March 11, 2010

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Functional Specifications for a Wireless Weight Distribution Scale

Dear Dr. Rawicz:

The following document contains the design specifications for Asäna's Wireless Weight Distribution Scale. The scale will be used to measure the user's body weight and center of balance. This information can be used for a wide range of health related applications. For example, a recovering stroke patient needs to relearn how to stand with an even weight distribution between both feet. The scale will wirelessly send the collected information to a nearby computer for data storage and analysis.

Details in methods of implementation are included in this design specifications document. Laying out our designs will give us a clear direction to our prototype development. It will also serve as a great set of standards that our prototype will meet for our project demonstration.

Asäna consists of four engineering students: Sam Leung, Sasan Naderi, Gurpal Sandhu, and Wil Gomez. For further inquiries about our company, please reach us by email at ensc440-asana@sfu.ca, or by phone at 778.861.3371.

Sincerely,

Sam Leung

Sam Leung
Chief Executive Officer
Asäna

Enclosure: *Design Specifications for a Wireless Weight Distribution Scale*

Asana

Design Specifications for a

Wireless Weight Distribution Scale

Prepared for

Dr. Andrew Rawicz – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Project Members

Sam Leung
Wil Gomez
Gurpal Sandhu
Sasan Naderi

Team Contact

ensc440-asana@sfu.ca

March 11, 2010

EXECUTIVE SUMMARY

In recent times the need for health monitoring has become more necessary as society as a whole has become further dependent on convenience. The statistics are overwhelming, particularly in North America, where a significant portion of the population is defined as either overweight or obese. The potential health risks associated with excessive weight are serious and include type 2 diabetes, cardiovascular disease, hypertension and stroke. Moreover, there are other significant indicators such as posture which can reflect one's overall health and well-being. The centre of balance of a standing individual contains vital information that can indicate whether a person is afflicted or recovering from many different ailments and injuries. By being conscious of weight and centre of balance, an individual can lead a healthier and richer life through more informed and intelligent decisions.

Asäna has already proposed an exciting and innovative product which addresses the current needs of the health industry. The Wireless Weight Distribution Scale is capable of tracking weight and weight distribution of a user over a large span of time. The scale is also capable of transmitting all measured data wirelessly to a nearby PC for storage. The data is graphically represented on the PC so that the general public as well as medical professionals can interpret the results. Along with the scale's ease of use, it boasts an impressive battery life and a range of wireless connectivity which will accommodate even the loftiest of environments.

The primary requirements of the Wireless Weight Distribution Scale provide an excellent reflection of the system's general features. Some of the key requirements are:

- The scale will be battery-operated and will last at least 1200 hours before battery replacement is necessary.
- All data will be wirelessly transferred to a local PC which is within the maximum operating range of 50 meters.
- The scale will have the capability of entering into a power-saving mode when not in use in order to preserve battery power.
- The process of connecting the scale to the local PC will be simple and straight forward for the average user.
- The wireless method of communicating the data will be unsecured.
- The scale will conform to standards and guidelines as stipulated by the CSA and FCC.

TABLE OF CONTENTS

REVISION HISTORY

TABLE 1: REVISION HISTORY

Version	Date	Summary of Changes
1.0	24/02/2010	Initial template created
1.1	06/03/2010	Rough draft completed
1.2	08/03/2010	Added figures and tables
1.3	09/03/2010	Finished editing
1.4	11/03/2010	Completed formatting

GLOSSARY

Coronal Plane	A coronal plane (also known as the frontal plane) is any vertical plane that divides the body into anterior and posterior (belly and back) sections.[2]
Sagittal Plane	A sagittal plane of the human body is an imaginary plane that travels vertically from the top to the bottom of the body along the Y axis, dividing it into left and right portions.[3]
Entire Scale	The physical scale including LCD, load cells, micro controller RF sender module.
Reticule	A shape superimposed on an image that is used for precise alignment of a device.
Bootloader	Software that runs immediately after a reset. Starts the application software.

ACRONYMS

USB	Universal Serial Bus
CRC	A cyclic redundancy check (CRC) is a non-secure hash function designed to detect accidental changes to raw computer data.[8]
CSA	Canadian Standards Association
FCC	Federal Communications Commission
OS	Operating System
MCU	Microcontroller Unit
ADC	Analog-to-Digital Converter
I/O	Input/Output
EEPROM	Electrically Erasable Programmable Read Only Memory
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
DIP	Dual In-line Package

I²C	Inter-Integrated Circuit
WPAN	Wireless Personal Area Network
PAN	Personal Area Network
CSMA/CD	Carrier Sense Medium Access/Collision Detect
IEEE	Institute of Electric and Electronic Engineers
PC	Personal Computer
MUX	Multiplexor
ADMUX	ADC Multiplexer Selection Register
ADSC	ADC Start Conversion
ADCSRA	ADC Control and Status Register A
TWI	Two Wire Interface (alternate name for I²C)
WDT	Watchdog Timer
BOD	Brown-out Detector
WDTCSR	Watchdog Timer Control and Status Register
SM	Sleep Mode
SMCR	Sleep Mode Control Register
SE	Sleep Enable
INT0	Interrupt Request 0
SREG	Status Register
ISC01/ISC00	Interrupt Sense Control 0 Bit 1 and Bit 0
EICRA	External Interrupt Control Register A
EIMSK	External Interrupt Mask Register
ISR	Interrupt Service Routine

IDE	Integrated Development Environment
MAC	Medium Access Control
ISM	Industrial, Scientific and Medical
RF	Radio Frequency
TX/RX	Transmit/Receive
CTS	Clear to Send
DI/DO	Data In/Data Out

1. Introduction

The Wireless Weight Distribution Scale is a digital weighing scale, which sends collected data to a personal computer. The data collected is the user's body weight and weight distributed along the coronal and sagittal planes. The weight information is displayed on the scale and wirelessly transmitted to a personal computer for analysis. This document describes the functional specification of the Wireless Weight Distribution Scale, as proposed by Asäna.

1.1. Scope

The development of this scale requires a set of design specifications. From proof-of-concept to final production, these requirements are needed to identify the design goals and verify the functionality of the scale. The design specifications provide a means to assess and ensure that all requirements have been met.

1.2. Intended Audience

Members of Asäna will refer to the design specification as a measure of progress throughout the development cycle. This document serves as a guideline for the project manager to ensure that the execution of goals and tasks are carried out accordingly. The design team will refer to this document for design details and unit tests.

2. Overall System Design

The Wireless Weight Distribution Scale can be broken down into 5 main components: the sensor unit, wireless communications unit, display unit, microcontroller unit, and the user software. The various design choices for each component are described in this section which will provide a complete view of how they will work together in the final product.

2.1. Mechanical Design

The design of the scale does not require many mechanical parts since most of the components which are used in the scale are electrical sensors and control units. Therefore in this section, we will mostly outline the physical characteristics of the scale such as the dimensions and placement of components within the chassis of the scale.

We purchased an Anyload ES300 scale from the manufacturer which came with four load cells, and a sturdy chassis. When choosing the scale we took into consideration several key factors such as surface area of scale, number of load cells, load cell sensitivity, total allowable weight and cost. The overall dimensions of the scale are shown in Figure 1.

