

# WIRELESS TELE CARE SYSTEM FOR INTENSIVE CARE UNIT OF HOSPITALS USING BLUETOOTH AND EMBEDDED TECHNOLOGY

C.Rajeshkumar, [Beeds93@gmail.com](mailto:Beeds93@gmail.com), K.Praveenkumr, [kpraveenece2011@gmail.com](mailto:kpraveenece2011@gmail.com)  
KONGU ENGINEERING COLLEGE

## DESIGN AND IMPLEMENTATION

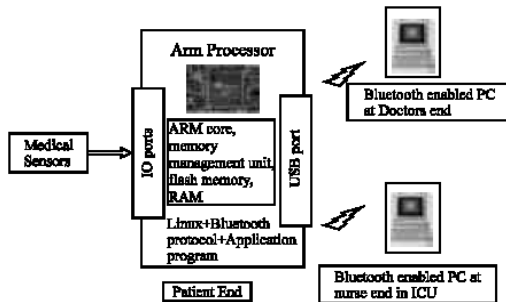


Fig. 3: Basic block diagram of the proposed tele care system

Fig.3 shows the basic block diagram of the proposed tele care system. This is constructed using ARM processor. To transmit the data to the remote PC, blue tooth USB adopter (Dongle) is connected to the USB port of both the handheld device and the remote PC. Multi processing is required for the handheld device as it has to transmit the data and acquire it simultaneously.

The ARM processor is the heart of the handheld device the ARM is a 32-bit machine with a register-to-register, three-operand instruction set. ARM920T core is most advanced of all the cores available amongst the ARM series. An Atmel product of AT920 architecture, AT91RM920 is used for this purpose. The ARM processor is a 32-bit RISC processor (Dandamudi, 2005). It is designed and developed by ARM limited (Advanced RISC machine). These ARM processors have high performance together with low power consumption and system cost. The ARM920T features instruction and data caches, Memory Management Unit (MMU) enabling support for all major Operating System (OS). It is a single development toolkit for reduced development costs and shorter development cycle time. The ARM processor provides solutions for open platforms running complex operating systems for wireless, consumer and imaging applications. It has wide applications in embedded real-time systems for mass storage, automotive, secure applications including smart cards, SIMs, industrial and networking applications.

Embedded Linux is the operating system which runs the AT91RM920. The embedded linux has three parts starting

from the boot loader to the kernel and the RFS (Root File System). U-boot is the boot loader used in AT91 core and it is burnt in starting location of the memory. As soon as the handheld device boots, U-boot fetches the kernel from the flash and puts it into the RAM. The kernel then takes the RFS after it loads itself completely. The U-boot is readily available in the net which can be downloaded and fused to the flash memory. Similar is the case for the kernel and the RFS but little difficulty arises as it has to comply with and get patched for the specific target (AT91RM920 core in this case). Once the compilation and patched are over, it is ready to be uploaded in the flash itself.

For the purpose of converting the analog data acquired to digital, an ADC 0808 is used. This is further given as input to the parallel port of the AT91RM920 processor. The sensor which would measure the digital data such as drip level could be connected to the processor directly. Signal conditioning circuit is constructed for the SpO2 sensor which would measure heart rate.

One of the important functions of the kernel is to recognize the hardware connected it by Means of device driver program. The device driver program for the ADC and the rest of the sensors is written in C and compiled along with the kernel. The RFS is the user environment in which the program the acquiring data and transmitting the written. The program is compiled along with RFS and burnt in the flash after the kernel location.

BlueZ is the official Linux Bluetooth Protocol Stack which is also readily available as a open source over the internet. The protocol stack is also compiled along with the kernel and burnt in the flash. A Bluetooth USB adapter is used as a physical device to transmit the data to the desired location. Each USB adapter has it own unique hardware ID called the MAC ID. Program is written so as to transmit the information only to the desired MAC ID and rest of the system will ignore it.

An LCD display is also fixed in the device to receive for the suggestions from the doctors apart from the bio medical sensors. The appropriate Kernel and FFS level programs are also written and fused in the flash.

The Base station which would acquire the data is also enabled with a USB Bluetooth adapter. The handheld device with the patient would transmit the data to the MAC ID of this device alone and ignore the rest. The front end is constructed using visual basic and displays the acquire information and updates it continuously. The handheld device is unique for each patient.

