1 Table of Contents

List of Figures

List of Tables

1 Track User Tracker Executive Summary

The Track User Tracker system (TUT) is both a health care professional's tool and an exercise assistant. Track refers to an indoor running and walking track found at many health clubs and physical therapy locations. As a tool for health care professionals, TUT allows "prescriptions" of exercise regimens related to track activities to be written for patients. Monitoring of a patient takes place via a TUT website and patients can access their own exercise log in the same manner. Patients, while using the track facilities can monitor their progress during a prescribed session. For example, a patient prescribed an hour walk per day would be able to see time remaining data as well as distance covered and an average speed display. This data is displayed on a wearable device, the Wrist Band Tracker or WBT. A patient cannot override a prescribed regimen and can only add to the regimen if the prescriber allows. Health care professionals with patients using the TUT system can monitor progress and adherence to the prescription.

As a personal exercise assistant, TUT allows general track users to set exercise goals, "self-prescribe", and to monitor performance during a track session. The system will maintain an exercise diary for these users in the same from as that of a patient's. The system keeps track of an individual's exercise patterns over time, I.E. track usage, miles covered, time spent walking or running and several other relevant data items. This data takes the form of the personal exercise diary residing on the TUT website as already described. A user's diary is accessible by both the user and health care prescriber via the internet. Security procedures to ensure privacy have been implemented in the TUT application software and no financial or personal information, other than a user code, is kept by the TUTapp or website.

The hardware for track session monitoring is maximized for ease of installation at a facility housing an indoor track. No wiring is required other than USB cables required to charge and interface a Windows PC to the WBT(s). Each component uses minimal power and is in large part self-sustaining. Maintenance is therefore straightforward and requires no technical expertise. The TUT monitoring hardware uses a low power RF frequency for all functions. A wireless shutdown feature has been designed to initiate a power shutdown such as at the end of the day or when no TUT user is on the track.

The design allows for a wide variety of users occupying the track at the same time, moving side-by-side, singly, walking, jogging or running. As is common at track facilities, user are expected to move in the same direction, however the system accommodates a general change in direction. In other words, all users may be moving clockwise around the track one day and counterclockwise the next.

A typical indoor track at a gymnasium is a rounded rectangle to maximize building space. Distance around the track varies but typically ranges from eight-tenths to one-tenth of a mile. The TUT user interface during a track session takes place through its OLED display packaged in a wearable device. The Wrist Band Tracker, WBT, is designed to be a wearable device and can be similar in size to a large wristwatch. The WBT periodically displays new information to the user. The intention is to provide periodic data and not to have the user monitor the WBT on a continual basis. At a session end, performance and goal achievement data is uploaded to the TUTapp, which updates the user's exercise log.

2 TUT Project Description

2.1 Motivation for building the project

Simple exercises such as walking or more intense exercise such as running or jogging have known health benefits that are truly out of proportion to the amount of effort required. Walking lowers low-density lipoprotein (LDL) cholesterol, raises good cholesterol, lowers blood pressure, helps control type II diabetes, heightens mood, and adds strength to the physical body. Outdoor exercise, such as that described above, is probably the most preferable choice. It is free and simply being outside is, in itself, a break from the stresses of the office or home. There are drawbacks. Weather, local traffic patterns, pedestrian routes all can prevent regular exercise.

Indoor tracks are a good option. They offer a better alternative to treadmills, which can be misleading and are routinely abandoned by a purchaser. The use of an door athletic track is however somewhat monotonous. Conversations with users, some of them hard-core competitors, reveal that most users soon lose track of performance data.
For example, it is difficult to keep track of the laps completed. If 1 lap around a track is 0.08 miles, and an individual wished to go 3 miles, 37.5 laps would have to be counted. There is no easy method of keeping track of speed and no method will give a relatively instantaneous velocity.

In addition, there are many people who are using these tracks in order to lose weight or for general health objectives. Some motivation would help these types of users meet their personal goals. With the costs of health care skyrocketing, preventative medicine holds the promise of some relief to the health care system. It makes sense that simple exercise regimens, prescribed and monitored by a health care professional, could both prevent costly treatments in the future of a patient and control or diminish existing medical maladies.

A system that takes advantage of already existing indoor tracks, allowing users of all types to get the most benefit from their exercise, is a small, added step in improving overall health and fitness of the public. The system is secure, accessible, easy to use, and easy to install on any track. User performance data in the form of an exercise log can offer encouragement and motivation. Health care professionals will be able monitor activities and offer advice or prescribe a suitable course of exercise for a patient. These “exercise prescriptions” could well prevent future prescriptions of insulin, blood pressure medications or antidepressants.

2.2 Technical Objectives

The technical aspects of the TUT system are simple and complex. The TUT host software bundle allows interactions between the healthcare professional and the patient. It allows casual users to keep records of exercise performed on the track, to set goals (self-prescribe) and monitor progress while on the track. The development of a website for maintaining and updating the user profiles a completed objective for the system. The website and database allow secure interactions concerning “prescribed” exercise regimens and a health care professional. The Windows XP based host used on the system interacts with the website at a track facility and uploads to a remote database exercise related data and progress reports. These are available to both users and to relevant professionals. Two classes of users are supported. Users under a prescribed exercise regimen and users who simply use the system features to either monitor themselves while using the track and/or to keep a personal exercise diary.

The system’s hardware components are designed to maximize ease of installation at a facility and for ease of use. Installation is an important criteria in this system as it is intended to be installed is in a pre-existing building. As such, power requirements and savings are important in the design. The design calls for minimal to no wiring of the track. Components are installable with no mechanical mountings other than for example double-sided tape. The hardware is flexible enough to record good measurements on any size track. The hardware is fully programmable and it is not track or location on the track specific. To keep wiring to a minimum the system design uses wireless technology. The system uses a series of transmitter nodes arranged around the track to provide a trigger to a device worn by each track user. The necessity exists for multiple users moving side by side to be determined and therefore a Radio Frequency (RF) beacon of a sort is used. The system hardware is able to determine how many laps are completed of a track circuit. It tallies the total distance covered, determine if a pre-set distance has been covered and determine time-to-go data for the user. The system indicates to the user whether or not current speed is to slow to reach a pre-set goal per lap and alternatively slowing down the user if a lap is completed to fast.

The WBT will be useful to both patients and to casual users of the system. Patients will be limited to certain user modes. For example, depending on the prescribed regimen, the speed up/down indication may not be displayed for some users. In other words data displayed is firstly tailored to an individual's prescription, then to individual allowable preferences. Casual users deemed self-prescribers have complete control over goal data they wish displayed. The WBT device will be comfortable to wear for an extended period (5.5 hours) and clearly display track session data to the user. The data presented can be user selectable through a mode feature. The data displayed will be limited in one mode (data only mode) and the WBT will display only speed, time, distance and laps in this mode. In prescribed mode, where the user is a patient the data is tailored to the prescribed regimen. This is an important issue in the design. Under any circumstance the WBT must not prompt a person who is medically contraindicated from walking, for example, at greater speed. The WBT will be designed to be easily understood and viewed by the by the user. The WBT must be non-distractive to the user. The device will periodically alert the wearer through a selectable alarm system that new data is available for viewing. A design challenge is to make the system both encourage users, via the display, and not overwhelm the user with data updates, warnings and the like. It is undesirable for the user to pay more attention to the WBT than in their direction of travel. The WBT device is intended to encourage the user through the display. Data such as time and distance are presented to the user however, information in the form of encouragements may also be presented. In this sense, it is the design goal that the device appear "intelligent" to the user as well as "encouraging".

The RF components of the system will be very low power and those components which transmit not worn by the user. The WBT is a receive only wireless device and eliminates the need to attach an RF transmitter next to the skin of the user. The transmitter nodes, as originally designed, each have two separate transmitters. The first will be bidirectional to the host computer. The objective is to create, in this section of the transmitter node(s), a small wireless network of limited power. Each node is uniquely identified via software when it is installed. However, in order to achieve a generic design, suitable for installation at any facility, the distances between transmitter nodes are largely unknown and will be transparent to the transmitters. The transmitter nodes achieve some generic ability by the use of a simple repeating transmission process in regards to the host wireless connection. This connection will be used for system maintenance. Both to save power and to limit RF transmission the transmitter nodes will have a shutdown mode controllable from the host in future designs.

The transmitter section dedicated to tracking a user will be a transmit only scheme which required directing the radiation pattern and controlling the radiated power. The transmitter nodes are completely independent of external wiring. These units will to be self-powered and a key technical objective was to make the transmitters self-sustaining as much as possible in an indoor environment.

2.3 Specifications and requirements

2.3.1 Wrist Band Tracker Specifications

2.3.1.1 Power Requirements

The WBT will be a portable electronic device and minimal power consumption is required. Because the anticipated usage will be limited to a single track session the power requirements have been limited to a include a fixed amount of time. This time has been determined from a worst-case scenario based on the Boston Marathon. The qualifying time for an 80-year-old woman is 5 ½ hours (2). Therefore, the WBT device will carry adequate charge for 5 ½ hours. The prototype system uses two, 1.5VDC AAA type batteries.

2.3.1.2 RF Receiver Characteristics

The WBT RF receiver needs only decode, in any field of a transmitter node, one repeating data packet which identifies the transmit node detected as a particular and valid TUT node. It receives at 2.425 MHz, MSK modulation format and a 250k BAUD data rate. The core processor receives an interrupt when a valid carrier is detected or lost. The receiver provides an approximate signal strength measurement which is then processed to determine a distance from the transmitter.

2.3.1.3 Spatial and Speed Accuracy

Accuracy is limited to errors in RF field detection and the accuracy of the timers in the core processor. The accuracy falls within 1.5 feet of a transmitter for the prototype system. Refinements, including better RF control of the transmitter nodes, including antennae, and better algorithm design for RF strength will improve the accuracy as the design matures.

For proof of concept, the 1.5 feet discrepancy, per transmitter node is deemed acceptable. In other words, the system will have an acceptable error of 1.5 feet per transmitter node. This is counter intuitive since it implies that the more transmitter nodes placed on a track the total system error increases. For the prototype system, this was again deemed acceptable. The advantage to more transmitter nodes is not accuracy but more information being provided to a user during a session and subsequent performance data records after. Anticipated use of the system is for total distances above a mile. For each number n of transmitter nodes node the error per mile is calculated as:

$$\% \, error = \frac{n * 1.5 feet}{52.8 * track \ size \ in \ miles}$$

The error will increase as the number of transmitter nodes is increased. The designed maximum number of nodes is 16. For a track size of 0.1 miles the error for a 1-mile session would be:

$$error = \frac{16 * 1.5}{52.8 * 0.1} = 4.54\%$$

The WBT can calculate two speeds. First, an average speed is calculated as a time versus distance between two transmitter nodes. A summation of node to node speeds is calculated for a lap speed function. Second, a pseudo-instantaneous speed can be derived (but has not been implemented) based on the known or calibrated RF field size. The pseudo-instantaneous speed is the time versus distance calculation while the WBT passes in and out of a transmitter field. To the user this speed would appear to be an instantaneous speed. A table of transitions through nodes, in seconds, is passed back to the host upon completion of an exercise session. This data is further processed by the host to recreate the session and generate an exercise log.

2.3.1.4 User Interface and Operating Modes

The system operates in two high-level modes, Patient Mode (PM) and Self-prescriber Mode (SM). These modes are not user selectable. PM limits the user to activities based on a prescribed exercise regimen. PM mode will limit user modifications of the device's input and output. Users receive information via the display based on prescription data. These take the form of a set time goal per lap and a time display of a user's progress. If a lap is completed ahead of time, time is subtracted from the next lap time goal. This indicates to the user that their pace is too fast. The opposite occurs when a lap has been completed too slowly. The objective is to "inspire" the user to complete a prescribed time-distance criteria.

Once the WBT is disconnected from the host USB interface, the user is prompted through the OLED panel and walked through a session. User input to the device is carried out through four switches incorporated into the WBT design.

2.3.1.5 WBT Software.

The WBT software has been developed in C using Code Composer™ Studio Version: 4.2.4.00033 from Texas Instruments (CCS). The GUI for CCS is shown in Figure 1. The embedded code runs as an interrupt driven state machine. All functions developed for the WBT system software by the development team. As an example, the OLED display is a graphics device.

Functions have been developed write a command and or data byte to the unit. A character write needed was written which expands bit mapped font data into a useable pixel format by pixel format. String write functions are developed to generate string writes to the OLED. WBT specific graphical functions have been written to generate bit map images on the OLED.

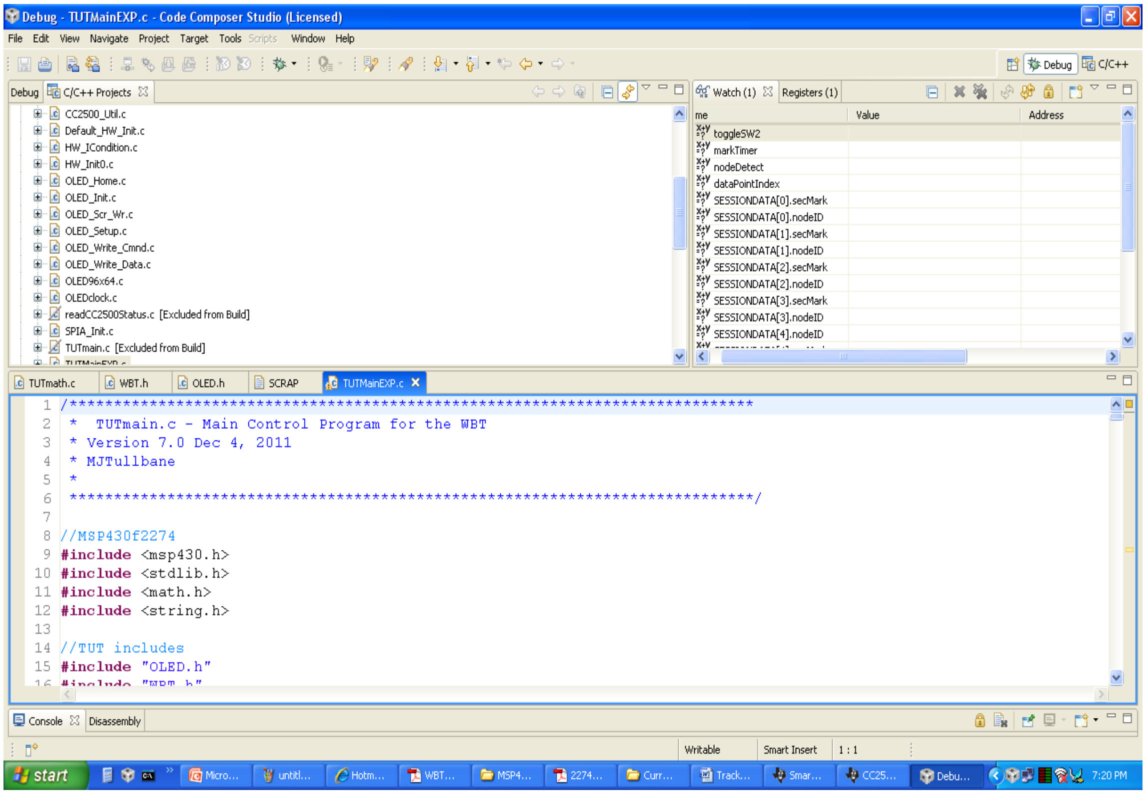

Figure 1-Code Composer™ Studio GUI

The transmitter node detection algorithm consists of monitoring the Relative Signal Strength Indicator (RSSI) register of the transceiver section. The algorithm determines spatial data based on the condition of this register and the validity of the received data packet.

The Tabulated Distance module determines total number of laps completed. It calculates a time goal for each lap based on the distance between transmitter nodes, desired pace and desired time for the entire session. Track configuration data is downloaded from the host along with the time of day and user prescription. This means that a WBT is not configured for any one track but is configured prior to every use. Thus a WBT from track A of distance 0.8 miles with two transmitter nodes is usable on track B of 1.0 miles with one node.

User input functions generate interrupts to the microcontroller and appropriate actions are taken based on the entirety of an exercise session. For example, the user cannot modify prescribed courses of exercise during a session and recording of data does not cease until the final lap is complete. The user can adjust contrast or turn the display off and on at any time but the unit will over-ride these option when a node transition has taken place.

2.3.1.6 Host USB Interface

The WBT interfaces to the TUT host via a USB port. The data is sent and received at 9600 baud, 8 data bits, 1 stop bit, no parity and no flow control. Figure 2 shows the host VCP settings. The USB interface on the host makes use of Future Technology Devices International (FTDI) drivers, which allow the creation of a Virtual Communications Port (VCP) on the host. The WBT RAM data is completely erased after each track session once that data has been uploaded to the host. User goal data and track configuration data must be downloaded prior to each use.

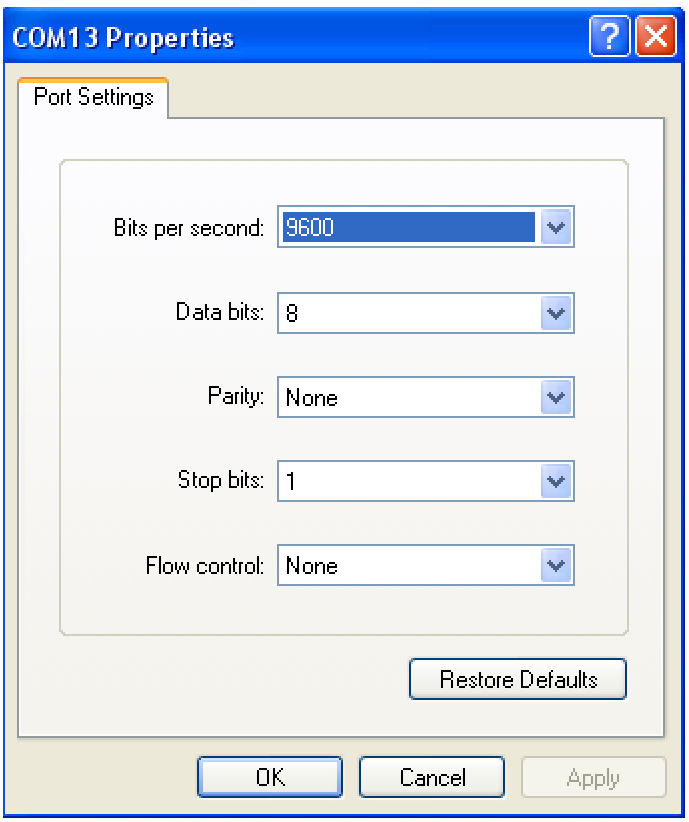

Figure 2-VCP setting for the host

2.3.2 **Transmitter Node Specifications**

The transmitter has been designed for very power low wireless application. Its basic function is to send signal to the WBT within a range of 5 meters. The transceiver interfaces with a host computer using a proprietary wireless connection, which transmits and receives data at the rate of 250 kbps. In active mode, the transmitter will use a current as small as 15 mA. It takes the transmitter 240 micro second to leave the standby mode for the wakeup mode.

The maximum power consumes by the transmitter is 12dBm. The transmitter supply voltage ranges from 1.8 volts to 3.6 volts. It is capable of operating at temperatures between negative 40 degrees Celsius and 85 degrees Celsius. We should keep in mind that the device for which we are designing the transmitter is an indoor device. Therefore, a reasonable temperature is a normal room temperature which is between 25 degrees Celsius and 35 degrees Celsius.

The transceiver is also a low power wireless device. Its voltage supply will be between 1.8 and 3.6 volts. In standby mode, it uses the minimal current of 0.7 micro amps. It will take 240 micro second to pass from the standby mode to the wakeup mode. It receives and transmits data at the rate of 250kbps. The output power is 16mW. It operates at a temperature between 25 degrees Celsius and 35 degrees Celsius. Antenna is vital to any communication systems. The transmitter node is composed of a transceiver (CC2500), a microcontroller (MSP430F2274), a power supply (batteries), and one directional antenna (patch antenna).

A directional antenna also called a patch antenna has been used in the transmitter node design. The basic function of the patch antenna is to transmit signal to the WBT. The patch antenna operates at a frequency of 2.4 GHz, has a matching impedance of 50 Ohms, and is a low cost antenna (less than 30 dollars). In addition, it has a narrow band width, a high gain (10db), and a low cost antenna (under 30 dollars). A narrow bandwidth is obtained when the directional antenna is in the vertical position. Figure 3 describes the radiation pattern of the patch antenna when it is mounted in the vertical and horizontal direction. In the horizontal direction, the forward lobe is larger than the forward lobe in the Vertical direction. The backward lobe is very small in both directions. A Very focused forward lobe is very important for the design of the transmitter node because very narrow is the radiation pattern, and more accurate is the WBT. The directional antenna has been used in the vertical direction during testing, and the demo. The vertical beam with, and horizontal beam with angle are 55 degrees respectively.

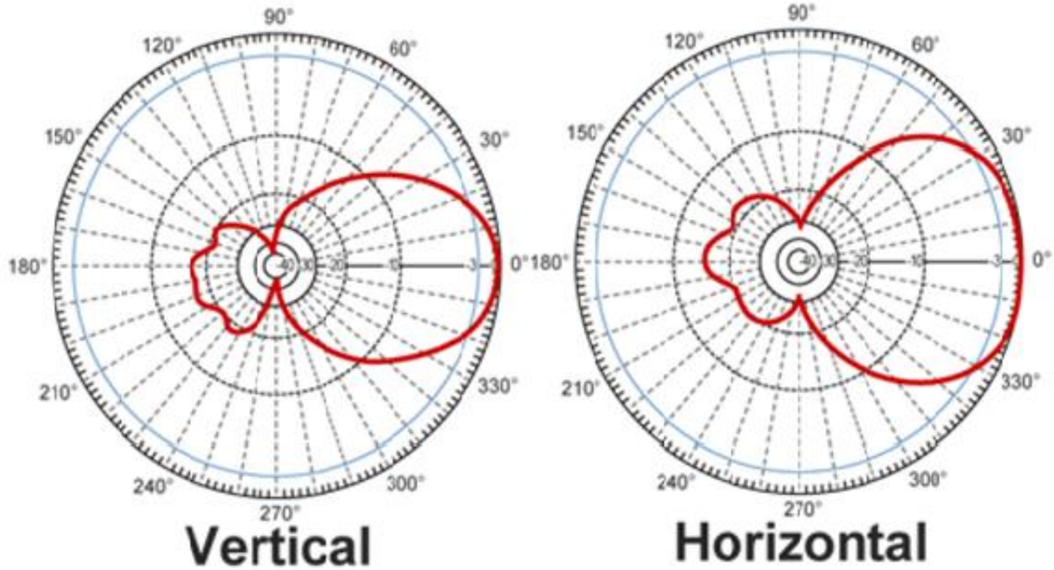

Figure 3: Directional Antenna Radiation Pattern

The directional antenna is designed for all weather operation. It has a compact size approximately 4 cubic inches by 4 cubic inches by 3.5 inches. The patch antenna has dual Integral N female connectors which can be installed for either vertical or horizontal polarization. The patch antenna weights approximately 0.5 Kg. For the purpose of our design, the directional antenna has been positioned in the vertical direction for optimal performance. This antenna operates at temperature ranging from – 40 degrees Celsius to 60 degrees Celsius. The maximum input power of the directional antenna is 200W.

2.3.3 **TUT Software Specifications**

2.3.3.1 Functional Requirements

Functional requirements are derived from actions that the TUT software must perform. They were used during the testing phase to help define the test software stubs that were written. These requirements need to be specific and measurable so that they can be concisely verified and validated during testing. An example of a functional requirement might be “The system shall display the calculated value of 3X + 7 where x is any integer between -512 and 256” This requirement would be very easy to test and it would be easy to write a software stub to automate the test procedure. All of our functional software requirements have been reduced to a similar specific and measurable form. Along with a description of the requirement we also list any dependencies with any other requirements and also how we tested whether the system implements our requirement satisfactorily. We have compiled a list of the text of our requirements in Table 1. Table 1 shows only the pertinent requirements definitions.

Req #	Requirement Text
1	The system shall display the TUT user options screen within 3 seconds of pressing the “Login” button.
2	The system shall present user with the “Accepted” prompt within 1 second after pressing the <Enter> after entering goal data into and goal entry box on the user GUI
3	The system shall present user with assigned WBT number within 5 seconds of pressing the “I’m Ready to Exercise!” button on the user GUI.
4	The system shall present a user prompt in large size text and appropriate color within 1 second after the user removes the wrong WBT from the dock.
5	The system shall logoff a user if there is 2 minutes of inactivity on the system while a user is logged in
6	The system shall prompt user for password within 5 seconds of returning WBT to the dock.
7	The system shall display the results options screen within 1 second of entering password and clicking on the ‘submit’ button. (after returning WBT to dock)
8	The system shall email results, as requested by the user, within 1 minute of entering user email address and clicking on the “Submit” button.
9	The system shall display results, as requested by the user, within 10 seconds of clicking on the “View Exercise Log” button on the GUI.
10	The system shall log user off of TUT system within 1 second of clicking on the “I’m Done for Today” button on the GUI.
11	The system shall turn on Transmitter / Receivers within 5 seconds of user removing the first assigned WBT from the dock..
12	The system shall turn off Transmitter / Receivers within 5 seconds of user returning the last active WBT to the dock.
13	The system shall turn off Transmitter / Receivers within 10 seconds of system time reaching pre-set “System Off” time.
14	If a WBT is still out and the track closing time is reached, the system shall wait at least 15 more minutes or until the last WBT is returned before turning off the Transmitter / Receivers.

Table 1 - TUT Host Software Functional Requirements

2.3.3.2 Interface Requirements

Interface requirements are derived from interactions that must occur between the TUT software and other entities. This includes interaction with the TUT website, interactions with the physicians, and interactions with the TUT user / patient. These requirements were also used in the testing phase and are specific and measurable so that they could be concisely verified and validated. An example of an interface requirement might be “The system must allow the user to enter an integer between -512 and 256 and display an appropriate prompt if the integer is outside these limits” This requirement would be easy to test and easy to write a stub to automate a variety of input. All these requirements were reduced this form. We have compiled a list of the text of our requirements in Table 2, which shows only the pertinent requirements definitions.

Req #	Requirement Text
15	The system shall present the user with a drop-down list box from which to select his / her user ID during log-on
16	The system shall allow the user to enter a default password to log-in the first time and then prompt the user to set a personal password.
17	The system shall allow the user to enter personal goal data.
18	The system shall allow the user to select a Prescribed Session which was previously uploaded from their physician.
19	The system shall allow the user select whether to display end of session data or email the data to the users desired email address. If email is selected, then the system shall allow the user to enter their desired email address.
20	The system shall allow the user to enter their password and log in to the system after returning the WBT to the dock.
21	The system shall allow the user to log out when ready.

Table 2-TUT Host Software Interface Requirements

2.3.3.3 Health Care Professional Interface

The physician does not directly interact with the TUT host. When the physician needs to issue a prescription for physical therapy, they would use their office computer to log-on to the TUT website. The physician would then be presented with options to prescribe exercise requirements for the duration of the prescription, retrieve session data, or perform database queries as desired for statistical analysis. The physician also has the option to add a new patient to the system. The TUT host retrieves the prescription data when the associated user logs in to the TUT at the exercise facility.

2.3.3.4 User Interface

User interaction with the host database is limited to the TUT user GUI. There are three instances when the user would require data from the database. If the user selects the “Retrieve Prescription” option, the system will then log-on to the remote database server and access the database to retrieve the prescription data. If the user chooses to display their session data using the “View Exercise Log” option, the system would log-on to the remote database server and access the database to retrieve the session data. Finally, if the user chooses to have session data emailed to their preferred email address using the “Email Exercise Log” option, the system would then log-on to the remote database server and access the database to retrieve the session data. The user only has access to their session data. All access to the database is controlled with appropriate privileges at the database level.

2.3.3.5 Resource Requirements

The TUT host software runs on any windows computer running Microsoft Windows XP 32 bit with service pack 2 or above, Windows Vista 32 bit, or Windows 7 32 bit operating system. The computer running the TUT host software must have an available universal serial bus adapter, and an Ethernet port. There are no special memory requirements and since all data is stored in a remote database, there are no special secondary storage requirements. The computer system running the TUT host software will be located in a health club or gym type environment. This will automatically insure that the system has sufficient cooling so there will be no chance of extreme high or low ambient temperatures.

The TUT web server is also a windows computer but with added hardware and software requirements. This computer is required to have data security, data redundancy, and sufficient processing power to handle multiple running applications and threads. The server will also require a fast dedicated internet connection as well as sufficient air conditioning to keep it cool. Since this system is initially designed for a patient base of up to 1000 patients, it does not require a full blown server system to operate. With this in mind we require the server to have at least a 2.5 GHZ AMD quad-core processor, 4 gigabytes of random access memory (RAM), three 1 terabyte hard drives configured in a redundant array of independent disk drives (RAID) with at least 1 more 1 terabyte hard drive configured as a ready spare. This hard drive configuration will provide data redundancy, sufficient storage capability, and very good performance at a reasonable price. The server will also need to have a 1000 watt power supply so it will be able to supply power to all hard drives, memory, and processor without overheating. Finally, we would require an uninterruptible power supply to protect the system from power surges and power loss events.

2.3.3.6 Security Requirements

Security requirements for a commercial release of the TUT system are based on the Health Insurance Portability and Accountability Act (HIPPA) which basically requires all data in transit to be encrypted, all stored patient data to be encrypted, all stored patient data to be backed up daily, and all stored patient data to be able to be completely and permanently deleted when no longer in use. Our prototype system does not implement all HIPPA security requirements but we do provide password protection for each user’s data and ensure that no user or the public may access another users’ data.

Access to the TUT remote database is restricted to system level access and is only accessible to patients, casual users, and physicians through our specialized interface. No user is able to exit the graphical user interface and access the operating system. Any attempt to exit the GUI is caught by the system and the user is properly advised. If repeated attempts are made to exit the GUI, the currently active user is logged out of the system and the users' password is randomly reset. That password is accessible by the system administrator so the user can regain system access after contacting the administrator.

There will be at least one trained administrator at each location where a TUT system is installed. Even system administration is quite simple and the on-site administrator still does not have access to patient data. The only person with access to patient data besides the patient / user is the prescribing physician. If the user is a casual user and not a physical therapy patient, then no other person besides the user has access to their data.

2.3.4 Host to Transmitter Node Auxiliary Wireless Specifications

The primary interface to the TUT host computer will be through USB. An auxiliary communication channel to the transmitter node was designed but not implemented. This wireless connection was intended to operate at 865MHz. The wireless network would operate on a repeater type arrangement with each transmitter node rebroadcasting packets received prior to any action. The purpose of the network was for shutting down the transmitter when the track facility closed.

3 Research efforts

3.1 Competing technologies

3.1.1 Visual Recognition or Tracking

3.1.1.1 Hardware Considerations

Visual tracking of users was contemplated for the TUT system. The advantages to such a system are in the limited amount of hardware required for acquiring images. It was felt that a system could be designed using a few relatively low-resolution cameras placed strategically around the track. This design called for wireless feeds from the cameras to a central host computer.

It was determined that a powerful host computer with several GPU's would be required to implement visual identification of individuals. It would require that an image of the user be captured prior to each use. In addition, a casual survey of potential users revealed a requirement of confidentiality. The original assumption was that flat panel displays, updated by the host, would display information to users. It was quickly evident that users would not wish this data displayed to potentially the entire facility. The determination that a public display would not be practical adds the requirement of a wearable device which is personalized to the user.

This design was discarded both for costs and complexity issues. The preliminary costs analysis was exorbitant even using relatively inexpensive commercial off the shelf components. It was further determined that with the exception of the wearable device, almost the entire project would be software oriented. It did add the requirement that a wearable, personalized display be used in the final design.

3.2 Relevant Technologies

3.2.1 MSP430 Microcontroller Family Technology

Texas Instruments' MSP430 family of microcontrollers was selected in the design of both the transmitter nodes and the WBTs. These microcontrollers offer several advantages to the project including good support, sampling and the Code Composer Studio Software Development Environment.

The MSP 430 family is extremely suitable for use in low power designs. Both transmitters and WBT, where long runtimes with minimal power consumption is a critical aspect, require power management. The MSP 430 was chosen for the design over the ATMEL ATmegafamily, used on the popular Arduino microcontroller boards. The selection was primarily based on the perceived technical support that Texas Instruments offers, the wealth of application data available and the free sample policy.

Texas Instruments offers a wide variety of microcontroller development and evaluation kits, such as the eZ430-RF2500-SHE Solar Energy Harvesting. It has a high-efficiency solar panel to operate indoors under low-intensity fluorescent lights. Its batteries are environmentally friendly and can be recharged a dozen times and they have a low self-discharge. Schematic diagrams for this development system as well as a bill of materials is available without costs from Texas Instruments and was intended to lend guidance for the implementation of the transmitter power conservation schemes employed in TUT.

3.2.2 CC2500 Transceiver IC

The MSP430 microcontroller coupled with an RF product the CC2500 radio IC is in use in many designs. This IC has been selected for the WBT and transmitter nodes. The CC2500 with supported RF filter section, 26MHz external crystal and chip antennae is available on the eZ430-RF2500 PCB also from Texas Instruments. A modified eZ430 is used in the design as the RF section for the WBT. Similarly, an eZ430 has been modified for use as a transmitter node.

3.2.3 **LCD versus OLED Display selection**

Display considerations for the WBT display included ease of viewing, power requirements and interface design challenges. An LCD display was at first glance the most reasonable choice for several reasons, among them power consumption and the fact the chosen microcontroller is offered with a built in 96 segment LCD controller.

