University of Puerto Rico
Department of Electrical and Computer Engineering
Mayagüez Campus

CPR Electronic Teaching Assistant Mannequin Proposal

February 26, 2009
ICOM 5047 – Computer Engineering Project Design Course
Professors: J. Fernando Vega-Riveros, Nayda G. Santiago and Manuel Rodríguez
EHS team: Arelis Perez Peña, Jose A. Lombay Gonzalez, Juan C. Gorritz García,
Jomar Rosario Muñiz, Edvier Cabassa Miranda

Abstract

Cardiopulmonary resuscitation (CPR) is a procedure used with unconscious victims of some type of trauma which has caused the heart to stop beating in order to simulate a functional heartbeat and oxygen intake through breathing [3]. Currently, CPR training is usually performed using a mannequin and supervised by an instructor. We would like to improve the quality of the training process to perform CPR. For that, we will create a tool to facilitate the job of the instructors by electronically monitoring students practicing CPR on a training mannequin. Our product will consist of a mannequin that will be monitored using different sensors and a software application which will retrieve and process the information in order to provide proper training and support the instructor. The mannequin will simulate a person who needs CPR and will respond to the steps if they were applied according to the rules of the CPR manual or instructor. The trainer will use software installed in a normal computer to control the mannequin's actions and obtain results of his or her students in real time, with the ability to save the data or retrieve previous session data from a database.

Table of Contents

Executive summary

Introduction

Cardiopulmonary resuscitation (CPR) is a practice used with unconscious victims of some type of trauma which has caused the heart to stop beating [3]. It consists of a series of steps, primarily compressing the victim's chest to get the heart to pump blood, richly oxygenated, throughout the body.

CPR training is essential in many working environments, including jobs ranging from law enforcement to babysitting [1]. However, if performed incorrectly, it can produce adverse effects. For this reason, the Red Cross provides training programs to certify the people that are capable of performing the procedure [2]. In order to obtain the certification, the person has to pass a final practical exam which is performed using a CPR mannequin that simulates a person in need of CPR. Instructors want to be sure that the steps are learned correctly since the practice may be used on a live person in the future.

We want to provide a more realistic way to perform the practical exam, improving the techniques of teaching this procedure. Using a mannequin that is monitored and controlled by sensors, a microprocessor, and computer software in a personal computer, the test can be an experience near to real life. It does not relieve the CPR trainer of the responsibility of evaluating the student, but it will be helpful in this task.

Marketing

Market Description

The market provides several CPR mannequins. CPR Savers & First Aid Supply offers a variety of CPR mannequin trainers, each with different features and set pieces amount. The Ambu CPR Pal Training Mannequin also has a first aid kit and a patented hygienic system that protects students and makes internal cleaning unnecessary. Some of the features that the mannequins have are that the airways open when the CPR mannequin’s head is correctly positioned and the lung inflation is indicated by a whistling sound.

MSR: Israel Center of Medical Simulation provides a CPR mannequin that is used in adult CPR training. Some features the mannequin has are natural airway obstruction, carotid pulse simulation, realistic chest compliance for chest compressions, and it allows head tilt lift. These features are basically the most common that companies offer in mannequin trainers. CPR Electronic Teaching Assistant Mannequin will be more than that. CPR e-TAM will be a complete CPR mannequin with the most realistic environment features. The trainers that use CPR e-TAM will have an accurate assistant and a complete grading system for each student when they take a CPR test.

The only mannequin we were able to find that connects to a computer, AmbuMan, uses a USB connection and is able to retrieve data sent from the mannequin, which by itself is realistic, however gives no visual or audible feedback. Our mannequin, apart from having a wireless connection for better portability, is specialized for universities or other teaching institutions rather than strictly for certification, which is where the custom grading profiles differs from the AmbuMan software, and considering materials and development costs is much more affordable, an important factor for public institutions.

Market Overview

Our product will be oriented to any institution that is interested in providing CPR training since our product will not be replacing actual training procedures, but complementing them with more realistic features. Currently there are several types of mannequins that consist of realistic features in the physical aspect, but they do not provide information on whether or not the steps are being done correctly and they completely depend on the supervision done by the trainer.

One of the models in the market that does provide electronic information about procedures being realized on the manikin is the “Ambu CPR Pal Training Mannequin”, also known as AmbuMan. While their product measures air ventilation and chest compressions, our product will be identifying if noise was produced to verify the consciousness of the victim and if neck position is correct in addition to measuring air ventilation and chest compressions. Also, we found their software to be very complicated, despite of the fact that 3 of the 5 group members are familiar with CPR. Our group will have feedback from at least one CPR trainer in order to provide the most usable and useful user interface possible. Their software, however, has the added functionality of a course planner, which, up to this point, is not a planned feature.

Document Description

This document presents a proposal for a CPR Electronic Teaching Assistant Mannequin. Our team identifies the issues and how we will solve them using Computer Engineering. It contains detailed specifications for both the hardware and software modules necessary for the system. The hardware consists of a CPR mannequin, different sensors, a microcontroller, and a wireless device which will be used to capture the information of what is being performed on the procedure. The software will consist of a Windows application which will be used to process that information and display it in an organized manner. It will also provide the user with logs of past progress of students. From the summary of the proposal to a work breakdown structure (WBS) and Gantt charts are provided in order to give all the information we have of the prototype to implement for our costumer. A risk analysis and budget are presented to complete the information regarding market analysis.

Solution

Objectives

To help in the task of preparing better trained people in CPR we will accomplish the following objectives:

- Create a tool within two months to facilitate the job of the instructors by electronically monitoring students practicing CPR on a training mannequin
- Provide the hardware and software to identify when a student is not following the correct procedure and present the results in tabulated form, according to CPR standards
- Develop specialized software to configure the mannequin and display the data provided from the real time situation in the mannequin with a delay of less than two seconds
- Develop a training system that will reduce the evaluation of a CPR procedure by at least a factor of four (one fourth faster)

Scope

We will provide a product to improve the current learning tools available in medical services related to CPR. We will develop a mannequin that contains several push-button switches to provide information such as if the student verified the pulse of the mannequin and if the chest compression is adequate. A pressure sensor would be used to measure the pressure of the air provided by the learner is not too high for the lungs of the patient, and a proximity sensor used to verify that the respiration of the mannequin was checked. LED and buzzers would be indicators of the status of the mannequin, if it has been resuscitated, if it is still unconscious, or if the learner failed the CPR process. We would be using a micro-controller to collect all the data from the different sensors and wirelessly transmit the information to a computer where the software application would be running showing the status of the procedure being carried out.

The aforementioned software application would be developed by us as well. We would have to develop a graphical user interface (GUI) with useful information about the procedure like the pressure of the air provided by the learner, the time duration of the session and if there have been any mistakes during the session. The application should have a way to create a report of a training session. We would be using a relational database to store all the information being generated for future reference. The application should also have a feature to configure how many times the CPR procedure would have to be repeated to complete the session successfully. All this will be deployed following the specifications established in the WBS and budget sections of this document.

System Limitations

Even though we would like to make our system as real as possible, we know that it is not possible to do it. Therefore, the system will not:

- Simulate the pulse throughout the whole body
- Reproduce the breathing of the mannequin after it has been "revived"
- Be able to evaluate the student without any type of supervision
- Be able to simulate an emergency with a large amount of noise disruption in the area, as this would interfere with the first step, which is shouting to the victim to determine consciousness
- Know if the user physically touched the mannequin to determine consciousness while shouting
- Monitor the hand position for the compressions

Schedule

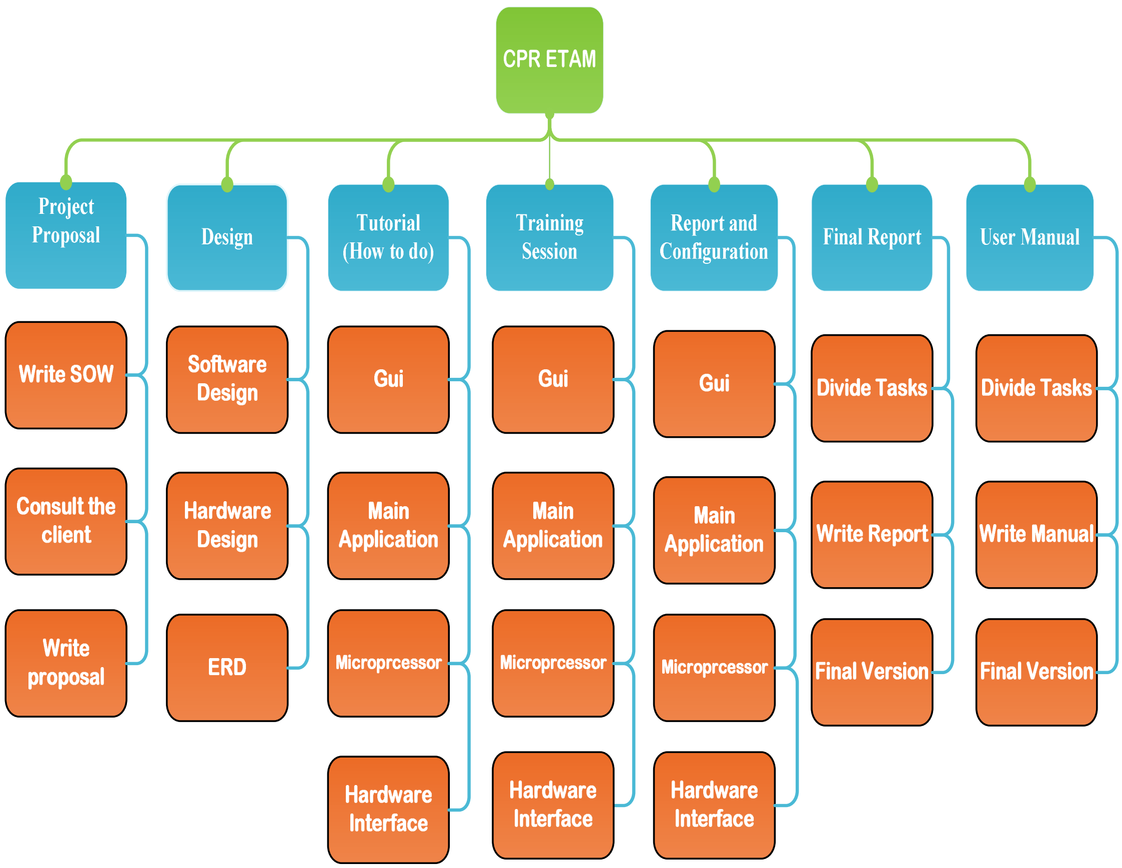
The schedule of this project is based on the three main tasks and deliverables. We divided the task into three areas: software, hardware and the microprocessor, which is in the middle of them. A list of deliverables has to be accomplishing in the dates that we will expose.*We expect to work 5 hours per day. Overtime can be optional if is required

	Deliverable	**Time (days)**	**Start**	**Finish**
1	Project Proposal	18	2/2/09	2/25/09
2	Progress Report	4	3/20/09	3/25/09
3	Tutorial (How to do)	21	2/11/09	3/11/09
4	Training	25	3/12/09	4/15/09
5	Report and Configuration	15	4/16/09	5/6/09
6	User Manual	3	5/7/09	5/11/09
7	Final Report	3	5/7/09	5/11/09
8	Total	75	2/2/09	5/11/09

Work Breakdown and Distribution

WBS Structure

The following structure has been populated with the tasks that have been assigned to our group to complete CPR e-TAM. It was divided into seven main tasks.