## SENSORS

Sensors play an important role in the monitoring of the parameters in the handheld device. The precision at which the handheld device is to be operated is decided by sensors in the system. Vital parameters of the patient such as temperature, glucose level, ECG, Drip level, pulse rate can be measured. The measured values are then transmitted to

the base station via blue tooth. The sensors have to be designed properly in order to make the handheld device not much prone to erroneous data. The sensors which we have chosen for our system are temperature sensor, drip level, ECG and SpO2 sensor.

#### Temperature Sensor

It is used to measure the body temperature and it is needed to be monitored at constant time intervals. The temperature measured at the handheld device is then sent to the base station. The temperature being measured is analog in nature and hence an ADC0808 is used for acquisition of the data and then for conversion in to digital values. The digital value obtained from the ADC is then fed to the ARM processors parallel port for further processing.

The device driver for ADC is written at the kernel level of the Linux OS and then a user program actually measures and controls the operation of this sensor. The device driver program is written and compiled in C and then attached along with the kernel through the zImage which has already been mentioned as the compressed Linux kernel. The user program for fetching the data and for further maneuvering is also written in C and then compiled along with the RFS. Both these are fused along with the flash memory of the handheld device.

#### Float Sensor:



Fig.4 – Float Sensor

Fig.4 shows the float sensor used to measure the level of drip in the saline bottle. This is again of prime importance in medication process. The drip once emptied, will indeed feed air to the body which could lead to the death of human being. In order to avoid this from happening, we are using a

float sensor which would actually measure the level of drip in the bottle and then transmit it to the base station. When empty, a trigger can be activated which would initialize a relay to cut down the drip.

The operation of this type of sensor depends on which level of contact it is. When it touches the bottom level it will indicate LOW and the trigger action has to be performed accordingly. The float sensor is indeed a contact based sensor which actually gives the digital value and hence the need for an ADC is eliminated. A user program can efficiently be used for acquisition of the data from the float sensor. The user space program is compiled along with the RFS and then fused along with others in order to transmit the acquire data in the base station.

#### ECG Signal

The ECG signal shown in fig.6 is being acquire simultaneously along with the other parameters and the received signal is transmitted to the base station. As usual the data is acquired by means of a user program at the root file system which controls the operation. The data is transmitted to the base station in the format of an image. The transmitted image is then obtained, assembled and then fed to the front end in Visual basic and the doctors use it for further analysis.

