MONITORING OF INDUSTRIAL PROCESS USING INTELLIGENT TOMOGRAPHY SYSTEM

By

MARC MACMILLEN

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

MONITORING OF INDUSTRIAL PROCESS USING INTELLIGENT TOMOGRAPHY SYSTEM

by

Marc Macmillen

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr. Taj Mohammad Baloch
Senior Lecturer
Electrical and Electronics Engineering
New Academic Block No 22
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh, Perak Darul Ridzuan, MALAYSIA

Dr Taj Mohd Baloch
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Marc Macmillen

ABSTRACT

The objective of this project is to come up with a monitoring system in a pipeline or vessel of an industrial process without intrusion of the pipeline/vessel by sensor equipment. This is done by taking a set of measurements using sensors that are distributed around the periphery of the vessel/pipeline. The information obtained is then used to reconstruct the cross sectional image inside the vessel/pipeline. In an industrial process, numerous data can be collected from materials (eg gas, liquid or solids) inside the pipeline/vessel such as flow regime, solids fraction profile, volumetric flow rate etc. Traditional monitoring systems required the sensing elements to be placed inside the vessel/pipeline thus making it intrusive. This intrusion will cause disturbance to the flow thus creating measurement errors. This project will utilize Electrical Capacitance Tomography (ECT) as its sensing method. The tomography system itself is divided into two parts namely data acquisition and data processing.

ACKNOWLEDGEMENT

I would like to convey my highest gratitude to my supervisor, Dr Taj Mohd Baloch for his guidance; Mr Patrick Sabastian for his big help and the other technicians namely Miss Siti Hawa, Miss Norliza and Mr Isnani, to name a few, for their support in making this final year project possible.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

CHAPTER 1
INTRODUCTION

1.1 Background of Study

In the mid-1980s work started that led to the present generation of Process Tomography systems. At University of Manchester, there began a project on Electrical Capacitance Tomography for imaging multi-component flows from oil wells. About the same time, a group at the Morgantown Energy Technology Center in the USA was designing a Capacitance Tomography system for measuring the void distribution in gas fluidized beds. The capacitance transducers used for both these systems were only suitable for use in an electrically non-conducting situation.

The basic aim of modern tomography is to determine the distribution of materials in some region of interest (eg. process vessels or pipeline) by obtaining a set of measurements using sensors that are distributed around the periphery. This information can be used to identify flow regime, solids fraction profile and volumetric flow rate. In industrial application, the materials could be oil or gas in a pipeline. Tomographic measurements are non-intrusive, perhaps penetrating the 'wall' of the vessel but not intruding into the medium, and also, ideally, non-invasive such that the sensors are located on the outside of the 'wall'.

An ECT systems attempts to compute the permittivity distribution by imposing a source of energy on the vessel from one orientation and a number of measurements are taken by distributed sensors to create a projection of data. The source is then moved to provide another projection and so on around the vessel until a frame of data is accumulated. Usually the frame of data is translated, in software, into a cross-sectional image representing the distribution of the materials.

1.2 Problem Statement

In industrial process, it is very common to have various kind of measuring techniques to monitor certain parts of the process within the plant for instance solid-laden gas flow, filtration/temperature monitoring and flame imaging to name a few. Traditional techniques required the measuring elements to be placed inside the process pipeline. This intrusive manner of measurement will create disturbance to the flow causing errors in measurements. Tomography technique is very advantageous as it is non-intrusive. It is able to provide cross-sectional distribution information which can later be derived to obtain flow regime, solid fraction profile, velocity profile and volumetric rate. ECT is based on measuring the change in capacitance to reconstruct the cross-sectional distribution of permittivity within the area of interest.

1.3 Objective and Scope of Study

This project is to develop a monitoring system that utilizes electrical tomography techniques for industrial process. The ECT consists of two main parts which are data acquisition and data processing. For data acquisition, capacitive values are obtained with the use of capacitance sensors distributed around the perimeter of the pipeline. These sensors electrodes are actually normal copper or aluminum plates and are connected to capacitance measuring circuit. In data processing, real time cross-sectional images are constructed with the help of image processing software such as Lab View or MATLAB. For this project, only data acquisition was done in order to obtain feasible data for image reconstruction. Image reconstruction itself is not included.

CHAPTER 2
LITERATURE REVIEW

Much of the early studies were based on literatures written by individuals who spent extensive duration of time for tomography development in the early years. One of the early published texts was PROCESS TOMOGRAPHY: Principles, Techniques and Applications (1995) written by R A Williams and M S Beck. The book describes in detail the fundamental of tomography. Summary of literatures, journals and conferences throughout the years by professional individuals can also be found in ELECTRICAL IMPEDEDANCE TOMOGRAPHY: Methods, History and Application (2005) written by David S Holder. Apart from published text, internet websites offers a great deal of information for tomography. These websites include www.tomography.manchester.ac.uk, www.tomoflow.com and www.itoms.com to name a few. These sites offer findings and even products and services for tomography applications.

CHAPTER 3
THEORY OF OPERATION

A basic capacitance-based tomographic imaging system consists of two parts – data acquisition (the primary sensor and the sensor electronics) and data processing (an image reconstruction computer). The functions of the sensor electronics include selection of electrode-pair combinations, measurement of capacitances between all possible combination pairs of the electrodes (and by doing so, interrogating different areas in the sensing volume), conversion of measured capacitance values into digital signals and interfacing the data acquisition part with the image reconstruction computer.

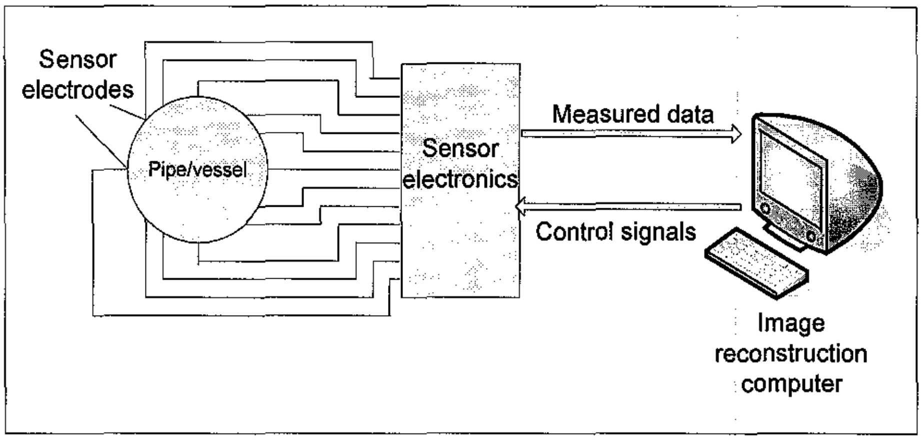


Figure 1: Basic ECT system [2]

3.1 About Capacitance and Permittivity

A capacitor is an arrangement of conductor plates that are insulated from each other so that electric charge can be stored. **Figure 2** shows two metal plates applied with a voltage from a battery. One plate will gain electrons and the other will lose electrons. If the battery is removed from the circuit, a voltage still exist across the plates one plate is negatively charged and the other positively charged. If the plates are now connected, a transient current will flow until there is no surplus or deficit of electrons on either plate. Thus capacitance is a measure of the amount of electric charge stored for a given electric potential and the relationship is given by:

Charge = capacitance x voltage

$$Q = C \times V$$

$$C = Q / V$$

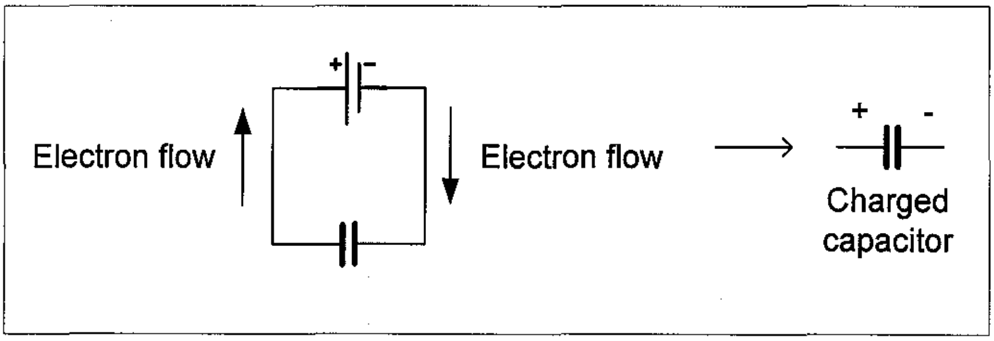


Figure 2: Charging of capacitor creates polarity

If any solids or liquid insulator is placed between the plates, the capacitance is increased and the factor by which the capacitance is increased is called the relative permittivity of the material between the plates. Permittivity relates to the material's ability to "permit" an electric field. The formula which can be used to calculate the capacitance is given as:

$$Capacitance, C = 8.84 \times \frac{\varepsilon_r A}{d} \qquad \textbf{(eq.1)}$$

ε_r = relative permittivity of insulating material

d = spacing between the plates

A = area of the plates

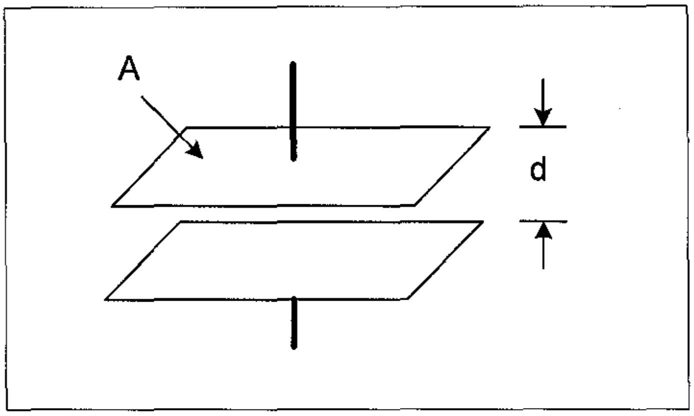


Figure 3: Basic construction of a capacitor

The basic of this project is based on this principle of capacitance and permittivity. As the medium inside the vessel changes in term phase, type, composition etc, the permittivity changes and the capacitance measured will also change accordingly.