Distribution

The following diagrams display the division of tasks between each member of the team and the budget assigned to each resource. The details of the task are displayed in the Gantt chart. (See the Appendix for more information)

Overall

Deliverables	Days	Cost
Project Proposal	18	
Progress Report	4	
Tutorial (How to do)	21	
Training	25	
Report and Configuration	15	
User Manual	3	
Final Report	3	
Total	75	$38,104.40

Members of EHS

Jomar Rosario – Software Engineer

Deliverables	Days	Cost
Project Proposal	18	
Progress Report	4	
Tutorial (How to do)	21	
Training	22	
Report and Configuration	6	
User Manual	3	
Final Report	3	
Total	73	$8,940.05

Juan Gorritz – Software Engineer

Deliverables	Days	Cost
Project Proposal	19	
Progress Report	4	
Tutorial (How to do)	17	
Training	20	
Report and Configuration	11	
User Manual	3	
Final Report	3	
Total	73	$8,577.50

Edvier Cabassa -
Software Engineer

Deliverables	Days	Cost
Project Proposal	19	
Progress Report	4	
Tutorial (How to do)	9	
Training	19	
Report and Configuration	10	
User Manual	3	
Final Report	3	
Total	63.5	$8,255.00

Arelis Perez –
Hardware Engineer

Deliverables	Days	Cost
Project Proposal	19	
Progress Report	4	
Tutorial (How to do)	8	
Training	16	
Report and Configuration	5	
User Manual	3	
Final Report	3	
Total	56	$5,600.00

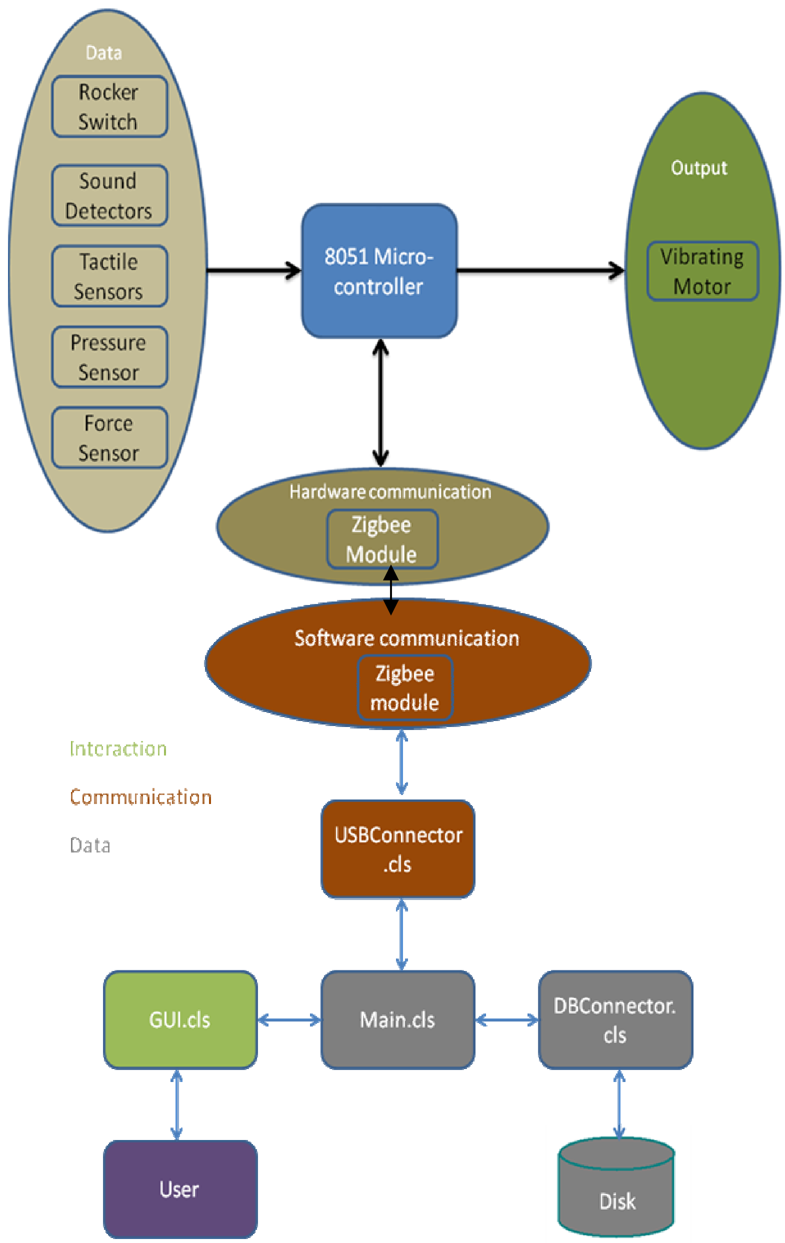
Jose Lombay –
Hardware Engineer

Deliverables	Days	Cost
Project Proposal	18	
Progress Report	4	
Tutorial (How to do)	8	
Training	21	
Report and Configuration	5	
User Manual	3	
Final Report	3	
Total	56	$6,729.40

Design Specifications

This part of the proposal will describe in more detail the two main components of our proposed design: hardware and software. The hardware will consist of the mannequin, the sensors, the microcontroller and the wireless communications device used to transmit the data to the computer. The software component will consist of a GUI developed in C# that will let the user to configure and control the mannequin from the computer. This GUI will be connected to a database created in MySQL language. These two main components will be described in more detail below.

Block Diagram



Hardware Specifications

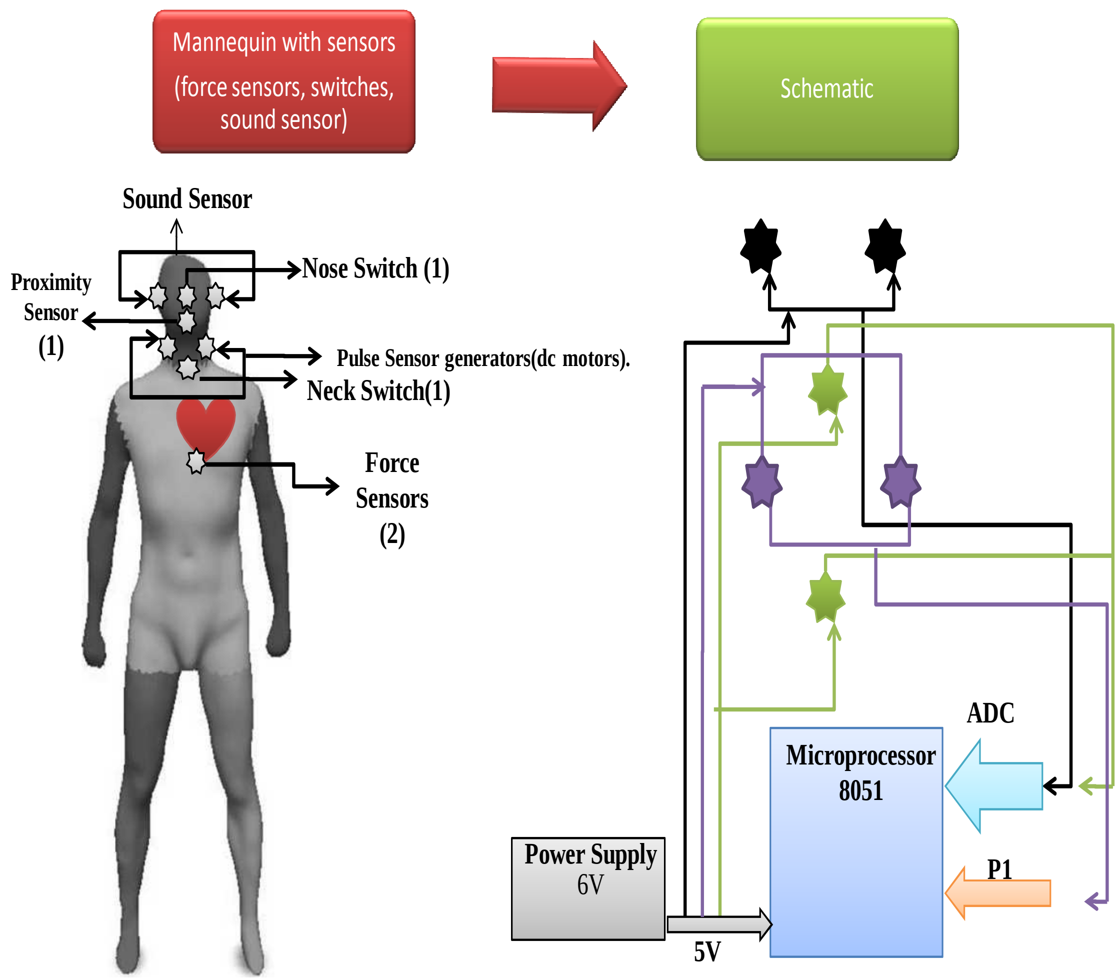
Hardware Considerations

- **Power Supply –** The system will be battery powered, and so we are going to use a series of 4 D batteries to provide a maximum amount of 6V. We will also integrate voltages regulators in order to satisfy all the power inputs necessary for the different devices.
- **Micro-Controller** – An 8051 micro-controller will collect the data of the different input devices strategically placed on the mannequin. It will contain software to verify that the correct procedure is being performed by the student. The input/output devices will be connected to the different ports of the controllers and a UART will be used to transmit information to a PC. We are going to use the C8051F040DK model.
- **Sound detector** (2) – An important part of the procedure is to verify if the victim is conscious. A way to do that is by talking to the victim. By using a sound detector, we will calibrate the sound detector to a threshold, which is the accepted volume of the speaker to verify if the victim is conscious or not. We are going to use the Index Sound Detector Board. It works with a built-in condenser microphone and can interface with microcontrollers and logic circuits. This sensor provides an output voltage from 0 to +5 Vdc which is proportional to the intensity of noise detected. If the noise is louder the voltage will increase and vice versa. This sensor will be calibrated by the system.
- **Rocker Switch** (1) –This switch will be placed on the neck of the mannequin to verify that the student tilted the head of the victim before listening for breathing. We are going to use the C&K SPDT Rocker Switch, model U11.
- **Tactile Sensors** (4) –These sensors would be adequate to confirm if the student verified the pulse of the victim. We are going to use the Interlinks Electronics 1.5” Square Force Sensing Resistor. It is optimized for use in human touch control applications, it uses a 1.5” square pad to sense applied force.
- **Vibrating Motors** (4) –It will be used to simulate the pulse of a person in the mannequin’s neck. We are going to use the Vibrating Disk Motor that gives 1G vibration at 12,000rpm from 3V, drawing less than 80mA.
- **Pressure Sensor** (2) –It will be connected to the mouth of the mannequin to verify that the correct amount/volume of air is being from the student. With it, we can measure if too much air is being provided. We are going to use the Dallas Semiconductor DS18B20+ Temperature Sensor, ideal for interfacing to a microcontroller. It is powered from 3 to 5.5V and measures -55 to +125°C (-67°F to +257°F). This sensor will be calibrated by the system.
- **Force Sensor** (5) – During the CPR process, the student will have to provide chest compression to the mannequin. In real life, if this is not done

correctly, you can produce more harm to the victim than what you would be helping. The sensor would help us to measure the amount of force applied in the chest of the mannequin. We are going to use a FlexiForce Sensor, A201 Model. It works with force ranges of 0-1 lb. for low, 0-25 lbs. for medium and 0-100 lbs. for high, and with supply voltage of 9V. This sensor will be calibrated by the system.