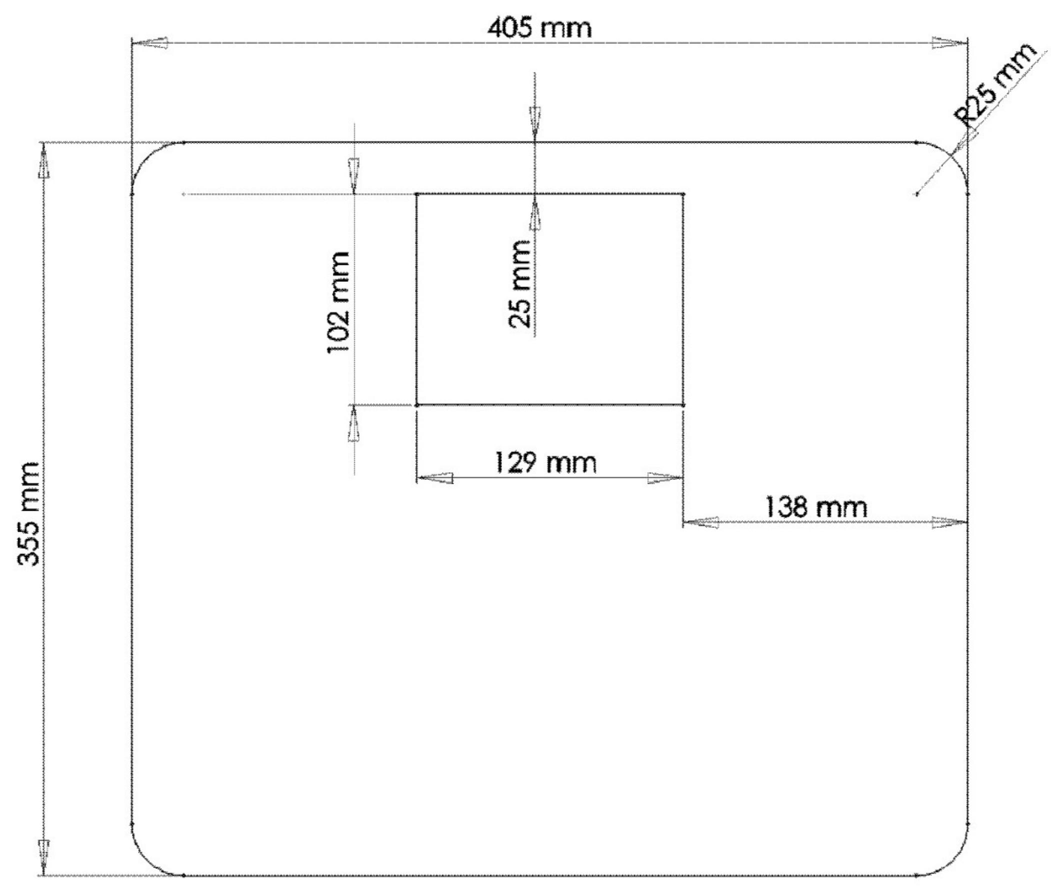

FIGURE 1: DIMENSIONS OF SCALE FROM TOP VIEW

The load cells are located in each corner. They are housed within the stabilizing columns which act as a leg for each corner of the chassis. On the bottom of each stabilizing column there is a non-slip pad to eliminate sliding of the scale on smooth surfaces. The column uses a spring in order to compensate for uneven surfaces.

The LCD which we plan on incorporating into the scale will be housed separately for our proof of concept prototype. All the peripherals which make up the scale will be neatly packaged into an external case only for the prototype. There are several reasons for this which include the fact that the manufacturing of the scale was done by Anyload and hence consideration was not taken to ensure that enough room would remain in the packaging of the scale for our hardware components. During our scale's production, the design of the scale's chassis will accommodate all of the additional hardware components such that they all fit in one package.

We also intend on creating patterns which resemble the outline of a bare footprint in varying sizes to visually allow the user to place their feet in the same position as their previous attempts. This will reduce any variance between measurements and increase accuracy. Figure 2 displays the scale with the graphic mentioned.

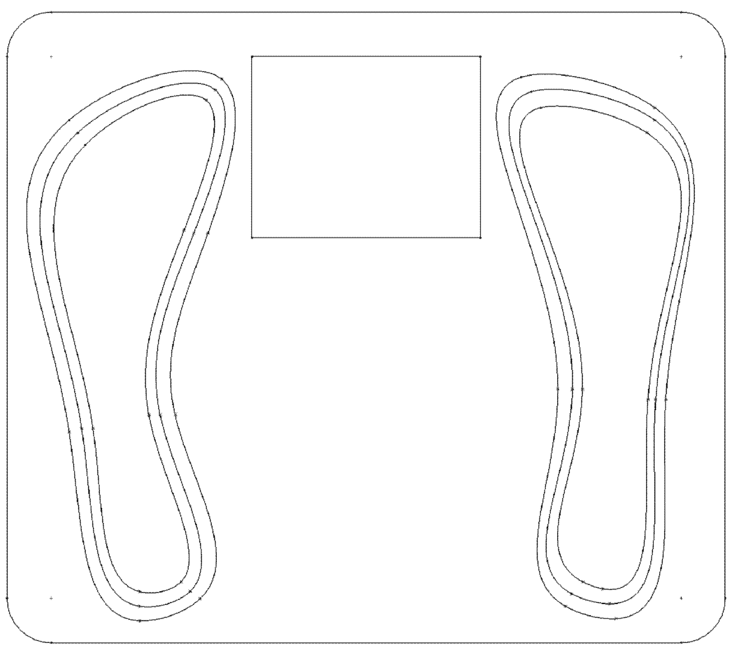

FIGURE 2: THE FOOTPRINT GRAPHIC IS SHOWN ON TOP OF THE SCALE

In order to wake up the MCU from a power-down state, we need to generate an external interrupt. The external interrupt will be generated when a user steps onto the scale. A simple pushbutton switch will be connected to the chassis such that a user's weight will cause the surface of the scale to lower and press the pushbutton switch thus alerting the MCU to turn on. The pushbutton will be located directly underneath the stainless steel surface in the centre of the scale since it is expected that the greatest displacement should occur there.

2.2. MICROCONTROLLER UNIT / DEVELOPMENT BOARD

The MCU is the heart of the scale. It is responsible for almost all of the operations of the scale from supplying ADCs for load cell signals to controlling external peripherals through digital I/O ports.

Careful consideration was taken when choosing the appropriate MCU. We decided upon a strict list of requirements that the MCU should possess like power efficiency, number of analog inputs, ADC resolution, internal flash memory capacity and available serial interfaces. In the end we concluded that the Atmel ATmega328P was the best fit for our design. The Arduino Duemilanove was chosen as our development board because it supported the MCU. The subsequent sections describe the features and specifications of the MCU and development board that are deemed crucial to the scale's operation. Please refer to Figure 3 to see the general physical connections between our system's peripherals and the MCU.

