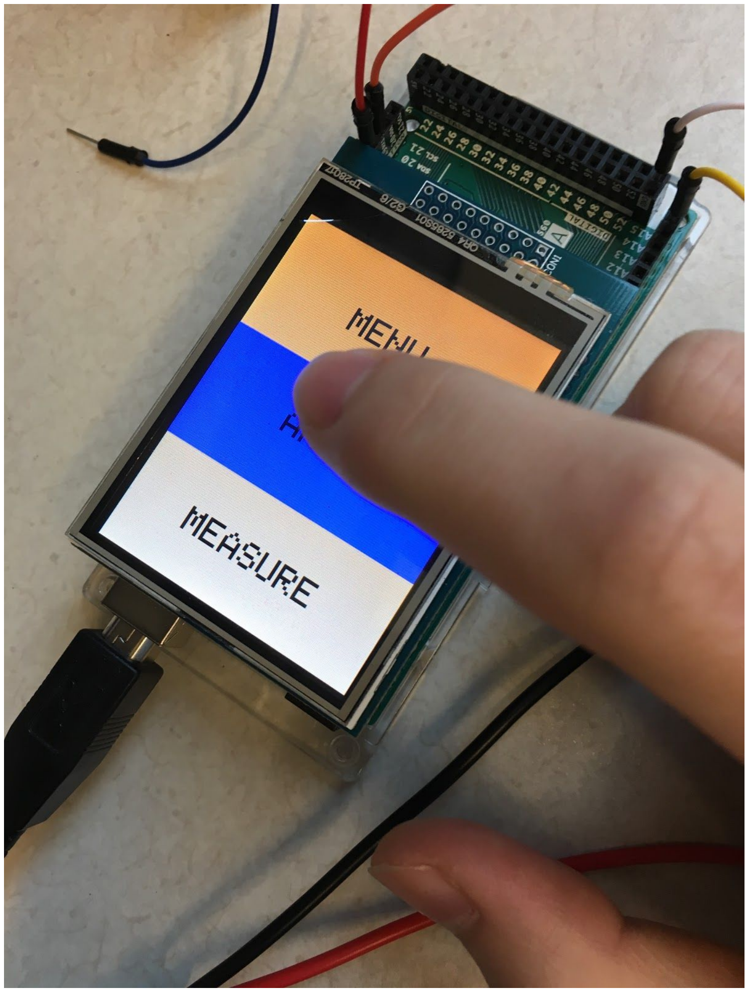

Figure 41 System no longer Responsive to Touch After Task Stopped

Figure 42 to 45 demonstrates that enter “D” command enables or disables the local display.