- **Zigbee Modules** (4) –Wireless Modules 2.4Ghz XBee-PRO ZB w/Wired-Whip-AT Router F/W. Two modules will be used in the system. One of them will be connected to the micro-controller, at the UART, while the other one will be connected to a USB to Serial connector which will provide the communication to the computer. The communication between the two modules is wireless.
- **USB to Serial connector** (4) –Will be used to facilitate the communication between the computer and the micro-controller. We are going to use the Industrial USB Connectors and Cables VS-04-BUB-FK-F/IP67 FEM INSERT USB B, 1653867 models.

Hardware Design Mannequin



(See the Appendix for more information)

Software Specifications

The e-TAM software application will be intended for the Windows operating system. It will be written in C# through Visual Studio 2008, due to its ease of use when programming user interfaces. It will only be used by instructors, but will require authorization due to the sensitive nature of the data being stored and retrieved, which will be handled with a simple login pop up window when attempting to access information; an authenticated user’s status will remain throughout the system until the user logs out or the program is closed. The software application will start with a splash screen with a variation of the company logo and the name of the program while the program loads.

High-level software module descriptions

The system refers to the program as a whole, including the database and file creation/editing by said system. High level refers to the level which the user will actively manipulate. The majority of the system architecture will be event-based, with the added functionality of a database-centric architecture for maintaining instructor, student, and course information. The main interface will have an option to create an account which will be saved to the database, or if internet is not available, to the database file on the hard drive using C# encryption methods.

- Tutorials
    - This module is responsible for educating the user to achieve full understanding of the CPR process and the program's functionality. Sub-modules will display interactive training videos for the selected option in a new window
        - Perform CPR: Will give a detailed, graphical description of all the steps necessary for a successful CPR session
        - Accessing results: Will show how to use the program to gain access to previous results stored either on the computer or on the database and print them to a file, or, using C#'s built-in methods, straight to a printer.
        - Add courses & students: Will show how to add a course to the system, as well as populate it with the students enrolled in it
- Training
    - This module contains the core functionality of the software application, and replaces the main interface with a new training interface. The main interaction between user and system occurs within the three important sub-modules:
        - View/Edit courses – Activated when the instructor is logged on. When selected, a new screen will appear showing the added courses, sections, and students, an add button, an edit/remove button, and a modify student button
            - Add course: add a course with course information such as course name, code, number of enrolled students, and section
            - Edit/remove course: will allow the instructor to change course information, or, if all fields are left blank, will remove the course from the system
            - Modify students: will be used to add students to a course, and modify student information, such as enrolled section. If all fields left blank, the student will be removed from the system
        - Begin training session: Enabled regardless of whether or not the instructor is logged on. When selected, a new window will appear asking for options to enable grading, live-preview mode, and to begin the session. The event-driven architecture comes into play here more than anywhere in the system, as this is where the program will react to the events, being the responses from the sensors

        - Enable/disable grading: If enabled, the grading profile will be retrieved to evaluate the student as he or she is performing CPR. The steps will be compared to the profile, and points will be deducted accordingly in a new window, where the instructor can see the results in real time.
        - Enable/disable live-preview mode: Live preview mode is a feature that will display an animation of the current step taking place on the mannequin. When enabled, the pictures will be retrieved as the session progresses. Steps will be generalized, not detailed (i.e. Compressions vs. Compression 23/30, 24/30, 25/30 etc)
    - View past results: Allows the instructor to view group or individual results of the selected course & section in tabular form by session. When selected, the results window will open in a new window
        - Get group results: Will display the entire group of students in the section in a table. Individual grades per session will be linked to more detailed results shown in individual mode
        - Get individual results: Displays results for a single student in a course throughout the course by session. Session details will be accessible through the grade for each session showing more detailed information, such as % of compressions, common errors, etc.
- Configuration
  - This module is responsible for modifying program settings for the grading profiles and default save locations. If selected, the main interface will replaced with a configuration interface.
    - Modify grading profiles: Used to create, edit, or delete a new grading profile
    - Modify save location: Will allow the instructor to choose where grading profiles are saved.

Low-level software module descriptions

Low-level refers to the part of the software that will be hidden from the user. Encryption/decryption will be handled using C#’s cryptography classes.

- USB connector
    - This module will be used to communicate with the wireless USB device. It will have methods that will encode and decode the information coming and going.
        - Decode data: The information that comes from the wireless USB device will need to be translated to be understood correctly by the program, this component will be responsible for decoding the data.
        - Encode data: This section will be responsible for putting the instructions in a format readable by the wireless USB device
        - Send/Retrieve data: Will be used either to send or retrieve data to or from the USB device
- Database connector –
    - This module will allow the instructor to make changes to the database that contains all the information about his/her students, courses and evaluation sessions.
        - Modify course: The instructor will be able to add more courses to the database, delete one of them or change the information of one of the courses, like session number, amount of students, etc.
        - Modify students: This part of the module will change the database information regarding the students the instructor has. He /she will be able to remove students from previous semesters, add new students every semester or change student information, like student number, session, etc.
        - Update results: This part of the module will allow the instructor to record the results of an evaluation session of CPR per student. For example, it will log the results that the student had on chest compressions, head tilt check, pulse check, breath check and other grading aspects that the instructor would like to verify during the CPR session. This part will also update the last grade the student achieved, adding the new results to the evaluation sessions.
- Hard disk connector
    - This module will be responsible for database manipulation on the hard drive. It will control data access in the case that an internet connection is not available.
        - Encryption/decryption: The database file will be encrypted before saving and decrypted upon reading using C#’s cryptography classes
        - Database access: After decryption of the database file, access will be made similar to the methods used in the Database connector module.

Risk Management and Considerations

Risk Management Plan

Absence of a team member

Mitigation

One resource will be assigned to each task, but another resource will have knowledge of its specifics.

Monitoring

Absence must be notified 24 hours in advance. A signed list (paper or electronic) will be in the work area to monitor assistance to the lab.

Management

If the assigned task is not critical it will be held until the member of the team returns. However, if it is a critical task it will be done by the other people that has knowledge or that person will explain the task to another member that is not working on a critical task to handle the task of the absent member.

Prolonged absence of a team member due to illness or treatment

Mitigation

One resource will be assigned for the responsibility of each task, but another resource will have knowledge of its specifics. The person can work from home and meet electronically through web conferencing software.

Monitoring

Notification of prolonged absence will be required in advance. Constant communication with the affected person will be maintained.

Management

If the assigned task of the affected person is not critical it will be held off until the member of the team returns. However, if it is a critical task it will be done by the other people that have knowledge of its specifics or he will explain the task to another member that is not working on a critical task to do the task of the absent member. If the person will be out more than 2 days,he or she will work from home if the condition allows. Also, if necessary, the team will bring the requisite materials to their location.

Delay in Hardware components shipping

Mitigation

The shopping of the components will take place before its assigned date to prevent a delay in the project due to some delay in shipping.

Monitoring

Monitor the shipping by internet or phone.

Management

If the components don’t arrive in time the resource in charge of the task that involves said components will be assigned to another task until the component arrives or the other task assigned is finished.

Unscheduled and unexpected academic recess

Mitigation

A team member will be assigned to have the official campus schedule and be aware of any meeting, bulletins, rumors, or news that lead to an uncommon event like a strike, lack of water or electricity on campus, etc.

Monitoring

This team member will be continuously verifying the schedule to see if one of these unexpected recesses affects the development of the project.

Management

The project manager will reschedule the events and/or meetings and will reassign a place to meet, in case the campus is affected by this recess.

Hardware component breaks in the development process

Mitigation

A hardware team member will be assigned to take care of all the hardware pieces. Any team member that needs to use one of these pieces will ask this team member for his/her approval.

Monitoring

The team member assigned to take care of the hardware will inspect all the hardware pieces (visually and electronically) every week (or other amount of time), in order to verify that all of them are working well and have no problems.

Management

The team member will discard the broken part or will replace that part with one of the backup parts. If necessary he/she will order a new one in case no backup parts exist. Also the team member will notify of the event to the PM.

Contractual Aspect

EHS group will be generating progress reports in the due dates scheduled by the capstone professors. The progress reports will contain information about the completed tasks, missing tasks, accountability, and any changes that may have occurred during the duration of the project. Two preliminary prototypes have to be shown in oral exams after the final prototype.

Legal Considerations

The whole purpose of this project is to facilitate the trainer's task in giving and evaluating CPR training sessions to students, it is not intended that this product replace the RED CROSS CPR certification, the trainers or any other methods for teaching CPR, the use of this product does not imply that you are certified to give CPR.

In order for this product to be usable in the United States all the radio and telecommunications components have to be FCC compliant. The component list of the project has been revised to have FCC compliant Zigbee modules.

Environmental Issues

After an evaluation of possible environmental issues due to the design and creation of this project, we agree that there are no issues, besides the disposal of the mannequin with the sensors. In this case, it is considered the responsibility of the person who bought the mannequin to dispose it in a green way. Also, since the product will use wireless communication, any radio and telecommunications components have to be FCC compliant. The component list has been revised to insure that these standards are met.

Budget

Human Resources

For the development of CPR e-TAM, EHS Group has three Level I Software Engineers and two Level I Hardware Engineers. The work of the project begins on January 14, 2009 and finishes on May 14, 2009. The following table describes the total cost that each engineer will earn.

Employees	***Position***	***Dollars/Hour***	***Days/Contract***	***Payment/ Contact***
J. Rosario	Software Engineer I	$24.50	73	$8,942.05
E. Cabassa	Software Engineer I	$26.00	63.5	$8,255.00
J.Gorritz	Software Engineer I	$23.50	73	$8,577.50
A.Pérez	Hardware Engineer I	$20.00	56	$5,600.00
J.Lombay	Hardware Engineer I	$24.25	55.5	$6,729.40
		Employment Cost		$38,104.40
		Unemployment Insurance (1.40%)		$533.46
		Retirement (15%)		$571.57
		State Insurance Fund (1.55%)		$674.45
		Social Security (6.20%)		$2,362.47
		Medicare (1.45%)		$552.51
		Total Employment Cost		$42,798.90

Table 1: Human Resources Cost

Hardware Components

For the hardware development of the CPR Mannequin, the following parts and components are needed:

Component	Model	Price	Quantity	Total Cost
Proximity Sensor	308S	$12.95	2	$25.90
Pressure Sensor	DS18B20+ Temperature Sensor	$4.95	2	$9.90
Force Sensor	FSM101	$0.43	5	$2.15
Vibrating Sensor	VPM2	$3.52	3	$10.56
Tactile Sensor	RB-Int-03	$8.75	3	$26.25
Neck Switch	U11	$1.00	1	$1.00
Microprocessor	Intel 8051	$179.00	1	$179.00
Mannequin		$70.00	2	$140.00
Total Hardware Cost				$394.76

Table 2: Hardware Components Cost

Software Components

Program	Price	Quantity	Cost
Visual Studio 2008	$236.00	3	$708.00
MySQL Server	$0	2	$0
My SQL Tools	$0	2	$0
Gimp	$0	1	$0
	Total Software Cost		$708.00

Table 3: Software Component Cost

Overall

This table contains the total cost of the project described by each part that it consists of:

Component	Cost
Total Employment Cost	$42,798.90
Total Hardware Cost	$394.76
Total Software Cost	$708.00
Project Cost	$43,901.60
Overhead	$48,291.80
Total Project Cost	$92,193.40

Table 3: Overall Project Cost

Assessments

In order to provide a tool for us to distribute and monitor tasks, we will use Jira dashboard. Each resource has access to the project to modify his or her tasks, and notifications are available as new tasks arise and as tasks are completed, therefore we will be able to track our team's progress. In the software module, we have identified issues and assigned them to the responsible resources in order to prevent bugs. In order to minimize errors, a C# coding standard has been adapted which will produce more readable code, facilitating integration of other programmers. Unit testing will be done using the NUnit framework, which will be performed after the creation of each class to verify that it works as it should. Also, a logging class has been created with a scale index for tracing program progress while running. Test cases will be tabulated and, as a requirement, will need to be observed and approved by the client. Code review will assure that new code is written following the coding standard and that it is well organized and properly documented.

