MEDSence: Accessible Pill Dispensing Device- Week 4 Report

Work Completed

The work completed this week centered on the Bluetooth technology. Last week I had problems programming the Bluetooth module using a serial connection. Since the user’s manual was vague regarding the pin-out diagram I did not try to establish a serial connection due to fear that an incorrect attachment could burn out the module and cause even more delays. On Friday of last week I received information about the correct way to attach all of the wires and was able to establish a connection with the Bluetooth.

Once we knew we had the module wired correctly we began to program the module to the settings we established we wanted. The programming was done by using the HyperTerminal program on the computer. This program allowed us to send commands to the microcontroller. This was especially difficult in the beginning since we thought we had not set up the connection correctly since we could not see the text we typed. We tried sending text files to the module which did not work. Upon further research about HyperTerminal we realized that we had not set the properties of the connection correctly. We checked the box in the settings tab that would echo back the commands we were sending the eb505. Once we did this we could see the text that we typed and then were able to successfully program the device through a serial connection. Figure 1.1 shows a screenshot of the HyperTerminal window. The commands seen in the window were some of the ones that were sent to the eb505.

![Figure 1.1- Sending Commands via serial connection through HyperTerminal](image-url)
The module we chose for this project does not have the ability to accept commands wirelessly. Before establishing a wireless connection I had to send all of the commands necessary to program the module fully. The following is a list of commands, their function and how they add to the project.

> ver all
This command returns the version of the firmware installed on the device. Knowing the version of firmware is important to ensure that we know the exact specifications of the module.

> get name and > set name
The get name command returns the name of the module. The name of the module is how it will be recognized by other devices. The second command is how the name of the module is changed. When other Bluetooth devices discover this module the name MEDSense will come up.

>get address
This returns the hexadecimal address of the module. Finding the address is important in trying to connect to other Bluetooth devices.

> con
This command is issued when connecting to another Bluetooth device. This tells the module to connect to the desired Bluetooth device.

There are only a few commands used in programming the hardware. Other commands set the baud rate and set security features to protect unwanted data from being sent to the eb505 module.

Figure 1.2- Wireless Connection between eb505 and Bluetooth Dongle
Once the programming was completed we turned our attention to establishing a wireless connection through a Bluetooth dongle through the USB drive in the computer. We found the address of the Bluetooth dongle by searching the properties within the dongle software. I then connected the eb505 to the dongle using the connect command through HyperTerminal. The command was a success and I disconnected the serial cable. There was two ways in which I knew that there was a wireless connection. Figure 1.2 shows the eb505 module without a serial connection. The LED is still on which means there was still a connection. Figure 1.2 shows that there was a wireless connection because of the line connecting the MEDSense device with the center sphere representing the dongle. Since there was evidence of a connection in both the hardware and the software the connection was established successfully.

![Figure 1.3- Interaction between computer and Bluetooth module](image)

Following my work on the Bluetooth module I concentrated on how to program the microcontroller. The Bluetooth technology has no value unless it can send data to the computer. To accomplish this task I must program the microcontroller to send data to the Bluetooth module. I have researched how to do this online and am making progress with the code. I also have studied C++ programming. I have not programmed in C++ for many years and it takes some effort to relearn some of the protocol. This information will be useful when I write the code to drive the motors and incorporate the real time clock. The information I have accumulated this week will allow me to make great
progress when I spend more time on the programming aspects of the project in the coming weeks.

**Future Work**

With the Bluetooth connection established I now turn to the software portion of the project. I ordered a book online that should aid in programming the microprocessor. The only experience I have with programming microcontrollers is when I programmed a microcontroller for the EKG project last semester. This experience is not particularly useful since I did not write the code for the microcontroller which will be by far the largest task I will have to complete in the upcoming weeks.

There are many aspects of the microcontroller programming that I will continue to work on next week. Most of my work this week outside of the Bluetooth module was learning as much as possible about how to program the microcontroller. This information will allow me to progress in programming the microcontroller to drive the motors and send data through the Bluetooth module. I hope to make great strides in driving the motors and am optimistic I will finish the code within the next two weeks so that the testing the motors can begin. This is a lot of work but I feel I am up to the task.

**Project Review**

I believe we are progressing with the project. The Bluetooth technology is working correctly which is a major part of the project. If the programming of the microcontroller goes smoothly the group should see great advances in the project. This will allow us to begin sending data to a computer, testing the motors and setting alarms. The circuits to drive the motors are built and ready for the microcontroller. All of our parts are in except for the vibrating motor and the text-to-speech module so work can begin or continue on all parts of the project excluding these two parts. I am confident that marked improvement in the development of the device will occur over the next two weeks.

**Hours Worked**

13