3.2 Measurement Circuit

A stray-immune measurement circuit is used for capacitance measurement. This circuit measures only the capacitance between the selected pair of electrodes and is insensitive to the stray capacitances between the selected and the redundant electrodes and those between the selected electrodes and earth [1]. An example of a stray-immune circuit is the switched capacitor charged-transfer circuit which is used in this project. This circuit offer a fast data capture rate and because of its simple circuitry, parallel measurement channels can be used to increase the excitation frequency without a substantial increase in cost or complexity of the circuitry [2].

The principle of the basic measuring circuit (primary sensor) can be explained with reference to **Figure 4**. One electrode of the unknown capacitance (source electrode) is connected with a pair of CMOS switches, S1 and S2, and another (detecting electrode) is connected with switches S3 and S4. In a typical operating cycle, the switches S1 and S3 are first closed (S2 and S4 open) to charge the unknown capacitance, C_x to voltage V_c, and the charging current flows into the input (at virtual earth potential) of the current detector CDl where it is converted into a negative voltage output. In the second half of the cycle, switches S2 and S4 close (S1 and S3 open) to discharge C_x to earth potential. The discharging current flows out of the current detector CD2, producing a positive voltage output. This typical charge/discharge cycle repeats at a frequency, and the successive charging and discharging current pulses DC output voltage:

$$V_o = fV_c R_f C_x \qquad \textbf{(eq.2)}$$

where R_f is the value of the feedback resistors of the current detectors. The capacitors C_{in}(0.1 μF) ensure that the virtual earth potentials at the input remain stable during the high speed charge and discharge operation. Since, during the operation, the source electrode is always connected with low impedance supplies *(V,* and earth), and the detecting electrode always at virtual earth potential, stray capacitances have virtually no effect on the measurement [1].

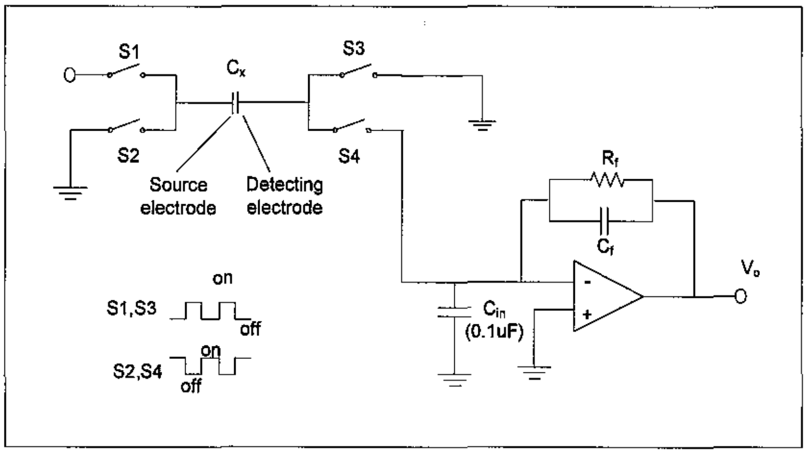


Figure 4: Basic capacitance measurement circuit [2]

3.3 Switching Circuit

The principle of the switching circuit is shown by **Figure 5**. Each switching channel consists of a MC14066 quad CMOS switch. For an 8-electrode system, sensor electrodes 2 to 8 are connected to the charge amplifier (capacitance measurement circuit) whereas electrode 1 is connected only to a pair of CMOS switches performing charge/discharge functions. The microcontroller PIC16F877A controls the status of the electrodes. Electrodes 2 to 7 can be selected as either the source or detecting electrodes; electrode 1 can be set to either active (source electrode) or idle (earthed), while electrode 8 always works as a detecting electrode.

First, electrode 1 is selected as the source and electrodes 2 to 8 as the detecting ones. The capacitances between electrode pair 1-2, 1-3, to 1-8 are measured one by one. These parallel measurements are independent because the capacitances are determined from the charge and discharge currents through the detecting electrodes and these are all held at virtual earth potential. The 7 parallel channel outputs are selected one by one by the microcontroller and fed to the charge amplifier to obtain the resulting output voltage. The microcontroller will then convert this analog voltage value to digital form with its built in Analog to Digital (A/D) function. In the second step, electrode 2 is selected as the source (electrode 1 set to idle) and electrodes 3 to 8 as the detecting ones. Capacitances between 2-3 to 2-8 are measured and the resulting voltage output is converted into digital signals. This process continues until electrode 7 is selected as the source and electrode 8 is the detecting one, and the capacitance between electrodes 7 and 8 is measured. Generally for an n-electrode system, the number of independent measurements N is given by the following equation of combination:

$$N = n(n - 1)/2 \qquad \textbf{(eq.3)}$$

For an 8-electrode system, this number is 28.

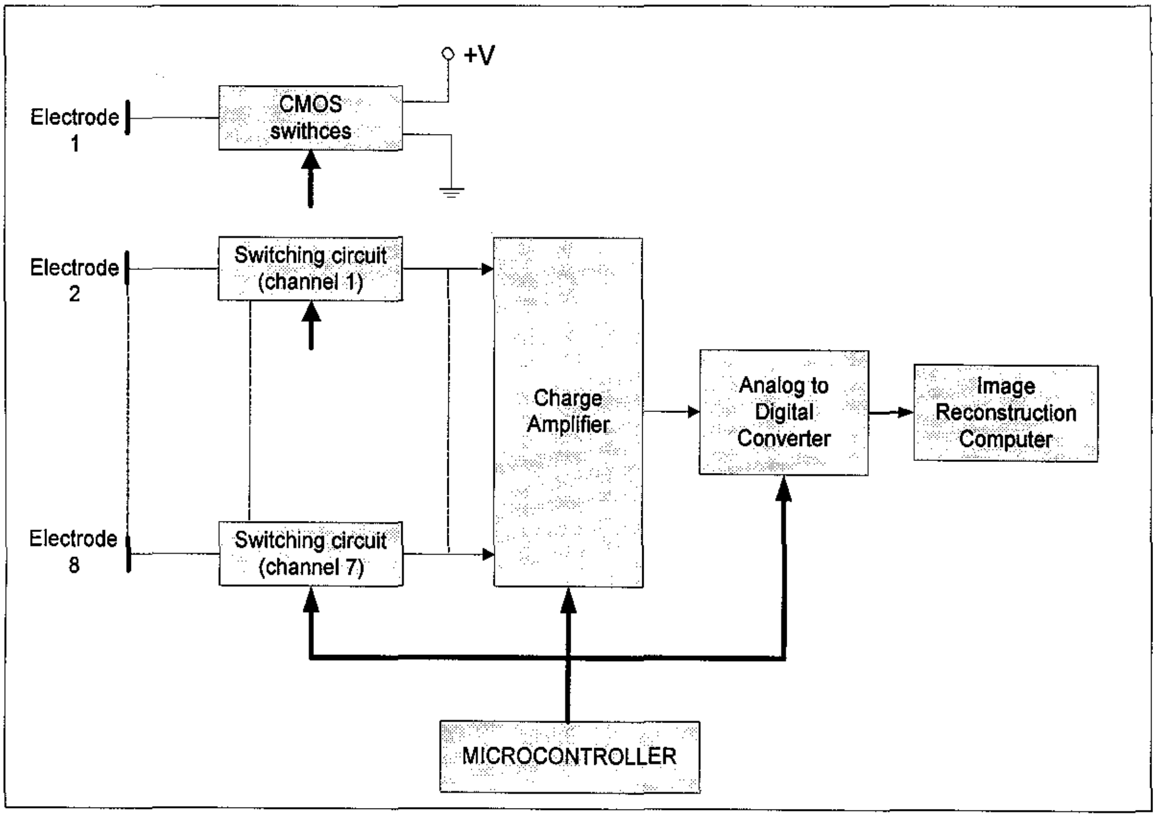


Figure 5: Block diagram of the ECT system

3.4 Offset adjustment

Due to unbalances in the op-amp's internal transistors and resistors, an error voltage is present at its output even when 0V is applied to its input. One way to eliminate this problem is by implementing offset resistors network at the non-inverting input of the op-amp. The full range of adjustment for voltage applied to the non-inverting input is ±15V(R2/(R1+R2) **(eq.4)**.

