

January 2012

Respiratory Monitoring System Based on the Thoracic Expansion Measurement

Fabiola Araujo Cespedes

University of South Florida, fabiola@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>



Part of the [American Studies Commons](#), and the [Electrical and Computer Engineering Commons](#)

Scholar Commons Citation

Araujo Cespedes, Fabiola, "Respiratory Monitoring System Based on the Thoracic Expansion Measurement" (2012). *Graduate Theses and Dissertations*.

<http://scholarcommons.usf.edu/etd/3953>

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Neonatal Respiratory Monitoring System Based on the Thoracic Expansion
Measurement

by

Fabiola Araujo Cespedes

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Stephen Sadow, Ph.D.
Andrew Hoff, Ph.D.
Andrew Raij, Ph.D

Date of Approval:
March 9, 2012

Keywords:
Wireless Network, RF modules, PIC Microchip Microcontroller, Remote Unit, Vital
Signal Measurement

Copyright © 2012, Fabiola Araujo Cespedes

DEDICATION

I dedicate this thesis in memory of Meredythe Scheflen (1926- Aug 23rd 2011), founder of the Universidad Evangelica Boliviana, first private university in Santa Cruz, Bolivia.

ACKNOWLEDGEMENTS

I would like to thank Dr. Andrew Hoff, Dr. Andrew Rajj and Dr. Stephen Sadow, for being part of the committee members for this Masters degree thesis. I want to give special thanks to Dr. Stephen Sadow, for being a great professor, academic advisor and provided guidance.

I would also like to thank my husband, Rodrigo, for his continued support and for his encouragement in the completion of this work and to my parents for having set an example in my life and having provided the academic foundation from the first moment of my life.

TABLE OF CONTENTS

List of Tables.....	iii
List of Figures.....	v
Abstract.....	viii
Chapter One: Introduction	1
1.1 Preliminary Information	1
1.2 Problem Statement	2
1.3 Previous Research.....	2
1.4 Proposed Design	4
1.5 Thesis Organization	5
Chapter Two: Respiratory Sensor PR2012.....	7
2.1 Generic Description of the Respiratory Sensor.....	7
2.2 Construction of the Respiratory Sensor.....	7
2.3 Features of the Respiratory Sensor.....	10
2.3.1 Physical Dimensions.....	10
2.3.2 Displacement.....	11
2.4 Basic Tests on the Respiratory Sensor	12
2.4.1 Voltage versus Distance.....	13
2.4.2 Varying Speed of the Displacement.....	15
2.4.3 Breaking Point.....	16
2.4.4 Holding Maximum and Minimum Values.....	17
2.5 Respiratory Sensor Features	18
2.6 Summary	18
Chapter Three: Hardware Design Of The Respiratory System.....	19
3.1 Generic Description of the Overall System.....	19
3.2 Description of the Respiratory Belt Unit.....	20
3.2.1 Generic Block Diagram Respiratory Belt Unit Description.....	20
3.2.2 Logic Block Diagram Respiratory Belt Unit Description.....	21
3.2.3 Respiratory Belt Unit Circuit Description.....	23
3.3 Description of the Central Monitoring System.....	29
3.3.1 Generic Block Diagram Description.....	29
3.3.2 Logic Block Diagram Description.....	30
3.3.3 Central Monitoring System Circuit Description.....	31
3.4 Criteria for Components Selection.....	33
3.5 Summary	34
Chapter Four: Programming Description	35
4.1 Master Microcontroller Programming.....	35

4.1.1	General Description of the Master Microcontroller Program.....	35
4.1.2	Subroutine “Update EEPROM”.....	37
4.1.2.1	Internal EEPROM Memory.....	37
4.1.2.2	Description of the Subroutine.....	38
4.1.2.2.1	Command 01: New Belt Registered.....	39
4.1.2.2.1.1	Subroutine “Store Baby ID”.....	39
4.1.2.2.1.2	Subroutine “Activate ID”.....	41
4.1.2.2.2	Command 02: Respiratory Belt Exit.....	42
4.1.2.2.3	Command 03: Reset List.....	45
4.1.3	Subroutine “Request Data from Remote IDx”.....	45
4.1.4	Subroutine “Microc_PC”.....	47
4.2	Remote Microcontroller Programming.....	48
4.3	Summary.....	49
Chapter Five: Analysis, Conclusions, Recommendations and Future Work.....		52
5.1	Tests Analysis.....	52
5.1.1	Description of Technique for Simulation.....	52
5.1.2	Tests Results Varying with Belt Tension.....	52
5.1.3	Tests Results Varying with Sampling Times.....	54
5.1.4	Tests Results Varying Amplitude Resolution.....	57
5.1.5	Rescaling Y-axis to Voltage and X-axis to Time.....	59
5.1.6	Frequency Calculations.....	61
5.1.7	Consistency in Sampling Time.....	64
5.2	Electrical Shock Hazard Safety.....	64
5.3	Conclusions.....	65
5.4	Future Work.....	66
References.....		68
Appendices.....		71
Appendix A: Assembler Code for Master Microcontroller.....		72
Appendix B: Assembler Code for Remote Microcontroller.....		82
Appendix C: Registers Configuration in Remote Microcontroller.....		87
Appendix D: MATLAB Code for Simulation Test.....		88
Appendix E: Simulated Test Results for Square Signal Conversion.....		90
Appendix F: Additional Simulated Test Results.....		97

LIST OF TABLES

TABLE 2.1: Relationship between displacement distance and average voltage.	14
TABLE 2.2: Measured respiratory sensor parameters.	18
TABLE C1: Register TRISB configuration in the remote microcontroller	87
TABLE C2: Register TRISA configuration in the remote microcontroller	87
TABLE C3: Register ANSEL configuration in the remote microcontroller	87
TABLE C4: Register ADCON1 configuration in the remote microcontroller.....	87

LIST OF FIGURES

FIGURE 2.1: First step in the respiratory sensor construction.....	8
FIGURE 2.2: Second step in the respiratory sensor construction.	8
FIGURE 2.3: Third step in the respiratory sensor construction.	9
FIGURE 2.4: Fourth step in the respiratory sensor construction.	9
FIGURE 2.5. Dimensions of parts of the model PR2012 respiratory sensor.	10
FIGURE 2.6. Graphic design of respiratory sensor in the starting position.....	11
FIGURE 2.7. Logic symbol of the respiratory sensor in the starting position.	11
FIGURE 2.8. Graphic design of the displaced respiratory sensor.	11
FIGURE 2.9. Logic symbol of the displaced respiratory sensor.	11
FIGURE 2.10. Connection of the respiratory sensor to the computer.....	12
FIGURE 2.11: Respiratory sensor with no displacement measured by the micrometer.	13
FIGURE 2.12: Respiratory sensor with displacement measured by the micrometer.	13
FIGURE 2.13: Sensor displacement and average voltage relationship.	15
FIGURE 2.14: Results while measuring sensor displacement at different speeds.	16
FIGURE 2.15: Test result to find breaking point at 4.844V.....	17
FIGURE 2.16: Minimum and maximum sensor displacement test.	17
FIGURE 3.1: Respiratory system individual monitoring mode.....	19
FIGURE 3.2: Respiratory system individual monitoring mode.....	20
FIGURE 3.3: Respiratory belt unit generic block diagram.	21
FIGURE 3.4: Respiratory belt unit logic block diagram.	23

FIGURE 3.5: Respiratory belt unit circuit.	24
FIGURE 3.6: Highlight of RA5 and individual mode interface connection.	25
FIGURE 3.7: Highlight of RB2/Rx pin connected only to D10 of the encoder.....	26
FIGURE 3.8: Dip switch connection to the encoders and decoders.	27
FIGURE 3.9: Dip switch bit open or closed.....	27
FIGURE 3.10: Highlight of RB2/Rx in the individual monitoring mode.....	28
FIGURE 3.11: Central monitoring system generic block diagram.	29
FIGURE 3.12: Central monitoring system logic block diagram.....	30
FIGURE 3.13: Central monitoring system circuit.....	32
FIGURE 3.14: Overall system logic block diagram.	34
FIGURE 4.1: Master microcontroller software main flow diagram.....	36
FIGURE 4.2: Master microcontroller EEPROM memory mapping example.	37
FIGURE 4.3: Update EEPROM memory subroutine.	38
FIGURE 4.4: Visual steps of the third level subroutine “store baby ID”.	39
FIGURE 4.5: Flow diagram of the third level subroutine “store baby ID”.	40
FIGURE 4.6: Flow diagram of the third level subroutine “Activate ID”.	41
FIGURE 4.7: First steps executed for eliminating an ID from memory.	42
FIGURE 4.8: Next steps executed for eliminating an ID from memory.....	43
FIGURE 4.9: Flow diagram of the third level subroutine “Eliminate ID”.	44
FIGURE 4.10: Flow diagram of subroutine “Reset list”.	45
FIGURE 4.11: Flow diagram of subroutine “Request data from remote IDx”.....	46
FIGURE 4.12: Flow diagram of subroutine “Microc_PC”.....	47
FIGURE 4.13: Remote microcontroller flow diagram.	48
FIGURE 4.14: First stage of program description	50
FIGURE 4.15: Second stage of program description.	51

FIGURE 5.1: Recording when the respiratory belt sensor is set too tightly.	53
FIGURE 5.2: Recording when the respiratory belt sensor is set too loosely.....	53
FIGURE 5.3: “Belt On Test” to determine correct tightening of belt.....	54
FIGURE 5.4: Test simulation using 102 samples in 12 seconds with time delay t1.....	55
FIGURE 5.5: Test simulation using 390 samples in 12 seconds with time delay t2.....	55
FIGURE 5.6: Test simulation using 1440 samples in 12 seconds with time delay t3.....	56
FIGURE 5.7: Test simulation using 306 samples in 36 seconds with time delay t1.....	56
FIGURE 5.8: Test simulation using 1170 samples in 36 seconds with time delay t2.....	57
FIGURE 5.9: Test simulation using 4420 samples in 36 seconds with time delay t3.....	57
FIGURE 5.10: Results with right justification in AD conversion.....	58
FIGURE 5.11: Results with right justification after physical adjustments.....	58
FIGURE 5.12: Simulated test 1 with new amplitude scale.	59
FIGURE 5.13: Simulated test 2 with new amplitude scale.	59
FIGURE 5.14: Simulated test scaled to time vs. byte value.	60
FIGURE 5.15: Simulated test scaled sample vs. voltage.	60
FIGURE 5.16: Assignment of lower and upper cut for square signal conversion.	61
FIGURE 5.17: Simulated test signal converted to square signal.....	61
FIGURE 5.18: Time elapsed calculation test.	62
FIGURE 5.19: MATLAB calculation of frequency for the corresponding test.....	63
FIGURE E1: Simulated signal test E1.....	90
FIGURE E2: Conversion to square signal test E1.....	90
FIGURE E3: Simulated signal test E2.....	91
FIGURE E4: Conversion to square signal test E2.....	91
FIGURE E5: Simulated signal test E3.....	92
FIGURE E6: Conversion to square signal test E3.....	92

FIGURE E7: Simulated signal test E4.....	93
FIGURE E8: Conversion to square signal test E4.....	93
FIGURE E9: Simulated signal test E5.....	94
FIGURE E10: Conversion to square signal test E5.....	94
FIGURE E11: Simulated signal test E6.....	95
FIGURE E12: Conversion to square signal test E6.....	95
FIGURE E13: Simulated signal test E7.....	96
FIGURE E14: Conversion to square signal test E7.....	96
FIGURE F1: Simulated signal test F1	97
FIGURE F2: Conversion to square signal test F1	97
FIGURE F3: Calculated parameters results test F1	98
FIGURE F4: Simulated signal Test F2.....	98
FIGURE F5: Conversion to square signal Test F2.....	99
FIGURE F6: Calculated parameters results test F2.....	99

ABSTRACT

The purpose of this research was to develop and a respiratory monitoring system using a reflective object sensor based belt to measure the thoracic expansion of a neonatal for future application at the medical center of the Universidad Evangelica Boliviana (UEB). This medical center, being founded by the UEB University, is dedicated to help and serve the poor, and currently has no respiratory monitoring system.

The methodology used was first to build and test the respiratory sensor belt and test the relationship between the belt expansion and the voltage generated. Then, to incorporate the respiratory sensor belt in a system that would allow individual testing as well as group testing in a wireless network. The system was simulated using an expandable plastic container that was expanded and extracted periodically, registering the results in a MATLAB software.

The system gave successful results and generated the frequency results of each cycle, average frequency and deviation frequency. The system demonstrated to be reliable, and to have repeatable results..

CHAPTER ONE: INTRODUCTION

1.1 Preliminary Information

Neonatal care monitoring is essential in the early detection of anomalies, and the rapid intervention in case of problems, preventing in many cases future complications and even death. With a recording feature added to neonatal monitoring systems, real time and recorded data enable the categorization and analysis of health information. In the United States, all hospitals with neonatal intensive care units are required to have monitoring equipment for newborns, as the Ministry of Health Guidelines for Hospitals with Neonatal Intensive Care Service not only requires its use, but specifies that “equipment used to monitor vital functions must have an alarm system that is operative at all times.” and even adds that “all life support and monitoring equipment shall be connected to an uninterruptable electrical supply” source⁽¹⁾. Nevertheless, in many developing countries, monitoring systems are limited to a few hospitals and medical centers that either have the technology and are overcrowded with lower economical class patients or are expensive and available only for the upper economical class.

According to the PLoS Medicine publication, research conducted in 193 countries over the period of from 1990 to 2009 showed that neonatal deaths (referring to the death of infants less than 28 days of birth) account for 41% of all child deaths. From all of the countries analyzed, only 38 countries had reliable data registration. ⁽²⁾ On January 14th, 2009, the World Health Organization estimated that 28% of neonatal deaths in the

United States are due to preterm births (babies born prior to 37 weeks), and that in developing countries, accurate and complete population data and medical records usually do not exist⁽³⁾. The research for the following project is intended for the medical center of the Universidad Evangelica Boliviana (UEB), which is a medical center constructed by the UEB university located five miles from the center of Santa Cruz, Bolivia, with no current neonatal respiratory monitoring system.

1.2 Problem Statement

The Universidad Evangelica Boliviana (UEB) University was founded by World Gospel Mission missionary Meredythe Schefflen in 1982, with the support of Evangelicals Friends Mission, United World Mission, South American Mission, Bolivian Holiness Mission, Church of the Nazarene in Bolivia and the Rio Nuevo Education Cooperative, being the first evangelical university in Spanish speaking South America. The first six academic areas included nursing, electronics, communications, English, music and theology. The UEB University operates its own medical center with operating rooms and maternity and child birth units, as well as its own education television station, education radio station and agricultural center. The medical center has limited resources and currently no respiratory monitoring system for early detection of abnormalities in new born babies.

1.3 Previous Research

The increasing demand for body vital signal measurement systems has enabled the development of a wide variety of respiratory measurement systems, including belts and wireless systems to cover care unit areas. Nowadays, two techniques are widely used for neonatal monitoring either based on techniques that measure thoracic or

abdominal expansion or based in measurement of changes in skin impedance. For the first technique, most systems use strain gauges made from a mercury filled Silastic™, which is a silicone elastomer that is extremely sensitive to movement. For the second technique, non-invasive skin electrodes are placed on the thorax, and the variation of the electrical impedance can be detected during respiration cycles. Other methods, such as nasal air flow sensors are not reliable and therefore not used for Neonatal Intensive Care Units⁽⁴⁾.

Texas Instrument developed the integrated circuit ADS1298R for the measurement of respiration rates based on impedance pneumography⁽⁵⁾. The technique consists of using two or four electrodes to measure the person's changes in electrical impedance in the thorax caused by respiration. Alternatives, such as the Respiratory Belt MR by Dr. Davide Riccobon, are built to make use of methods that are comfortable and non-intrusive in contrast to methods based on EEG and fMRI for measurement⁽⁶⁾. Hans Strasburger and Dieter Klenk, from the Max Planck Institute of Psychiatry developed an opto-electronic belt that measured the respiratory pattern⁽⁷⁾. The belt was based on an infrared-sensitive phototransistor and an infrared light emitting diode located on two sides of the belt. The distance of the light flow varied as the breathing of the patient separated the parts which were located on opposite sides of an elastic band. Brink, Muller and Schierz developed a contact free measurement technique for heart rate, respiration rate and body movements during sleep⁽⁸⁾. The method consisted of installing four high resolution force sensors under the bedposts. The heart rate movement and the lowering and lifting of the thorax during breathing causes small shifts in the center gravity of the bed, values which were used to measure the mentioned parameters. When compared this method with results using a strain gauge, it gave

comparable results. Nevertheless, if the amplitude changes during respiration are of interest, the method proved to be less reliable⁽⁸⁾.

The use of pyroelectric polymer (PEP) film as a transducer results in reliable and effective respiratory rate monitor, as fabricated by Dennis Dodds and Johnathan Purdy from the Department of Electronic Engineering, Bolton Institute and by and Chris Moulton by the department of Emergency Medicine University of Manchester.⁽⁹⁾ The respiratory sensor is cheap, disposable and has a high mechanical strength.⁽⁹⁾

The implementation of wireless transmission in the area of medical device research has resulted in the remote monitoring of patients. In the 2nd International Symposium on Medical Information and Communication Technology in 2007, the wireless belt project by the University of Oulu was conducted with a number of hospitals and they centered their development on a wireless belt to measure ECG, respiratory rate, tilt and temperature of a patient, sending the information to a computer over a wireless Zigbee data transmission link. The module was constructed using the 8051 microcontroller and a ZigBee transceiver that supported the IEEE 802.15.4 standard⁽¹⁰⁾. Jones, Deo and Lockyer from the Massachusetts Institute of Technology developed a wearable chest strap that incorporated a PPy Lycra strain gauge to detect respiratory rate and wrist strap with infrared light technology to detect heart rate based, connected to a wireless tactile control unit⁽¹¹⁾.

1.4 Proposed Design

The thesis goal is to design and construct a monitoring respiratory system for a neonatal care unit that is capable of working in a group monitoring mode using wireless transmission, as well as an individual monitoring mode using a dedicated wired connection to a computer. To complete the overall proposed design, the thesis is

divided into the following specific proposed designs: Design of a respiratory sensor belt, design of the hardware for the monitoring respiratory system and design of the software of the monitoring respiratory system.

The proposed respiratory sensor belt design is based on a reflective object sensor with output voltage variation depending on the thoracic or abdominal expansion during inhalation and exhalation of the baby. The design of the proposed hardware should enable the connection of the respiratory sensor to an individual computer or to a group central monitoring computer using a wireless transmission link. Finally, the design of the microcontroller software should be able to support individual as well group monitoring using a wireless connection.

1.5 Thesis Organization

Chapter 2 describes the design of a respiratory sensor based on a reflective object sensor and a displacement of a mechanism that depends on the expansion and compression of the thorax. The first part of the chapter includes a description of how the sensor is built and a description of the basic functional mechanism. The second part of the chapter includes tests conducted on the sensor manually, which was done to test the basic function of the respiratory sensor itself.

Chapter 3 is focused on the hardware design, describing the generic block diagram as well as the detailed circuitry of a respiratory belt unit and a central monitoring station. The chapter includes details of how the respiratory belt unit adapts to work as individual mode or as group monitoring mode.

Chapter 4 describes the software designed for the system. It includes a detailed description of the software routines and subroutines in assembly language for the microchip PIC microcontrollers. The first part of the chapter describes master

microcontroller programming and its interfacing programming to a computer as well as its wireless communication with the remote respiratory belt units. The second part of the chapter describes the remote microcontroller programming for sending the wireless data to the master microcontroller.

Chapter 5 describes the tests conducted using a simulated thorax and analyzing variations in sampling time, rescaling the amplitude resolution, rescaling the x-axis to time and the y-axis to voltage, as well as the method used to obtain the frequency of each cycle as well as the average frequency and the deviation of frequency. The chapter includes the analysis of the results, conclusions and recommendations for future research and development.

CHAPTER TWO: RESPIRATORY SENSOR PR2012

2.1 Generic Description of the Respiratory Sensor

The proposed respiratory sensor design will be referred to respiratory sensor PR2012 during the rest of the thesis document. The respiratory sensor PR2012 is based on a reflective object sensor that varies the output voltage depending on the displacement of an inner obstacle caused by the thoracic expansion during inhalation and exhalation of the patient. The respiratory sensor PR2012 is constructed to be connected to the respiratory belt unit, which is the name given to the complete hardware and respiratory sensor that enables registration of the respiration measurement of a patient. Details of the respiratory belt unit are discussed in the following chapter. This chapter gives detail of the proposed PR2012 respiratory sensor including its construction, features and specifications and the basic tests conducted during this thesis research.

2.2 Construction of the Respiratory Sensor

The respiratory sensor PR2012 is made of an immovable part attached to the edge a two sided adaptable size belt, a displaceable part connected to the other side of the belt that displaces on each thoracic expansion and a spring that forces the displaceable part to return to its initial position. The immovable part is composed of a plastic tube that acts as a container for the respiratory sensor and an optical based

sensor fixed on one end of the plastic tube. The displaceable part consists of an axis that moves in and out of the plastic tube and a dark barrier (referred to as “bumper edge” from this point on). Figure 2.1a shows a sketch of the sensor and its principle components.

The plastic tube container is open on both ends, with one of the open ends slightly smaller than the other. The axis is attached to a cylinder with a stop that contains a ring. The first step to assemble the sensor is to insert the axis into the smaller opening of the plastic tube as shown in Figure 2.1. The axis slides easily in and out, stopping at the stop that is attached to the ring with the opening of the plastic tube container.

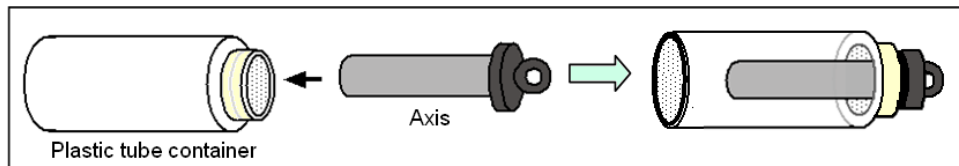


FIGURE 2.1: First step in the respiratory sensor construction. The axis is inserted in the plastic tube container and can easily slide in and out of the plastic tube container thus allowing for a means to measure axis displacement.

A spring is then inserted through the larger open end of the plastic tube container. The spring will wrap around the axis, still allowing the axis to move in and out of the plastic tube container as shown in Figure 2.2

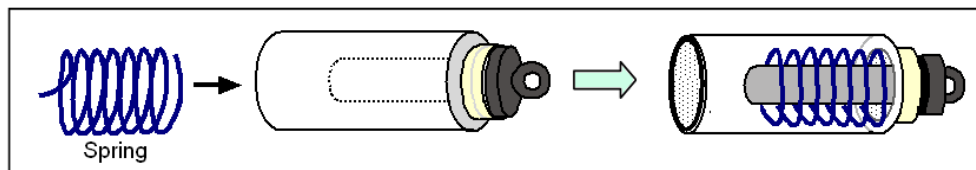


FIGURE 2.2: Second step in the respiratory sensor construction. A spring is inserted into the tube container with the axis as shown.

The bumper edge has one straight side and one hollow side. The hole fits by pressure in the axis. The bumper edge is inserted with the hole facing the open end of the plastic tube container. The bumper edge fits exactly through the open end and easily moves in and out. When the bumper edge reaches the axis, it is attached by

applying pressure. Figure 2.3 shows the third step of the construction as the bumper edge is inserted in the plastic tube container with the axis and the spring and is attached by pressure to the axis.

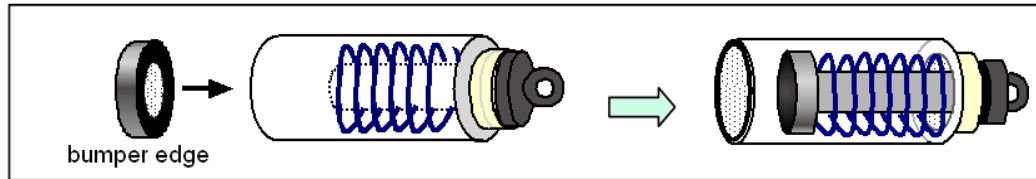


FIGURE 2.3: Third step in the respiratory sensor construction. The bumper edge is inserted and attached to the axis by pressure.

Finally, the reflective object sensor is inserted through the wider open end of the tube and is sealed in position as shown in Figure 2.4.

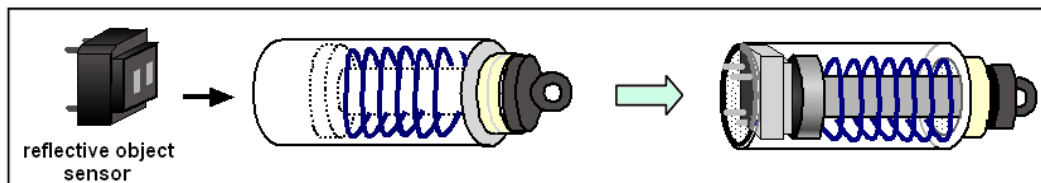


FIGURE 2.4: Fourth step in the respiratory sensor construction. The reflective object sensor is inserted and sealed.

The reflective object sensor (OPB606A from TT electronics - OPTEK Technology Inc.) consists of an infrared Light Emitting Diode (LED) and a PNP phototransistor mounted side by side on parallel axes, so the level of light detected by the phototransistor depends on the distance of the obstacle⁽¹²⁾. They are unfocused for sensing diffuse surfaces and encapsulated in a filtering epoxy to reduce ambient light noise. Therefore, as the thorax expands, the bumper edge moves away from the reflective object sensor, varying the reflective distance and as a consequence, the radiation amplitude detected by the phototransistor. The axis maintains the equilibrium of the bumper edge as it moves closer to the reflective object sensor or farther away from it, while the spring forces a gradual return of the bumper edge to its original position.

2.3 Features of the Respiratory Sensor

2.3.1 Physical Dimensions.

Each of the parts of the proposed model PR2012 respiratory sensor was built and measured using a vernier. Details of the physical dimensions of each of the respiratory sensor parts are shown on Figure 2.5.

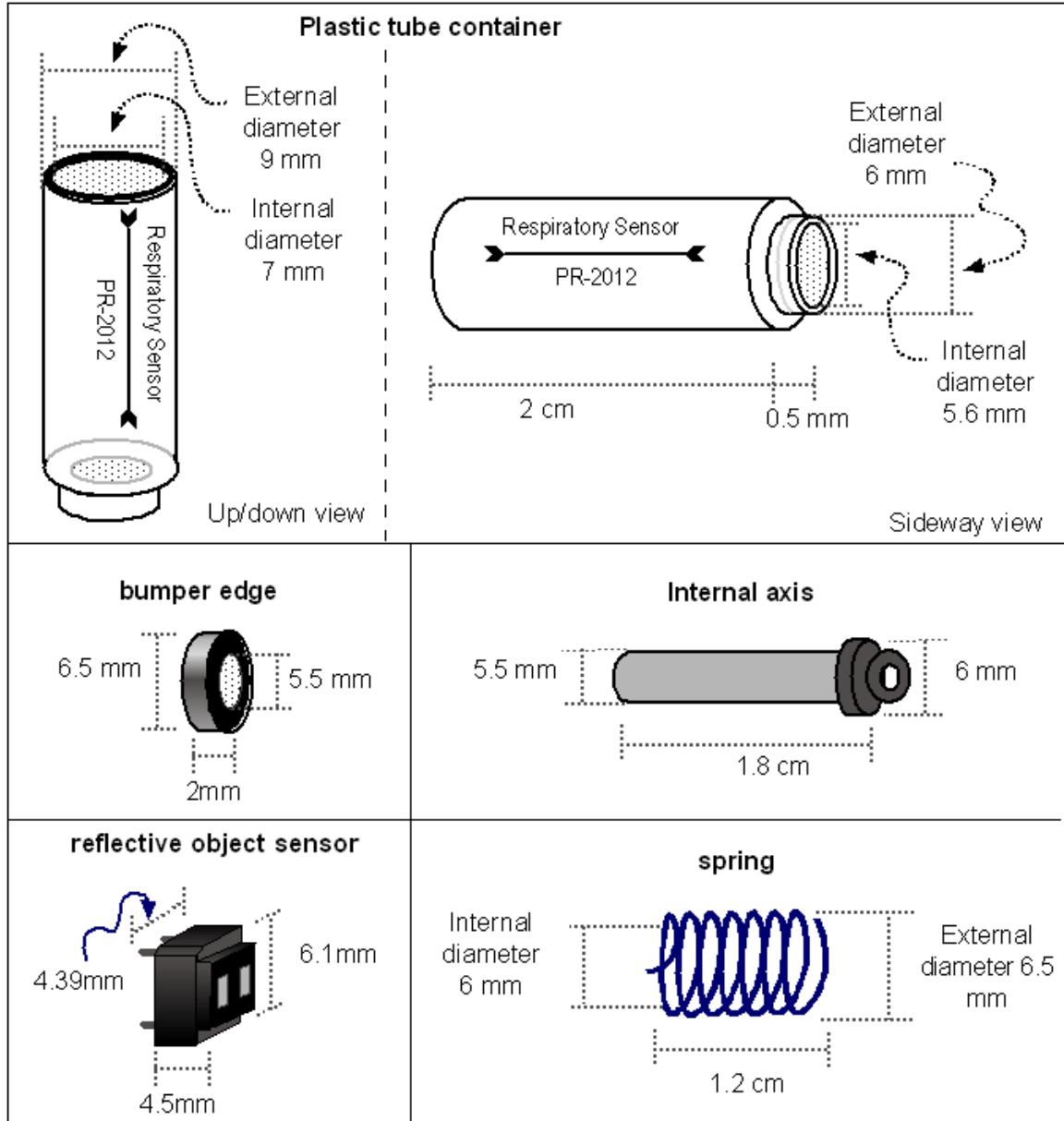


FIGURE 2.5. Dimensions of parts of the model PR2012 respiratory sensor.