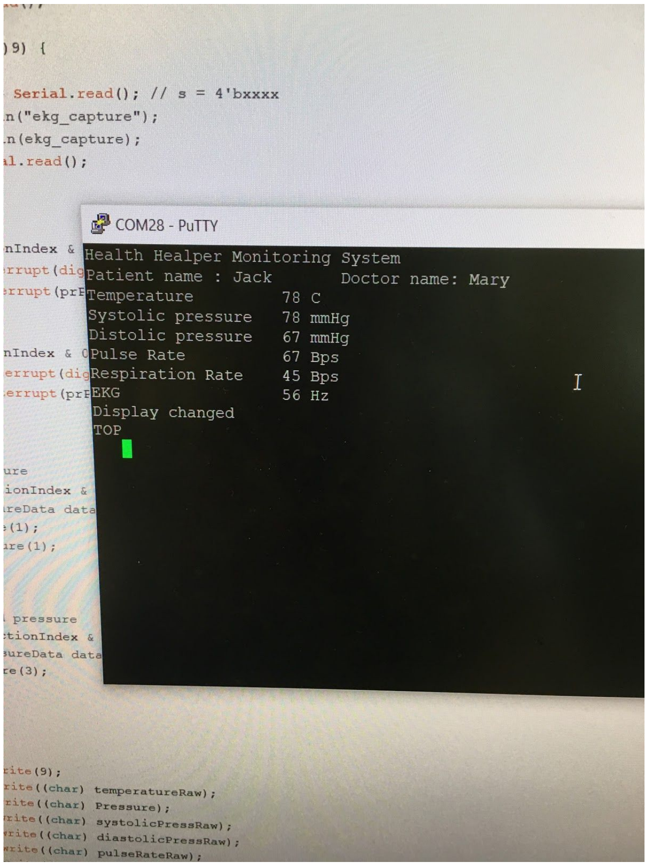

Figure 42 Remote Computer System Display Changed Prompt

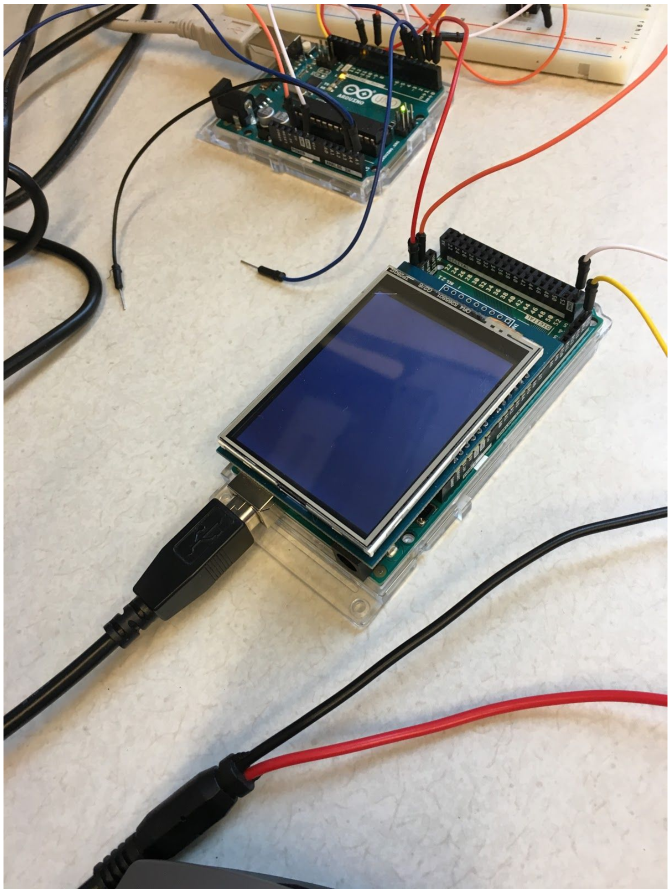

Figure 43 System Display is Off After the D Command

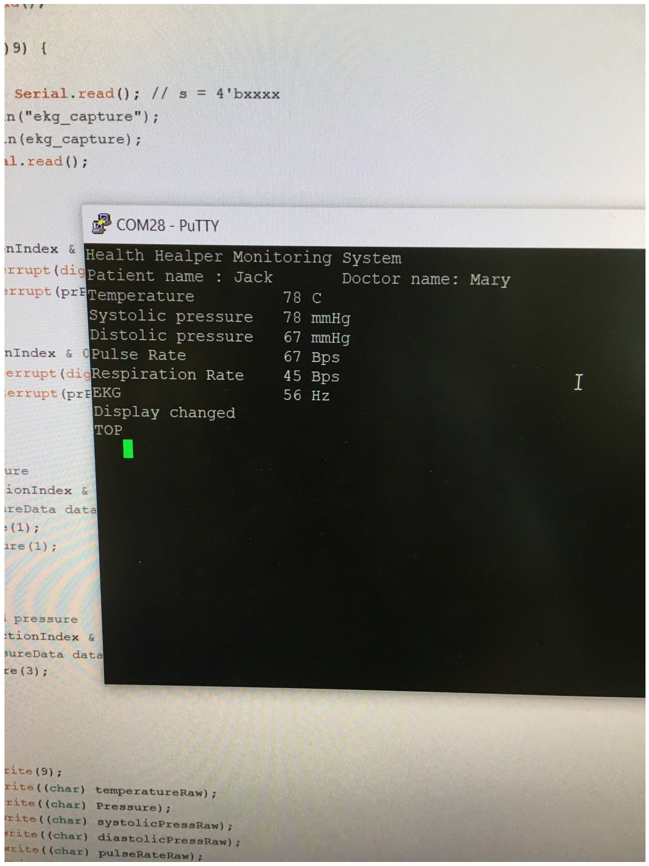

Figure 44 Remote Computer System Display Changed Prompt

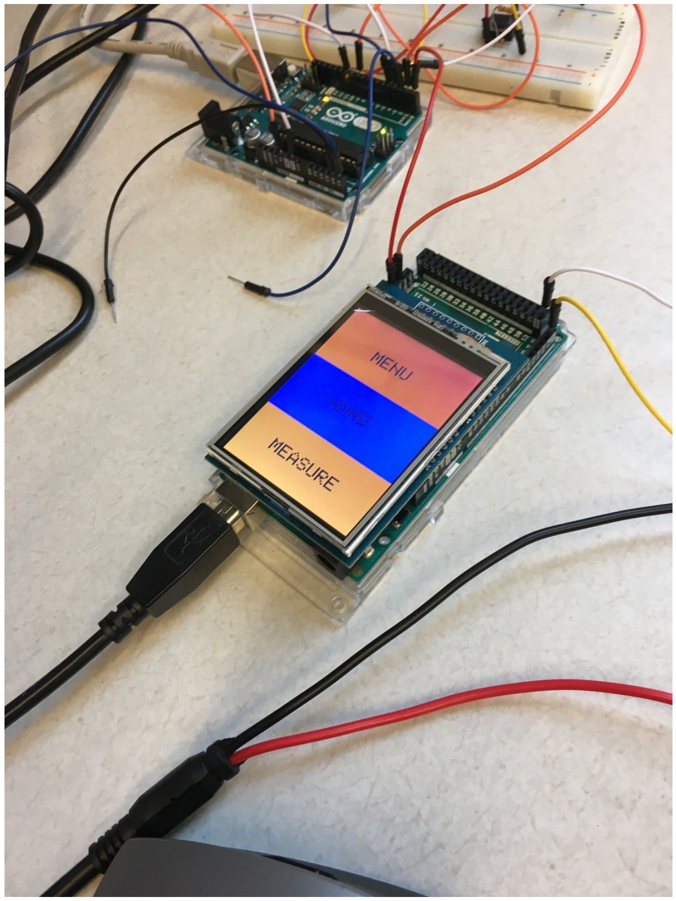