Measurements for success

When all of the goals are met for each phase, it will be considered successful:

I. First Phase
   a. Software – GUI prototype, hardwired database interaction
      i. Contain the frames (windows) that will be used on the software application to demonstrate usability and features.
      ii. "How to" sections about how to operate the system.
      iii. A database should be created and show data interaction with GUI (hardwired information).
   b. Hardware
      i. Basic connections to switches, in order to simulate usage of the port and ADC.
      ii. Should have ordered the manikin that will be used for the project.
      iii. Should have a list of the components that will be ordered in order to complete the second phase.

II. Second Phase
   a. Software – Completed GUI, full database interaction, micro communication
      i. Imitate live-reading of results using streaming data
      ii. One-way communication with the micro-controller should be possible
      iii. There should be communication from the application to the database, using data created by the user.
   b. Hardware
      i. The hardware components should be integrated with the manikin.
      ii. One-way communication with the application should be made.

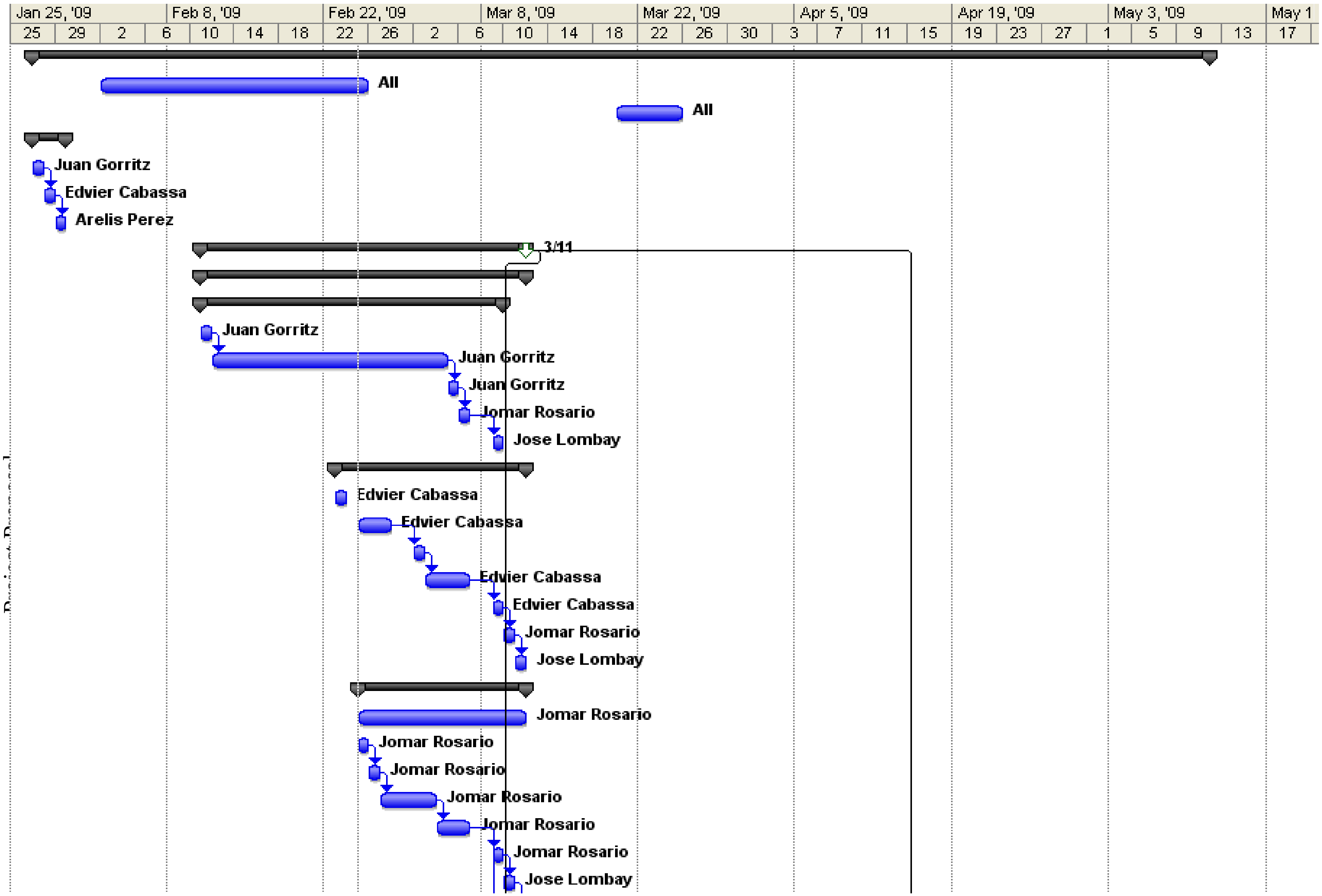
III. Third Phase
   a. Software – Micro communication, GUI configuration, CPR grading
      i. Two-way communication with the micro-controller
      ii. Configuration should be done by utilizing the application
      iii. Should be able to monitor a simple cycle of resuscitation.
      iv. A cycle of resuscitation should be graded utilizing the configuration chosen by user.
   b. Hardware
      i. Two-way communication with the application
      ii. Should be accepting configuration parameters from the application

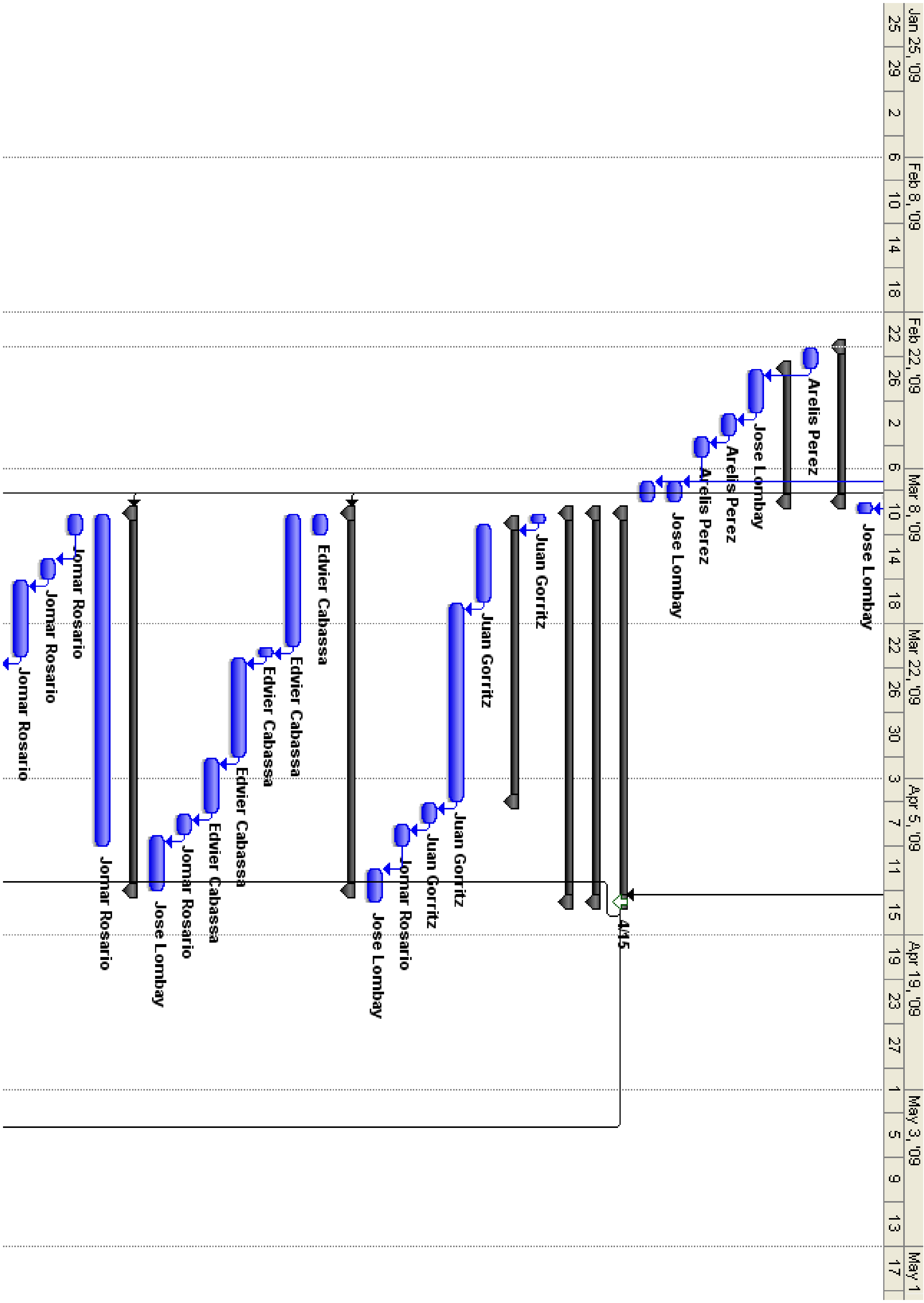
iii. Should run a cycle of resuscitation.

IV. Final
  a. Software & Hardware
    i. A fully completed mannequin that successfully connects to the application and does everything proposed.