2.3.2 Displacement.

As the original resting position of the respiratory sensor, the bumper edge is located 0.8 mm from the object reflecting sensor. This distance is denoted as the starting point distance. Visual designs for the starting point position are shown in Figures 2.6 and 2.7.

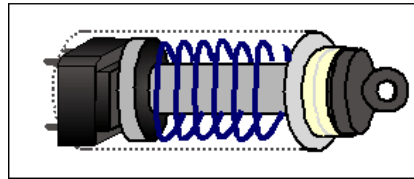


FIGURE 2.6. Graphic design of respiratory sensor in the starting position.

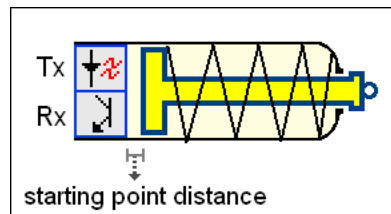


FIGURE 2.7. Logic symbol of the respiratory sensor in the starting position.

As the thorax expands, the bumper edge is pulled away, varying the distance from the starting point, that is related to the physical distance displaced. Schematic representations for displaced bumper edge are shown in Figures 2.8 and 2.9.

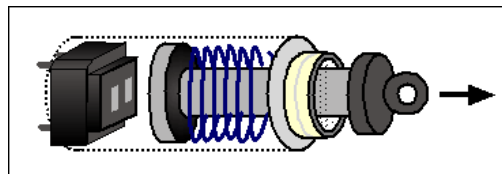


FIGURE 2.8. Graphic design of the displaced respiratory sensor.

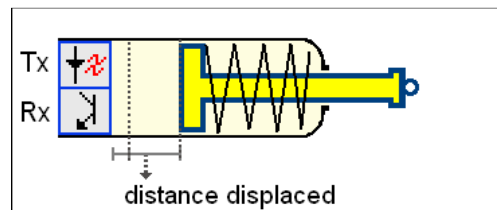


FIGURE 2.9. Logic symbol of the displaced respiratory sensor.

2.4 Basic Tests on the Respiratory Sensor

In order to conduct tests on the respiratory sensor, the LED and the phototransistor of the reflective object sensor were connected through resistors of 330Ω and 20KΩ, respectively to a voltage supply of 5V as shown in Figure 2.10. The analog voltage was connected to the microcontroller PIC16f88 analog input, converted to digital and sent to the computer using the RS-232 interface. The results were graphed in MATLAB for each test conducted.

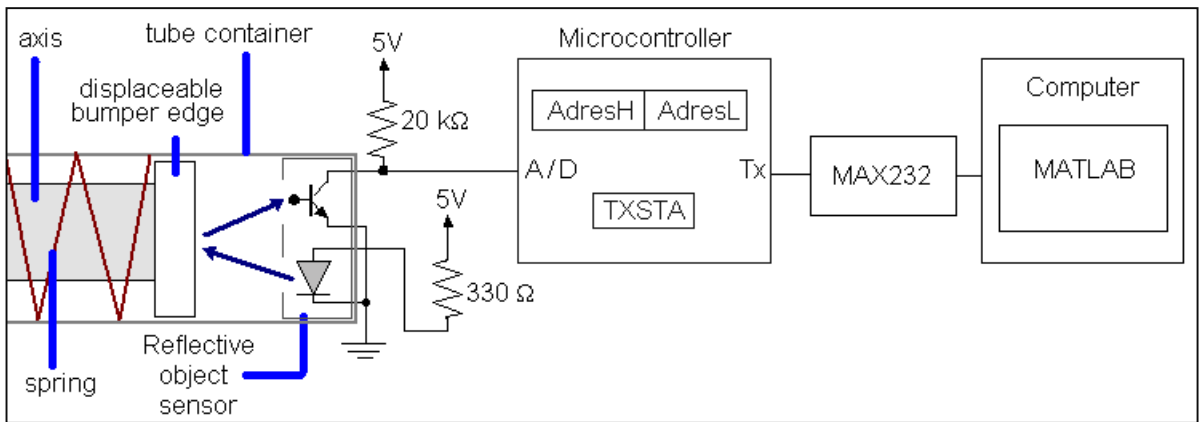


FIGURE 2.10. Connection of the respiratory sensor to the computer. The sensor was connected to the computer running MATLAB for testing purposes MATLAB was used to process the data so that a graphic plot could be displayed automatically.

The microcontroller was programmed for an analog to digital conversion with left justification, referring to the eight most significant bits of a 10 bit conversion being stored in register AdresH, and the least significant bits remaining stored in bits 6 and 7 of register AdresL. The two least significant bits were disregarded, and only the 8 bits stored in AdresH were considered.

The voltage varying in each case was calculated as shown below.

$$Voltage = \frac{Vmax}{2^{bits\ resolution}}$$

For 8 bit resolution: $Voltage = \frac{5V}{2^8} = 0.01953125V$

2.4.1 Voltage versus Distance.

The belt was then attached on one end and displaced on the second end using a crimp to hold in the sensor in place. A Mitutoyo micrometer was then calibrated with no displacement, as starting point (referred to as distance 0) as shown on Figure 2.11. The distance was displaced every 0.3 mm from the starting until the maximum distance of displacement was found, recording 360 samples. The micrometer the displaced distance as shown on Figure 2.12. The voltage for each distance and the corresponding distance were recorded.



FIGURE 2.11: Respiratory sensor with no displacement measured by the micrometer. The micrometer is calibrated to distance 0 as a starting point.



FIGURE 2.12: Respiratory sensor with displacement measured by the micrometer. The internal axis is displaced 0.16120 inches (4.0mm)

The software on MATLAB calculated the average displacement and standard deviation of each set of recorded distances and voltages. Table 2.1 shows the relationship between the displacement distances, the average of the total voltage samples recorded, the standard deviation of the samples, and the equivalent bit conversion.

TABLE 2.1: Relationship between displacement distance and average voltage.

Displacement distance (mm)	Voltage (V)	Standard deviation (V)	Bits conversion
0	4.1021	0.0176980	210
0.3	4.1494	0.0097246	212
0.6	4.1886	0.0097405	214
0.9	4.2152	0.0075679	216
1.2	4.2524	0.0157610	218
1.5	4.2925	0.0081673	220
1.8	4.3216	0.1639700	221
2.1	4.3458	0.0100990	223
2.4	4.4684	0.0158730	229
2.7	4.5173	0.0091679	231
3	4.5405	0.0097656	232
3.3	4.5983	0.0096919	235
3.6	4.6772	0.0098723	239
3.9	4.6971	0.0106120	240
4.2	4.7186	0.0131600	242
4.5	4.7470	0.0062947	243
4.8	4.7574	0.0120130	244
5.1	4.7743	0.1179300	244
>5.1	4.8440	N/A	248

The information was transferred to an excel file to graph and calculate the linear approximation using the “Add trendline” option. The data and linear approximation were plotted as shown on Figure 2.13. The least square fit linear equation of the voltage output as a function of displacement with R^2 equivalent to 0.981 was approximated to:

$$\text{Voltage output}[V] = 0.144(\text{displacement}[mm]) + 4.095$$

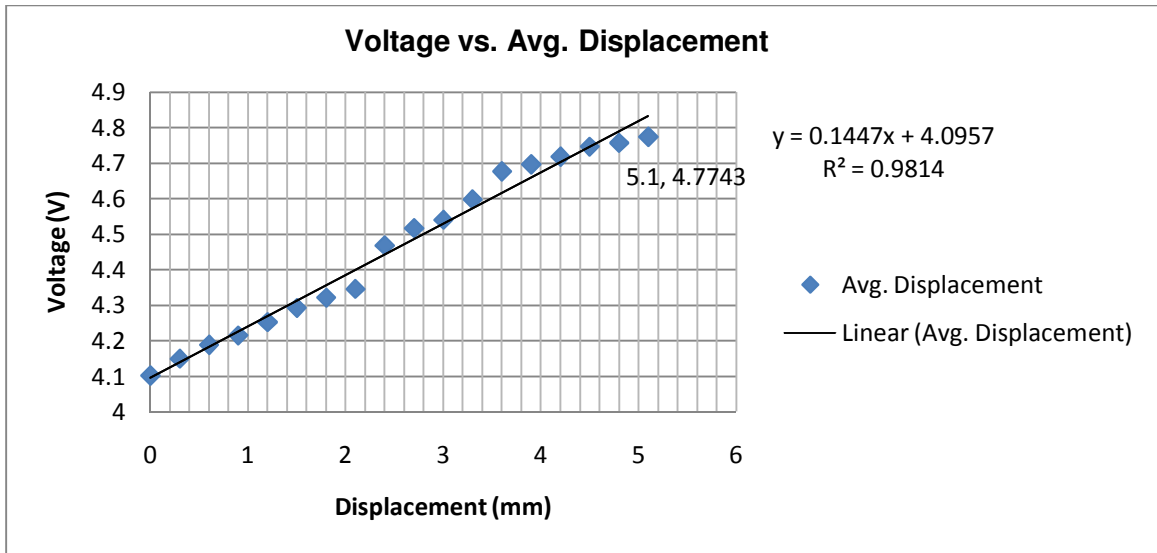


FIGURE 2.13: Sensor displacement and average voltage relationship. Based on measurements taken in the tests, the data is fairly linear. A least squares fit (solid black line) is shown along with the data (blue dots).

2.4.2 Varying Speed of the Displacement.

A second type of test was performed, testing the maximum and minimum voltages generated by the respiratory sensor as the speed of the displacement varied from resting point until the first tension point was detected. One side of the belt was attached to a desk and the other side was pulled using a manual rotating device in order to vary the speed as uniformly as possible. The rotating device had an LED indicator which would activate when an overload was detected, and which was used to indicate the first tension point. This operation point was then used as a stop indicator parameter.

The test was repeated a total of 20 times, during which different with different displacement speeds were tested. During the tests, the minimum was consistently measured to be close to 4.08V and the maximum slightly over 4.7574V, which was equivalent to 4.2mm of displacement. Figure 2.14 displays the pattern results of some of the recordings.

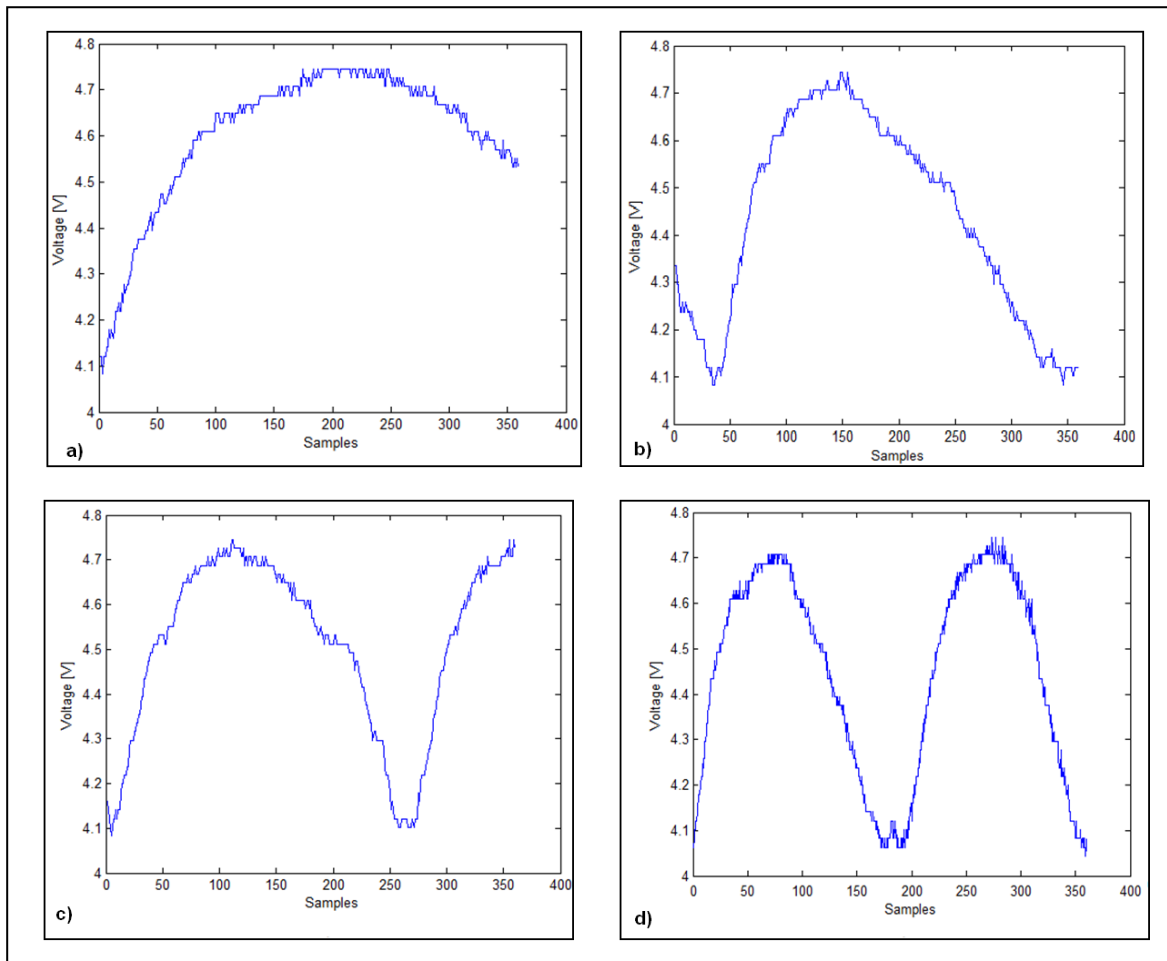


FIGURE 2.14: Results while measuring sensor displacement at different speeds.

- a) Continuous displacement from minimum to maximum at a rate of 1.05 mm/second
 b) at a rate of 1.68 mm/second from maximum to minimum and 1.4 mm/second from minimum to maximum.
 c) at a rate of 1.68 mm/second
 d) at a rate of x mm/second
 d) at a rate of 2.8 mm/second

2.4.3 Breaking Point.

As the internal bumper edge of the respiratory sensor was set further away from the distance point, the output voltage increased, resulting in a maximum at a displacement greater than 5.1mm. The tension point starts at around 4.7475V, and has a rapid increase to the breaking point at 4.844V. Figure 2.15 shows the result of the test conducted.

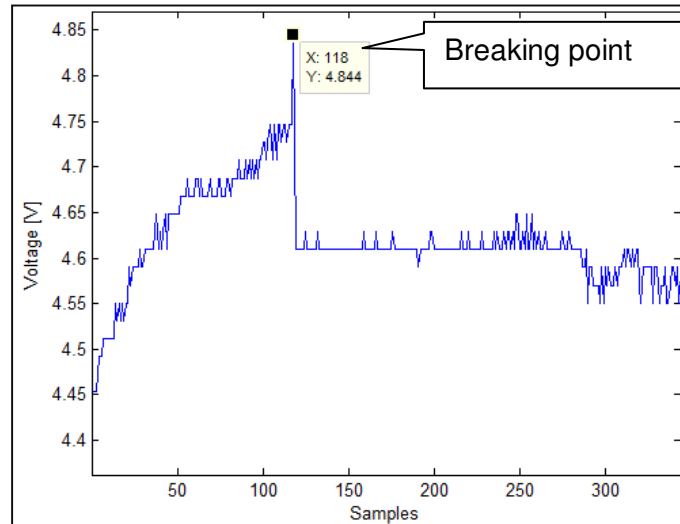


FIGURE 2.15: Test result to find breaking point at 4.844V.

2.4.4 Holding Maximum and Minimum Values.

The sensor was stretched from the initial position to the tension point where the sensor was then, held in position-while recording the voltage output. The results, shown in Figure 2.16, show that the minimum and maximum values are very consistent for their initial position point to the maximum displacement point where tension is first detected as was the case in the previous detected test.

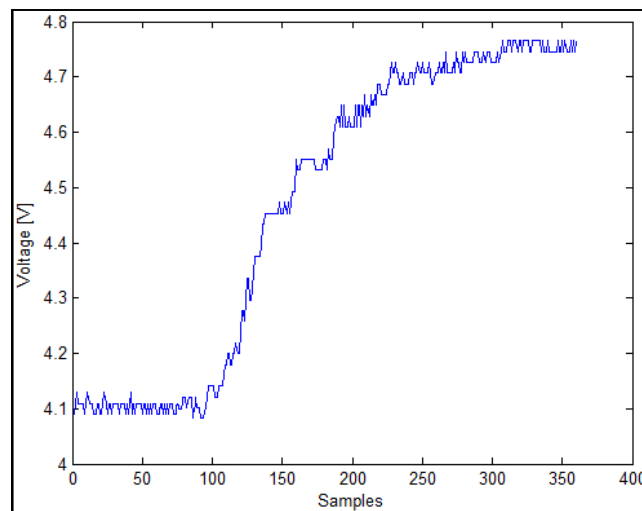


FIGURE 2.16: Minimum and maximum sensor displacement test. Sensor output test result when holding the displacement at the starting point and then varying the displacement to the breaking point.

2.5 Respiratory Sensor Features

In the following table, the main features of the respiratory sensor are listed:

TABLE 2.2: Measured respiratory sensor parameters.

Description	Value	Comment
Minimum voltage detected	4.08V	No displacement
Voltage at maximum recommended displacement	4.7574V	4.4 mm displacement
Voltage at breaking point voltage	4.844V	>5.1 mm
Weight	0.42 oz	Weight includes the respiratory sensor, 3 wires 50 cm long to connect to system, and belt. Weight excludes the hardware circuit.
Resolution	19.56mV	Resolution taken when connected to the microcontroller with 8 bit AD conversion
Pulling force	17N/m	A pulling force of 0.017N/mm stretched

2.6 Summary

In this chapter, the construction of the proposed respiratory sensor PR2012 is described, including the assembly steps to reproduce the sensor, the dimensions of the parts and features of the overall finished product. With the tests conducted on the product, we could conclude that the variation of the distance displacement is close to a proportional variation of the voltage, finding these two variables to have an approximation of a linear equation. With the tests conducted, the minimum voltage point was found at 4.08 and the maximum linearly operable voltage at 4.7574V, equivalent to a displacement of 4.2 mm. After this maximum voltage, the slope or the linear relationship changes sharply, until a breaking point to 4.844 in less than 1 mm of distance. For a proper use of the respiratory sensor, the maximum safe displacement to be used when measuring the thorax expansion is then concluded to be 4.2mm, range of operable distance.

CHAPTER THREE: HARDWARE DESIGN OF THE RESPIRATORY SYSTEM

3.1 Generic Description of the Overall System

The neonatal respiratory system is a complete respiratory monitoring system designed to allow two types of operating mode: Individual monitoring mode or group monitoring mode. When selected as individual mode, the respiratory belt unit is connected to an individual monitoring computer using the individual mode interface in order to have the baby's respiration pattern displayed and analyzed by an individual computer. Figure 3.1 shows the individual monitoring mode. This mode is intended for circumstances in which the baby is not in the neonatal care unit and instead, is at the mother's room or is being held temporarily aside from the other babies.

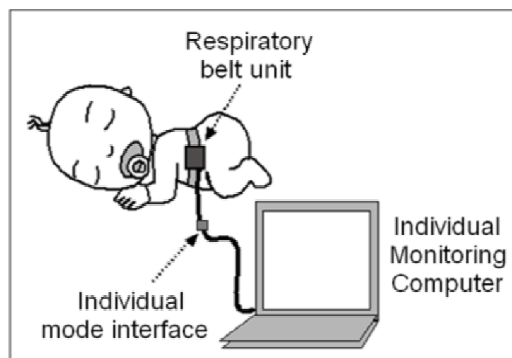


FIGURE 3.1: Respiratory system individual monitoring mode. The respiratory belt unit is connected to an individual monitoring computer using a serial port cable.

When selected as group monitoring mode, each respiratory belt unit is given a unique ID. This unique ID should be registered in the central monitoring computer to enable it to be connected to the network. Once the ID is entered, a master

microcontroller connected to the central monitoring computer is in charge of requesting data from the respiratory belt unit and sending and receiving data to and from the computer. Therefore, the central monitoring system can be described as being in two parts: the central monitoring computer where the information will be analyzed and displayed and the master microcontroller in charge of synchronizing the sent and received information to and from each respiratory belt unit and sending this information to the central monitoring computer. Figure 3.2 shows the group monitoring mode.

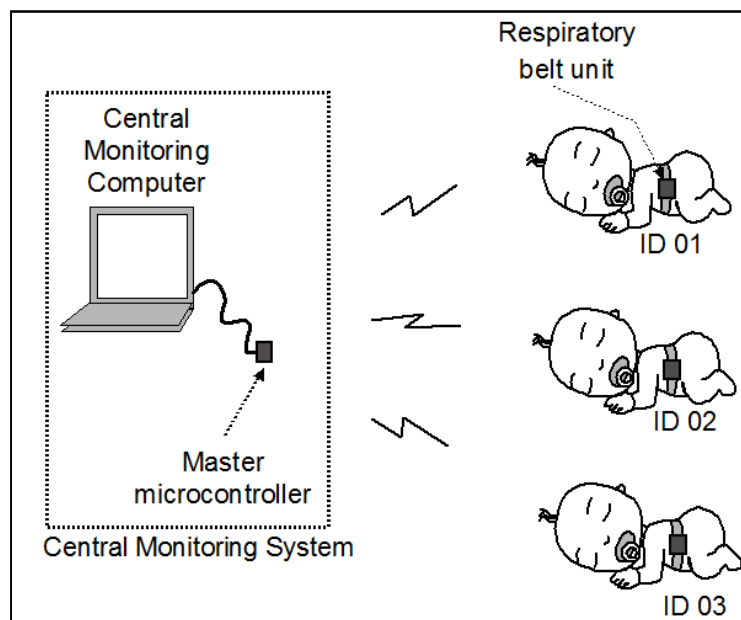


FIGURE 3.2: Respiratory system individual monitoring mode. The respiratory belt unit is connected to the central monitoring computer by a wireless network.

3.2 Description of the Respiratory Belt Unit

3.2.1 Generic Block Diagram Respiratory Belt Unit Description.

The Respiratory Belt Unit refers to the belt that has attached the respiration sensor and the electronic circuit that allow the belt to operate in the individual or group mode. The belt should first be placed around the baby's thorax in order for the respiration sensor to detect the respiration based on the thoracic circumference expansion. The analog data of the respiratory sensor PR2012 is the input of an analog

to digital converter of the microcontroller. This information is either sent to the individual monitoring computer using the individual mode interface or sent to the group monitoring computer wirelessly depending on the “Mode Selection” input. The mode of operation is selected by an input in the microcontroller. An “Alert” output indicates data being input from the belt out of range and the output “connected When the system is connected wirelessly, an output indicator “Connected to System Indicator” is high. When the system has detected a problem, the output “Alert” is high. Figure 3.3 shows the block diagram of a respiratory belt unit with the two alternative monitoring mode.

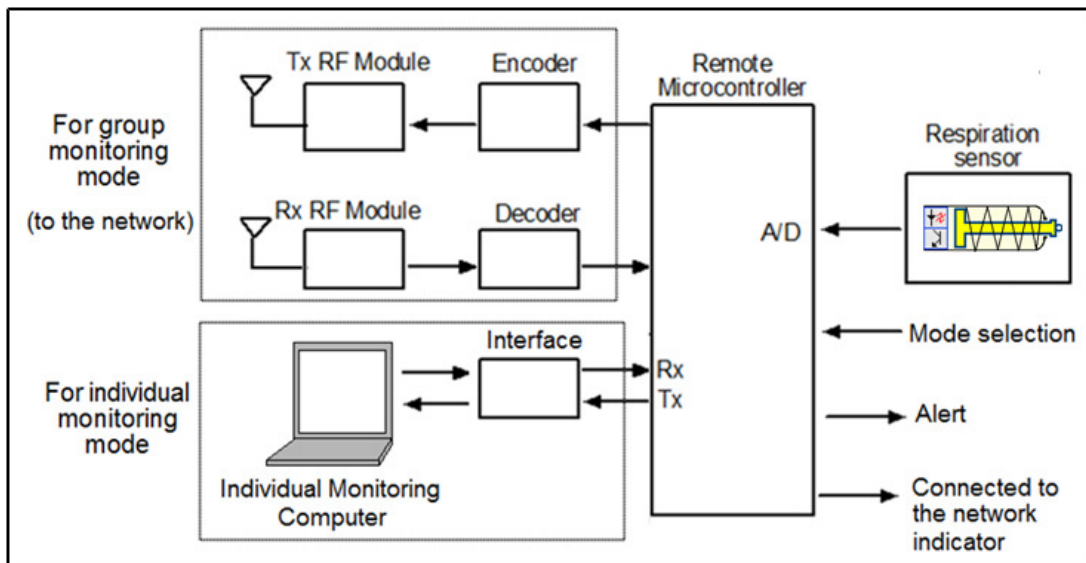


FIGURE 3.3: Respiratory belt unit generic block diagram. An input in the microcontroller “mode selection” selects either the group monitoring mode or the individual monitoring mode.

3.2.2 Logic Block Diagram Respiratory Belt Unit Description.

For the remote microcontroller, the respiratory belt unit uses the microcontroller PIC 16F88 from microchip. The respiratory sensor signal enters the RB7/AN6 input of the microcontroller configured as an analog to digital input. The mode selection is defined by the digital input RA5. A low input in pin RA5 indicates group monitoring mode, which connects the unit wirelessly to the network. In this mode, microcontroller will send the information of the respiratory sensor input to the encoder through pins RB0-

RB4 and sent to the RX RF module to the central monitoring system. The microcontroller will receive information from the central monitoring system through pins RA0 to RA3 validated by the VT signal from the decoder through pin RA7. A high input in pin RA5 indicates the individual monitoring mode and the information from the respiratory sensor input will be sent through the serial output using pins Tx and Rx. Pin RB2/Rx is the RB2 multiplexed with the Rx signal. This pin is used in both the group monitoring mode and individual monitoring mode and is configured by software the mode of operation depending on the input mode selected in pin RA5.

The digital outputs RA6 and RB6 indicate respectively a signal of alert when abnormal respiration rhythm is detected and a signal when the microcontroller is connected to the network.

For encoders and decoders, the respiratory belt unit uses the HT-12E and HT-12D respectively, from Holtek. For both, pins from D8 to D11 are connected to the microcontroller to send and receive information, as described previously and pins from A7 to A0 are connected to a dip switch to determine the address configuration for the given decoder or encoder. All of the respiratory belt units have the same given address for the decoder since all the stations send their information to the same central monitoring system. Nevertheless, each respiratory belt unit has a unique address for the encoder since each station will receive information from the central monitoring system that is intended for only that remote station. The decoder will send the information to the RF TX module TLP 434 through its pin Dout, and the encoder will receive the information from the RF RX module RLP 434 through its pin Din.

For the interface to the individual monitoring computer, the respiratory belt unit uses the MAX232 for a RS232 serial port interface. Pin Tx from the microcontroller is connected to pin Tin from the MAX232, which adapts the voltage level to connect to the

computer's Rx serial port. RB2/Rx is configured as Rx and connected to pin Rout of the microcontroller, which adapts the voltage level to connect it to the computer's Tx serial port. Figure 3.4 shows the respiratory belt unit logic block diagram with details of the logic connection between the components.