However, data presented to the user must be easily understood and "encouraging" in some respects. For example, goal data such as time to go should not be simply a four-digit time countdown. The message should include "Time left to reach your Goal" or some other easily understood message. This requirement implied that a simple numeric display would be insufficient for the purposes of motivation. The user in mind is not necessarily an athlete, in fact, the motivation for the TUT system itself is for those needing encouragement in order to improve their overall health.

LCD panels are not emissive devices and it was determined that the use of a back light would be required. The viewing angle of the LCD is limited with the best super-twist nematic devices offering a viewing angle of 120 degrees (3). The backlight for this device draws 25 milliamps and so reduces power saving relative to a passive LCD panel significantly.

3.2.4 **Organic Light Emitting Diode**

As alternative to the LCD an OLED panel is much more attractive. Despite the higher costs, the wider viewing angles (160 degree) and full color graphics capabilities justified the selection of an OLED panel over a similar LCD device. The OLED devices do require a bias voltage (approximately 15 VDC) which can add both components and costs to the design. However, an alternative panel with built in bias generation and as an added benefit, a serial SPI interface has been selected with commiserate costs. The OLED appears to be the new trend in portable device displays. They are relatively simple to manufacture and are considered "greener" in the manufacturing process than LCD displays.

3.2.5 **Directional control of RF signals**

An antenna has to be added to the design to improve the RF performance. When designing an antenna, parameters such as radiation patterns, efficiency of the antenna, its bandwidth, and antenna matching for maximum power transfer have to be taken into consideration. Antenna gain (in dBi or dBd) is the ability to radiate power in a certain direction when connected to a source. G = DD = directivity, μ = efficiency, and m = mismatch loss, dBd = 2.15 dBi, G = antenna's gain

Directivity of an antenna describes a radiation pattern. The antenna can radiate better in some direction than others. Effective Isotropic Radiated Power (EIRP) is a term used to describe the effective radiated power from an antenna taking the gain of the antenna into account. Regulative specifications often refer to EIRP = G P (P = output power from transmitter) for maximum output power measurements. Efficiency (η = R r / (R r + Rd)) is the most important term when talking about small antennas. Rr = radiation resistance (wanted), and Rd is = dissipation resistance (unwanted). In addition, to determine how well the antenna will work, we use the Friis transmission equation. Pr = PtGtGr * (λ * λ / (16*3.14*3.14*R*R); where **λ** = Wavelength in Meters, Pr = Received Power in dBm, Pt = Transmit Power in dBm, Gt = Transmit Antenna Gain in dBi, Gr = Receive Antenna Gain in dBi, R = Distance between Antennas in Meters

Q-factor (Q) (or *Quality* factor) is used to describe the antenna as a resonator. A high Q-factor means a sharp resonance and narrow bandwidth. Q = antenna reactance / antenna resistance. Qmin = (1 / (k*a) ^3)**.** The bandwidth (BW) of a small antenna depends on the Q-factor and the efficiency. BWmax = (16*(πr) ^3 / ηλ^3)**.** Where the antenna is confined within a sphere of radius r, λ is the wavelength and η is the radiation efficiency. There is an inverse proportional relationship between bandwidth and efficiency. For instance, large bandwidth means low efficiency for a given antenna size. The maximum power transfer (VSWR) comes from Moritz von Jacobi's Maximum Power theory. Zo = Z'a (Zo = transmission line impedance, and Z'a = antenna's impedance).The design of an antenna from scratch is time consuming, and a challenge. Too many factors have to be taken into consideration. In addition, an antenna is not expensive ($20) and therefore it is not worth the time and the effort that we want to put in. As a group, we decide to buy one, so that we will not end up design a wrong one.

During our research we find out that a directional or an Omni directional antenna will not be the best for our project even though they are the best antenna for indoor. A directional antenna can divert the RF energy in a particular direction to farther distances. It then covers long ranges, but the beam width decreases. The directional antenna is useful in covering hallways, long corridors, and isle structures for instance. Its disadvantage is the fact that it doesn't cover large areas (small angle coverage).The TUT system needs an antenna with a narrow angle. The WBT has to be able to calculate with the minimum possible error the instantaneous velocity. The only antenna that can give us such angle and work at a low frequency such as 315MHz is the Pulse Helical Wire antenna with a matching impedance of 50 Ohms.

The transceiver will use a wire antenna. The wire antenna will be designed. The wire antenna length will be calculated using the wavelength divided by four. The wire antenna is cheap, and does not require too much engineering knowledge. A comparison of antennae types and their relative advantages and disadvantages is shown in Table 3. We have selected the Wire type antennae for the transmitter to host wireless implementation.

Antenna types	Advantages	Disadvantages
PCB Antenna	No extra cost development, and good to high range.	Require more board area, size impact at low frequencies; require skilled resources, and software.
Chip Antenna	Less expensive (less than $1)	Lower range
Whip antenna	Best for matching theoretical range, size not limiting application	Cost (from $1)
Wire antenna	Application for the 1 sub GHz, very cheap, good performance	Difficult to put on a PCB, variation of the position
IP based Antenna	Designed for directional operations	External cost for the antenna design

Table 3-Antenna Type Comparison

3.2.6 Solar Panel Selection

A solar cell or photovoltaic cell is a device that converts light directly into electricity by the photovoltaic effect. Solar panels are eco-friendly generator. They generate free power using the sun. They convert sunlight to electricity with no emission, and no maintenance. A single solar cell can approximately produce ½ of a volt. If a design requires 12volts, 36 solar cells wired in series will be needed so that the peak output voltage will around 17 volt. A high voltage output is needed because small wire (they are less expensive) can be used to transfer the electric power from the solar panel to the solar regulator and the batteries. Multiple solar panels are wired in parallel to increase the power (current capacity), and in series to increase the voltage capacity. They are mainly four types of solar panels.

Mono-crystalline solar panels are the most efficient and the most expensive solar panels. They are made from a single large crystal (silicon) during a very complicated crystal growth process. Polycrystalline solar panels are also called multi-crystalline cells. They are the most common solar panels. They are less expensive and less efficient than mono-crystalline solar panels because the cells are made from a large block of many crystals.

Amorphous solar panels are made by depositing a thin film of silicon onto a sheet of another material such as a thin layer of silicon onto a sheet of another material such as metal or glass. They are often used in calculators or garden lamps. They are cheaper than Mono-crystalline solar panels and Polycrystalline solar panels, but their energy efficiency is also less than the previous one. If more energy is needed, more panels have to be added, and therefore more space is taken. Dye-sensitized solar cells are a low-cost solar cells belonging to the group of thin film solar cells. It manufacture is less expensive than the other cells design. It is mechanical robust, and do not require any protection from minor events such as hail or tree strikes. Dye-sensitized solar cells work even in low light condition. Some of the Dye-sensitized solar cells disadvantages are the facts that it is made of the liquid electrolyte, which has temperature stability problems and contains volatile organic solvents. A solution to this problem will be replacing the liquid electrolyte with a solid. This is actually an ongoing field of research. Solar panels have to be installed in a sunny (facing the equator) and non-shaded area to get the maximum sun. Shading of the solar panels reduced their output and damages them. It is also important to know that the efficiency of solar panels decrease as temperature increases. A solar panels system should allow for spacing around the individual solar panels for air circulation in order to solve that problem.

Today, the average life time of a solar panel is between 25 and 30 years. This is good news because for 75 or 85 watts output power, a solar panel may cost around $500, and it is a life time investment. There are many ways to connect the solar panel to the battery holder, but for the purpose of this design, two methods will be mainly used. As shown in Figure 3, the solar panel will be connected to a charge controller and then to the batteries. A charge controller regulates the voltage and the current coming from the solar panels going to the battery. If there is no regulation, the batteries will be damaged from overcharging. They also add protection to prevent any leakage of current from the batteries to photovoltaic cell at night.

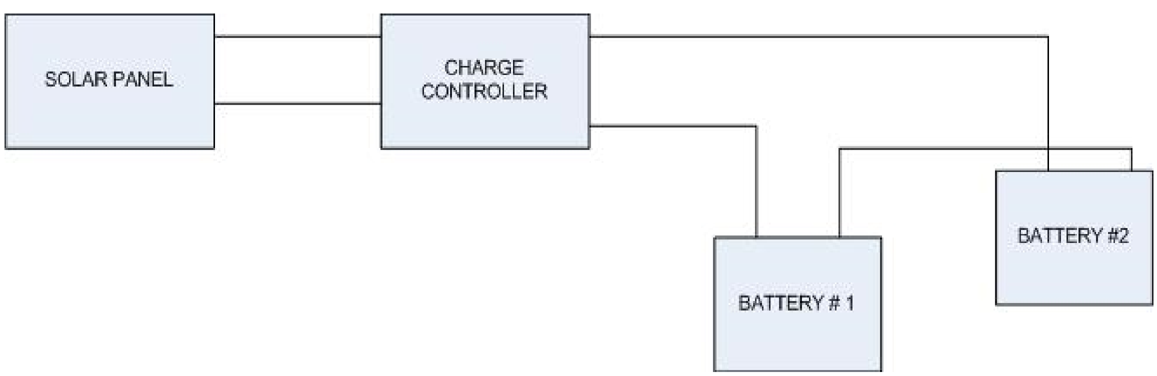

Figure 3: Serial battery connection with charge controller

3.2.7 **Type of charge controls**

There are a variety of Solar panel regulators. The simplest one is the shunt regulator. It has the advantage of simplicity, extremely small power dissipation, low cost, high reliability. It disadvantage is the fact that the voltage on the battery is always going slightly up and down, that the battery is switched between full charging current and no charging current. The second type of charge controller is the PWM (Pulse Width Modulation).It charges the battery with constant voltage or constant current. They are the most commonly used in solar panel system. The third type of charge controllers is the MPPT (Maximum Power Point Tracking) controller. They are the most expensive, and are used for large systems. The choice of a regulator depends on the application. Simplicity is the key to success. The simplest charge controllers will be used in our design. Rechargeable Batteries will be used to store the energy produces by the solar panels (5).

Rechargeable batteries save money, and preserve the environment. There are approximately five different types of rechargeable batteries. The nickel-cadmium batteries (NiCad) batteries are rarely sold due to restrictions on poisonous cadmium that is used in their manufacture. They are mainly used now for low drain application. It disadvantage is the fact it can suddenly run out of power. The second type of rechargeable batteries is the nickel-zinc (NiZn) batteries. They are recommended for high-drain applications such as flashlights and outdoor equipment. The third type is the lithium ion (Li-ion). They are easy to manufacture, and are used in personal electronics such as camera, laptop, and computers. The forth type of rechargeable batteries is the alkaline batteries. They are less common than the other types. They are best used in low-drain applications. They are ideal for backup or emergency use because once charged it last longer than the other rechargeable types. The last type is the Nickel-Metal-Hybrid (NiMH) batteries. They are relatively inexpensive. They power most hybrid and electric vehicles. In addition, they can be relied on for most applications. After reviewing all the types of rechargeable batteries, we decided to use alkaline rechargeable batteries for our design.

3.2.8 **Software Design**

3.2.8.1 TUT Host Software Design Considerations

Having a basic understanding of the TUT system we were prepared to investigate any commercial off the shelf options that may exist for a similar application. Our system is unique in the way that it performs the tracking and in its' intended purpose. There are many off the shelf software systems for the medical industry. There are systems for medical records, systems for medical billing and insurance, and complete systems for medical clinic patient records. All of these systems are large and expensive and still do not provide the functionality required by our TUT system. Some experts assert that the going

price of a web based electronic medical records system is three to four thousand dollars plus monthly fees. This price was prohibitive and the system had too many unnecessary features. Many features that we need in our system are also not available in the commercial package. For these reasons we decided to develop our own software system to support the TUT system (4).
The decision to write our own software for the TUT system raised many more questions that needed to be researched and answered before we could proceed further with the design process. We needed to decide which type of programming style to use which lead to the question of which programming language or languages to use. Since there are three different software subsystems in our TUT software system, we needed to investigate options for each subsystem. We considered that may be able to simplify the process and use the same language and programming techniques for all three subsystems. We also planned for the system requiring us to use a different language for each subsystem or even more than one language for each subsystem. Other considerations included the design of a graphical user interface versus a text based interface. We looked at the possibility of using a commercial database system or maybe an open source database if it has a proven record of security and stability. And, finally, we had to consider the option of either paying for a web hosting service or writing our own web server and host the website ourselves? Our research continued until we had at least three options for each subsystem in our TUT software system.

First subsystem we will discuss is the TUT host. Considering the customer requirements for the TUT host, we knew we needed to select a programming language which easily lends itself to the development of a graphical user interface. Other considerations for choice of languages were the skill the team has in programming in the chosen language and the ability of the language to interface with the other subsystems. Our choices, then, were between "C", "C++", or "Java". We have the most programming experience in "C". "C" is very fast and efficient, but it is not a good language for designing a graphical user interface. "C++" is a versatile language that has object oriented capabilities and a good set of graphical development libraries available, but we do not have the experience in graphical programming with "C++". That leaves the Java programming language. We have quite a bit of experience in Java and it is very well suited for graphical development. There are many tools readily available for designing graphical user interfaces in Java which greatly reduces the development time for producing a functional prototype. Considering time to develop, ease of use, previous experience, and cost effectiveness, it only made sense to use Java to write the TUT host software subsystem.

3.2.8.2 TUT Data Base Design Considerations

There were many options for the TUT database design and implementation. We had many of the same considerations in deciding on a database solution. Cost was still high on our list of determining factors and, fortunately, there were several free options. We also needed to consider how well a database solution works with the other software used in our system. Since we had already chosen Java as our programming language for our TUT host, we needed the database to be easily accessible from a Java subsystem. We also needed to use a database that is easily integrated into our TUT web server. As with our TUT host, we first narrowed the options down to the top three candidates. One of the resources we used to research free database software was the website www.freedatabasesoftware.net. This website actually reviews four free database programs and contains links to many other database resources. Another web resource we checked was the website www.freebyte.com. This website lists about nineteen different free databases with links for their descriptions and downloads. Using this information, we were able to narrow the choices down to three. The free databases that we further examined for suitability in our project were MySQL, Firebird, and PostGreSQL. Before we evaluated the features of each of these choices, we needed to further define our basic feature set. We already said we wanted it to be a free software package. We also needed it to support SQL so that we can perform efficient data retrieval operations. SQL is the language used to ask a relational database to show you the data that you are looking for. The other considerations were similar to our TUT host software considerations. We needed to look at whether or not we were already familiar with this software and how long it would take us to implement our database design with the chosen software. The last consideration was scalability. If we were to produce a commercial system, would this database software be able to support the increased customer base and also would the software cost for commercial use be reasonable? Let us look at our three database choices using our criteria to rate the advantages each choice.

Firebird is free, stable, supports SQL, and it is scalable to meet future expansion requirements. It has support for languages such as C/C++, PHP, and Python, but the current release does not support access from Java. PostGreSQL is free and open-source, it supports SQL and it is fully scalable with many extra features such as support for creating your own custom data types. It has been around longer than Firebird but also does not natively support access from Java. MySQL was our last database program to evaluate. This software has both commercial and free versions and there are also many free development tools available from MySQL. There is a free graphical database management suite called MySQL Workbench that is available from the MySQL website and was written by MySQL AB. MySQL has native support for access from Java along with support for multi-user access. MySQL supports web-server access with PHP and Python and is known as one of the most popular open-source databases. Now when we also considered the factors of development time and previous experience, the clear choice was MySQL for developing the TUT database subsystem. In fact we used MySQL with the MySQL Workbench for modeling, designing, and administration of the TUT database.

3.2.8.3 TUT Website Server Design Considerations

Our initial design of the TUT website would require several different development tools to implement and there are many different flavors of each to choose from. Our research then needed to look at which web server software to use, which server side development software to use, and which client side application software to use. Web server software is our framework for the design of the software that actually communicates with the rest of the internet and manages our addressing and domain. Server side software is the software that displays web content, and calls other applications running on the server. Client side software in our case is limited to standard web browsers such as Internet Explorer, or Mozilla Firefox. Our website is designed to display properly with either of these popular web browsers. The software that powers the majority of all web sites is called Apache. Apache is a very extensive set of software packages. Apache is free, but it is software that we have no programming experience with and it has so many extra features that we did not need for our system. The next most used free web server software is the Information Interchange Service (IIS) included with newer windows server packages as well as XP Pro service pack 2, Vista, and Windows 7. IIS is not installed by default when you install windows, but you can enable the service after installation of the operating system. IIS has many advantages over Apache in that it is easily integrated with other Microsoft products. With IIS, you can set up a web server with all Microsoft packages, such as Expression Web for web page design, visual basic or C# for server side scripting, and Microsoft SQL server for database access. This package is also highly scalable we thought it would be a very good option for a future commercial release of the TUT system. For our prototype purposes there was another package that we thought would be most efficient at implementing our web server. The main component of this package consists of the Python programming language. Python is an interpreted language and it supports object oriented programming. One of the goals of the Python language is rapid development and deployment which makes it very suitable for our prototype development. Another component of this web server package is Django. Django works with Python to quickly design and implement a complex web server using native Python code. Web server development with Django focuses on a database driven design which we thought would be perfect for our prototype since database access is the primary function of our TUT web site. Since Python also communicates seamlessly with MySQL databases, our decision was then easy and we would, therefore, attempt to use Django and Python to develop our TUT web server prototype. It did not take very long to realize that this was the wrong option for our programming skill level and our web site functionality. After re-assessing our options, we chose to host our website using Information Interchange Service from Microsoft.

3.3 FCC Restrictions and the Impact(s) on the Project

3.3.1 General Information

The FCC is the federal agency directed by the congress to regulate broadcasting throughout the United States of America (USA), the District of Columbia, and its possessions (Guam, Puerto Rico, Guam, and the American Virgin Island). The FCC licenses all transmitters whose signal can travel distances, although there are a few exceptions for very low power radio transmitters. The FCC licenses radio transmitters according to geography and certain other common ownership rules that are intended to help prevent radio stations from interfering with the signals of other stations. The spectrum of available radio and television frequencies is limited, so the FCC can issue only a limited number of licenses. In large cities, Broadcast licenses are extremely valuable, and the FCC limits the number of stations in order to control the number of licenses issue. The decision of the FCC to allocate new broadcast stations is based on the need of various communities, and specified engineering standards in order to prevent interference among stations and other communication users. In the United State, 87.7-91.9 bands is allocated to educational broadcasts.

The FCC rules and regulations are codified in Title 47 of the Code of Federal Regulations (CFR). They are initially published in the Federal Register. After October 1 of each year, the GPO compiles all the changes, additions, and deletions to the FCC rules and publishes an updated CFR. Unlicensed operation of small transmitting devices is discussed in "Part 15" of the FCC Rules. These Rules are published in 100 "Parts," covering everything imaginable concerning the FCC regulation for unlicensed transmitter. Our design can be classified under part 15 as Radio frequencies devices and under unlicensed national infrastructures devices. It is the responsibility of builder device which emits RF energy to study, and understand the FCC regulations. It is the sole responsibility of the builder-user of any FM broadcast-band device to research and fully avoid any and all interference to licensed FM broadcast transmission and reception.

3.3.2 Power Limitations on Unlicensed Transmitters (1/2)

The FCC sets 100mW as the maximum permitted power output for unlicensed, home-built transmitting devices, and the combined length of your antenna and feed line must not exceed 10 feet. Under part 15.221, broadcasting on the school ground using AM is permitted, but it has to be between 525 and 1705 KHz. If operating in the 88-108 MHz FM broadcast band, the bandwidth of the transmission is limited to 200 KHz (FCC rule number 15.239). In addition, the RF field strength (it is determined by the antenna and by the RF output of the signal itself) of the signal must not exceed 250 micro volts at a distance of 10 feet from the transmitter (FCC rule number 15.239). FCC rule 15.215 and FCC rule 15.5 are a little bit controversial because rule 15.215 seems to give the builder the freedom to use any types of operations, and rule 15.5 says: "

Operation...is subject to the conditions that no harmful interference is caused and that interference must be accepted that may be caused by the operation of an authorized radio station, by another intentional or unintentional radiator, by industrial, scientific and medical equipment, or by an incidental radiator. The operator of a radio frequency device shall be required to cease operating the device upon notification by a Commission representative that the device is causing harmful interference." The builder of any unlicensed devices has to comply with the FCC regulations, and when there is a doubt, call a technician to check the device to make sure that it is within the regulations.

3.3.3 Protocol Restrictions

As an unlicensed operator, our design has to stay within to the regulations of the FCC so that we will not cause any interference to licensed broadcast services. To do so, we should not modify in any way all the kits that have been purchase because doing so can generate interferences. We also have to read all the notices intended for unlicensed operators. In addition, we also would like to stay in the 88 – 92 bands FM or check our intended operating frequency carefully to see if there is any interference. To avoid any complaints from the FCC, we should identify the locations, and purpose of our transmission from time to time. Our design must conform to the FCC regulations which have been discussed above.

4 System Design Details

4.1 Hardware Components

4.1.1 Wrist Band Tracker Design Detail

4.1.1.1 Major Sections

The block diagram for the WBT, Figure 4, shows the five major sections of the unit, the MPS430F2274 microcontroller, FT232RQ USB section, CC2500 transceiver section, and OLED graphics display. The user input switches, the CC2500 transceiver IC and the FT232RQ IC all interrupt the MSP430 microcontroller at various times.

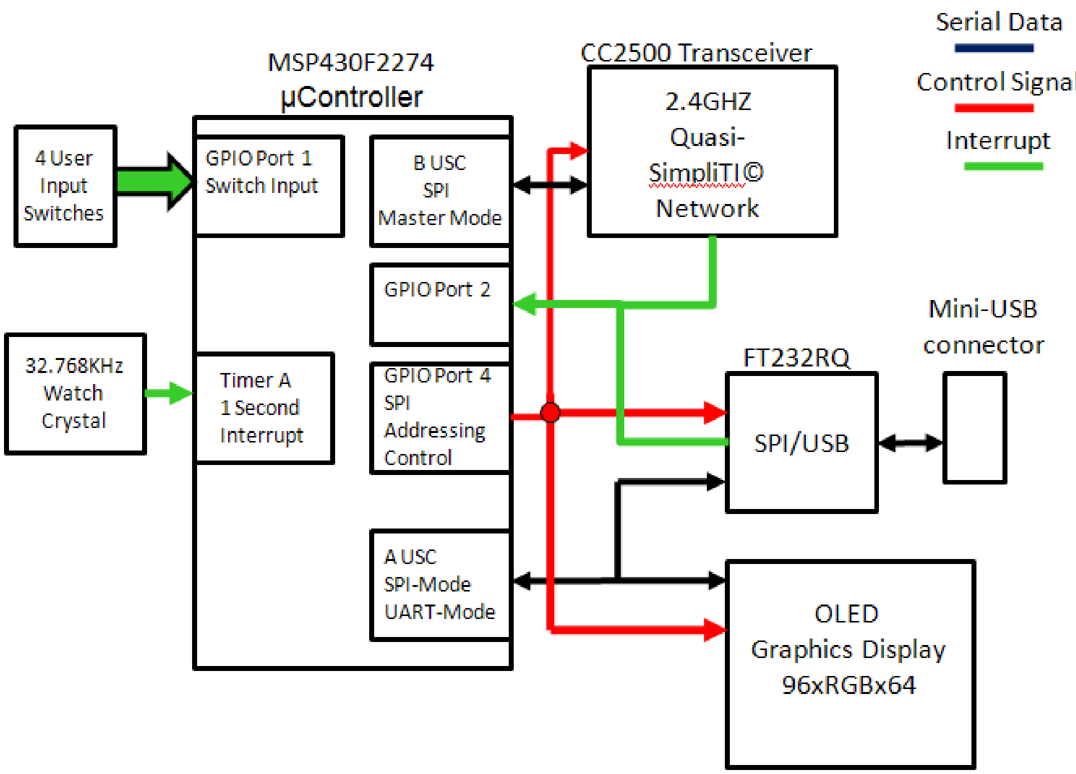

Figure 4-WBT Block diagram

Timing functions for the WBT are generated with a 32.768 kHz watch crystal used to generate a 1-second interrupt from Timer A of the MSP430. The four user input switches, SW1 through SW4, allow user interactions with the unit. Visual data is presented to the user via the OLED graphics display.

4.1.1.2 User Input

The SW1 input is used as a start for a track session, cessation of sessions is automatic. SW2 is turns the OLED panel on and off. SW 3 and SW4 adjust the OLED panel contrast up and down. These adjustments, other than start, are done prior to a track session. Figure 5 shows the switch locations on the WBT prototype.

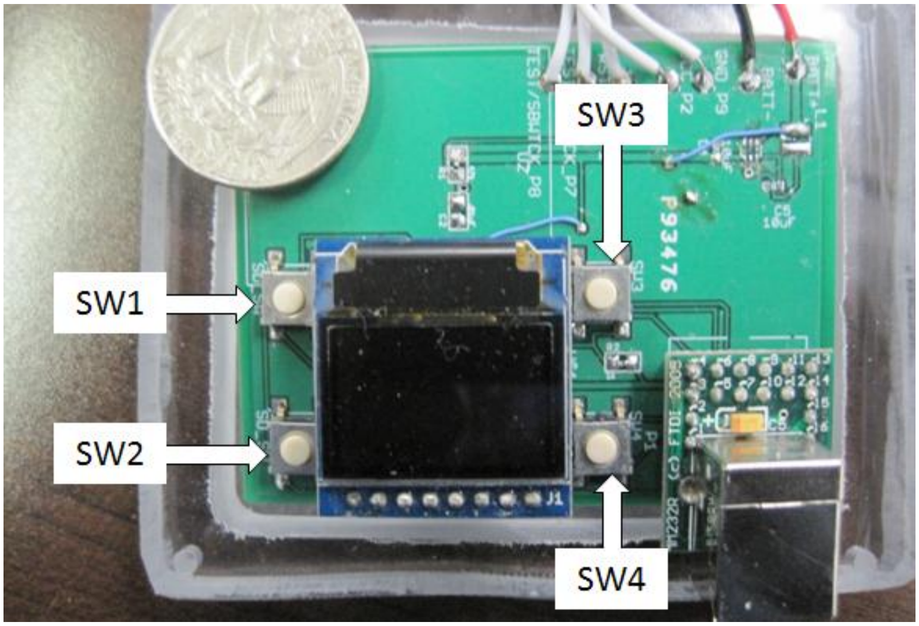

Figure 5-WBT Prototype user input switches

4.1.1.3 WBT Connectors and Mechanical Data

The main printed circuit board of the WBT contains the microcontroller, the USB to UART conversion module, the power subsystem, the OLED display and the modified EZ430-RF2500 section. External connection locations to the PCB are hardwired to solder pads and are shown in Figure 6.

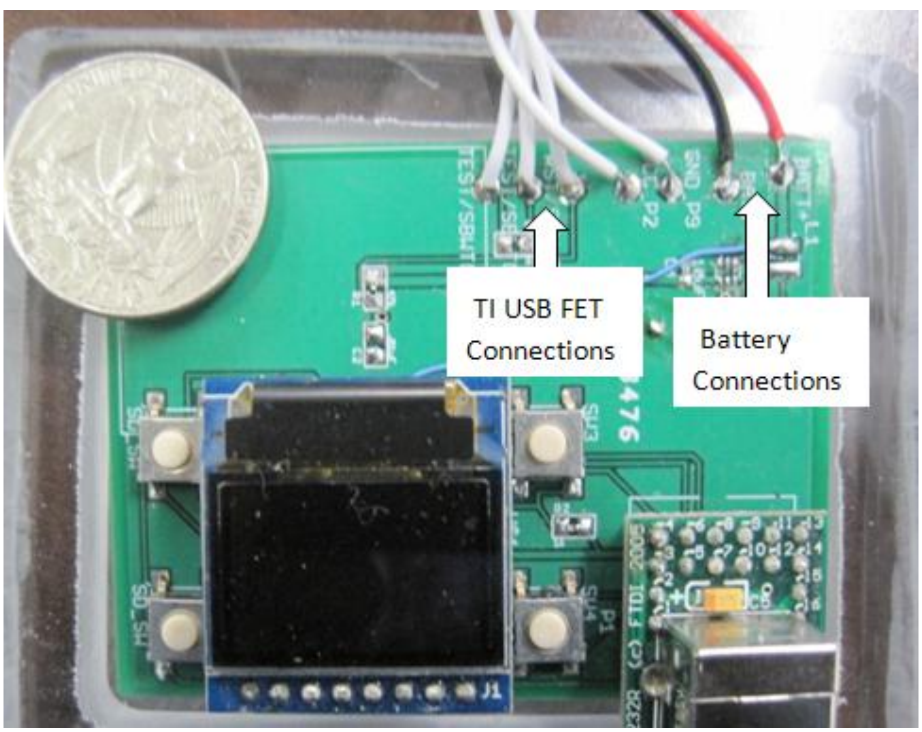

Figure 6-WBT Main PCB Connections

The schematic diagram for WBT prototype is shown in Figure 7. Detailed descriptions and sub-section schematics for the WBT PCB and the OLED panel operation details are given in sections 4.1.1.4 through 4.1.1.8.

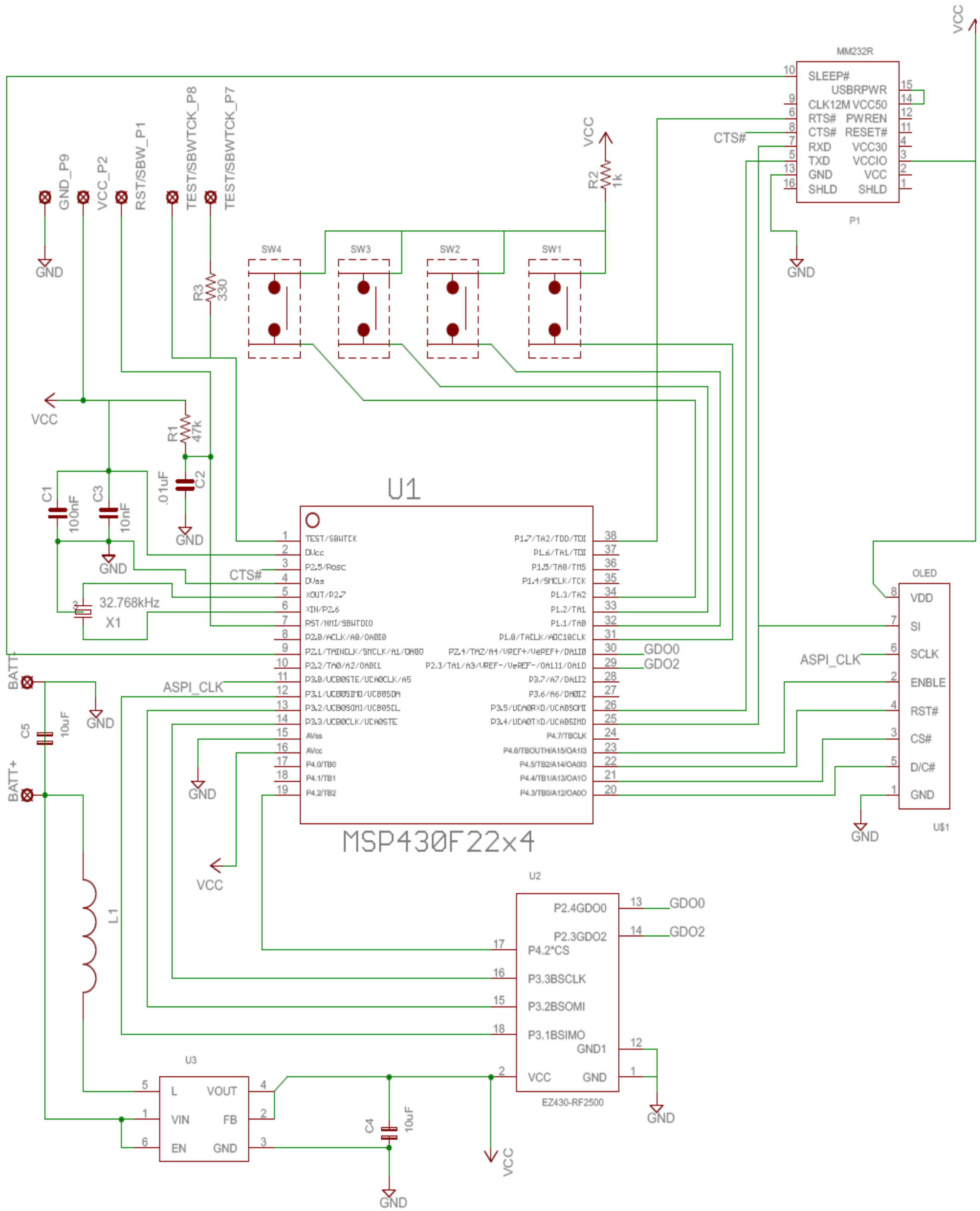

Figure 7-WBT prototype schematic diagram

4.1.1.4 Texas Instruments MSP430F2274 microcontroller

The processor of the WBT is the MSP430F2274 microcontroller, U1 in Figure 7 and Figure 8. This IC is a 38-pin thin-shrink small outline packaged (TSSOP) microcontroller. Two internal Universal Serial Controllers (USCA and USCB) are available. USCB is operated in Serial Peripheral Interface (SPI) mode only and provides the bidirectional serial data interface to the transceiver section. The second USC, A is switched in software between UART mode and SPI mode. UART mode is bidirectional to the USB module while write only SPI is used to the OLED panel. In addition to the USCA SPI signals, several of the digital I/O pins are used for control of the OLED display detailed in section 4.1.1.7. Programming and emulation takes place through a two-wire, spy-by-wire interface. Crystal X1 is the 32.768KHz clock input to the MSP430. This timing frequency makes second, minute and hour timing functions easier to realize.