Using Thevenin Theorem:

Given $V_{ios} = 10\text{mV}$

$V = |V_{cc}| = |V_{ee}| = 12\text{V}$

$$R_{pmax} = \frac{R_p}{2} \| \frac{R_p}{2} = \frac{R_p}{4} \qquad \textbf{(eq.5)}$$

$$\frac{V}{V_{ios}} = \frac{R_1}{R_2} = \frac{12}{10\text{m}} = 1.2\text{k} \qquad \textbf{(eq.6)}$$

$R_2 < 100\Omega$

Therefore, $R_2 = 10\Omega$, $R_1 = 12\text{k}\Omega$

$$R_{pmax} = \frac{R_1}{10} = 1.2k\Omega$$

$$VR = 4 \times R_{pmax} = 4.8\text{K}\Omega \approx 5\text{K}\Omega \qquad \textbf{(eq.7)}$$

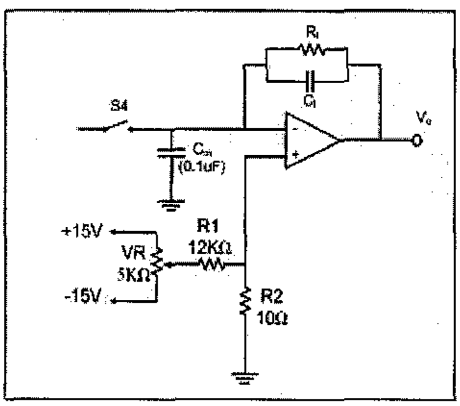


Figure 6: Zero offset network

CHAPTER 4
METHODOLOGY

4.1 Vessel Prototype

The first vessel prototype was very large and all the electrodes were fitted on the interior of the vessel. This had broken the conditions of the correct electrodes design. The electrodes should be non-invasive and non-intrusive. Non-invasive means the electrodes should not necessitate rupture of the wall of the process vessel, for instance, by the introduction of the electrodes. Non-intrusive means the electrodes should not disturb the nature of the process being examined. Because of these requirements, a second vessel was built.

The vessel was made out of common plastic cylindrical containers. The vessel was fitted with eight electrodes made out of aluminium plates each with a dimension of 25mm x 80mm. To reduce stray capacitance around the electrodes and the body of the vessel, screening plate was placed around the electrodes with no contact with those electrodes. This screening plate was also made out of aluminium. The outlay position of the electrodes and the screening plate is shown in **Figure 7**. To connect the electrodes to the measuring circuit, coaxial cables were used instead of a normal unshielded cable. The internal main conductors were connected to each of the electrodes respectively and the outer shielding mesh was connected to the screening plate. **Figure 9** shows the finished product of the second vessel prototype. On the other end of the cable, the shielding mesh will be connected to ground using BNC connectors. The use of coaxial cable with mentioned connection was to reduce stray capacitance around the vessel and along the cable itself. Since Electrical Capacitance Tomography is very sensitive to stray capacitance, this grounding method is necessary in order to obtain a more accurate reading.

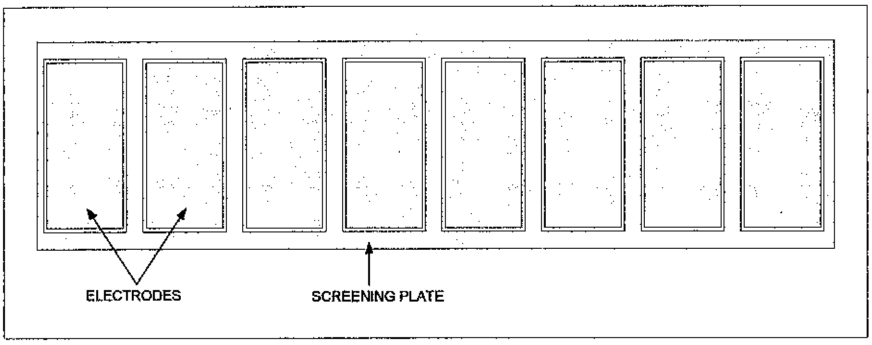


Figure 7: Outlay of the electrodes and the screening plate.

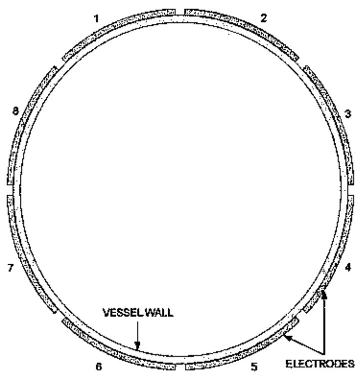


Figure 8: Electrodes position around the vessel

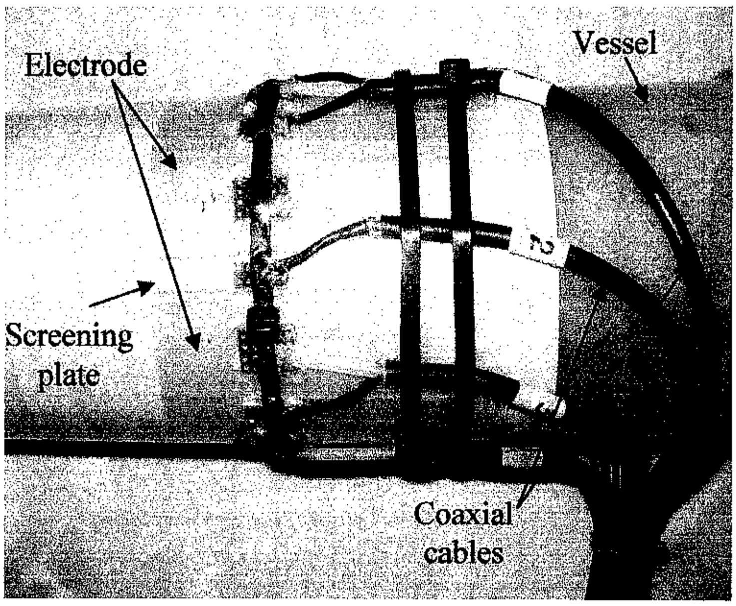


Figure 9: Completed vessel fitted with 8 electrodes connected with coaxial cables.

4.2 Measuring Circuit

In the final design, only a single measuring circuit was built. All electrodes were connected to this measuring circuit through individual switching channels.

Initially, measuring circuit for every electrode was made but some problems occurred. TL084 quad amplifier ICs were used for all the seven detecting electrodes (electrodes 2-8) and the circuits were built on the same set of breadboards. Doing this, created coupling problem in the design since all amplifiers shared the same power supply. Few decoupling methods were used such as using separate voltage regulators and decoupling capacitors but the problem persisted. The final decision was to have a single measuring circuit for the detecting electrodes. Having a single measuring circuit meant that voltage measurement could not be done simultaneously on all detecting electrodes whenever a source electrode is driven. The measurement must be done one by one controlled by a microcontroller. The microcontroller used was a PIC16F877. To reduce noise effect from stray capacitance, an operational amplifier with very low input noise is preferred. LM833 dual op-amp was used in the design. It has a relatively low input noise of 4.5nV/√Hz.

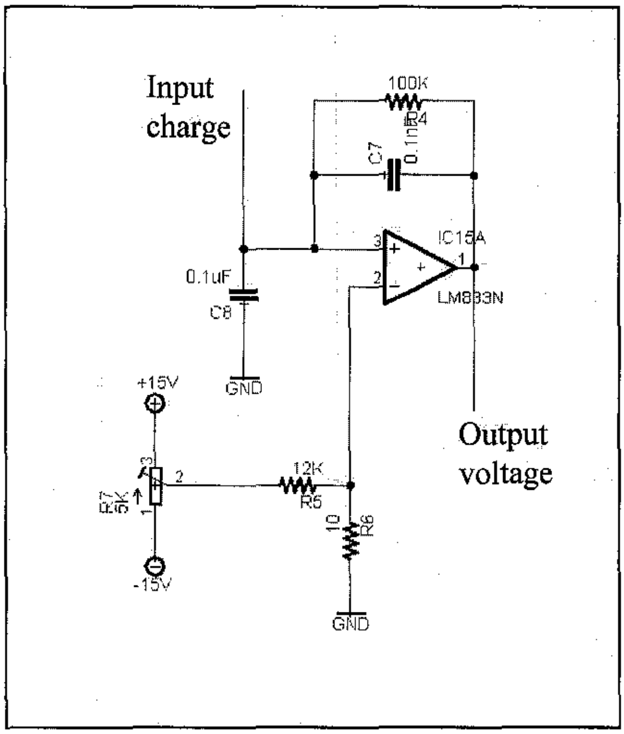


Figure 10: Measurement circuit schematic diagram.

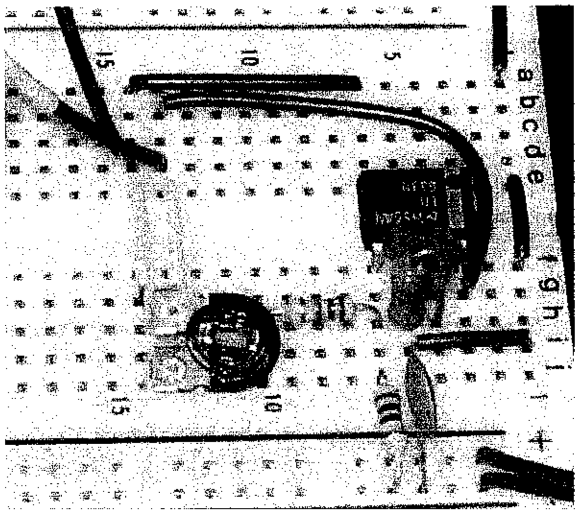


Figure 11: Measuring circuit based on charge amplifier design.