Figure 45 System Display is On After Reenter the D Command

Figure 46 demonstrates that enter “M” command return the the most recent value(s) of all measurement data

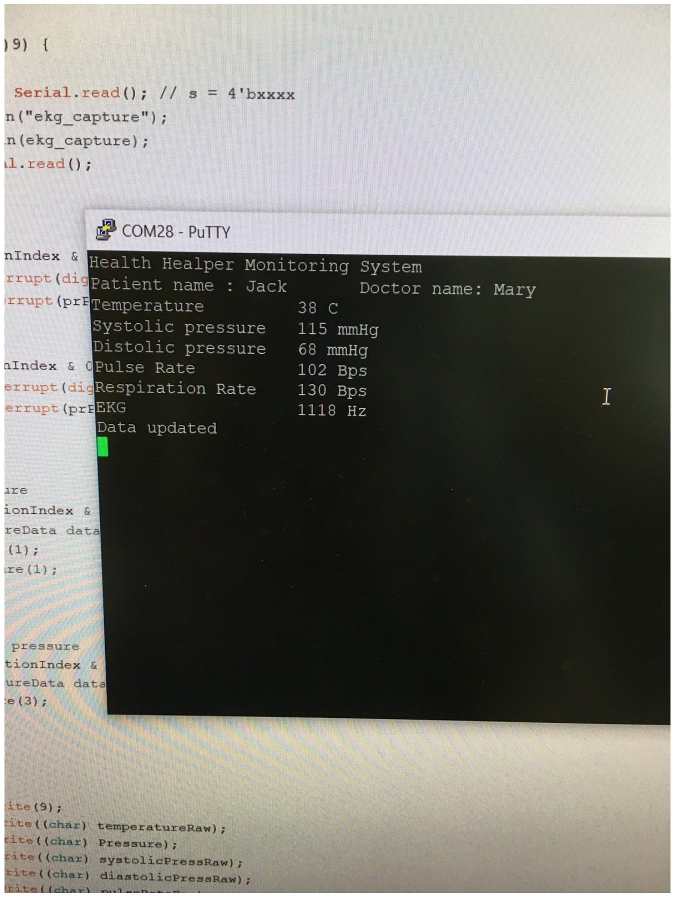

Figure 46 Remote Computer System Data Updated Prompt

Figure 47 demonstrates that enter “W” command indicates return the most recent value(s) of all warning and alarm data.

