

Wearable Sensors

Bi-Weekly Report
13 April 2004

Table of Contents

| | |
|--|----------|
| <u>Sensors Team</u> | 3 |
| <u>Temperature Sensor</u> | 3 |
| <u>GSR Sensor</u> | 3 |
| <u>BVP Sensor</u> | 3 |
| <u>Sensors Team Remaining Tasks</u> | 4 |
| <u>Board Team</u> | 5 |
| <u>Board Team Remaining Tasks</u> | 5 |
| <u>Software Team</u> | 6 |
| <u>Chipcon-Cygnal Communication (Leader- Clay)</u> | 6 |
| <u>USB – Board Connection (Leader- Adam)</u> | 7 |
| <u>GUI (Leader- Daniel)</u> | 7 |
| <u>Sensor Data Acquisition (Leader- Josh)</u> | 8 |
| <u>Testing and debugging the Chipcon communication software (Leader- Josh)</u> | 9 |

Sensors Team

Temperature Sensor

Since our last bi-weekly report, we have recalibrated the temperature circuit with the 3.5V battery supply. The linear relationship still has remained constant, but the constant of multiplication changed according to the smaller input voltage. It should be noted that the temperature on the surface of the skin is approximately one degree Fahrenheit lower than the average oral temperature. Our calibration has taken this into account.

We have also made a design decision to attach the temperature sensor to the tip of the index finger with a small Velcro band to securely hold it in place.

GSR Sensor

We have successfully implemented and calibrated the GSR. The skin resistance is relative to each individual and an initial zero or baseline value must be established for each person. Our circuit easily handles this design constraint. We have also decided that this sensor will be placed on the ring finger with the Velcro band.

BVP Sensor

We have successfully modified the circuit to operate on the 3.5V battery supply. To solve the motion artifact issue, we have decided to implement two identical circuits, one that will measure BVP values, the other will only detect the baseline values. We will then use software capabilities to subtract the motion artifact from the output resulting in a clean signal.

Sensors Team Remaining Tasks

- Design the three sensor circuits on Eagle software (Leader- Gouri)
- Order the parts for sensor board population (Leader- Rosy)
- Submit order for PCB manufacturing
- Populate the board
- Test and debug the board
- Implement it with the transmission board
- Write the software for the subtraction of the BVP signals

Board Team

The board should be arriving on Monday, April 12, 2004. Christina and Clay will be working on populating the board with the parts that have arrived from Digikey. This task should be completed by Wednesday, April 14, 2004. After the board has been populated, the board team (which consists of Clay, Christina, and Adam) will be working together to combine the sensors board with the transceiver board as an immediate task. The remaining tasks for the board team are as follows:

Board Team Remaining Tasks

- Complete population of the transmission board with the ordered parts. (Leader-Christina)
- Combine the transmission board with the sensors board.
- Test and Debug system.
- Combine the boards with the transmission / receiving software.
- Test and Debug system.
- Combine system with GUI (graphical user interface).
- Test and Debug entire system.
- Integrate system into wearable device (wristwatch)
- Project completion.

Software Team

The software team, which currently consists of Josh, Daniel, Adam, and Clay, has been working toward the following goals.

Chipcon-Cygnal Communication (Leader- Clay)

Successful communication between the Chipcon and Cygnal processor working. Below is a list of our debugging steps:

1. We first hooked up the boards to the EB63, and determined we could read and write registers from the Chipcon daughter boards. This verified that the old code was working and that at least one of the daughter boards could send and receive using serial communication.
2. We then swapped out the first daughter board (board 1), for the second (board 2) to see if it was in working condition. When we plugged it in, it did not work. We then plugged the other board (board 1) back into the computer to see if we were missing something. It did not work. Unfortunately that was when we noticed we had a pin that was set to Vcc, that was not supposed to be set. Our next step would then be to see if this error had fried the serial communication capabilities of both the daughter boards.
3. To determine if our theory was correct, we went ahead and placed both of the daughter boards back into the Chipcon motherboards, and proceeded to check to see if we get them to communicate using the software we got from the manufacturer. The boards worked perfectly. We were able to get both boards to send and receive without any CRC errors.
4. We went back to check our connections on the ChipCons to see what was causing the issues found in step 2. We first decided to simplify the number of pins being

plugged into the board, by soldering all of the ground pin wires together so that we just had to plug in one cord. We did the same for both power supplies. Next we looked at the code to see what was occurring when the program ran. We simplified the read procedure, and verified the write procedure. Next we hooked up a logic analyzer and found quite a few errors with the code. The errors found through this debugging process have now been fixed.

After all of the commands, timing requirements, and voltages were checked, we still came up short, and the ChipCon is still not responding to commands.

Our Theory: We believe we fried one of the buffers on the ChipCon, thereby preventing serial communication from occurring. The motherboard is using the UART interface, so the next step is to develop the code to write to the ChipCon using UART. This will be completed and tested by Wednesday of next week. Hopefully this will lead to successful communication.

USB – Board Connection (Leader- Adam)

The USB Adapter has arrived from Mouser. We will use this to create a connection between the USB adapter and the board. Since we have the code for this connection from a previous semester, we should be able to get this connection stable relatively easily.