4.3 Switching Circuit

As mentioned in the earlier chapter, charge/discharge process was done using MC14066 CMOS switches. These switches were controlled by microcontroller PIC16F877A. Previously, logic control based on logic ICs was made but due to massive wiring and strict wiring requirements of logic gates of this size, the system could not work properly even though it work properly in simulation done using Electronic Workbench program. Because of this, a microcontroller was used instead. It has greatly reduced the size of the whole switching circuit. Modifications were also possible since the microcontroller can be reprogrammed according to needs. The complete schematic diagram is shown in **Appendix C**.

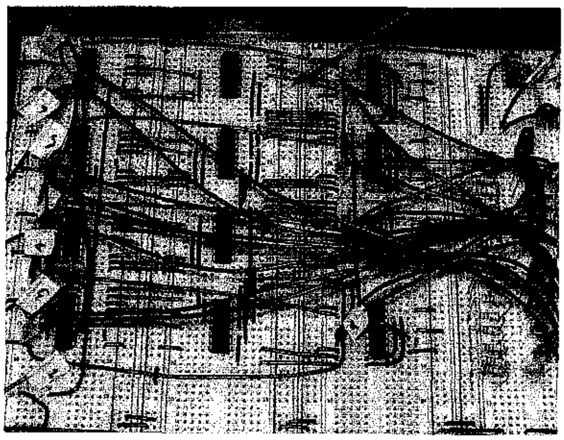


Figure 12: Switching circuit built on breadboard

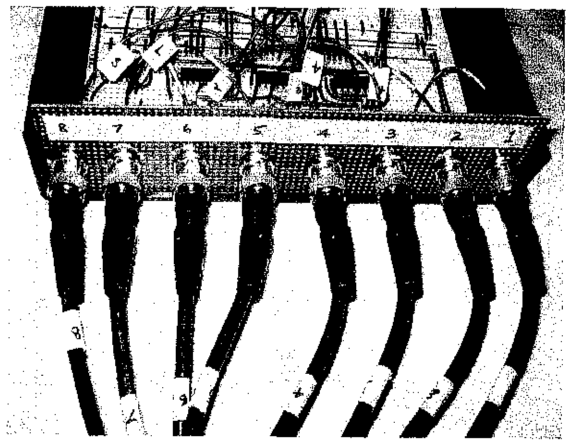


Figure 13: Electrodes connected to the switching circuit using coaxial cables

4.4 Microcontroller PIC16F877

This microcontroller is the heart of the whole system as it controls the correct sequence of the switching for the measuring to be done correctly. It gets a 24 MHz clock from an external crystal oscillator. Even with this high frequency clock, the microcontroller is only capable of driving the CMOS switches at a frequency of approximately 273 KHz due to its physical limitation and the big cycle number it needs to generate for the switching cycle per electrodes pair. Port B and Port D are used to control the CMOS switches. The C code for the microcontroller is attached in Appendix A.

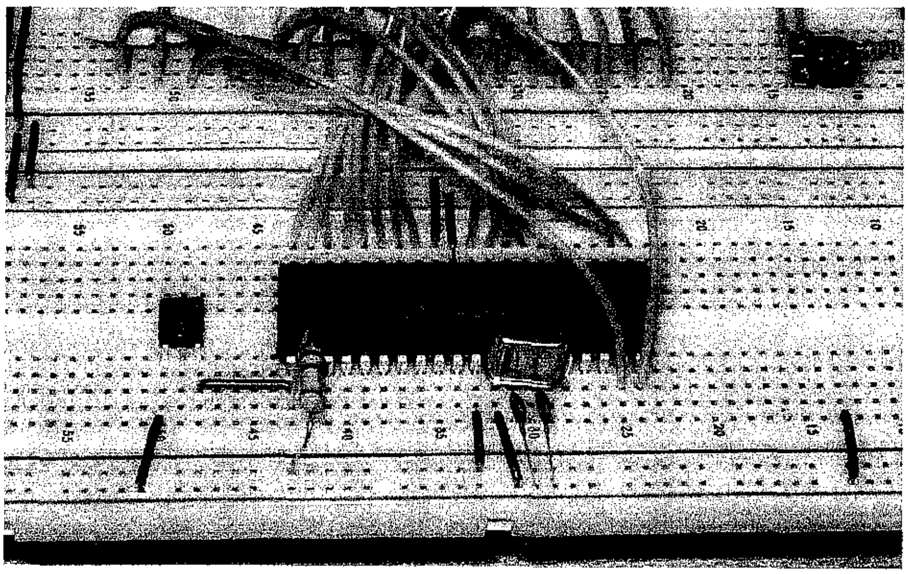


Figure 14: PIC16F877 microcontroller with 24MHz clock

4.5 Analog to Digital Converter

From the measuring circuit, analog output voltage is obtained which is proportion to the capacitance value of the medium inside the vessel sensed by the electrodes. This result can be shown in real time using a digital multi-meter in real time. But in order to save the result automatically, the result must be stored in a computer. This is done by first converting the analog output into digital output which can later be transferred to the computer using an Analog to Digital Converter (ADC). For this purpose, microcontroller PIC16F877A can be used. It has a built in ADC function which can perform the conversion process. It is the same microcontroller that is used for the switching circuit.

Pin A0 of Port A of the microcontroller is used for the analog input and the digital output is from Pin C6 of Port C. First, Port A must be predefined as ADC input by specifying Pin A0 as the input pin. Pin A3 is set as the reference voltage input used for the ADC conversion. For a reference voltage of 5V, Pin A3 is left as it is as the Vcc if the microcontroller is also 5V and the voltage reference assumes this voltage. For other values of voltage reference, Pin A3 must but specified as the voltage reference input. The C code for this ADC is shown in **Appendix B**.

Serial communication is used for the digital output and Pin C6 is set as the transmitter for this method. The digital output obtained after the A/D conversion is sent out to Pin C6. From Pin C6, the digital data is sent to the computer through MAX232 chip. This chip is to interface hardware and the computer using serial communication. HyperTerminal is a program used to display the result in the computer.

Formula use for this conversion is:

$$\frac{V_{in}}{V_{fullscale}} = \frac{DigitalOutput}{2^n - 1} \qquad \textbf{(eq.8)}$$

n = number of bit used

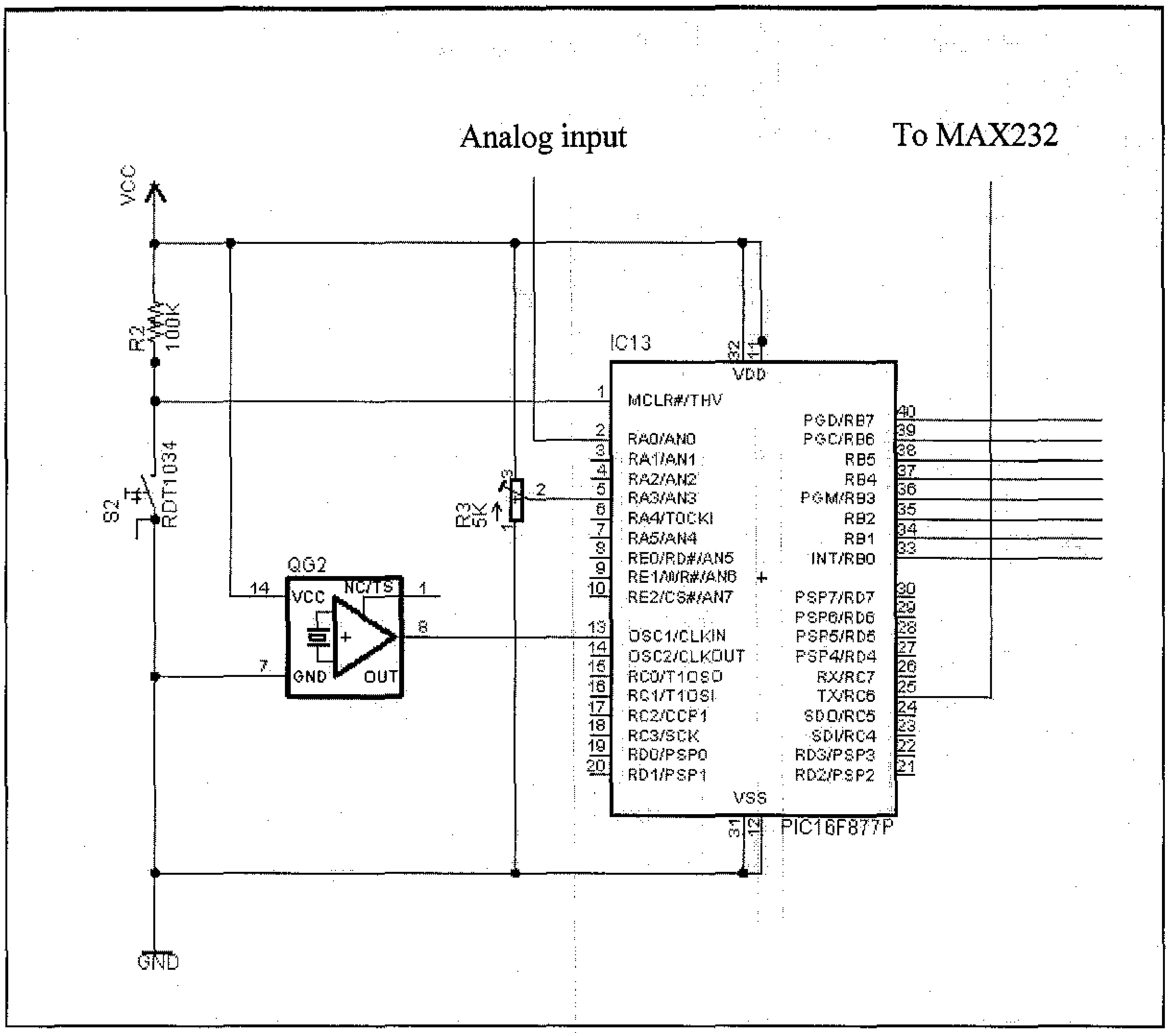


Figure 15: Schematic diagram of ADC using PIC16F877A.