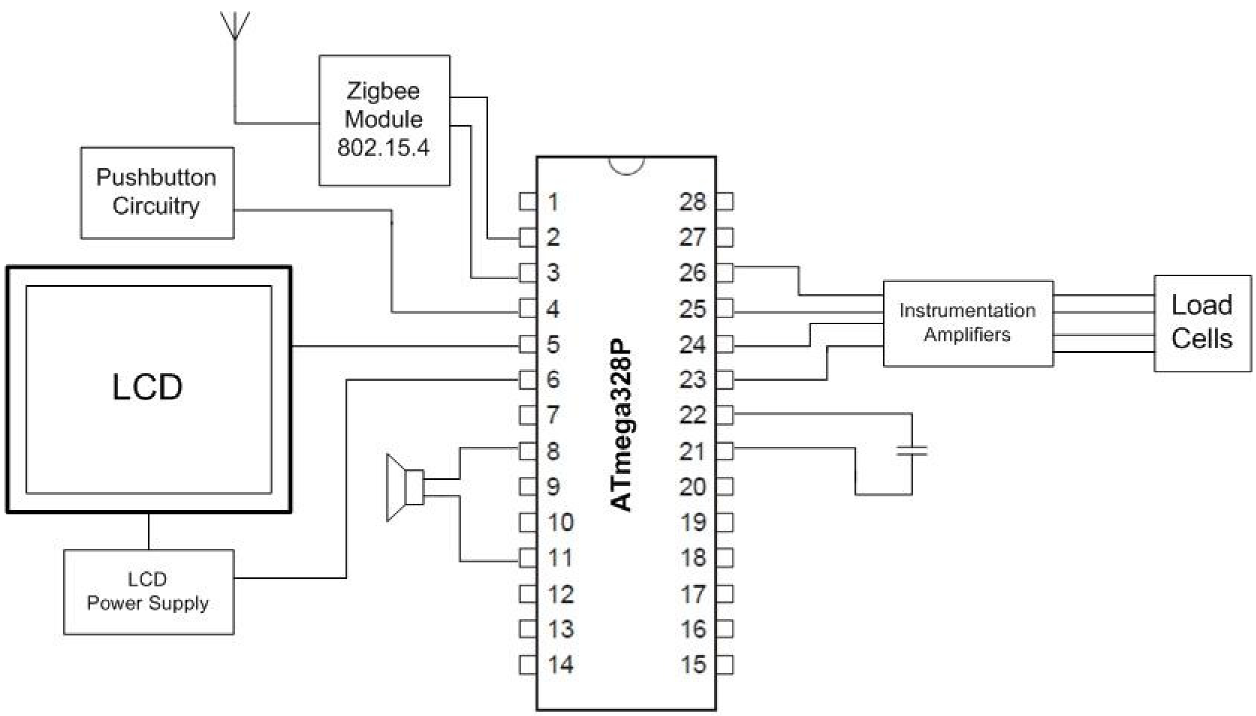

FIGURE 3: A BROAD SYSTEM OVERVIEW OF SYSTEM PERIPHERALS CONNECTED TO MCU

TABLE 2: USEFUL PIN ASSIGNMENTS OF THE ATMEGA328P

Pin #	Pin Name	Functionality
1	PC6	RESET
2	PD0	RXD
3	PD1	TXD
4	PD2	INT0
5	PD3	INT1
7	VCC	+5V
8	GND	
11	PD5	Speaker Output
17	PC3	ADC3
18	PC2	ADC2
19	PC1	ADC1
20	PC0	ADC0
21	GND	
22	AREF	
23	AVCC	

2.3. SPECIFICATIONS

TABLE 3: ATMEL ATMEGA328P SPECIFICATIONS

Flash Memory	32K Bytes In-System Self-Programmable Flash program memory
EEPROM	1K Bytes
Internal SRAM	2K Bytes
Peripheral Features	
Timer/Counter	Two 8-bit Timer/Counter with Separate Prescaler and Compare Mode
ADC	6-channel 10-bit ADC
Interrupts	Two External Interrupts (INT0 and INT1)
Serial Interfaces	SPI, I^2C, UART
I/O and Packages	
I/O	23 Programmable I/O Lines
MCU Package	28-pin DIP
Power Consumption at 1 MHz, 1.8V, 25°C	
Active Mode	0.2 mA
Power-down Mode	0.1 µA

2.4. ARDUINO DUEMILANOVE

The Duemilanove is a simple development board designed to support the Atmel ATmega328P and ATmega168P MCUs. The board can be powered through a USB port or a barrel jack connected to either an AC adaptor or a battery source. We intend on powering the entire system with a single 9 V battery connected to the barrel jack. However, during development and testing we will power the board through the USB port. The USB port is also used to download binary images to the MCU from a PC.

The Duemilanove uses an on-board 16 MHz crystal oscillator as the main clock for the MCU. In order to provide power for the MCU, the board regulates an output of 5 V and 3.3 V which can also be used for other purposes. We intend on using the regulated 5 V output to power the instrumentation amplifiers for the load cells. There is a hardware reset push-button switch on the board and all of the programmable I/O ports are connected to easily accessible jumpers.

2.4.1. Analog to Digital Converter

The 28-pin DIP ATmega328P, features a 10-bit successive approximation ADC. There are 6 individual analog channels which can be selected through an internal MUX. The analog input can be selected by manipulating the MUX3…0 bits in the ADMUX register. Each channel is connected to a port on the MCU called an analog input. Analog inputs are generally used for interfacing with a wide range of sensors.

For our design, we are using 4 load cells. Each load cell will be connected to the analog input of the MCU. By default the ADC will run in free running mode which is enabled by using the ADC interrupt flag as a trigger source. This makes the ADC start a new conversion as soon as the ongoing conversion has finished. The initial conversion is triggered by writing a '1' to the ADSC bit in the ADCSRA. Although this may seem inefficient, our design calls for such an implementation so that we can determine a stable reading once a user’s weight is sensed on the load cell.

We also took into consideration the conversion time since it will directly affect the channel selection delays. For instance, referring to Table 4, the conversion time will differ depending on if an initial conversion is taking place as opposed to subsequent conversions.

TABLE 4: ADC CONVERSION TIME

Condition	Sample & Hold (cycles)	Conversion Time (cycles)
First conversion	13.5	25
Normal conversions (e.g. Free Running mode)	1.5	13

“In free running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.”

This would imply that after the MUX3…0 bits of the ADMUX register are modified, there needs to be a delay to guarantee that the ADC conversion will reflect that of the new channel rather than the previously selected channel. Since the conversion time could take anywhere from 13 to 260 µs (depending on the sampling frequency), a 1 ms delay between analog input changes will ensure data correctness.

2.4.2. Power Reduction Management

The ATmega328P features 6 sleep modes which help in reducing power consumption through disabling specific unused peripherals. These modes are outlined in Table 5.

TABLE 5: ACTIVE CLOCK DOMAINS AND WAKE-UP SOURCES IN THE DIFFERENT SLEEP MODES

	Active Clock Domains					Oscillators		Wake-up Sources							
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	Software BOD Disable
Idle			X	X	X	X	$X^{(2)}$	X	X	X	X	X	X	X	
ADC Noise Reduction				X	X	X	$X^{(2)}$	$X^{(3)}$	X	$X^{(2)}$	X	X	X		
Power-down								$X^{(3)}$	X				X		X
Power-save					X		$X^{(2)}$	$X^{(3)}$	X	X			X		X
Standby(1)						X		$X^{(3)}$	X				X		X
Extended Standby					$X^{(2)}$	X	$X^{(2)}$	$X^{(3)}$	X	X			X		X

Notes: 1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

Through careful research, it become evident that the power-down mode was the most power efficient and could only be woken up through an asynchronous external interrupt level change, a TWI address match by an I^2C peripheral, or a WDT or BOD reset. Also, the power-down state automatically disables many of the peripherals that we would have otherwise had to manually disable.

One thing to note is that the BOD and I^2C interface are disabled by default, therefore, we do not need to concern ourselves with them since they won't be draining any unnecessary power. However, the WDT will need to be disabled only once during the initialization of the system. This is done by writing 0x00 to the entire WDTCSR.

In order to set the sleep mode, the SM2…0 bits in the SMCR must be set to power-down which are '010' respectively. Also the SE bit must be set to '1' to make the MCU enter the sleep mode when the SLEEP instruction is executed.

“To avoid the MCU entering the sleep mode unless it is the programmer’s purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.”

When the MCU is in the power-down mode, only an external interrupt level change on INT0 can wake it up. Upon waking up, the MCU executes the instruction directly following the SLEEP instruction. The interrupt required for this function is described in the next section.

2.4.3. INTERRUPTS

Our design will utilize externally generated interrupts solely for the purpose of awakening the MCU from the sleep mode. In our design, a simple pushbutton switch will be used to create the level change required to cause the interrupt. It should be noted that only a level change on an external interrupt pin can satisfy the conditions to awaken the MCU from the power-down state.

"When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock, described in ”Clock Systems and their Distribution” on page 26. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interrupt can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode."

We've chosen to configure the interrupt sense control such that any logical change on INT0 generates an interrupt request.

The process of enabling and configuring interrupts is quite simple with respect to the application. During initialization of the application, the I bit of the SREG must be set to '0' which in effect disables all interrupts. This allows us to make critical changes to interrupt configurations without the risk of having the MCU service an interfering interrupt. We can then set ISC01 and ISC00 bits in the EICRA register to '01' respectively to indicate that any logical change will generate an interrupt on INT0. The term INT0 refers to the external interrupt pin which corresponds to pin 4 of the MCU package.

Prior to entering the power-down state, we must enable INT0 by writing a 1 to the INT0 bit of the EIMSK register. This will enable the external interrupt request 0 pin. Finally, we must reset the I bit of the SREG to 1 so that interrupts are globally enabled. The application enters the power-down state and remains in this state only until a level change has been detected on INT0.

