Senior Project ~ Medical Astronaut Monitoring System (MAMS)

Phillip Fultz and Ryan Hollinger

Dr. Frank Merat

April 20, 2001

Executive Summary

With rapid advancements in the space program, space applications are becoming more affordable, leading to an increase in space related projects. From construction of the space station to repairing space satellites, more and more astronauts are working outside the space shuttle. To ensure the safety of the astronauts, it is important to continuously monitor the astronaut's vital signs as well as their equipment's performance. This monitoring technology is currently limited to electrodes on the skin directly wired to a computer that monitors vital signs. This technology limits the astronauts mobility, their efficiency, and performance while performing their duties in space.

A low power, wireless solution is the answer. With the recent emerging technology of Bluetooth, we have been able to build an application to communicate an astronaut's vital signs and equipment performance.

____

Phillip Fultz

____

Ryan Hollinger

____

Dr. Frank Merat – Project Advisor

Introduction

Current astronaut monitoring systems involve a direct cable connection from the astronaut to a recorder attached at their hip. This leads to a cumbersome array of cables protruding from the body limiting comfortable movement. The large number of wires also adds unnecessary weight to the space suite. The MAMS (Medical Astronaut Monitoring System) provides a low cost, low power wireless solution using commercially available Bluetooth Technology. With a range of 100 meters, along with a customized software package, astronauts and their equipment can be monitored continuously using Bluetooth technology.

Methodology

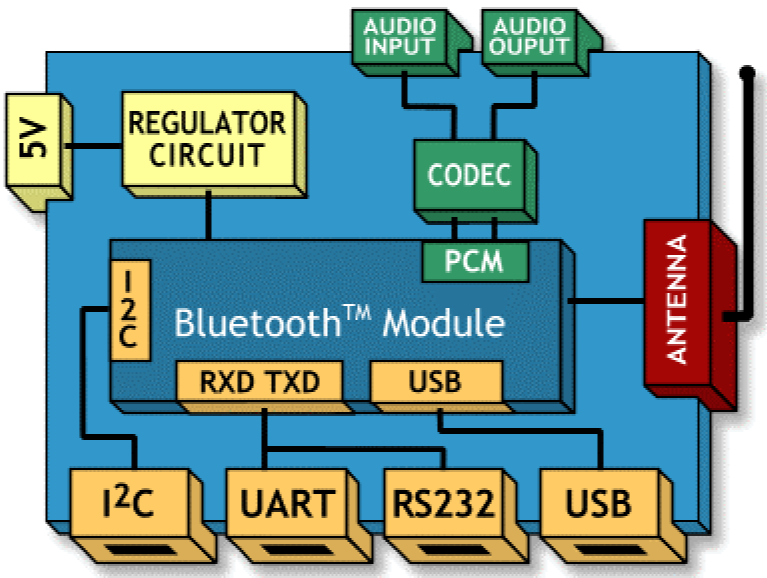
The success of the MAMS project hinged on procuring Blue Tooth development boards and writing appropriate software. First was the procurement of two Bluetooth development boards. This was not an engineering feat but an information-gathering portion of the project. The next major component to MAMS was the driver software. The largest portion of our project is called the Host Controller Interface (HCI). The HCI takes care of setting up serial communications, establishing a connection between the boards, sending, and receiving data. On top of the Host Controller Interface is the graphical application we have designed to mimic an astronaut monitoring system. This gives the user easy access to the MAMS functionality he desires.

Bluetooth Development Board

After extensive research we found our best option to allow development using Bluetooth was a local company Audiopack Technologies. Our research revealed Bluetooth is still a very expensive area in which to work; development boards cost around $2,500. The board we obtained for temporary use from Audiopack technologies was made by Stone Street One and used Ericsson chipsets.

A simplified diagram of the development board layout is shown below. As can be seen 5 volts is provided from a power supply. This is regulated to 3.3 volts by the regulator circuit to power all of the devices on the board. The board has the Bluetooth Module, which is a chip designed by Ericsson. Ericsson's chip is compatible with the version 1.0 of the Bluetooth specification. The current version of the specification is 1.1, but for our purposes 1.0 was adequate.