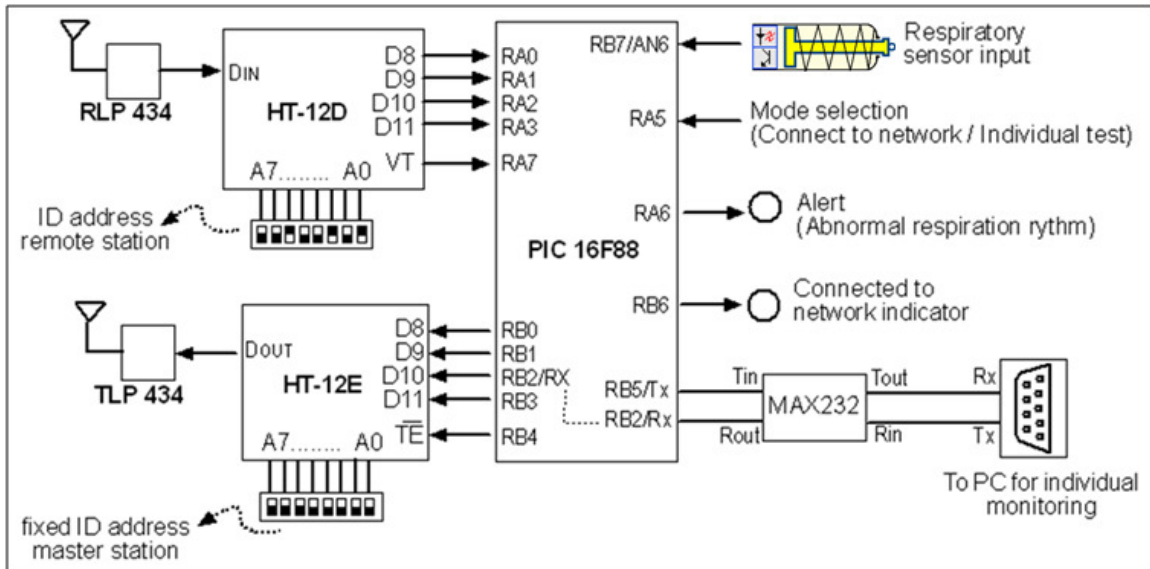


FIGURE 3.4: Respiratory belt unit logic block diagram. Pin RA5 defines the mode selection as individual or as group mode. Pin RB2/Rx is used in both modes of operation.

3.2.3 Respiratory Belt Unit Circuit Description.

The respiratory belt unit circuit can be observed in Figure 3.5. The respiratory sensor circuit is based on an infrared sensor. Details of the complete respiratory sensor are described in the previous chapter. The signal from respiratory sensor PR-2012 enters the pin RB7/AN6 of the microcontroller, which is configured as analog input. Pins RA6 and RB6 of the microcontroller are connected to LED since both outputs are “Alert” indicator and “Connected to the system” indicator.

The mode of operation is selected based on the digital input of RA5 which will depend whether or not the individual mode interface is connected by the 6 pin min DIN PCB. As can be observed, when the individual mode interface is not connected, RA5

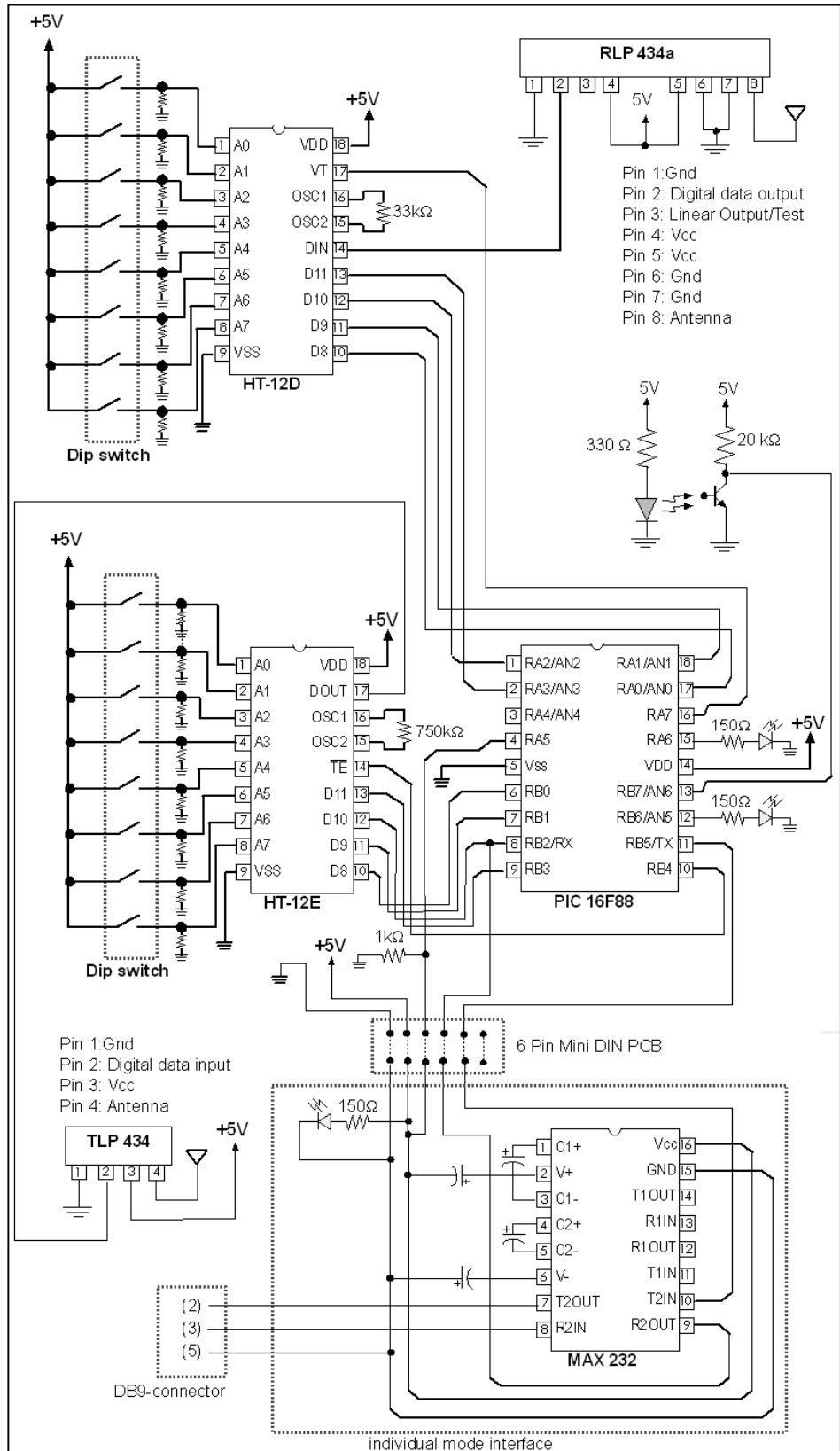


FIGURE 3.5: Respiratory belt unit circuit.

has a low input, selecting this way the group monitoring mode and when the individual mode interface is connected, a physical path of voltage is set that is input on the RA5 pin, selecting this other way the individual monitoring mode. Figure 3.6 shows how RA5 has either a low input or a high input depending on whether the individual mode interface is connected or not connected.

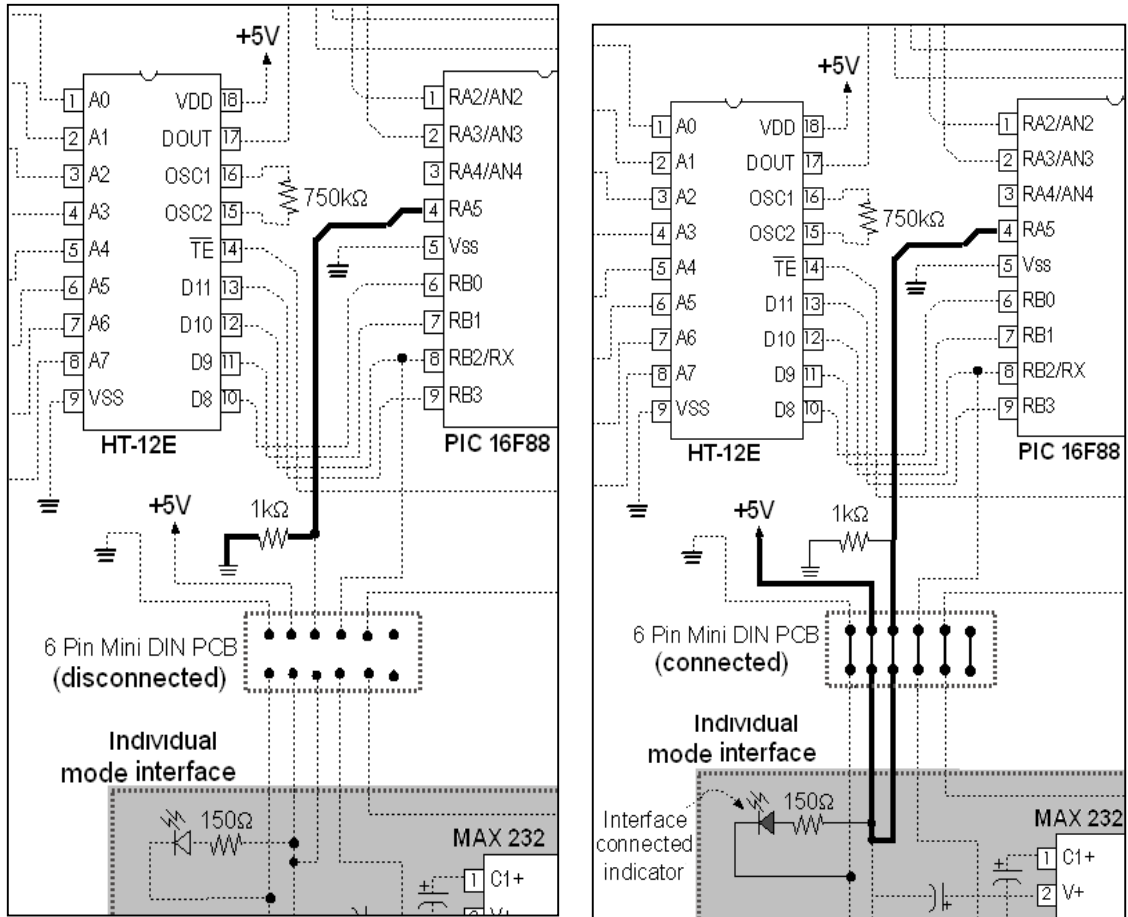


FIGURE 3.6: Highlight of RA5 and individual mode interface connection. The RA5 is low when the individual mode interface is disconnected and high when the individual mode interface is connected. Refer to Figure 3.4 for the complete circuit diagram.

When the system is selected in group monitoring mode, the information is sent from the microcontroller to the central monitoring system using the encoder HT-12E and is received from the central monitoring system using the decoder HT-12D. The wireless

signal from the central monitoring system enters the respiratory belt unit through the antenna input (pin8) of the RX RF module RLP 434a, and is connected to the pin DIN of the decoder HT-12D by the digital data output (pin2) of the module. When a data is received, the HT-12D generates a pulse in the VT output, which is connected to the microcontroller digital input RA7, and the data from the decoder's pin D8 to D11 is connected to the digital inputs RA0 to RA2.

Similarly, the signal is sent to the central monitoring system from the microcontroller's digital output RB to RB3 to the encoder's pin D8 to D11. Everytime a data is sent, the microcontroller generates a low pulse in RB4, which is connected to the encoder's pin TE/. The encoder transmits the information from pin DOUT to the RF TX module pin 2 (digital data input), and the information is sent wirelessly to the network through the antenna pin 4. As can be highlighted in Figure 3.7, the microcontroller pin RB2/RX is only physically connected to the encoder when the individual interface mode is disconnected.

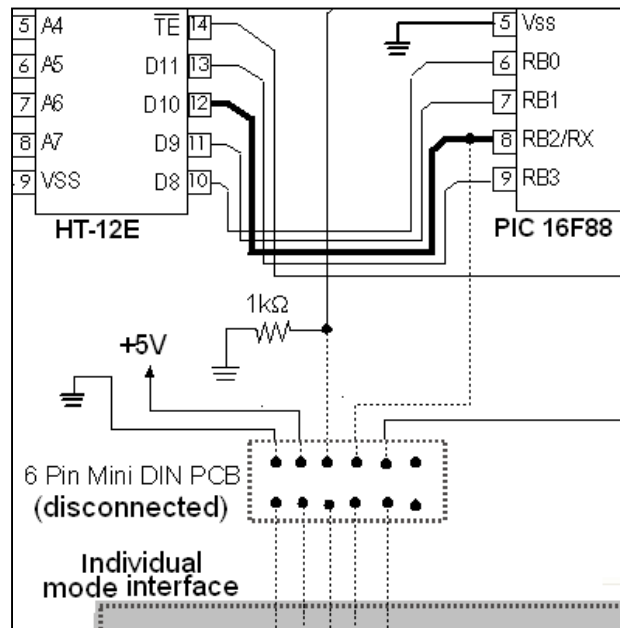


FIGURE 3.7: Highlight of RB2/Rx pin connected only to D10 of the encoder. The pin RB2/Rx pin of the microcontroller is connected only to D10 when the individual mode interface is disconnected. Refer to Figure 3.4 for the complete circuit diagram.

The address output A0 to A7 of the decoder HT-12D and the encoder HT-12E are each connected to an eight bit dip switch as shows Figure 3.8.

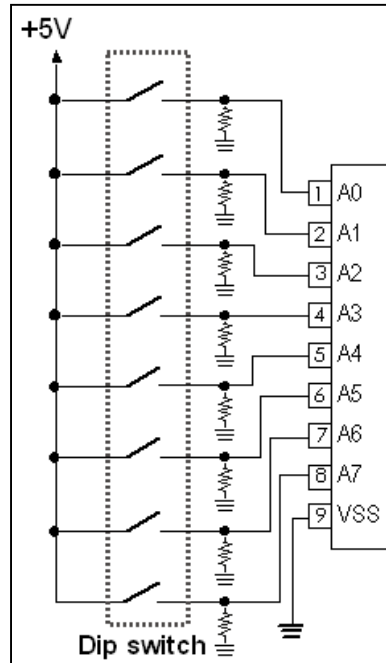


FIGURE 3.8: Dip switch connection to the encoders and decoders. Address A0 to A7 from the encoder and decoder are connected to a dip switch to enable a logic high input or low input. Refer to Figure 3.4 for the complete circuit diagram.

Each dip switch bit is set to zero (when the switch is open) or to one (when the switch is closed) as shows the Figure 3.9. The decoder of each respiratory belt unit is given a unique address, while all of the encoders of the respiratory belt unit have the same address, which coincides with the address of the decoder of the central monitoring system.

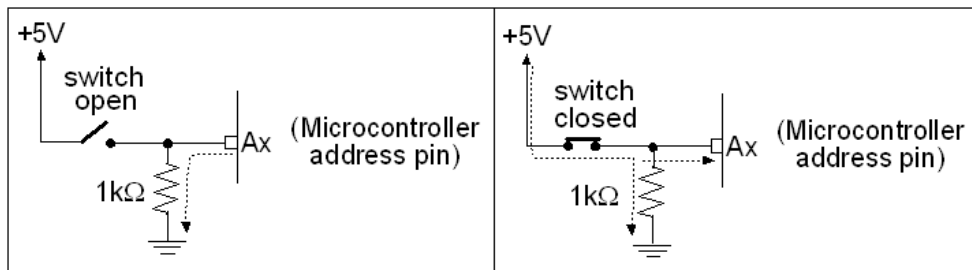


FIGURE 3.9: Dip switch bit open or closed.

The individual mode interface connection also physically sets a circuit path to turn on the “Individual interface mode indicator” as ON. The voltages of the microcontroller pins Tx and Rx are adapted to the computer RS-232 voltages. The pin TX of the microcontroller is connected to T2IN of the MAX232 and the adapted voltage output T2out is connected to pin 2 of the DB9 connector. The pin Rx of the microcontroller is connected to R2OUT of the MAX232 and the adapted voltage is R2IN, which is connected to pin 3 of the DB9 connector.

3.3 Description of the Central Monitoring System

3.3.1 Generic Block Diagram Description.

The central monitoring system is designed to send and receive information from and to each respiratory belt unit using a wireless connection. The information is received by the RX RF module and send to a decoder in order to send the digitalized decoded data to a master microcontroller. Similarly, the information the microcontroller sends to the respiratory belt units are sent first to an encoder and then to the TX RF Module. The master microcontroller is connected to the central monitoring computer through an interface that adapts the logic state voltages.

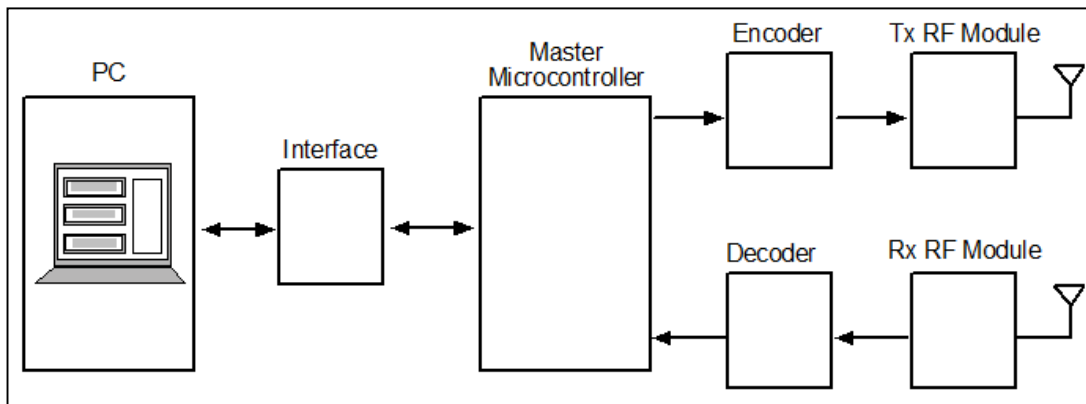


FIGURE 3.11: Central monitoring system generic block diagram. The master microcontroller is connected to an encoder and decoder to send and receive information to an from the respiratory belt units, and also connected to a central monitoring computer.

3.3.2 Logic Block Diagram Description.

In the logic block diagram of the central monitoring system from Figure 3.12 we observe the decoder HT-12D has a fixed address, while the encoder HT12E has an address that is defined by PORTB of the microcontroller. This way, the central monitoring unit can have wireless communication with several stations by changing the address by software. The decoder data D8 to D11 is connected to the microcontroller RA0 to RA3 and VT is connected to RA4. The encoder data D8 to D11 is connected to the microcontroller RD0 to RD3 and TE/ is connected to RD4. The microcontroller is connected to the central monitoring computer through a serial port using the MAX232 as interface. Tx of the microcontroller is connected to Tin of the MAX232. The voltages are adapted and the output Tout is connected to Rx of the computer's serial port. RX of the microcontroller is connected to Rout of the MAX232 and the voltages are adapted. Rin of the MAX232 is connected to Tx of the computer's serial port.

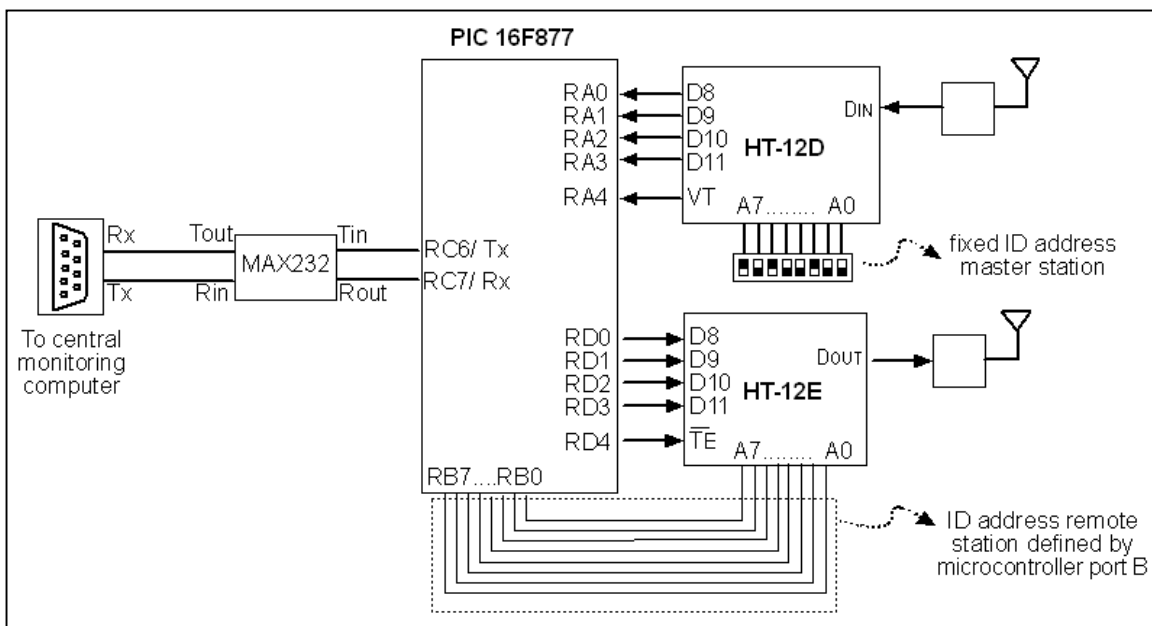


FIGURE 3.12: Central monitoring system logic block diagram. The microcontroller PIC16f877 is connected to the decoder HT-12d and HT-12e to send and receive information. The decoder address is fixed, while the encoder address depends on the microcontroller's PORTB.

3.3.3 Central Monitoring System Circuit Description.

Figure 3.12 shows the central monitoring system circuit. The connections between the decoder HT12d and the RX RF module RPL 434 and the encoder HT12e and the TX RF module TLP 434 are similar to the connections in the respiratory belt unit. As in the previous configuration, HT12e and HT12d have resistances in the oscillator pin inputs of $750\text{K}\Omega$ and $33\text{K}\Omega$ respectively.

The dip-switch connected to A0 to A7 from the HT12d defines the master microcontroller receiving address. All of the remote respiratory belt units should have exact address in their encoder units to enable the transmission from the remote units to the master unit. The master microcontroller transmitting address is defined by port B of the microcontroller allowing the microcontroller to send wireless information to different remote units depending on the program port B output.

Unlike the respiratory belt unit, whose microcontroller PIC16F88 used an internal oscillator, the microcontroller PIC 16F877 in the central monitoring system requires an external oscillator. It is connected to a 4MHz crystal oscillator connected to inputs 13 and 14 (OSC1 and OSC2) and two capacitors of 22pF. Pin 1 of the microcontroller is the master clear entrance, resetting each time there is a low input. As can be observed in Figure 3.13, when the reset push button is not pressed, the pin receives +5V. When the reset push button is pressed, the +5V is derived to ground and the pin receives a low input.

As interface to the central monitoring computer, the microcontroller is connected to the MAX232 using pins RX and TX. These pins are only connected to the interface and are not multiplexed to be used for any other function. The serial port voltages of the microcontroller are then adapted to the RS-232 serial port voltages required to transmit and receive information to and from the central monitoring computer.

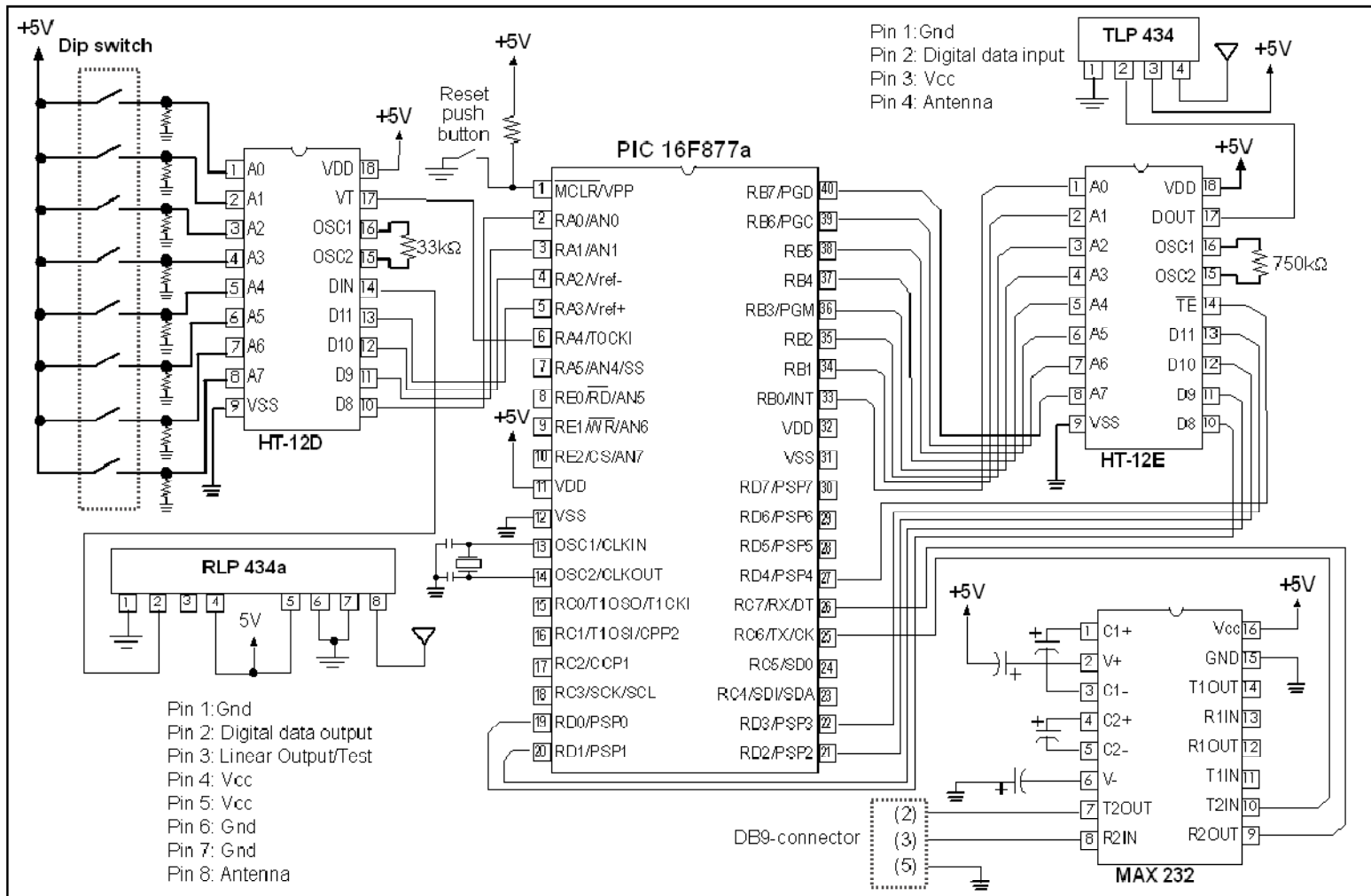


FIGURE 3.13: Central monitoring system circuit.

3.4 Criteria for Components Selection

In order to select the electronic components, price and commercially accessible devices for Bolivia were taken into consideration, as well as components that were able to be connected between them.

For RF module, the selected modules were the RLP434 and TLP434 from FUTURELEC. Their low price compared to other similar devices, and the characteristics made these ideal for the design and implementation. Both operate using ASK modulation at a frequency range of 433.92 Mhz. This frequency is an unlicensed band and has a range of up to 1000 feet. The RLP operate at a data rate of maximum 4800bps and channel spacil of +/-500Khz, and a supply voltage of 5V. The TLP434 operate at a data rate of maximum 8Kbps and a supply voltage of 3 to 12V.

The microcontrollers, from MICROCHIP were selected taking into consideration price, commercial availability, and characteristics of the microcontrollers. For the remote units to be applied in the belts, the microcontroller selected was the PIC16F88, which is a midrange PIC microcontroller of 8 bits with 16 bits digital input/output, 7 of which can be analog inputs (AN0, AN1, AN2, AN3, AN4, AN5, AN6), with internal oscillator and with serial interface. For a future design, other analog inputs could be used for other sensing parameters input. For the master unit, the microcontroller selected was the PIC16F877 having additional input/output ports the microcontroller and Rx/Tx port, and for being a microcontroller widely used in Bolivian electronic made applications, making the commercial price lower than other microcontrollers with similar characteristics.

The encoders/decoders selected for the application were the HT12D and HT12E mainly for being commercially available for a low price compared to other similar devices.