2.4.4. TIMER

As a power reducing feature in our design, we have implemented a method of forcing the MCU into a sleep mode when there are no further operations to be completed. This scenario occurs after all of the information has been calculated, transmitted wirelessly and displayed on the LCD for an appropriate length of time such as 5 seconds. In order to keep track of the time in seconds that the information has been displayed on the LCD, we require a timer that will generate an interrupt to notify the application that the sleep condition has occurred.

The 16-bit timer/counter will be used to set the total time in seconds and provide the comparison between the current time and set time. When the comparison is evaluated to be true, an interrupt is issued. Our ISR will be simple and quick and will only set a global flag to alert the main loop of the application that the condition to put the MCU into the power-down state has been satisfied.

2.4.5. UART

The UART provides us with the capabilities of serially communicating with the Arduino board. This feature was very helpful during the development and testing phases in which binary files containing the application were downloaded to the MCU through the UART. The UART connection was made directly through the USB connector on the Arduino board. The same port that can be used to supply power to the board and MCU. A key benefit to using the Arduino Duemilanove was the fact that the ATmega328P was preloaded with the Arduino bootloader which allows the download of new code to the MCU without an external hardware programmer.

The actual design of the system calls for the Xbee 802.15.4 module to be connected to the external TX and RX pins of the UART. Initial testing done with the LCD proved that an external peripheral connected to the UART will not hinder the downloading of the process from the PC to the Arduino board. A more detailed explanation of the Xbee 802.15.4 module interface is provided in section 5.

2.4.6. DIGITAL PORTS

The 23 programmable I/O ports featured on the ATmega328P can be used for a multitude of purposes. These ports are pre-configured for a specific subset of functions like external interrupts, SPI or UART. However, they can also be configured as generic digital I/O ports.

Due to the fact that our Xbee 802.15.4 module will be connected to the pre-configured UART ports, we will implement a software serial port that will emulate the function of a UART through another set of ports. This is important because the LCD and Xbee 802.15.4 modules were designed to connect to the MCU through the same interface and now they can, with the exception that the LCD will use a software emulator that is provided through an Arduino IDE library.

Also, a couple of the ports will be configured as outputs in order to interface to a speaker and control the power supply of the LCD which will effectively turn on and off the LCD when not in use. The speaker will be used to alert the user that all of the calculations have taken place and the user can step off the scale. It is assumed that before hearing the audible alert from the speaker, the user is steadily standing in a natural posture.

2.4.7. Application Programming Interface

By choosing the Arduino Duemilanove, we were fortunate enough to inherit a rich collection of pre-defined libraries through the generosity of online communities. Since the Duemilanove has hardware and software which are open source, there are libraries which exist to implement almost every feature of the ATmega328P. In our design we require the use of libraries that will implement interrupts, ADCs, timers, serial interfaces, and I/O ports.

2.4.8. General System Flowchart

The following flowchart outlines the general system behavior. The system is based on interrupts and timers in order to determine the current state. Based on that information the scale will know whether to calculate, display or transmit weight information or power-down to a sleep state.

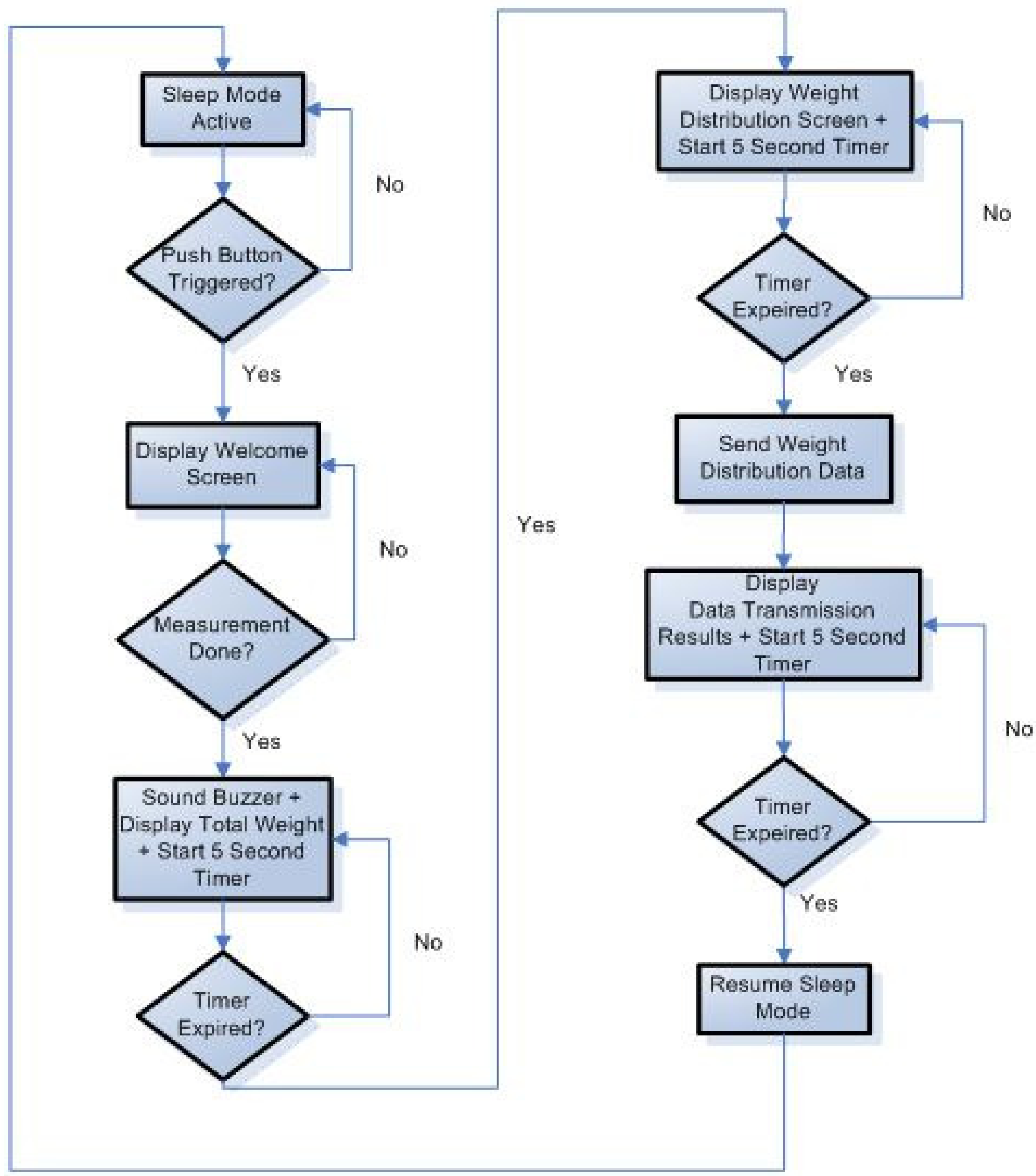

FIGURE 4: FLOWCHART OF GENERAL SYSTEM BEHAVIOUR

3. Weight Sensor System

3.1. Strain Gauge Sensors

The sensor units are used to measure weight on the scale. To measure weight, we have chosen to use strain gauge sensors to convert a physical strain force into an electrical signal. Strain gauge sensors were chosen over other types of load cells such as a hydraulic load cell or load buttons. This is because strain gauge sensors suit the range of weights we will be measuring and can provide accurate results with proper signal conditioning all for a very reasonable price.

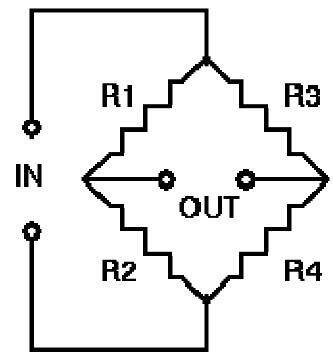

FIGURE 5: WHEATSTONE BRIDGE CIRCUIT

As the diagram in Figure 5 shows, the strain gauge sensor consists of a wheatstone bridge circuit. This circuit works by an excitation voltage supplied to the circuit, and in the presence of a strain on the resistor, the resistance changes, thus resulting in a changing output voltage measured at the “OUT” nodes.