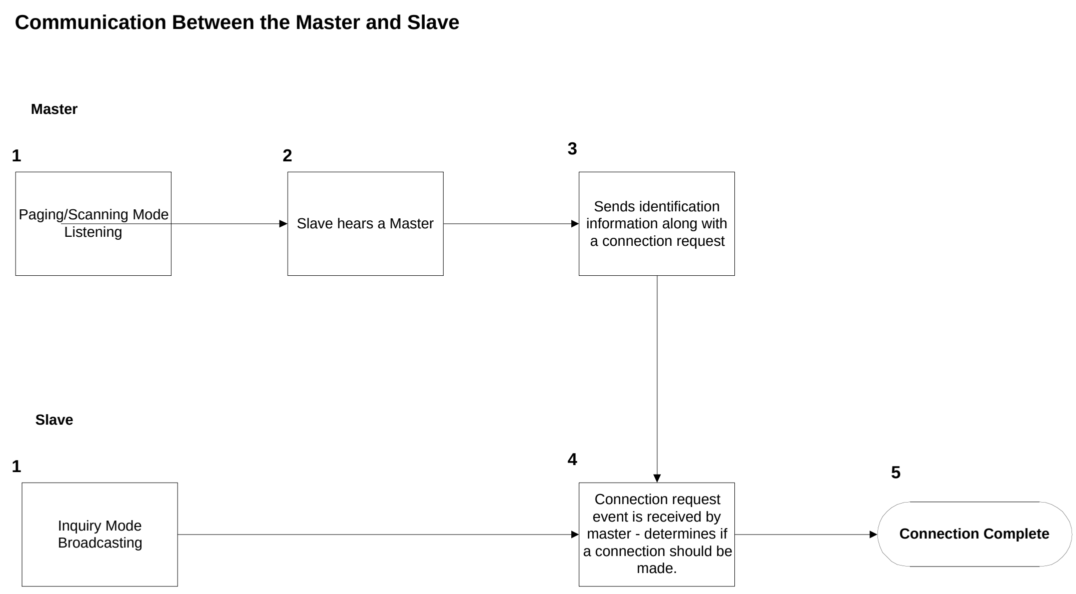
As can be seen, there are a wide variety of methods to communicate with the board. We elected to use RS232, as sending and receiving data this way is straightforward and well documented. A second forms of communication, I^2C, is designed to integrate the Bluetooth module with chipsets on the board. Both I^2C and USB, the third option offer faster data throughput. This would be beneficial for high bandwidth data like video. USB also offers a slightly cheaper solution from a hardware point of view, with only two wires. Audio circuitry is include on the board. This is very useful, as it allows audio to be transmitted without having to go through the host controller. This is what allowed our project to have audio, support as we could not have attained sufficient data rates across the 57600 bps serial link. The CODEC, the main audio component uses Log PCM encoding. This is one of the audio coding techniques mandated by the specification. Log coding using a logarithmic transfer function compressing the data significantly. The specification is the International Telecommunication Union, recommendation G.711.



Driver Software

The driver software is broken up into a few sections. The first is called the comm section and deals only with the very basic input output with the serial port. It is also responsible for the initial set up of the serial port. This involves setting the various settings such as baud rate and parity. It also gets a handle for writing and reading from the comm port and ensures that the port is initialized properly. Another major part is the separate thread created to read incoming data from the port. The thread blocks while checking for data; upon receipt of data the OnRecieveData function is called. This function deciphers the incoming information.