During the compiling of the C code for the ADC function, it was found that the built in memory (ROM) of the PIC16F877A was not sufficient. The microcontroller had already spent around half of it memory for the switching circuit and it did not have enough spare memory. To overcome this problem, PIC18F452 was used instead separately just for the ADC function. This means the switching circuit uses one PIC16F877A and ADC used one PIC18F452. A total of two microcontrollers were uses for this ECT system. The PIC16F452 is a pin for pin replacement for a 16F877A but with much bigger memory.

4.6 HyperTerminal

HyperTerminal is a program that can be used to connect to other computers, Telnet sites, bulletin board systems (BBSs), online services, and host computers, using either your modem or a null modem cable. For this ECT system, it is used to connect to the system through RS232 cable and to display the data in real time.

In order to use the HyperTerminal, it must beforehand be setup according to the correct configuration:

- Create a new HyperTerminal connection.
- Enter the desired connection name.
- Chose the desired icon.
- Chose appropriate COM port which will be used to connect to the RS232 cable.
- The COM port properties must be set according to the setting configured in the C codes used for the serial communication setup for A/D converter.

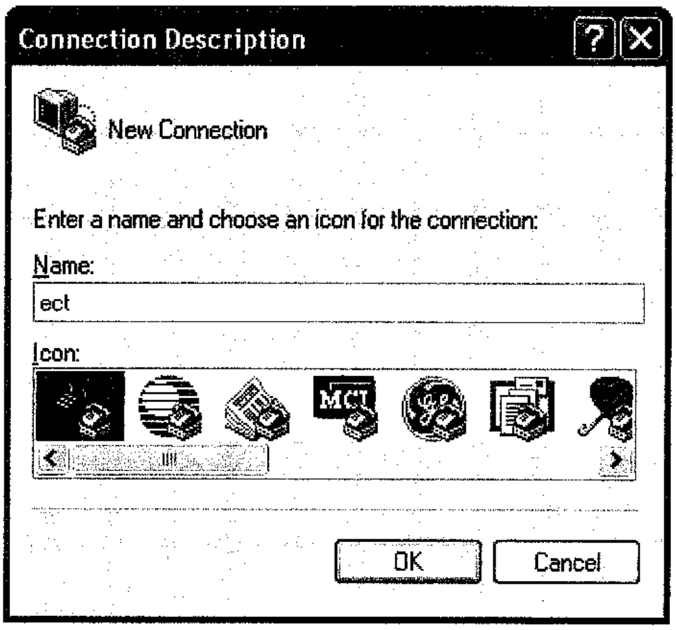


Figure 16: Creating a name and icon for the new connection.

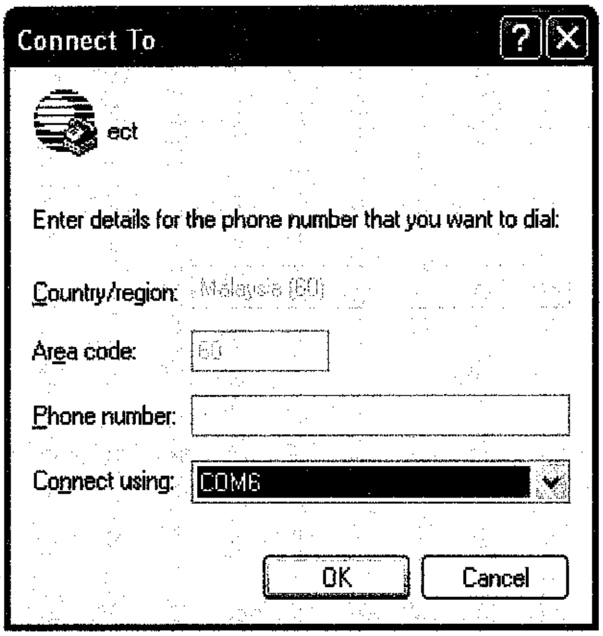


Figure 17: Select the correct COM port used for the computer.

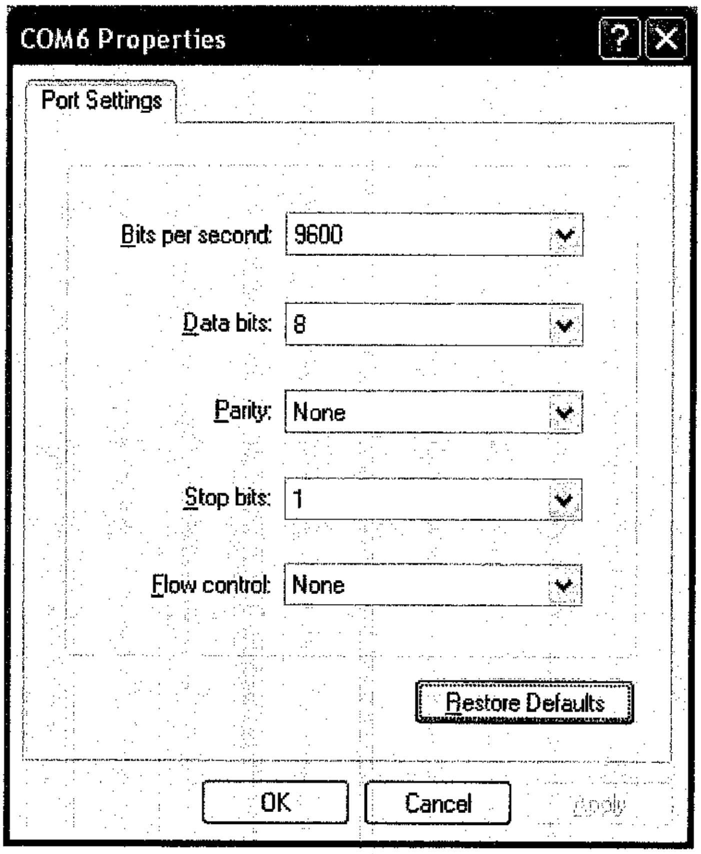


Figure 18: Set the right setting for the COM properties.

4.7 Grounding and Shielding

This ECT system is very sensitive to noise such as stray capacitance. Stray capacitance exists around the vessel, in between electrodes and on the surface of the cables connecting the electrodes to the switching circuit. The effect of this stray capacitance was obvious when little precaution was taken to overcome it. During the initial experiments, voltage measurements varied significantly whenever any part of the vessel or the cables connecting the electrodes to the measuring circuit were touched or even when got closed to. The higher the switching frequency used in the system, the more sensitive the system is to stray capacitance as shown in **Equation 1**.

Some methods were implemented in order to minimize the effect of stray capacitance to the system:

a) Screening plate was placed in between electrodes and around the vessel. This plate is then grounded.

b) Using amplifier with very low input noise. LM833N dual op-amp was used. It has an input noise of $4.5nV/\sqrt{Hz}$.

c) All electrodes were connected to the switching circuit using coaxial cables. The outer mesh of the coaxial cables was used to connect the screening plate to ground. This mesh also served to direct stray capacitance along the cable to ground. Appropriate SMA connectors were used to terminate the connection of the coaxial cables to the switching circuit.

CHAPTER 5

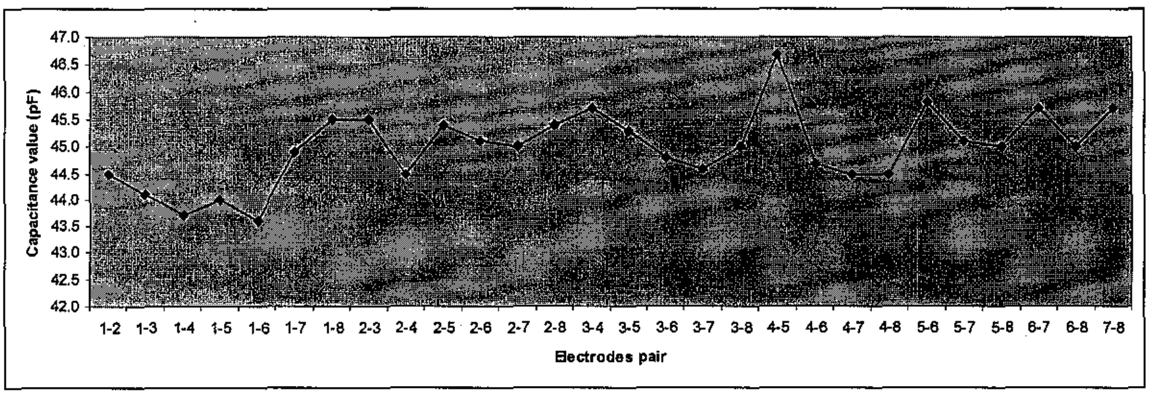
RESULTS AND DISCUSSION

Experimental work was done on the vessel. Rice grains were used as the medium being tested. 2/3 of the vessel was filled with the rice. Using a capacitance meter or Q-Meter, values of capacitance were measured according to respective electrodes pairs manually without any switching. The results are shown in **Table 1** and **Graph 1**.