The strain gauge sensors chosen to be used in our scale are designed by Anyload. These strain gauge sensors were used in their scale model ES300-150kg. Each of these sensors have a capacity of 80kg. In order to accurately measure weight with these load cells, we will characterize them by placing known weights on the load cell, and plot the weight-to-voltage relationship on a graph to find the load cell's load factor.

3.2. Instrumentation Amplifier

Because a strain gauge sensor will show only a few millivolts change at maximum load capacity, we require an amplifying circuit to magnify the change in voltage to be able to accurately measure weight. An instrumentation amplifier is used in our design to perform this task. The schematic for this circuit is shown below.

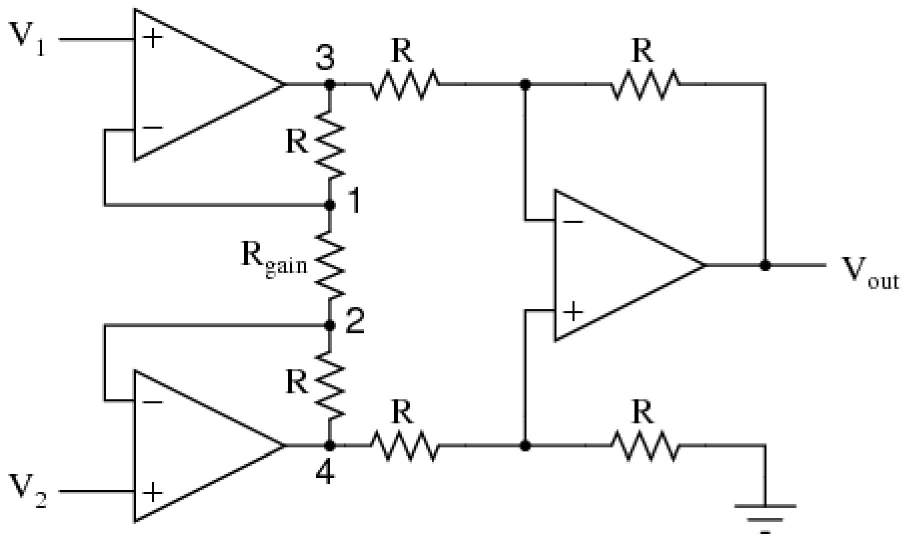

FIGURE 6: INSTRUMENTATION AMPLIFIER

The sense (output) wires of the load cell will serve as the input to our instrumentation amplifier circuit. The instrumentation amplifier circuit's gain is determined by using the following equation:

$$A_v = \left\{1 + \frac{2R}{\mathrm{R_{gain}}}\right\}$$

From the equation above, R was calculated to be 120 KΩ where R_{gain} is 1.2 KΩ to produce a gain of approximately 201. This gain will result in a swing of approximately 1.546 V from no load to maximum load. The output of the amplifier will be connected to our ADC where the voltage signal will be converted to a digital value for processing.

3.3. SENSOR PLACEMENT

In order to measure weight distribution along the sagittal and coronal planes, our scale will house a total of four load cells. The load cells will be distributed near each corner of the scale. By placing the load cells where the four feet of the scale rest, there will be minimal mechanical additions to ensure the load cells detect all the weight presented on top of the scale. The following diagram shows the location of the load cells.

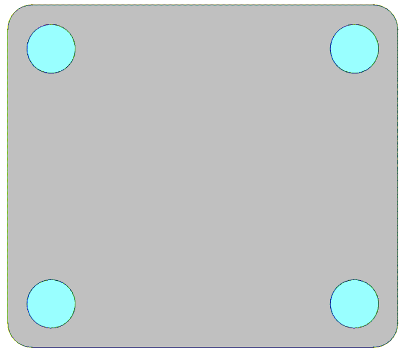

FIGURE 7: PLACEMENT OF SENSORS (HIGHLIGHTED IN BLUE)

4. Display Unit

4.1. Physical Design

The display unit has physical dimensions shown in Figure 8. When considering choice of LCD's a criterion was set on what functionality it should provide. Many factors were considered but most importantly it needed to be large enough so that results could be read from a distance of at least 2.0 meters. At first we considered getting a simple character LCD but we then realized there would be no way of showing weight distribution in a graphical form to the user. Also the size of each character was hard coded so there was no way we could enlarge the font. For this reason we chose a graphical LCD where each pixel is programmable giving endless possibilities on how to tailor it to our needs.

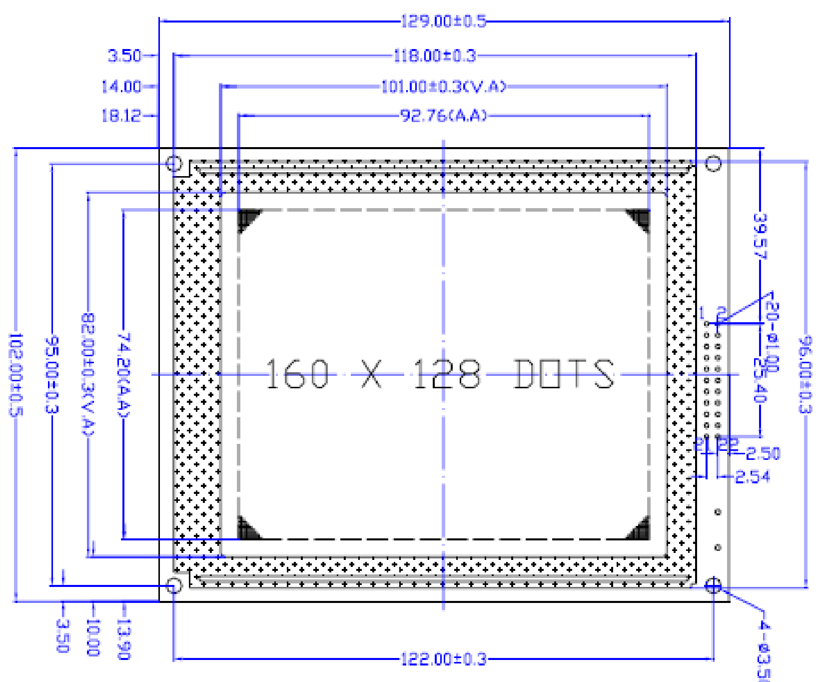

FIGURE 8: PHYSICAL DIMENSIONS OF GRAPHICAL LCD

4.2. Electrical Design

The display unit has the following electrical characteristics shown in Table 6. It is being controlled through serial communication which is essential since our development board has a limited amount of output pins it can drive. Using serial communication required just one digital output pin. To conserve power, the backlight was adjusted to a 25% level where the display unit is still readable with ambient lighting at a viewing distance of 2.0m.

TABLE 6: ELECTRICAL CHARACTERISTICS OF GRAPHICAL LCD

Item	Value	Unit
Max Reverse Voltage	5.3	V
Current at 100% Backlight	200	mA
Max Power Dissipation	1000	mW
Current at 25% Backlight	75	mA

4.3. Display Flow of Events

The display's flow of events is a description of how the display will communicate with the user through the set of steps involved in taking a measurement. Initially the LCD will be turned off during power save mode. Once the user steps on and triggers the push button the LCD will power up and show the welcome screen. It tells the user to stand still and wait until an audible sound is heard which notifies the user that the measurement is complete. The welcome screen is shown below in Figure 9.

FIGURE 9: LCD WELCOME SCREEN

FIGURE 10: TOTAL WEIGHT SHOWN ON LCD

Next, the total weight is displayed on the LCD as in Figure 10. This screen is shown for 5 seconds then weight distribution is shown after replacing the total weight. This screen (Figure 11) is shown for five seconds, giving the user a more then an adequate amount of time to review the results. Finally the success of the transmission is shown on the last screen (Figure 12) which is also shown for 5 seconds. After that time the weight scale returns back into sleep mode.