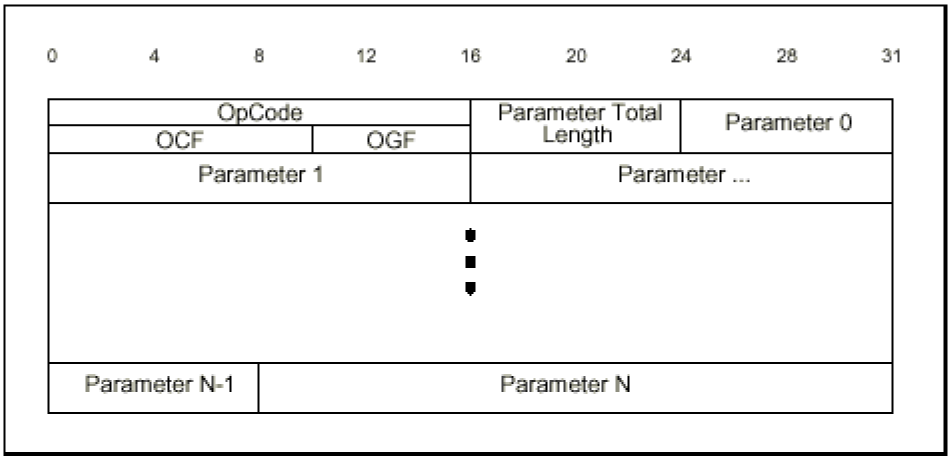
The HCI section is the parent to all other classes in the MAMS project. One of the major components is the HCI_init function, which allows a connection to be established. In order to establish a connection there are many commands to be sent and events to be processed. The initialization procedure differs based upon the role of the unit, which can be either slave or master. The diagram below shows how a connection is made



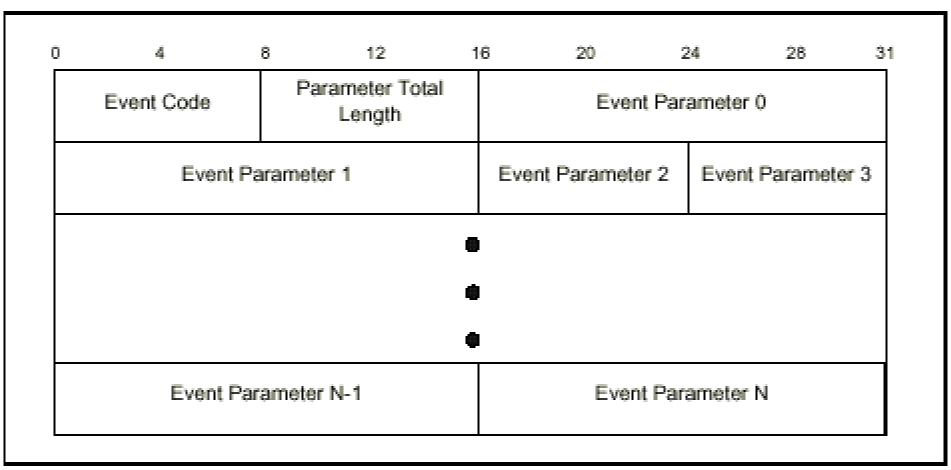
HCI also has a function which establishes a Synchronous Connection Oriented (SCO) link. This enables audio transmission. An Asynchronous Connection Less (ACL) link is automatically established when a connection is established. This enables data transmission. The HCI_establish_SCO function sets up the audio connection and immediately begins transmitting. This activates the microphone and line out jack on the boards.

The last major part of HCI is the receive data function. This simply takes in the data the comm thread has received and begins processing it. Data and events must be differentiated, and acted upon accordingly. The major complexity in this function is that it is recursive to deal with multiple events or data packets in the same read from the buffer.

The next major section is HCI_CMD which deals with the sending of commands to operate the board. The format for a command can be seen below. This class is rather large because of the numerous commands. It has all of the commands we have needed to implement our project with the proper parameters for each. There was a great deal of time spent here as many parameters needed values, were unknown. For the header of the commands OpCode Group Field (OGF) and OpCode Command Field (OCF) can be seen. The OGF specifies which group the command belongs to while the OCF declares the actual command.