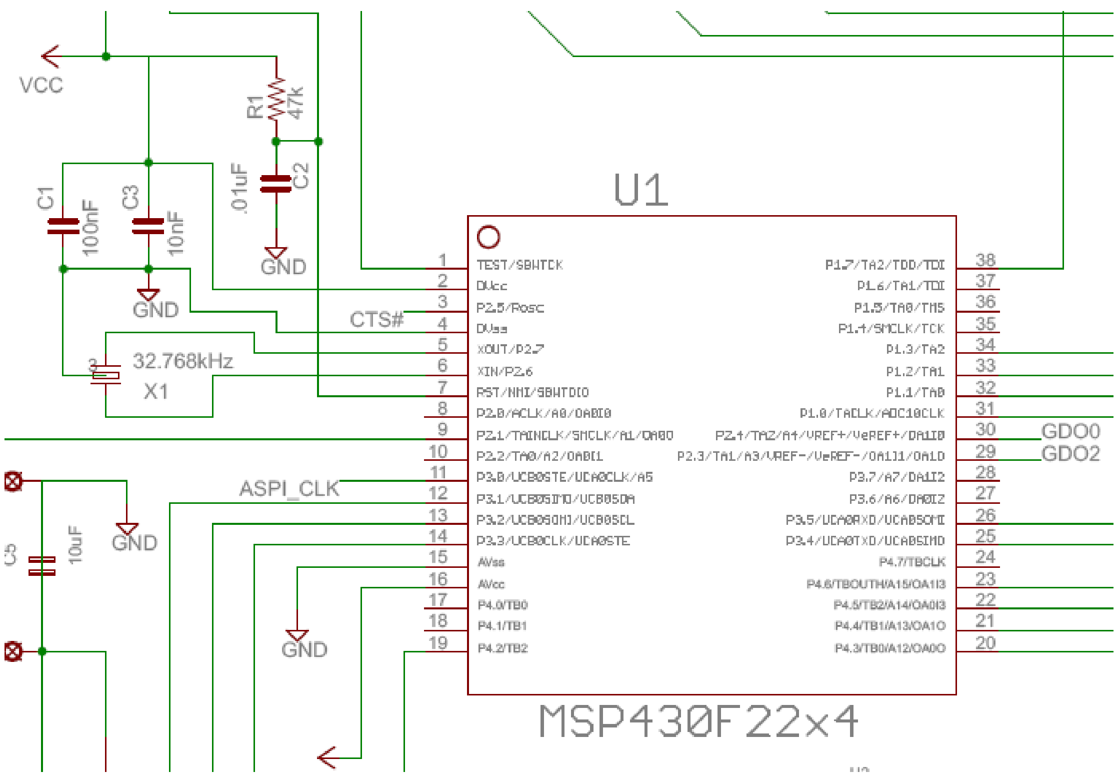

Figure 8-MSP430F2274 section and 32.768kHz crystal

The MPS430 and the 32.768kHz crystal are mounted beneath the OLED panel as shown in Figure 9. The OLED panel has been removed from its socket in this figure. The design saves real estate on the PCB and the use of a socket for the OLED panel was considered necessary for troubleshooting of the prototype.

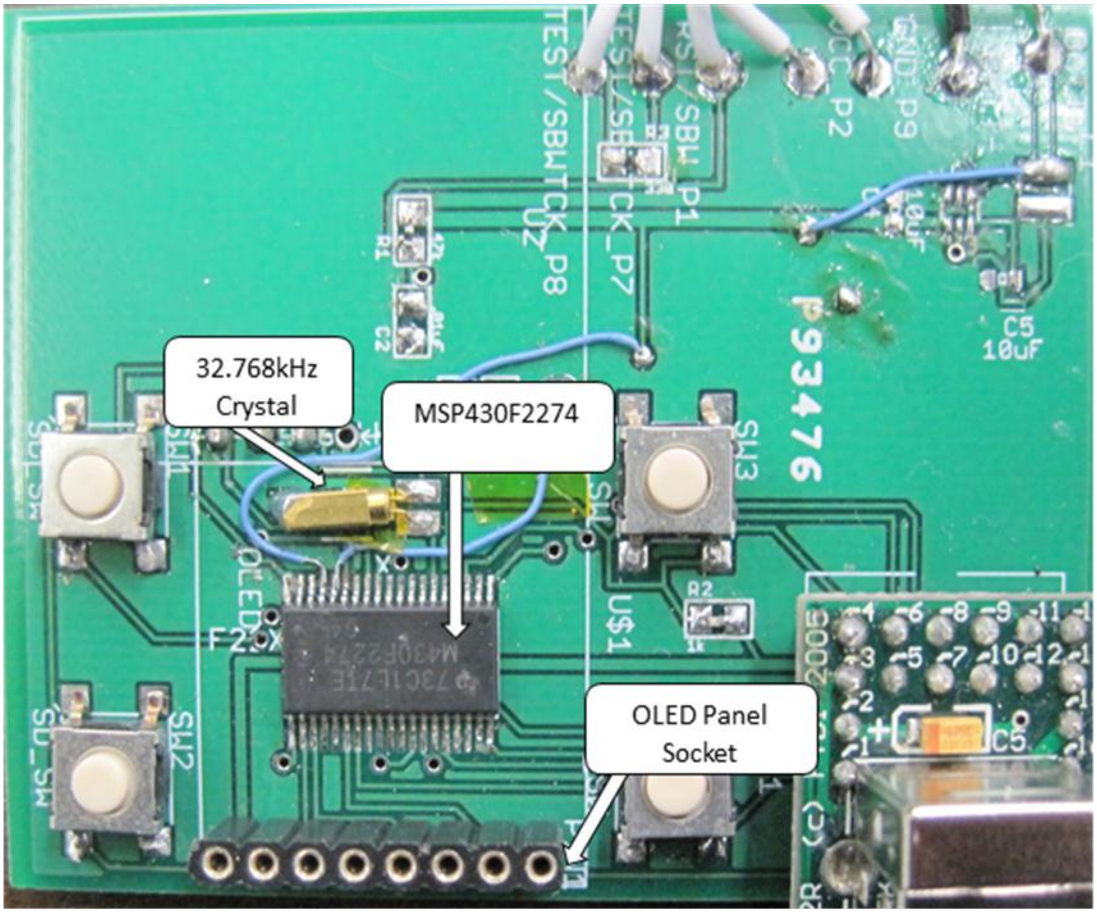

Figure 9-MSP430F2274, 32.768 Crystal and OLED Socket

4.1.1.5 Transceiver Section

The prototype system makes use of an off the shelf Texas Instruments eZRF430-2500 development board. The board is modified for use in the WBT and treated as an integrated circuit for the PCB layout. The MSP430 microcontroller installed on the board from the factory has been electrically removed (with an exacto-knife and microscope). The unit comes with pin holes giving easy access to the chip select, SPI interface, and GDO0and GDO2 pins of the CC2500 radio IC. The RF filter, 26.00 MHz crystal and the chip antennae remain undisturbed and form the radio for the WBT. Figure 10 shows a detailed image of the eZ430-RF2500 PCB treated as U2 (circled with dashed lines) in the detailed view from the main WBT schematic in Figure 7.

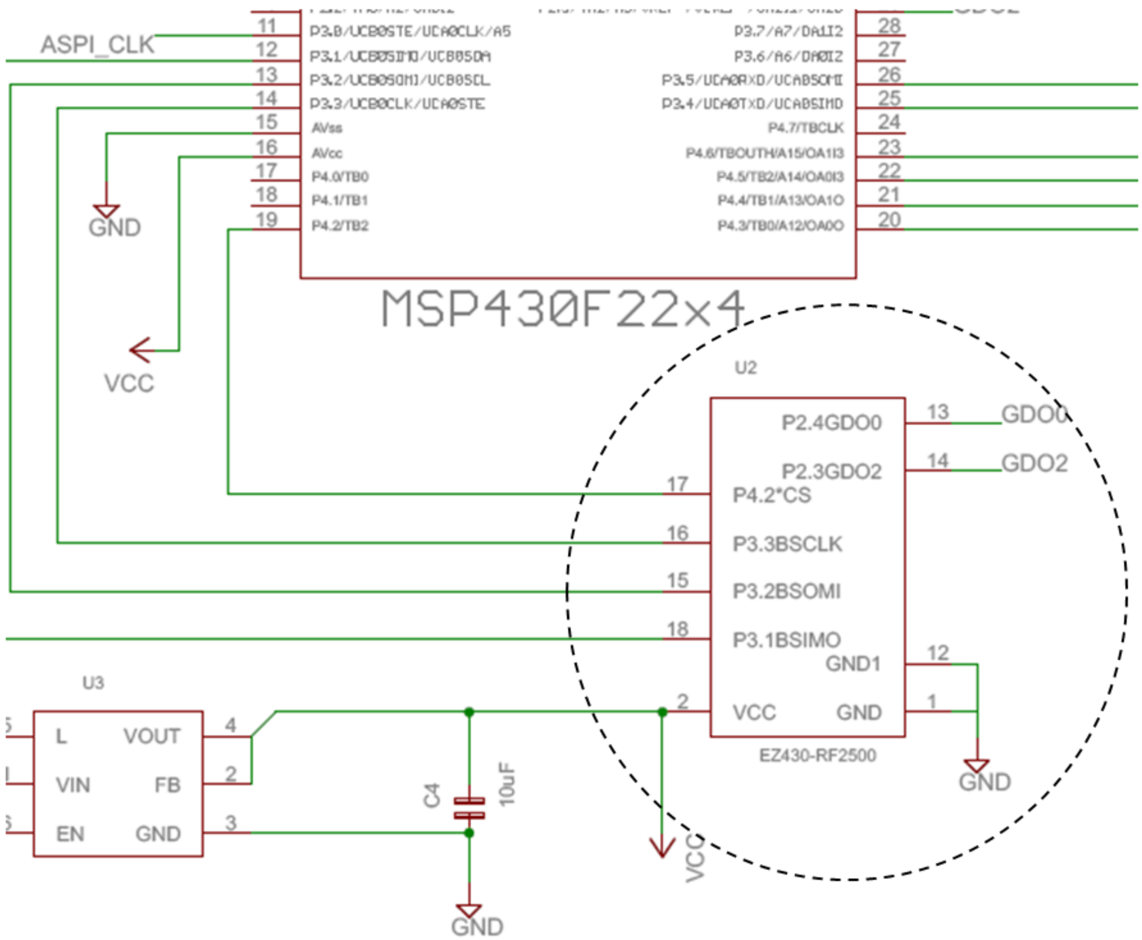

Figure 10-eZRF430-2500 section schematic detail

Figure 11 is the schematic diagram of the modified eZ430-RF2500. The pins in the dashed circle are connected to the WBT prototype. For the purposes of the prototype board design and in the main schematic diagram, these pins are treated as input and output pins of an integrated circuit. This saves development time for the prototype and eliminates the need to lay out a separate 2.4 GHz RF filter and antennae section.

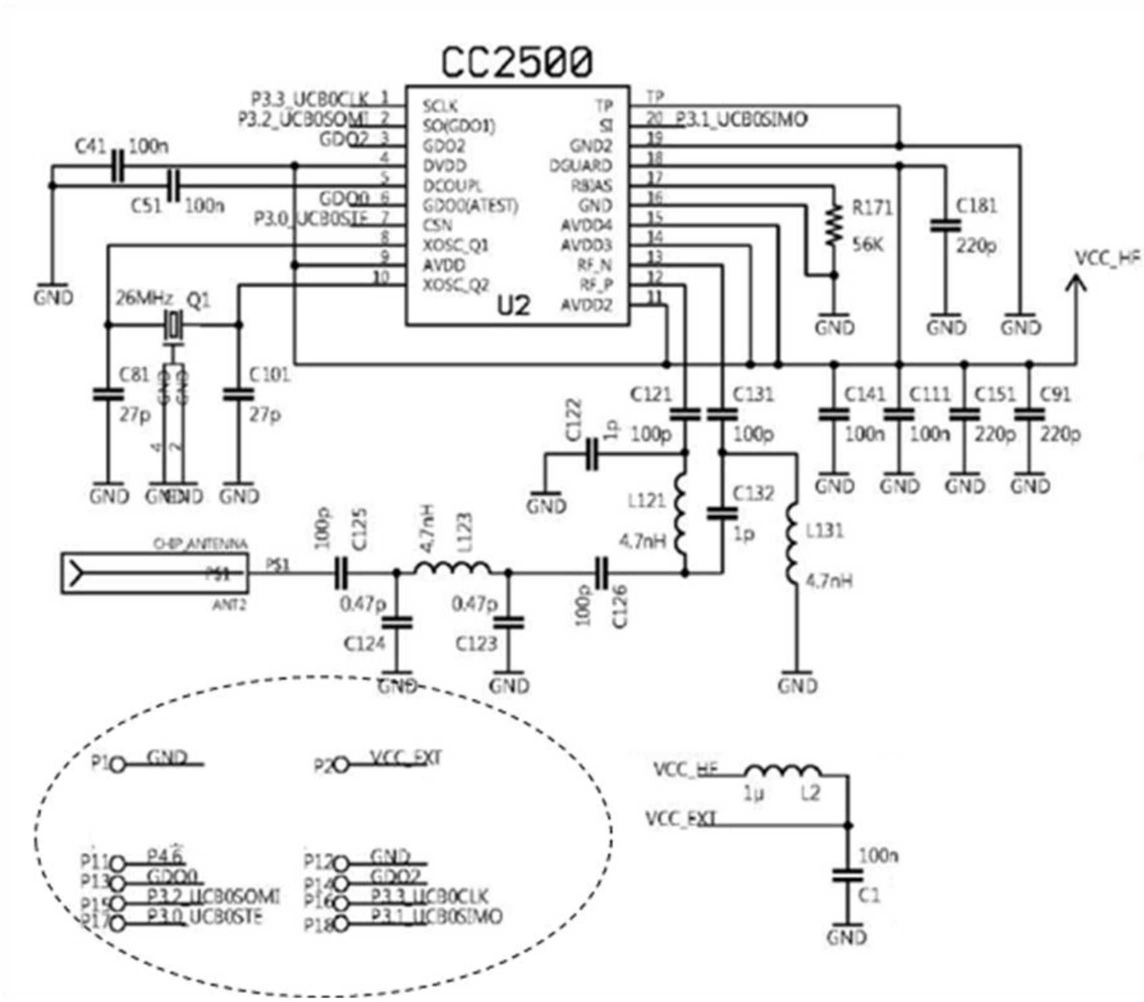

Figure 11-WBT modified eZRF430-2500 PCB (modified from TI schematic)

Figure 12 shows the eZ430-RF2500 soldered to the bottom side of the WBT prototype PCB. The MSP430 microcontroller is disabled on the eZ430 PCB.

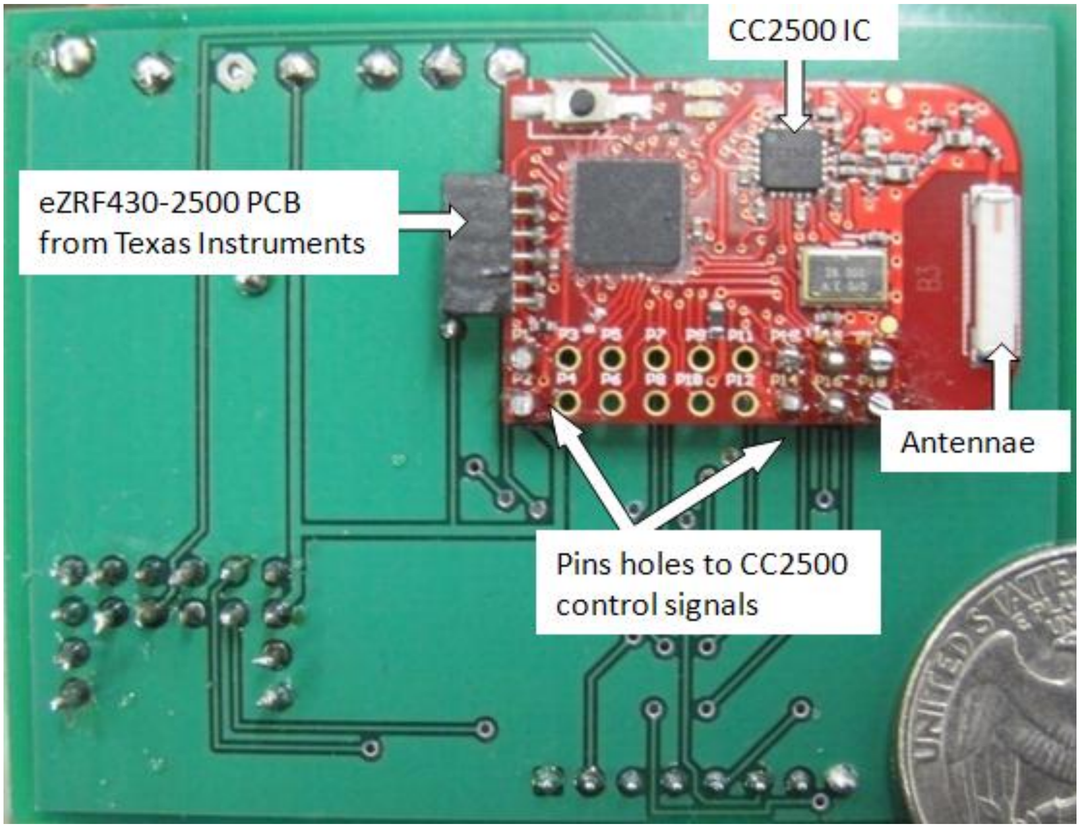

Figure 12-WBT Prototype Bottom View, eZRF430-2500 PCB location

4.1.1.6 Power Subsystem

The power subsystem as designed is shown in Figure 13**.** This design was not implemented for the testing of the prototype. However the design was subsequently tested. The design reduces the required input voltage to the WBT to less than 1.0 VDC. U3 is a TPS61221 boost convertor from Texas Instruments with a fixed 3.3 VDC output. The OLED panel requires a minimum of 2.4 VDC, the MSP430 and radio section will operate down to 2.1 VDC. The minimum VCC acceptable is 2.4VDC. 3.3 VDC is more than required. However, the improved performance achievable in both the radio section and the MSP430 speed indicates that this is an acceptable voltage level versus power savings compromise.

Power consumption in the WBT has been measured at 24mA with both the receiver in receive and the OLED panel in track display mode. Power drops as the radio section leaves a transmitter node by 13mA. No power management is done in software but a realistic expectation is that the unit will, on average, have a 1mAh power consumption while in use on a track with 16 transmitter nodes. The power section consists of L1, C5,C4 and U3.

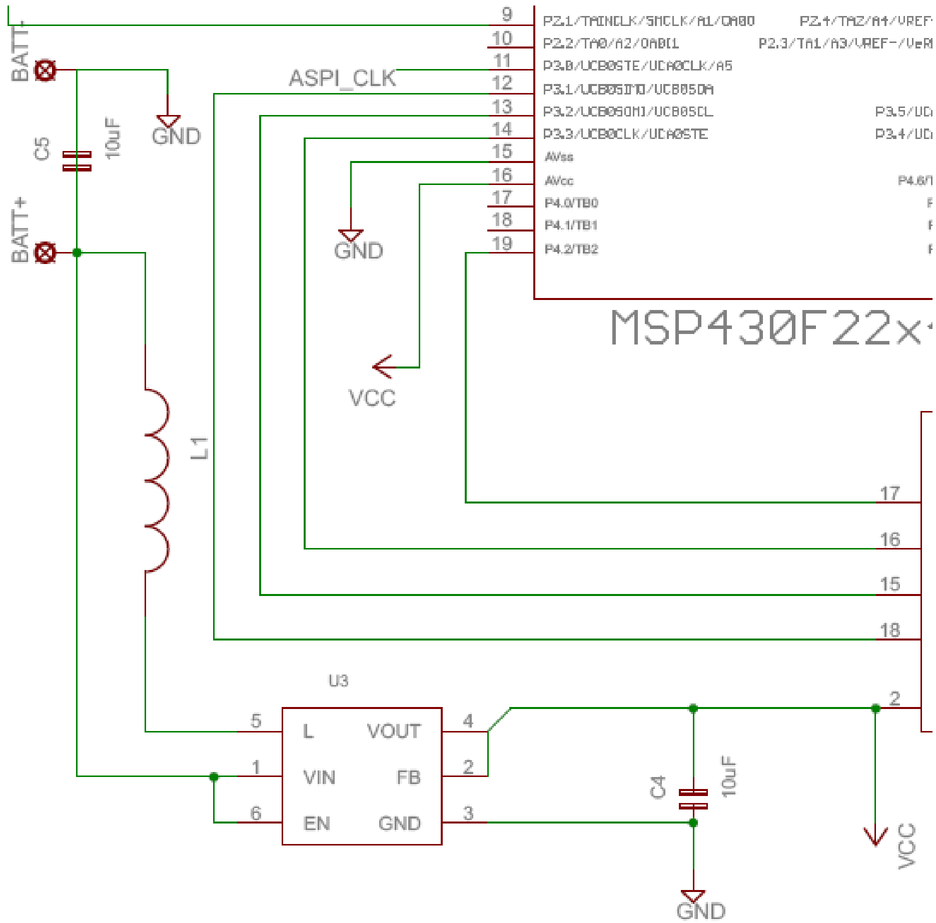

Figure 13-WBT Main PCB Power section

4.1.1.7 OLED Display and User Interface Subsystem

The OLED display is a CFAL9664B-F-B1 graphics OLED module from Crytalfontz America Inc. The resolution is 96x64 pixels. Color depth is either 65k colors in which case a 16 bit RGB value is used (5 red, 6 green, 5 blue) or 256 color mode using an 8 bit color value (3 red, 3 green, 2 blue). The WBT operates in 256-color mode to save MSP430 flash RAM space and to save write cycles to the panel. A 65k color write cycle takes twice as long as a 256. The display has a dark room contrast ration of 2000:1. The viewing angle is 160 degrees and the active viewing area is 0.95 inches diagonal. The unit comes mounted on a 2BM-10000 carrier unit that provides the voltage boost for the anode to cathode bias of the OLED elements. A SPI serial interface is also available on the carrier version. The carrier unit pins are inserted into a socket mounted on the prototype. Several I/O pins are needed from the microcontroller to ensure that data in not present when power is applied. Control of the power up sequence is mandatory.

The panel segments are controlled on board the carrier by a Solomon Systech SSD1332 CMOS controller and driver. This IC has a 96x64x16 graphics data SRAM. The device has a 256 step contrast control and a 16 step master current control. The reset line, connected to an I/O pin on the MSP430, will blank the display and reset the GSRAM address pointer to 0. The Data/*Command line from the MSP430 signals the SSD1332 whether the current SPI input is a command or data.

4.1.1.8 USB Section

P1 in Figure 14 forms the USB section of the prototype. P1 is a MM232R module from FTDI. A schematic diagram, courtesy of Future Technology Data International is shown in Figure 15. The IC used on the module is a FTDI FT232RQ. The IC is internally terminated for USB and automatically implements the USB to UART conversion. The RS-232 signals CTS# and RTS# are sent to the MSP430 but are not used. The SLEEP# signal goes high when a USB connection is made and provides the only interrupt from the USB section to the MSP430.

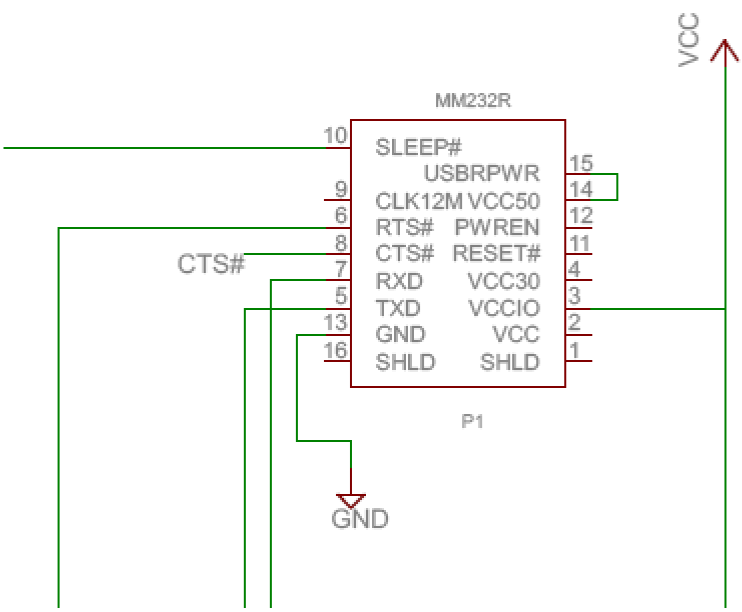

Figure 14-USB Section

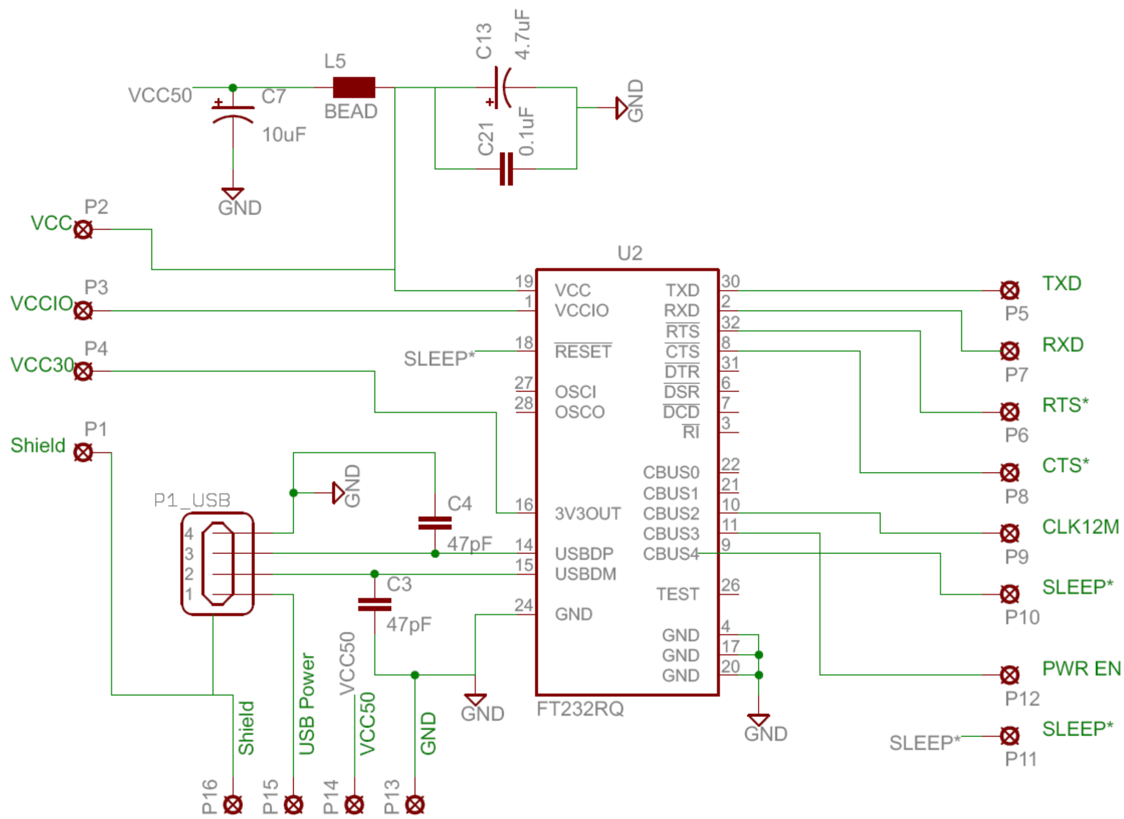

Figure 15-FDTI MM232R Module Schematic

If the TUT host system some interaction with the WBT may be undertaken from a PC running hyper-terminal or a similar communication program. The WBT must have just been reset (batteries removed and replaced). The OLED panel will indicate that no user is present. A USB cable is then attached (the interrupt for USB sensing does not occur when a USB cable is detached or statically in place). The hyper-terminal is started and for the VCP generated by the windows drivers. To start an interactive session with the WBT a carriage return must be sent from the PC. The WBT will prompt for data items, indicating the form the data item MUST take (including leading zeros). A sample session is shown in Figure 16.

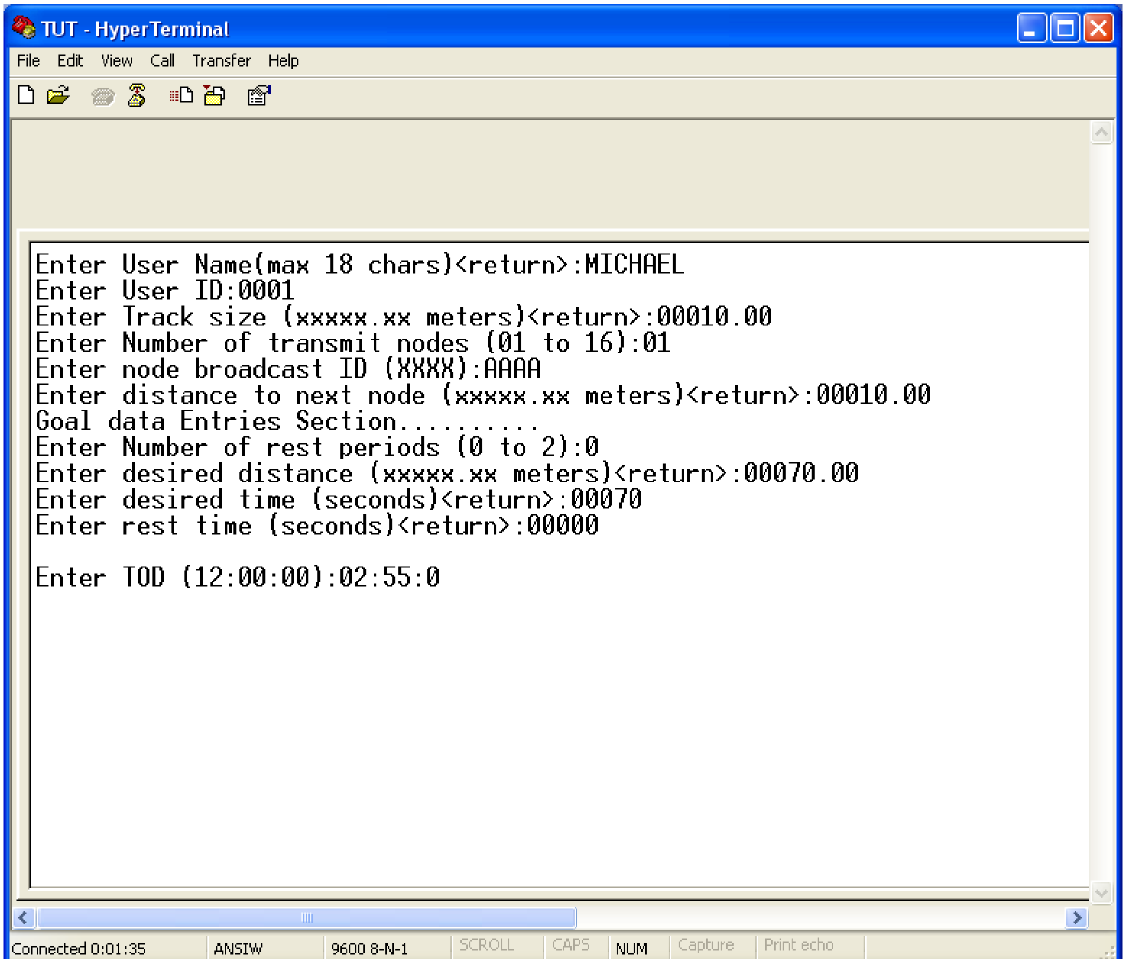

Figure 16-Sample hyper-term session

4.1.2 **Transmitter Node Design**

4.1.2.1 Transceiver and Transmitter Section

The RF filter section for the 315 MHz directional antenna portion of the transmitter node is shown in Figure 17. It has six capacitors and four inductors.

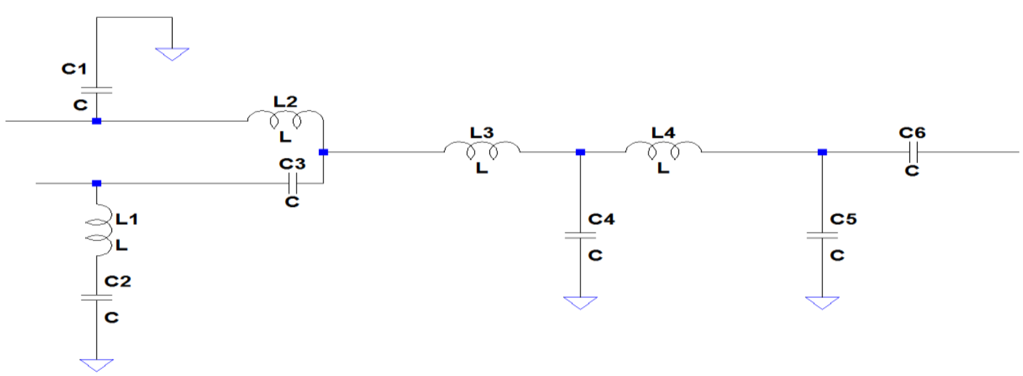

Figure 17: RF Transmitter filter

The filter values have been select to transmit a 315MHz FM signal. Values for the transmitter are listed in Table 4.