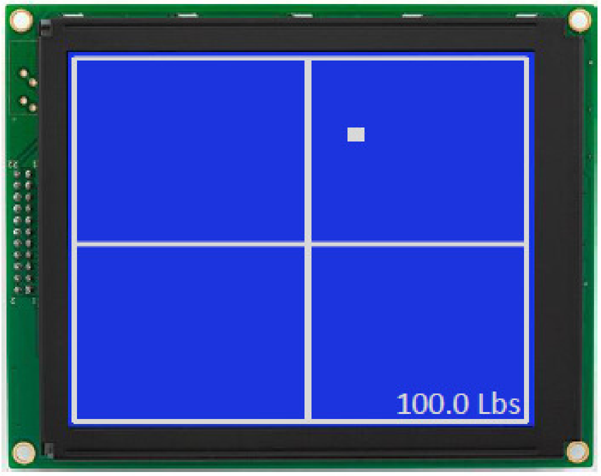

FIGURE 11: WEIGHT DISTRIBUTION SHOWN ON LCD

FIGURE 12: SUCCESSFUL TRANSMISSION MESSAGE ON LCD

5. WIRELESS UNIT

The wireless unit provides communication between the digital scale and the PC. The data captured by the digital scale is sent wirelessly to the PC for post analysis of total weight and weight distribution. The Zigbee protocol suite is an ideal wireless solution for this application. It is a low-cost and low-power alternative to WiFi and Bluetooth wireless protocols. The Zigbee specification is a simple and reliable protocol offering long battery life and communication range.

5.1. ZIGBEE PROTOCOL

The Zigbee protocol is based on the IEEE 802.15.4 standard for low-rate WPAN. The protocol is intended for use in embedded application requiring low data rates and power consumption.

The IEEE 802.15.4 standard specifies the lower protocol layers for Zigbee, which is the physical layer, and the MAC portion of the data link layer. As such, Zigbee uses CSMA/CD for basic channel access and it can operate within 868 MHz, 915 MHz or 2.4 GHz ISM bands. A 2.4 GHz frequency band is desired for long range, which is less susceptible to external device interference.

Zigbee supports beacon and non-beacon enabled networks. In a beacon network, a Zigbee node is only active when a beacon is being transmitted, whereas in a non-beacon network, the device is always active or sleeping most of the time. For the purpose of our application, we are only concerned with a non-beacon network since the device will only be active when it is ready to transmit data to the PC.

5.2. Wireless Transmitter: XBee 802.15.4

When selecting a Zigbee RF transmitter, a set of requirements was considered for its implementation. The most significant requirement was power consumption and transmission range. The Xbee 802.15.4 is a low-cost and low-power RF module that operates within the ISM 2.4 GHz frequency. The RF module provides an acceptable range of operation and consumes very little power during inactivity or transmission. Table 7 lists the specification of the Xbee 802.15.4 RF module, which satisfies the requirement of a Zigbee transmitter.

TABLE 7: SPECIFICATIONS OF XBEE 802.15.4

Parameter	Value
Indoor/Urban	Up to 100’ (30 m)
Outdoor line-of-sight	Up to 300’ (100 m)
Transmit Power	1 mW (0 dBm)
Receiver Sensitivity	-92 dBm
TX Current	45 mA @ 3.3 V
RX Current	50 mA @ 3.3 V
Power-down Current	< 10 mA

The transmitter uses a UART interface to communicate directly with the microcontroller. The RF module only supports non-beacon networking, and it operates in the default Unicast mode, which is the only mode that supports re-transmission. Figure 13 is the mechanical drawing of the XBEE 802.15.4 module.

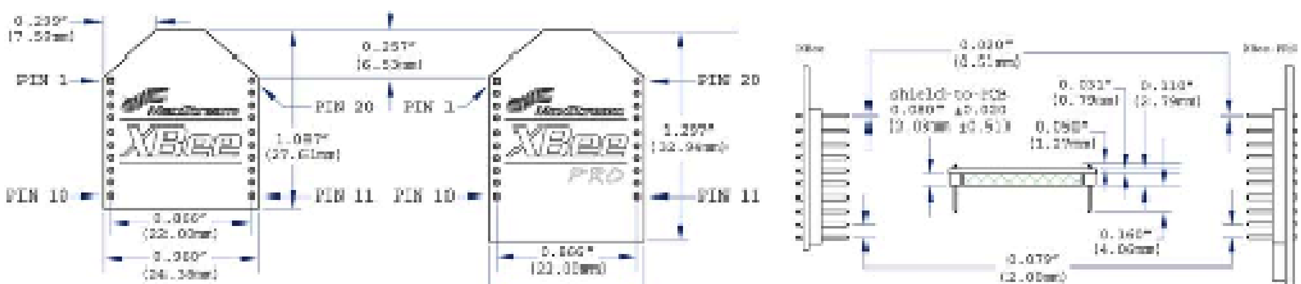

FIGURE 13: MECHANICAL DRAWING OF XBEE 802.15.4

5.2.1. Antenna

The XBEE 802.15.4 RF module is available in either the whip antenna or the low-profile chip antenna. A series of range tests were performed by Maxstream, which documented the results between the XBee whip antenna and chip antenna. The results from the document are shown in Table 8.

TABLE 8: COMPARISON BETWEEN THE CHIP AND WHIP ANTENNA

Antenna Type	Outdoor Distance (Visual Line-of-Sight)	Indoor Distance (Office Building)
Chip	470 ft. (143 m)	80 ft. (24 m)
Whip	845 ft. (258 m)	80 ft. (24 m)

From Table 8, the whip antenna has a better outdoor range advantage than the chip antenna. However, the digital scale is used indoors, and as a result, the range distance between the whip and chip antenna is identical. The housing of the digital scale is the only exception to using the chip antenna. The digital scale is housed by an aluminum/stainless steel casing, which hinders the transmission of wireless signals. As such, the whip antenna is the optimal choice for overcoming this hindrance by orientating the module where the antenna is outside of the casing.

5.2.2. POWER CONSUMPTION

The XBEE 802.15.4 RF module offers three sleep modes to achieve low power consumption. These modes are outlined in Table 9.

TABLE 9: SLEEP MODE CHARACTERISTICS & POWER CONSUMPTION OF XBEE 802.15.4 RF MODULE

Sleep Mode	Characteristics	Power Consumption
Hibernate	Lowest Power	< 10 mA
Doze	Fastest Wake-up	< 50 mA
Cyclic Sleep	RF Module wakes in pre-determined time intervals to detect if RF data is present	< 50 mA when sleeping

Hibernate mode is a voltage level-activated mode that offers the lowest power consumption during inactivity. When a sleep request is asserted, the module will finish its transmission, enter idle mode and then sleep. During this state, the module will not respond to either serial or RF activity as desired. The module will only wake-up when the sleep request is de-asserted and begin to transmit when the CTS line is low.

5.2.3. XBEE SHIELD

The Xbee shield allows the Arduino board to conveniently communicate wirelessly using the XBee 802.15.4 RF module. This shield can be used as a serial/usb replacement or it can operate in command mode where it can be configured for a particular networking option. The module adds convenience and simplicity in integrating the XBee 802.15.4 transmitter with the Arduino board. Figure 14 illustrates the Arduino board with the mounted Xbee shield.

FIGURE 14: XBEE SHIELD MOUNTED ON THE ARDUINO BOARD

5.3. Wireless Receiver: RZUSBSTICK

The Zigbee RF receiver requires simplicity in adaptation with the PC. The RZUSBSTICK is a Zigbee RF module that provides a USB interface with the PC. The hardware uses a USB microcontroller to communicate with the PC and a Zigbee radio transceiver that manages the RF protocol stacks. The RZUSBSTICK is a 2.4GHz radio transceiver with low power consumption that supports the IEEE 802.15.4 standard. The antenna on the module is a folded dipole antenna with a net peak gain of 0 dB.

5.4. Wireless Communication Process

The XBee 802.15.4 contains an input and output buffer that controls the flow of serial to RF packetization. In Figure 15, serial data enters the RF module through the DI pin where data is stored in the DI buffer waiting to be transmitted.