3.5 Summary

In summary, the respiratory system is composed by two basic subsystems: the respiratory belt units and the central monitoring system. The respiratory belt unit can be selected to work independently, as an individual unit, in which the information is displayed in an independent and individual computer, or can be selected to work as part of a network, in which the information is displayed in a central monitoring computer. Figure 3.14 shows an example, in which respiratory belt units 1, 2 and 3 are working in the network mode, and have the connected indicator output ON. Each is distinguished by a unique address in the decoder. Respiratory belt unit 4 is working in the individual mode and the information is display in an independent computer.

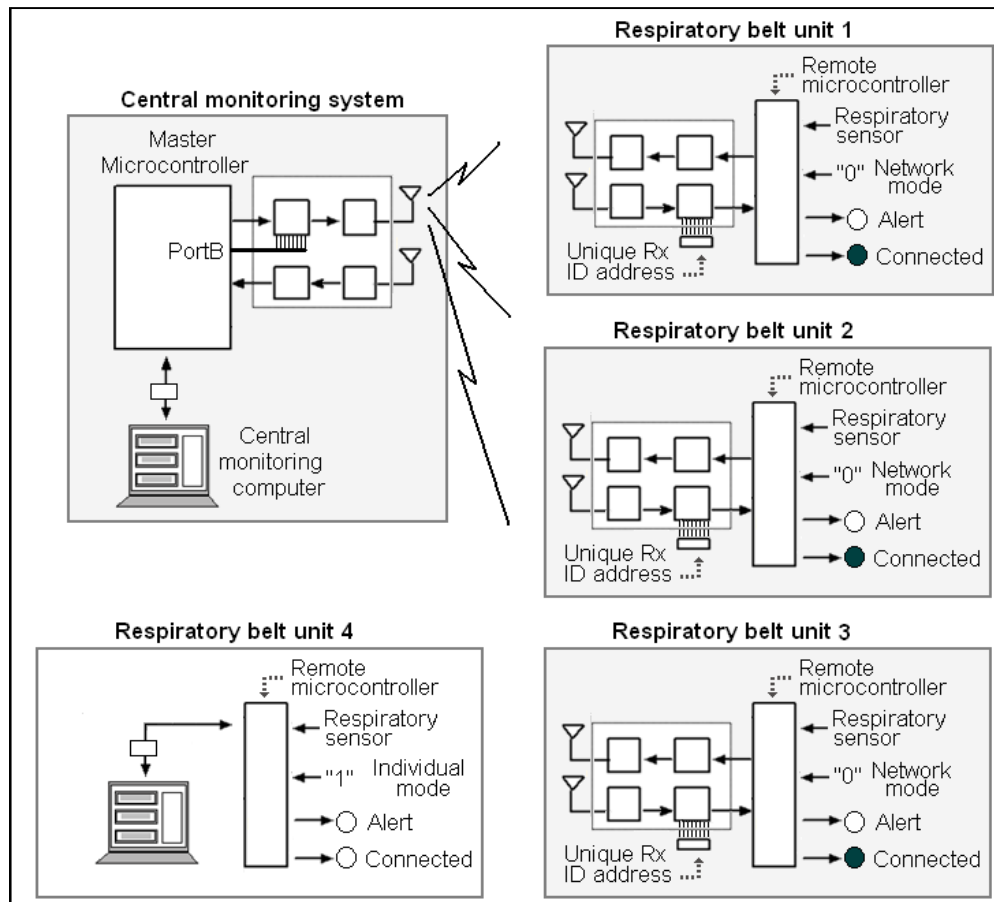


FIGURE 3.14: Overall system logic block diagram. Three respiratory belt units working in network mode are connected to the central monitoring computer, while one respiratory belt unit is working individually.

CHAPTER FOUR: PROGRAMMING DESCRIPTION

4.1 Master Microcontroller Programming

4.1.1 General Description of the Master Microcontroller Program.

The synchronism of the wireless communication between each respiratory belt unit and the central monitoring computer is achieved by the master microcontroller's program using information stored in its internal EEPROM.

At the beginning of the program, the microcontroller receives information from the central monitoring computer. This information contains the number of respiratory belt units registered and the ID of each one, and stores this information in EEPROM memory. If there are no belts registered, the microcontroller will remain in a monitoring until there are belts registered and stored in the EEPROM. Using the ID address of the belt, the microcontroller proceeds to request and receive respiratory measurement data from the unit. The data received is then sent to the central monitoring computer for analysis and sends an alert signal of alert in case of an abnormality. Once this task has been completed, the program proceeds to repeat the same task for the next ID registered, repeating this task for each ID stored in memory. Figure 4.1 shows the flow chart of the master microcontroller program and a detailed explanation next to each block.

In the program, the variable n indicates the number of belts registered and the variable x is used as an address pointer or address location count. Each address

contains the baby's ID. The program increments the variable x from 0 to n and repeats the task of requesting respiratory data and sending it to the computer.

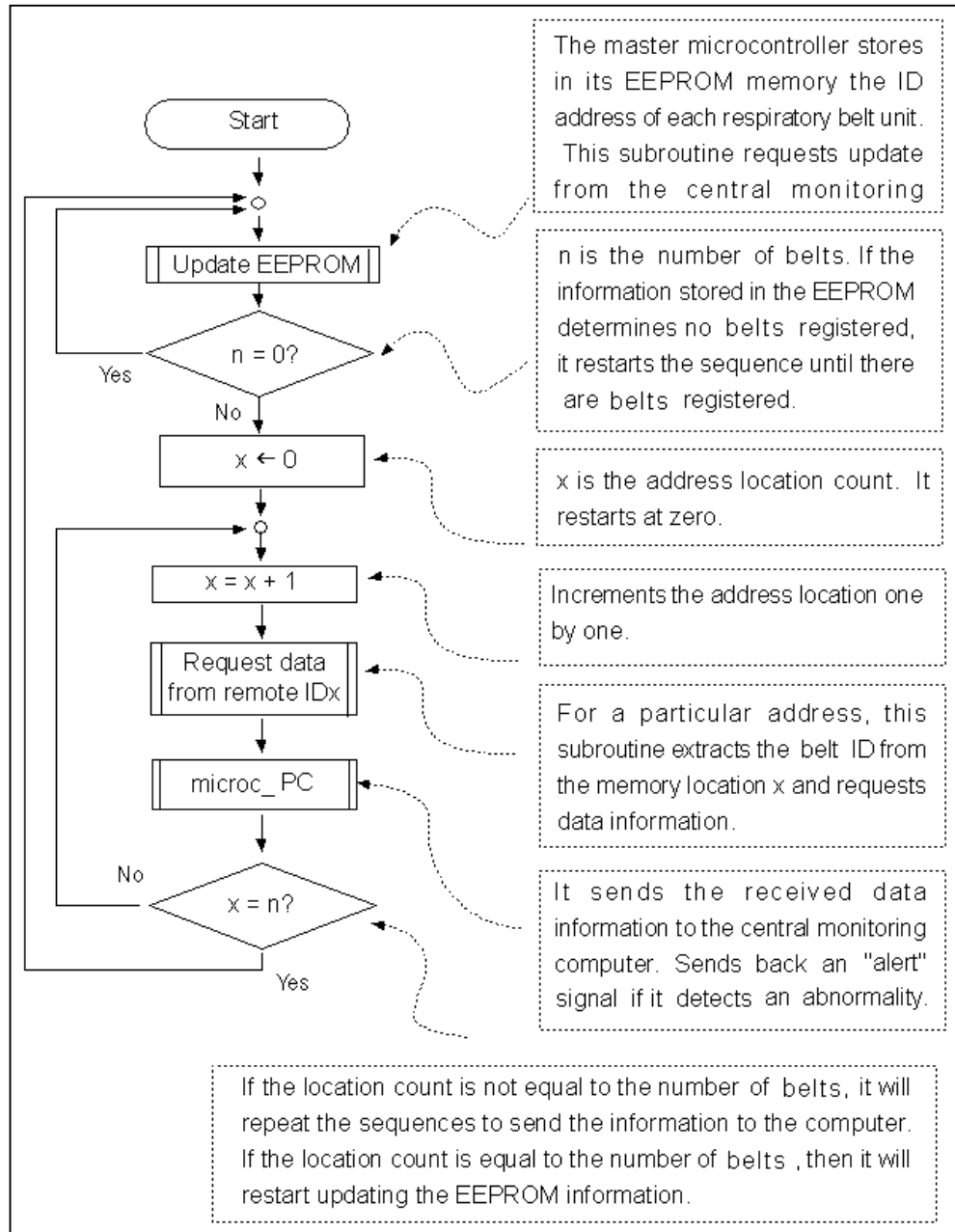


FIGURE 4.1: Master microcontroller software main flow diagram. The information from the EEPROM, which stores the ID address of the remote units, is continuously updated. The microcontroller requests information from each remote ID and sends the information to the central monitoring computer.

The program has three subroutines, the “Update EEPROM” subroutine, the “Request data from remote IDx” subroutine and the “microc_PC” subroutine. In the program, these subroutines are given the names “PC_EE”, “Rqst_ID”, and “Microc_PC” respectively.

4.1.2 Subroutine “Update EEPROM”.

4.1.2.1 Internal EEPROM Memory.

The microcontroller 16F877 can store 256 bytes of data in EEPROM memory. Address 0 stores the number of registered belt units and starting address 1, the memory stores the unique belt unit ID. Figure 4.2 shows an example of an EEPROM memory with five belt units registered. Address “0” contains the data “5” indicating there are five belts registered. Addresses “1” to “5” contain the ID of each belt unit. In the example, the belt unit number four has an ID of “132”.

The memory has a capacity to store 255 ID codes. Nevertheless, the program limits the number to 32 belt units. This amount can be redefined according to the necessities of the application.

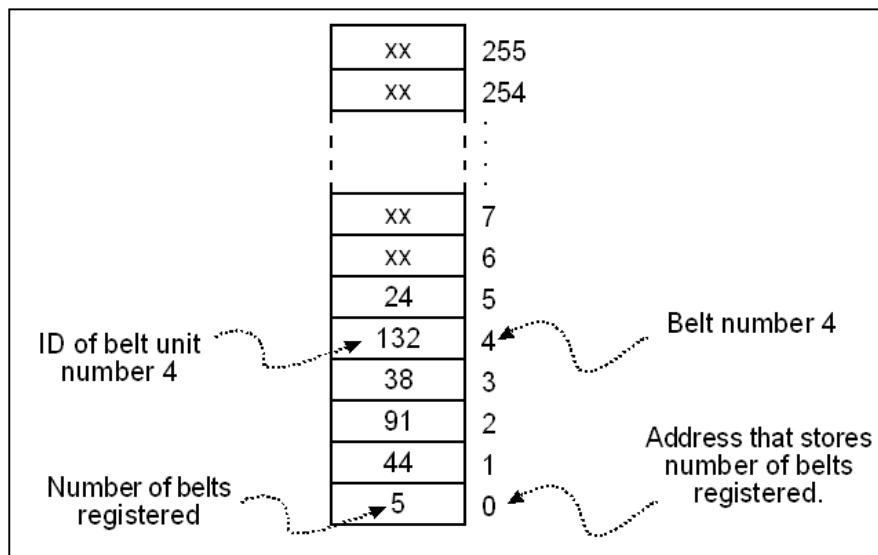


FIGURE 4.2: Master microcontroller EEPROM memory mapping example. Address 0 is reserved for the storage of the number of belts registered and starting address 1 for the storage of the belt unit unique ID.

4.1.2.2 Description of the Subroutine.

A “call EE_PC” from the main microcontroller program executes the “Update EEPROM” subroutine. The program verifies the reception of data from the central monitoring computer. If there is no data, there is no update information for the memory EEPROM, and the subroutine returns to the main program without modifying any information. If there is data, it verifies the type of information that it is. The central monitoring computer can send three types of commands: A new respiratory belt unit has been registered in the computer (command 01), a respiratory belt unit is leaving the network system (command 02), or the system is resetting the previous list (command 03). Figure 4.3 shows the flow diagram for the “Update EEPROM” subroutine.

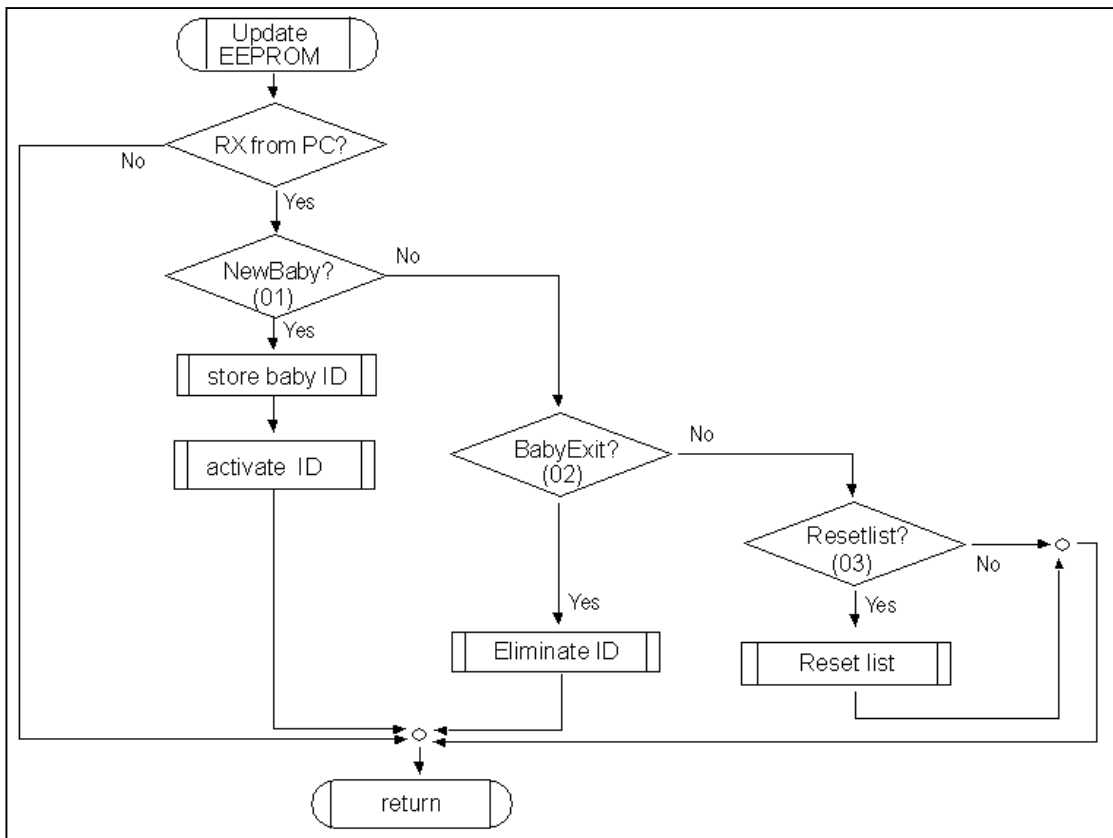


FIGURE 4.3: Update EEPROM memory subroutine. The main microcontroller can receive the command 01 (for a new belt registered in the network), command 02 (to eliminate a previous belt registry), command 03 (to reset the list) or simply no information update.

4.1.2.2.1 Command 01: New belt registered.

When the command 01 is detected, the new belt ID has to first be stored in the EEPROM memory. Then, the microcontroller must activate the remote respiratory unit that has the given ID. This subroutine has two third level subroutines: store baby ID and activate ID.

4.1.2.2.1.1 Subroutine "Store Baby ID":

The third level subroutine "Store Baby ID" is in charge of storing the new ID in the next available memory location. If the memory had the last ID stored in address x , the new ID should be stored in address $x+1$. The subroutine starts by reading the information stored in address "0" which indicates the current number of active registrations, and assigns this value to the variable n . The program increases the value of the variable n by one. This new value becomes the address location for storing the new ID. Finally, the value in address "0" is updated with the new value of variable n to continuously store the number of registered belts. Figure 4.4

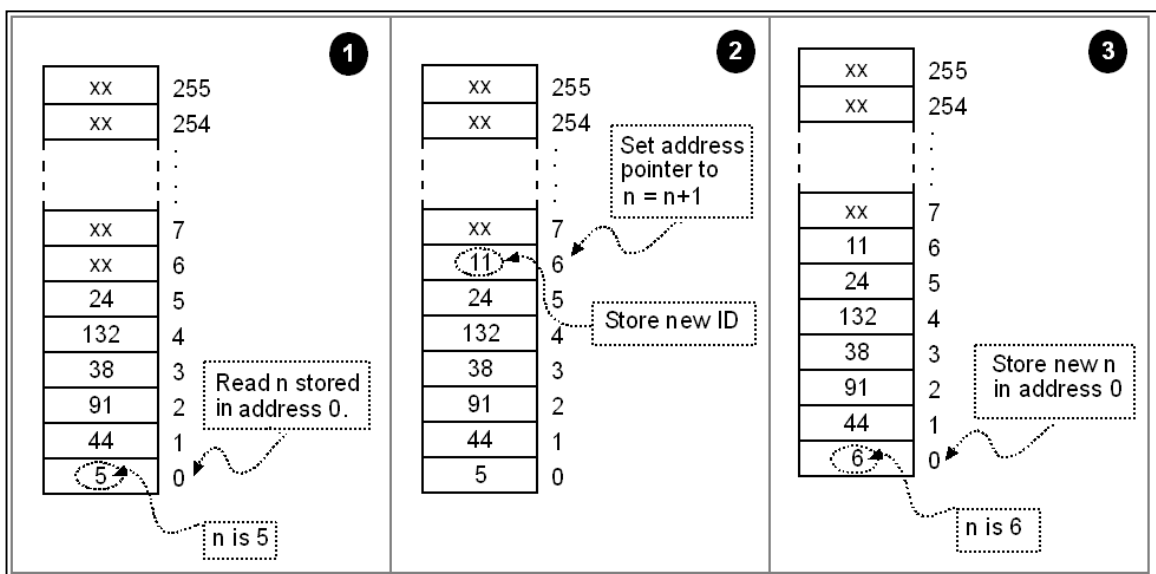


FIGURE 4.4: Visual steps of the third level subroutine "store baby ID". The EEPROM memory is updated storing a new baby ID in the next memory location.

The same procedure can be observed in the flow diagram from Figure 4.5. The subroutine uses the variables EEADR as the memory address pointer, and EEDATA as the data stored in memory. For every read operation, the address is assigned to variable EEADDR and then the fourth level subroutine “read” is executed. The data read is located then in EEDATA. For every store operation, the address is assigned to EEADDR and the data to be stored in EEDATA, and the fourth level subroutine “store” is executed.

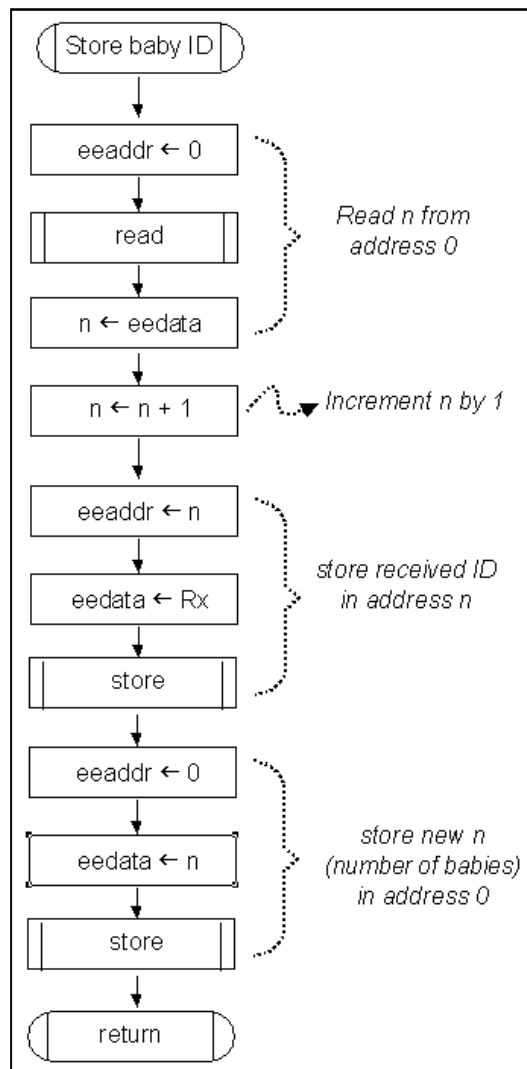


FIGURE 4.5: Flow diagram of the third level subroutine “store baby ID”. Address “0” is read and the content is variable n. This is incremented by 1, and the value is used as the address location. The ID is stored in this address location, and address “0” is updated.

4.1.2.2.1.2 Subroutine "Activate ID":

After the new ID has been stored in memory, the master microcontroller sends the data "55" to the remote microcontroller and expects the remote microcontroller to receive this command, and return the same data. The microcontroller then sends the central monitoring computer the message that the ID is connected. In case the master microcontroller does not receive a confirmation after a given amount of time, a message of "unable to connect" is sent to the computer and the ID is eliminated. The elimination of this ID is executed in a fourth level subroutine and consists in reading the address "0" to extract the number of belts registered, decreasing this amount by one and restoring this information. Figure 4.6 shows the flow diagram of this third level subroutine "Activate ID" and the fourth level subroutine "Eliminate last ID".

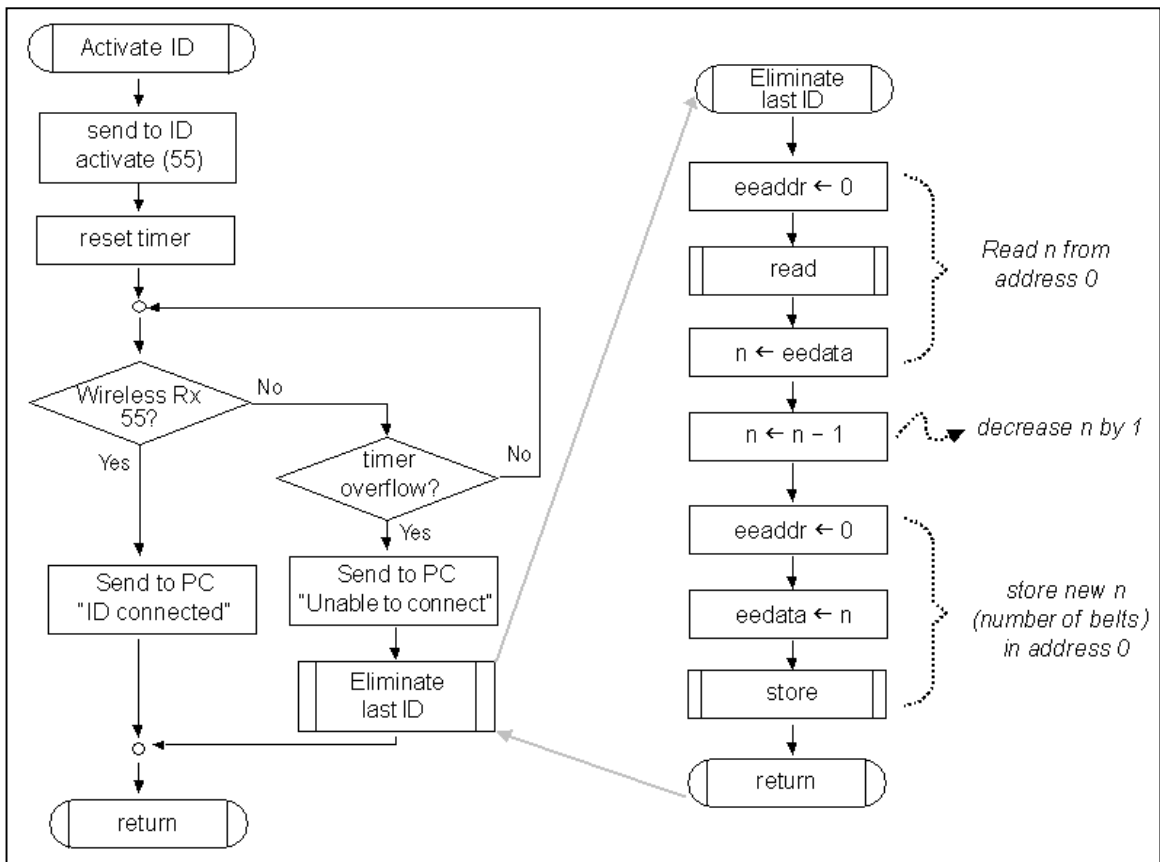


FIGURE 4.6: Flow diagram of the third level subroutine "Activate ID". The microcontroller sends command 55 to remote microcontroller. In case there is no confirmation, the last ID is eliminated.

4.1.2.2.2 Command 02: Respiratory Belt Exit.

When the command 02 is detected, a respiratory belt is removed from the system, so its ID must be eliminated from the master microcontroller EEPROM memory. The program first determines the total number of belts registered, and assigns this value to the variable *y*. Then, it verifies each memory address for the current location of the ID to be eliminated, starting from address 1 until address *y*. Once it finds the memory location, it assigns this address the variable *n*. Figure 4.7 shows an example of the first steps executed for eliminating an ID from memory.

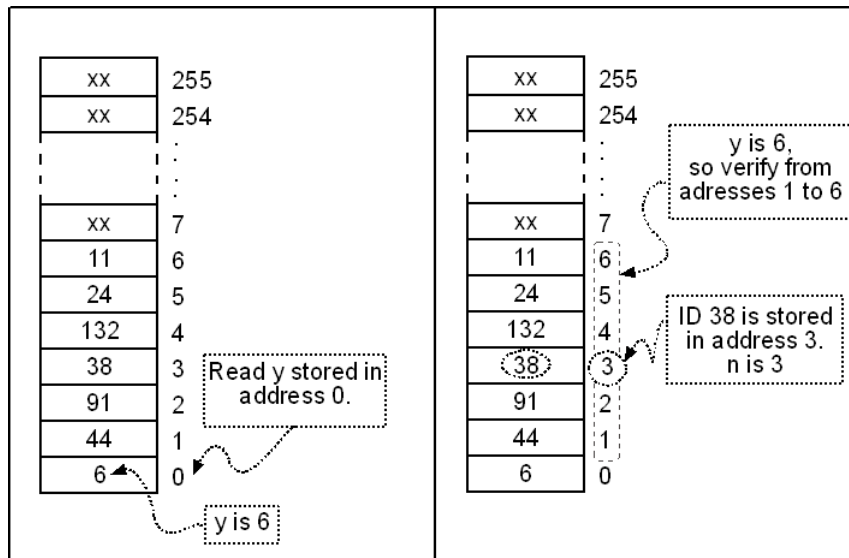


FIGURE 4.7: First steps executed for eliminating an ID from memory. The location of the ID to be eliminated first has to be located.

If all of the valid addresses are checked, and none of the contents match the ID to be eliminated, then, the program sends a message to the central monitoring computer indicating that the ID is not found. Otherwise, the program proceeds to execute a one by one displacement of the content of the address starting at the next address of the ID eliminated and ending in the last valid address. Then, the content of data of address “0” is decreased by updating in this way the information of the number of belts registered in

the system. Figure 4.8 displays an example of the displacement process and the update of address “0”.

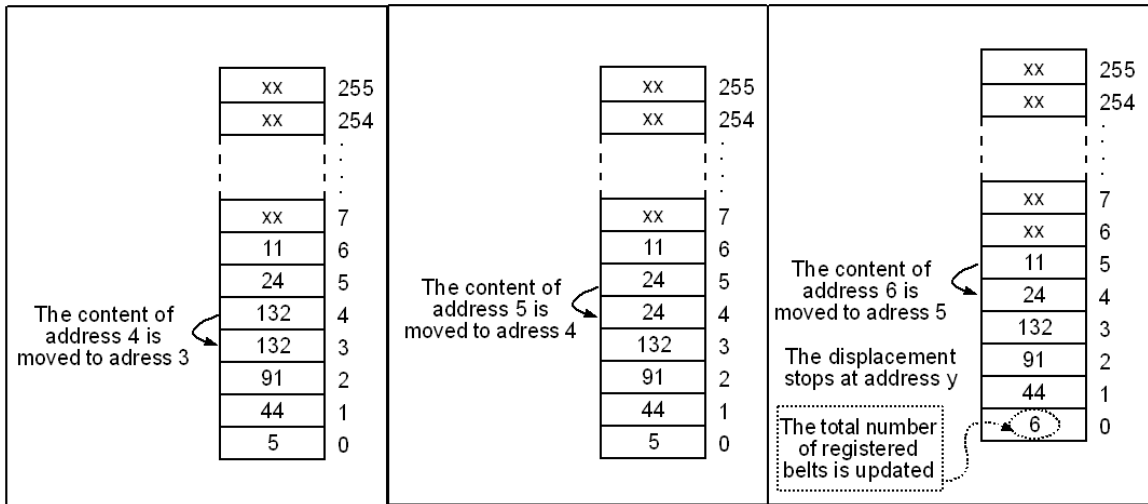


FIGURE 4.8: Next steps executed for eliminating an ID from memory. The content of the location are displaced one by one until reaching address y and the total number of registered belts is updated.