#### Pulse

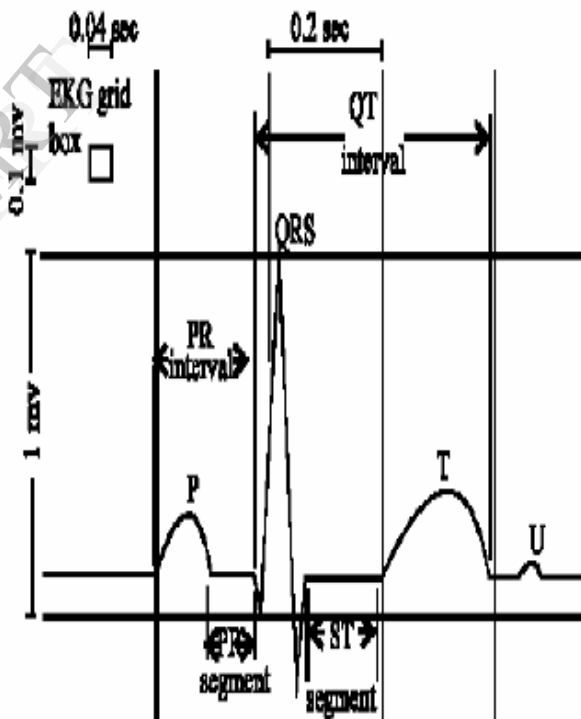


Fig 6:ECG signal

#### Pulse Oximeter Sensors:

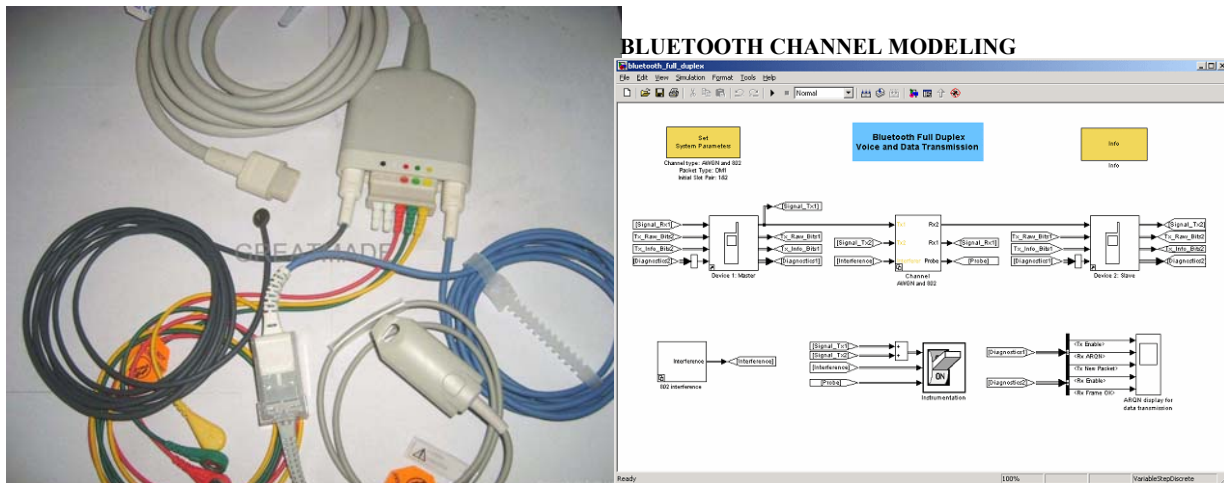


Fig.8 – Simulink Model of Bluetooth Transmission



Fig.7 – Pulse Oximeter Sensor

The heart rate of the patient is measured using the pulse oximeter sensor as shown in the fig.7. The operation and type are as follows; basically a two-sided probe transmits infrared light through body tissue, usually a finger tip. Most light will be absorbed by the tissue between the probe. The small amount of light that is not absorbed is detected by sensors on the other side of the probe and this is used to measure hemoglobin saturation. Absorption variation between oxygen-rich and oxygen-poor produces a difference that enables the microchip to calculate hemoglobin saturation. SpO2 can be measured at any place where a pulse is accessible. In practice, oximeter is usually measured on fingers, the earlobe or with infants at the bridge of the nose.

The sensor types depend on the different applications and usage. Broadly there are two types namely disposable and reusable sensors. We have used reusable sensor since it can be used more than once. The program for fetching the data from the reusable finger probe is written and compiled which is then fused in the flash memory of the ARM processor along with the kernel and RFS.

After encountering problems in some base band parameters of our physical hardware implementation, Bluetooth transmission was successfully implemented by building, simulink model. Although the existing model fully supports transmission of voice packets (HV1, HV2, HV3), it has been extended to include support for the transmission of basic data packets (DM1). The simulink model relating each of the block components is shown in fig.8. The simulink model is composed of several core blocks to create the simulation of the entire Bluetooth system. System parameters block configure the whole model's packet type, slot pair and channel type. There are two possible channel models: 1) Additive White Gaussian Noise (AWGN) and 2) AWGN with an 802.11 interferer. The simulink model shows the interplay between the Bluetooth master and slave device. The transmission and reception of the signals are event driven in nature and the algorithms are modeled using state-floor, a finite state machine tool, which defines the device's state as being either "idle", "transmitting", "training", or "synchronizing".

#### Data Transmission

Data transmission uses Asynchronous Connection-less (ACL) packets. Currently, the implementation of data transmission requires that no errors are allowed in transmission and packets are retransmitted otherwise. This characteristic is inherent feature of Bluetooth where packets are sent repeatedly by the transmitter until it receives an acknowledgement back from the receiver indicating that the packet was received correctly. In future directions of our project involving the support of more data packet types such as DH1,DM3,DH3,DM5 and DH5, this feature will need to be disabled such that an estimate can be made about bit error rates in transmitting JPEG images via Bluetooth. Essentially, without any type of Forward Error Correction (FEC) or cyclic redundancy checking, the packet would be an AUX packet type, transmitted as best effort UDP. This would increase throughput given that we no longer have to use bytes for the CRC code. Also, this would decrease idle time in the master since it would no longer have to wait for an acknowledgement packet sent by the slave device before sending another data packet.

### Data reception

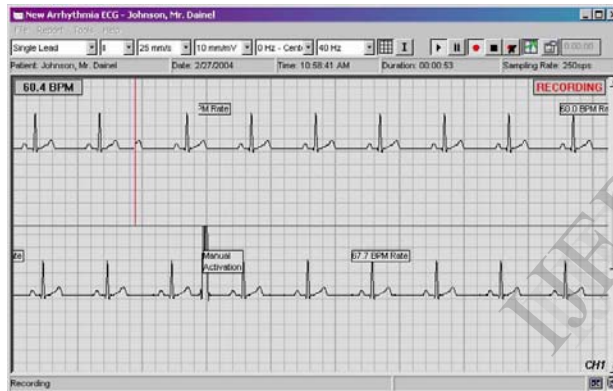
The receiver is always waiting for a new packet. When a new packet is detected, it must register it and enable the decoder. Currently, the majority of the receiver code involves checks to ensure the correct detection of a packet. Future changes in this design to incorporate JPEG image transmission will no longer need this feature as BER will be the parameter of interest.