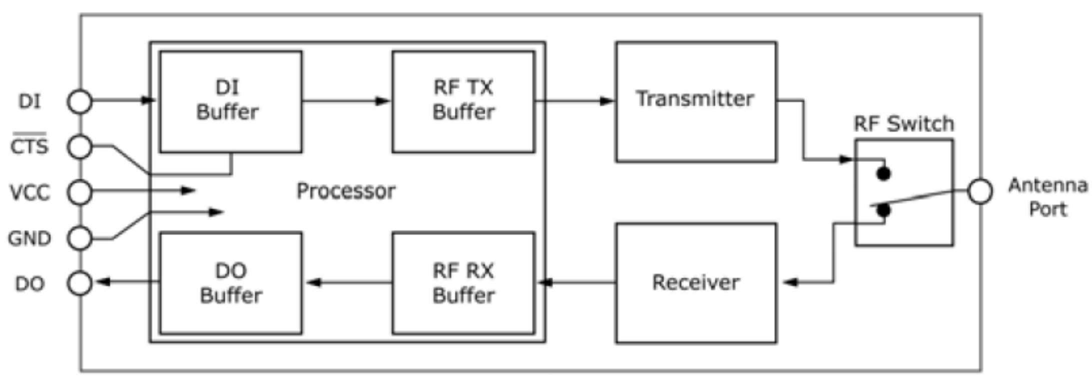

FIGURE 15: DATA FLOW CONTROL OF THE XBEE 802.15.4 RF MODULE

The transmitter is set-up to use the same communication channel as the receiver. When a transmit request is sent to the transmitter, RF data is packetized and delivered to the receiver. The transmitter waits for an acknowledgment from the receiver to verify the successful transmission of a packet. If transmission fails, the transmitter will retry up to three times before indicating an error in transmission.

6. User Software

Application software for the computer will provide a means of organizing data collected from the scale. It will allow the user to easily store measurements as well as the ability to view previous measurements. Aside from presenting the data, the software will also interface with the wireless receiver to manage the handshaking of the receiver/transmitter communication. It will only run on a windows environment initially, but if time permits the application will also be developed for other operating systems. The Application software will be coded using C#.

To receive a measurement from the scale the application must first be launched by the user. Upon receiving a measurement, the application software places the measurement as an 'un-organized' type. The 'un-organized' type means that details for that measurement have not yet been filled out as shown in Figure 16.

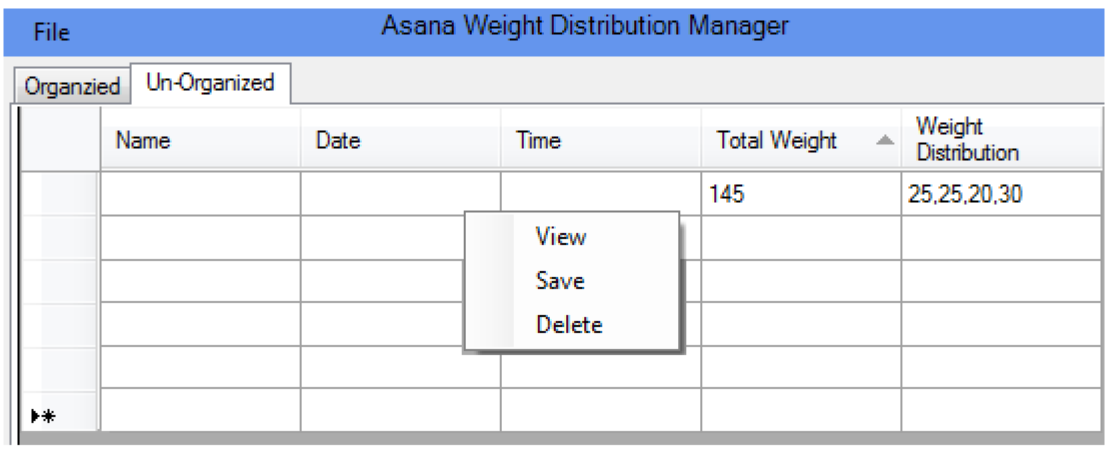

FIGURE 16: UN-ORGANIZED DATA

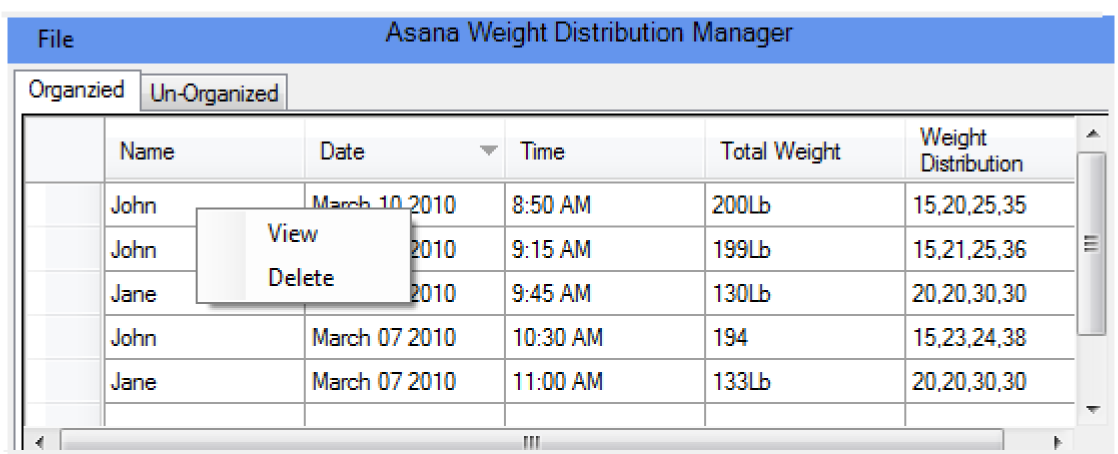

FIGURE 17: ORGANIZED DATA

Once it's filled out by the user the application software will place it as a 'organized' type, see Figure 17. To view each individual entry the user clicks on the View button beside each entry and a pop up comes up as in Figure 18. They also have the option to side-click on each entry and either view or delete it. Once the user is done viewing the results they can close the application which will cause the communication channel to close, assuming that the scale was already paired with the PC.

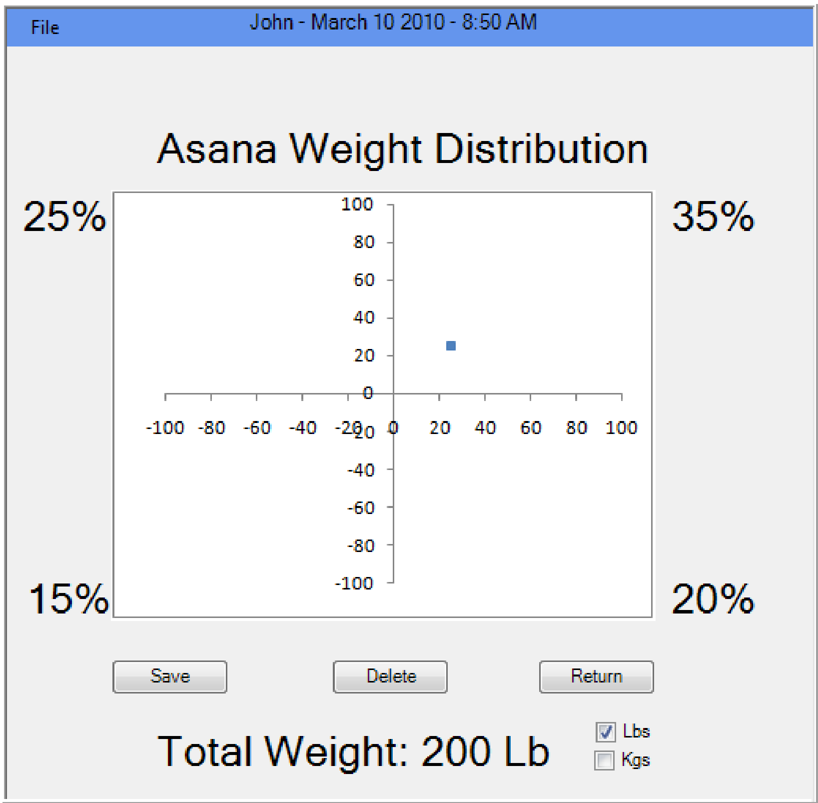

FIGURE 18: ENTRY DETAILS

6.1. USABILITY DESIGN

The user interface was designed in the simplest way possible allowing any kind of person to become familiar with the software. When an entry is received, the user is allowed to enter the following:

- Date
- Time
- Name

The entries can be sorted by date, weight, or user. In the pop up view, the user can:

- Delete - Erases the entry, then the program returns to the organized view
- Save - Saves the entry if it hasn't been saved already
- Return - Returns the organized view

- lbs/kg - View result in pounds or kilograms

The Graphical form has four percentages shown at each corner. They correspond to the amount of weight as a percentage of the total weight that each load cell is sensing.