Figure 4.9 shows the flow diagram of the subroutine “Eliminate ID”. In this flow diagram, four variables are used which are detailed below:

ID: This is the variable that holds the value of the ID that the user programmed for elimination.

Y: This is the variable that holds the value of the number of belts registered before elimination. The value is extracted from address memory “0”.

N: The value in this variable acts as an address pointer. The program first locates the address that stores the ID to be eliminated and sets this value to be the starting address pointer value. The value is incremented one by one as the data stored in each memory location is displaced to the previous address.

X: This is a variable that stores temporarily the value stored in memory location N in order to accomplish the displacement.

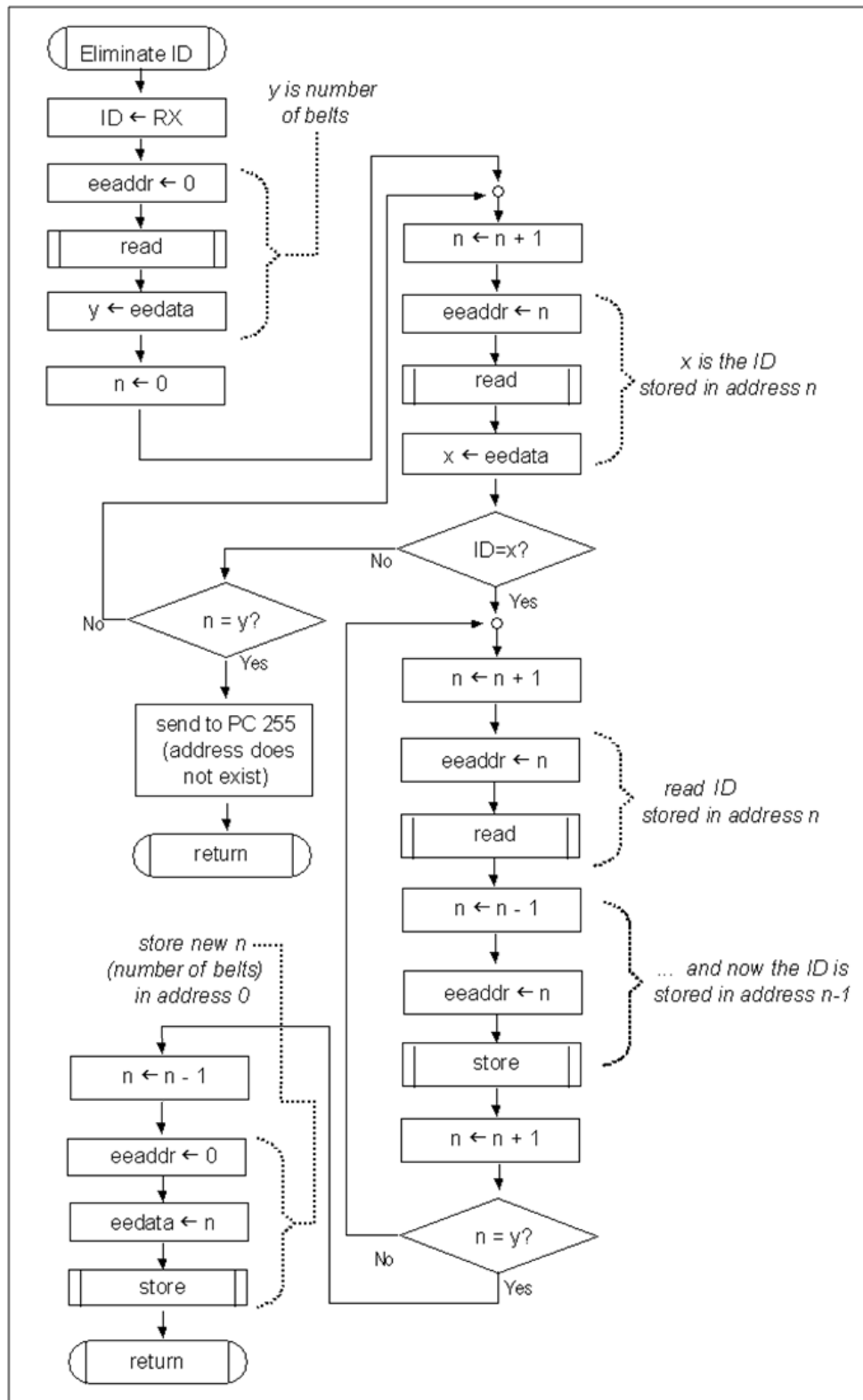


FIGURE 4.9: Flow diagram of the third level subroutine “Eliminate ID”. The address where the ID to be eliminated is first located, and then the contents of the following locations are displaced one location.

4.1.2.2.3 Command 03: Reset List.

The command reset list stores the data “0” in the memory address “0”. Even if some memory locations had previously data stored in them and are not physically erased, since the program first reads the information “0 respiratory belt units”, it will not read the previously stored memory locations. A future command of “New Baby” would overwrite the next pointed memory location. Figure 4.10 shows the flow diagram of this subroutine.

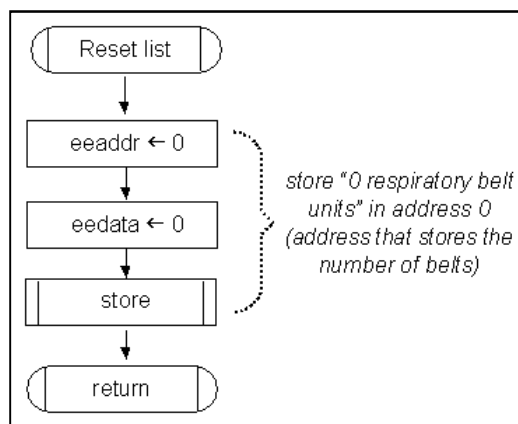


FIGURE 4.10: Flow diagram of subroutine “Reset list”. The content of address “0” is set to “0”.

4.1.3 Subroutine “Request data from remote IDx”.

Figure 4.11 shows the flow diagram of the diagram for the subroutine “Request data from remote IDx”. When the master microcontroller program reaches the instruction “call Rqst_ID”, the program has entered a loop until receiving data information for each of the belts registered in the memory. The subroutine requests data from the remote microcontroller with the address ID defined by variable X, a value which is incremented before the program jumps to the subroutine. The request consists of sending the command “AA” and expecting the reception of the remote data. If no reception is received in a given amount of time, the program returns from the subroutine

with the value 255 in W, which indicates the ID is disconnected. If a data is received successfully, the microcontroller repeats the previous operation to avoid errors, and

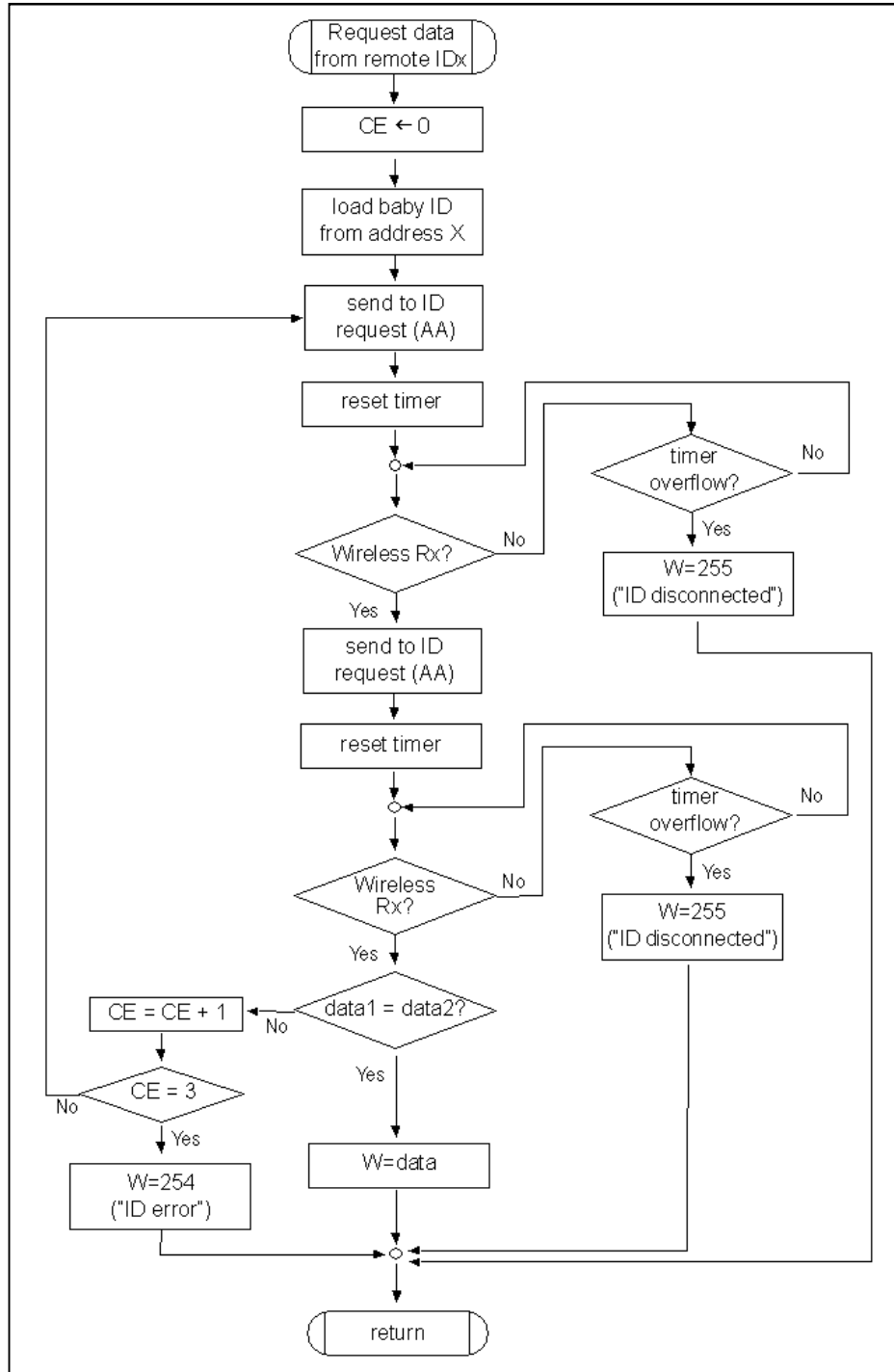


FIGURE 4.11: Flow diagram of subroutine “Request data from remote IDx”. The command “AA” is sent requesting data twice.

compares both values received. If they coincide, the program returns from the subroutine with the value of the data in W, otherwise, the process repeats. If after repeating the process three times there is still no match, the remote ID must be recalibrated and the program returns from the subroutine with the value 254 in W. This value indicates “ID error”.

4.1.4 Subroutine “Microc_PC”.

The subroutine “Microc_PC” is the subroutine that achieves the communication between the remote microcontroller and the central monitoring computer. The data received in the previous subroutine from the remote station is sent to the computer. If the values were 254 or 255, the program returns from the subroutine to the main program since these values indicate an abnormality. Otherwise, the data sent from the remote microcontroller is processed by another program in the central monitoring computer. The main microcontroller program expects a code “0” or “1” from this other program, indicating “Everything is normal” or “Alert”, and sends this information back to the remote microcontroller. Figure 4.12 shows the flow diagram of this subroutine “Microc_PC”.

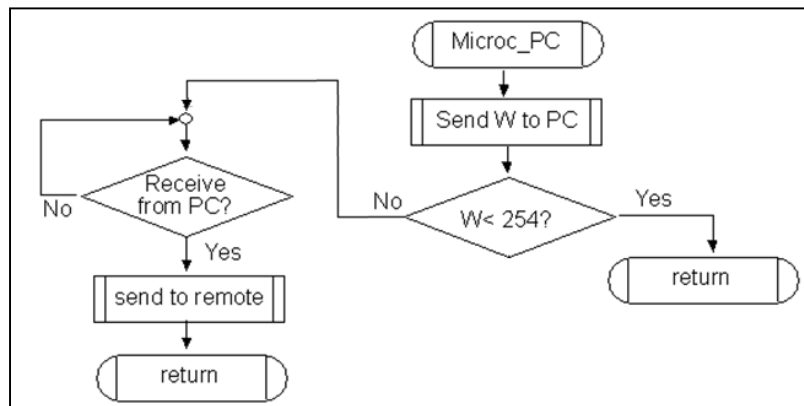


FIGURE 4.12: Flow diagram of subroutine “Microc_PC”. If the values sent to the computer are not 254 or 255, a value that is received from the computer is sent to the remote station.

4.2 Remote Microcontroller Programming

The remote microcontroller program first has to detect the logic state of the selection mode to operate in either individual mode or in network mode. Figure 4.13 shows the flow diagram of the remote microcontroller program.

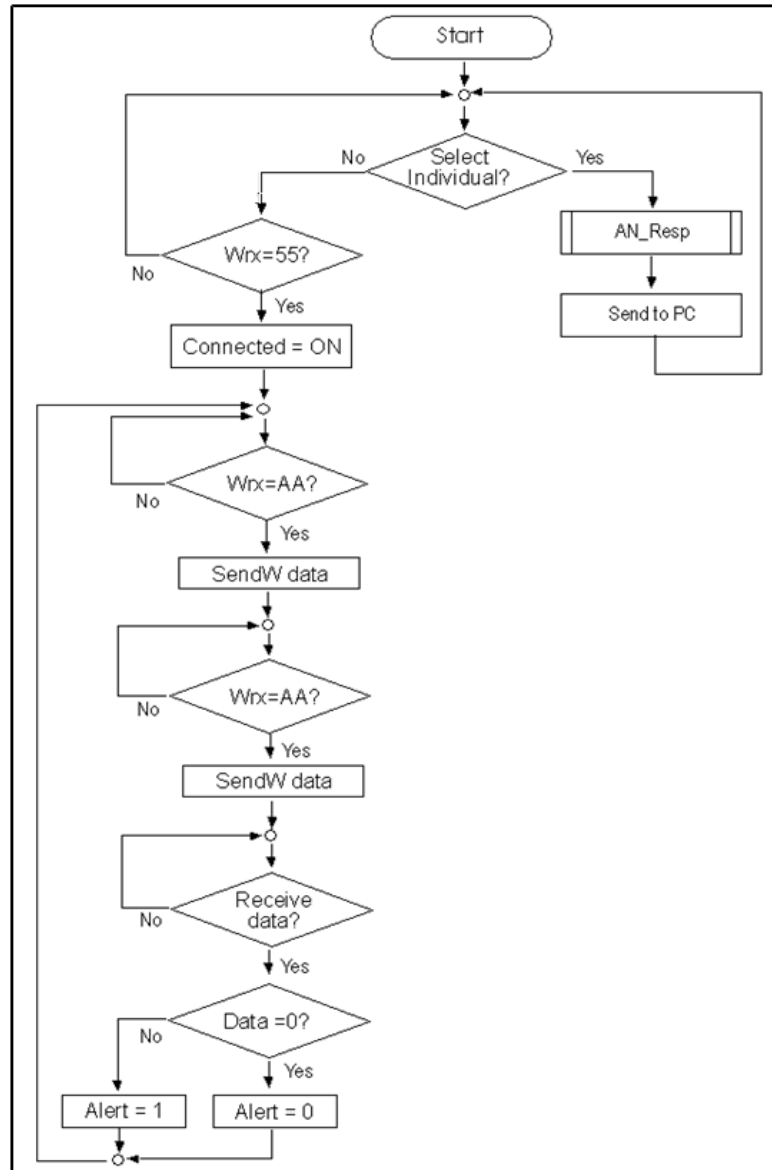


FIGURE 4.13: Remote microcontroller flow diagram.

If the mode is selected as individual mode, then the program converts the respiratory sensor analog data to digital and sends the information to the individual

monitoring computer. Otherwise, the network mode is selected and the program first awaits the wireless data “55”, which is the command to activate. Once active, the program awaits twice the data “AA” in order to send the respiratory sensor data to the master microcontroller. Afterwards, the program awaits the reception of the result of the analysis of the data sent.

4.3 Summary

The respiratory monitoring system is designed to work in individual mode or group mode. When connected as individual mode, the remote microcontroller program converts the analog data to digital data and sends the information to the computer using a serial port interface. When connected as group mode, in order for a remote microcontroller to send the respiratory data to one central monitoring computer without interfering with other remote microcontrollers, a master microcontroller controls the sequence of events, sending each remote microcontroller an initial one time signal to activate, and a signal each time the respiratory data is requested. Therefore, the remote microcontroller program in group mode consists of first waiting for an initial wireless signal for activation and then waiting for the signal to send the analog conversion.

The program in the master microcontroller coordinates information between each remote microcontroller and the central monitoring computer. In order to properly request data from each computer, the master microcontroller holds in its internal memory the information of how many remote microcontrollers are active and the ID of each remote microcontroller. The master microcontroller receives the information from the computer whether a new baby is registered, a baby has exited the system or a reset has been set on the computer, updating with this information its internal memory, and sending an activation signal to a remote microcontroller in case of a new baby command. This

subroutine was called uEEPROM. Figure 4.14 summarizes the interaction between the main microcontroller, the remote microcontroller and the central computer program for this first stage in group mode.

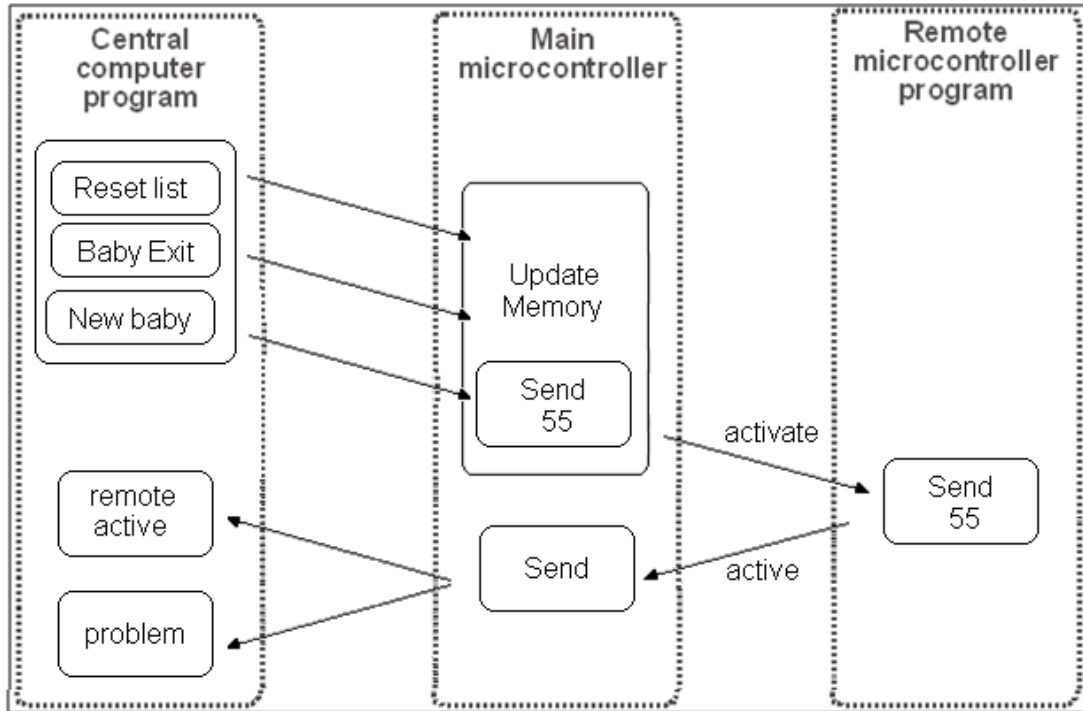


FIGURE 4.14: First stage of program description.

Using the information incoming from the central monitoring computer, the main microcontroller updates the memory. In case of a new baby, it sends the activation signal to the remote microcontroller and expects the signal back. If positive, sends the signal to the computer that the new remote ID is active, or if negative, sends a problem in detection signal.

Using the information in its memory, the master microcontroller proceeds to request data from each active remote microcontroller. Then, it sends the received data to the PC, and the status of respiratory data back to the remote microcontroller. This process is repeated for each of the remote microcontrollers active in the system, before returning and updating its internal memory. Figure 4.15 summarizes the interaction between the programs for this second stage in group mode.

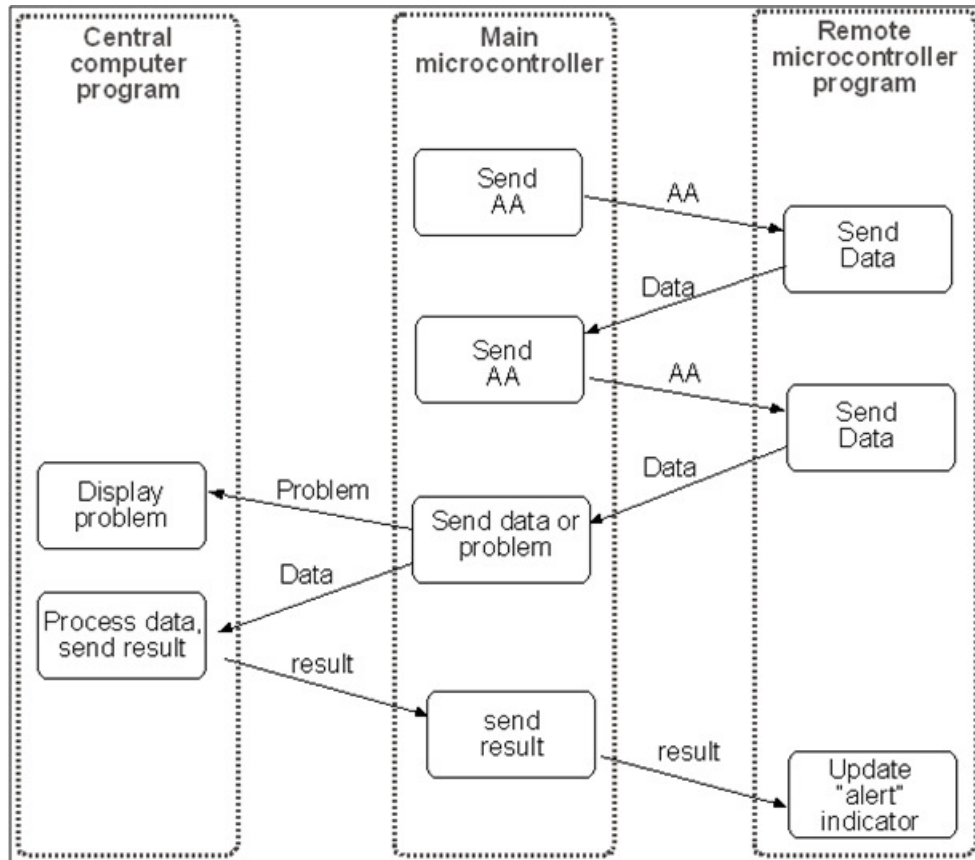


FIGURE 4.15: Second stage of program description. Master microcontroller sends the remote microcontroller a request for data, and receives the data twice and sends the information to the central monitoring computer. If data was received successfully, the computer analyses the data and sends the result back to the remote microcontroller in the main microcontroller. This process is repeated for each active remote unit in memory.

CHAPTER FIVE:
ANALYSIS, CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

5.1 Tests Analysis

5.1.1 Description of Technique for Simulation.

The respiratory sensor system was tested using an expandable plastic oval closed container which varied its circumference depending on the amount of air pumped into or out of the container. The belt was wrapped around the container which simulated the thorax of a neonatal, and air was pumped into and out of container cyclically in order to simulate the thoracic expansion from 0 to 4mm. The system was connected to a MATLAB program as described on figure 2.5 and the data was plotted in order to analyze the results.

In the MATLAB program, the reception of data was set using baud rate of 9600. After opening the serial port (`fopen(s)`), the data was read and assigned to variable `out` (`out=fread(s)`). Since the sampling time for the microcontroller was not yet set, and was varied during different tests, the sampling times were detected by the MATLAB program using the commands `tic` to start the timer, and `toc` to stop the timer. The x-axis was set either by samples or time, and the y-axis by either the digital value or by the voltage.

5.1.2 Tests Results Varying with Belt Tension.

During the tests conducted, it was observed that the belt tightening (i.e., tension) had a key role in the changes of the sensor amplitude during respiration. If the belt was

set too tightly with initial resting position above 233 byte value (i.e., 4.55V), the signal undergoes a limited amplitude variation, making the pattern loose resolution with a maximum variation of 5 byte value (10mV). Figure 5.1 shows the pattern generated in a condition of the belt being set too tightly.

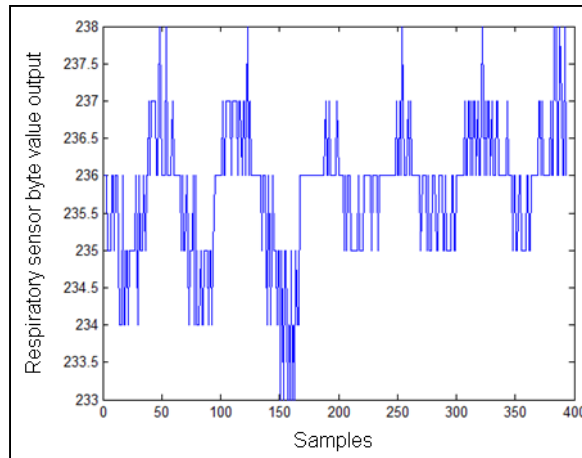


FIGURE 5.1: Recording when the respiratory belt sensor is set too tightly.

If the belt is set too loosely, with the initial position with a maximum value below 220 byte value (i.4.3V), the thoracic movement does not generate proper displacement in the belt. Figure 5.2 shows a pattern generated from a belt with initial position below 217 byte value.

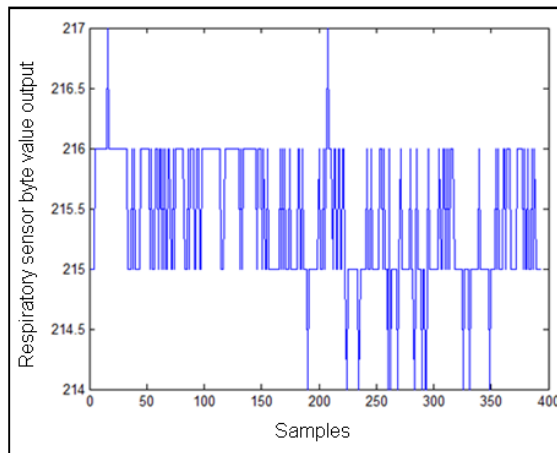


FIGURE 5.2: Recording when the respiratory belt sensor is set too loosely.

To solve these problems, before initiating the program, a pre-program was added to detect when the belt is in optimum tension. At the start, the microcontroller records sensor output samples and then calculates the difference between the maximum and minimum value. If the difference between the values is less than 8 digital values, the program will not start and the belt must be relocated. Once the user adjusts the belt tightening, and the value difference is greater than 8 digital value, a BELT ON indicator lights up. Figure 5.3 shows the flow diagram for the “Belt On” test.

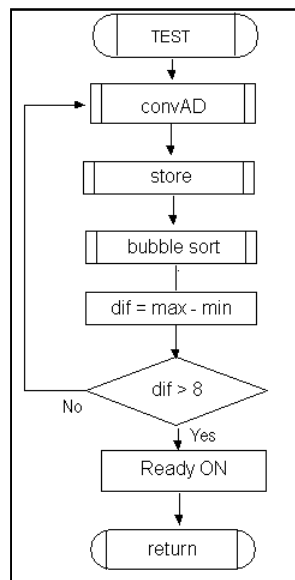


FIGURE 5.3: “Belt On Test” to determine correct tightening of belt.

5.1.3 Tests Results Varying with Sampling Times.

A breathing pattern was simulated with the container for twelve seconds. The test was repeated changing the time interval in-between each sending of samples of data, with the following conditions:

One hundred and twenty samples were taken during the twelve second period with a time delay between samples t_1 (40 samples/second), 102 samples, time period with approximately 6 simulated breaths. The belt was removed from the container and

relocated in the same position and the test was repeated for a total of twenty tests. An example of the test is shown in Figure 5.4.