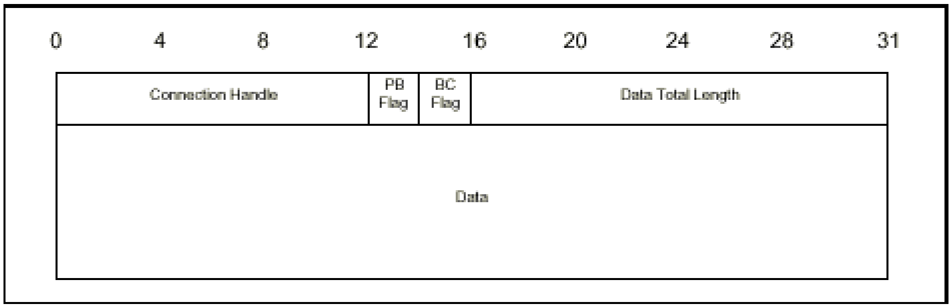


The event class governs the flow control, such as announcing when a certain event has been received. This has to be watched for so execution of the code can continue once the event has been received. It also removes and saves necessary data that is returned in the events. There are far fewer events than commands so only one byte is needed for the event code.



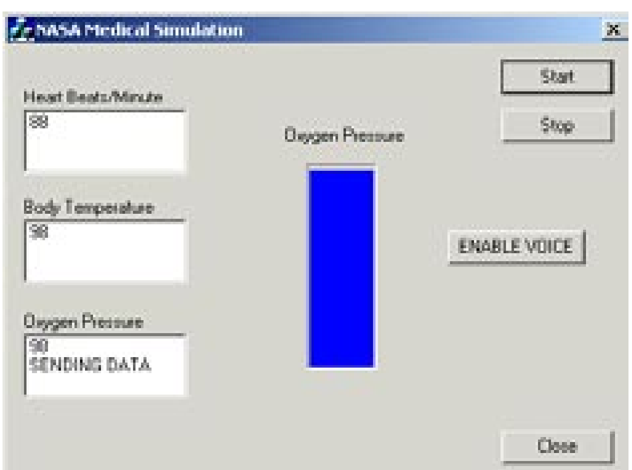
The last major section is HCI_send_data, which formats the data packets. This connection handle is known from the connection complete event. In addition there is an Logical Link

Controller Adaptation Protocol (L2CAP) header that needs to be placed first in the data section. This is necessary and if not present the receiving board will throw the packet out. The L2CAP header only includes another length parameter as well as a channel identifier string. This ID string allows multiple streams of data to be identified from one another.

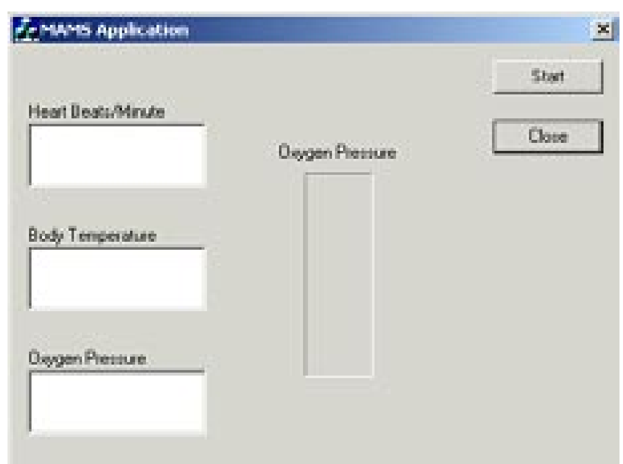


Application Software

There are two separate applications, a controlling application for the master Bluetooth chip and a controlling application for the slave Bluetooth chip. These applications provide the means to send and receive data. Below are screen shots of the master and slave applications.



Medical Simulator located on Slave Application



MAMS Application located on Master Application

Master Application:

The master application as shown above is the main driver for sending data. This Graphical User Interface (GUI) is a simulation for what would be embedded in electrodes placed on the astronaut's skin or equipment. Once an astronaut is within range of the master Bluetooth chip[1], the master initiates a connection. The "Start" button places the Master Bluetooth device in "listening" mode. Until the device "hears" another device, the application will continue to listen idly.