Note: Units in pF

1-2	44.5												
1-3	44.1	2-3	45.5										
1-4	43.7	2-4	44.5	3-4	45.7								
1-5	44.0	2-5	45.4	3-5	45.3	4-5	46.7						
1-6	43.6	2-6	45.1	3-6	44.8	4-6	44.7	5-6	45.8				
1-7	44.9	2-7	45.0	3-7	44.6	4-7	44.5	5-7	45.1	6-7	45.7		
1-8	45.5	2-8	45.4	3-8	45.0	4-8	44.5	5-8	45.0	6-8	45.0	7-8	45.7

Table 1: Capacitance values between electrodes pair from pair 1-2 to 7-8.



Graph 1: Plotted values of the capacitance measurements.

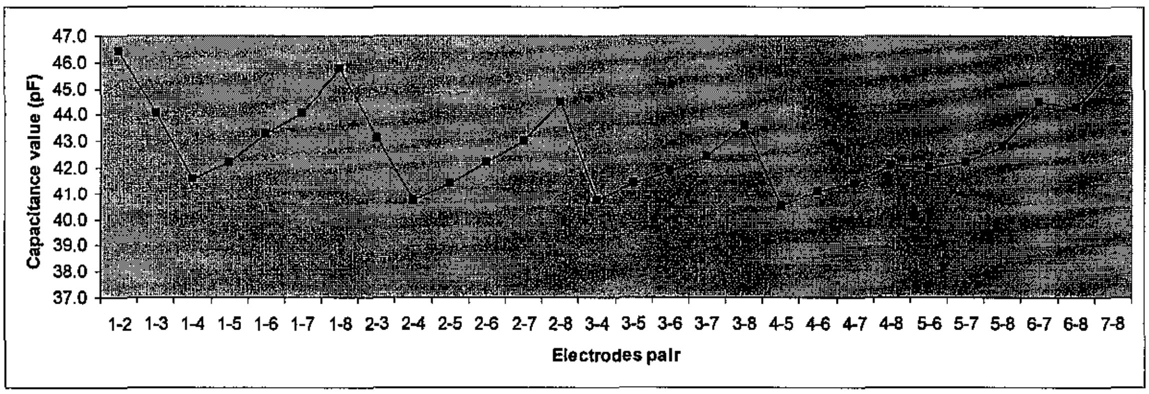
As shown in **Graph 1**, there is no significant trend in the measurements. It was noticed that during the experiment, the value of the capacitance changed whenever the vessel or the cables were held by hand. This indicated that the vessel was very sensitive to surrounding disturbance and the capacitance value measured was easily affected by stray capacitance surrounding the vessel. Touching the vessel created an electric loop from the vessel, through the human body and to the ground. A better grounding method was required for the vessel in order to eliminate this problem.

Another set of measurement was done. This time great care was taken in order not to touch any part of the vessel or the coaxial cable. The result is shown in **Table 2** and **Graph 2**.

Note: Units in pF

1-2	46.4												
1-3	44.1	2-3	43.1										
1-4	41.6	2-4	40.7	3-4	40.7								
1-5	42.2	2-5	41.4	3-5	41.4	4-5	40.5						
1-6	43.3	2-6	42.2	3-6	41.9	4-6	41.1	5-6	42.0				
1-7	44.1	2-7	43.0	3-7	42.4	4-7	41.4	5-7	42.2	6-7	44.5		
1-8	45.8	2-8	44.5	3-8	43.6	4-8	42.1	5-8	42.8	6-8	44.3	7-8	45.8

Table 2: Second measurement



Graph 2: Second measurement plot

Few more experiments were done. The vessel's coaxial cables were properly connected with BNC connectors and were fitted on a piece of strip board in order to maintain the distance of each cable to make sure all the cables were in place during experiment.

Few sets of measurements were done in order to obtain the capacitance values and the voltage values using a single measurement circuit with the switching provided by a function generator. The function generator was used to activate/deactivate the CMOS switches at a frequency of 1MHz. The measurements were done pair by pair manually. Below are the results of the experiments:

1-2	46.7												
1-3	45.5	2-3	44.7										
1-4	43.5	2-4	42.5	3-4	42.3								
1-5	44.4	2-5	43.2	3-5	42.6	4-5	41.3						
1-6	45.1	2-6	43.9	3-6	43.3	4-6	41.8	5-6	42.8				
1-7	46.0	2-7	44.7	3-7	44.1	4-7	42.4	5-7	43.2	6-7	44.4		
1-8	47.4	2-8	46.1	3-8	45.3	4-8	43.6	5-8	44.4	6-8	45.1	7-8	46.6

Table 3: Experiment 1 - Capacitance measurement with empty vessel. All values in pF

1-2	49.9												
1-3	48.4	2-3	47.7										
1-4	46.5	2-4	45.4	3-4	45.5								
1-5	47.3	2-5	46.0	3-5	45.5	4-5	44.7						
1-6	48.0	2-6	46.8	3-6	46.2	4-6	44.7	5-6	46.2				
1-7	48.9	2-7	47.7	3-7	46.9	4-7	45.2	5-7	46.1	6-7	47.7		
1-8	50.6	2-8	48.9	3-8	48.2	4-8	46.3	5-8	47.2	6-8	48.2	7-8	49.7

Table 4: Experiment 2 - Capacitance measurement with vessel filled with rice grain. All values in pF.

1-2	51.1												
1-3	49.4	2-3	48.6										
1-4	47.4	2-4	46.3	3-4	46.2								
1-5	48.0	2-5	46.8	3-5	46.3	4-5	45.0						
1-6	49.0	2-6	47.7	3-6	47.2	4-6	45.4	5-6	46.4				
1-7	49.8	2-7	48.6	3-7	47.9	4-7	46.2	5-7	46.8	6-7	48.0		
1-8	51.3	2-8	49.8	3-8	49.2	4-8	47.3	5-8	47.9	6-8	49.0	7-8	50.1

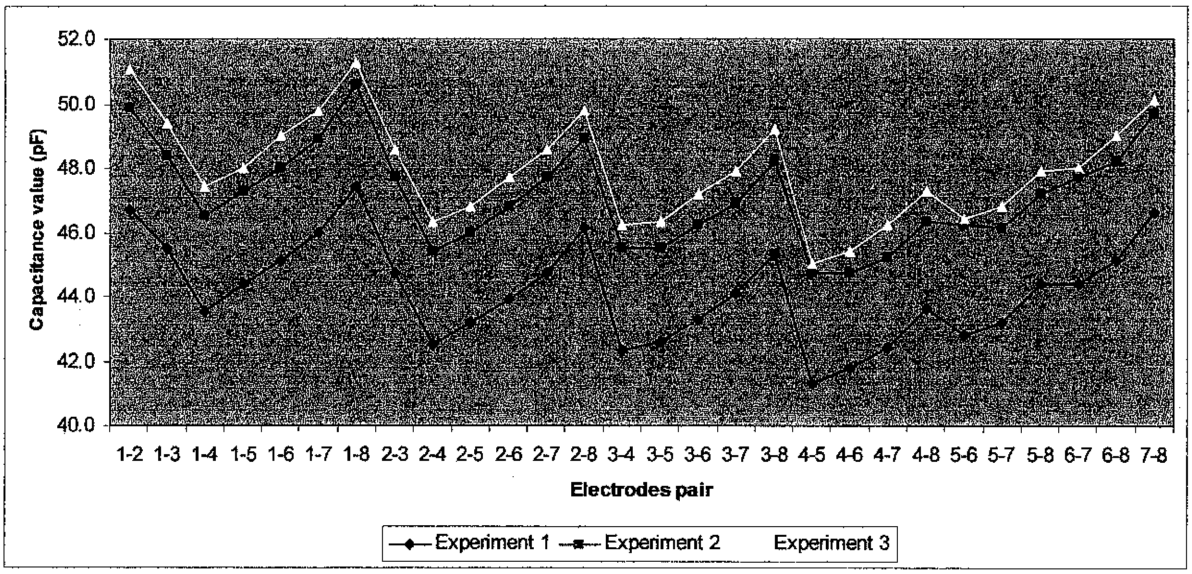
Table 5: Experiment 3 - Capacitance measurement with tap water bottle inside the rice grain filled vessel. All values in pF.