Components	Values at 315 MHz
C1	6.8pF +- 0.5pF
C2	6.8 pF ± 0.5 pF
C3	12 pF ± 5%
C4	6.8 pF ± 0.5 pF
C5	220 pF ± 5%,
C6	220 pF ± 5%,
L1	33 nH ± 5%,
L2	33 nH ± 5%,
L3	18 nH ± 5%,
L4	33 nH ± 5%,

Table 4: RF Transmitter Filter Components Values

The Transceiver section for the host wireless connection is as show in Figure 18. Table 5 list the component values and tolerances chosen to form an 868MHz FM RF filter. Both the transmitter and transceiver section will use SMD components.

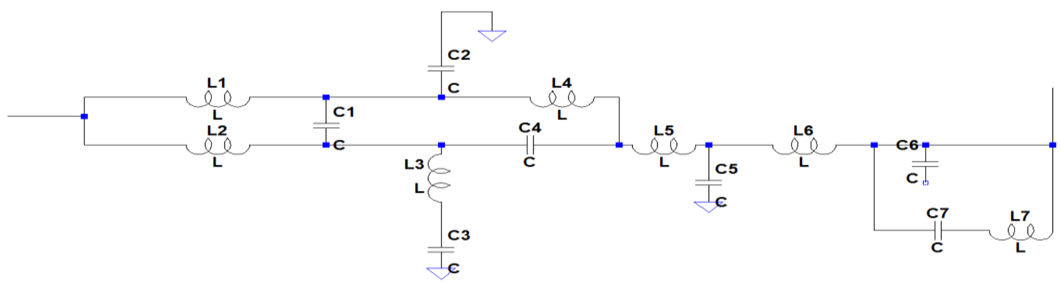

Figure 18 - RF Transceiver Filter (CC1101)

Components	Values at 868MHz
C1	6.8 pF ± 0.5
C2	6.8pF +- 0.5pF
C3	220 pF ± 5%
C4	12 pF ± 5%,
C5	6.8 pF ± 0.5 pF
C6	220 pF ± 5%,
C7	220 pF ± 5%,
L1	33 nH ± 5%
L2	6.8 pF ± 0.5 pF
L3	18 nH ± 5%,
L4	33 nH ± 5%,
L5	33 nH ± 5%
L6	33 nH ± 5%
L7	33 nH ± 5%

Table 5: RF Transceiver filter components values

Figure 19 below, describes the block diagram of the transmitter node. On the left right are the power supply and the microcontroller MSP430F2274. The microcontroller interfaces with the RF transceiver CC2500. Above the RF transceiver is the RF filter section which is connected to the directional antenna via the RF Connector. The main function of the transmitter node is to broadcast signal to the WBT. This transmitter node is designed for short range and indoor activities. The transmitter nodes do not have to be equidistant around the track. The main function of the transmitter node is to be able to give a user every time he or she passes near a transmitter node, he or she speed, prescription, and how far he or she has to exercise.

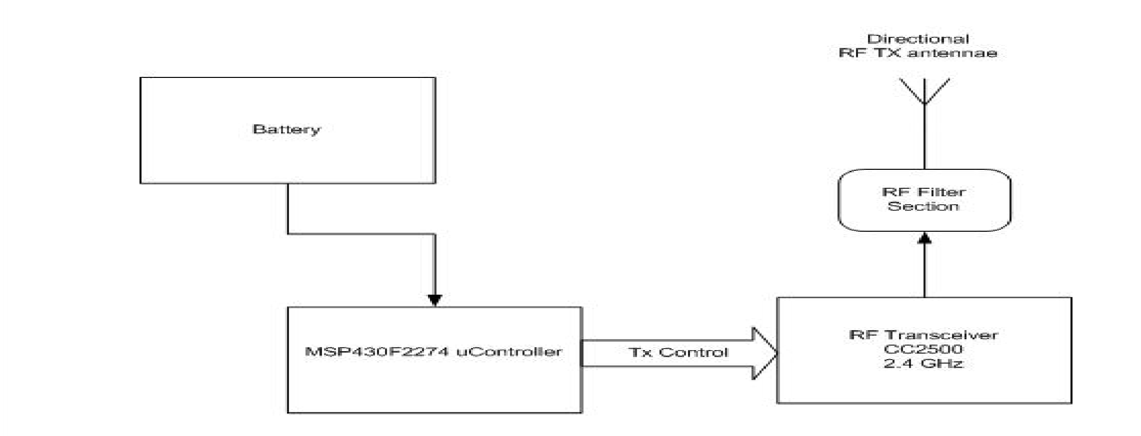

Figure 19 – Transmitter node block diagram

Figure 20 is the schematic of the RF transceiver. The basics function of the RF transceiver CC2500 is to transmit signal to the WBT via the patch antenna. the CC2500 is a low cost device. It works at 2.4 GHZ frequency. It is designed for very low power wireless application, and short range device frequency range. It has 20 I/O pins, and it is manufactured by Texas Instruments.

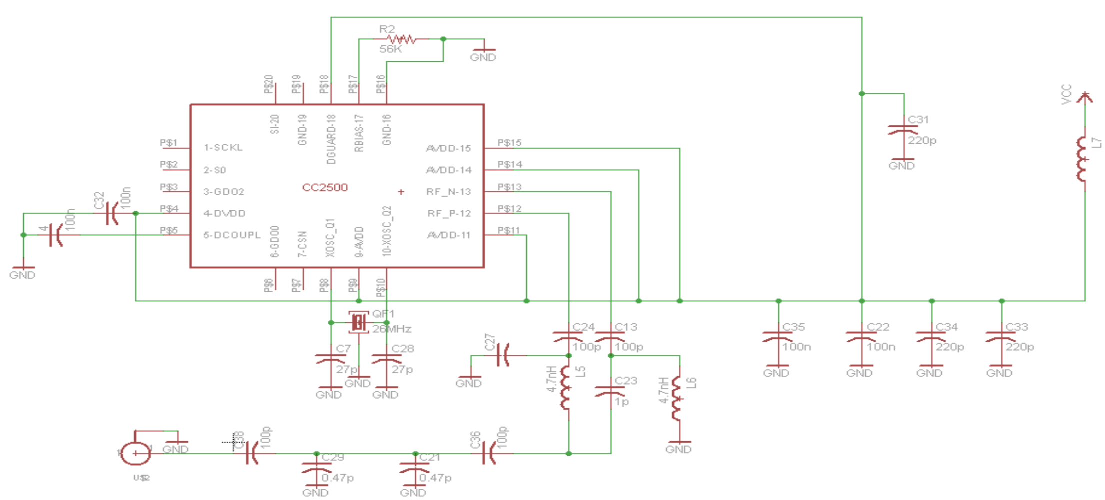

Figure 20: Schematic of the RF transceiver CC2500

4.1.2.2 Transmitter node design section

Figure 21 below is the transmitter node schematics. This schematic is based on the EZ430-RF2500T schematic. The MSP430F2274 microcontroller is in center of the schematics. On the right, we have the RF transceiver CC2500 and the RF filter section. The filter section is connected to the RF connector. The RF connector has two pins connected to the last capacitor of the RF filter, and to the ground respectively. On the far left of the schematic, we have the voltage supply, and the Connector header used to program the transmitter node.

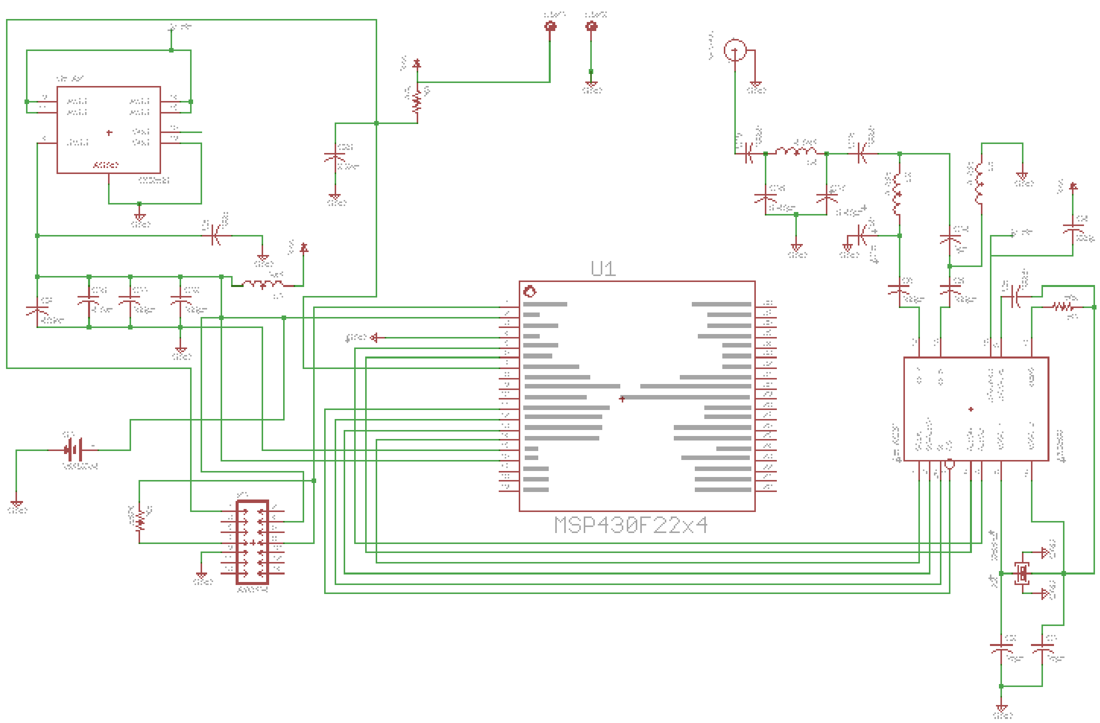

Figure 21: Transmitter node schematics

Six pins of the microcontroller MSP430F2274 was used to program the CC2500. Four of the pins form the serial communication interface (SPI). These pins are pin9, pin 10, pin 11, and pin 12. The remaining two pins (pin 2 and pin 3) are used by the RF transceiver to wake up the microcontroller. The microcontroller configures the RF transceiver to trigger GDO0 for proper operation.

Figure 22 is the actual transmitter node designed by the group. The red device connected to the directional antenna is the EZ430-RF2500T manufactures by Texas Instruments. It has been modified to meet our design specifications. The chip antenna of the target board has been removed from the board, and replaced by the RF connector. The RF connector has been solder to the pins of the old chip antenna. The RF connector was connected now to the directional antenna via a cable which was custom made by L.COM. For good performance the directional antenna has been mounted in the vertical position. The voltage supply is 2 AA batteries located in the black boss on the table. The transmitter is controlled by a switch mounted on top of the batteries holder. The turn the transmitter on, the switch is pushed toward the yellow mark on the right. To turn the switch off, push the silver switch on the left (the actual position).

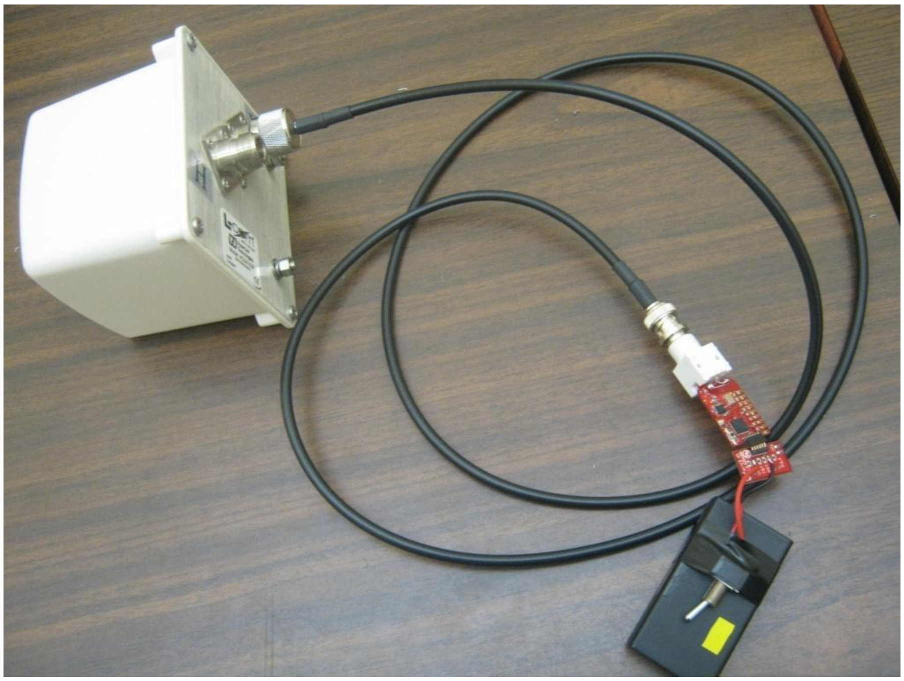

Figure 22: Transmitter node

4.1.2.3 Design Issues

The transmitter design node has been a challenge. Many reasons explain why it was difficult to design the transmitter node. First all we do not have the background to design RF transmitter. The RF filter section required precision and some knowledge about how far the resistors, Inductors, and capacitors are away from each other. Failing to do so creates a filter inside a filter which changed the design completely. The second reason is the cost of a PCB. We found out that buying a transmitter was actually cheaper than designing one. A transmitter PCB cost 70 dollars while buying one cost 20 dollars. The third reason is the fact that the resistors, capacitors, and inductors were all surface mount. Those parts are so tiny hat they were not easy to solder. It took time, patience, and precision to solder the RLC components on the board.

Figure 23 below represents the designed transmitter in green, and the target board in red. The filter section of the designed PCB is bigger than the target board from Texas Instruments. This is the problem that the group faced when trying to design this transmitter node. The components of the RF filter section have to be as close as possible to avoid creating a filter inside a filter. The target board has the MSP430F2274 microcontroller as well as the RF transceiver CC2500, and the RF filter section, but it is still small. The lack of experience in RF design, and the tools to design one have been one of the main reasons why the designed transmitter has not been able to work.

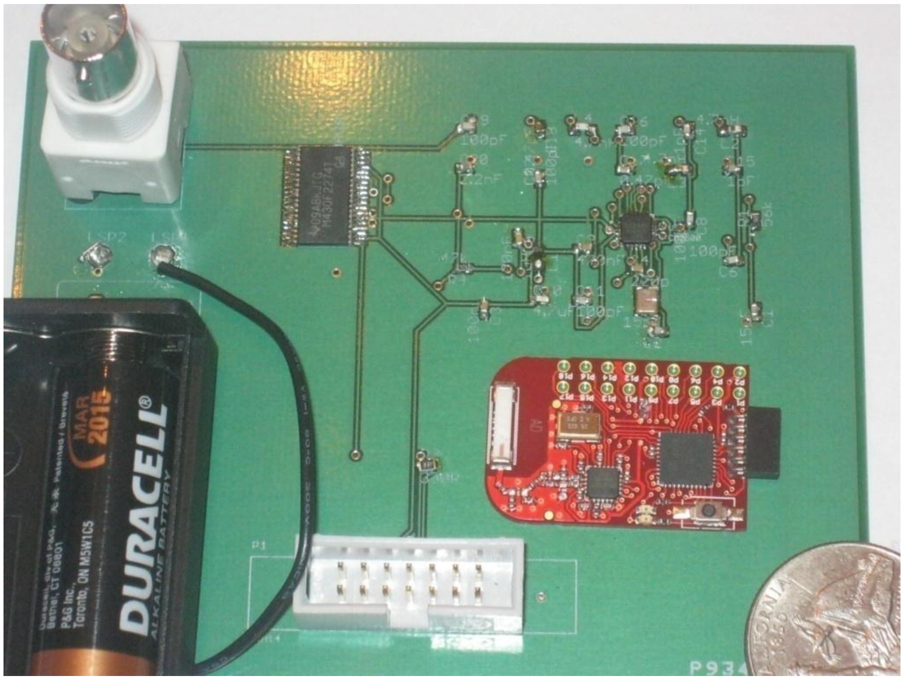

Figure 23: Transmitter PCB versus EZ430-RF2500T

Considering the fact that there is no sponsor for our project, budget constraint, and we have the assurance that the EZ430-RF2500T worked, we decided to buy the EZ430-RF2500 from Texas Instruments that we modified to meet the requirements of our design. The chip antenna of the target board has been removed. A RF connector has been solder to the pins of the chip antenna in order to be able connect the directional antenna to the transmitter.

The transmitter is self-powered. At the beginning of the design, we intended to use a solar panel as voltage supply, but we had to change it to simple batteries because of the problems that we were facing with the solar panels. First of all the solar panels were supposed to give a maximum output of 6 V and 50mA. When tested, the solar panels only provided 1.5 volts and 1mA under light in the lab. We put in series 2 solar panels in order to generate at least 3V, but when tested, we only had an output of 2.4 V under the best condition. To avoid during demonstration to run below the required voltage of the transmitter, we decided to use batteries as voltage supply. In addition, the EZ430-RF2500 came with two AAA batteries that we decided to keep, and use.

4.2 Software Design

4.2.1 Data Base Design Details

The TUT database was the easiest subsystem to implement. Before we could proceed with the design of the database we first needed to identify what types of data we would maintain and the relationships between the data. As the TUT design summary suggests, there are four different user types each with their own associated data. We created a data table for each user type and specifically described every data field in the table. To define the data fields, we broke down the data into its component parts. For example if we wanted to store an address in the database we first separated the address into the house number, street, city, state, and zip code. That means that we have five separate fields or columns for the address data. Similarly, name was separated into first name, middle initial, and last name. The first user type is called a patient. The patient has the most data to maintain because the data will also be used by the patients' physician. The data fields for the patient table consist of a unique patient identification number followed by patient name, address, email address, birth date, height, and weight. When we define a column in a database we also must define the type of data each column will hold. The data types are standard data types such as integer, decimal, date, time, character, or text. You can see in the entity–relationship diagram in Figure 24 that the data type is listed immediately after the column name in each table. The number in parentheses tells us how many characters are in this field. The next user type is the casual user. A casual user is a person that is not a physical therapy patient, but they use the gym and they are able to use the TUT system to enter goals for their exercise session and retrieve past session data in the form an exercise log. The casual user table does not need as much data as a patient table because the casual user is the only one able to access their information and it is only used as a log of their track usage. The casual user table consists of a unique user identification number, user name, email, birth date, height, and weight. The third user type is the physician. The physician is a user that can issue a prescription to a physical therapy patient, and retrieve patient data for statistical analysis. The physician table has a unique physician identification number, physician name, name of the practice where the physician works, practice phone number, practice address, and email. Finally there is the administrator user type. The administrator has access to system parameters such as track size, and number of transmitter nodes installed. The administrator can also add new casual users to the TUT system. The only data maintained for this user type is a unique identification number, name, and password.

In addition to the user type tables, we maintain a data table to store prescriptions as well as tables to store the exercise logs for both patient and casual users. The first column for the prescription table has the patient identification number for whom the prescription is issued. This allows us to link the prescription to the proper patient. This table also has a unique prescription identification number which is just an absolute sequence number. This number starts at one and is automatically be incremented for each new prescription issued. Following the identification number, the prescription table has from and to dates for which the prescription is valid, the length of time that the patient must exercise, the days of the week that patient must exercise, the distance that the patient must travel during each session, and a text field for any additional notes to the patient. The prescription may not use every goal field, but it must use at least one of the goal types. Finally, the prescription table contains the physician identification number of the physician that issued the prescription.

The last tables in the TUT database are the exercise logs. These tables are exactly the same except that one is for patients and the other is for casual users. These two tables hold the exercise data from all previous exercise sessions. The data for the casual user is only accessible by the user whose identification number appears in that particular record. The data for a patient will is accessible by the patient and by the physician who is linked to that patient. A physician does not have access to any casual user data or any patient data for whom they did not issue a prescription. The log tables have a unique identification which consists of the date / time stamp for when the record was entered. These tables also contain a filed for the encoded exercise log sent form the TUThost subsystem which consists of the users identification number and the time is took the user to travel between each transmitter node for the duration of the exercise session. There are also separate fields for the user identification number and any user entered notes for the session. The user identification field is used to restrict access to the log and the encoded session data is used by the TUThost to display or email the exercise log to the user.

Figure 24 illustrates the entity-relationship diagram for the TUT system database. This diagram shows each table in the database along with the column names and data types for each column. This table also shows the relationships among all tables in the database. The diagram uses specific symbols on each relationship connector to define the type of relationship and whether it is a mandatory relationship or not. As it turns out, all of the relationships depicted are mandatory in both directions. A patient does have at least one prescription and may have several prescriptions. This is called a one-to-many relationship and is represented by the crow's foot symbol on the prescription side of the relationship. A patient also has the same type of relationship with the Plog table. Every time a patient exercises, they will have a record of the session. Therefore, the patient table has another one-to-many relationship which is between the patient table and the Plog table. The casual user table has a one-to-many relationship to the Clog table but no relationship to the prescription table. The physician table has a one-to-many relationship to the prescription table since a physician may issue many prescriptions to many different patients. The physician table has no relationship to the patient table because we already have the prescription relation to the patient and the prescription table has the physician identification number in it. The admin table has no relationships since the administrator has nothing to do with the exercise entities.

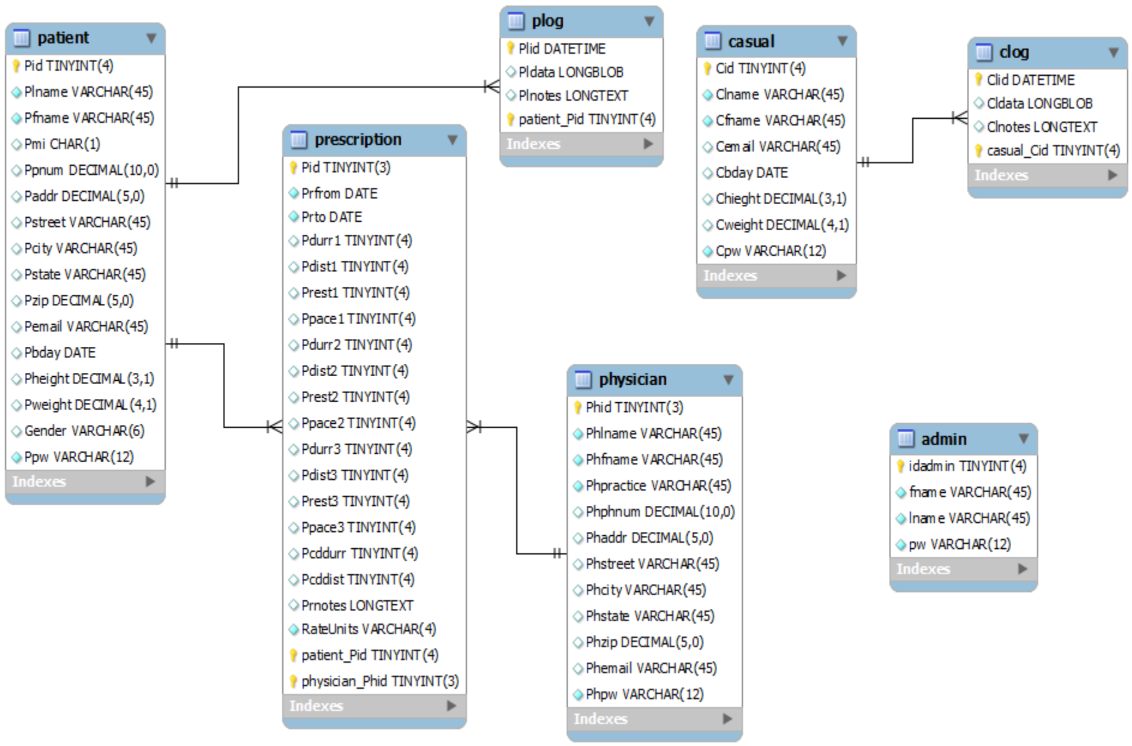

Figure 24-TUT database entity-relationship diagram

4.2.2 TUT Host Design Details

The TUT host software subsystem is written in Java using the object oriented programming paradigm. As such, we have broken our TUT host problem down into unique functional objects or modules. The host software consist of thirteen modules that we call classes which are instantiated as objects during program execution. Each class has its own attributes and operations that it can perform representing the communications available with each class object. All classes have GUI's for their interface except for the User, Wbt, Prescription, and ExLog classes. These four classes perform operations for other classes that do include a GUI. The main class, meaning the class which contains our controlling methods, is called TUTmain. The main method of this class is called main. Since this is a GUI based program, the main method instantiates the TUTmain GUI which performs the initial interaction with the users (log in functions). TUTmain also has methods for retrieving and updating data in the TUT database, emailing the exercise logs, issuing the WBT's, and telling the system that a wbt has been returned.

The Patient_main class provides the main GUI for the patient. This class also has methods to retrieve a prescription and sent it to the WBT, view the current exercise log after returning a WBT, and it has a method that requests TUTmain to email the users current exercise log. Casual_main has similar methods to Patient_main, but in Caual_main the user enters their own time and distance goals instead of retrieving a prescription.

The Phy_main class was implemented with two main options. There is a method to add a new patient to the TUT system and there is a method to write a prescription. Both methods call other classes to perform their work. If the physician selects the option to write a new prescription, the Phy_Main classes writescr method would call the Write_Scrip class which would display the GUI for entering new patient information. The Write_Scrip class would then send all new patient data entered to the database and then return to the Phy_Main GUI.

Our last “main” class is the Admin_main class. This class has the GUI for the administrator and provides methods for calling the administrator functions. There is a method for adding a new casual user to the system which reads the data input to the AddUser GUI and transfers that data into the TUT database. We also have methods for calling the Track_Config GUI to set up track size for WBT configuration and a method for testing system communications with a new WBT before it is put into service.

The Add_User and Add_Patient classes just provide the GUI for entering the user or patient information into the TUT system These two classes are called from the Admin_ Main and Phy_Main classes respectively. They also both contain what we call “getter” methods to retrieve the user data and send it to the database.

The Prescription class is in charge of retrieving and formatting prescription data prior to sending the prescription to the WBT. It has methods for retrieving the prescription form the database, determining if the prescription is new, if it is still a valid prescription, the date it was issued, and the physician who issued the prescription. We need to know the date it was issued to determine if it is still in effect or if it has expired. We need to know the physician who issued the prescription in order to validate whether it is legitimate and if it belongs to the current patient. These decisions are included in the isValid method. The isNew method simply checks the prescription issue date and compares with the last exercise date to see if it was issued subsequent to that date. The other methods just access the database and retrieve and format the desired information.

The ExLog class is responsible for retrieving and formatting the previous exercise session data and sending it to either the Casual_Main class or the Patient_Main class for viewing. If the user selects the “Email Exercise Log” option from their main GUI, then the prescription data is sent to the Elog method in the TUTmain class. It has a method to retrieve the log data from the database according to the date selected by the user. It also has a method to read the email address if it is to be emailed, and a method to allow the user to enter their email address if they don't already have one stored in the database.

In the WBT class we have all the methods needed to be able to configure a WBT for the current track layout, upload an exercise session to a WBT, assign a WBT to a particular user, and retrieve exercise session information from a WBT when it is returned. The isAssn method checks to see if a particular WBT is currently assigned to a user by looking at the "assigned" attribute in the WBT class instance. This attribute would be set by the TUTmain class method used to assign the WBT to a user. The isCharged method would use a system call to check the charge current from the USB connection to see if it is below the minimum set by the WBT designer to indicate a full battery charge. The isConf method is just checking a single bit in the WBT. Both load methods transfer data across the USB to the WBT memory. This data is supplied by the TUThost class and was either collected from the prescription class or the GUI from user input. Whether prescription or user entered exercise data, it is converted to the same format prior to upload to the WBT. The conversion takes place in the TUTmain class in either the wscript or wgoals method. The getEx method will retrieve session exercise data from the WBT when it is returned to the USB dock after an exercise session. This method then sends the data back to the TUTmain class. The TUTmain class will use the wbtRtn method to read the data and record it in the TUT database.

The User class is responsible for retrieving and setting user data needed by the TUT system while a user is participating in an exercise session. The system needs to be able to identify the user in order to retrieve the correct prescription data or previous session data. This class has methods for setting and retrieving user identification number and user name. The user name is divided into first name and last name attributes. The "get" methods retrieve data from the TUT database and send it back to the TUTmain class. The "set" methods get information from the GUI's and save it to the "User" class attributes.

Since there are two different user types, other than administrator, that will be accessing the TUT system, we have separate GUI's for each type. Each GUI is displayed and controlled by the users' main class and provides the options corresponding to their user type. Following are examples of both user GUI's with a description of the options for each. After logging in to the TUT host a physical therapy patient would be presented with GUI screen shown in Figure 25. The patient would first enter their patient identification number and their password in the designated text boxes. They would then click on the "Login" button to be taken to their main menu. The user would then see their first name shown in the welcome box and their user Id number so they could verify proper login. Now the patient is ready to select their options for the session. The options are limited for a patient because they have been issued a specific exercise prescription from their physician and cannot enter their own exercise plan. For this example we will say that the patient clicks the button labeled “Retrieve Prescription”. What will happen is the TUT system retrieves the prescription data, sends the data to the WBT, and prompts the patient that the prescription data was successfully retrieved. The patient then clicks the button labeled “I’m Ready to Exercise” and takes their WBT from the USB dock. The system will then log the user out of the GUI until they return the WBT after the exercise session.

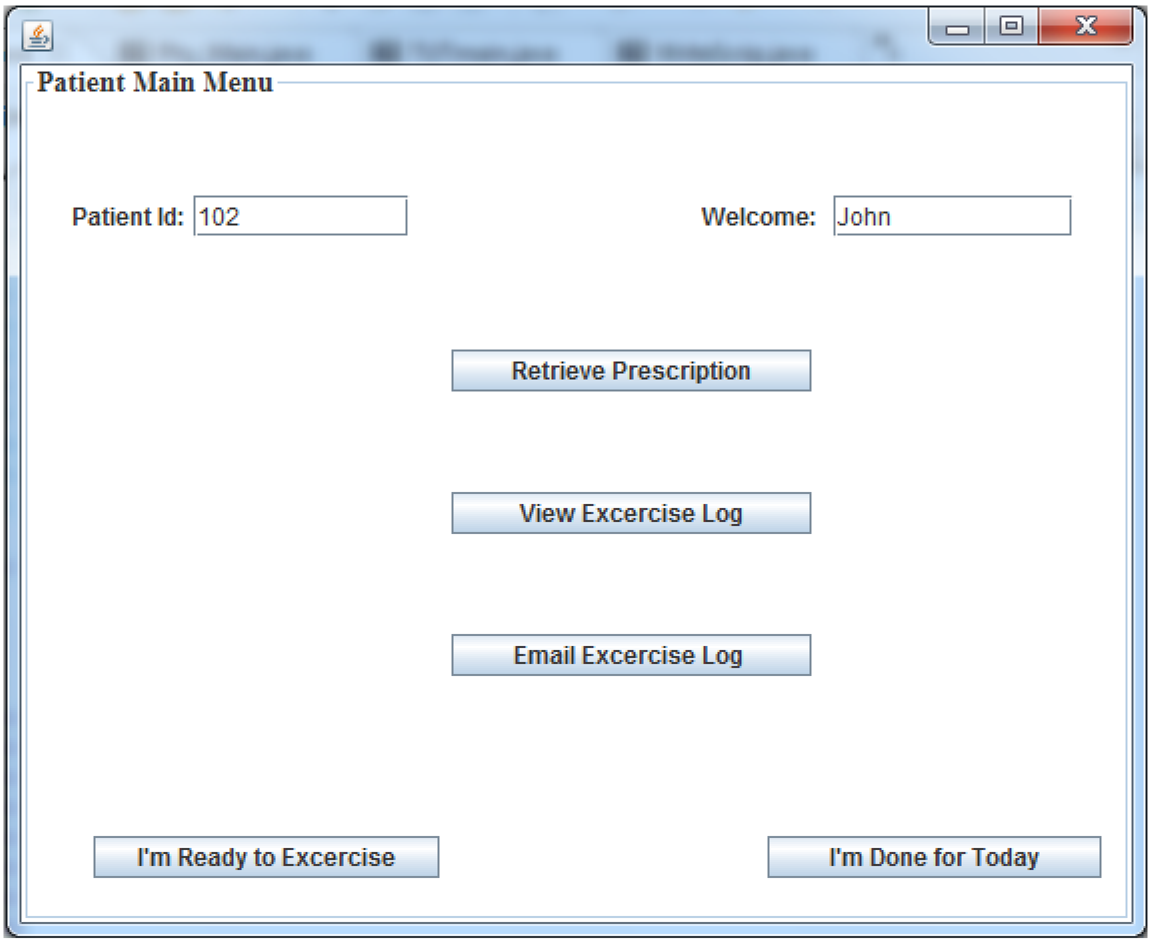

Figure 25-PM User GUI

A non-patient user, AKA SM user, has a similar interface. After logging in to the TUT host an SM user is be presented with the GUI screen shown in Figure 26. The user would first enter their patient identification number and their password in the designated text boxes. They would then click on the “Login” button to be taken to their main menu. Now the user is ready to select their options for the session. The options for a non-patient user include time and distance goals for the exercise session. The user does not have to specify both time and distance but must enter at least one goal type. For this example we will say that the user enters the number 45 in the box next to the time label and then click the button labeled “I’m Ready to Exercise!”. What will happen is the TUT system will then send the users goal data to the WBT. After that, the user will remove the WBT from the USB dock and proceed to the track for their exercise session. The user is automatically logged out of the GUI until they return the WBT after the exercise session.