Once a connection has been successfully made, the application polls for data and immediately updates the display once data has be read and sorted. Upon receiving data, a flashing "RECEIVING DATA" is displayed.

Slave Application

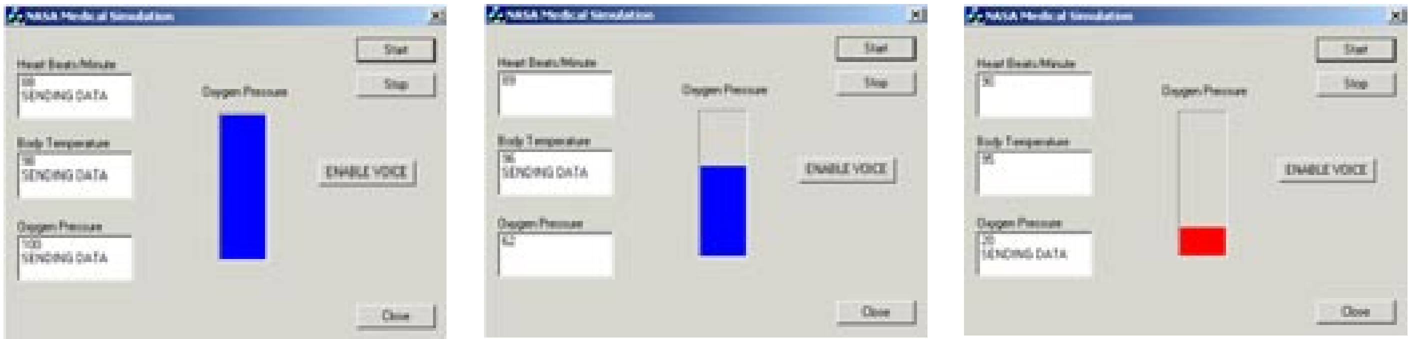
The slave application is responsible for simulating a slave Bluetooth chip located on an electrode either on the astronaut's skin or equipment. The slave application is responsible for sending information to the master. Once a connection has been initiated by the master application, the slave will start sending data.

The "Start" button will place the slave application into page scan mode. Once in page scan mode, the master will be able to hear the device and attempt to connect to it. As soon as a connection has been established, data will be sent.

The "heart beats per minute" data will be sent every minute, the "body temperature" will be sent every thirty seconds, and the oxygen pressure data will be broadcast every six seconds. Each process contains it's own thread to ensure optimum processor efficiency. A flashing "SENDING DATA" will be displayed each time data is sent. The progress bar will indicate graphically the status of the oxygen pressure. Once the pressure drops below twenty-five percent, the bar turns red as shown in the screen shots below.

The "Stop" button will destroy the three threads and reset the file pointer. The connection between the master and slave will still be enabled. Once the "Start" button is pressed again, the three threads will be recreated and data will again send from the beginning of the file. By pressing the "Close" button, the connection will be immediately dropped. The "Enable Voice" button will enable real-time voice transmission.

[1] Approximately 10 meters for our application but more expensive chips are capable of transmitting up to 100 meters.



Screen shots of the slave "medical simulation" application as data is sent.

Results

Our project has been a complete success. We have established a connection between Bluetooth boards, created an ACL (asynchronous connection less) link, allowing us to send data between the two boards and established a SCO (synchronous connection oriented) link, which has enabled us to send real-time voice communication. We have also created a Windows application geared towards a NASA inspired proposal to create a wireless medical monitoring system. Our application, the MAMS (Medical Astronaut Monitoring System) simulates data collected from electrodes and sensors located on the astronaut and his equipment. This data is then transmitted to the MAMS application residing on a PC controlling the master Bluetooth chip.

Another key element to our success was to measure how portable and expandable our code is. We have found our class design to be very robust. We had little difficulty adding to our project as it expanded, and do not anticipate any problems in future development of our code. To ensure complete encapsulation, we have compiled our Bluetooth code into a dll (dynamic link library), which is directly linked to our MFC application. Our MFC application then provides the user interface.