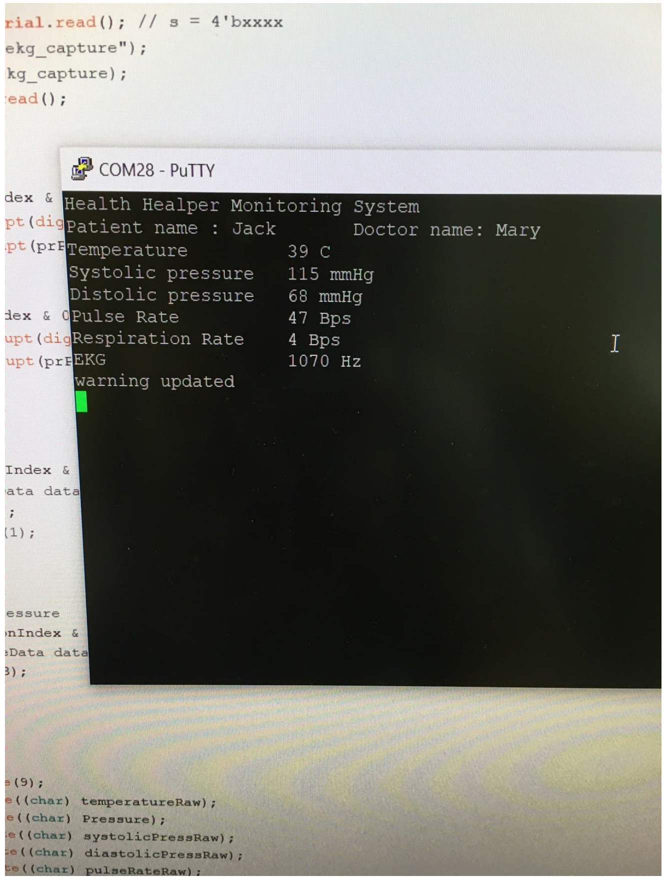

Figure 47 Remote Computer System Display Warning Updated Prompt

<u>Remote Display:</u>
Figure 48 demonstrates that the remote display displays all required information which includes:

- The name of the product
- The patient's name
- The doctor's name.
- Temperature: <temperature> C
- Systolic pressure: <systolic pressure> mm Hg
- Diastolic pressure: <diastolic pressure> mm Hg
- Pulse rate: <pulse rate> BPM
- Respiration rate <pulse rate> BPM
- EKG < Measured Frequency reading > Hz
- Battery: <charge remaining>

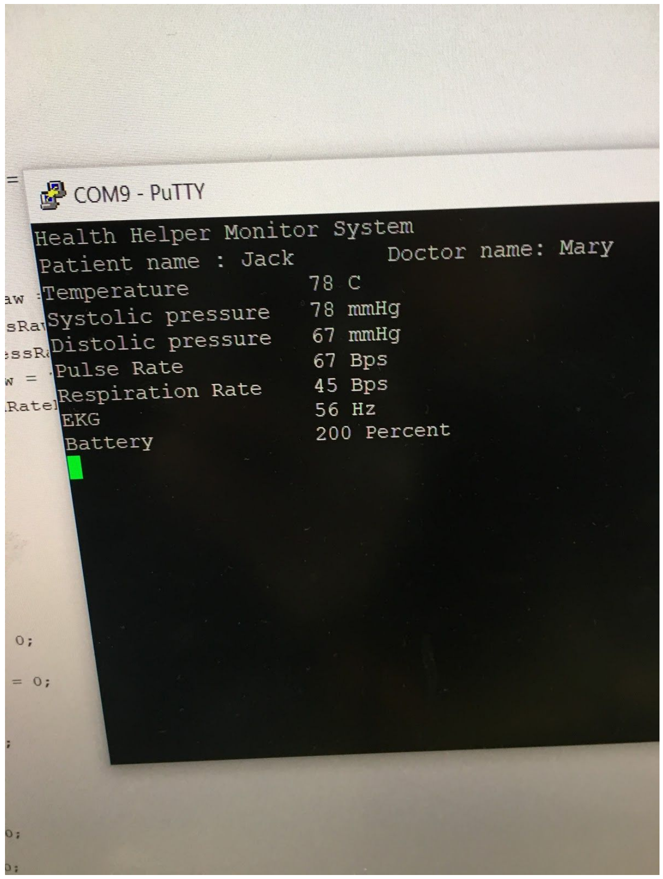

Figure 48 Remote Computer Terminal With Required Information Displayed

4. Analysis of errors needed to be resolved

Our design is error-free and we do not have any unresolved error to be fixed.