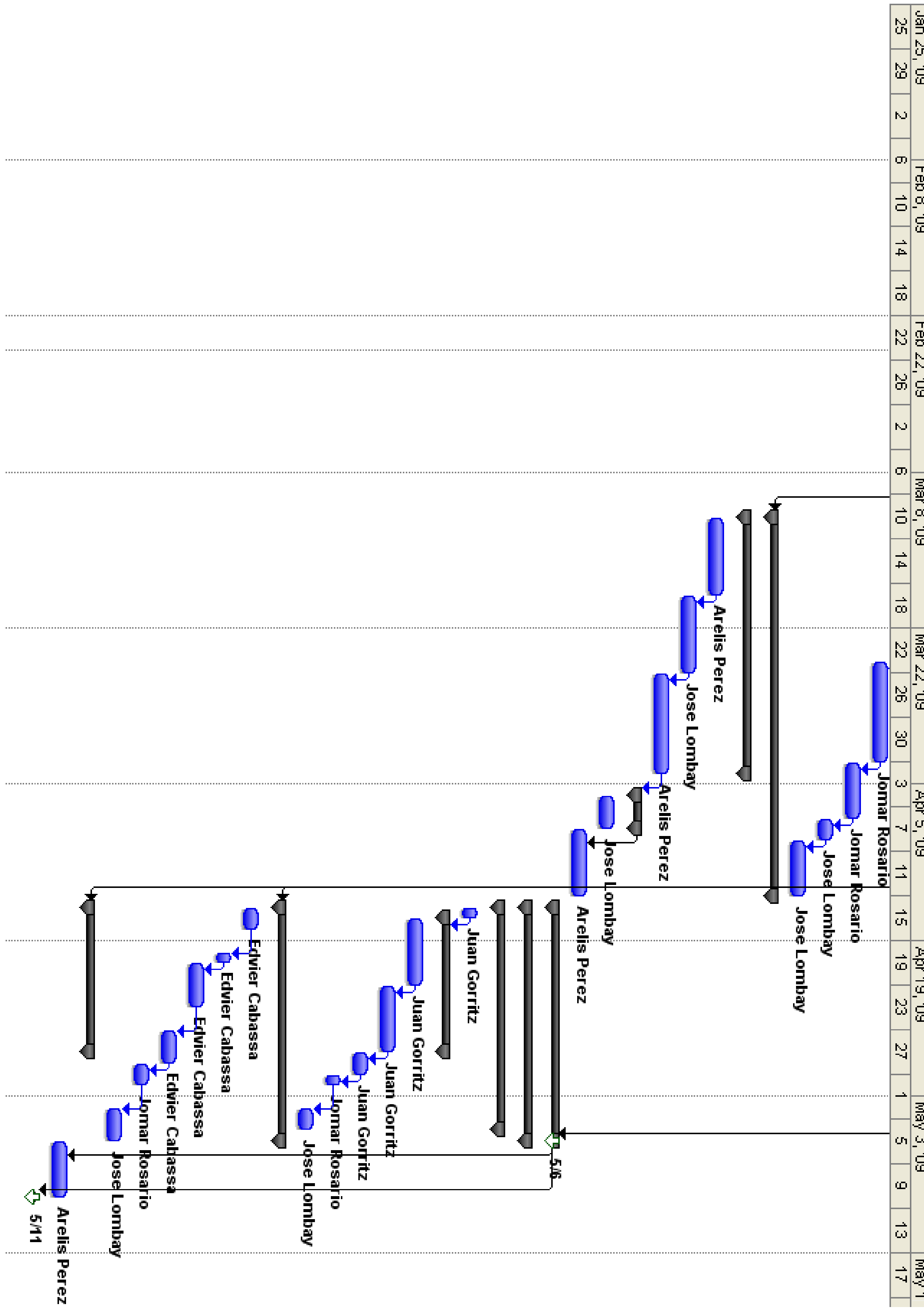
Gantt chart

To create the Gantt chart we used Microsoft Project 2003. This software provides useful tools to manage the project assigning tasks to the resources and deadlines to accomplish them.


Jan 25, '09
Feb 8, '09
Feb 22, '09
Mar 8, '09
Mar 22, '09
Apr 5, '09
Apr 19, '09
May 3, '09
May 1
25 29 2 6 10 14 18 22 26 2 6 10 14 18 22 26 30 3 7 11 15 19 23 27 1 5 9 13 17
All
All
Juan Gorritz
Edvier Cabassa
Arelis Perez
3/11
Juan Gorritz
Juan Gorritz
Juan Gorritz
Jomar Rosario
Jose Lombay
Edvier Cabassa
Edvier Cabassa
Edvier Cabassa
Edvier Cabassa
Jomar Rosario
Jose Lombay
Jomar Rosario
Jomar Rosario
Jomar Rosario
Jomar Rosario
Jomar Rosario
Jomar Rosario
Jose Lombay


Jan 25, '09
Feb 8, '09
Feb 22, '09
Mar 8, '09
Mar 22, '09
Apr 5, '09
Apr 19, '09
May 3, '09
May 1
25
29
2
6
10
14
18
22
26
2
6
10
14
18
22
26
30
3
7
11
15
19
23
27
1
5
9
13
17
Arelis Perez
Jose Lombay
Arelis Perez
Arelis Perez
Jose Lombay
Jose Lombay
Juan Gorritz
Juan Gorritz
Juan Gorritz
Juan Gorritz
Jomar Rosario
Jose Lombay
4/15
Edvier Cabassa
Edvier Cabassa
Edvier Cabassa
Edvier Cabassa
Edvier Cabassa
Jomar Rosario
Jose Lombay
Jomar Rosario
Jomar Rosario
Jomar Rosario
Jomar Rosario

Project Proposal
CPR Electronic Teaching Assistant Mannequin



Personnel Information

Juan C. Gorritz Garcia

Juan C. Gorritz is a computer engineering undergraduate specializing in computing systems, and will be responsible for the software application development. Relevant courses include Programming, Advanced programming, Data structures, Programming languages, Database systems, and Operating systems. His main interest area is the area of usability of software. He has experience through a variety of projects, including designing an e-Commerce website, a microprocessor simulator, experience as a webmaster, and community service presentations. In all of these the main purpose was to present information to the user in the most usable way possible. His programming experience includes C, C#, Java, JSP, HTML, CSS, Javascript, MySQL.

Edvier Cabassa Miranda

Edvier Cabassa is a Computer Engineer from the University of Puerto Rico at Mayagüez. He has programming experience in JAVA, C, MATLAB, IDL, HTML, Javascript, PHP and Database systems (MySQL). He also has experience in the design of web pages using HTML and CSS. He had worked on university projects like design a 16 bit microprocessor, made an airline reservation web page with a database, and implemented a Java Operating System. At this time he is working with the project NOAA CREST at UPRM. In 2008, he worked for the CSOIS MAS-net REU program at Utah State University in a project that involved a MATLAB GUI and real time wireless sensor network data. In 2007 he worked with AT-Sciences, in collaboration with the Human Engineering Research Lab (HERL) and University of Pittsburgh. He was responsible to developed, designed and tested a new drop off sensor for an autonomous power wheel chair. In the summer of 2008, he was invited to present a poster about this work, and he was awarded with 4 other students with an honorable mention award. For the summer of 2006 he worked with the Arecibo Observatory in collaboration with UPRM. He worked with Dr. Julio Urbina to improve an algorithm wrote in IDL. Scientific papers he has written: Development and Testing of a New Drop-Off Detection System for Powered Wheelchairs, and collaborated in many others.

Jomar Rosario Muñiz

Jomar Rosario is a Computer Engineer from the University of Puerto Rico, Mayaguez Campus with an emphasis in software engineering. He worked with the Professor Jimenez and CPES in the research of the Parasitic Estimation Algorithm and published a paper in Virginia Tech University. He has worked with Medtronic in the Software Development Department, where he developed his skills and promoted the usage of a new programming language in the entire company.

He has experience using these languages: C, C++, C#, Java, JSP, HTML, VBA, VB, assembly. He also has experience with microprocessors and computer architecture, Oracle and SQL databases. In EHS he will lead the team as Project Manager and will be in charge of the programming and interface of the microprocessor.

Arelis Perez Peña

Arelis D. Pérez is a Computer Engineering student from the University of Puerto Rico at Mayaguez. She has taken Electronics II, Analogical Circuit Design and Digital Signal Processing and has experience in programming languages including Java and C++. She is taking courses to complete the Project Manager and Business Administration Certifications. She did a six month research in the debugging process developing a program in Pascal capable of eliminating Pin Table errors when testing a device using the VLCT Tester.

Jose A. Lombay Gonzalez

Jose Lombay is a Computer Engineering student at the University of Puerto Rico, Mayaguez Campus. He is specializing in the Computing System area. He did a one-year long research at UPRM in the Nanoimprint Lithography. Also, he was part of a “Research Experience for Undergraduate” at Virginia Tech for 10 weeks, in the area of Human Computer Interaction. Due to his work during that summer, he was able to represent both universities at the “Richard Tapia’s Diversity in Computing Conference”. For the last six months he was working at *National Aeronautics and Space Administration,* in the *Flight Software Engineering branch.* He has experience in a variety of programming languages, such as C, C++, SQL, PHP, and Java. He also is experienced in the hardware area, since he took the Microprocessor 2 course and was part of a group who developed a blood pressure monitor device.

Legal Requirements

We will be receiving consultation from a CPR professor from the Physical Education Department at the University of Puerto Rico, Mayaguez Campus. As part of our agreement with him, he will be providing his support during the process and happy hours, and we will provide our final product, for use in his course.

It must be clarified that this product in no way, shape, or form intends on replacing conventional, professional teaching of CPR procedures. Its only intention is the facilitation of teaching and evaluating CPR students. Proper use of the product does not imply that the user will be certified to give CPR.

It falls under the responsibility of the instructor to take into consideration the legal aspects concerning CPR instruction and performing CPR.

References

[1] Thygerson, A. (2005). *First aid, CPR, and AED, 4th ed.* Jones and Bartlett: Sudbury, Massachusettes.

[2] National Safety Council. (1999). *Heartsaver facts: First aid, AED, CPR training system.* Jones and Bartlett.

[3] National Safety Council. (1997). *CPR for the professional rescuer*. Jones and Bartlett.

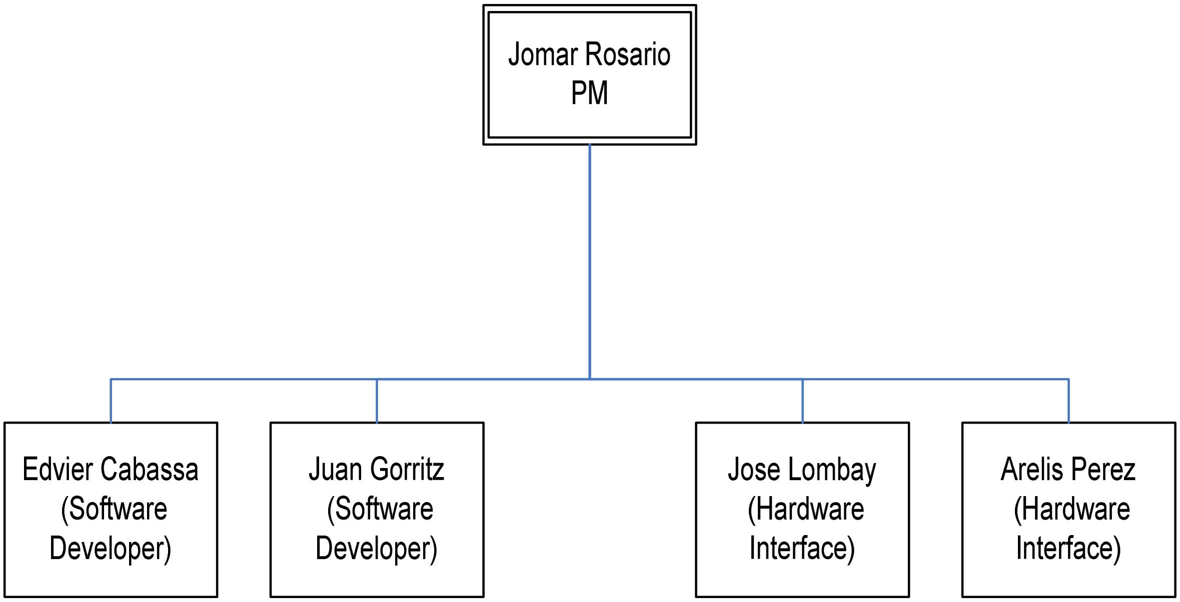
[4] Jones, A. (2003). *C# for Java developers*. Microsoft Press.

[5] Microsoft Corporation. (2002). *Microsoft visual C# .NET: Language reference*. Microsoft Press: Redmond, WA.