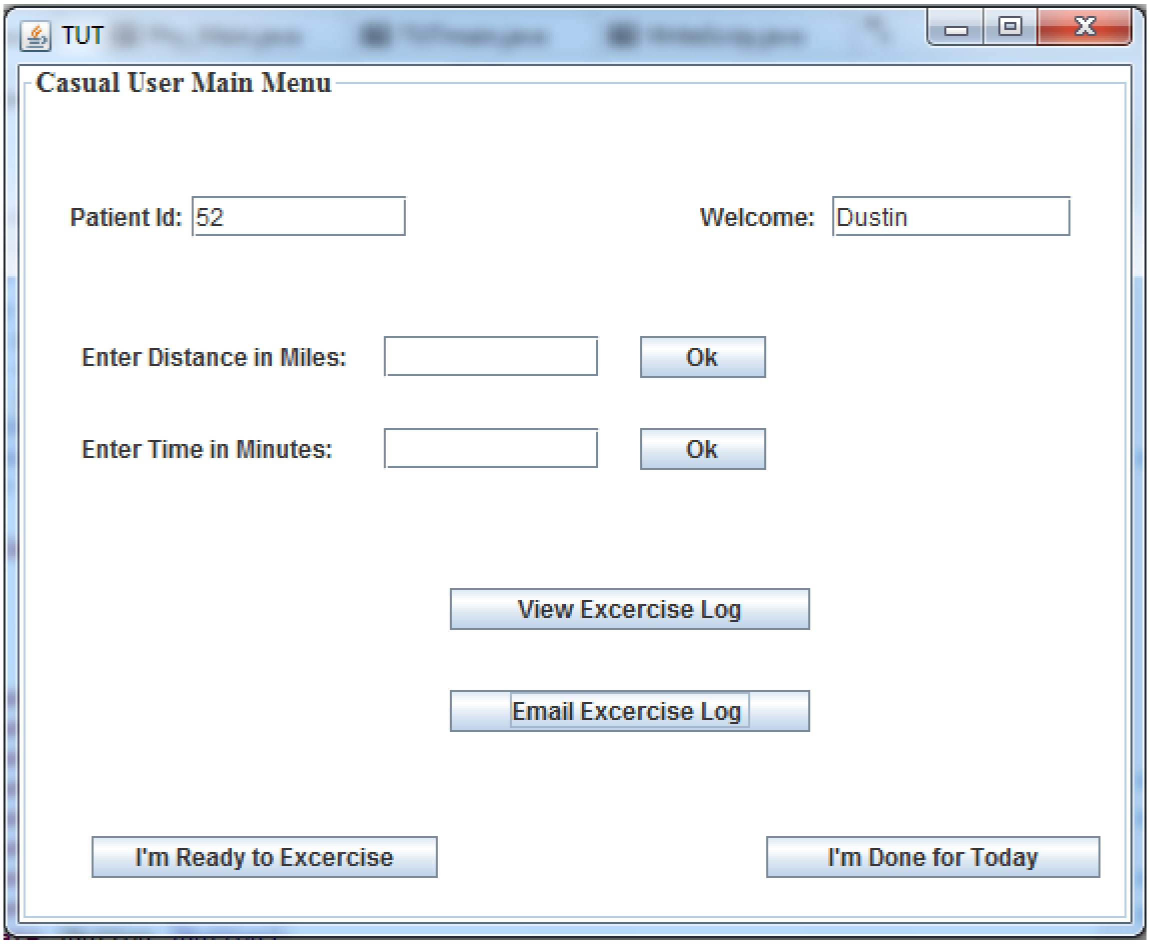

Figure 26-SM User GUI

The TUT host class diagram, shown in Figure 27, outlines the modules included in the host software subsystem. This diagram shows the relationships between modules, the data fields contained in each module, the operations performed by each module, the type of data contained in each field, and the type of data returned by each operation. Each box represents a class or module and each line represents an interaction between modules. For example, the TUTgui interacts with human users and the TUThost controls data flow between modules and interacts with the other modules.

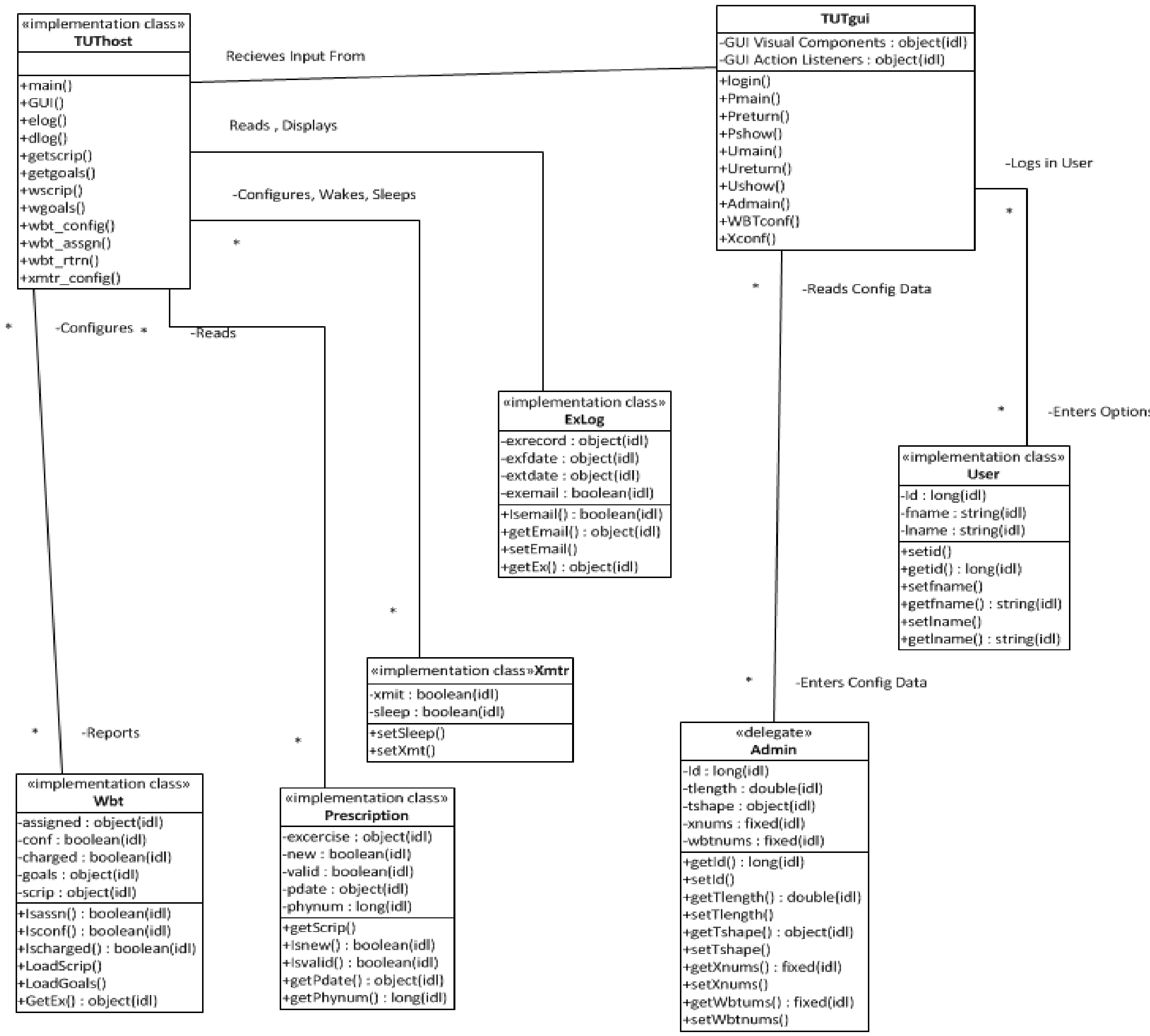

Figure 27-TUT Host Class Diagram

4.2.3 TUT Website Design Details

After logging in to the TUT website and selecting the option to issue a prescription they are presented with the graphical user interface (GUI) screen shown in Figure 28. The physician would first select the patient from the drop down menu box. They would then enter exercise parameters, cool down parameters, and select the units for the distance and rate. When they are finished entering parameters, they would click on the button labeled "Submit Prescription" and the data would then be saved to the database for retrieval by the corresponding patient. After that the physician can return to the main menu and from the main menu they can log back out of the system.

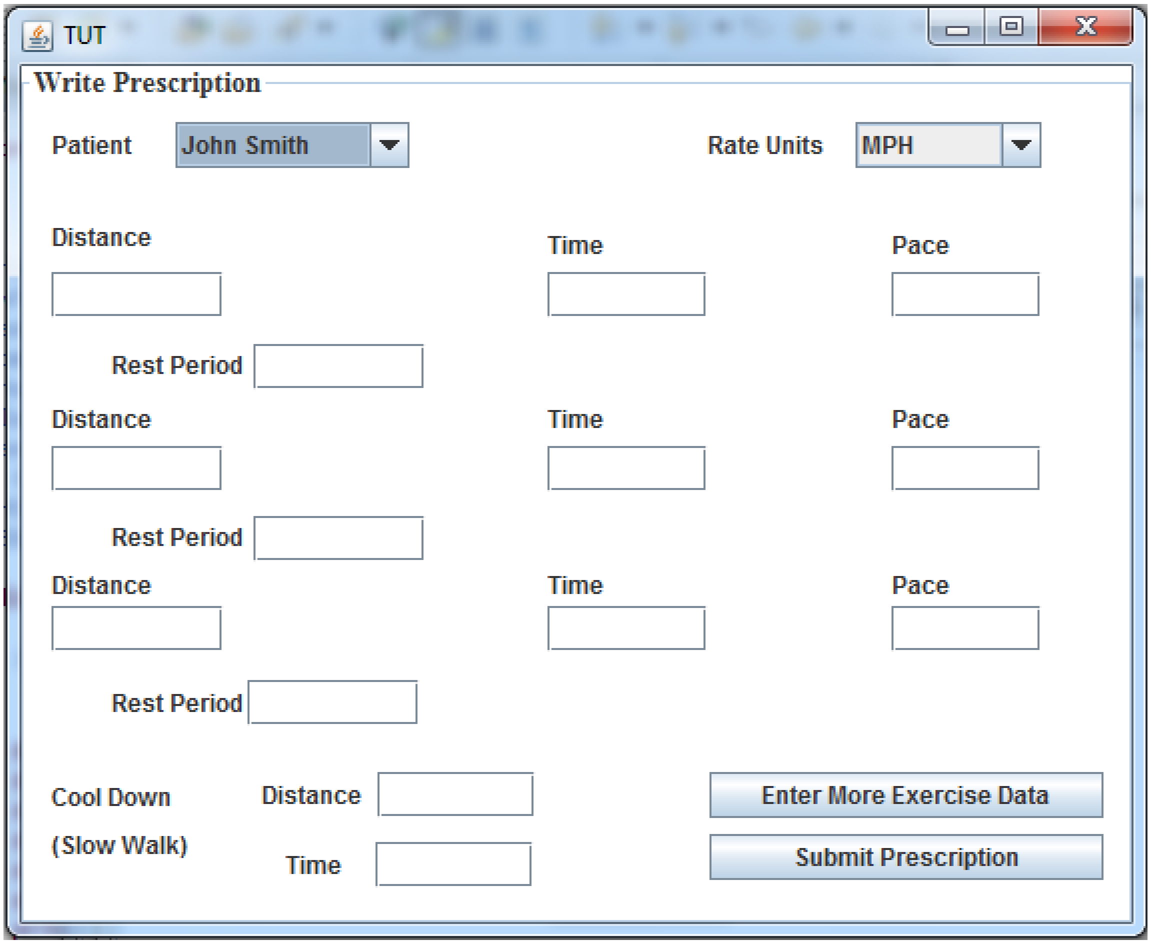

Figure 28-TUT Prescription Entry GUI

5 Design Summary

5.1 System Architecture

The TUT system is composed of both software and hardware. The TUT hardware is installed at participating facilities housing an indoor track. The TUT software components resides on a Microsoft Windows PC at the track facility and on a dedicated website as shown in Figure 29-Health Care Professional & Patient Interaction.Figure 29. User interaction with the system takes place both at the facility and through the internet. Health care professionals interact with and monitor patients through the website. Two TUT software applications, TUTapp and TUTconfig, are resident on the track facility host. The software has several functions and two distinct users, Patient Mode users (PM) and Self-Prescriber mode patients (SM).TUTapp handles user login, retrieval of exercise regimen prescriptions (for PM users), setting or checking of desired goal data (SM users), uploading and initializing a Wrist Band Tracker (WBT), and downloading track session data. It updates a personal exercise diary of the relevant user on the website database. Each health care professional with patients using TUT has access to that patient's exercise data. The professional will also have the ability to "prescribe" and modify exercise regimens for their patients at their discretion.

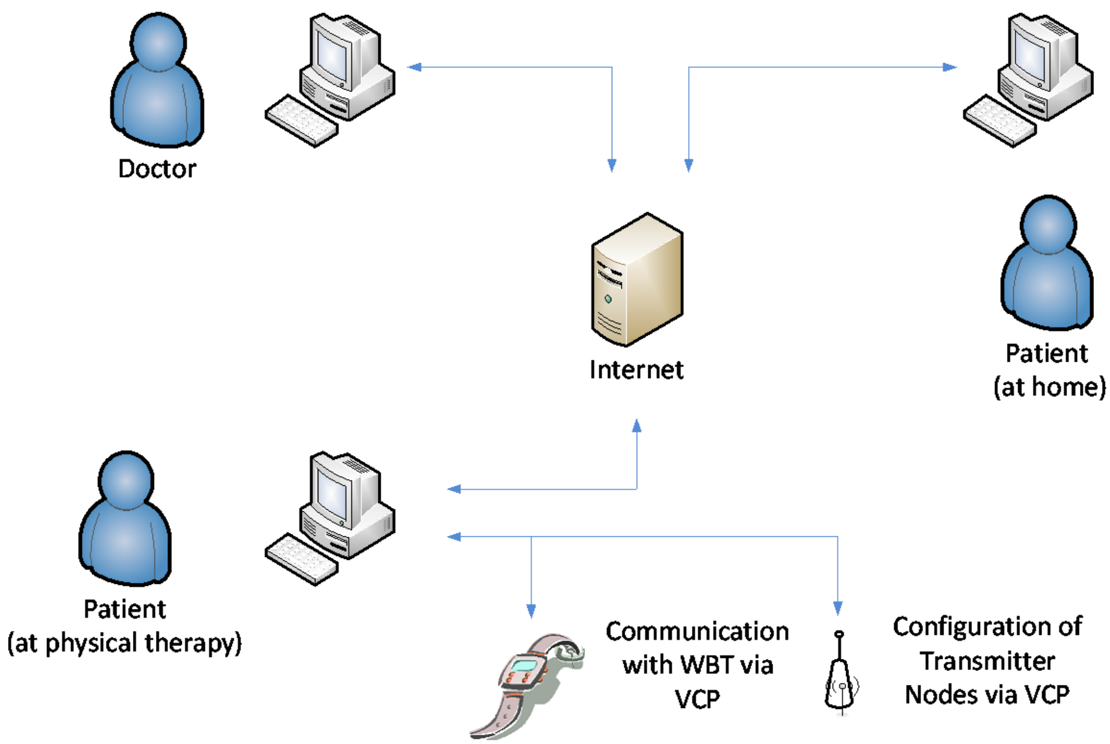

Figure 29-Health Care Professional & Patient Interaction.

SM users of TUT will have access both onsite and via the internet to their personal exercise diary. They will have the ability to "self-prescribe" exercise regimens, monitor their own progress towards set goals, and to set goals prior to a track session. TUTapp logs a user onto the system prior to track use via an already existing ID card and self-defined password. Health clubs normally issue such a card to each member. For example, the health and recreation center (HRC) at UCF uses each student's ID card; the password requirement is in order to protect the user's privacy. Each user who logs onto the system will be issued a WBT preprogrammed to the users particular needs or prescription data retrieved from the TUT website.

Interactions between the TUT hardware and the TUT host software takes place through two avenues. The host downloads track configuration and session related data and uploads completed session data to and from a WBT via a USB interface as shown in Figure 30. The software manages the transmitter nodes

over a wireless connection. The software will shut down all TUT transmitters at the end of the day or when no WBT is checked out by a user.
The system software does not monitor a user while they are on the track; the WBT stores the data for upload.

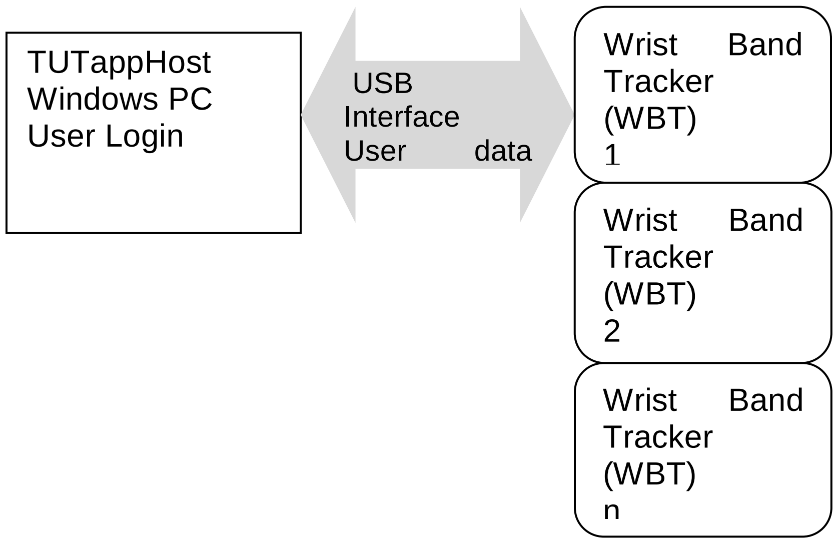

Figure 30-Interface WBT to TUTapp.

A utility function of the software package is used to configure a given TUT system during installation at a facility. The system is designed to be flexible enough such that track size and number of transmitter nodes per lap is configurable. The configuration software will determine the optimal placement of the transmitter nodes and allow the installer to override a node placement if an obstruction prevents placement in an optimal location. Each transmitter node is programmed by the TUTconfig utility with a unique identifier that it then broadcasts over two separate wireless systems. Once the install is complete, this data is permanently stored on the host. This data is uploaded to each WBT whenever it is issued to a user.

The hardware of the system consist of multiple copies of two core components, the WBT devices, worn on the wrist by each user, and transmitter nodes which provide a positional reference for the WBT. The WBT has a 96x64 pixel graphical OLED display to present session data. It reacts to the transmitter nodes placed around a track and computes average speed, a periodic pseudo-instantaneous speed, duration, distance, and prescription specific data.

The method used to track a user is shown in Figure 31. The WBTs determine speed and position by first detecting the edges of a transmitter node’s RF field and determining how long the field was present (how long the WBT was in the field). The RF lobe pattern is known and a pseudo-instantaneous speed is calculated. Secondly, from decoding the unique identifier in the transmitted field, it will determine which node's field was transited. Based on track configuration data downloaded from the host an average per-node speed and position are calculated, stored, and displayed for the user.

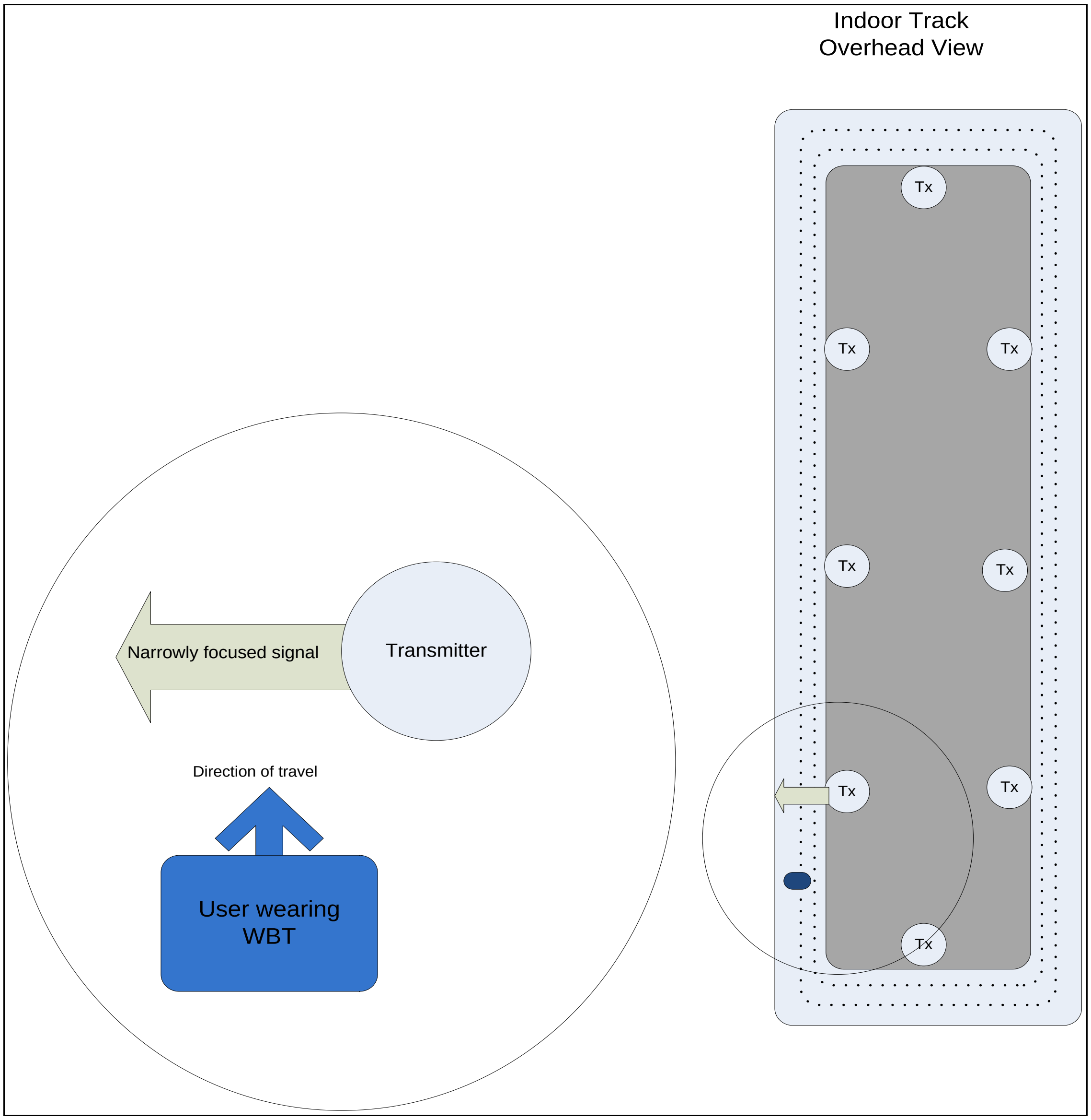

Figure 31-TUT tracking scenario.

It is undesirable to have the WBT present to much information to often. The monitoring of the WBT should be "at a glance". As designed, the user receives only periodic display updates and the relevant alerts after a user passes a transmitter node. At the end of a session, the user presses the stop button on the WBT. The unit is next connected to the host USB and an upload of session data occurs initiated by the host. The WBT automatically clears itself of session data and prompts the operator to press switch 1 when a new user is ready to download.

The transmitter nodes each emit the same low power RF signal with a unique identifier code embedded in the transmission. The WBT detects each signal as it passes near a transmitter and calculates speed distance data based on the stored distances between transmitter nodes and the time taken to pass between them. It calculates a pseudo-instantaneous velocity based on time spent in the known size of the transmitted RF field.

It is therefore necessary to tightly control the radiation pattern. To determine proximity to a centerline from a transmitter node there are two options, either sensing the peak signal strength or detecting the thresholds of the radiation pattern and extrapolating a centerline. If the WBT merely detects, for example, a localized wireless network, there is no easy way to discern how close to the source the WBT is.

The resolution of the system is dependent on how closely the WBT is able to discern where it is relative to any individual transmitter node's centerline with the track. As Figure 32 shows, the WBT uses the threshold of the pattern, and the pattern spread determines the accuracy of the system. If the pattern is tight enough, the threshold method becomes correspondingly more accurate.

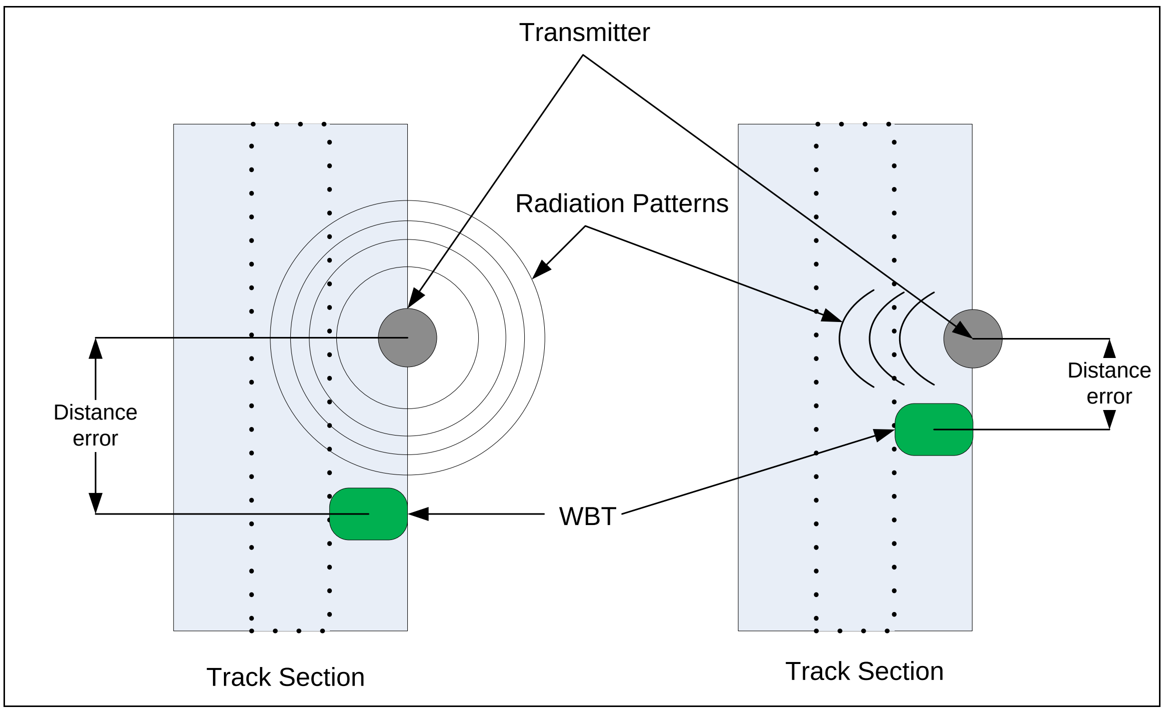

Figure 32-Detecting the threshold of different radiation patterns

The TUT is designed to limit exposure to RF. The WBT is worn on the body, users of the TUT can be assumed to be health conscious and the effects of RF on human tissue is somewhat in debate. The WBT emits no RF energy other than extremely small secondary receiver transmissions.

The procedure a user will follow once issued a WBT is straightforward. The user enters the track near a transmitter station, an indication will be presented to the user to begin, the user depresses a start button on the WBT and begins their session. As they progress around the track, the WBT updates itself at each transmitter node, monitoring and storing session data. The WBT displays a goal time to achieve for each lap segment in seconds and a time the user has spent transiting that segment.

5.2 Wrist Band Tracker (WBT)

The WBT is equivalent in size to a large watch in theory. It is intended to be worn comfortably on the wrist for an extended period. For determining position, the WBT is essentially an RF signal detector. It bases its position on signals received from each transmitter node it passes through. The major functions of a WBT are to record track session data, inform the user of speed and distance data and to update and present goal and or prescription related data (for example time remaining) to the user. It presents data to the user through a 96 x64 pixel RGB OLED display. User can access and enable or disable certain features through four touch sensitive switches located on the unit.

The block diagram of the WBT is shown in Figure 33. The controller of the WBT is the Texas Instruments MSP430F2274 microcontroller, which is coupled with a T.I. CC2500 2.4MHz transceiver. The WBT uses only the receive portion although future expansion of its capabilities may make use of the transmitter. The WBT has 1 Kbytes of RAM which stores track configuration and session data. 32k 16 bit bytes of flash RAM store the main control program as well as bitmap and font data.

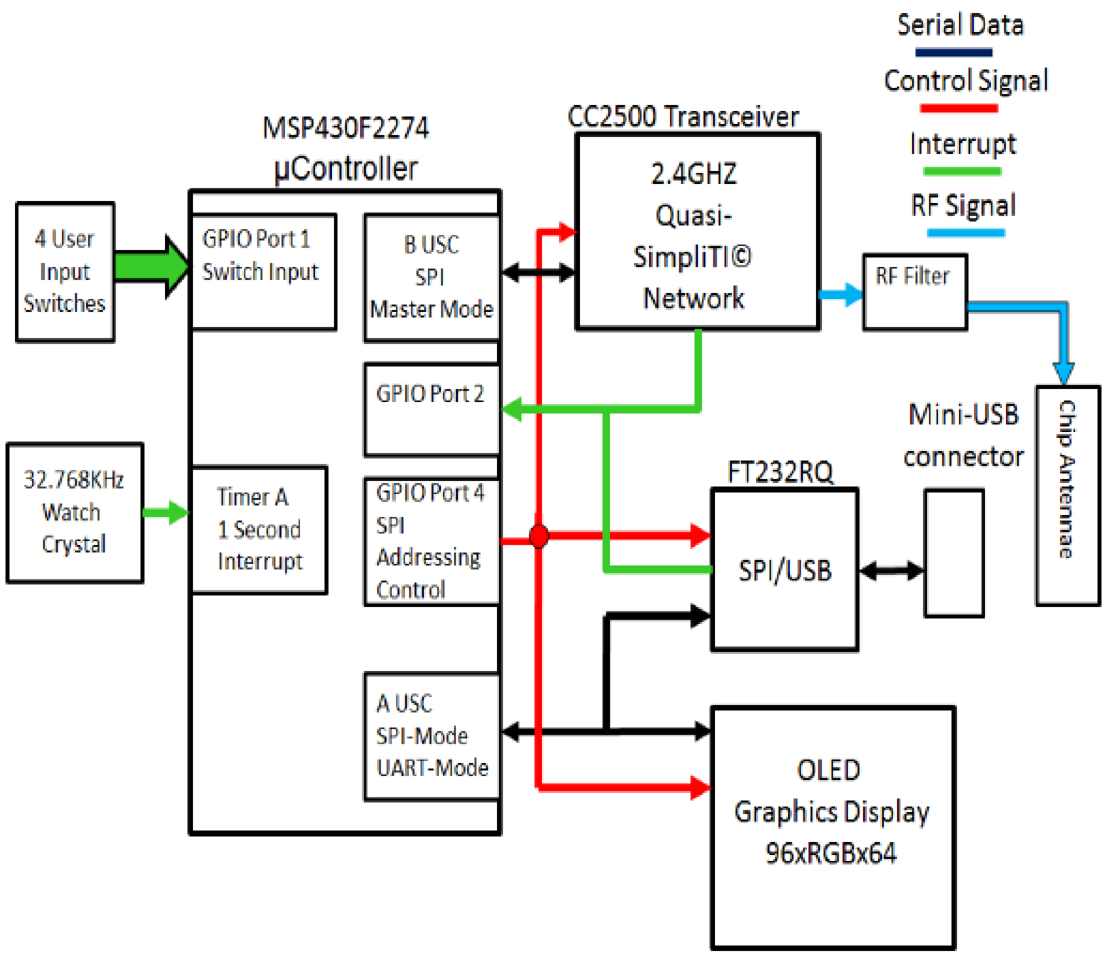

Figure 33-WBT Block Diagram.

5.3 TUT Transmitter Node

Each transmitter node shown in Figure 34 has a directional transmitter operating at 2.4GHz serving as a reference for a passing WBT. A separate transceiver section was designed but not implemented to accept and rebroadcast shutdown/wakeup commands from the TUT host computer. Each directional transmitter emits a unique code with which the WBT(s) determines its position on the track. The code is preprogrammed and does not change after the node is installed. The TUT transmitter node's placement on a track is setup in the host computer and this information is passed to WBT each time a user is downloaded.

The transmitter node peak power usage is during transmit. The power saving scheme involving a secondary wireless network would involve the host transmitting a shutdown to the first transmitter (physically closest to the host) which will then transmit a repeat shutdown prior to shutting down. In addition, the transmitters will operate at a duty cycle which will both minimize total power and ensure that a WBT passing at 23 MPH will not miss the radiation pattern thresholds. Currently the transmitter broadcast at a 75% duty cycle. The power saving scheme would be necessary to prevent any additional wiring at an installation. In other words, the transmitter will be battery operated and battery power conservation is important.

The transmitter software is almost entirely SimpliciTI© wireless networking software from Texas Instruments. Slight modifications were made to control radiated power and to control packet transmission timing.