5. Analysis of errors encountered

1. Memory Size Limit
   Cause: The dynamic Memory Size of UNO is only 2048 bytes, and due to the FFT, we required 13% bytes more. We previously use char array to store sample, but the accuracy is not good then cause the unstable FFT Output.
2. EKG measurements are not stable when input Frequency is larger than 3.3KHz
   Cause: We changed real and imaginary from integer to character so that EKG measurement can be put on UNO, which caused frequent overflow when calculating FFT.
   Fixation: Put EKG measurement on ATMega and use integer type.

6. Test

Test Plan

* Tests for data structures and some features are the same as those in lab 3 and we tested them using the same plan and cases as we did for lab 3. Please refer “EE474 Lab3 report” for more information.

UNO side

We need to verify that functionalities about peripheral on UNO side works properly.

1. We need to verify that cuff value are changed by buttons and uno can capture systolic and diastolic pressure from analog signal from function generator
2. We need to verify that Uno can be measure frequencies of incoming square wave and be able to respond to interrupt we have attached.
3. We need to verify that analogRead value are reasonable

ATMega side

1. We need to verify that remote display window can display data sent from Mega and Mega is able to receive command characters and responds properly to those commands.
2. We need to verify that new added UI works properly (EKG selection button & Display mode button).
3. New measurement EGK can return stable value

Test Specification

Uno side:

1. Cuff value should have the same behaviors as described below in table:

Cuff value	Switch value (indicating direction of changes)	Button Pressed
Stay the same	1	Not pressed

+10%	1	Pressed
Stay the same	0	Not pressed
-10%	0	Pressed

Uno should only return a new blood pressure measurement only if one of conditions stated below is met.

Which pressure to update	Cuff value	When
Systolic pressure	Between 110 to 150 mmHg	First time cuff value is in this range
Diastolic pressure	Between 50 to 80 mmHg	First time cuff value is in this range
None	Any other value	NONE

2. We need to verify that square wave given to UNO is VPP 3.3 V and 50 % duty cycle and value obtained by UNO read is alternating between 0 and 1. We make sure interrupt function is called every time interrupt conditions are meet. We put a counter in interrupt function to monitor how many times interrupts are enabled.

Respiration rate and pulse rate are all handled by interrupt, value returned by function should be approximately the same as frequencies of input square wave.

3. We need to verify that analog read in uno can return reasonable value. Analog read value should be in the range of 0 - 300 (we tune the function generator so that analogread value should be in that range).

Measurements use analogread should return value in that range.

ATMega side

1. We should verify that remote control system can change state of our cardio-monitor and should follow specifications described in table below.

Command sent	Behavior	Can monitor respond to touches and continue to make measurements
I	Enable remote command system	As long as user start system
S	Start monitoring system	Yes
P	Pause monitoring system	No
M	Send newest data to remote display	Yes
W	Send newest warning data to remote display	Yes
D	Turn off display	No

2. We added more UI buttons to ATMega

EKG selection button can be tapped and change color.

We can enter display mode by tapping Measurement in Top menu.

3. For this section we divide this into three part

1. Make sure that analogread can return value ranging from 0 - 1024
2. Make sure that repressed value sent to TFF ranges from -31 to 31.
3. Make sure that index return by TFF is stable, if not increase sampling rate

Test Cases

UNO side:
Cuff testing:
Case 1 : switch at 1 press button for five time
Case 2 : switch at 0 press button for five time
Case 3 : when buff is in range observe whether blood pressure will only change for once

Interrupt test:
Case 1 : Frequency 2k

Case 2 : Frequency 34
Casd 3 : Frequency 1k

Analog read:
Case : provide analog pin a tuned sinusoidal wave, record printed read value

ATMega side:

Remote command
We enter a series of commands and record system responses.