1-2	544.10												
1-3	540.71	2-3	539.06										
1-4	535.92	2-4	531.77	3-4	531.49								
1-5	535.39	2-5	532.06	3-5	530.31	4-5	527.47						
1-6	538.79	2-6	535.48	3-6	533.51	4-6	529.30	5-6	531.60				
1-7	543.53	2-7	539.26	3-7	536.92	4-7	532.61	5-7	533.20	6-7	537.97		
1-8	548.58	2-8	543.44	3-8	541.24	4-8	536.72	5-8	536.66	6-8	540.26	7-8	545.24

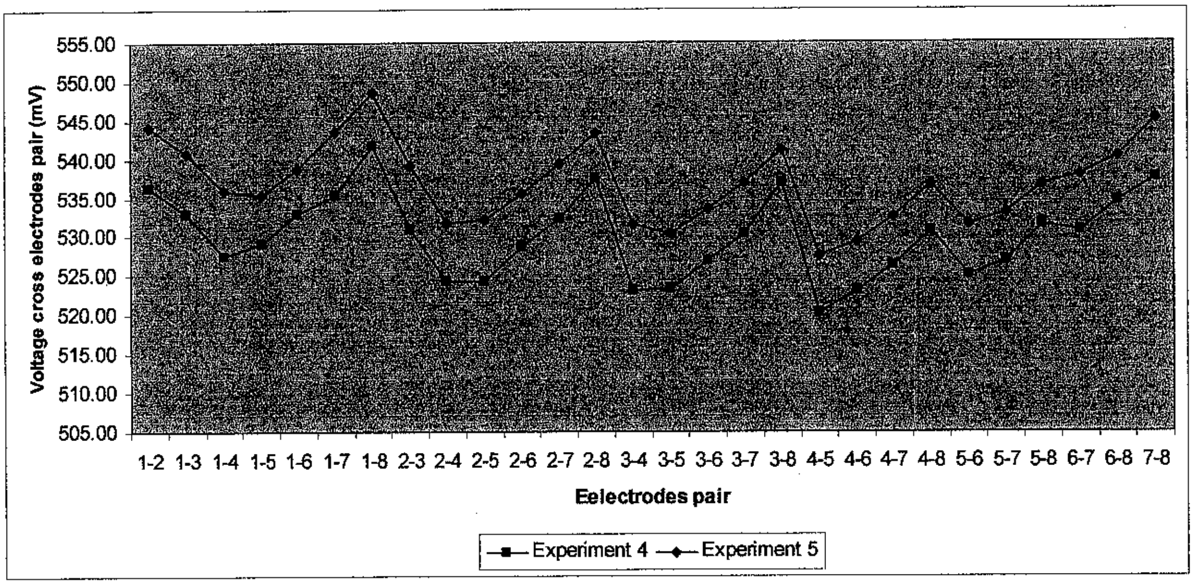
Table 6: Experiment 4 - Voltage measurement with empty vessel. All values in mV.

1-2	536.39												
1-3	532.96	2-3	530.84										
1-4	527.39	2-4	524.19	3-4	523.06								
1-5	529.12	2-5	524.20	3-5	523.28	4-5	520.33						
1-6	532.89	2-6	528.61	3-6	526.85	4-6	522.98	5-6	525.15				
1-7	535.36	2-7	532.23	3-7	530.39	4-7	526.32	5-7	526.98	6-7	530.62		
1-8	541.79	2-8	537.50	3-8	536.87	4-8	530.77	5-8	531.42	6-8	534.54	7-8	537.59

Table 7: Experiment 5 - Voltage measurement with vessel filled with rice grain. All values in mV.



Graph 3: Capacitance values for different condition.



Graph 4: Voltage values for different condition.

In Figure 1, it is noticeable that the capacitance value is the highest between electrodes pair 1-8. This is because due to the arrangement of the cables, referring to **Figure 12**, electrodes 1 and 8 have the longest length comparing to the rest of the cables thus creating a higher capacitance along the cable itself. Any combination pair with electrodes 1 or 8 will exhibit a high value of capacitance. Same goes with the results in **Graph 4**. Voltage level is the highest between pair 1-8.

For Experiments 1, 2 and 3, capacitance values increased when the vessel was filled with rice grain comparing to the time when the vessel was empty. The capacitance increased even more when a bottle of tap water was placed in the middle of the rice grain. This showed that the capacitance measured across the electrodes pairs change with the change of medium inside the vessel. Consequently, so thus the voltage measured across the electrodes pairs. The voltage measurement experiment was done using a single capacitance measuring circuit placed across all pairs one at a time while recording the voltage output using a digital multi-meter. The switching speed of the measuring circuit was set to 1MHz. Frequencies of 2MHz and 4MHz were also used and the results showed similar trend in the graph with slightly higher values. The results are not included in this report.

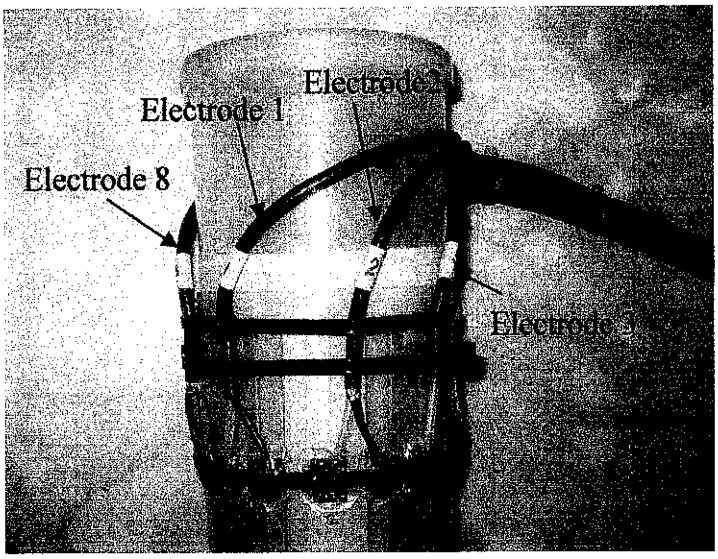


Figure 19: Cables arrangement

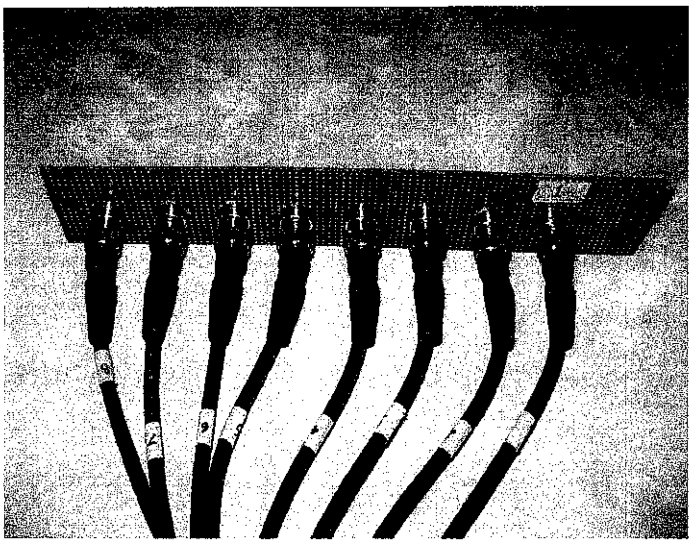
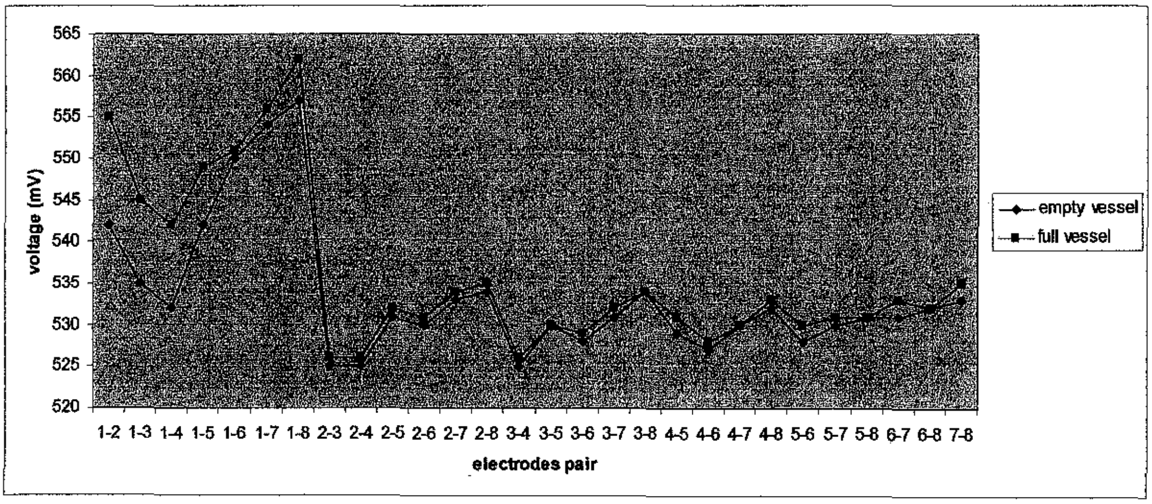


Figure 20: Coaxial cables fitted with BNC connectors on a strip board

Using the microcontroller, the voltage measurement for every electrodes pair combinations (28 combinations) was done again. Electrodes pair change was done by the microcontroller and so was the switching of the CMOS switches but due to the limitation of the microcontroller, the switching speed was only 273KHz and this had greatly reduced the sensitivity of the measuring circuit. **Graph 5** shows the voltage comparison between empty vessel and full vessel. Output voltage is higher when the vessel is filled with the rice grain consistent with the results obtained previously when the CMOS switching and the pair change was done manually.



Graph 5: Output voltage with switching and electrodes pair change done by microcontroller

When the ECT was set up and the output displayed using a digital multi-meter and HyperTerminal, it was noticed that the value shown on the multi-meter and on HyperTerminal screen was slightly different. The multi-meter showed a more accurate and consistent reading as compare to the ones shown by HyperTerminal. Due to its sole purpose and more accurate design, the multi-meter was able to give more reliable information. Referring to PIC16F877 datasheet, serial communication using Universal Synchronous Asynchronous Receiver Transmitter (USART) has a slight error in the output. From the datasheet, using baud rate of 9600 and clock frequency of 10MHz yielded an error of 0.16%.