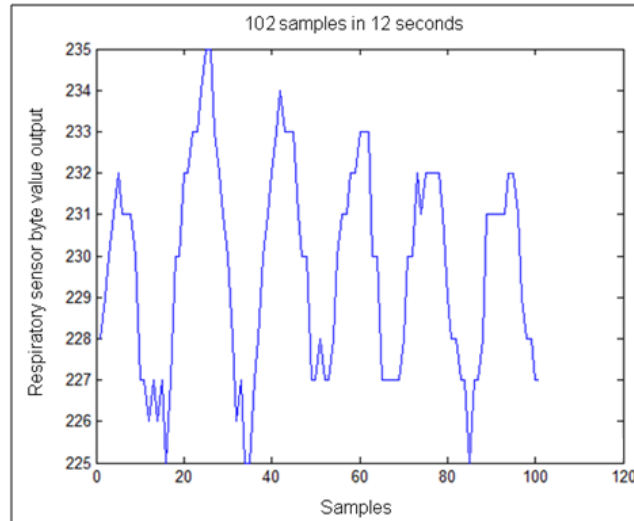


FIGURE 5.4: Test simulation using 102 samples in 12 seconds with time delay t_1 .

The tests were repeated an additional twenty times recording three hundred and ninety samples during the twelve second period with a time delay t_2 (150 samples/second). The belt was removed from the container completely in-between each test and returned to the same position. An example of the test is shown in Figure 5.5.

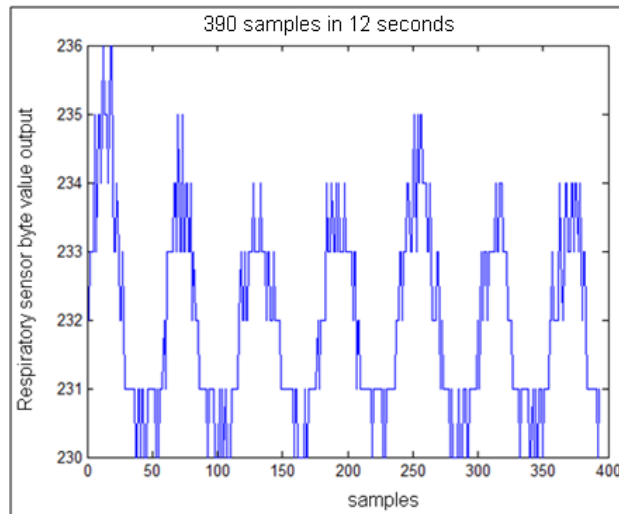


FIGURE 5.5: Test simulation using 390 samples in 12 seconds with time delay t_2 .

The tests were again repeated an additional twenty times with 1440 samples during the twelve second period with a time delay between samples t_3 (400 samples/second). The belt was removed from the container completely in-between each test and returned to the same position. . An example of the test is shown in Figure 5.6.

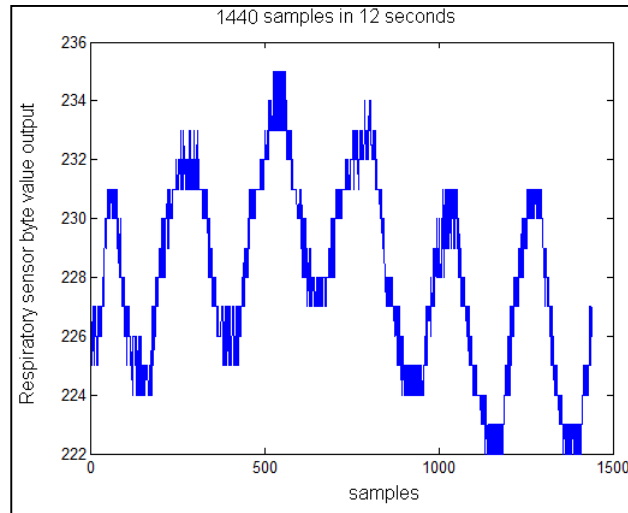


FIGURE 5.6: Test simulation using 1440 samples in 12 seconds with time delay t_3 .

These tests were repeated during a time period three times longer using the same sampling periods of t_1 , t_2 and t_3 . The results of one of each of the tests are shown in Figure 5.7, 5.8 and 5.9 respectively.

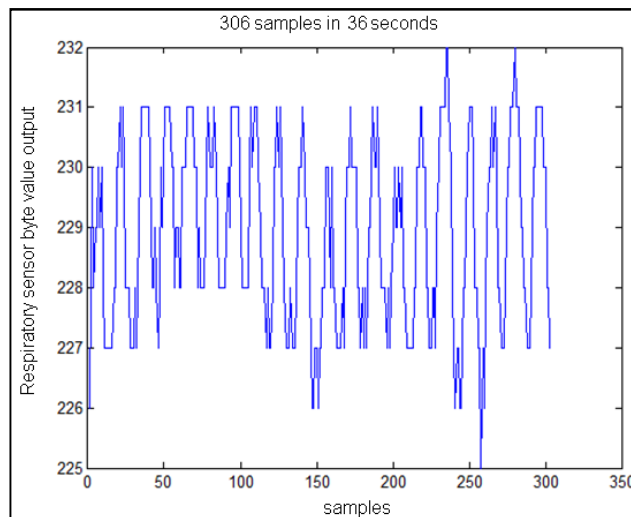


FIGURE 5.7: Test simulation using 306 samples in 36 seconds with time delay t_1 .

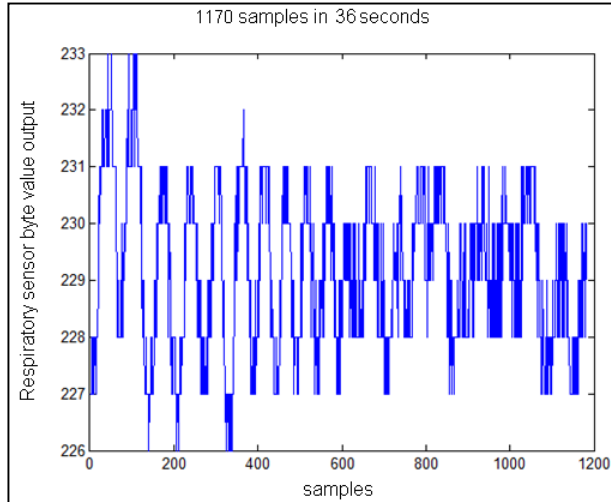


FIGURE 5.8: Test simulation using 1170 samples in 36 seconds with time delay t2.

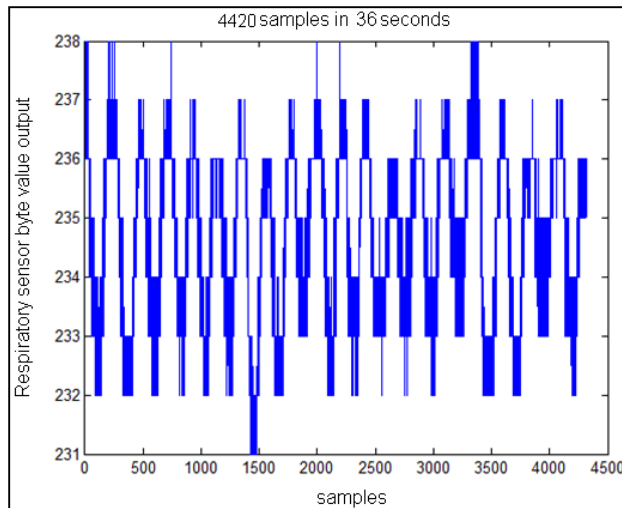


FIGURE 5.9: Test simulation using 4420 samples in 36 seconds with time delay t3.

5.1.4 Tests Results Varying Amplitude Resolution.

The analog/digital conversion subroutine was substituted with an alternative of right justification referring to the eight least significant bytes being stored in the ADRESL register and the two most significant bytes being stored in bits 0 and 1 of register ADRESH. During the test, only the register ADRESL which carried 8 least significant bits of a 10 bit resolution conversion was graphed. For an initial testing, the belt was

manually pulled from the starting point zero to the maximum possible movement. Several tests were then taken. In all of the results the scale reading (byte value) the values increasing from the byte value of 75 changes abruptly to 190. Additionally, when the value is lowering from 0, changes abruptly to 255. The valid scales are shown to be between 0 and 75 and between 190 to 255. The non use of the two most significant bits creates the abrupt changes in the increase above 75 and the decrease below 0, therefore with a repetitive cycling pattern as shown in Figure 5.10.

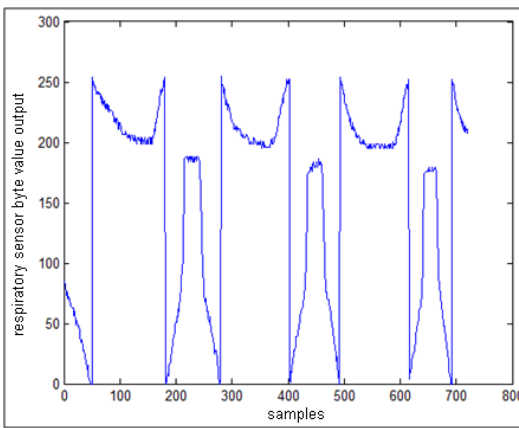


FIGURE 5.10: Results with right justification in AD conversion.

Adjusting the sensor physically created slight changes, but remained with the pattern, as shown in figure 5.11.

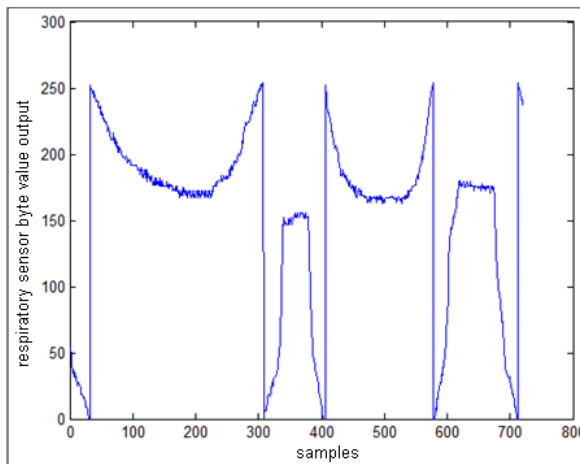


FIGURE 5.11: Results with right justification after physical adjustments.

The next step was to analyze the rescaling results while still only reading ADRESL and disregarding ADRESH. With this adapted scale, twenty 8-second tests were run first on the simulated container shown in Figure 5.12. The test was repeated varying the expansion of the container method and frequency, with results shown on Figure 5.13.

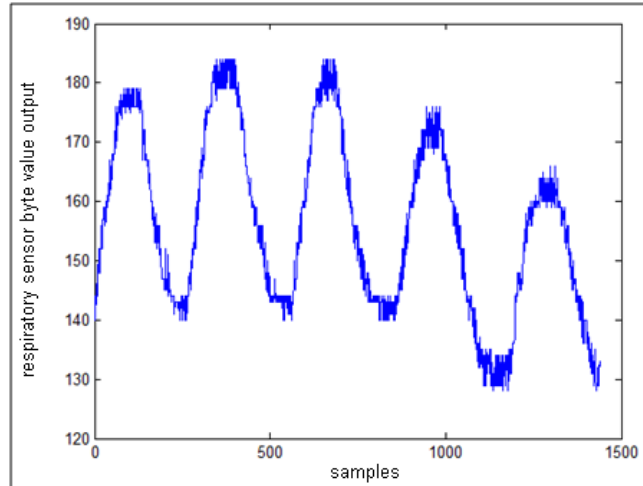


FIGURE 5.12: Simulated test 1 with new amplitude scale.

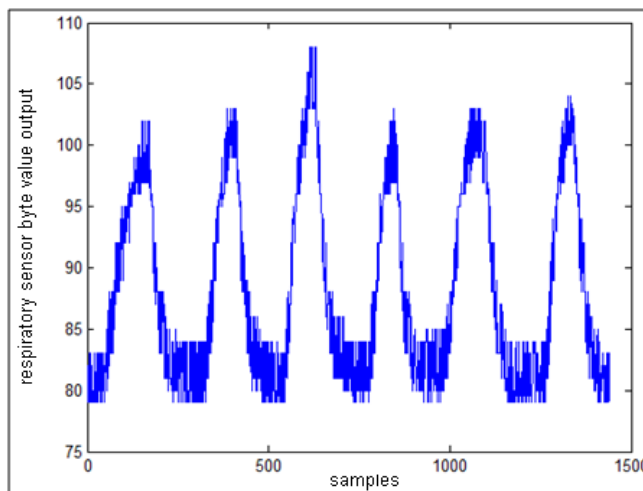


FIGURE 5.13: Simulated test 2 with new amplitude scale.

5.1.5 Rescaling Y-axis to Voltage and X-axis to Time.

In order to observe the changes during respiratory breathing related to the voltage output of the PR2012 respiratory sensor, the Y-axis was scaled to voltage by

using variable volt equivalent to $V_{max}/256$ times “out”. The X-axis was scaled to time using the instruction tic to start a timer and toc to stop the timer. After the period of timing elapsed, the time was assigned to variable t. The samples taken during the period of test were determined by the length of “out”. Therefore, the sampling time was determined by calculating $t/\text{samples}$. Figure 5.14 shows a simulation breathing pattern scaled in the X-axis to time and in the Y-axis to the respiratory sensor byte value. Figure 5.15 shows the same simulation breathing pattern scaled in the X-axis to samples and in the Y-axis to the voltage.

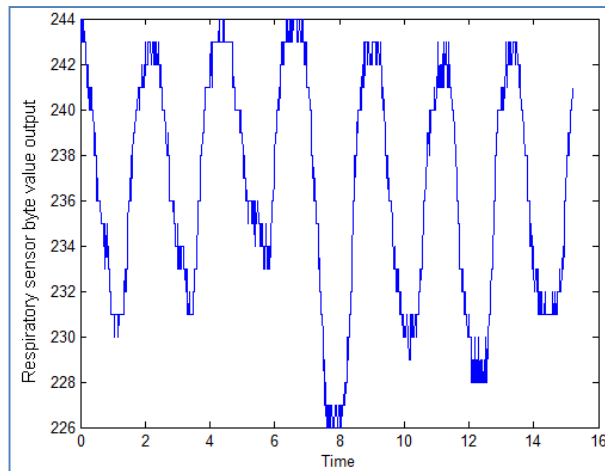


FIGURE 5.14: Simulated test scaled to time vs. byte value.

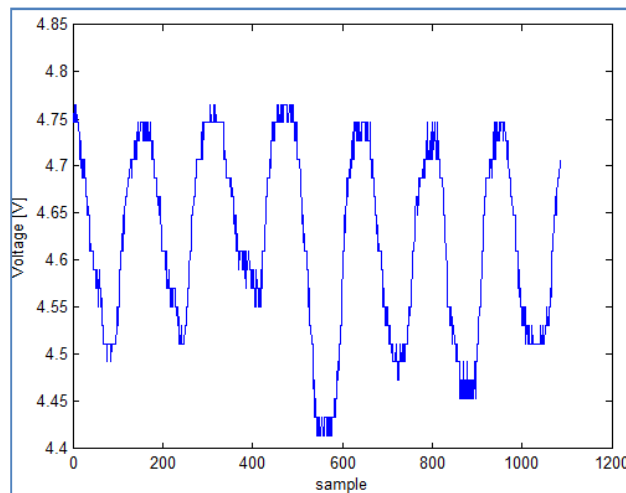


FIGURE 5.15: Simulated test scaled sample vs. voltage.

5.1.6 Frequency Calculations.

To determine the frequency, the signal was converted to a square signal by assigning a value of 1 to the data above a determined value and a -1 to the data below a second determined value. Do avoid erroneous assignment due to noise, such as the example set in figure 5.16, the median first was calculated, and a threshold above and below the median set the upper and lower cuts for the conversion. Once detecting the upper cut and assigning a value of 1, the value was maintained until reaching the lower cut, and therefore eliminating the risk of erroneous detection in possible noise between the upper and lower levels of cut.

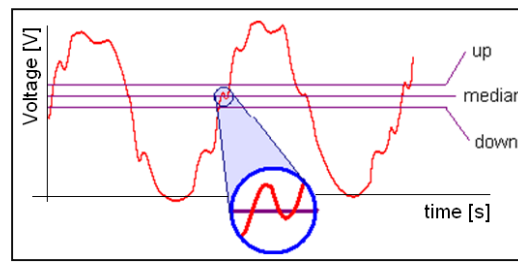


FIGURE 5.16: Assignment of lower and upper cut for square signal conversion.
Note: No scale included. Generic example only for illustration.

Figure 5.17 shows results of conversion of the simulated breathing signal converted to square signals. Appendix E shows additional examples of tests results.

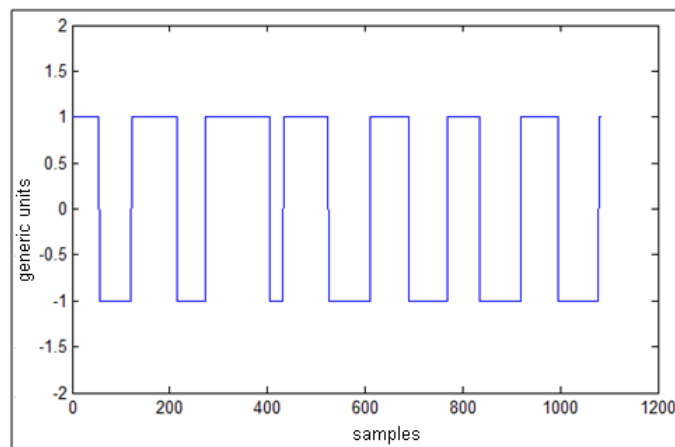


FIGURE 5.17: Simulated test signal converted to square signal.
Refer to figures 5.13 and 5.14.

To determine the time elapse in each cycle, a program runs calculating for all of the set of samples, the difference between the sample and the next sample. If the difference is 2, there is a change in cycle, and therefore a timer initiates for the next cycle and ends for the current cycle. Figure 5.18 shows a print screen showing the results of the times calculated for each of the cycles in the example. Set points have been approximated for each of the times calculated by software.

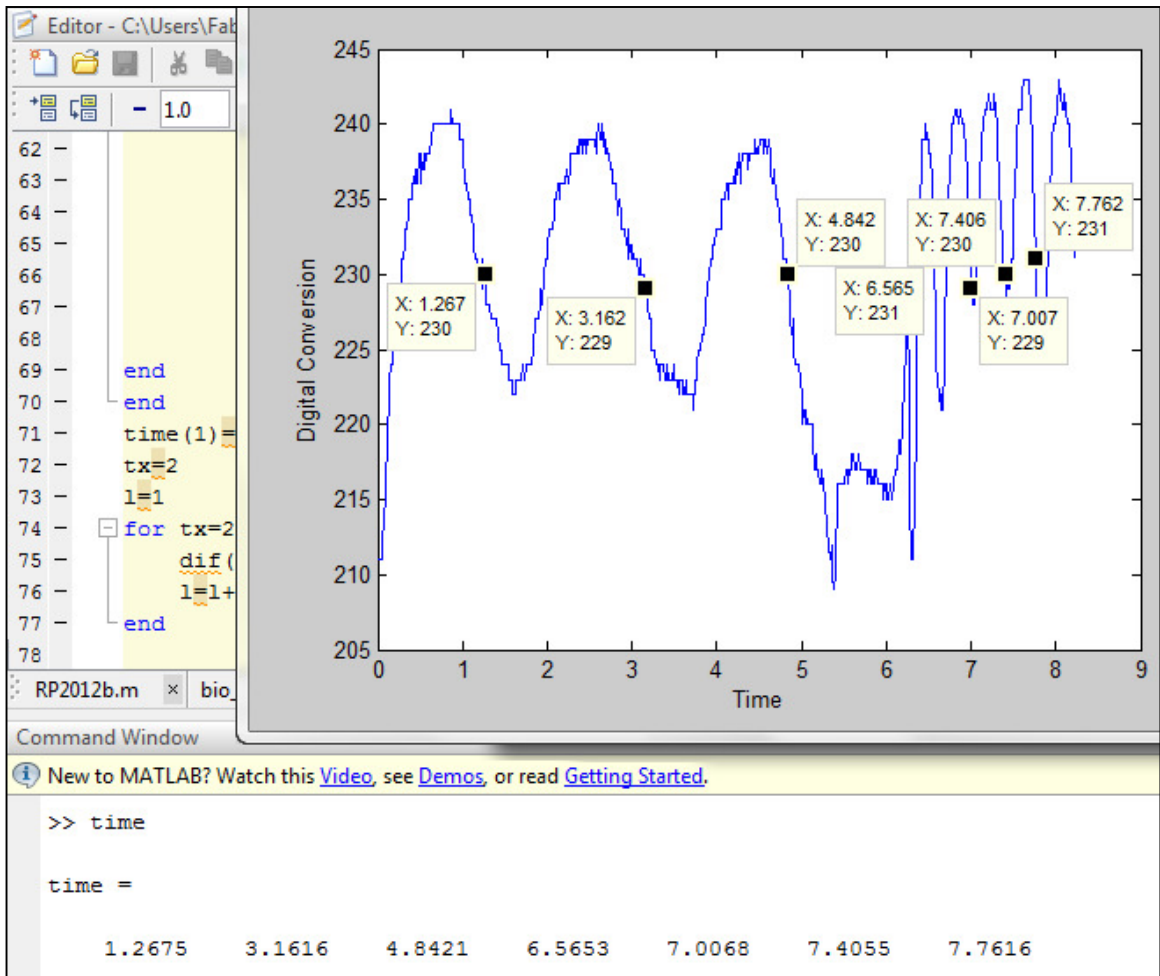


FIGURE 5.18: Time elapsed calculation test.

Using these values, the average time elapsed in each cycle and the deviation of times is calculated, as well as the frequencies calculated in breaths/minute for each cycle, the average of all the frequencies determined by each cycle and the frequency

deviation. Figure 5.19 shows the print screen showing the results of the frequency calculation. As observed visually, the first three cycles have much slower frequencies than the next four cycles. The results show the frequencies of 46.6658, 28.6931 and 28.1231 breaths/minute for the first three cycles and 84.9317, 114.7726, 114.7726 and 106.1646 breaths/minute for the next four cycles, which corresponds to the visual graph. The average frequency is given to be 53.6572 breaths/minute and the deviation is given to be 82.2892 breaths/minute. Such a high deviation indicates an unstable frequency rhythm. Additional test results are included in Appendix F.

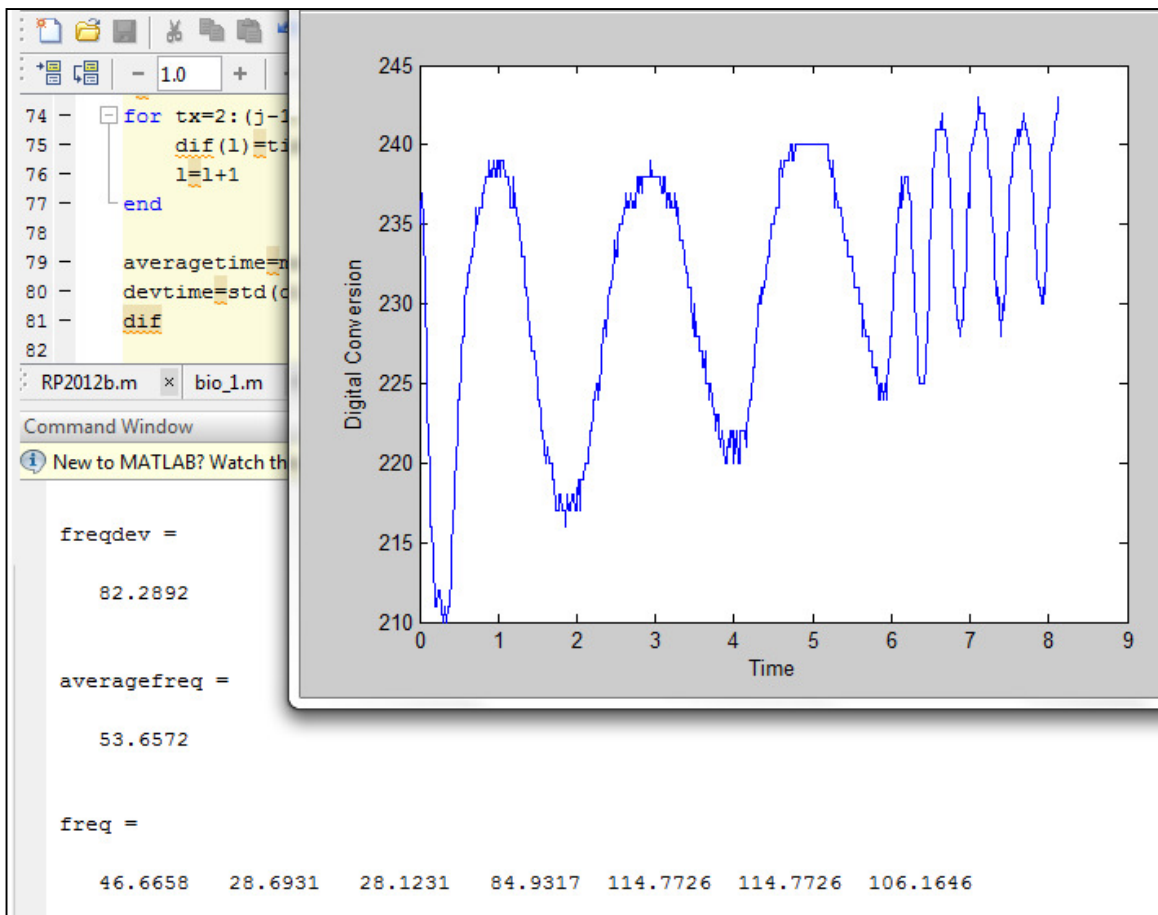


FIGURE 5.19: MATLAB calculation of frequency for the corresponding test. The frequency deviation, average frequency and frequencies of each cycle is calculated, which corresponds visually to the breathing rhythm.

5.1.7 Consistency in Sampling Time.

For the group monitoring mode, to enable the system to have consistent time delay in between the sampling of two of the same belt monitoring unit regardless of the amount of belt monitoring units connected to the system, the system was limited for 16 babies. An intentional time delay is added at the end of the program multiplied by 16 minus the number of belt monitoring units connected. This way, the system had the same timing between each sample collected, and could the respiratory rate could be calculated more uniformly.

5.2 Electrical Shock Hazard Safety

According to the OSHA publication 3075 “Controlling Electrical Hazards”, the current that poses an electrical hazard on a human being depends on the humidity of the skin⁽¹³⁾. Estimates for effects of electrical current in human body describe 5 mA as the point in which slight shock is felt without it being painful, with possibilities of indirect injuries caused by involuntary reactions. Painful shocks start being detected from 6 to 25 mA in women, and 9 to 30 mA in men. Between 50 to 150 mA, additional symptoms such as extreme pain and respiratory difficulties, and even death⁽¹⁴⁾.

The International standard IEC 60601-1 sets a lists of standards for approval for medical electrical equipment ⁽¹⁵⁾. Some of the standards specifically for medical electrical equipment include the IEC 60601-1-2 for the collateral standard for electromagnetic compatibility and for requirements and tests, the IEC 60601-1-3 for general requirements for radiation protection in diagnostic X ray equipment, and IEC 60601-1-6 for usability ⁽¹⁶⁾. The Sub-clause 2.2.15 of the IEC 60601-1 defines the “equipment, provided with not more than one connection to a particular supply mains and intended to diagnose, treat, or monitor the patient under medical supervision and

which makes physical or electrical contact with the patient and/or transfers energy to or from the patient and/or detects such energy transfer to or from the patient.”⁽¹⁷⁾ This description includes examples such as battery operated thermometer.⁽¹⁷⁾

Proper electrical safety is considered during the elaboration of the respiratory monitoring system. The system is battery-operated and is covered with isolated material material. Each wire entering the sensor is individually isolated and the group of isolated wires has a second isolated for additional safety. The equipment does not require to have direct contact with the baby’s skin.

5.3 Conclusions

The neonatal respiratory system hardware to allow individual monitoring as well as group monitoring system was built, including the respiratory sensor for such hardware. The respiratory belt unit microcontrollers as well as the master microcontroller were programmed. Each part of the system was tested separately. The respiratory sensor was built and tested, recording the relationship between the distance of displacement and the voltage, using the microcontroller to interface the computer and connect the respiratory sensor results in a MATLAB program. The test showed the respiratory belt has consistent and repeatable results if the respiratory sensor belt unit is set with the correct tightening. Otherwise, the results were unreliable, not periodic and contained significant noise. The belt tension problem was solved by adding a pre-program to indicate when the belt was located in the correct position and the system detecting reliable data. Several tests were conducted varying the sampling time to determine the effect of the signal. Increasing the sampling time does not result in more reliable data, and will require the microcontroller to send the information at a faster rate. An intermediate sampling time from the tests conducted was elected for the rest of the

tests, equivalent to 40 samples / second. The resolution of amplitude was increased by using left justification and storing the analog conversion in the register ADRESL. This did not result in significant change, and added the problem of distortion of the signal if the least significant bytes stored in ADRESH were to be ignored, so the option was discarded. Tests were conducted rescaling the X-axis to time and the Y-axis to voltage.