Case 1 : S S D D P
Case 2 : I S P P S D D
Case 3 : I S wait for some seconds w, m

UI testing:
Case 1: click EKG option for multiply times and observe effect
Case 2 : click measurement mode to see whether measurements are up to date

EKG testing
Case 1 : print analog read value
Case 2 : print adjusted analog value
Case 3 : print final index and make ture it is stable and linear

Test Result
Uno side :
Original Cuff value is 80

Case Number	Result
1	80 -> 88 -> 96 -> 105 -> 115 -> 126
2	80 -> 72 -> 65 -> 59 -> 54 -> 49
3	If in range corresponding value will only change once and remain stable until next time cuff value is in range

Interrupt test:

Case Number	Result
1	1990 Hz
2	40 Hz

3	1010 Hz

Analog read value:

Analog read value ranges from 0 to 300 which is what we expected.

ATMega side:

Remote tests:

Case 1 :

Command sent	Response
s	Error message
s	Error message
d	Error message
d	Error message
p	Error message

Case 2 :

Command sent	Reponse
i	Command enabled
s	System started
p	System pausd
p	nothing happened
s	System started
d	Display turned off
d	Display turned on

Case 3 :

Command	Reponse

i	Command enabled
s	System started
m	Measurement Data updated
w	Warning data updated

UI testing:

Case 1: Ekg selection button will change color as long as we tap it and it will remain the same color as we left it when we re-enter selection menu.

Case 2 : Display mode display temperature, blood pressure, respiration rate, pulse rate and EKG properly

EKG testing

Case 1 : Analog raw data ranges from 0 to 1023 periodically and they are evenly spaced.

Case 2 : Adjusted analog raw data ranges from -31 to 31 and they are evenly spaced.

Case 3 :

Frequency	Index returned
36	1
1000	83
2000	171
3000	240

Index returned are stable at given frequency and approximately linear with frequencies.

7. Summary and Conclusion

Summary

Lab4 and Lab5 are based on Lab1 Lab2 and Lab3. According to the infrastructure, we additionally use Interrupt Service Routine to implement Measure Task including Pulse Rate, Respiration Rate and Blood Pressure on Uno, measure EKG by implementing a hard real time task and implement remote control. In Lab4, we model our patient use resistors, and switch and model Cuff using a button. In Lab5, our main task is to build remote control communication using Putty through serial communication port, and experience hard real time task to measure EKG. In the end, we integrate our whole system and implement a medical embedded system which can be controlled by either remote control from other terminal or by local display. Now the system support to measure Temperature, Blood Pressure, Pulse Rate, Respiration Rate and EKG, and allow remote measurement.

1. Timing Issue

    Due to the refresh rate of Screen on Uno, the minimum user friendly output's rate should be larger than 10ms.

    When choose measurement items, user should click the button and release the button as soon as possible, otherwise it will not change the previous status of measurement.

    When choose annunciation, the whole system will stall for 2 seconds due to the process time which is inevitable.

2. Potential Bug.

    Continuing press on button without release will make system stall until users release.

3.Test Issue

    It is hard to read time in second with small error range. Because our delay can be accurate in 10x ms, so we need to test our output delay is correct or acceptable.

4. Tradeoff

    Due to the dynamic Memory Size, we measure and process EKG in ATMega 2560 rather than in Uno. In our original design, we support put our RTOS kernel in ATMega2560 and do all the measurement in Uno which could be more convenient for the future upgrade. However, the dynamic memory size in Uno has only 2048 bytes which is 13% bytes smaller than our original expected value, so we put EKG measure in ATMega 2560 as well.

Conclusion

This is the end of our 474 project but is also the start of our new journey. We start from Code Block to simulate some functions to integrate whole measurement and can even use remote control to control the whole system. We experience how to design a RTOS kernel to control the system, how to use ISR to measure data, how to implement hard real time task, how to communicate with two MCU and how to use remote control to control the whole systems. We experience bugs and challenges, complaint and beer. Now, our system, the medical embedded system can fully operate. It can measure Temperature, Blood Pressure, Pulse Rate, Respiration Rate, and EKG and users can remotely use it!

We appreciate Dr. James Peckol and our TA: Daniel Predmore, Christy Truong, Jonathan Hamberg for their help and advice. Congrats to your graduation! We really enjoy this quarter and this course!