GUI (Leader- Daniel)

For the software interface, Daniel is working on creating version 2.0 of the GUI. By Wednesday, April 14, 2004, Daniel will have completed the display functionality of the GUI: The remaining portions of the task include:

- write a better frequency detector for use in class FrequencyDisplay (for pulse rate)
- some minor layout changes to the main form

For now, the "sensor readings" used to test the GUI are emulated in software. The next major task is to interface the software with the sensor signals being sent through the USB port.

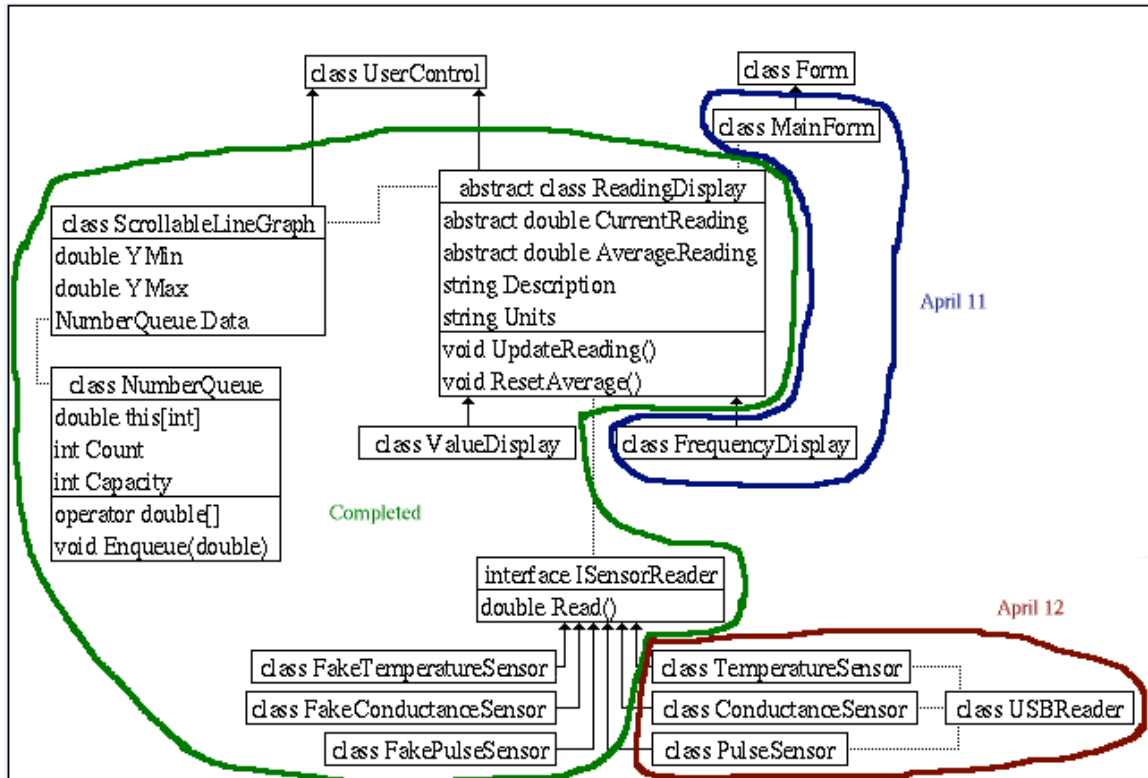


Figure 1- An outline of the classes needed for completion of the GUI. The green section has already been completed, whereas the dates for completion of the blue and red sections can be found alongside the section.

Sensor Data Acquisition (Leader- Josh)

The data acquisition team is currently working on reading in data collected by the sensors (temperature, galvanic skin conductor, and pulse oximeter). So far, data from 3 analog voltages can be input to the MCU and converted to digital values. To say that the data acquisition portion of the project has been completed, we must be able to read in 6 analog signals (3 pairs of 2 differential signals) and perform an operation to create three single coherent signals from each pair. The specifics of each signal and the method of creating

a single signal from each pair are necessary knowledge to reach this goal. Knowing these things, modifying the existing code should be fairly straightforward.

Testing and debugging the Chipcon communication software (Leader- Josh)

Since the last biweekly report, the following has been worked on by the initial software team of Josh and Daniel:

The first of the debugging tests involved writing a “snooping” function in an attempt to use the Cygnal MCU to read the values of packets sent between two Chipcon evaluation boards (which were using the SmartRF software to send/receive groups of specified packets). Josh reasoned that this did not work correctly because the Chipcon has a faster clock speed than the Cygnal MCU. Since this is the case, it is possible that the values sent over the SI/SO/SCLk/CSn pins that we were “snooping” were changing too rapidly for the Cygnal MCU to read correctly.

Daniel and Josh then ran several tests to determine how exactly the MCU was writing values to the pins used to communicate with the Chipcon. We noticed that when the MCU's output pins (SI/SO/SCLk/CSn) were connected to the corresponding pins on the Chipcon evaluation board, the values on these pins could not be changed from within the MCU code. From this, we guessed that we could not use the evaluation boards to test our code, and would have to set up the Chipcon daughterboard on a breadboard by its own, so that we would not get interfering signals from the Chipcon EB.

Since that discovery, Josh has been busy setting up a breadboard to supply all of the correct values to the pins on the Chipcon daughterboard. This has involved making several smaller circuits to regulate the voltage, etc.