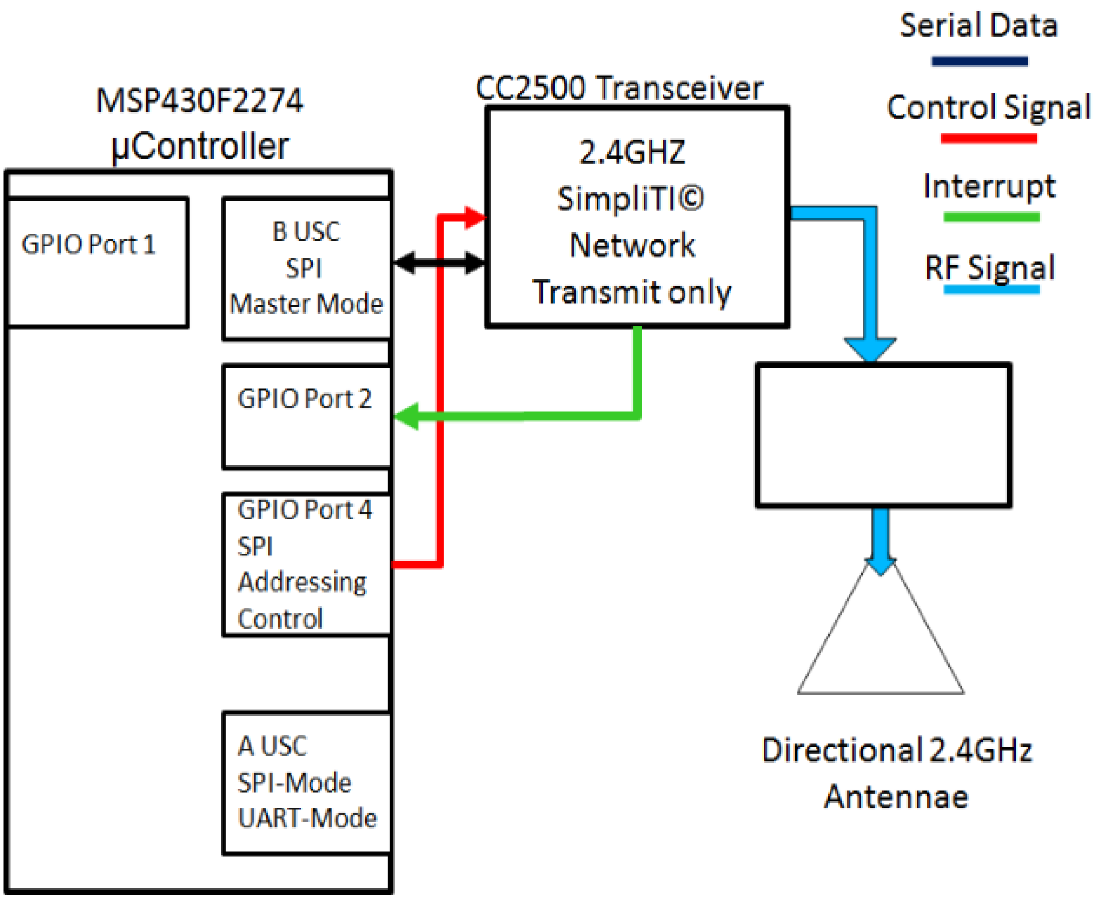

Figure 34-Transmitter Node Block Diagram.

5.4 TUT Software

5.4.1 TUT Utilities

A restricted utility program, TUTconfig, is accessible by means of the start=>run menus of the Windows XP operating system. A user name and password, “Installation”, “TUTmaster” will be available to qualified individuals. This utility is used at installation of the TUT at the track site and for maintenance. The utility’s GUI will, based on an input of “Track Size” and “Number of Transmitters”, calculate the optimal placement of transmitter nodes based on these two inputs. If obstructions in the building housing the track prevent optimal placement a secondary GUI function allows the installer to place nodes manually. The utility programs each transmit node with a unique identifier prior to its indicated installation position and a subset of this function allows for the replacement of nodes due to failures. The utility will write a file, TUTconfig.txt, to the host hard disk drive and a copy of this file will be kept at the site.

5.4.2 TUT Host Application

The main application software, TUTapp, interfaces with users through its login screen and to the TUT website via the internet. The application determines a user's status at login. Status is determined by querying the TUT website data base for an existing prescription, I.E. a Prescription Mode Patient (PM), an existing Self-Prescriber Mode (SM) user or the application will add an authorized (facility member, health club member) to the system. New users are assumed to be SM users and can either create an account or make use of the system for an individual session. The application interfaces to the WBT through a USB port and loads track information (including transmitter node identification coding and track configuration), the PM-SM user level field, and exercise session goal data. The application uploads exercise data from a completed track session and updates the user's personal exercise diary on the TUT website. A secondary function of the application will be to monitor and control the transmitter nodes power status and the number of WBT's currently in use and/or available for use. The application will shut off the transmitters under two conditions, the track facility is closed or no WBT's are issued. A shutdown packet is sent from the host to any receiving transmitter node. Receiving nodes then rebroadcast the shutdown signal, transmit a "I am shutting down" signal, wait for ten seconds during which time each node monitors and rebroadcasts any "I am shutting down" packet received from any other node. The application monitors and verifies that is has received the shutdown acknowledge from each node. The auxiliary shutdown system has not been incorporated into the transmitter design at this time and was not part of the prototype system.

5.4.3 TUT Website

The TUT website serves the primary function of health care provider to patient interface. The website will have three access levels, health care professional, PM user and SM user. Health care professional with patients using the system will have access to those patient's exercise logs. They will use the website to write and update exercise regimen prescriptions as needed and to interface with their patients, receiving and giving advice and consents.

The TUT website serves hosts the user database containing the exercise diaries of all users. Data base references to a user is coded by a patients number for both SM and PM users. The TUT website login protocols and database are secure. The information stored in the database is not necessarily medical records however some information pertaining to, for example, a patients restriction on the amount of time spent walking, will be securely kept and the use of patient numbers, as opposed to names will prevent any misuse of the website by, for example, health insurance companies.

6 Design Implementation Strategy

6.1 Hardware Specific Prototypes

A development kit was purchased from Texas Instruments and hardware and WBT, transmitter software development will take place simultaneously. The development kit contains two of the boards shown in Figure 36. One of these was ultimately used as the WBT receiver section. The design was broken into sections and each section used a piece of the development hardware. Sections were hardwired together to form a prototype of first WBT. Next, a WBT, including the display was mocked up to prove most, if not all, of the hardware design.

The development kit includes a limited Code Composer license and development tools which will be used for the software development of both the transmitter node and WBT. It includes 2 of the boards shown in Figure 35, two antennas, battery packs and all necessary hardware to prototype a wireless network.

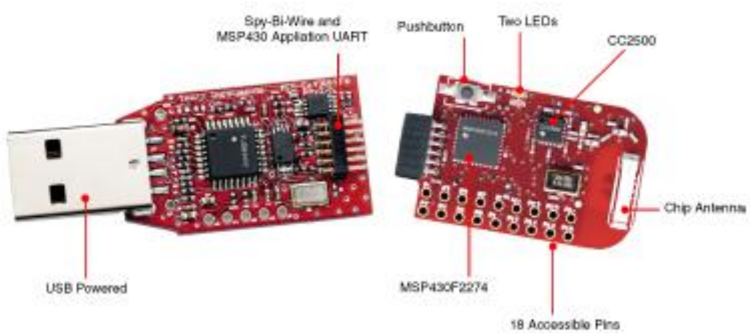

Figure 35-eZ430-RF200 Development boards (with permission Texas Instruments)

This board will be operating in the 2.4GHz region and will also be used to prove the transmitter node characteristics and secondly to prove the host to transmitter node wireless network. Code developed during this proving stage will be directly applicable to the final prototype. One of the TI development boards will be hardwired to the OLED display for proof of concept and to develop the software functions to drive the display. The OLED pins are fully accessible and the I/O pins needed to implement the designed microcontroller interface are available on the TI board.

A USB development Kit from Future Technology Devices International (FTDI), MM232R, was incorporated into the WBT mockup. The USB portion of the WBT design was proven, host drivers tested, and TUTconfig and TUTapp interfaces tested. The other one of the Texas Instruments development boards was designated for the transmitter node development and antennae testing.

All knowledge gained during the preliminary mockups was simultaneously transferred to the ongoing final prototype design and PCB layout. The WBT design required more time for software and hardware development and was ongoing as an attempt at fabricating the transmitter PCB. The completed transmitter node was used to analyze the antenna purchased for the directional RF. Additionally and apace of this research the WBT mockup and the WBT PCB was tested.

Testing of host software and utilities will be on going during the entire process with priorities given to those modules needed during hardware development. Section 8.3, Milestone Chart, gives the planned software development and hardware dates.

Part numbers and development costs are included in section

Tool Purpose	Description & Part Number	Vendor	Costs	Qty
WBT Prototyping Code development	Texas Instruments MSP-FET430U38: MSP430 38-Pin Target board and USB Programmer	Texas Instruments	$169.00	1
RF Development	EZ430-RF2500: MSP430 Wireless Development Tool	Digikey 813-1026-ND	$59.00	1
		Total Costs	$228.00	

Table 9-Development BOM.

6.2 Host Software Development

Before designing the software systems required for the operation of the TUT system, we needed to select the appropriate software development model which best supports our teams goals and level of interaction. There were many different software development models to pick from each having their own primary focus, required steps and milestones, and interactions with the customer. All of these considerations were important to insure that the software design process chosen was structured in a way that we could correctly apply each development phase to our overall project requirements. We decided that by selecting the proper model we would be more successful in producing a software design that fulfills all operational and functional requirements of the

TUT system. For the purposes of the software development side of our project we have designated Mike Tullbane as the customer since he is the one who initiated and outlined the design of the TUT system This makes it easier to fully define the software lifecycle into a manageable and traceable process using the chosen software lifecycle model. It also gives us a single point of contact for requirement identification and definition while moving forward in the design process.

In order to select the proper model we reviewed notes from a software engineering class attended by our computer engineer and began research on the internet so we could have a greater variety of opinions to use in our decision. We found that most of the internet searches we performed on the same subject agreed with Pfleeger and Atlee, so we based our conclusion on their work (5). We ended up with three possible options to consider that could work well for our project. The first model we considered was the traditional model called the “Waterfall Model”. This model is very straightforward and has the structure that we wanted for our project software. It does, however, have a significant drawback in that it is inflexible to changing requirements. Since we are designing a prototype system here, we decided against the waterfall model based on that feature. The next model to consider is called “Agile Methods”. This model is very flexible to changing requirements, but lacks the structure and documentation of the waterfall model. This model also requires more interaction with the customer and aims to quickly produce a final product. We rejected this model because we needed less interaction with the customer so he has time to work on his own part of the hardware design and more structure to keep us focused and organized. The last option we considered is called the “Prototyping Model”. We concluded that this model would work best for our goals and timeline. This model allows us to produce prototypes at different stages in the product development cycle to verify requirements and functionality, and to be able to make revisions as necessary throughout the process thereby ensuring that we properly implement all required functions in our final product. Figure 36 illustrates the “Prototyping” software lifecycle model.

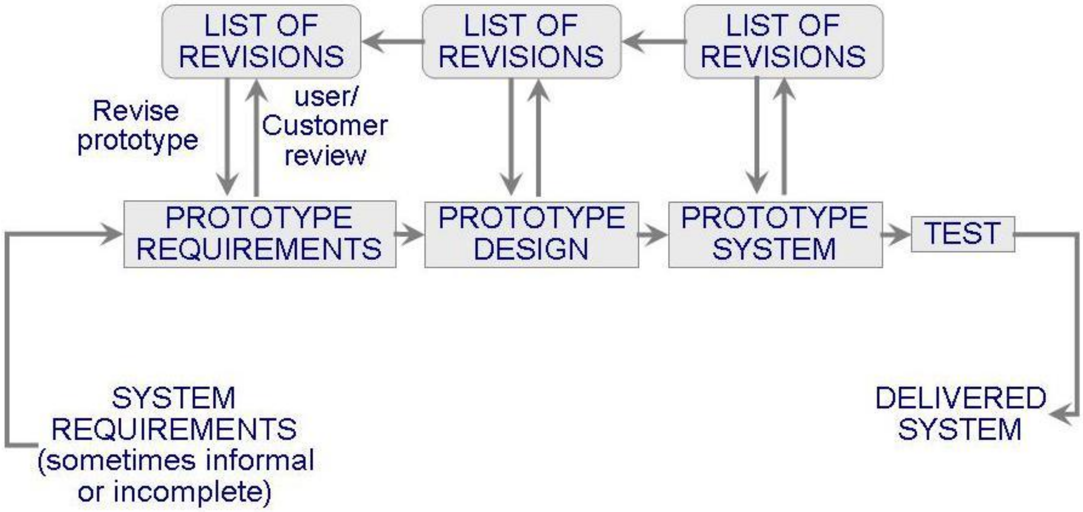

Figure 36 - Prototyping Model (With Permission Dr. Damla Turgut, UCF, CECS)

As you can see, this model gives ample opportunity to make revisions so we can be sure that all requirements are implemented prior to the start of the testing phase.

Now that we have chosen the software lifecycle model to follow, we can begin the analysis of the overall system and start to capture the system the software requirements. We began this phase by looking at the high level view of the system using direct communication with the customer. While discussing the overall system with the customer, we identified at least three major software subsystems for this project. These three subsystems will operate independently to provide the full set of capabilities for our system. The first subsystem we identified is the TUT host system. This is the part of the system which performs the initial configuration of the transmitter / receivers and the wristband tracker devices, and will provide for all user interaction with the TUT system at the gym. This subsystem also provides secure access to the users exercise diary or patient physical therapy record. The next subsystem is be the TUT website. The website allows secure remote storage of user and patient data, easy secure retrieval of exercise records from any location with an internet connection, and simple efficient delivery of prescriptions to physical therapy patients. The last subsystem identified is the patient and user records database. This subsystem provides the data storage and retrieval component of the TUT system. It will allow physicians and patients or users to retrieve past and current exercise data as well as maintain the data in a secure manner restricting access to authorized data requests only.

With these three subsystems identified, we were able to define the basic high level architecture of the software system. The only possibility that resembles the interaction between software components of our system is the basic client-server architecture. In this configuration, we have a central database server and a web server with all other access points as clients. This configuration is shown in Figure 29, section 5.1. In this illustration, the component labeled “Internet” functions as both the web server and the database server. The component labeled “Patient at Physical Therapy” represents the TUT host subsystem. All other components are user, patient, or physician computers used to access the TUT system.

All of the software development tools were chosen to be compatible with a Windows environment. The operating system installed on the development machine is the thirty-two bit version of Microsoft Windows 7 Professional. We developed the TUT host software subsystem using the Eclipse Enterprise Edition IDE version Indigo with the Jigloo GUI builder plug-in. Eclipse was chosen due to the superior support for rapid GUI development and integrated web development features. This IDE supports every aspect of our TUT host software subsystem including an integrated testing environment, GUI previews, and automatic creation of Java code from the GUI model. Eclipse also supports database connections and is able to test our TUT host database connections.

For the database development we used MySQL Workbench v5.2.35 community edition with MySQL Community Server v5.5.18. MySQL Workbench is a graphical development front end for modeling, creating, and administering a database using the MySQL database software. Using MySQL workbench significantly reduced development time by allowing us to create the database model graphically, generate the SQL code, and then run the SQL code to implement the database design. The administration features of MySQL Workbench provided us with the tools needed to populate the database with test data and run actual queries on the data for verification.

Web server development was done in a couple of stages using the Information Interchange Service version 7.5 from Microsoft. First, we added a new site and specified it’s function to implement our server. Then we designed the welcome page with required hyperlinks to our application. This type of web server is considered a static sever since it only serves HTML documents. The link in the HTML document then uses a file with the .jnlp extension to start our application. JNLP stands for Java Network Launch Protocol which is required for Java Web Start. Java web start actually downloads the compressed java byte code file for the application to the client computer and runs it in a secure shell. This method significantly reduced our coding and development time for our web site. After that we only had to use a free domain name service to register our site and it was up and running in one day.

7 TUT Testing Strategy

7.1 Test Environment

7.1.1 Description of Test Environment

All components of TUT are designed for the environment housing the indoor track. These environments are kept at relatively low ambient temperatures and humidity levels. Typically the ambient temperature is kept between 75 and 80 degrees Fahrenheit (24˚ to 27˚ C) with a relatively humidity of 60%.

7.2 WBT Standalone Test Procedures

7.2.1 User Interface

A PM user will be loaded into to the WBT with a prescription of 1 mile in 1 hour. The WBT will prompt the user to configure the unit through the 4 input switches. Table 6 lists the WBT prompt, the required action, and results.

User Input Tests			
WBT Prompt (in quotes)	Expected Result/ User Input	Switch number	P/F
"Press Start To Begin" A red X appears on the OLED corner near SW1.	Presses SW1. WBT switches into track mode. Tracking data appears on the OLED.	1	
OLED displays any item	Press SW2 OLED panel turns off/on	2	
OLED displays any item	Press SW3 multiple times Contrast is adjusted down	3	
OLED displays any item	Press SW4 multiple times Contrast is adjusted up	4	

Table 6-WBT User Input Tests PM mode

7.2.2 **Accuracy Validation**

7.2.2.1 Transmitter node detection

The WBT will be loaded with test code that displays field strength detection data, incoming identification of the transmitter node, and distance to the node. A measurement box will be placed on the floor in a suitable environment in front of and perpendicular to the transmitter. 3 Physical measurements between the WBT and a line perpendicular to the transmitter centerline will be taken and compared to the WBT display. Each measurement and the corresponding WBT displayed distance will be recorded in Table 7. The difference between measured and displayed distance shall not exceed 1 foot.

WBT Accuracy		
Action Required and/or Expected Result	Measured dbm	WBT Measured Distance
OLED panel displays: The dbm bar graph		
The dbm bar graph		
The dbm bar graph		
Expected Result: Measured versus WBT displayed measurements shall not exceed 1.5 feet		

Table 7-WBT Spatial Accuracy Test Table

7.2.3 **Display Testing**

Test code will be loaded to facilitate this test. Button 1 on the WBT will be used to cycle through all allowable characters and graphics. Button 2 will be used to set the mode of testing and cycle through the data only, normal and sleazy normal display modes. Each available graphic, encouragement message or data value will be cycled through.

7.3 Transmitter Test Procedures

7.3.1 **DIRECTIONAL ANTENNA TEST PROCEDURES**

The directional antenna and the transmitter node have been tested to make sure all the parts work properly before any user can get on the track. The test environment was the Senior Design lab located in engineering 1. The temperature of the lab is between 25 degrees and 30 degrees Celsius. Friis' transmission equation has been used to predict the power delivered by the directional antenna.

$$P_r = P_t * G_r * G_t * [[\frac{\lambda}{4*\Pi*R}]^2]$$

The values of the parameters of Friis' equation are as follow: P_t = 200W, Gt = 10db, G_r = 0.5 db, R = 1 meter and $\lambda = \frac{3*10^8}{2.4*10^9}$ = 12.5 cm. after changing all the values in linear for calculation purposes, the power delivered by the directional antenna is 0.22 W. the difference between the transmit power, and the received power is due to the path loss during signal transmission. The power delivered has been calculated for R = 0.25 meter, 0.5 meter, and 1 meter. For R = 0.25 meter, Pr = 3.55 W, and for R = 0.5 meter, Pr = 0.89 W. figure 29 below describes how the relation between the range and the received power. We can notice that when the range is increasing, the power received decreased.

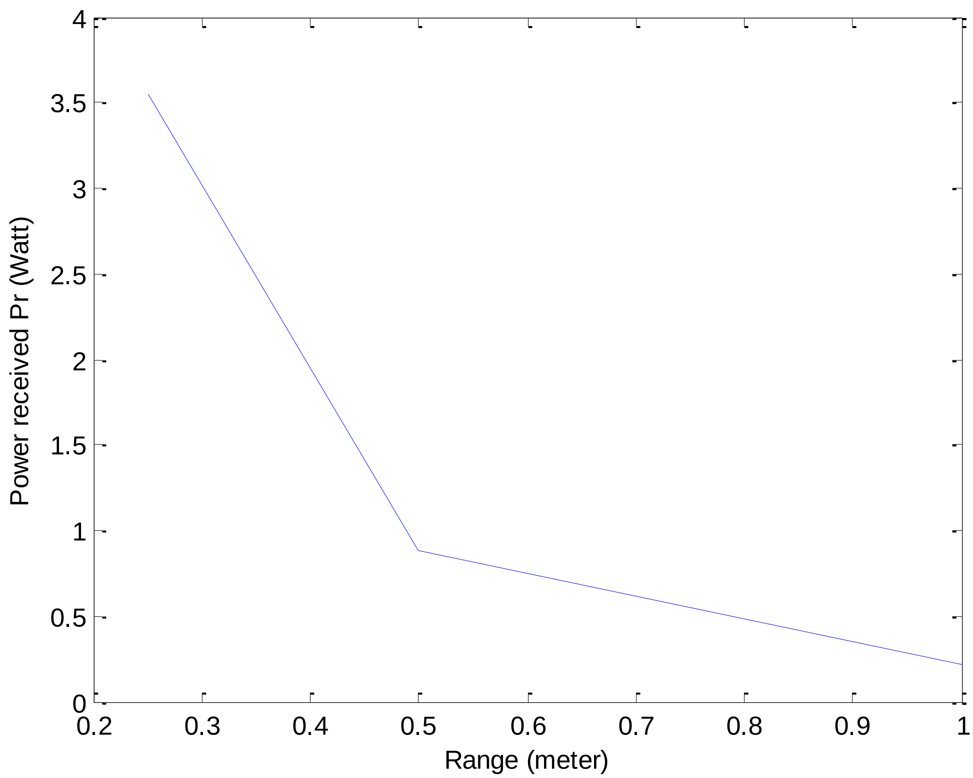

Figure 37: Power received in function of the range

7.3.2 Transmitter Node General Testing

The directional antenna has been tested with the WBT. The directional antenna has been placed at a fixed location, and the WBT has been placed the first time at 10 inches away from the directional at zero degree angles. The result of the power received is the table below. The power delivered has also been measured at 45 degrees and negative 45 degrees when the WBT is still at 10 inches away from the directional antenna. We also measured the power delivered by the directional antenna at 18 inches, 26 inches, and 32 inches respectively. At each location, measurements have been taken at 45 and negative 45 degrees. The testing of the antenna gave us an idea about the real radiation pattern of the antenna. We also learnt how far the transmitter can broadcast signals to the WBT.

Distance away from the directional antenna							
10 inches		18 inches		26 inches		32 inches	
Range (feet)	Power received(μW)	Range (feet)	Power received(μW)	Range (feet)	Power received (μW)	Range (feet)	Power received(μW)
-12	0.282	-12	0.794	-12	1.259	-12	1.259
0	0.251	0	4.467	0	3.162	0	1
12	1.259	12	0.282	12	0.501	12	0.2511

Table 8: Range testing of the Directional antenna

The functionality of the transmitter node has been tested with the WBT. The transmitter node was attached at a fixed position with the directional antenna in the lab. The WBT is also placed at a fixed position on a table. The antenna was moved closer to WBT to simulate the runner reaching each transmitter node. Each time the WBT will display the speed at which the runner was going. At the end of the section, the WBT will be place via a USB to the host to upload the information of the user. In formation means, the speed, the goal, and what the user had done on the track.

7.4 Host Software Test Environment

7.4.1 Description of Test Environment

The TUT software subsystems was tested on the development computer. This system is windows based system with the thirty-two bit version of Windows 7 professional installed as the operating system. All software tests that did not require the WBT were run on this system The hardware of the test system consists of an AMD Athlon II X4 620 processor running at 2.6 giga-hertz (Ghz) with four giga-bytes (GB) of random access memory (RAM) running at 400 MHz. Our secondary storage has two one terra-byte (TB) hard disks using a Serial Advanced Technology Attachment version 2.0 (SATA2) interface capable of transferring data at the rate of 3 giga-bits per second (Gb/s). The disks are arranged as a mode one redundant array of independent disks (RAID1) which provides complete data redundancy between disks to prevent data loss from a single point disk failure. Internet connection is provided by a Realtek network interface which can transfer data at a rate of up to one (Gb/s). The TUT host software, TUT web server software, and the TUT database software will normally be running in a similar environment. We cannot guarantee that the production hardware environment will be exactly the same as the test hardware environment, but we can say that TUT software will run within the environment described in the resource requirements section. The initial (low level) software unit testing was performed by the programmer. The high level unit testing as well as the integration testing were performed by the programmer and the TUT hardware designers. For the integration testing we ran the TUT host on a laptop computer running the windows Vista Business 32 bit operating system while the TUT web server and the TUT database were be running on the previously described test system with the database populated with test data.

7.5 Host Software Test Procedures

7.5.1 Introduction

The software testing is a major portion of the software development life cycle. There are test procedures for each major subsystem and for each class in each subsystem. Our testing efforts verified proper communications between classes, correct data formats, and correct results from any computations. Once all data details were verified we tested the user interfaces, and the interfaces between software subsystems, for proper function and timing. When we finished the software integration testing we deemed our TUT software system prototype ready to participate in the hardware integration testing. There were tests performed in the software integration test phase that required some hardware support in order to test for proper software reaction to a hardware event. For the most part these tests were focused on the software subsystems.

7.5.2 Stopping Criteria

During integration testing, with the assistance of the hardware development team, we identified any remaining deficiencies using our predefined functional and interface requirements. Each requirement was tested as defined and an electronic stop-watch was used for response time verification. Some of our times were in the milliseconds so our accuracy was limited but all response times were faster than specified. The hardware team had a checklist to keep track of tests performed, tests passed, and tests failed. There are also columns on the checklist for priority of faults which will determine when and how the fault is addressed. If the fault, or failed test, was determined to be correctable within 5 minutes the programmer would make the proper corrections and restart testing at the point before the failure. If the fault has an undetermined correction time required and it was not a fatal error, then testing continued and the fault was documented for correction after the testing session was completed. If the fault was ranked as fatal it means that either the system crashed or the system functionality was diminished to a point where testing could not continue. In this case the testing was stopped and the programmer rescheduled another testing session after all identified faults were corrected. When the team finished a testing session without any documented faults, then the system software was deemed deliverable.

7.5.3 **Description of Test Cases**

Before we describe the test cases in detail we will explain the roles of the test team members and show the checklist that was used during all testing sessions. Our objective was to only have one testing session to validate our system, but that was an unrealistic goal considering the number hardware and software subsystems involved. What we planned was to perform comprehensive integration testing without overlooking any test failures even if it was just a slower than required response time. All failures were to be documented for correction and any changes and / or enhancements were to be documented for possible implementation. All failures would be prioritized as to whether they halted the system, returned the incorrect data, or degraded system performance. Failures would take priority over enhancements but every effort would be made to accommodate enhancements prior to release of the TUT prototype. Once documented failures were corrected, the team would start another testing session. We planned to test in this manner until there were no failures and all team members agreed that the system was ready for prototype demonstration. We were not able to perform our testing as planned. We did test every part of the system before System Demonstration, but due to other delay factors we could not dedicate the time required by the team to perform all formal testing that we planned. Figure 38 below shows the checklist that was to be used by our test team. This checklist was to be used for every testing session and then filed as part of our quality assurance record. The test team members were to have a printed copy of Table 1 and Table 2, section 2.3.3 for cross reference to requirement descriptions.

Name:			Session #:	Date:	
Req. #	**W**itnessed or **P**reformed	**P**assed or **F**ailed	Priority (1,2,3)	Category (a,b,c)	Failure Mode / Comments
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
Additional Comments:					
Categories: a -> functional b -> interface c -> enhancement / change			Priorities: 1 -> system halted 2 -> incorrect data 3 -> system reacted slowly		

Figure 38-Software Test Procedures Check List

Test case #1: This test will verify that the method used to display the user GUI is properly and efficiently coded. It will also verify that the system displays the correct GUI for the user ID number and password entered during log-in. The system will detect the user type by reading the user ID number and password. The test team member will enter five incorrect user ID numbers to verify that the system properly prompts the user to “please enter a valid user ID number”. The tester will then enter the correct user ID number in the login text box followed by pressing <Enter>. The system will then present another blank text box with the “Password:” prompt next to it. The test team member will enter five incorrect passwords to verify that the system properly prompts the user to “please enter password for user number XXXXXXX”. The tester will then enter their correct password followed by pressing <Enter>. The tester will use the stop-watch to verify that the system will present the user with their proper GUI screen within three seconds of pressing <Enter>. The test team will perform this same test for all three user types to verify that the system displays the proper user options for the user type.

Test case #2: This test will verify that the method used to accept the entry of user goals in the non-prescription user GUI is properly and efficiently coded. It will also verify that the system correctly displays the “Accepted” prompt within 1 second after the user enters goal data and then presses <Enter>. The test team member will enter five different numbers for a distance goal, followed by pressing <Enter>, and use the stop-watch to verify that the “Accepted” prompt displays within one second. The test team member will enter five different numbers for a time goal, followed by pressing <Enter>, and use the stop-watch to verify that the “Accepted” prompt displays within one second. The test team member will enter alpha-numeric data in the distance goal input box, press <Enter>, and verify that the system properly prompts the user to “please enter integer values only”. The test team member will enter alpha-numeric data in the time goal input box, press <Enter>, and verify that the system properly prompts the user to “please enter integer values only”.

Test case #3: This test will verify that the method used to assign a WBT is properly and efficiently coded. It will also verify that the system displays the correct prompt and issues a WBT to the user within 5 seconds of clicking the “I’m Ready to Exercise!” button on the user GUI. The test team member will login five different times using five different user ID’s and passwords, retrieve a test prescription, click the “I’m Ready to Exercise!” button on the user GUI. The tester will then use the stopwatch to verify that the system assigns a WBT and displays the WBT number within 5 seconds. The above test will be performed for each user type.

Test case #4: This test will verify that the method used to monitor WBT’s in the USB dock is properly and efficiently coded. It will also verify that the system displays the correct prompt when the user attempts to remove a WBT that was not yet assigned. This is in place to insure that when the system assigns a WBT to a user, the user removes only the assigned WBT from the dock. The test team member will login five different times using five different user ID's and passwords and they will attempt to remove the wrong WBT three times before removing the correct WBT. The system shall properly prompt the user every time the user attempts to remove the wrong WBT and the system shall still correctly allow the user to remove the correct WBT.

Test case #5: This test will verify that the method used to monitor user activity on the system is properly and efficiently coded. It will also verify that the system displays the correct user prompt when there has been two minutes of inactivity on the system with a user still logged in. The system will detect that there is a user logged into the system and that there has been no activity for two minutes and then present the "System Will Log You off in 10, 9, 8, … seconds" prompt. If there is no user response the system will log off the user after the ten second count down. The system will allow the user to log back on and will not lock the user out. The test team member will log in and wait and use the stop watch to verify that the system begins the log off countdown and presents the proper prompt when there has been two minutes of inactivity. The tester will then press any button on the keyboard or move the mouse on the screen or click one of the mouse buttons to verify that the countdown timer and prompt disappear. The tester will again wait and use the stopwatch to very that the system again begins the logoff countdown and displays the proper prompt after two minutes of inactivity. This time the tester will allow the system to log him / her out and then the tester will log back into the system. This will verify that the auto logout does not affect normal system operation. The above tests will be performed for each user type.

Test case #6: This test will verify that the method used to monitor the return of an assigned WBT is properly and efficiently coded. It will also verify that the system correctly displays the password prompt within five seconds after an assigned WBT is returned to the USB dock. After the user enters the correct password, the system will display the proper GUI based on the user type. The test team member will return an assigned WBT to the USB dock and use the stopwatch to verify that the system displays the password prompt within five. The tester will then enter five incorrect passwords to verify that the system properly prompts the user to "Please Enter Correct Password". The tester will then enter the correct password in the login text box followed by pressing <Enter>. The system will then present the user with the correct GUI based on their user type. The above tests will be performed for each user type.

Test case #7: This test will verify that the method used to email the exercise log to the user is properly and efficiently coded. It will also verify that the system properly requests a valid email address if one is not currently present in the database and then confirm the address by clicking on the "OK" button next to the email address display box. If the required email address is present in the database, the system will display the email address and prompt the user to confirm the address by clicking on the “OK” button next to the email address display box. Once the user confirms the email address, the system shall display the “Message Sent” prompt within one minute. The test team member will log in under five different user ID numbers and passwords and click on the “Email Exercise Log” button on the GUI. The system will then either display the stored email address for that user or display a text input box with a prompt for the user to enter a valid email address. The tester would then either verify the displayed email address, or enter a new email address and then click on the “OK” button next to eh email address input box. The test team member will now use the stopwatch to verify that the system responds with the “Message Sent” prompt within one minute after clicking “OK”. The test will also verify that the system returns to the user GUI screen after the “Message Sent” prompt appears for two seconds. The above tests will be performed for each user type.

Test case #8: This test will verify that the method used to display the exercise log to the user is properly and efficiently coded. It will also verify that the system properly retrieves previous exercise data that is available from the database. If there is historical data present in the database for the current user the system will prompt the user for the date range of the data to display. After the user clicks on the “Display Exercise Log” button on the GUI the system will either display the results downloaded from the WBT or display the historical data GUI screen to prompt the user for a date range to display. If the system displays information downloaded from the WBT, there will be no significant delay. After the user enters a valid date range and clicks on the “OK” button, the system shall display the selected data within ten seconds. The test team member will log in under five different user ID numbers and passwords and click on the “Display Exercise Log” button on the GUI. The system will then either display the data just downloaded from the WBT or present the user with the historical data GUI screen. The test team member will then enter five different invalid date ranges to verify that the system correctly recognizes and invalid date range and properly prompts the user. The tester would then enter a valid date range and use the stopwatch to verify that the system correctly displays the requested data within ten seconds. The test team member will perform five requests for data that was downloaded from the WBT and five requests for data from the database and verify that all data displayed is correct and it is properly displayed. The above tests will be performed for each user type.