8. Individual Contribution

Name	Contribution
Gaohong Liu, Shawn Xu, Yongqin Wang	Software design & testing
Gaohong Liu, Shawn Xu, Yongqin Wang	System integration
Gaohong Liu, Shawn Xu, Yongqin Wang	Lab Report

table 14

9. Appendices

User Manual

There are two parts of our system. Users can either operate our systems using remote control, or using local display.

Remote Control has following command:

I The I command initializes the network communications between your system and the browser page.

E The E error response is given for incorrect commands or non-existent commands.

S The S command indicates START mode. The command shall start the embedded tasks by directing the hardware to initiate the measurement tasks. In doing so, the command shall also enable all the interrupts.

P The P command indicates STOP mode. This command shall stop the embedded tasks by terminating any running measurement tasks. Such an action shall disable any data collecting interrupts.

D The D command enables or disables the local display.

M The M command. The M command requests the return of the most recent value(s) of all measurement data.

W The W command. The W command requests the return of the most recent value(s) of all warning and alarm data.

IMPORTANT

User has to type I to initialize the remote control and type S to use whole system, otherwise the system will stall until activate.

After system is ready to work, users now have two thing to choose

1) Measurement Menu

   Inside Measurement Menu, Users have three things to choose, left button is body temperature, middle button is pulse rate, and right button is blood pressure. If they are in green, it means users are currently measuring and they are in red, it means they are disabled. After finish measurement selection, users could press return button to back to mean menu.

2) Annunciation

   If users press Annunciation button, the system will begin to measure and calculate and display panel will display measured data on panel. User still could use return button to return back to main menu.

3) Measurement

   If users press Measure button, the system will begin to measure and calculate everything(include EKG) without warning. User still could use return button to return back to main menu.

Source Code

Source code locates at ee474_proj5.zip

Extra Question

1. **Q:Hard copy of your pseudo code**

   A: Included in Discussion of Lab Section.

2. **Q: Hard copy of your source code.**

   A: Included in Source Code Section.

3. **Q: Empirically measured individual task execution time.**

   A: We have following data:

   For the first run, due to the cold start:

   Measure: 16µs
   Compute: 236µs
   WarningAlarm: 88µs
   Display: 718676µs
   Status: 12µs
   EKGMeasure: 2.5ms
   EKGProcess: 17µs

   For the rest run (almost all of rest run):

   Measure: 8µs
   Compute: 8µs
   WarningAlarm: 80µs
   Display: 8µs
   Status: 8µs
   EKGMeasure: 2.5ms
   EKGProcess: 8µs

4. **Q: Include a high-level block diagram with your report.**

   A: Included in Discussion of Lab Section.

5. **Q: If you were not able to get your design to work, include a contingency section describing the problem you are having, an explanation of possible causes, a discussion of what you did to try to solve the problem, and why such attempts failed.**

   A: We have finished our Lab with all specification required.

6. **Q: The final report must be signed by team members attesting to the fact that the work contained therein is their own and each must identify which portion(s) of the project she or he worked on.**

   A: Included in Individual Contribution Section.

7. **Q: If a stealth submersible sinks, how do they find it?**

   A:First, try to find the signal sent from the submersible.

   Second, if technicians can not track signals, try to locate the track of submersible which should be in some limited range.

   Third, use sonar to locate it.

8. **Q: Does a helium filled balloon fall or rise south of the equator?**

   A: Rise. Because the density of helium is smaller than Air.

10. Reference

[1]Class.ee.washington.edu. (2018). ***EE474 Workload and Grading*. [online] Available at: https://class.ee.washington.edu/474/peckol/admin/expectations.html [Accessed 6 Apr. 2018].**

[2]Hannaford, B., Ecker, A., Emerson, B., Nguyen, A., Lee, G., Aasheim, J. and Cornelius, T., Peckol, J. (2018). ***EE 474 Project 2 Learning the Development Environment – The Next Step* [ebook] James Peckol, p.18. Available at: https://class.ee.washington.edu/474/peckol/assignments/lab2/project5-Spring18.pdf [Accessed Apr 19, 2018].**

[3]Wang, B.(2009). ***fftHelper.* [online] Available at:https://class.ee.washington.edu/474/peckol/assignments/lab5/ [Accessed June 6, 2018]**

[4]Joung, J., Plump, B. (1998). **Fixed Point Version of Fast Fourier Transform source code [Source code]. https://class.ee.washington.edu/474/peckol/code/FFT/brent_fft/**