[6] Obviex. (2009). *How to: Encrypt and decrypt data using a symmetric (Rijndael) Key (C#/VB.NET).* Retrieved February 10, 2009 from http://www.obviex.com/samples/Encryption.aspx#

Appendix

Organization Chart



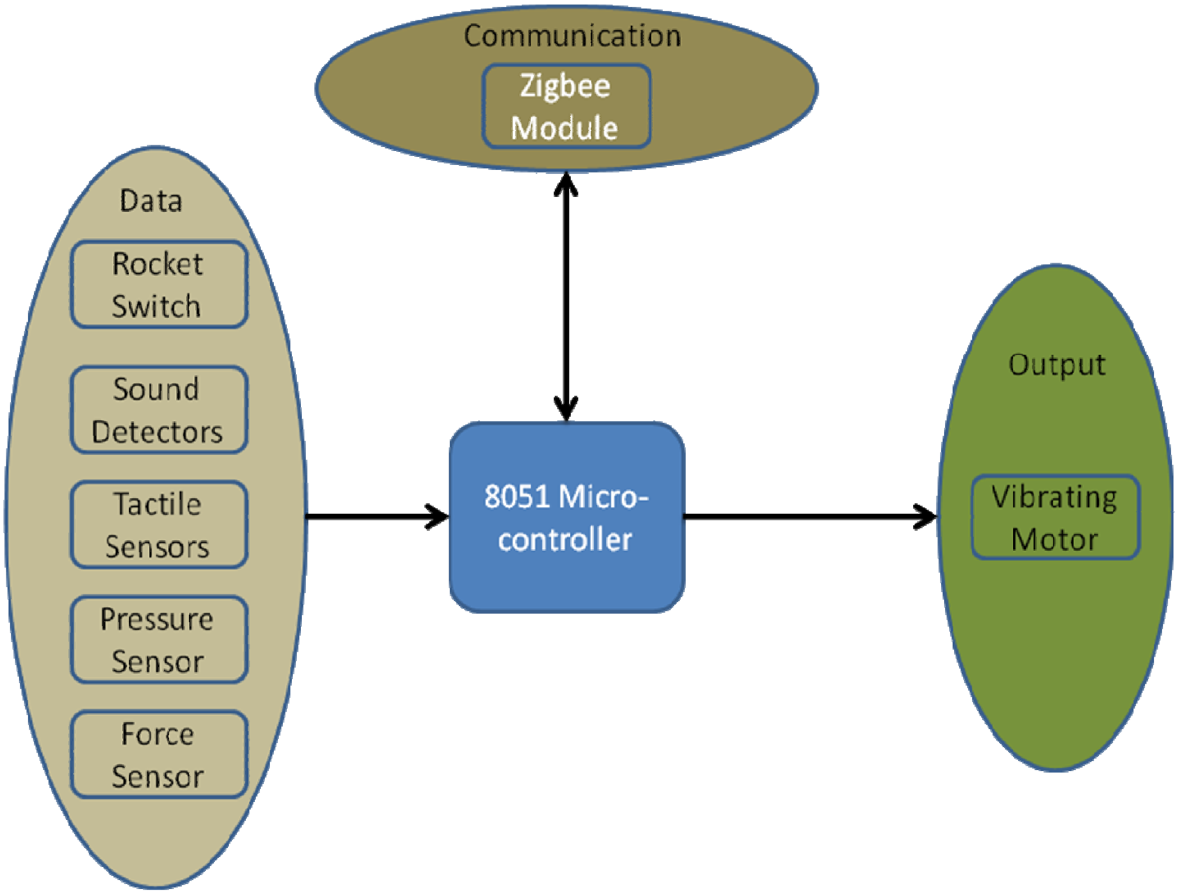
CPRETAM Work Breakdown

Tasks	Time	Responsible Party	Seq. No.
CRPETAM	75 days		1
Proposal	18 days	All	1.1
Progress report	4 days	All	1.2
Design	3 days		1.3
Software Design	1 day	Juan Gorritz	1.3.1
ERD	1 day	Edvier Cabassa	1.3.2
Hardware Design	1 day	Arelis Perez	1.3.3
Tutorial (How to do)	21 days		1.4
Software development	21 days		1.4.1
GUI	19 days		1.4.1.1
Frames design	1 day	Juan Gorritz	1.4.1.1.1
Frames development	15 days	Juan Gorritz	1.4.1.1.2
Debugging	1 day	Juan Gorritz	1.4.1.1.3
Code Review	1 day	Jomar Rosario	1.4.1.1.4
Testing	1 day	Jose Lombay	1.4.1.1.5
Main Application	13 days		1.4.1.2
Log class	1 day	Edvier Cabassa	1.4.1.2.1
Serial Port Handler class	3 days	Edvier Cabassa	1.4.1.2.2
Serial Port Event Handler	1 day		1.4.1.2.3
Communication Protocol class	4 days	Edvier Cabassa	1.4.1.2.4
Debugging	1 day	Edvier Cabassa	1.4.1.2.5
Code Review	1 day	Jomar Rosario	1.4.1.2.6
Testing	1 day	Jose Lombay	1.4.1.2.7
Microprocessor Software Development	11 days		1.4.1.3
Main subroutine	11 days	Jomar Rosario	1.4.1.3.1
Assign pins and ports to sensors	1 day	Jomar Rosario	1.4.1.3.2
Pins input/output	1 day	Jomar Rosario	1.4.1.3.3
ADC input	3 days	Jomar Rosario	1.4.1.3.4
UART (micro to pc)	3 days	Jomar Rosario	1.4.1.3.5
Debugging	1 day	Jomar Rosario	1.4.1.3.6
Code Review	1 day	Jose Lombay	1.4.1.3.7
Testing	1 day	Jose Lombay	1.4.1.3.8
Hardware interface	10 days		1.4.2
Power Supply	2 days	Arelis Perez	1.4.2.1
Sensors	8 days		1.4.2.2
Pulse switch	2 days	Jose Lombay	1.4.2.2.1
Head position switch	2 days	Arelis Perez	1.4.2.2.2
Nose switch	2 days	Arelis Perez	1.4.2.2.3

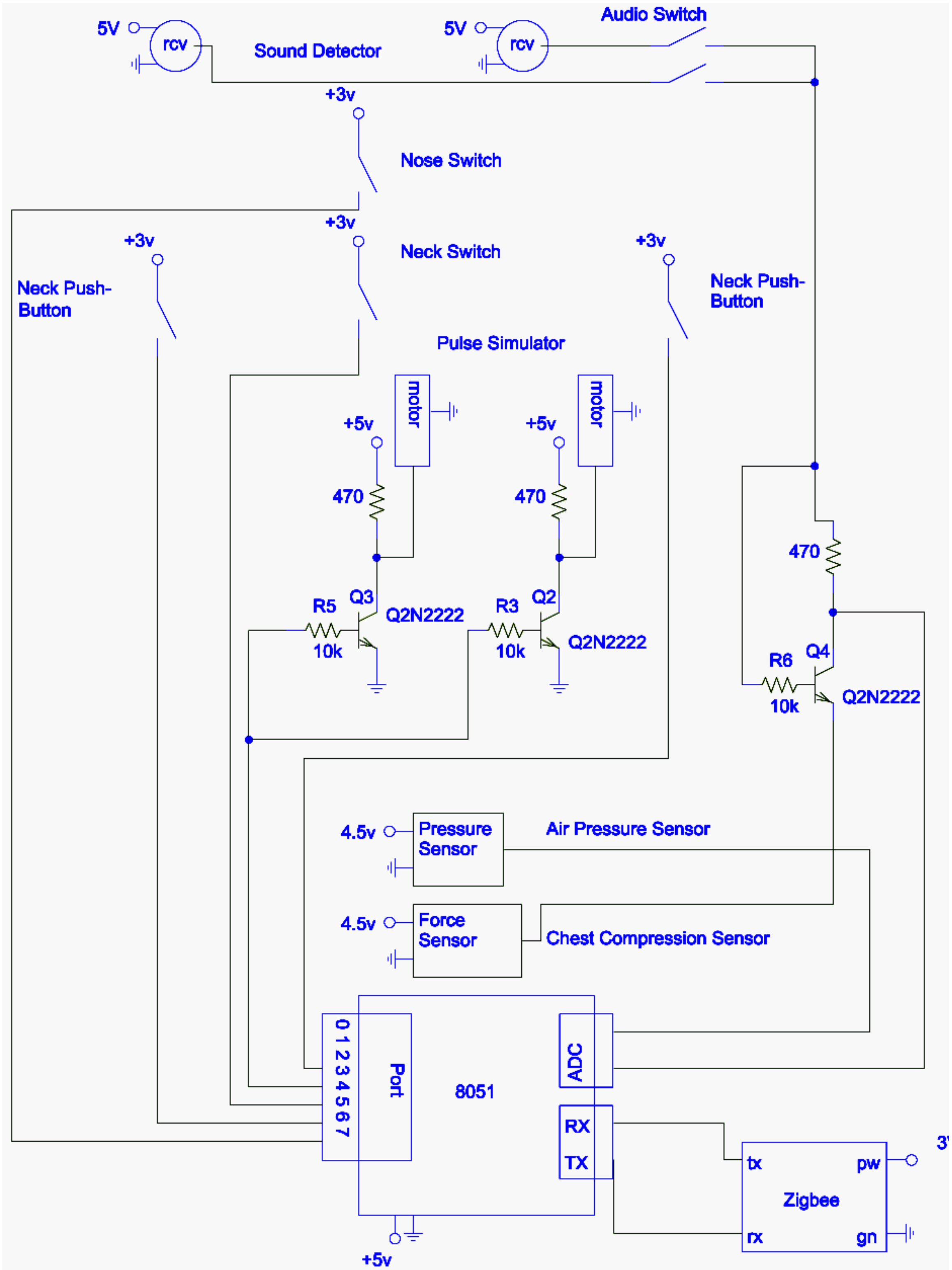
Breathing (pressure sensor)	2 days	Jose Lombay	1.4.2.2.4
Testing	2 days	Arelis Perez	1.4.2.2.5
Training	25 days		1.5
Software development	25 days		1.5.1
GUI	25 days		1.5.1.1
Frames design	1 day	Juan Gorritz	1.5.1.1.1
Frames development	17 days		1.5.1.1.2
View courses	5 days	Juan Gorritz	1.5.1.1.2.1
Begin training procedure	12 days	Juan Gorritz	1.5.1.1.2.2
Debugging	2 days	Juan Gorritz	1.5.1.1.3
Code Review	2 days	Jomar Rosario	1.5.1.1.4
Testing	3 days	Jose Lombay	1.5.1.1.5
Main Application	24 days		1.5.1.2
DB connector class	2 days	Edvier Cabassa	1.5.1.2.1
DB Queries class	8 days	Edvier Cabassa	1.5.1.2.2
Serial Port Event Handler	1 day	Edvier Cabassa	1.5.1.2.3
Communication Protocol class (cont.)	7 days	Edvier Cabassa	1.5.1.2.4
Debugging	3 days	Edvier Cabassa	1.5.1.2.5
Code Review	2 days	Jomar Rosario	1.5.1.2.6
Testing	3 days	Jose Lombay	1.5.1.2.7
Microprocessor Software Development	24 days		1.5.1.3
Main subroutine (cont.)	22 days	Jomar Rosario	1.5.1.3.1
Assign pins and ports to sensors (cont.)	2 days	Jomar Rosario	1.5.1.3.2
Pins input/output (cont.)	2 days	Jomar Rosario	1.5.1.3.3
ADC (cont.)	5 days	Jomar Rosario	1.5.1.3.4
UART (pc to micro)	7 days	Jomar Rosario	1.5.1.3.5
Debugging	3 days	Jomar Rosario	1.5.1.3.6
Code Review	2 days	Jose Lombay	1.5.1.3.7
Testing	3 days	Jose Lombay	1.5.1.3.8
Hardware Interface	24 days		1.5.2
Sensors	17 days		1.5.2.1
Tactile Sensors (neck)	5 days	Arelis Perez	1.5.2.1.1
Sound detectors	5 days	Jose Lombay	1.5.2.1.2
Force Sensor (chest compression)	7 days	Arelis Perez	1.5.2.1.3
Simulators	3 days		1.5.2.2
Pulse motor	3 days	Jose Lombay	1.5.2.2.1
Testing	4 days	Arelis Perez	1.5.2.3
Configuration and Report	15 days		1.6
Software development	15 days		1.6.1
GUI	14 days		1.6.1.1
Frames design	1 day	Juan Gorritz	1.6.1.1.1