Test case #9: This test will verify that the method used to log the user off the TUT system is properly and efficiently coded. It will also verify that the system properly displays the logoff message “See You Next Time”. The system will then display the login screen and prompt for user ID number. The test team member will log into the TUT system five times for each user type and then verify that the system properly logs the user off the system within one second after clicking on the “I’m Done for Today” button.

Test case #10: This test will verify that the wake up section of the method used to monitor the status of transmitter / receivers is properly and efficiently coded. It will also verify that the USB wireless device is operating within the correct frequency and reaching the required distance when the wake up command is sent. The system will detect that the first WBT is issued for use and then take the proper action to transmit the “Wake-up” signal to the Transmitter / Receivers. The test team member will log-into the TUT system and properly begin an exercise session. Another team member will be standing by at the Transmitter / Receiver nearest to the TUT host. When the team member who logged-into the system removes the assigned WBT from the USB dock, that tester will inform the team member who is standing by at the Transmitter / Receiver that the WBT has been removed from the USB dock. The test team member at the Transmitter / Receiver will then use the stopwatch to verify that the unit wakes up and begins transmitting within five seconds. The tester at the USB host will verify that host receives the signal back from the transmitters that they are now transmitting. The above test will be performed for each user type to verify that the system manages the transmitters correctly independent of user login.

Test case #11: This test will verify that the sleep section of the method used to monitor the status of transmitter / receivers that is based on the number of WBT’s in use is properly and efficiently coded. It will also verify that the USB wireless device is operating within the correct frequency and reaching the required distance when the sleep command is sent. The system will detect that the last WBT in use is returned to the USB dock and then take the proper action to transmit the “Sleep” signal to the Transmitter / Receivers. The test team member who was logged-in to the TUT system in test case #10 above will return the active WBT to the USB dock. Another team member will be standing by at the Transmitter / Receiver nearest to the TUT host. When the team member who logged-into the system returns the active WBT to the USB dock, that tester will inform the team member who is standing by at the Transmitter / Receiver that the WBT has been returned to the dock. The test team member at the Transmitter / Receiver will then use the stopwatch to verify that the unit powers down and enters sleep mode within five seconds. The tester at the USB host will verify that host receives the signal back from the transmitters that they are in sleep mode. The above test will be performed for each user type to verify that the system manages the transmitters correctly independent of user login.

Test case #12: This test will verify that the sleep section of the method used to monitor the status of transmitter / receivers that is based on the operating hours of the gym is properly and efficiently coded. It will also verify that the system time process in the TUT host computer operating system is functioning properly and that the TUT host software properly reads the system time parameter. The system will detect that the preset closing time of the gym has been reached and then will take the proper action to transmit the “Sleep” signal to the Transmitter /

Receivers. This test will be performed with no user logged-in to the system. One test team member will monitor the system time on the TUT host computer and another team member will be standing by at the Transmitter / Receiver nearest to the TUT host. The team member at the host will inform the team member at the Transmitter / Receiver that the system time is now equal to the preset closing time of the gym (programmer will simulate this parameter by adjusting the system time to match closing time). The test team member at the Transmitter / Receiver will then use the stopwatch to verify that the unit powers down and enters sleep mode within ten seconds. The tester at the USB host will verify that host receives the signal back from the transmitters that they are in sleep mode.

Test case #13: This test will verify that the sleep section of the method used to monitor the status of transmitter / receivers that is based on the operating hours of the gym and active WBT's is properly and efficiently coded. The system will detect that the preset closing time of the gym has been reached and that there is still at least one active WBT in use. The system will then wait for 15 minutes before initiating Transmitter / Receiver sleep mode. If the WBT has still not been returned to the USB dock, then the system will take the proper action to transmit the "Sleep" signal to the Transmitter / Receivers. One test team member log in to the system and check out a WBT. The tester will then monitor the system time on the TUT host computer while another team member stands by at the Transmitter / Receiver nearest to the TUT host. The team member at the host will inform the team member at the Transmitter / Receiver that the system time is now equal to the preset closing time of the gym (programmer will simulate this parameter by adjusting the system time to match closing time). The test team member at the Transmitter / Receiver will then use the stopwatch to verify that the unit does not power down and enter sleep mode until fifteen minutes later. The tester at the USB host will verify that host receives the signal back from the transmitters that they are in sleep mode. The above test will be performed for each user type to verify that the system manages the transmitters correctly independent of user login.

Test case #14: This test will verify that the user identification number section of the method used to display the log-in GUI is properly and efficiently coded. It will also verify that the system correctly accesses the database to display a drop down combination box from which to select the user identification number for log-in. The test team member will use a predefined list of test ID numbers to verify that the system correctly displays all identification numbers in the database when the drop down list is selected.

Test case #15: This test will verify that the password section of the method used to display the log-in GUI is properly and efficiently coded. It will also verify that the system correctly properly recognizes whether the user has used the system previously or the user is logging in for the first time. If the user has used the system previously, then the system will just allow the user to enter their current password and log in. If the user is a “New” user, the system will allow the user to enter a default password (“Exercise”) to login for the first time. The system will then prompt the user to create a personal password to use for future log-ins. The test team member will login and out five times with their personal (test) password to verify that the system properly allows their log-in. The tester will then log off of the system and then log in with a user identification that does not yet exist in the database to verify that the system correctly recognizes the user ID as a new user and properly prompts the user to set their password. The test team member will enter a password, click on the “OK” button next to the password input box, and the system will display the prompt “Password Set”. The tester will verify that the system has returned to the login GUI to allow the user to log in with their personal password. The test team member will then select their user identification number form the drop down list and enter their new password when prompted. The system shall now recognize the user as a registered SM user and display the correct user GUI screen.

Test case #16: This test will verify that the method used to allow an SM user to enter time and distance goal data is properly and efficiently coded. It will also verify that when a user enters goal data that the data is properly recorded and that the system correctly displays the “Goal Set” prompt after the user presses <Enter>. The test team member will enter five different numbers in the time goal input box followed by pressing <Enter>. The test team member will verify that the proper prompt is displayed and the programmer will verify that the data is correctly stored in the database. The tester will then perform the same sequence in the distance goal input box with the programmer again verifying that the system correctly stores the data in the database. The test team member will click on the “I’m Ready to Exercise!” button and the WBT designer will verify that the proper data is uploaded to the WBT.

Test case #17: This test will verify that the method used to retrieve a prescription for a PM user is properly and efficiently coded. It will also verify that the system correctly retrieves the prescription from the database and uploads the data to the WBT. The system will detect the user type by reading the user ID number and password and present the PM user with the proper GUI. The test team member will enter a correct user ID number and password for a PM user. The system will then properly display the PM user GUI. Then tester will click on the “Retrieve Prescription” button and the system will retrieve the prescription data from the database and upload the prescription to the WBT. The WBT designer will verify that the prescription was properly uploaded to the WBT. The test team will perform this test for five different PM users.

Test case #18: This test will verify that the method used to select whether to email or display exercise log data is properly and efficiently coded. It will also verify that the system properly displays the “Enter Preferred Email Address” prompt after selecting the “Email Exercise Log” option from the GUI. The team member will log in to the TUT system with a user ID and password for a user with a previously stored email address in the database. The tester will then click on the “Email Exercise Log” button and the system will respond with the message “Email Sent”. The programmer will then verify that the email address used was the correct email address for the current user. The test team member will now log out of the system and log back in with a user identification number and password of a user who has never entered an email address into the system. The tester will then click on the “Email Exercise Log” button and the system will respond with the message “Enter Preferred Email Address”. The tester will enter a valid test email address and the system will respond with the “Email Sent” message. The programmer will verify that the email address entered is properly stored in the database for the current user ID. The above test will be performed for five different user ID numbers with and without currently stored email addresses.

Test case #19: This test will verify that the method used to check in a returned WBT is properly and efficiently coded. It will also verify that the system correctly displays the prompt “Enter Password” when a WBT is returned to the USB dock after an exercise session. The system will present the correct user GUI for the current user type. The test team member will log in to the system as a PM user and check out a WBT. The tester will then wait two minutes and return the WBT to the USB dock. The system will then prompt the user for a password and the test team member will enter the correct password. The tester will verify that the system properly displayed the correct user GUI. The test team will perform this test five times for each user type.

Test case #20: This test will verify that the method used to log a user out of the TUT system is properly and efficiently coded. It will also verify that the system correctly displays “See You Next Time” prompt after the user clicks on the “I’m Done for Today” button on the GUI. The test team member will log in to the system as a PM user and then click on the “I’m Done for Today” button on the user GUI. The tester will then verify that the system properly displays the log off message. The test team will perform this test five times for each user type.

7.6 System Test

The final system test will be conducted at the Recreation and Wellness Center (permission pending) on the indoor track. The host system and hardware will be installed on the indoor track. A two transmitter node configuration and one WBT will be available for the test and all members of the development team will participate in the system integration testing. The physical placement of the nodes and distances between them will be determined by mechanical measurements. Placement of the nodes will be determined automatically by the TUTconfig utility.

The transmitter designer will log in to the TUT host and click on the “TUTconfig” button on the GUI. The team member will connect the transmitter to the host using a mini USB cable. The team member will enter all track configuration data into the GUI and then click the “Configure Transmitter” button. The system will then display the “Transmitter Node Placement” GUI showing the location of the transmitter that was just configured. The team member will then perform the above steps for all remaining transmitter nodes. When all transmitters have been configured the “Transmitter Node Placement” GUI will show the location of all transmitter nodes. The team members can now install all transmitters on the track in the correct locations and the transmitter designer can now log off of the TUT host.

The WBT designer will log in to the TUT host as a PM user and click on the “Retrieve Prescription” button on the GUI. The system will then upload the WBT configuration file to the WBT, retrieve the prescription for the current user ID, upload the prescription to the WBT, and display the assigned WBT number. The team member will remove the WBT from the USB connection enter the track and proceed to the designated start position. The team member will verify that the WBT correctly displays “Ready” when the designated start position is reached. Once the ready prompt is displayed on the WBT, the team member will start the prescribed exercise routine. After one lap around the track the, the tester will slow down to a casual walking pace and verify that the WBT properly prompts the user to increase speed. The team member will then increase speed back up to the prescribed pace and verify that the WBT properly indicates that the proper pace has been achieved. The team member will increase speed to a pace that is above the limit prescribed and verify that the WBT properly prompts the user to decrease speed. At the prompt, the team member will decrease speed until they are again within the prescribed limits and verify that the WBT properly indicates that the proper pace has been achieved. The team member will continue the exercise session prescribed and verify that the WBT correctly prompts the user at each stage of the prescribed routine.

Upon completion of the exercise session the team member will return the WBT to the TUT host USB connection and follow the prompt to log back in to the system. The team member will select the option to “Display Exercise Log” and verify that the data displayed correctly reflects the completed activity.

The above test sequence will be repeated as an SM user. In addition to the above sequence, the team member will verify that the WBT alarms and audible prompts can be silenced. A PM user is not allowed to silence and alarm or prompt since the routine has been prescribed by their physician.

8 Schedule and Costs

8.1 Milestone Chart

Date	Objective
08/23/11	Acquire and install all software tools on the development system. Initial funding of $200.00 per team member.
08/23/11	Order hardware development software tools
09/09/11	TUT host USB to WBT configuration utility mock-up (extends to transmitter) complete
09/16/11	Transmitter node mock-up to host working
09/16/11	TUT database implemented and tested
09/19/11	WBT mock-up display operational
09/26/11	WBT mock selection switches operational
09/26/11	TUT host GUI's implemented and tested
09/26/11	Transmitter node design proven
09/27/11	CDR transmitter node
09/30/11	CDR WBT
10/14/11	WBT mockup assembled and testing begins
11/07/11	WBT core code implemented
11/14/11	Host, WBT, Transmitter integration complete
11/21/11	TUT Website operational
11/25/11	TUT website Interface to TUT database complete
11/28/11	Software bugs fixed and hardware modifications complete
12/07/11	Final System testing complete

8.2 Cost Analysis

8.2.1 Hardware Development Tools

The costs of development tools are given in Table 9.

Tool Purpose	Description & Part Number	Vendor	Costs	Qty
WBT Prototyping Code development	Texas Instruments MSP-FET430U38: MSP430 38-Pin Target board and USB Programmer	Texas Instruments	$169.00	1
RF Development	EZ430-RF2500: MSP430 Wireless Development Tool	Digikey 813-1026-ND	$59.00	1
		Total Costs	$228.00	

Table 9-Development BOM

8.2.2 Hardware Bill of Materials

The bill of materials (BOM) for the transmitter is shown in Table 10. The WBT is shown in Table 11.

Transmitter BOM					
Schematic designator	Value tolerance	Description Manufacturer Part Number	Vendor Part number	Costs	Qty
Patch antenna	N/A	822335127708	HG2410DP	$29.99	1
RF Connect or	N/A	TE Connectivity / AMP 5226993-1	Mouser Electronics 571-5226993-1	$5.31	2
Antenna Cable	N/A	N/A	1 CC-BMNMA005	$35.00	1
Target Board	N/A	Texas Instruments EZ430-RF2500T	Digikey 296-23125-ND	20.40	2

Table 10-Transmitter Node Bill of Materials

Wrist Band Tracker BOM					
Schematic designator	Value tolerance	Description Manufacturer Part Number	Vendor Part number	Costs	Qty
U1	N/A	Texas Instruments MSP430F2274DA 38 pin microcontroller	Digi-Key	$19.41	3**
P1	N/A	FTDI MM232RQ USB-Serial UART development module	Digikey 768-1021-ND	$18.50	1
U2	N/A	Texas Instruments eZRF430-2500D	Texas Instruments eZRF430-2500D	$20.00	1
U3	N/A	Texas Instruments TPS61221DCKT Boost Converter SOT-23-6 PKG	Texas Instruments Sampled	$0.00	1
X1	32.768 kHz	Epson Toyocom XTAL 32.7680KHZ 9.0PF SMD	Digikey SER2416CT-ND	$1.40	1
U$1	N/A	Crystalfontz OLED display	Crystalfontz 2BM-10000	$48.00	1
R1	47k Ω 5%	Resistor SMD 402	Digikey 311-47KJRDKR-ND	$0.12	10*
R2	330 Ω 5%	Resistor SMD 0402	Digikey 311-10KJRDKR-ND	$0.12	10*
R3	1.0k Ω 1%	Resistor SMD 0402	Digikey 985-1323-2-ND	$1.06	10*
L1	N/A	Coil Craft Inductor 4.7uH EPL3015-472MLB	Coil Craft	$10.00	2*
C4,C5	10 µF 10%	AVX TPSB105K020R1000 Tantalum Capacitor 1 Ω ESR1210 (3528 Metric)	Digikey 478-2402-1-ND	$1.25	2
C2,C3	10pF 5%	Ceramic capacitor SMD 0402 AVX - 04025A100JAT2A	Digikey 478-1072-1-ND	$1.10	10*
C1	100pF 10%	Ceramic capacitor SMD 0402 AVX - 01016D101KAT2A	Digikey 478-6436-6-ND	$2.75	10*
PCB	N/A			$50.00	1
			Total Costs	175.89	

Table 11 - WBT Bill of Materials

8.2.3 **Mechanical Packaging and Mounting**

The WBT case will be fabricated in house from Plexiglas scrap material and micro-machine CNC tools available to a team member.
The transmitter nodes will be housed in a 6” x 4” x 2” deep plastic box and is included in the transmitter BOM, Table 10.

8.2.4 **Software Bill of Materials**

The 3rd party software requirements for developing and implementing the TUT application software, GUI’s, and utilities is given in Table 12, Software bill of materials (BOM).

Software BOM				
Schematic designator	Value and tolerance	Description Manufacturer Part Number	Vendor Part number	Costs
N/A	32bit Windows 7 Version	Microsoft Information Interchange Service	IIS 7.5	$0.00
N/A	32bit Windows Version	Java Runtime Environment	Java JRE v7	$0.00
N/A	32bit Windows Version	Java JDK	JDK7	$0.00
N/A	32bit Windows Version	Eclipse EE IDE (integrated development environment)	Eclipse Indigo	$0.00
N/A	N/A	Jigloo GUI builder for Eclipse	4.6.6	$0.00
N/A	N/A	JavaMail Mail API	1.4.4	$0.00
N/A	N/A	COMM2.0 Java Communications API	2.0	$0.00
N/A	32bit Windows Version	MySQL relational database software	MySQL Community Server 5.5.18	$0.00
N/A	32bit Windows Version	MySQL Workbench integrated development tools – GUI for MYSQL	MySQL Workbench 5.2.35 CE	$0.00
			Total Costs	$0.00

Table 12-Software Bill of Materials

8.2.5 **Costs Summary**

The total costs are calculated to be $592.95. A breakdown is given in Table 13-Total Costs breakdownTable 13.

Cost Summary	
Item Description	Costs
Development Costs	$177.95
WBT Prototype	$120.00
2 Transmitter Nodes	$220.00
Administrative-Shipping-Coffee	$75.00
Total	$592.95

Table 13-Total Costs breakdown

9 Operation

9.1 WBT Operation

9.1.1 Switches and Connections

Figure 39 shows the locations of the four user input switches, OLED display panel USB connector and battery pack.

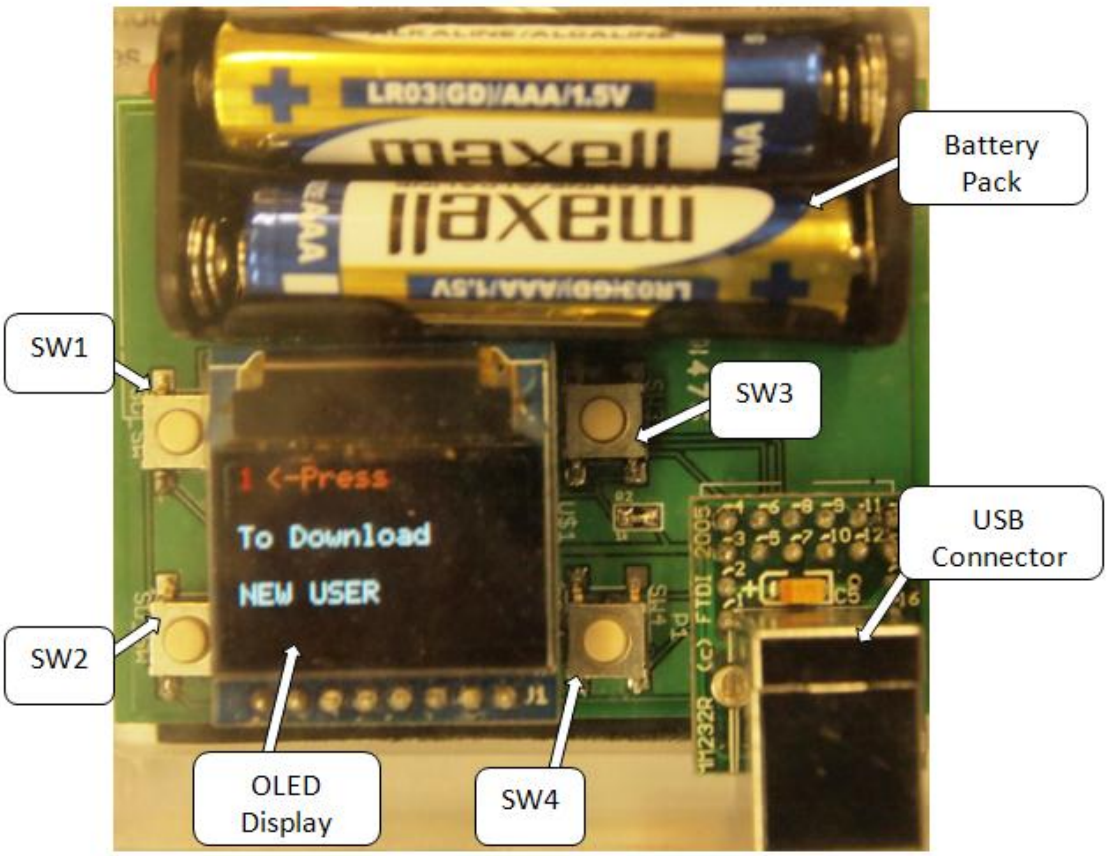

Figure 39-Switch and component locations

The switch functions are:
1. SW1-Various, function indicated by the display.
2. SW2-Display On/Off.
3. SW3-Contrast Up.
4. SW4-Contrast Down.

9.1.2 **WBT Session operation**

On Installation or changing of the batteries, or upon a reset the WBT will display the message shown in Figure 40. This indicates no user or track information is stored in the unit. The unit is waiting for the USB connection to the host.

Figure 40-Displayed message indicating no user is present

Connect the USB cable, the display flashes the messages “USB connected” followed by ”bye bye….” .

Follow the procedure outlined in retrieving a prescription. After the user data is loaded, the display will indicate success as shown in Figure 41.

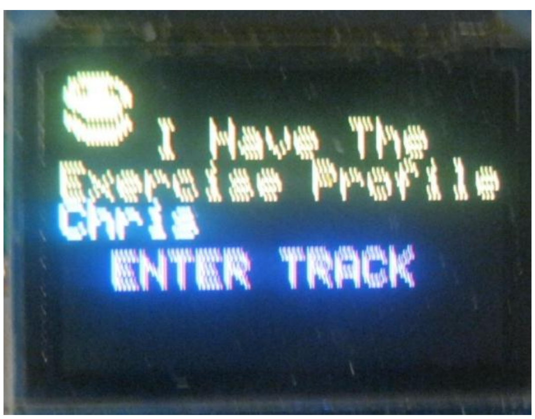

Figure 41-Successful download of a user prescription

The WBT is detached from the USB and the user enters the track near the first transmitter node. When the user is in the proper place on the track, the display will be as shown in Figure 42.

Figure 42-Display when user is at first node

When ready to begin the exercise session, the start button (SW1) is depressed and the session menu is displayed. The meaning of the display items is shown in Figure 43. The timers in the WBT will not start until the user leaves the first node.

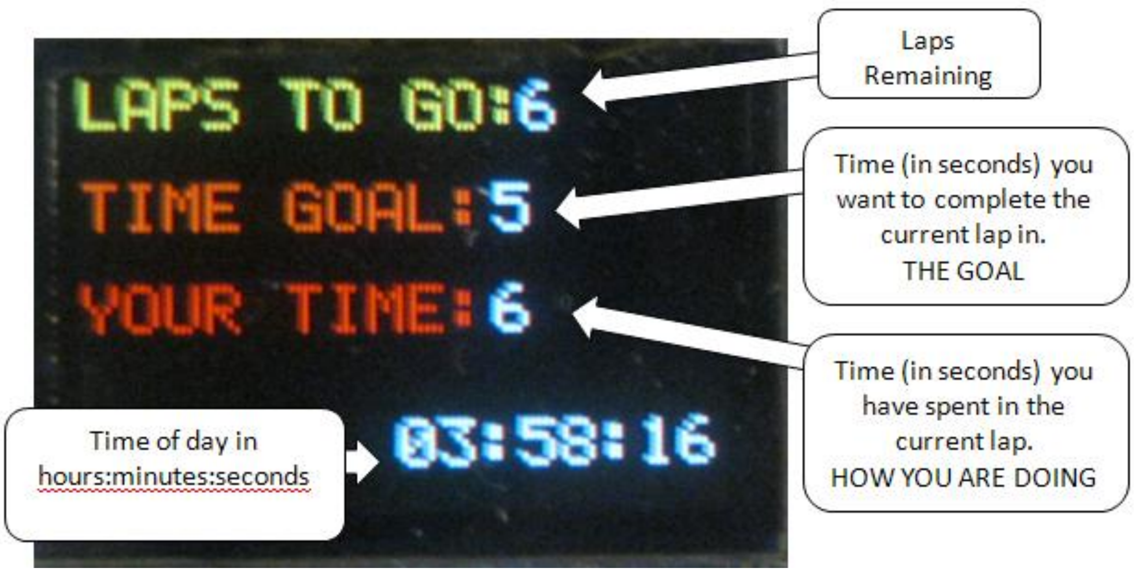

Figure 43-Track session display

The WBT will continue to set a time goal for each lap based on the performance of the previous lap or laps and the prescription information. When the last lap has been completed the display will indicate as shown in Figure 44.

Figure 44-Finished session display

Reconnect the WBT to the host via the USB port. The display will flash the message “USB reconnected” , “Uploading user data”, “Bye Bye”.
The display will disappear until the upload is complete. It will then display as shown in Figure 45.

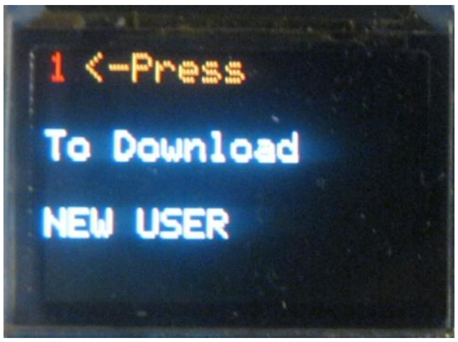

Figure 45-Display after user data uploaded successfully to the host.

Pressing SW1 will indicate to the WBT a new user is ready for download and the process will repeat.

9.2 TUT Software Operation

9.2.1 Interactions and Procedures

The user interfaces of the TUT system were designed to be quite simple. Each user type has only a few operations that they would have to perform to use the system. This guide will show the steps required for each user and the results of each selection. All users login the same way and the TUT login GUI is the same for all users. This screen is shown in Figure 46 below.

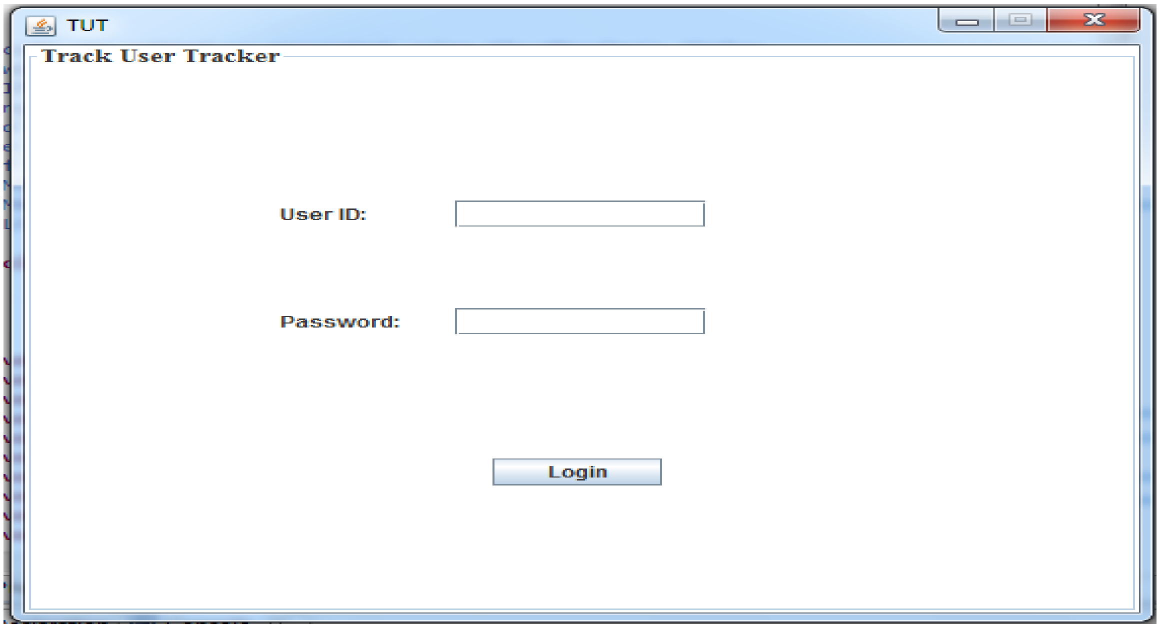

Figure 46-TUT Main Login Screen

Administrator
Enter your user ID and password then click on the “Login” button and you will be taken to your main menu screen. This is the GUI shown in Figure 47 below.

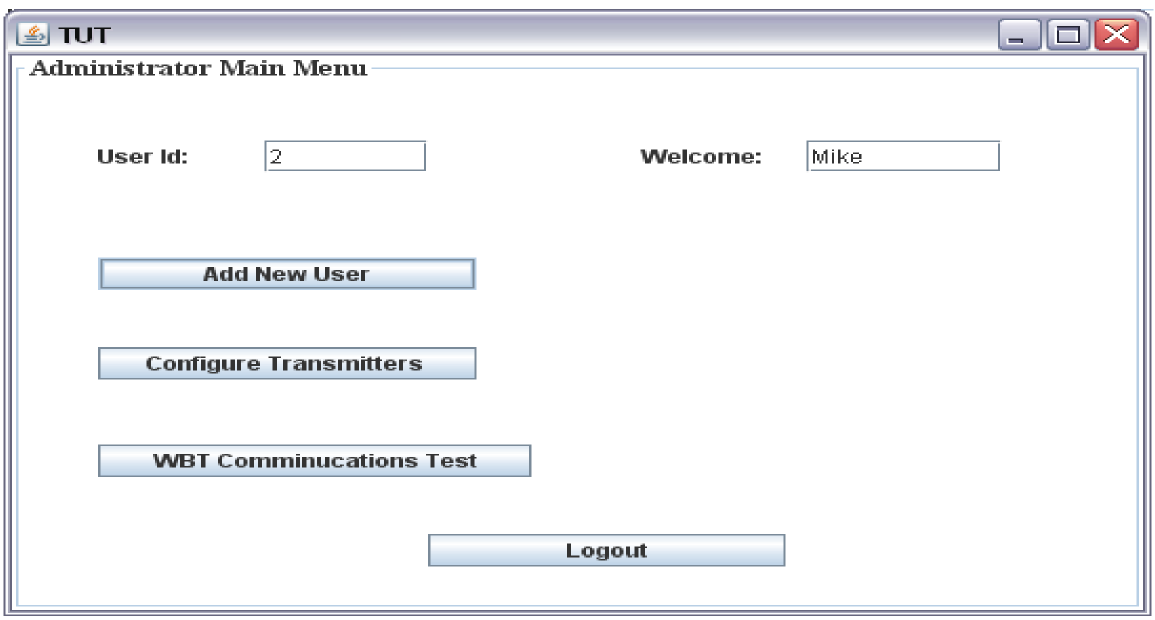

Figure 47-Administrator Main Menu

All users will be shown their user ID followed by a greeting with their first name so they can verify proper login. There are three choices from the main menu. The administrator can only add a new casual user to the TUT system. If they click on the “Add New User” button they will be taken to the new casual user sub menu shown in Figure 48.

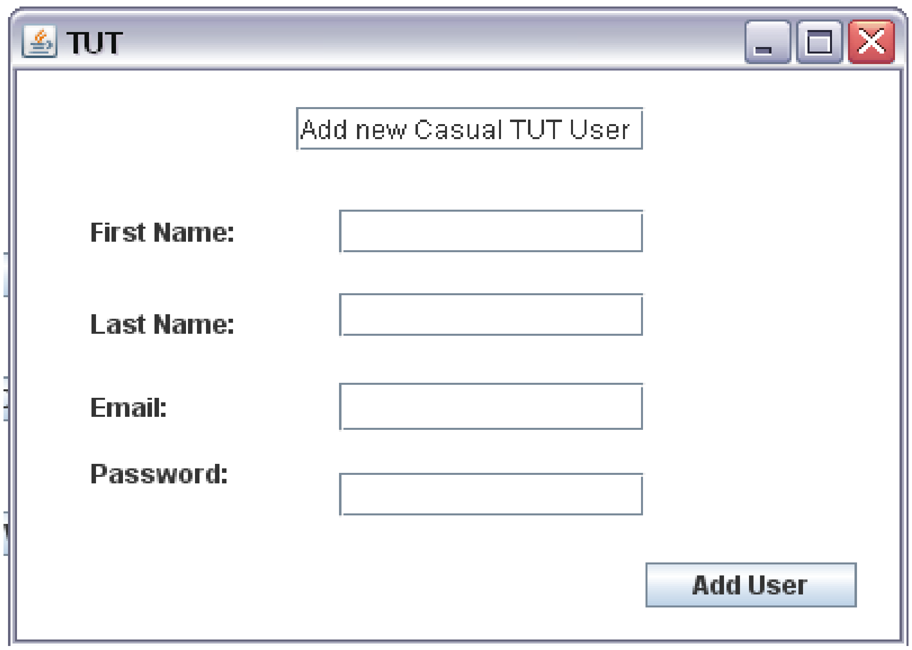

Figure 48-Add new casual user GUI

After the administrator finishes entering the new user information and clicks on the “Add User” button they would be presented with the new user’s Id number. The administrator would then give the Id number to the user so they could use it to log in to the system. This would look like the screen shot shown in Figure 49.