The tests completed during simulation gave reliable data for calculation of breathing rate; calculating time elapsed for each cycle, average time of each cycle. Additionally, the frequency of the breathing pattern was calculated. Since the breathing pattern could have a frequency breathing rhythm that changes over time, a program was done calculating the individual frequency at which each cycle was completed, having the individual frequencies for each of the cycles. The program gave the results of the average of the frequencies and the deviation from each frequency in breaths/minute.

5.4 Future Work

As a future work, the project needs to seek IRB approval in order to proceed to test the current monitoring system in babies, children and adults to detect differences in results of respiratory patterns and rate and observe specific needs for the respiratory system for babies. Record the results for each case, and the adjustments that need to be made in the system.

A next step includes testing the system under different distances to determine the maximum range of wireless signal and effects of the different signals in close-by proximity. Additionally, to improve the system, a program in an object oriented high level programming could facilitate visual results and use by an operator. The belt can be improved by industrial fabrication.

To compliment the current monitoring system, adding the detection of other vital signals as well, such as heart rate detection, temperature and blood pressure can be added. The current remote microcontroller has only one additional analog input. Another microcontroller with more input pins must be used.

REFERENCES

- (1) Guidelines for hospitals with neonatal intensive care service: regulation 4 of the private hospitals and medical clinics regulations [cap 248, Rg1]. Licensing and Accreditaion, Ministry of Health.
- (2) Oesterrgaard MZ, Inoue M, Yoshida S, Mahanani WR, Gore FM, et al 2011 Neonatal Mortality Levels for 193 Countries in 2009 with Trends since 1990: A Systematic Analysis of Progress, Projections and Priorities. PLoS Med 8(8): e1001080 doi: 10.1371/journal.pmed.1001080
- (3) WHO Bulletin volumes 88/1/08-062554
- (4) Ivana Murkovic, Matthew Steinberg, Branimir Murkovic. Sensors in neonatal monitoring: Current practice and future trends. Technology and Health Care 11 (2003) 399/412 IOS Press.
- (5) Amit K. Gupta. Respiration Rate measurement based on Impedance Pneumigraphy. Texas Instrument . Application Report SBAA181- February 2011. Data Acquisition Products.
- (6) Riccobon, Davide, Respiration Belt MR, Special Issue ISMRM 2010. www.brainproducts.com
- (7) Hans Strasburger and Dieter Klenk. Opto Electronic Belts for recording respiration in psychophysiological experimentation and therapy. Psychophysiology 1983. The society for psychophysiological research, inc. vol. 20 no 2.
- (8) Mark Brink, Christopher H. Muller, Christoph Schierz. Contact-free measurement of Heart Rate, respiration rate, and body movements during sleep. Behavior research methods 2006, 38(3), 511-521.
- (9) Dennis Dodds, Jonathan Purdy, Chris Moulton. The PEP transducer: a new way of measuring respiratory rate in the non-intubated patient. Accid Emerg Med 1999; 16:26-28.
- (10) Alamaki, Sliz, Ashraf, Sorvoja, Alasaarela and Hamalainen Measurement of ECG, Respiratory Rate, Tilt and Temperature of a patient and wireless Zigbee Data Transmission By. University of Oulu 2nd International Symposium on Medical Informantion and Communication Technology. 2007.

- (11) Lynette Jones, Nikhila Deo and Brett Lockyer. Wireless Physiological Sensor System for Ambulatory Use. Massachusetts Institute of Technology. Proceeding of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06) 0-7695-2547-4/06 2006 IEEE
- (12) Reflective Object Sensors Types OPB606A, OPB606B, OBP606C. OPTEK. Product Bulletin OPB606. June 1996.
- (13) "Controlling Electrical Hazards". OSHA 3075. 2002.
- (14) W.B. Kouwenhoven. "Human Safety and Electric Shock". Electrical Safety Practices, Monograph, 112, Instrument Society of America, p. 93. November 1968.
- (15) Turnbull, Aidan. Medical – Eco Design. The use of IEC 60601-1 in supporting approvals of medical electrical devices and the role of the new collateral standard IEC 60601-1-9. September 2007.
- (16) International Standard. IEC 60601-1. Third Edition 2005-12. Part 1. General requirements for basic safety and essential performance.
- (17) Leonard Isner, Robert M. Brown and Dan Modi. A Primer for IEC 60601-1. Published MDDI. September 2003.
- (18) HT12A/HT12E 2¹² Series of Encoders datasheet. Holtek. April 11, 2000.
- (19) HT12D/HT12F 2¹² Series of Encoders datasheet. Holtek. April 11, 2000.
- (20) PIC16F87xA Data sheet. 28/40/44 Pin Enhanced FLASH Microcontrollers Microchip. 2003 Microchip Technology Inc. DS39582B
- (21) PIC16F87/88 Data sheet. 18/20/28 Pin Enhanced FLASH Microcontrollers with nano Watt Technology. Microchip. 2003 Microchip Technology Inc. DS30487B
- (22) PICkit™ 2 Programmer/Debugger User's Guide. Microchip. 2008 Microchip Technology Inc. DS51553E.
- (23) PICmicro™ Mid-Range MCU Family. Reference Manual. 1997 Microchip Technology Inc. December 1997/ DS33023A.
- (24) MPLAB® IDE User's Guide. Microchip. 2005 Microchip Technology Inc. DS51519A.
- (25) MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian. User's Guide. Microchip. 2005 Microchip Technology Inc. DS33014J.
- (26) RLP434A RF ASK Hybrid Modules for Radio Control datasheet. Laipac Technology , Inc.
- (27) TLP434 RF ASK Hybrid Modules for Radio Control datasheet. Laipac Technology , Inc.

- (28) Walteneus Dargie and Christian Poellabauer. Fundamentals of Wireless Sensor Networks. Theory and Practice. Wiley Series on Wireless Communications and Mobile Computing. A John Wiley and Sons, Ltd. Publication. First Edition 2010. ISB 978-0-47099765-9 (H/B)
- (29) F. Egan, William. Practical RF System Design. Wiley-IEEE Press. 2003. ISBN 9780471200239.
- (30) C.S. Raghavendra, Krishna M. Sivalingam and Taieb Znati. Wireless Sensor Networks. ISBN: 1-4020-7883-8. 2004 Springer Science+Business Media, Inc.
- (31) Elgloria A. Harrison, MS, RRT-NPS. Neonatal Respiratory Care Handbook. Jones and Barlett Publishers. 2011.
- (32) Michael E. Wiklund. Medical Device and Equipment Design. Usability Engineering and Ergonomics. ISB 0-935184-69-4 1995 Interpharm Press, Inc.
- (33) Sergey Y. Yurish. Digital Sensors and Sensor Systems: Practical Design. Internation Frequency Sensor Association (IFSA) Publishing. 2011. ISBN 978-84-615-6957-1.
- (34) J. Witt, K. Krebber (BAM), D. Kinet and F. Narbonneau (MUL). Optical Fibre Sensors Embedded into technical Textile for Healthcare. D4.1 Respiratory motion sensor design. Information Societies Technology. 2007
- (35) K. Nakjima, T. Tamura, T Ohta, H. Milke and PA Oberg. Photoplethysmographic measurmente of hear and respiratory rates using digital filters. Engineering in Medicine and Biology Society, 1993. Proceeding of the 15th Annual Internation Conference of the IEEE. 1993. Pages 1006 and 1007. ISBN 0-7803-1377-1. Version August 2002.

APPENDICES

Appendix A: Assembler Code for Master Microcontroller

```
. ***** 16F877 MASTER MC *****
;
;
; *
; * Name: Fabiola Araujo Céspedes
; * Title: MASTER MICROCONTROLLER
; * Description:
; *
; *
; *   RB0-RB7... Remote Unit ID (Encoder Address)
; *   RD0-RD3... Encoder Data
; *   RD4..... TE/ (Encoder)
; *
; *
; *   RA0-RA3... Decoder Data
; *   RA4..... VT (Decoder)
; *
; *
; *   RC6/TX ... Transmit
; *   RC7/RX ... Receive
; *
; *
; *****
;

__CONFIG_CP_OFF&_WDT_OFF&_PWRTE_ON&_LVP_OFF&_XT_OSC

list p=16f877a
#include<p16f877a.inc>
#define VT    PortA,4
#define TE    Portb,4

cblock 0x21
LSB
MSB
cont1
cont2
wireless
dataRX
CE
N
X
Y
ID
EEADRM
EEDATAM
TOIF
DATA1
DATA2
TXWIRELESS
TEMP
endc
```

Appendix A: (continued)

```
ORG 0
GOTO inicio

; ***** MACROS *****
Bank0 MACRO
    bcf    STATUS,RP0
    bcf    STATUS,RP1
ENDM

Bank1 MACRO
    bsf    STATUS,RP0
    bcf    STATUS,RP1
ENDM

Bank2 MACRO
    bcf    STATUS,RP0
    bsf    STATUS,RP1
ENDM

Bank3 MACRO
    bsf    STATUS,RP0
    bsf    STATUS,RP1
ENDM

; ***** INITIAL CONFIGURATION *****

inicio  bank1

    clrf    trisb    ; Port B output (Encoder address)
    clrf    trisd    ; Port D output (Encoder data)
    movlw   b'111111'
    movwf   trisa    ; RA0-RA4 input (Decoder)
    movwf   trise    ; RE0-RA3 input (Extra input)
    movlw   0x0c
    movwf   ADCON1    ; Ports Digital
    MOVLW   d'25'; 9600 BAUDIOS
    MOVWF   SPBRG
    MOVLW   0X24; tx DE 8BIT, HIGH SPEED
    MOVWF   TXSTA; MODO ASINCRONO, TXEN
    movlw   b'00000010'
    movwf   OPTION_REG
    BANK0
    MOVLW   0X80; rx DE 8BIT
    MOVWF   RCSTA
    movf    rcreg,w
    movf    rcreg,w
    movf    rcreg,w
```

Appendix A: (continued)

```
; ***** MAIN PROGRAM *****

Start  call      uEEPROM
        movlw    0x00
        xorwf    N,w
        btfsc   status,z
        goto    start
loop   clrf      X
        incf     X,f
        call    Rqst_ID
        call    microc_PC
        movf    X,w
        xorwf    N,w
        btfss   status,z
        goto    loop
        goto    start

; ***** SECOND LEVEL SUBROUTINES *****

uEEPROM
        bsf     rcsta,rcen    ; enable reception
        btfss   PIR1,RCIF
        return
        movf    RCSTA,w
        movwf   dataRX
        movlw   0x01
        xorwf   dataRX,w
        btfsc   status,Z
        goto    NEWBABY
        movlw   0x02
        xorwf   dataRX,w
        btfsc   status,Z
        goto    BABYEXIT
        movlw   0x03
        xorwf   dataRX,w
        btfss   status,Z
        return
        call    RESETLIST
        return

NEWBABY
        nop
        call    STOREBABYID
        call    ACTIVATEID
        return
```

Appendix A: (continued)

```
BABYEXIT
    nop
    call    ELIMINATEID
    return

Rqst_ID
    clrf    CE
    movf    X,w
    movwf   EEADRM
    call    read
    movf    EEDATAM,w ; reading ID
    movwf   portb
first    movlw 0xaa
    call    txW
rtime    bcf    INTCON,TOIF ; empty TMR0 overflow flag
    movlw  0x36
    movwf   TMR0 ; value TMR0
Rloop1   btfsz VT
    goto    TO1
    call    RXW
    movf    wireless,w
    movwf   data1
second   movlw 0xaa
    call    txW
rtime2   bcf    INTCON,TOIF ; empty TMR0 overflow flag
    movlw  0x36
    movwf   TMR0 ; value TMR0
Rloop2   btfsz VT
    goto    TO1
    call    RXW
    movf    wireless,w
    movwf   data2
    movf    data1,w
    xorwf   data2
    btfsz   status,z
    goto    countCE
    movf    data1,w ; will send to PC data
    return

countCE
    incf    CE,f
    movlw  0x03
    xorwf   CE,w
    btfsz   status,z
    goto    first
    movlw  0xFE ; PC code for "ID error"
    return
```

Appendix A: (continued)

```
TO1  btfss      INTCON,TOIF
      goto      Rloop1
      movlw     0xff          ; PC code for "ID disconnected"
      return

TO2  btfss      INTCON,TOIF
      goto      Rloop2
      movlw     0xff          ; PC code for "ID disconnected"
      return
```

```
microc_PC
      call      SENDPC
      movwf     temp
      movlw     0xFE
      btfsc     status,C
      return
      call      RECEIVEPC
      call      TxW
      return
```

```
; ***** THIRD > LEVEL SUBROUTINES *****
```

```
sSTOREBABYID
      clrf      EEADRM
      call      read
      movf      EEDATAM,w
      movwf     N
      incf     N,f
      movf     N,w
      movwf     EEADRM
      call      receivePC
      movf     dataRX,w
      movwf     EEDATAM
      call      store
      clrf      EEADRM
      movf     N,w
      movwf     EEDATAM
      call      store
      return
```

```
ACTIVATEID
      movf     dataRx,w
      movwf     PortB
      movlw     0x55
      call      TXW
rtimer bcf      INTCON,TOIF ; empty TMR0 overflow flag
      movlw     0xFF
      movwf     TMR0          ; value TMR0
```


Appendix A: (continued)

```
loopA1
    btfss    VT
    goto    TOA
    call    RXW
    movlw   0x55
    xorwf   wireless,w
    btfss   status,z
    goto    TOA
    movlw   0xff ; PC code for "ID connected"
    call    SENDPC
    return

TOA
    btfss   INTCON,TOIF
    goto    loopA1
    movlw   0x02 ; PC code for "unable to connect"
    call    SENDPC
    call    E_Last_ID
    return

E_Last_ID
    clrf    EEADRM
    call    read
    movf    EEDATAM
    movwf   N
    decf   N,f
    clrf   EEADRM
    movf   N,w
    movwf  EEDATAM
    call   store
    return

ELIMINATEID
    call    receivePc
    movf   dataRX,w
    movwf  ID
    clrf   EEADRM
    call   read
    movf  EEDATAM,w
    movf  y ;y indicates the number of belts
    clrf  N
loop2   incf  N,f
        movf  N,w
        movwf EEADRM
        call  read
        movf  EEDATAM,w
        movwf X
```

Appendix A: (continued)

```

        xorwf      ID,w
        btfsc     status,z
        goto     loop3
        movf     N,w
        xorwf     Y,w
        btfss    status,z
        goto     loop2
        movlw    d'255'      ; PC code for "address does not exist"
        call    SENDPC
        return
loop3   incf     N,f
        movf     N,w
        movwf    EEADRM
        call    read
        decf     N,f
        movf     N,w
        movwf    eeadrM
        call    store
        incf     N,f
        movf     N,w
        xorwf     Y,w
        btfss    status,z
        goto     loop3
        decf     N,f
        clrf     EEADRM
        movf     N,w
        movwf    EEDATAM
        call    store
        return

RESETLIST
        clrf     EEADRM
        clrf     EEDATAM
        call    store
clrf    N
        return

; ***** SUBROUTINES *****

STORE
        movf     EEADRM,W
        bank2
        movwf    EEADR
        bank0
        movf     EEDATAM,W
        bank2
        movwf    EEDATA

```

Appendix A: (continued)

```
BANK3
bsf      EECON1,WREN
movlw   0x55
movwf   EECON2
movlw   0xAA
movwf   EECON2
bsf     EECON1,WR
bank0
btfss   PIR2,EEIF
GOTO    $-2
bcf     PIR2,EEIF
bank3
BCF     EECON1,WREN
bank0
RETURN

READ
movfw   EEADRM
bank2
movwf   EEADR
BANK3
bsf     EECON1,RD
BANK2
movf    EEDATA,W
bank0
movwf   EEDATAM
RETURN

SENDPC
Btfss   Pir1,TXif
goto    $-1
movwf   TxReg
call    retardito
return

RECEIVEPC
bsf     RCSTA,CREN;continuous reception
btfss   PIR1,RCIF
goto    $-1
movf    RCSTA,w
movwf   dataRX
return

TxW
movwf   LSB
movwf   MSB
bcf     LSB,4
swapf   MSB,f
```

Appendix A: (continued)

```
        bcf          MSB,4
        movf        LSB,w
        call       txW2
        movf        MSB,w
        call       txW2
        return

txW2
        movwf      PORTb
        call       delay
        bsf        TE
        call       delay
        return

RXW    movf        portA,w
        movwf      LSB
        btfsc     VT
        goto      $-1
        btfss     VT
        goto      $-1
        movf      portA,w
        movwf      MSB
        btfsc     VT
        goto      $-1
        movlw     b'00001111'
        andwf     LSB,f
        andwf     MSB,f
        swapf     MSB,w
        iorwf     LSB,w
        movwf     Wireless
        return

retardo
        movlw     d'2'
        movwf     cont2
        call      retardito
        decfsz   cont2,f
        goto     $-2
        return

retardito
        movlw     d'255'
        movwf     cont1
        decfsz   cont1,f
        goto     $-1
        return
```

Appendix A: (continued)

```
delay  movlw    d'64'  
        movwf   cont1  
        movlw   d'255'  
bucled2  
        movwf   cont2  
        decfsz  cont2,f  
        goto    $-1  
        decfsz  cont1,f  
        goto    bucl2  
        return
```

```
end
```

Appendix B: Assembler Code for Remote Microcontroller

```
. ***** 16F88 REMOTE MICROCONTROLLER*****  
,  
,  
, *  
, *  
, * Name: Fabiola Araujo Céspedes  
, *  
, *  
, * Title: REMOTE RESPIRATORY BELT UNIT  
, *  
, *  
, * Description:  
, *  
, *  
, * RA5=0 selects network mode (group monitoring)  
, * RA5=1 selects individual mode  
, *  
, *  
, * RA6 indicates alert "ALERT"  
, * RB6 indicates connected to network "CONNECTED"  
, *  
, *  
, * RB7/AN6 Analog input of respiratory sensor  
, *  
, *  
, * RA0...RA3 Data INPUT Decoder  
, * RA7 VT Decoder  
, *  
, *  
, * RB0...RB3 Data OUTPUT Encoder  
, * RB4 TE/  
, *  
, *  
, * RB2/RX ... Receive  
, * RB5/TX ... Transmit  
, *  
, *  
, *****  
,
```

```
_CONFIG _CONFIG1, _CP_OFF & _CCP1_RB0 & _DEBUG_OFF &  
_WRT_PROTECT_OFF & _CPD_OFF & _LVP_OFF & _BODEN_OFF & _MCLR_OFF  
& _PWRTE_ON & _WDT_OFF & _INTRC_IO
```

```
list p=16f88  
#include<p16f88.inc>  
  
#define Select PortA,5 ; 0 Network, 1 indiv  
#define Alert PortA,6  
#define Connected Portb,6  
#define VT PortA,7  
#define TE Portb,4  
  
cblock 0x21  
Wireless  
cont1  
cont2  
count
```

Appendix B: (continued)

```
Reg_dir  
Reg_data  
Regcont  
Temp  
LSB  
MSB  
Endc
```

```
ORG 0  
GOTO inicio
```

```
; ***** MACROS *****
```

```
Bank0 MACRO  
    bcf    STATUS,RP0  
    bcf    STATUS,RP1  
    ENDM
```

```
Bank1 MACRO  
    bsf    STATUS,RP0  
    bcf    STATUS,RP1  
    ENDM
```

```
Bank2 MACRO  
    bcf    STATUS,RP0  
    bsf    STATUS,RP1  
    ENDM
```

```
Bank3 MACRO  
    bsf    STATUS,RP0  
    bsf    STATUS,RP1  
    ENDM
```

```
; ***** INITIAL CONFIGURATION *****
```

```
inicio
```

```
Bank1  
movlw    b'01101000' ; 4Mhz  
movwf    OSCCON
```

```
movlw    b'10111111'  
movwf    TRISA
```

```
movlw    b'10100000'  
movwf    TRISB
```

```
movlw    b'01000000'  
movwf    ANSEL  
clrf     ADCON1
```

```
MOVLW    d'25'; 9600 BAUDIOS
```

Appendix B: (continued)

```

        MOVWF    SPBRG
        movlw   0x24
        movwf   TXSTA
        bank0

        MOVLW   0X80; rx DE 8BIT
        MOVWF   RCSTA
        movf    rcreg,w
        movf    rcreg,w

; ***** MAIN PROGRAM *****

        bcf     connected
        bcf     ALERT

bucle  btfss   select
        goto   group
        goto   Individual

Individual
        call   AN_Resp
        call   SendPC
        call   delay
        goto   bucle

Group  call   RxW
        movlw  0x55
        xorwf  wireless,w
        btfss  status,Z
        goto   bucle

        bsf    Connected

UNO    call   RXW
        movlw  0xAA
        xorwf  wireless,w
        btfss  status,Z
        goto   UNO

        call   SendWdata

DOS    call   RXW
        movlw  0xAA
        xorwf  wireless,w
        btfss  status,Z
        goto   DOS
```


Appendix B: (continued)

```
Results
    call    SendWdata
    call    RxW
    movf    wireless,f
    btfss   status,Z
    goto    Abnormal

    bcf     Alert
    goto    UNO
```

```
Abnormal
    bsf     Alert
    goto    UNO
```

```
; ***** SUBROUTINES *****
```

```
AN_Resp
    movlw   b'00110001' ; 00xxx000 xxx=AN6
    movwf   ADCON0
    call    retardito
    bsf     ADCON0,2 ; GO
    btfss   ADCON0,2 ; DONE?
    goto    $-1
    movf    ADRESH,w
    return
```

```
SENDPC
    Btfss   Pir1,TXif
    goto    $-1
    movwf   TxReg
    call    retardito
    return
```

```
RxW
    btfss   VT
    goto    $-1
    movf    portA,w
    movwf   LSB
    btfsc   VT
    goto    $-1

    btfss   VT
    goto    $-1
    movf    portA,w
    movwf   MSB
    btfsc   VT
    goto    $-1
    movlw   b'00001111'
    andwf   LSB,f
```

Appendix B: (continued)

```
    andwf      MSB,f
    swapf     MSB,w
    iorwf     LSB,w
    movwf    Wireless
    return

SendWdata
    CALL      AN_Resp
    Call      TxW
    return

TxW
    movwf    LSB
    movwf    MSB
    bcf      LSB,4
    swapf   MSB,f
    bcf      MSB,4
    movf     LSB,w
    call     txW
    movf     MSB,w
    call     txW
    return

txW2
    movwf    PORTB
    call     retardito
    bsf      TE
    call     retardito
    return

; ***** DELAY subroutines *****

delay    movlw    d'16'
         movwf    cont1
         movlw    d'255'
bucled2  movwf    cont2
         decfsz   cont2,f
         goto     $-1
         decfsz   cont1,f
         goto     bucled2
         return

retardito
         movlw    d'255'
         movwf    cont1
         decfsz   cont1,f
         goto     $-1
         return

end
```

Appendix C: Registers Configuration in Remote Microcontroller

TABLE C1: Register TRISB configuration in the remote microcontroller

	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
Connection	Analog input (AN6)	“Connected” indicator output	TX	TE output	Encoder digital output			
Configuration	1	0	1	0	0	0	0	0

TABLE C2: Register TRISA configuration in the remote microcontroller

	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
Connection	VT	Alert output	Select mode	***	Decoder digital input			
Configuration	1	0	1	X	1	1	1	1

TABLE C3: Register ANSEL configuration in the remote microcontroller

	X	AN6	AN5	AN4	AN3	AN2	AN1	AN0
	Digital	Analog	Digital	Digital	Digital	Digital	Digital	Digital
Configuration	0	1	0	0	0	0	0	0

TABLE C4: Register ADCON1 configuration in the remote microcontroller

	ADFM	ADCS2	VCFG1	VCFG0	****	****	****	****
	Right	clock	V_{REF+} AV_{DD}	V_{REF-} AV_{SS}	****	****	****	****
Configuration	1	0	0	0	0	0	0	0

Appendix D: MATLAB Code for Simulation Test

```
%% RESPIRATORY SENSOR TESTING/SIMULATION
clc;close all;clear all;
s = serial('COM4', 'BaudRate', 9600);
display('starting.....')
s.InputBufferSize=10000;
s.Timeout = 15;
fopen(s)
tic
out = fread(s);
volt = 5/256*out;
fclose(s)
toc
t = toc

samples = length(out)
st = t/samples
tt=(0:samples-1)*st;
average = mean(volt)
dev = std(volt)

figure(1)
plot(volt)
ylabel('Voltage [V]')
xlabel('sample')
title(['Average of ',num2str(average), ' and standard deviation of ',num2str(dev)])

figure(2)
plot(tt,out)
ylabel('Digital Conversion')
xlabel('Time')

up=median(out)+3
down=median(out)-3
y=1
i=1
    for i=1:(length(out))
        if y==1
            if out(i)>down
                x(i)=1
                y=1
            else
                x(i)=-1
                y=0
            end
        elseif out(i)>up
            x(i)=1
            y=1
        end
    end
```

Appendix D: (continued)

```
        else
            x(i)=-1
            y=0
        end
    end
end

figure(3)
plot(x)
j=1;
s=0;
for i=1:(length(x)-1)
    z(i)=(x(i)-x(i+1))/2
    if z(i)==1
        if s==0
            s=1
            timei(j)=tt(i)
            j=j+1
        else
            time(j)=tt(i)
            j=j+1
        end
    end
end
end

time(1)=timei(1)
tx=2
l=1
for tx=2:(j-1)
    dif(l)=time(tx)-time(tx-1)
    l=l+1
end

averagetime=mean(dif)
devtime=std(dif)
dif

averagefreq=1/averagetime*60
freq=[1./dif*60]
freqdev=std(freq)
```

Appendix E: Simulated Test Results for Square Signal Conversion

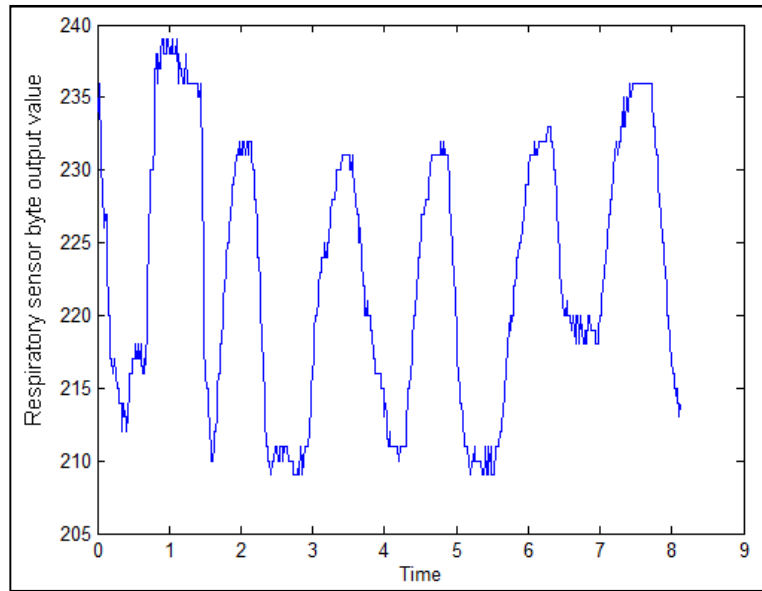


FIGURE E1: Simulated signal test E1

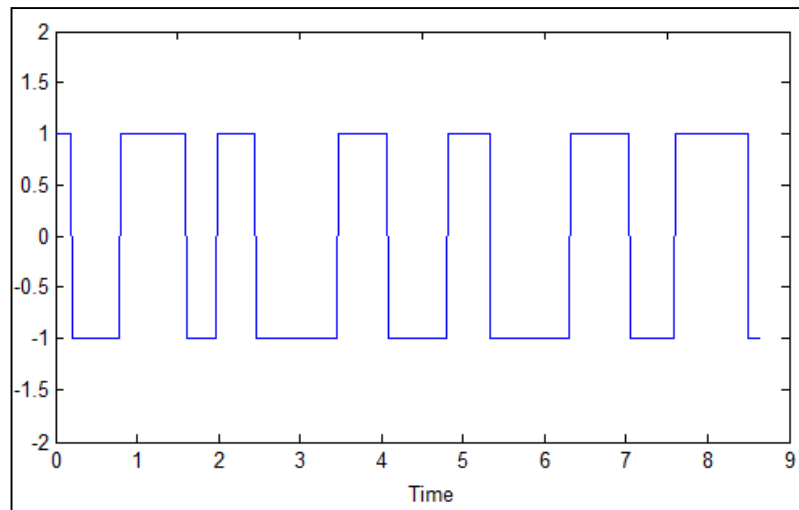


FIGURE E2: Conversion to square signal test E1

Appendix E: (continued)