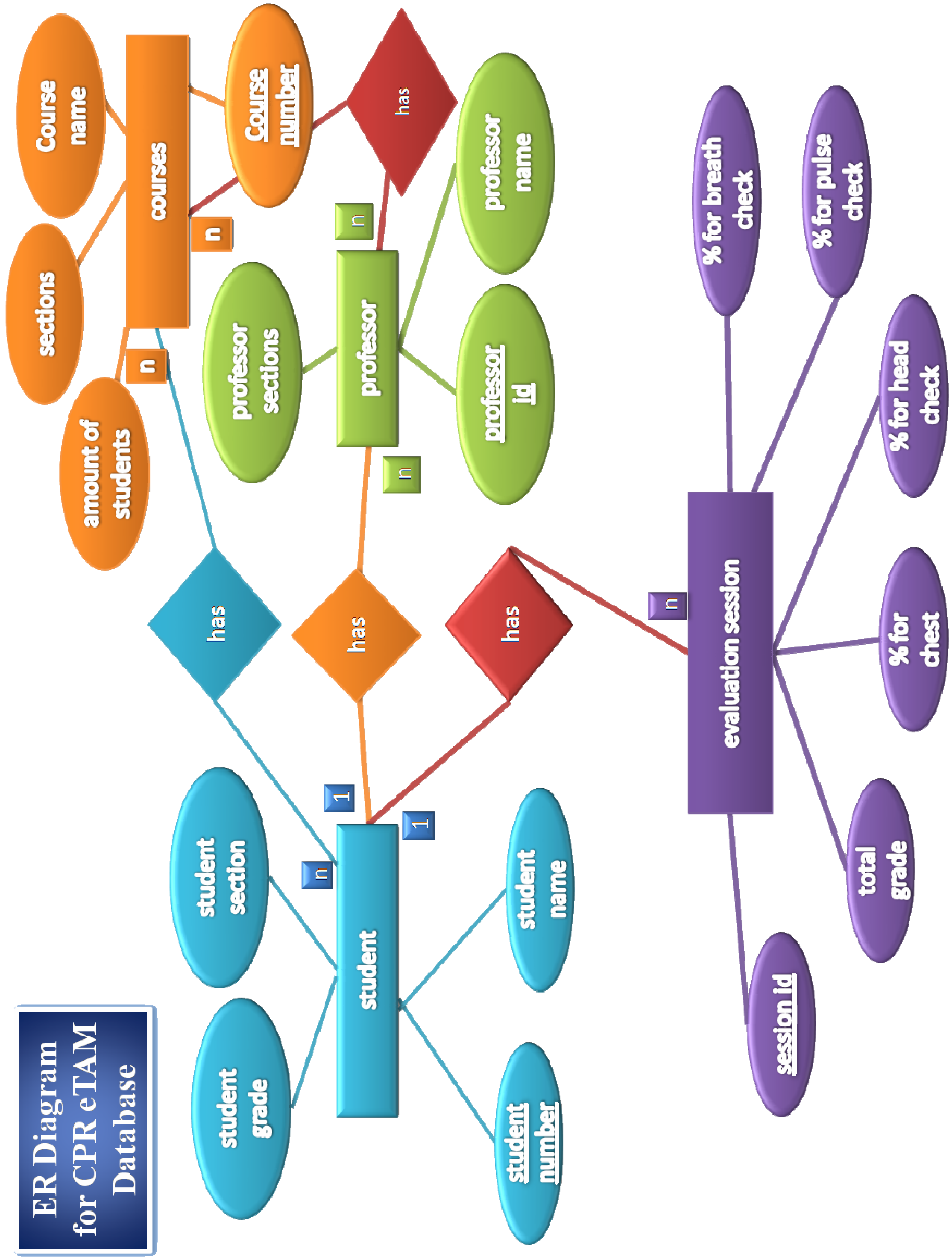
Frames development	8 days		1.6.1.1.2
Configuration frame	4 days	Juan Gorritz	1.6.1.1.2.1
View Results	4 days	Juan Gorritz	1.6.1.1.2.2
Debugging	2 days	Juan Gorritz	1.6.1.1.3
Code Review	1 day	Jomar Rosario	1.6.1.1.4
Testing	2 days	Jose Lombay	1.6.1.1.5
Main Application	15 days		1.6.1.2
DB Queries class (cont.)	2 days	Edvier Cabassa	1.6.1.2.1
Serial Port Event Handler	1 day	Edvier Cabassa	1.6.1.2.2
Communication Protocol class (cont.)	4 days	Edvier Cabassa	1.6.1.2.3
Debugging	3 days	Edvier Cabassa	1.6.1.2.4
Code Review	2 days	Jomar Rosario	1.6.1.2.5
Testing	3 days	Jose Lombay	1.6.1.2.6
Microprocessor Software Development	3 days		1.6.1.3
UART (pc to micro)	3 days	Jomar Rosario	1.6.1.3.1
Hardware Interface	2 days		1.6.2
Wireless Communication	2 days		1.6.2.1
Zigbee installation	2 days	Arelis Perez	1.6.2.1.1
Testing	2 days	Jose Lombay	1.6.2.2
User Manual	3 days	Arelis Perez	1.7
Final Report	3 days	All	1.8

Hardware Block Diagram



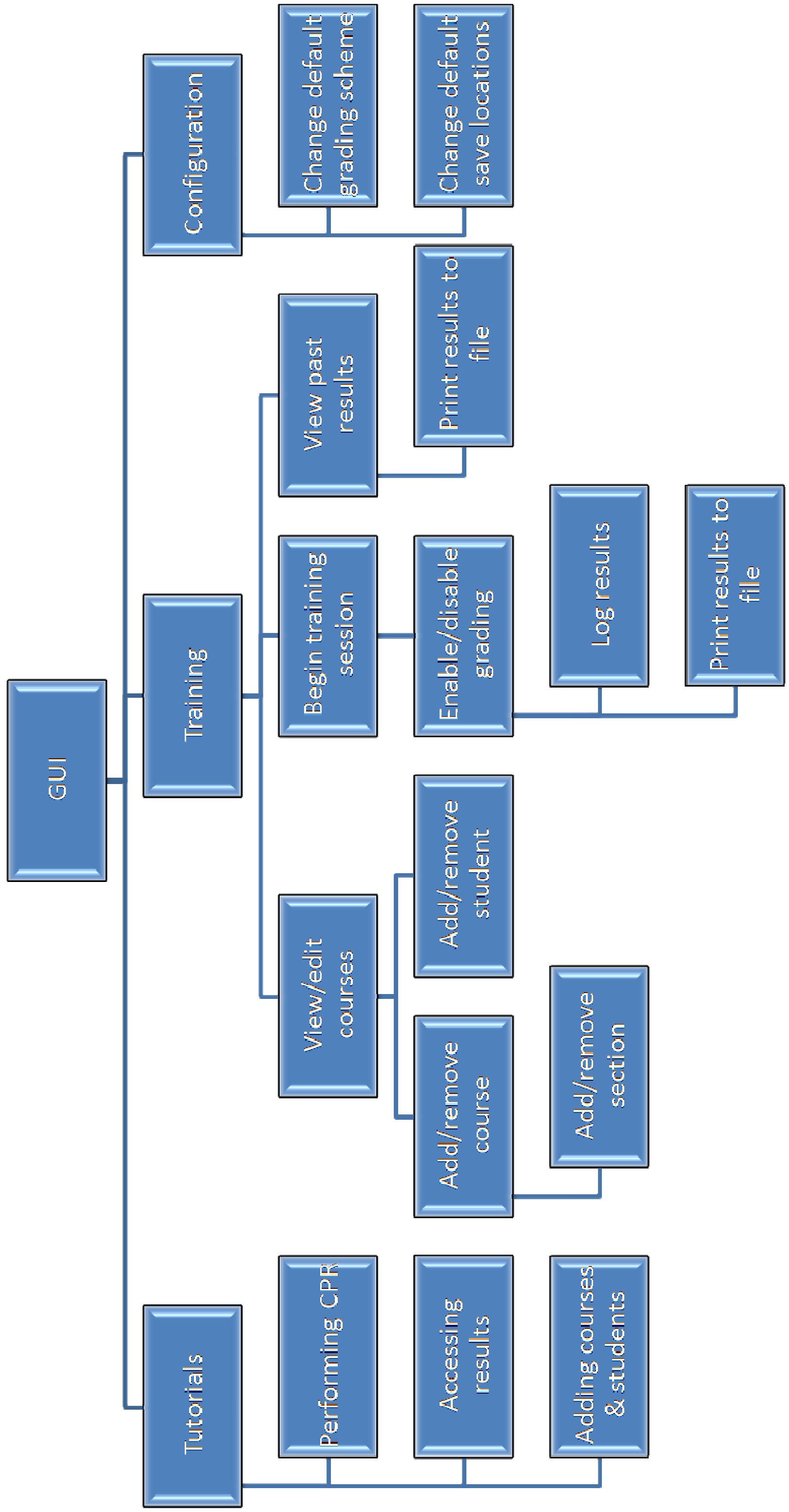
Hardware Mannequin Schematic




ER Diagram for CPR eTAM Database
courses
Course name
sections
Course number
amount of students
has
professor
professor name
professor sections
professor id
student
student section
student grade
student name
student number
evaluation session
% for breath check
% for pulse check
% for head check
% for chest
total grade
session id
n
1