Review of Other Implications

- ✓ *Environment* – Our devices have been approved by the FCC. They do not emit interfering signals that conflict with other devices.
- ✓ *Sustainability* - Our software has been designed very well so that it can be easily ported to other chipsets and expanded upon in the future.
- ✓ *Health and Safety* – The Bluetooth board has been purchased from a reliable retailer. We do not anticipate any electrical damage to be caused due to the development board.

- ✓ *Economics* – The majority of our project is software, which is of little cost to port to other PC's. Audiopack Technologies located in Cleveland, OH allowed us to use their Bluetooth development boards.

Conclusions

We have achieved all of the goals we had set forth at the beginning of the project. Bluetooth technology was very new when we decided to pursue this project. Our only tool was the Bluetooth specification. We have written software to send data and voice communication using Bluetooth. We have designed an application geared towards monitoring astronaut's vital signs and equipment status using a Bluetooth link. We have designed our code for portability and further expansion. We have compiled our Bluetooth communication code into a dll (dynamic link library) and our MFC application statically, enabling the code to be easily ported from PC to PC. Our software has paved the way for future applications wishing to use Bluetooth technology.

Recommendations

There are numerous applications that can take advantage of Bluetooth technology. Wireless keyboards, mice and printers are all concepts being developed now. With the state of the Bluetooth technology, very few tools are available for development purposes or reference. In fact, much is unknown about the technology's capabilities. Can multiple Bluetooth enabled devices talk to each other in network. The specification talks about a PAN (Personal Area Network), however Bluetooth development kits as of four months ago were not equipped with the hardware to support this feature. However, Bluetooth has excellent potential. Development tools are becoming less expensive and more advanced. More support is becoming available as more people become acquainted with the technology. We had the privilege of getting to learn about a very hot technology in its infancy stages. Although at times we wished we could have worked with a technology with more support features, we realize that the potential for Bluetooth is endless and we were able to take part in advancing it.

After talking with Dr. Matthiesen and Dr. Prahl, both former astronauts, we got a very good idea of space related applications that Bluetooth could impact in the future and how our project could be expanded further in the future. Currently, electrodes are mounted onto the skin, usually seven in the chest area, two on each ankle, and two on each arm. The wires from the electrodes are run through the space suite to a large box that hangs from the hip. Physiological data is recorded all day as they perform their duties. Many times even more electrodes are mounted on the body during experiments. For example, during re-entry into the Earth's atmosphere, water that initially gets shifted from the legs to the head while in space instantaneously reverses course and flows from the head to the legs, causing many different physiological consequences in different people. One major complaint expressed by both Dr. Matthiesen and Dr. Prahl was the amount of uncomfort felt while wearing the electrodes. Another issue was that the wires commonly broke while being stretched in the suit. Both were enthusiastic about the implementing Bluetooth with the electrodes.

Another application discussed was related to the oxygen/nitrogen content in the shuttle and space station. Oxygen and nitrogen are supplied though vents and circulated by fans. Many times the air does not mix well and pockets or nitrogen are left as a result. Also, while working in a corner, ones airspace may become filled with CO_2 causing the astronaut to become dizzy or faint. If each astronaut could wear a sensor with Bluetooth technology, the sensor could report the levels of CO_2 and report these to the master computer, which could alert the crewman who is working in that area.

A third application that was suggested was geared toward the scientists on board the shuttle. Most scientists want to have their own laptops on board with them because the mainframe computer on board is very limited and old. However, due to solar flares and other solar radiation, these laptops become unstable and unreliable. If one main computer could be used to load all of the experiments and relay the information via Bluetooth to the main terminal, reliability increases and fewer experiments are interrupted. This also makes for a more organized system of experimentation.

There are many types of applications that can be applied to Bluetooth. While we have only discussed applications related to astronauts, there are many other areas that could benefit from this technology. From space systems to cell phones, we believe Bluetooth will be the next wave of technology.