7. System Test Plan

The system test plan is divided into individual sections where the functionality of each specific unit is tested. The testing focuses on the functionality of the weight sensor unit, LCD unit, and wireless unit. Each individual unit is already integrated with the MCU to verify the correctness of the algorithm and desired output. As such, tests are evaluated from verifying the measurements of weights with respect to position, and validating the transmission of data under certain conditions.

7.1. Weight Positioning Verification

This verification test analyzes how the weight sensor will correspond to the LCD. The test focuses on examining the reticule's position with respect to weights placed in different areas on the scale. The LCD will be used to confirm the accuracy of the weight sensors for the following tests:

1) Weight is placed on the top left of the scale where the reticule is positioned on the top left of the LCD.
2) Weight is placed on the top right of the scale where the reticule is positioned top right of the LCD.
3) Weight is placed on the bottom left of the scale where the reticule is positioned on the bottom left of the LCD.
4) Weight is placed on the bottom right of the scale where the reticule is positioned on the bottom right of the LCD.
5) Weight is placed at the center of the scale where the reticule is positioned at the center of the scale.
6) Weight is placed in the middle of the first quadrant of the scale where the reticule is positioned in the middle of the first quadrant of the scale.
7) Weight is placed in the middle of the second quadrant of the scale where the reticule is positioned in the middle of the second quadrant of the scale.
8) Weight is placed in the middle of the third quadrant of the scale where the reticule is positioned in the middle of the third quadrant of the scale.
9) Weight is placed in the middle of the fourth quadrant of the scale where the reticule is positioned in the middle of the fourth quadrant of the scale.

7.2. Wireless Transmission Verification Testing

The wireless unit is evaluated by examining the packets on both the transmitting and receiving endpoints. Wireshark is a free and open source packet analyzer that can be used to evaluate the contents of packets. The packets are examined and analyzed by ensuring that the destination PAN of the transmitter matches the source PAN of the receiver and verifying that the data transmitted is correct. The following tests must be performed to verify functionality of the wireless unit:

1) Data is transmitted to receiver after measurement.
2) Data fails to transmit to receiver after 3 retries because the receiver is detached from the PC.
3) Data fails to transmit to receiver after 2 retries but transmits successfully on the third attempt.
4) Data transmits to correct receiver when another wireless receiver is attached to the PC.

7.3. Normal Case #1 – User complies to all instructions

User Input: The user steps onto the scale

Conditions: The user is standing on the centre of the scale with legs comfortably spread apart in a natural posture. The user will remain still throughout the calculation process until an audible sound is heard. User will then step off of the scale.

Expected Observations: The user’s weight and centre of balance are calculated and displayed on the LCD and transmitted wirelessly to a locally paired PC for storage and representation.

7.4. Normal Case #2 – User leans heavily on right side

User Input: The user stands on the scale when instructed to.

Conditions: The user has already powered on the scale, and when the scale instructs the user to stand on the scale, the user stands with his feet evenly spaced while leaning to the right as much as possible while still standing on the scale until a sound from the scale alerts the user that the weighing process is finished.

Expected Observation: The body weight of the user should be presented on the LCD display, followed by a graphical representation of the user's center of balance to be very skewed towards the right side.

7.5. Extreme Case #1 - Amount of time needed to take a measurement is not met because of user's inability to follow direction

User Input: The user steps off the scale before the MCU is finished performing it's calculations for weight distribution.

Conditions: The user has already powered on the scale by stepping on. The scale instructs the user to stand on the scale and wait for audible sound that indicates that the measurements are completed. The user steps off before this indication.

Expected Observations: The scale will sense that the user has stepped off of the scale before the measurements could be calculated. The MCU is put into a power-down state.

7.6. Extreme Case #2 - Someone Over the maximum weight allowed steps onto the scale

User Input: The user exceeding the weight limit steps onto the scale

Conditions: The user is standing in the center of the scale remaining still awaiting for the calculation. A distinct audible sound is heard and the user steps off the scale.

Expected Observations: The LCD on the scale indicates that the user has exceeded the weight limitations of the digital scale and is unable to perform the necessary calculations.

8. CONCLUSION

The details of our design and implementation are documented in this design specifications. Each element of our design has been carefully thought out and chosen. Development of our prototype will follow the specifications discussed in this document as closely as possible. This document clearly lays out milestones and objectives for the development of the scale prototype.

9. References

[1] World Health Organization, "Obesity and Overweight", 2010, http://www.who.int/dietphysicalactivity/publications/facts/obesity/en/

[2] Wikipedia, "Coronal Plane", 2010, http://en.wikipedia.org/wiki/Coronal_plane

[3] Wikipedia, "Sagittal Plane", 2010, http://en.wikipedia.org/wiki/Sagittal_plane

[4] FCC Rules, "Unlicensed Spread Spectrum radio systems", July 23, 1996, http://www.iprocessmart.com/techsmart/Spread%20Spectrum%20FCC%20Regulations.htm

[5] "Battery Life Calculator", 2010, http://jamesrbass.com/batterylife.htm

[6] ATMEL, “AVR2016: RZRAVEN Hardware User's Guide”, 2008, http://www.atmel.com/dyn/resources/prod_documents/doc8117.pdf

[7] ATMEL, “ATmega48PA/88PA/168PA/328P”, 2009, http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf

[8] MaxStream, “Xbee/XBee-PRO OEM RF Module”, 2007, http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Manual.pdf

[9] MaxStream, “Xbee/XBee-PRO OEM RF Module Antenna Considerations”, September 2005, http://www.maxstream.net/support/knowledgebase/article.php?kb=153

[10] Wikipedia, “IEEE 802.15.4-2006”, 2010, http://en.wikipedia.org/wiki/IEEE_802.15.4

[11] Wikipedia, “Zigbee”, 2010, http://en.wikipedia.org/wiki/Zigbee

[12] ATMEL, “AVR2016: RZRAVEN Hardware User's Guide”, 2008, http://www.atmel.com/dyn/resources/prod_documents/doc8117.pdf

[13] ATMEL, “ATmega48PA/88PA/168PA/328P”, 2009, http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf

[15] Arduino Playground, “Xbee Shield”, 2010, http://www.arduino.cc/playground/Shields/Xbee01

[16] Arduino, “Arduino Hardware”, 2010, http://arduino.cc/en/Main/Hardware

[17] Arduino, “Xbee Shield”, 2010, http://www.arduino.cc/en/Main/ArduinoXbeeShield

[18] Slattery C. & Nie M, “A Reference Design for High-Performance, Low-Cost Weigh Scales”, December 2005, http://www.analog.com/library/analogDialogue/archives/39-12/weigh_scale.pdf

[19] Wheatstone, “The Wheatstone Bridge”,2010, http://www.play-hookey.com/dc_theory/wheatstone_bridge.html

[20] Sparkfun Electronics, “Graphic LCD Serial Backpack Datasheet”, September 2009, http://www.sparkfun.com/datasheets/LCD/Monochrome/Corrected-SFE-0016-DataSheet-08884-SerialGraphicLCD-v2.pdf

[21] Distar Technology, “Specification for LCD Module DS-G160128STBWW”, 2010, http://www.sparkfun.com/datasheets/LCD/DS-G160128STBWW.pdf