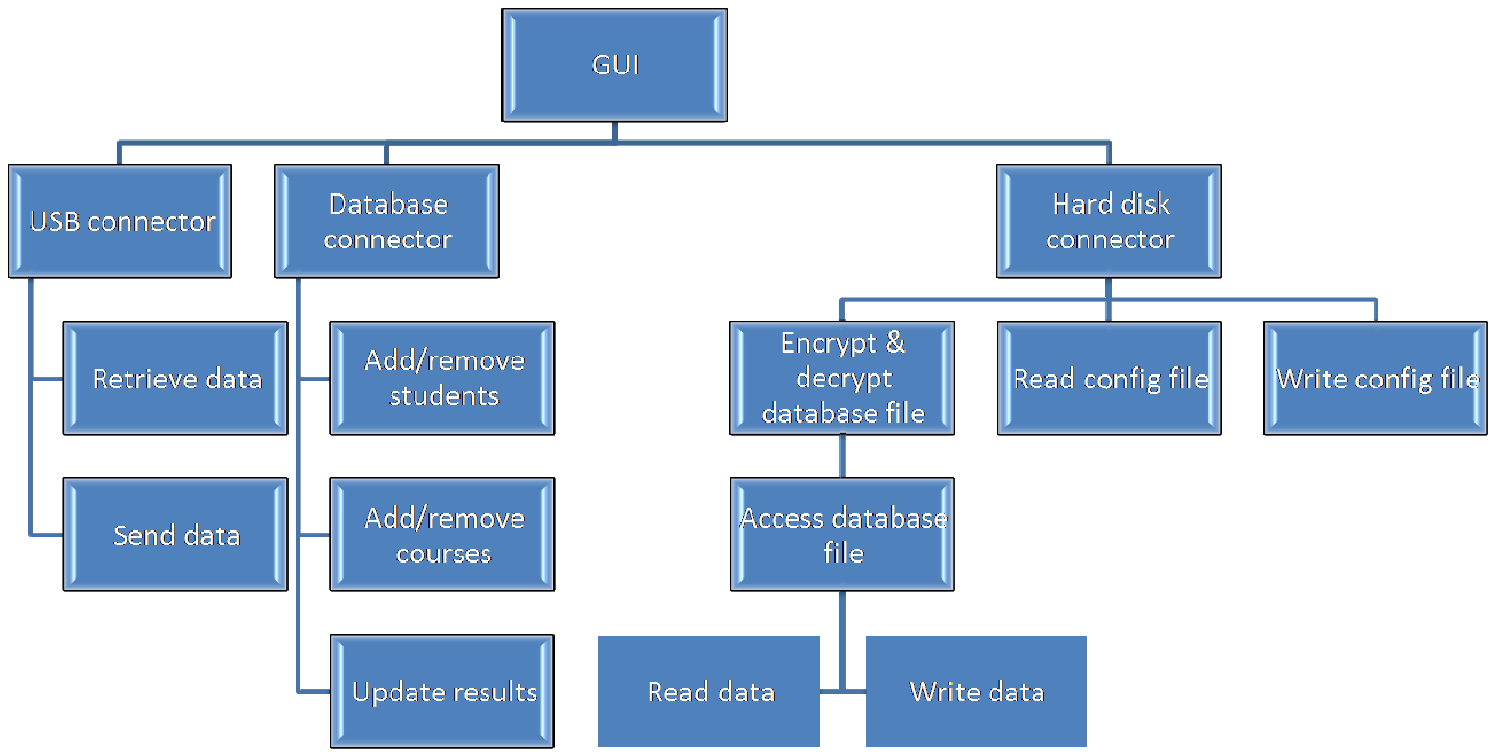
High-level software modules

Modules the user will actively interact with and see displayed through the GUI

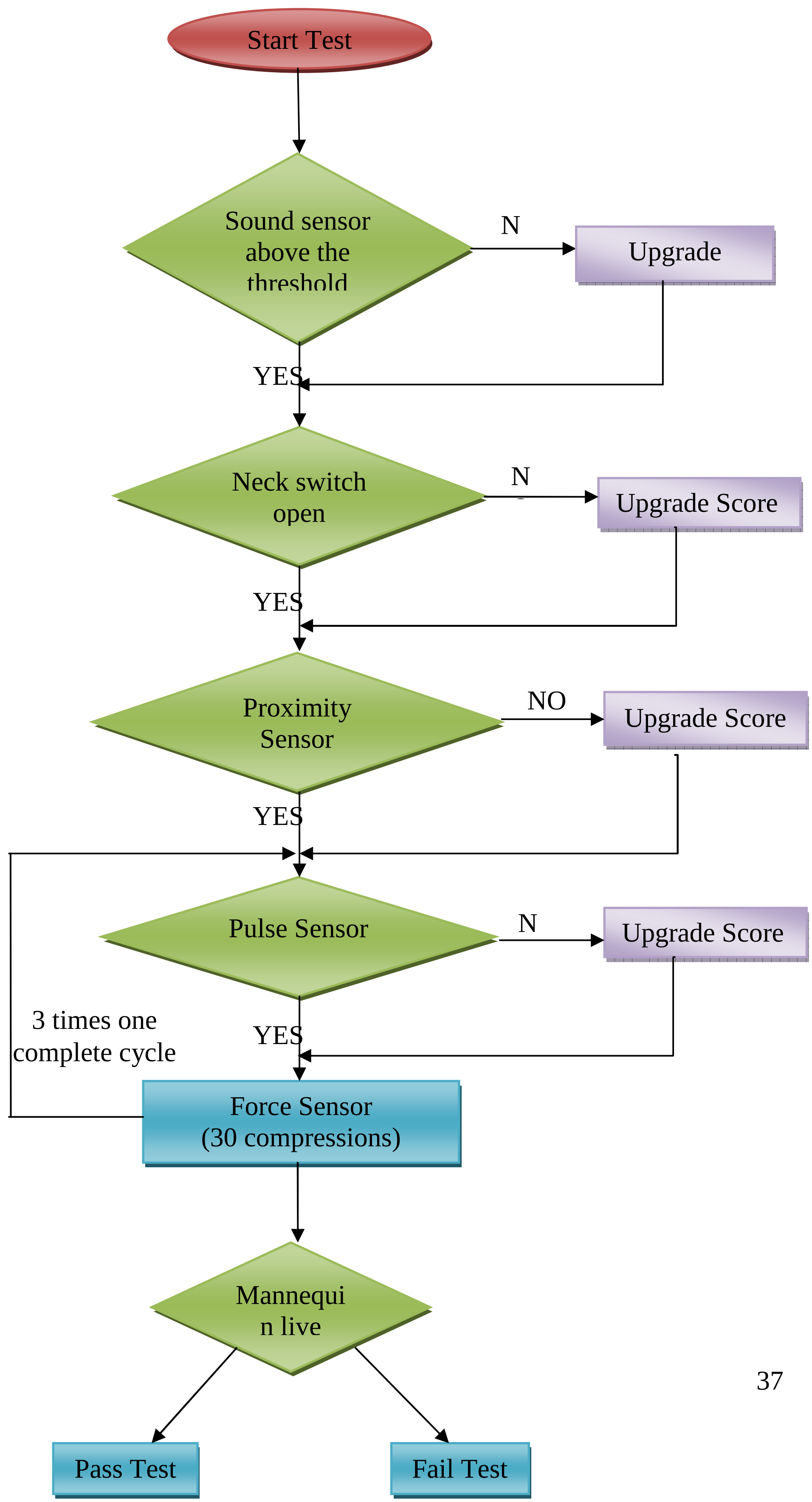


Low-level software modules

Modules the GUI will actively interact with , not the user.



Hardware Flow Chart



1. SYSTEM OVERVIEW

The C8051F000 family are fully integrated mixed-signal System on a Chip MCUs with a true 12-bit multi-channel ADC (F000/01/02/05/06/07), or a true 10-bit multi-channel ADC (F010/11/12/15/16/17). See the Product Selection Guide in Table 1.1 for a quick reference of each MCUs' feature set. Each has a programmable gain pre-amplifier, two 12-bit DACs, two voltage comparators (except for the F002/07/12/17, which have one), a voltage reference, and an 8051-compatible microcontroller core with 32kbytes of FLASH memory. There are also I2C/SMBus, UART, and SPI serial interfaces implemented in hardware (not "bit-banged" in user software) as well as a Programmable Counter/Timer Array (PCA) with five capture/compare modules. There are also 4 general-purpose 16-bit timers and 4 byte-wide general-purpose digital Port I/O. The C8051F000/01/02/10/11/12 have 256 bytes of RAM and execute up to 20MIPS, while the C8051F005/06/07/15/16/17 have 2304 bytes of RAM and execute up to 25MIPS.

With an on-board VDD monitor, WDT, and clock oscillator, the MCUs are truly stand-alone System-on-a-Chip solutions. Each MCU effectively configures and manages the analog and digital peripherals. The FLASH memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. Each MCU can also individually shut down any or all of the peripherals to conserve power.

On-board JTAG debug support allows non-intrusive (uses no on-chip resources), full speed, in-circuit debug using the production MCU installed in the final application. This debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, and run and halt commands. All analog and digital peripherals are fully functional when using JTAG debug.

Each MCU is specified for 2.7V-to-3.6V operation over the industrial temperature range (-45C to +85C). The Port I/Os, /RST, and JTAG pins are tolerant for input signals up to 5V. The C8051F000/05/10/15 are available in the 64-pin TQFP (see block diagram in Figure 1.1). The C8051F001/06/11/16 are available in the 48-pin TQFP (see block diagram in Figure 1.2). The C8051F002/07/12/17 are available in the 32-pin LQFP (see block diagram in Figure 1.3).

Table 1.1. Product Selection Guide

	MIPS (Peak)	FLASH Memory	RAM	SMBus/I2C	SPI	UART	Timers (16-bit)	Programmable Counter Array	Digital Port I/O's	ADC Resolution (bits)	ADC Max Speed (ksps)	ADC Inputs	Voltage Reference	Temperature Sensor	DAC Resolution	DAC Outputs	Voltage Comparators	Package
C8051F000	20	32k	256	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F001	20	32k	256	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F002	20	32k	256	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F005	25	32k	2304	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F006	25	32k	2304	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F007	25	32k	2304	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F010	20	32k	256	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F011	20	32k	256	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F012	20	32k	256	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP
C8051F015	25	32k	2304	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F016	25	32k	2304	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F017	25	32k	2304	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP

2. ABSOLUTE MAXIMUM RATINGS*

Ambient temperature under bias .... -55 to 125°C
Storage Temperature .... -65 to 150°C
Voltage on any Pin (except VDD and Port I/O) with respect to DGND .... -0.3V to (VDD + 0.3V)
Voltage on any Port I/O Pin or /RST with respect to DGND .... -0.3V to 5.8V
Voltage on VDD with respect to DGND .... -0.3V to 4.2V
Maximum Total current through VDD, AV+, DGND and AGND 800mA
Maximum output current sunk by any Port pin 100mA
Maximum output current sunk by any other I/O pin 25mA
Maximum output current sourced by any Port pin 100mA
Maximum output current sourced by any other I/O pin 25mA

*Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

3. GLOBAL DC ELECTRICAL CHARACTERISTICS

-40°C to +85°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Analog Supply Voltage	(Note 1)	2.7	3.0	3.6	V
Analog Supply Current	Internal REF, ADC, DAC, Comparators all active		1	2	mA
Analog Supply Current with analog sub-systems inactive	Internal REF, ADC, DAC, Comparators all disabled, oscillator disabled		5	20	µA
Analog-to-Digital Supply Delta (| VDD – AV+ |)				0.5	V
Digital Supply Voltage		2.7	3.0	3.6	V
Digital Supply Current with CPU active	VDD = 2.7V, Clock=25MHz VDD = 2.7V, Clock=1MHz VDD = 2.7V, Clock=32kHz		12.5 0.5 10		mA mA µA
Digital Supply Current (shutdown)	Oscillator not running		5		µA
Digital Supply RAM Data Retention Voltage			1.5		V
Specified Operating Temperature Range		-40		+85	°C
SYSCLK (System Clock Frequency)	C8051F005/6/7, C8051F015/6/7 (Note 2)	0		25	MHz
SYSCLK (System Clock Frequency)	C8051F000/1/2, C8051F010/1/2 (Note 2)	0		20	MHz
Tsysl (SYSCLK Low Time)		18			ns
Tsysh (SYSCLK High Time)		18			ns

Note 1: Analog Supply AV+ must be greater than 1V for VDD monitor to operate.
Note 2: SYSCLK must be at least 32 kHz to enable debugging.

C8051F000/1/2/5/6/7
C8051F010/1/2/5/6/7

Figure 4.1. TQFP-64 Pinout Diagram