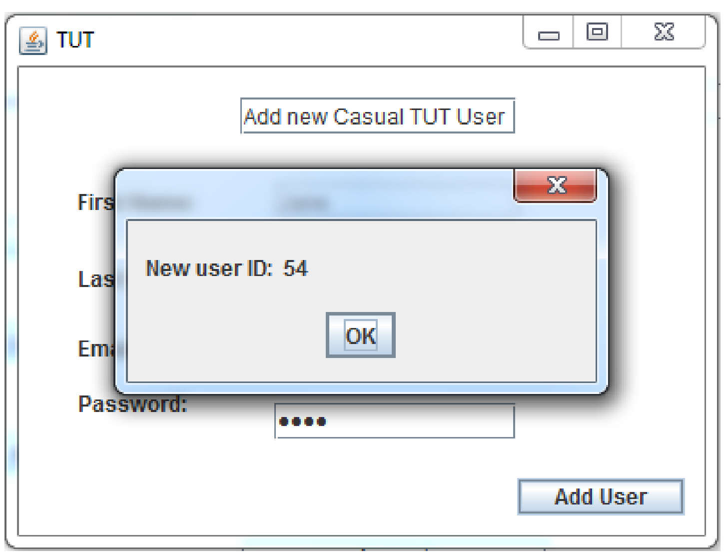

Figure 49-New User Id

When you are ready you will click on the “Ok” button and the system will take you back to your main menu. From there you may click “Logout” when you are finished. The other two options for the administrator were not implemented in the prototype release of the TUT system so there are no screen shots to show for these options.

Casual User
Enter your user ID and password then click on the “Login” button and you will be taken to your main menu screen. This is the GUI shown in Figure 50 below.

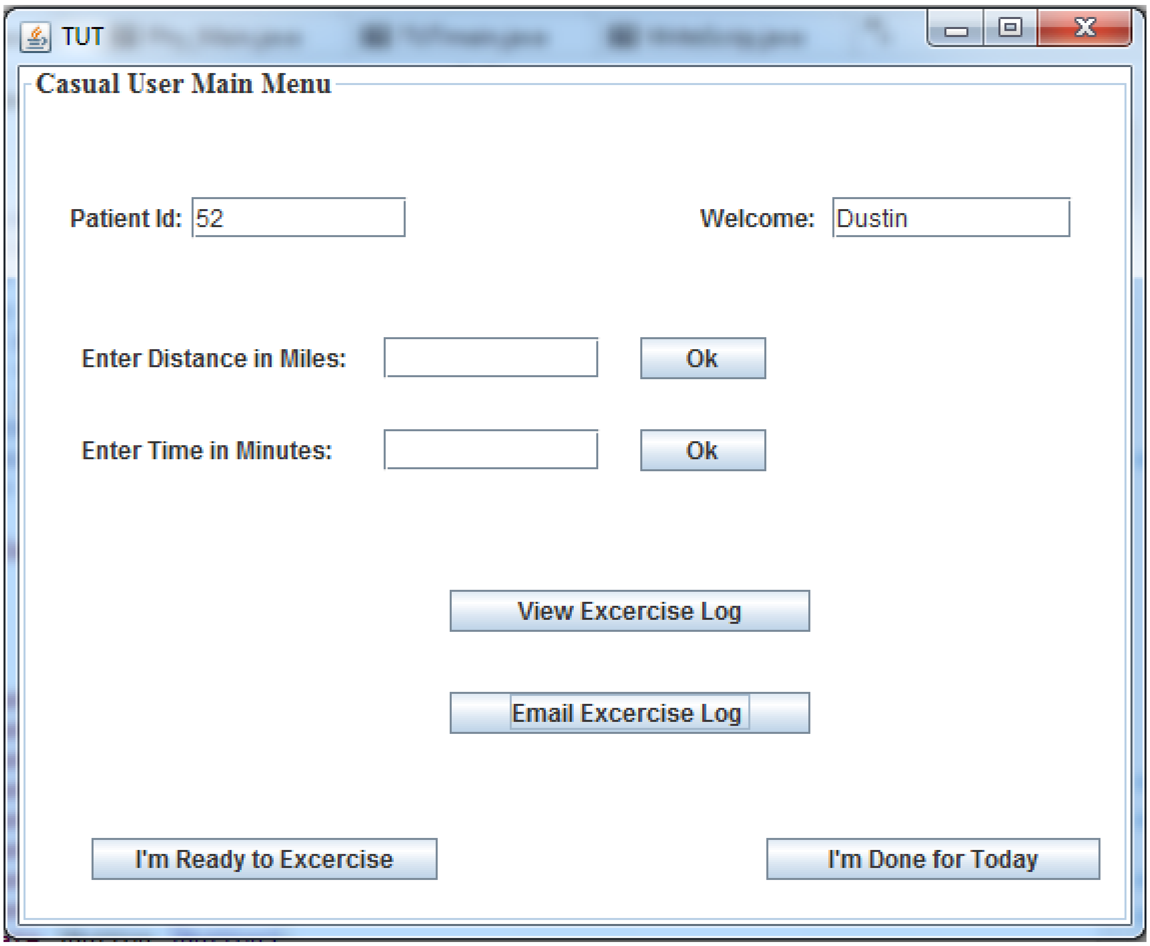

Figure 50-Casual User Main Menu

Before you start your exercise session you will select either a distance goal or a time goal, enter the corresponding value in the text box and then click on the “Ok” button next to your entry. After that you will be prompted that your WBT has been programmed with your exercise goals and you may press click the “I’m Ready to Exercise” button when ready. Now you will be prompted to take your assigned WBT and proceed to the track for your exercise. When you have finished and you return your WBT, you will log back in to the system and select either “View Exercise Log” or “Email Exercise Log”. If you select the view option you will be presented with your exercise statistics for the current session. If you select the email option, the system will email you your exercise log to the email stored in the database. If you do not have an email address already in the database, the system will prompt you for your email address. This prompt is shown in Figure 51.

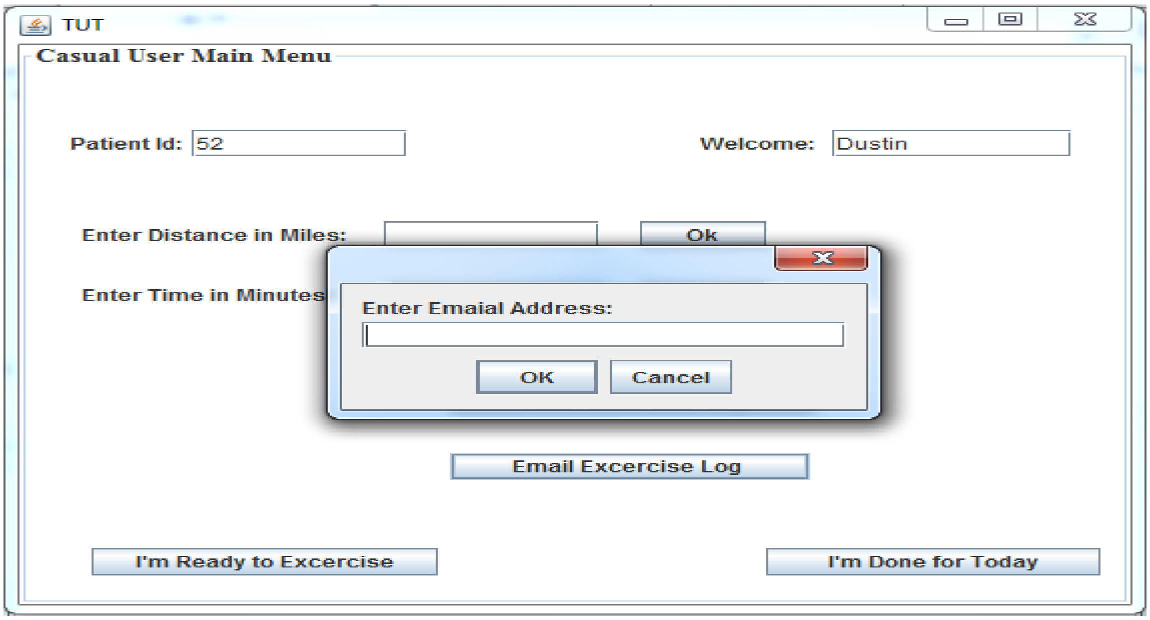

Figure 51-Casual user - no email address in database

After you have finished you will just click on the “I'm Done for Today” button and the system will log you out.

Patient

Enter your user ID and password then click on the “Login” button and you will be taken to your main menu screen. This is the GUI shown in Figure 52 below.

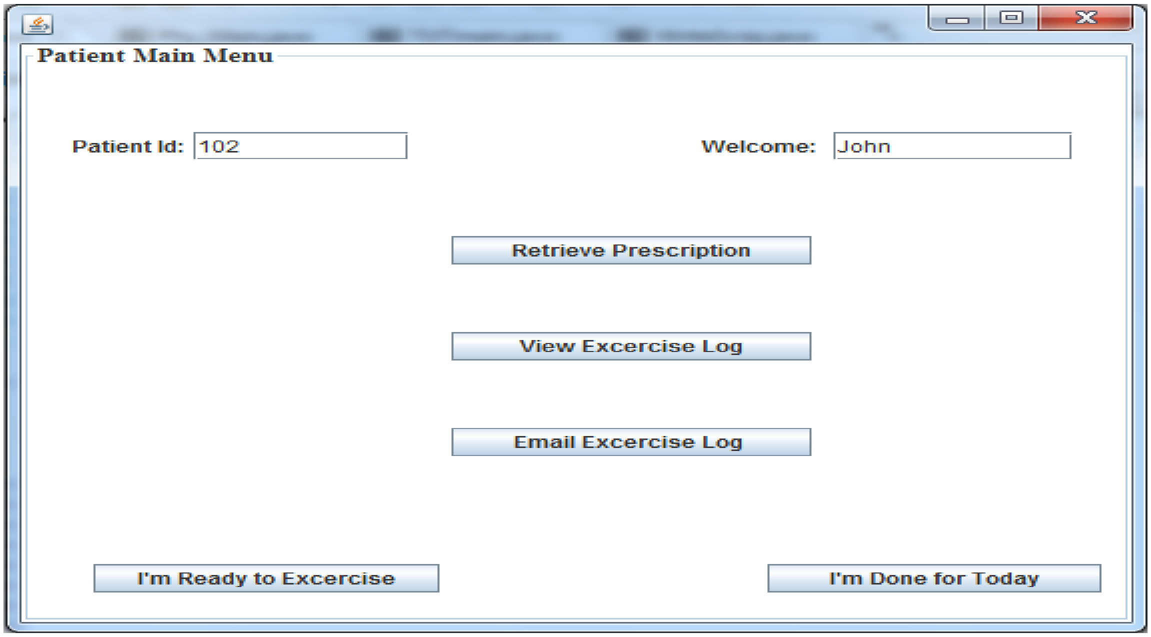

Figure 52-Patient Main Menu

As a patient you do not have the option of setting your own goals. You must follow your prescribed routine. From your main menu click on the "Retrieve Prescription" button and the system will retrieve your prescription and load it into a WBT. The system will then inform you that your prescription has been sent to the WBT and you may proceed when ready. This is shown in Figure 53 below.

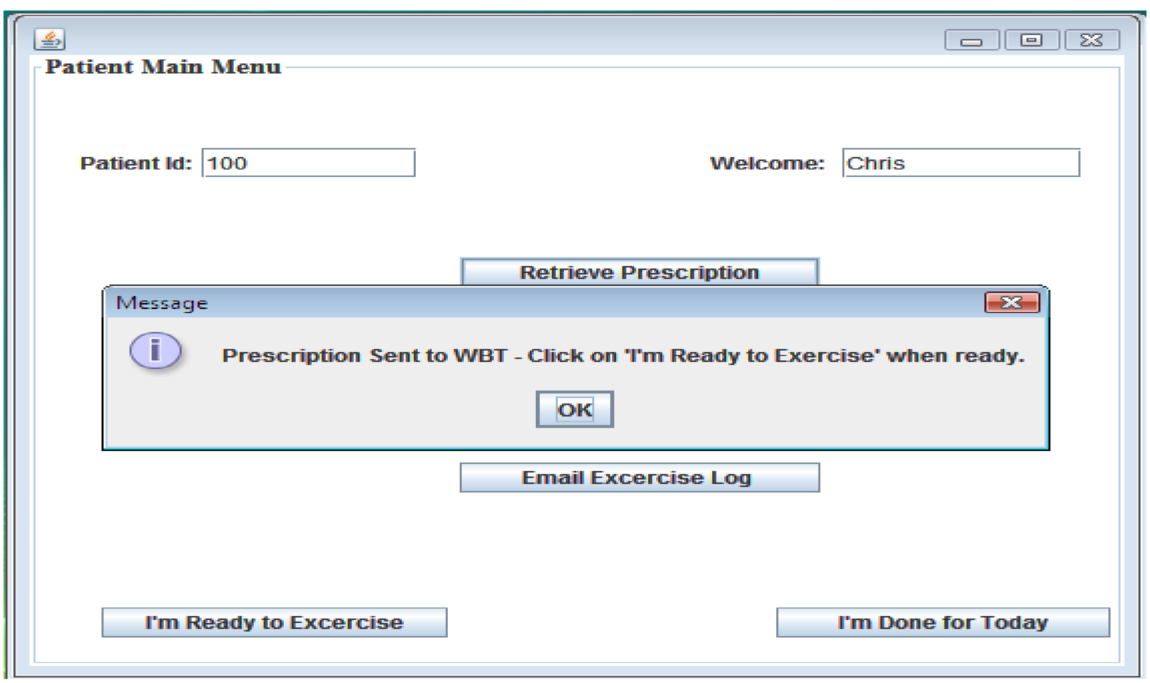

Figure 53-Prescription sent to WBT

You will then click on the “I’m Ready to Exercise” button, take your WBT and proceed to the track. When you have finished and you return your WBT, you will log back in to the system and select either “View Exercise Log” or “Email Exercise Log”. If you select the view option you will be presented with your exercise statistics for the current session. This exercise log view is shown in Figure 54.

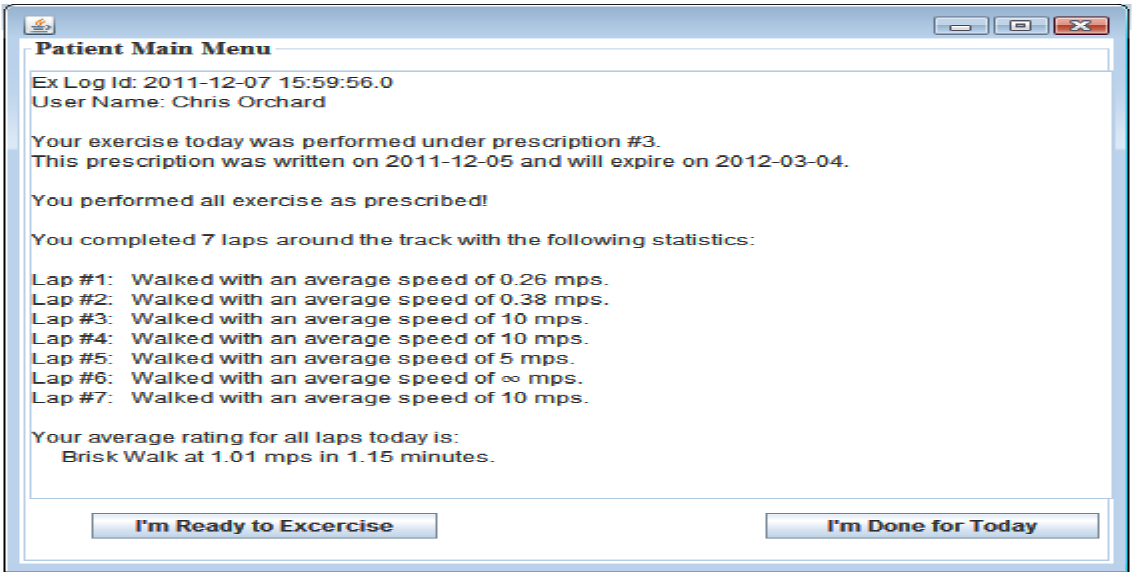

Figure 54-Patient Exercise Log

If you select the email option, the system will email you your exercise log to the email stored in the database. If you do not have an email address already in the database, the system will prompt you for your email address. This prompt is shown in Figure 55.

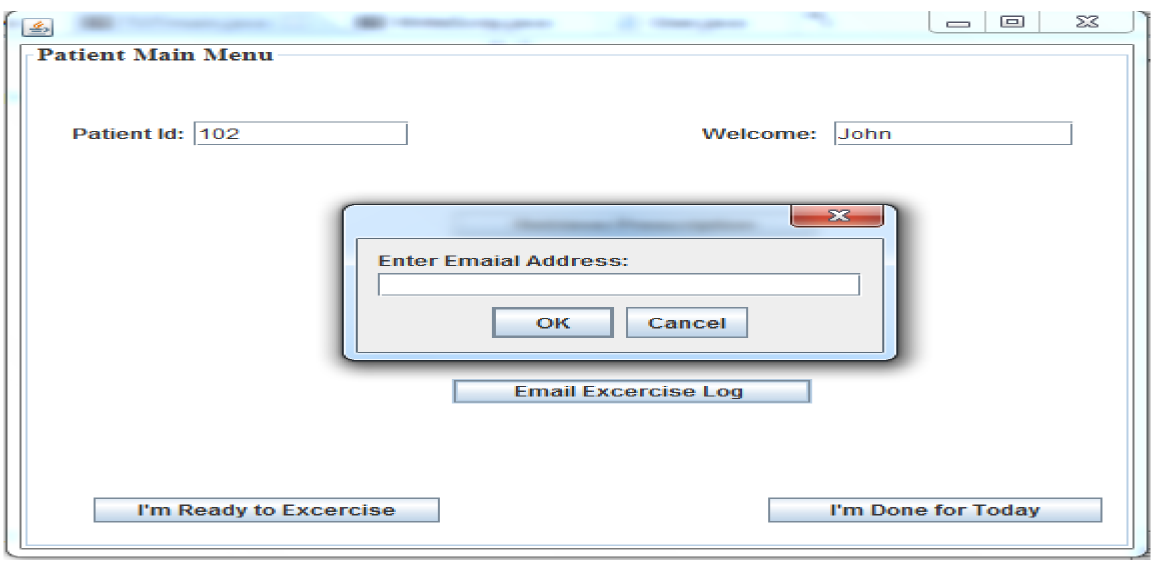

Figure 55-Patient - no email address in database

After you have finished you will just click on the “I’m Done for Today” button and the system will log you out.

Physician

As a physician you will log in to the TUT system from the TUT website. The website is located at www.tutweb.zapto.org. Once you are on the TUT web site, you will click on the link labeled “Start TUT Web App”. After the application starts you will be presented with the TUT main login screen where you will enter your user ID and password then click on the “Login” button and you will be taken to your main menu screen. This is the GUI shown in Figure 56 below.

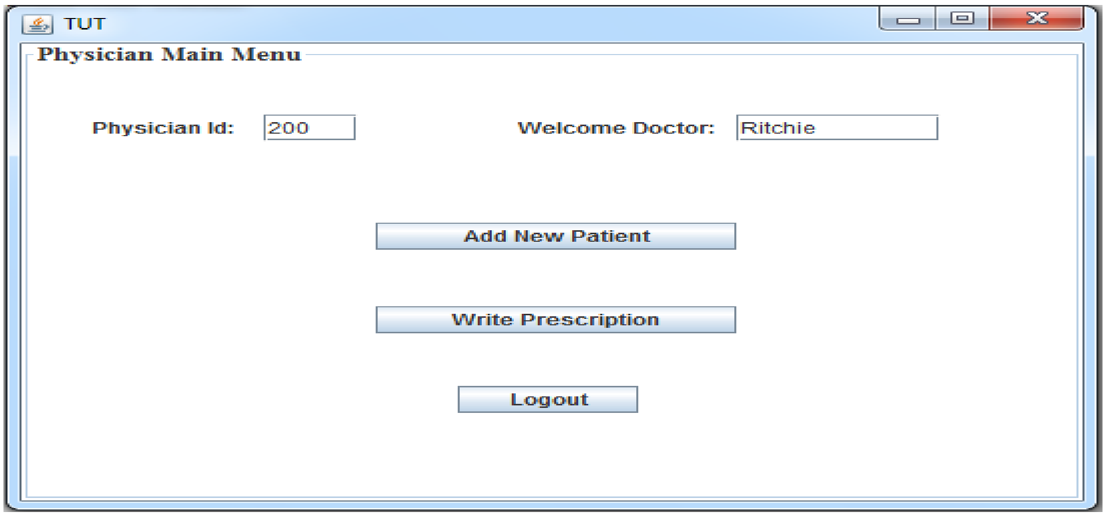

Figure 56-Physician Main Menu

If you select the “Add New Patient” option you will be taken to the add patient sub menu shown in Figure 58 below.

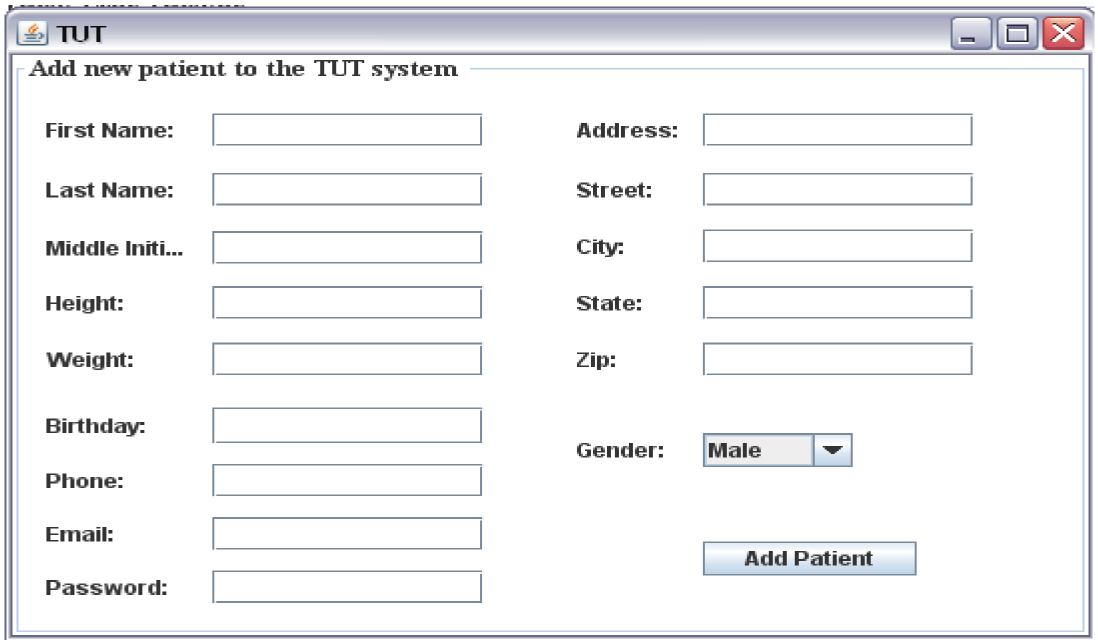

Figure 57-Add New Patient GUI

After the physician finishes entering the new patient information and clicks on the “Add Patient” button they would be presented with the new patient’s Id number. The physician would then give the Id number to the patient so they could use it to log in to the system. This would look like the screen shot shown in Figure 58.

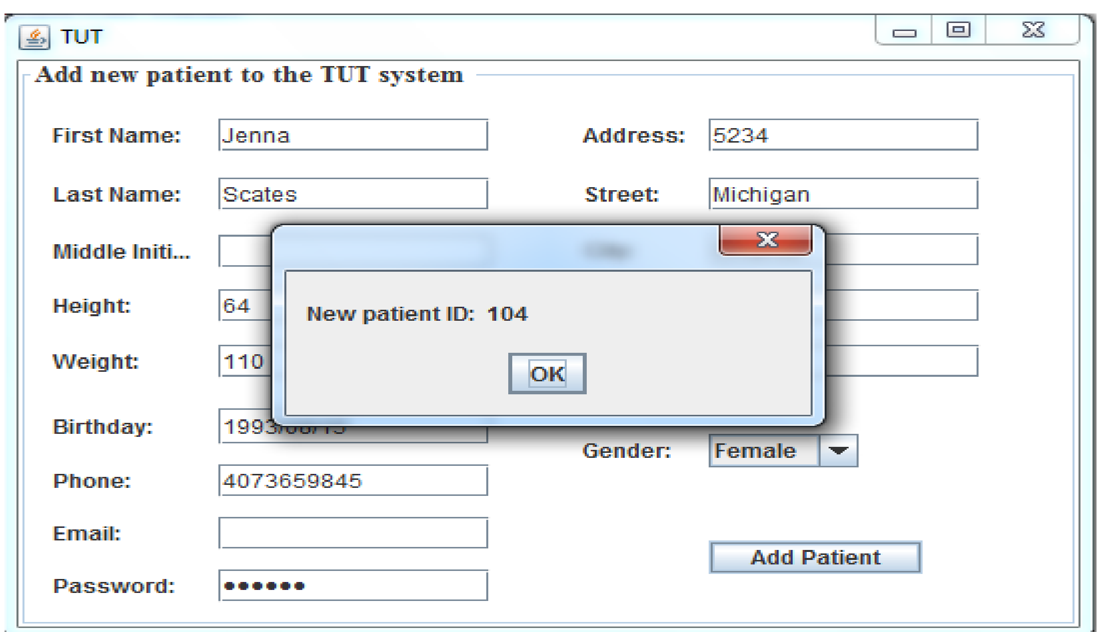

Figure 58-New Patient Id

When you are ready you will click on the “Ok” button and the system will take you back to your main menu. From you main menu you may also choose to issue a prescription to a patient by clicking on the “Write Prescription” button. If you make this selection you will be taken to the prescription sub menu show in Figure 59 below.

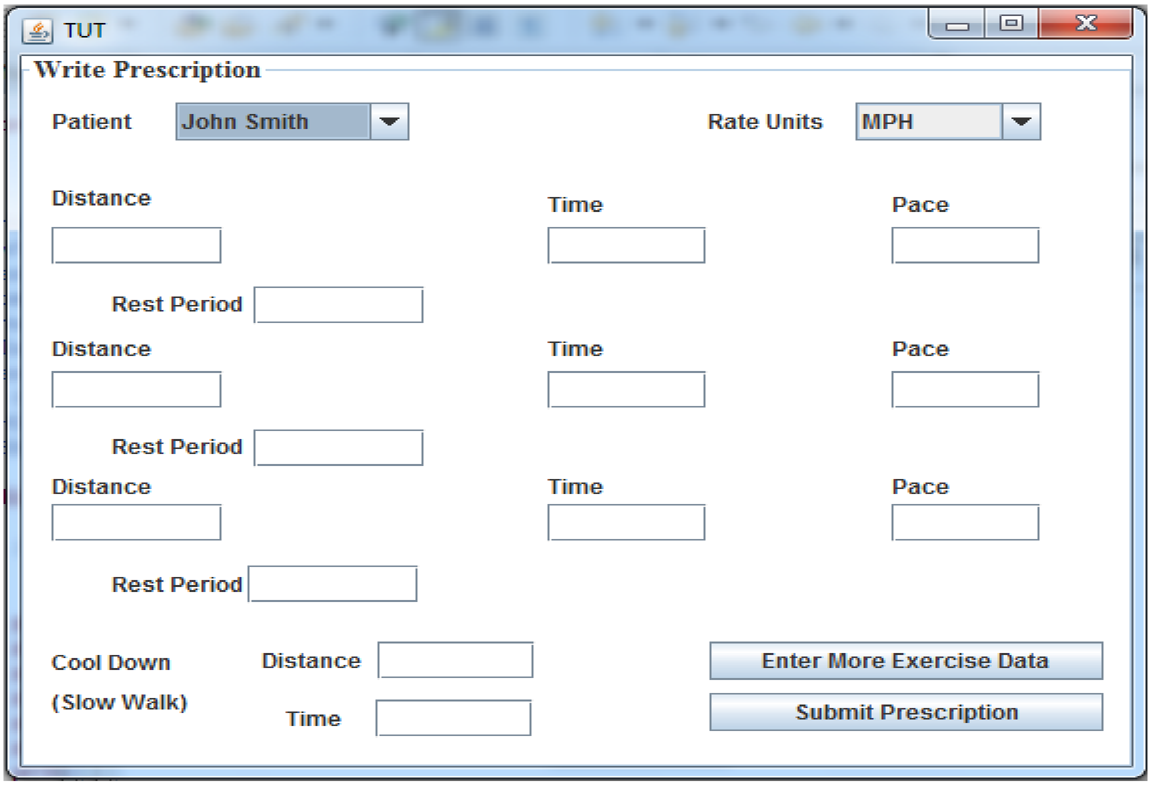

Figure 59-Write Prescription GUI

From here you will first select your patient by name from the drop down menu in the upper right, select the units for distance and pace, and then enter your prescription data. You do not need to use all spaces but you must enter at least one distance and one pace or one time and one pace or one time and one distance. The rest periods and the cool-down are also optional. After you have finished entering prescription data just click on the “Submit Prescription” button and you will be taken back to your main menu where you may log out when ready.

APPENDIX 1

Works Cited

1. **Mayo Clinic Staff.** Walking: Trim your waistline, improve your health. *http://www.mayoclinic.com.* [Online] December 18, 2010. [Cited: June 07, 2011.] http://www.mayoclinic.com/health/walking/HQ01612.

2. **Boston Athletic Association .** Participant Information: Qualifying. *2011 Boston Marathon.* [Online] Jan 01, 2011. [Cited: July 5, 2011.] http://www.baa.org/Races/Boston-Marathon/Participant-Information/Qualifying.aspx.

3. **Quach, Jeremy Gurski & Lee Ming.** Display Technology Overview. *http://www.lytica.com.* [Online] July 1, 2005. [Cited: June 14, 2011.] http://www.lytica.com/content/files/Documents/WhitePapers/Display%20Technology%20Overview.pdf.

4. Web Based EMR: Perhaps more affordable than you think. Tough economic times allow opportunities for Physician Practice **.** *www.emrexperts.com.* [Online] Jan 15, 2011. [Cited: July 14, 2011.] http://www.emrexperts.com/articles/emr-WebBased.php.

5. **Pfleeger, Shari L and Atlee, Joanne M.** *Software Engineering, Theory and Practice.* 4th Edition. Upper Saddle River : Prentice Hall, 2009.

APPENDIX 2 PERMISSIONS

Permission 1
Subject:
Re: COP4331
From:
Damla Turgut <turgut@eecs.ucf.edu>
Date:
3:09 AM
To:
Chris Orchard <corchard@cfl.rr.com>

Chris,

Sure, I remember you. You are most welcome to use the diagram as long as you cite the reference as the textbook since the slides were provided by the author.

Wishing you the best in completing your SD I class.

Cheers,
Damla

On Fri, Jul 29, 2011 at 1:11 AM, Chris Orchard <corchard@cfl.rr.com> wrote:
> > Dr. Turgut, I was in you COP4331 class in the Fall 2010 semester. I am in
> > Senior Design I this semester and I wanted to know if I could have
> > permission to use one graphic from your power point lecture notes? The
> > graphic that I would like to use is from the slide describing the
> > "Prototyping Model" from chapter 2 in the text. (chapter 2, slide #18) Thank
> > you for your consideration.
> >
> > Chris M. Orchard
> >

-- Damla Turgut, Ph.D. Associate Professor Department of EECS University of Central Florida Orlando, FL 32816-2362 Tel: (407) 823-6171 Fax: (407) 823-5835 Email: turgut@eecs.ucf.edu Web: http://www.eecs.ucf.edu/~turgut

Permission 2

Date: Wed, 13 Jul 2011 16:34:41 -0500
From: support@ti.com
Subject: RE: Service Request # 1-680916883
To: fevri@knights.ucf.edu

Hello Fevri,

Please find below the links that will assist you in retrieving the information you requested. The schematic for the CC1101 is located in the CC1101EM434MHz Reference Design 2.0.0 Zip File which can be found in the CC1101DK433 Product folder, and the antenna information can be found in the CC1101-Q1 Product Folder under the "Application Notes" section of the "Technical Documents" heading.

CC1101EM434MHz Reference Design 2.0.0 (Rev.A)

http://www.ti.com/litv/zip/swrr046a

CC1101-Q1 Product Folder

http://focus.ti.com/docs/prod/folders/print/cc1101-q1.html

In regards to your inquiry on using the datasheet in your project, Texas Instruments is pleased to provide the information on these pages of the World Wide Web. We encourage you to read and use this information in developing new products.

TI grants permission to download, print copies, store downloaded files on a computer and reference this information in your documents only for your personal and non-commercial use. But remember, TI retains its copyright in all of this information. This means that you may not further display, reproduce, or distribute this information without permission from Texas Instruments. This also means you may not, without our permission, "mirror" this information on your own server, or modify or re-use this information on another system.

TI further grants permission to non-profit, educational institutions (specifically K-12, universities and community colleges) to download, reproduce, display and distribute the information on these pages solely for use in the classroom. This permission is conditioned on not modifying the information, retaining all copyright notices and including on all reproduced information the following credit line: "Courtesy of Texas Instruments". Please send us a note describing your use of this information under the permission granted in this paragraph. Send the note and describe the use according to the request for permission explained below.

http://www.ti.com/corp/docs/legal/copyright.shtml?DCMP=TIFooterTracking&HQS=Other+OT+footer_copyright

I hope this helps. If you have any other questions or concerns, please feel free to contact us.

Best Regards,
Darian Martin

TI assumes no liability for applications assistance or customer product design. Customer is fully responsible for all design decisions and engineering with regard to its products, including decisions relating to application of TI products. By providing technical information, TI does not intend to offer or provide engineering services or advice concerning Customer's design. If Customer desires engineering services, the Customer should rely on its retained employees and consultants and/or procure engineering services from a licensed professional engineer (LPE).

Please do not delete the below Thread ID when replying to this email, doing so will delay our response to your inquiry

[SR THREAD ID:1-B9EETV]