### Frequency hop generation

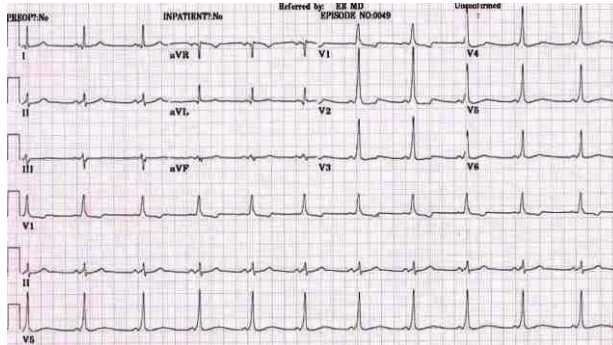
The Bluetooth wireless specification defines a technology that operates in the unlicensed 2.4 GHz radio spectrum. It uses a spread spectrum, frequency hopping, full-duplex signal up to 1600 hops/sec. The signal hops among

79 frequencies at 1 MHz intervals to give a high degree of interference immunity.

In the simulink model, there exists a frequency hop generation block that emits a new frequency every cycle for the data to use for transmission. Some study has been done already to support different data packet types that need to be transmitted over the same frequency over multiple clock cycles (e.g., DM3, DH3, DM5, DH5), but further refinements are needed.



Transmitted ECG



Received ECG

### RESULTS AND DISCUSSION

One of the most difficult problems we encountered in our implementation involved modifying the Simulink model to support file I/O. As given, the model modeled the bits in an image and just a random series of integers generated from a random number generator. We needed to have the code actually read in binary information from a image file that had been compressed using JPEG. Similarly, we needed a way to output the received image to a file so that we could

subsequently examine the effect of JPEG image transmitted using Bluetooth.

ECG data which is to be sent to the base station is in the form of image. The ECG image is taken and the image is decomposed in to matrix by using `imread()` function. The image is 256\*256 and hence after the decomposition of the matrix into pixels, the dimensions of the matrix will be the same as its size. The data fed to the 'integer to bit conversion' block should be in vector form. The matrix is converted to vector form by using `reshape()` function. Appropriate payload header is multiplexed along with the payload. CRC block enables error check to be performed on the data being transmitted.

The payload bits then moves to the AWGN channel block where the actual simulation of Bluetooth channel occurs. In this block, the

characteristics of the Bluetooth channel such as frequency of transmission are set. The output from the AWGN channel block is given to the Rx input of the slave block. The bits are then arranged in to data of 8 bit format. Payload header is then separated from the data received. BER should be zero otherwise the ARQ requests to resend the bit being transmitted. Image is reconstructed and the received matrix which is in vector form is converted to 256\*256 matrix by using the `reshape` function. The reshaped matrix is put in to image format using `imwrite()` function available in matlab. All the images are of `uint8` datatype which has to be taken care of before using `imwrite` function otherwise the image would be blurred.

The faithful reproduction of received data is verified by finding variance of the transmitted and received images and then taking the log of the variance of the transmitted image to the variance of the received image.

### CONCLUSION

The ratio of the variance of the transmitted and received images is calculated and found to be unity. This shows that transmitting the medical parameters using Bluetooth will not contribute to the error rate as to that of its counter part. In addition, the BER of the data is zero but increases over the time. This advantage is overcome by means of ARQ hard wired in the Bluetooth functionality. Hence the information transmitted would reach the destination with minimal error which also can be corrected.

In the future this handheld device can be enhanced as homecare system, so that handheld device would log on the patient's vital parameters to the hospital server from the web server through internet.

### REFERENCES

- John.H.W.(Ed.) 1991. Operating Room and Intensive Care Alarms and Information Transfer. Symposium paper held at Zurich, Switzerland, 27 Feb, 1991.
- Paul.N., M.D. Lanken, C.WilliamIII., M.D. Hanson and S.Manaker, 2000. The Intensive Care Unit Manual. W.B.Saunders Company.
- Nathan, S.M., 2001. Bluetooth Demystified. Tata McGraw-Hill, India