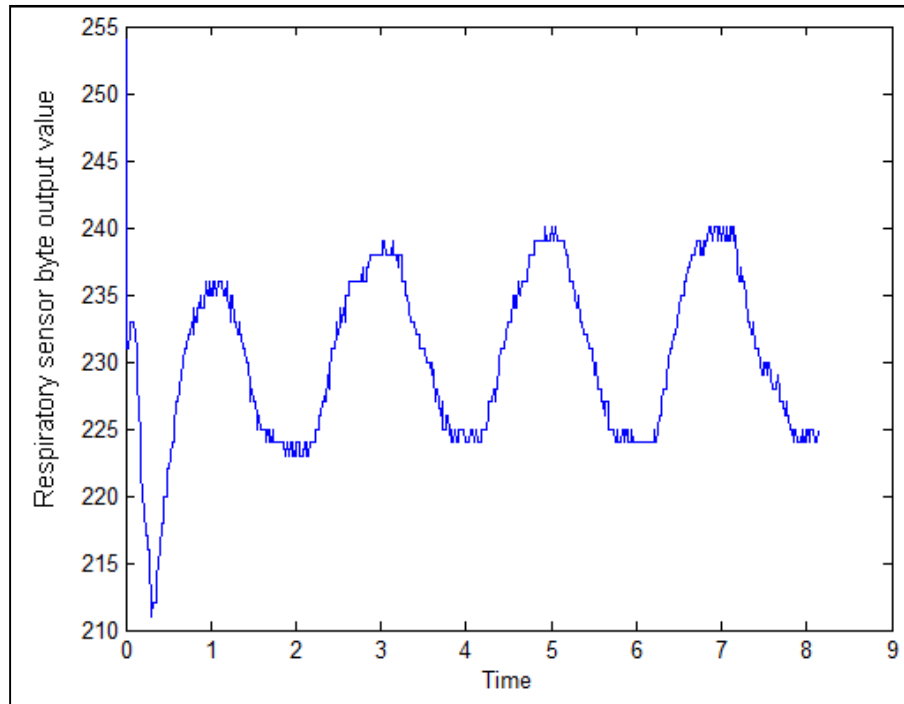


FIGURE E3: Simulated signal test E2

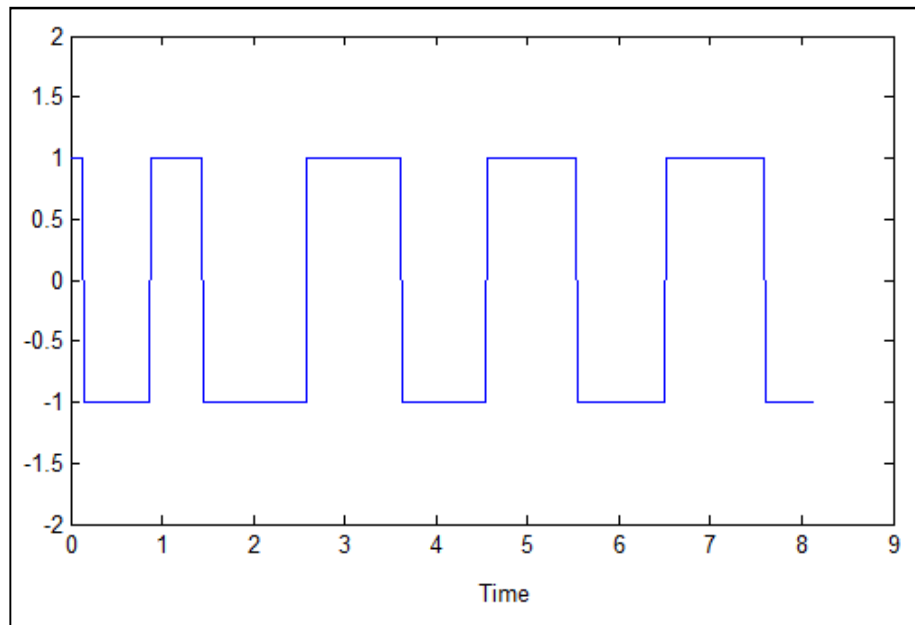


FIGURE E4: Conversion to square signal test E2

Appendix E: (continued)

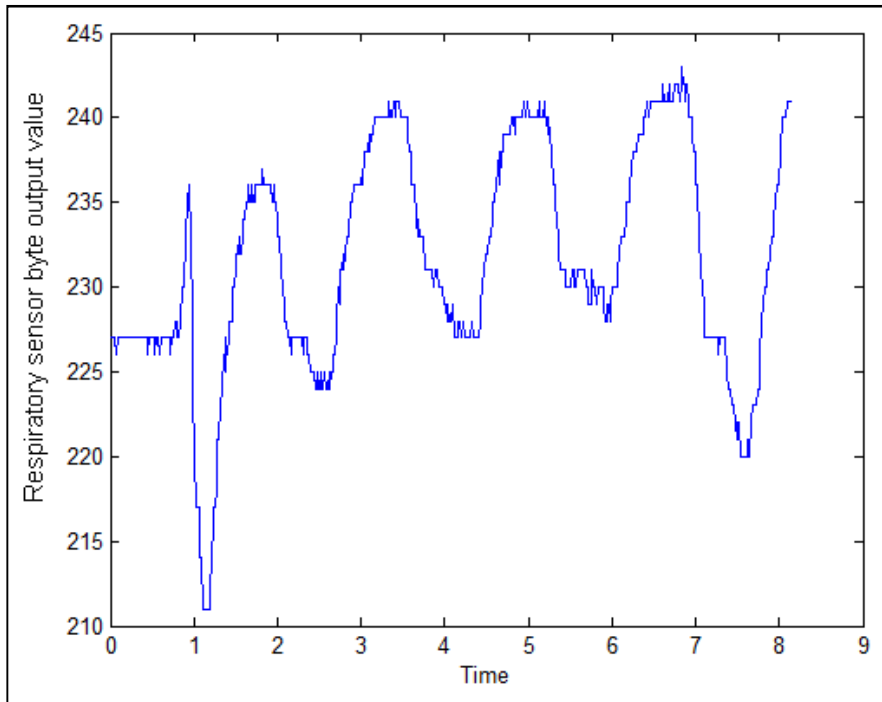


FIGURE E5: Simulated signal test E3

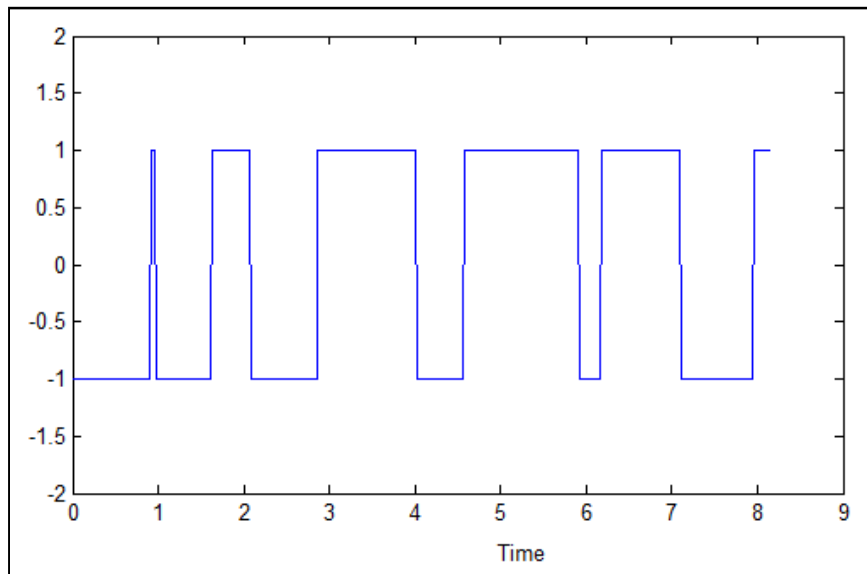


FIGURE E6: Conversion to square signal test E3

Appendix E: (continued)

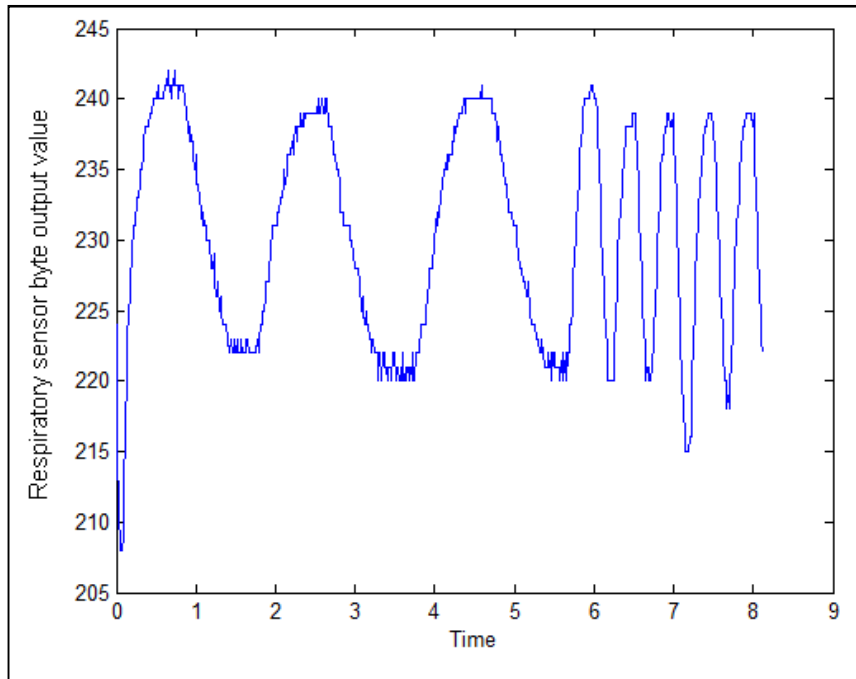


FIGURE E7: Simulated signal test E4

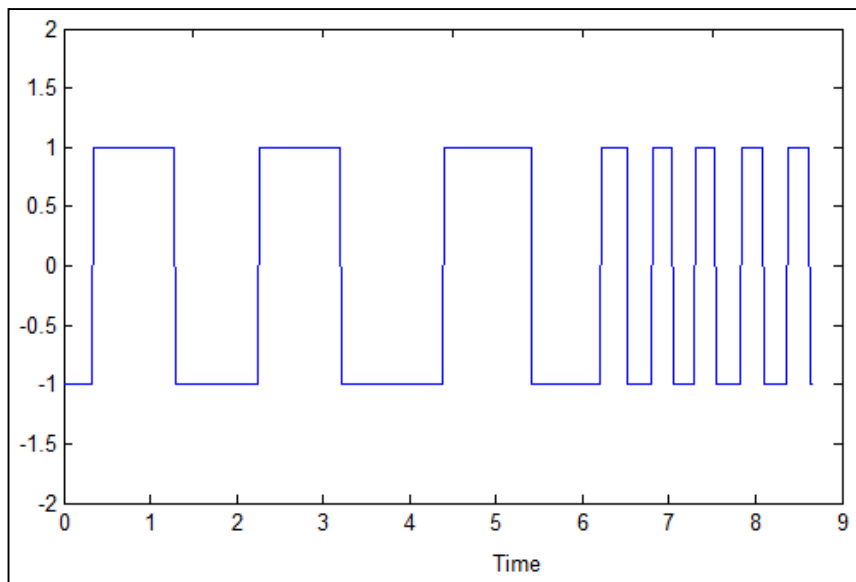


FIGURE E8: Conversion to square signal test E4

Appendix E: (continued)

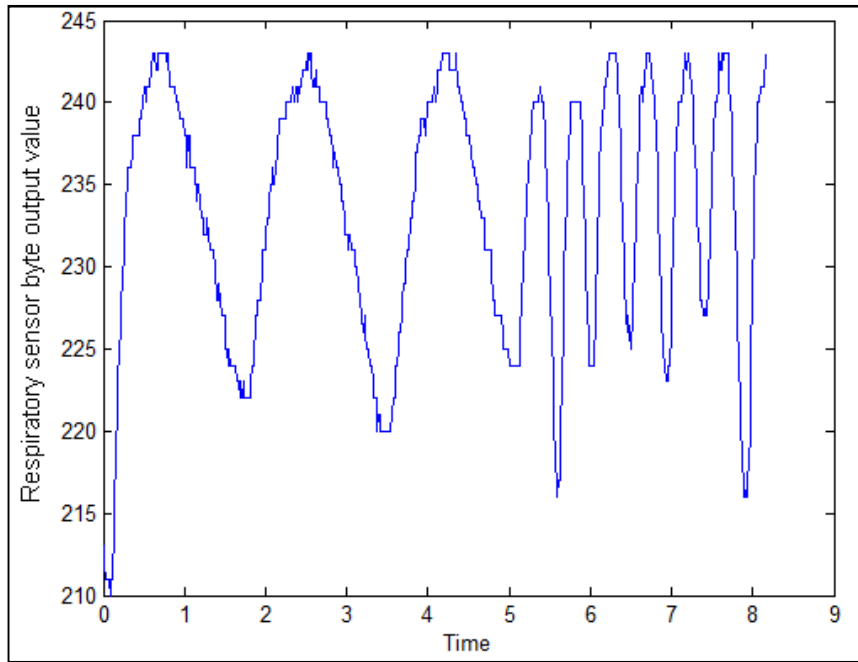


FIGURE E9: Simulated signal test E5

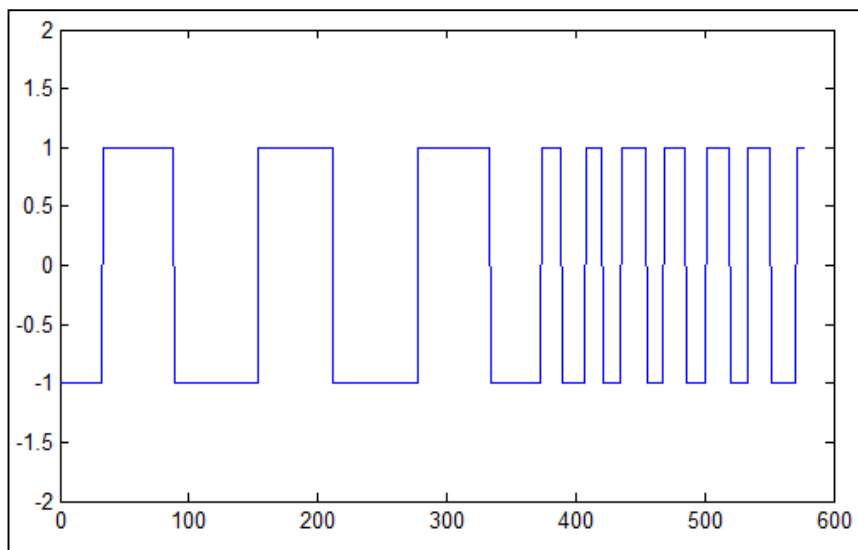


FIGURE E10: Conversion to square signal test E5

Appendix E: (continued)

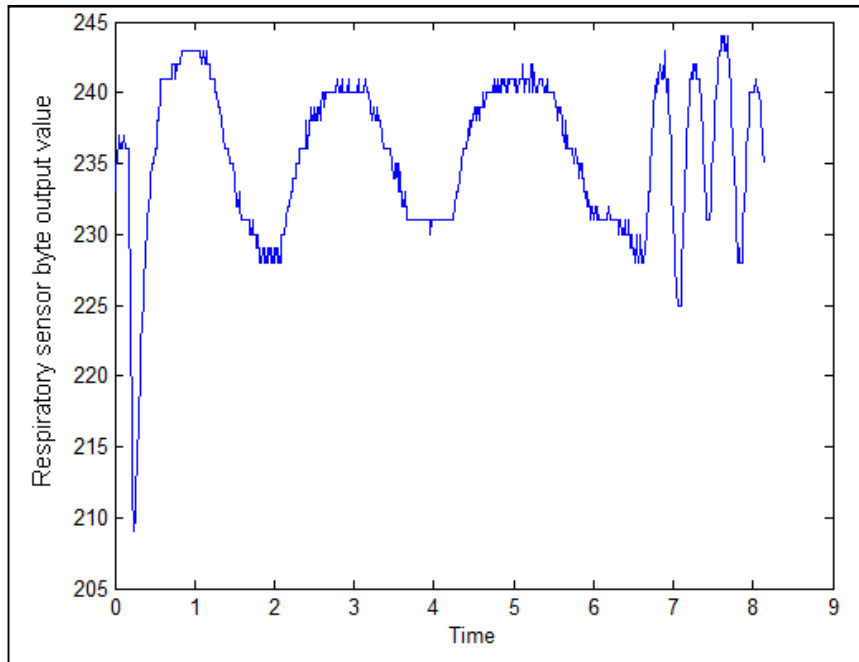


FIGURE E11: Simulated signal test E6

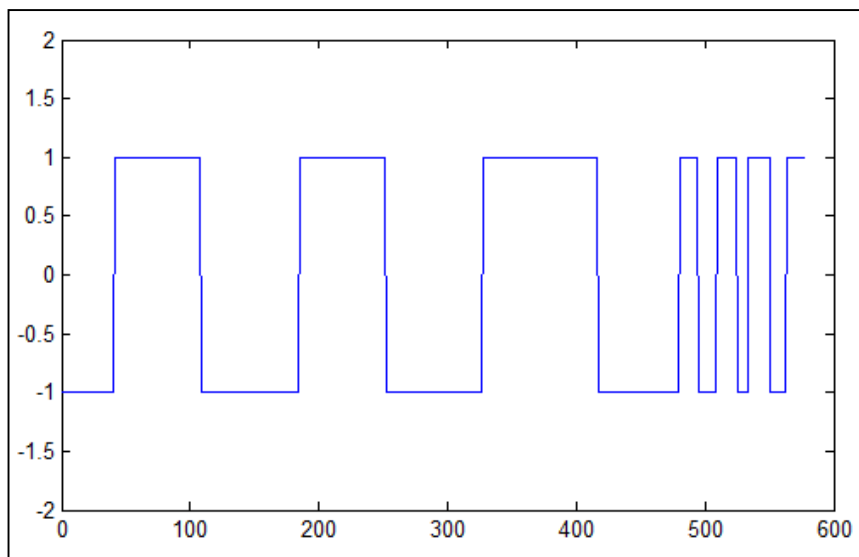


FIGURE E12: Conversion to square signal test E6

Appendix E: (continued)

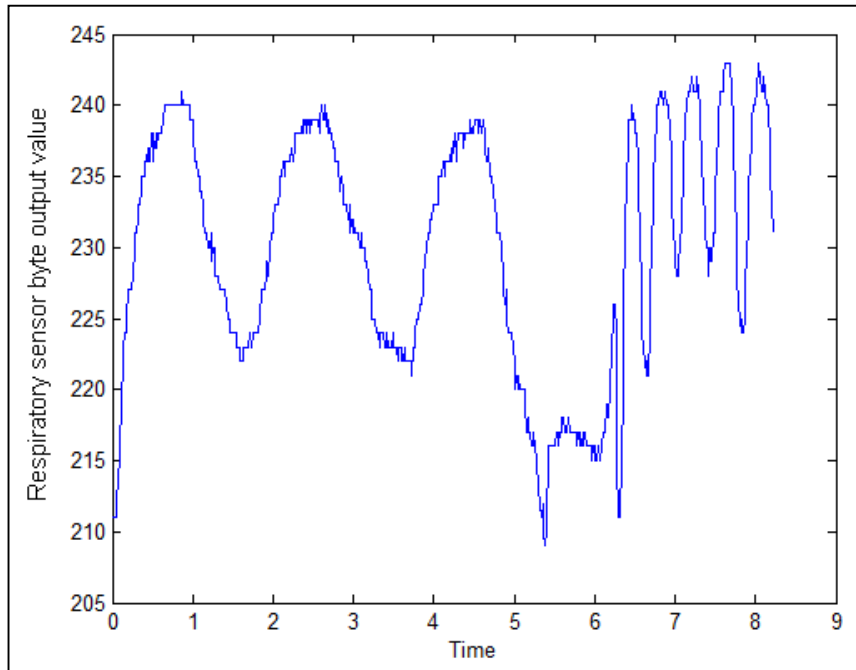


FIGURE E13: Simulated signal test E7

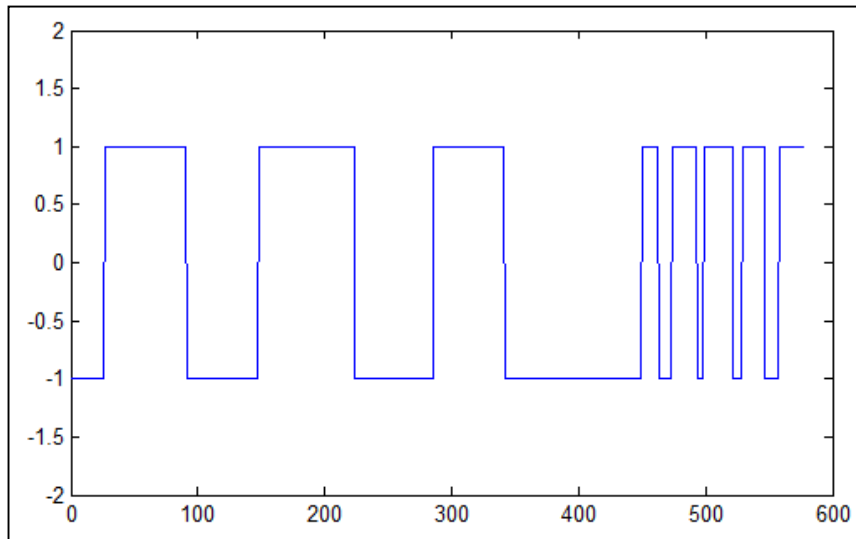


FIGURE E14: Conversion to square signal test E7

Appendix F: Additional Simulated Test Results

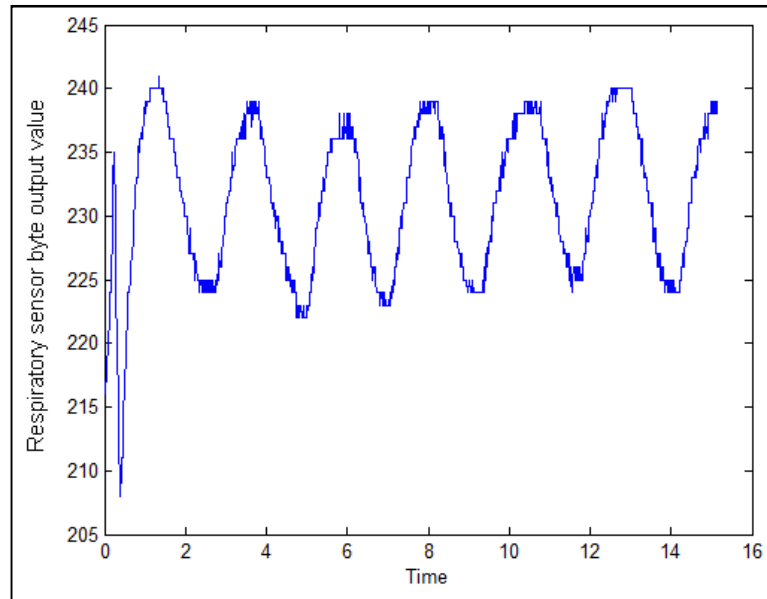


FIGURE F1: Simulated signal test F1

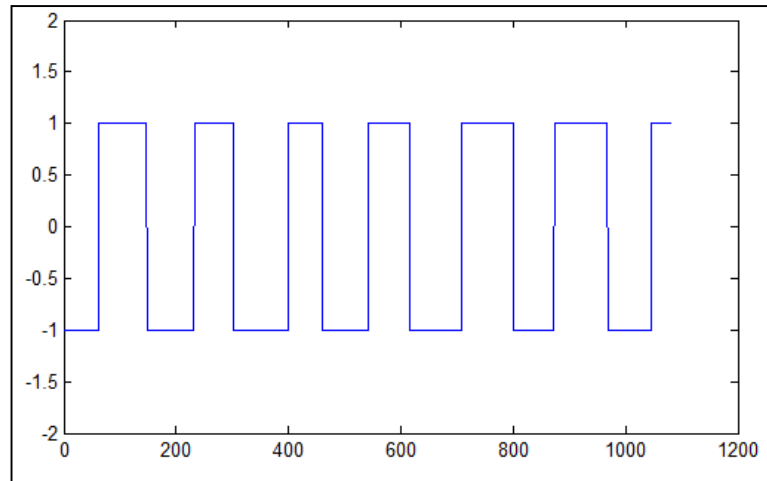


FIGURE F2: Conversion to square signal test F1

Appendix F: (continued)

```
dif = 2.1744 2.2165 2.1885 2.5813 2.3428
l = 6
averagetime = 2.3007
devtime = 0.1704
dif = 2.1744 2.2165 2.1885 2.5813 2.3428
freqdev = 1.8207
averagefreq = 26.0792
freq = 27.5935 27.0695 27.4166 23.2445 25.6107
|
```

FIGURE F3: Calculated parameters results test F1

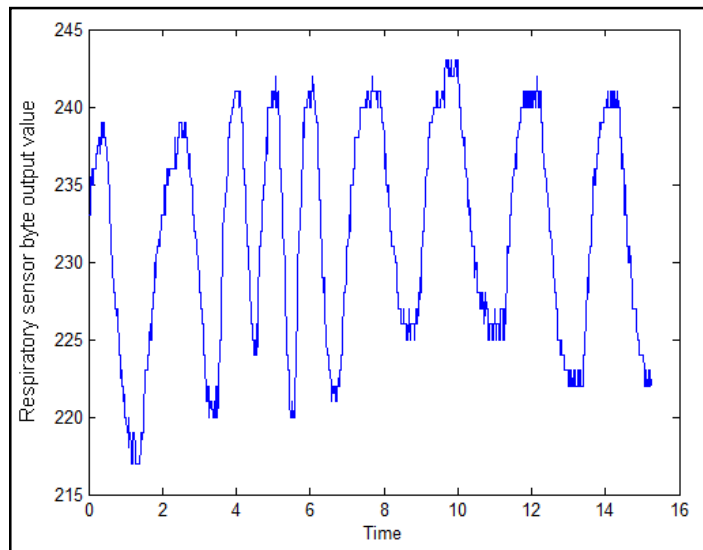


FIGURE F4: Simulated signal Test F2

Appendix F: (continued)

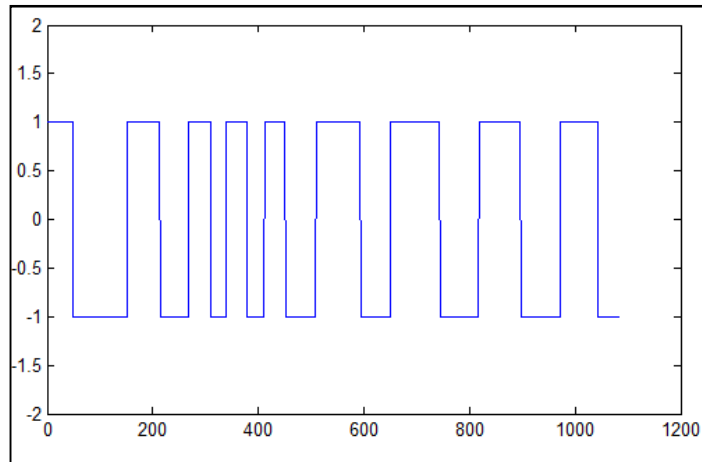


FIGURE F5: Conversion to square signal Test F2

dif =	2.3239	1.3521	0.9718	1.0141	2.0141	2.1127	2.1549	2.0704
l =	9							
averagetime =	1.7518							
devtime =	0.5481							
dif =	2.3239	1.3521	0.9718	1.0141	2.0141	2.1127	2.1549	2.0704
averagefreq =	34.2512							
freq =	25.8181	44.3749	61.7390	59.1665	29.7901	28.3999	27.8431	28.9795
freqdev =	14.8523							

FIGURE F6: Calculated parameters results test F2