The voltage reference used for the ADC was obtained from a variable resistor tapped from the 5V supply to get a 1V voltage reference. To adjust the variable resistor in order to get an exact value of 1V is rather difficult. Deviation in voltage reference value also contributed to the reading error shown by HyperTerminal as the ADC was configured using 1V voltage reference.

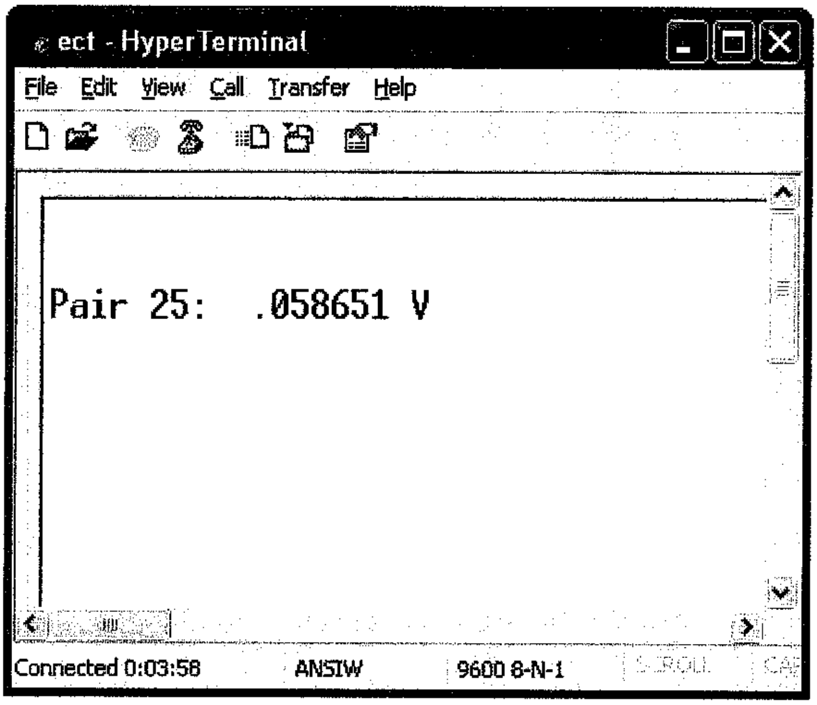


Figure 21: Real time reading displayed using HyperTerminal.

CHAPTER 6

CONCLUSSION & RECOMMENDATION

Feasible data was obtained and tabulated accordingly. Data shows that the ECT system is much more sensitive to changes inside the vessel with a higher frequency switching (above 1MHz). Sensitivity drops when the switching speed is decreased. Even though a high frequency is desired, it is limited by the microcontroller capability when large number of loop is used in the C code to generate the switching requirement. But even so, at a lower frequency of 273KHz, the ECT system was able to indicate changes in the vessel by producing different output voltage when the vessel was full comparing to the time when the vessel was empty. Great care must also be taken in the vessel design as to reduce the effect of noise due to stray capacitance around the vessel and throughout the cabling. Proper grounding method and the use of screening plates around the electrodes improve the immunity of the system to noise.

The length of the coaxial cables contributes to the total capacitance measured by the system. In order to overcome this, separate measuring circuit with adjustable gain should be used for each electrode by first overcoming the coupling effect between all the circuits. Adjustable gain option gives the opportunity to each measuring circuit to compensate for the extra capacitance introduced by the different length of the coaxial cables.

REFERENCES

[1] R A Williams and M S Beck, "Process Tomography-Principles, Techniques and Application", Butterworth-Heinemann, 1995.

[2] Huang, S. M., Stott, A. L., Green, R.G. and Beck, M.S., Electronic transducers for industrial measurement of low value capacitances, 1988.

[3] Quak Foo Lee, Advanced Chemical Technology Centre, "Electrical Resistance Tomography (ERT)", 2nd Jan 2006, http://www.chmltech.com/tomography.htm.

[4] Jun-Wen Liu and Feng Dong, "Electrical Resistance Tomography Based On The Single Drive Electrode Method", IEEE Transactions on Machine Learning and Cybernetics, Vol. 1, pp632-637, Aug. 2004.

APPENDIX A

C code for PIC16F877A microcontroller (switching circuit)

```
#include <16f877a.h>
#use delay(clock=24000000) /* Using a 24 Mhz clock */
#use fast_io(B)
#use fast_io(D)
#use fast_io(C)
#fuses HS,NOWDT,NOPROTECT,NOLVP
main()
 {

   unsigned int i;
   unsigned int j;
   unsigned int k;
   set_tris_b(0x00); //Set port B as output
   set_tris_d(0x00); //Set port D as output
   set_tris_c(0x00); //Set port A as output
   output_b(0x00);
   output_d(0x00);


   while(1)

   {

   output_C(0x01);

   for(i=0;i<15;i++)
   {
    for(j=0;j<255;j++)
    {
```

```
 for(k=0;k<255;k++)
 {
 output_high(PIN_B0);
 output_high(PIN_D1);
 delay_us(1);
 output_low(PIN_B0);
 output_low(PIN_D1);
 delay_us(1);
 }
}
}

output_C(0x02);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
 for(k=0;k<255;k++)
 {
 output_high(PIN_B0);
 output_high(PIN_D2);
 delay_us(1);
 output_low(PIN_B0);
 output_low(PIN_D2);
 delay_us(1);
 }
 }
}

output_C(0x03);

for(i=0;i<15;i++)
{
```

```
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B0);
  output_high(PIN_D3);
  delay_us(1);
  output_low(PIN_B0);
  output_low(PIN_D3);
  delay_us(1);
  }
 }
}

output_C(0x04);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B0);
  output_high(PIN_D4);
  delay_us(1);
  output_low(PIN_B0);
  output_low(PIN_D4);
  delay_us(1);
  }
 }
}

output_C(0x05);
```

```
for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B0);
  output_high(PIN_D5);
  delay_us(1);
  output_low(PIN_B0);
  output_low(PIN_D5);
  delay_us(1);
  }
 }
}

output_C(0x06);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B0);
  output_high(PIN_D6);
  delay_us(1);
  output_low(PIN_B0);
  output_low(PIN_D6);
  delay_us(1);
  }
 }
}
```

```
output_C(0x07);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B0);
  output_high(PIN_D7);
  delay_us(1);
  output_low(PIN_B0);
  output_low(PIN_D7);
  delay_us(1);
  }
 }
}

output_C(0x08);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D2);
  delay_us(1);
  output_low(PIN_B1);
  output_low(PIN_D2);
  delay_us(1);
  }
 }
```

```
}

output_C(0x09);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D3);
  delay_us(1);
  output_low(PIN_B1);
  output_low(PIN_D3);
  delay_us(1);
  }
 }
}

output_C(0x0A);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D4);
  delay_us(1);
  output_low(PIN_B1);
  output_low(PIN_D4);
  delay_us(1);
```

```
  }
 }
}

output_C(0x0B);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D5);
  delay_us(1);
  output_low(PIN_B1);
  output_low(PIN_D5);
  delay_us(1);
  }
 }
}

output_C(0x0C);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D6);
  delay_us(1);
  output_low(PIN_B1);
```

```
  output_low(PIN_D6);
  delay_us(1);
  }
 }
}

output_C(0x0D);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B1);
  output_high(PIN_D7);
  delay_us(1);
  output_low(PIN_B1);
  output_low(PIN_D7);
  delay_us(1);
  }
 }
}

output_C(0x0E);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B2);
  output_high(PIN_D3);
```

```
  delay_us(1);
  output_low(PIN_B2);
  output_low(PIN_D3);
  delay_us(1);
  }
 }
}

output_C(0x0F);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B2);
  output_high(PIN_D4);
  delay_us(1);
  output_low(PIN_B2);
  output_low(PIN_D4);
  delay_us(1);
  }
 }
}


output_C(0x10);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
```

```
 {
 output_high(PIN_B2);
 output_high(PIN_D5);
 delay_us(1);
 output_low(PIN_B2);
 output_low(PIN_D5);
 delay_us(1);
 }
 }
}


output_C(0x11);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
 for(k=0;k<255;k++)
 {
 output_high(PIN_B2);
 output_high(PIN_D6);
 delay_us(1);
 output_low(PIN_B2);
 output_low(PIN_D6);
 delay_us(1);
 }
 }
}


output_C(0x12);

for(i=0;i<15;i++)
```

```
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B2);
  output_high(PIN_D7);
  delay_us(1);
  output_low(PIN_B2);
  output_low(PIN_D7);
  delay_us(1);
  }
 }
}


output_C(0x13);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B3);
  output_high(PIN_D4);
  delay_us(1);
  output_low(PIN_B3);
  output_low(PIN_D4);
  delay_us(1);
  }
 }
}
```

```
output_C(0x14);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B3);
  output_high(PIN_D5);
  delay_us(1);
  output_low(PIN_B3);
  output_low(PIN_D5);
  delay_us(1);
  }
 }
}


output_C(0x15);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B3);
  output_high(PIN_D6);
  delay_us(1);
  output_low(PIN_B3);
  output_low(PIN_D6);
  delay_us(1);
```

```
   }
  }
 }


 output_C(0x16);

 for(i=0;i<15;i++)
 {
  for(j=0;j<255;j++)
  {
   for(k=0;k<255;k++)
   {
   output_high(PIN_B3);
   output_high(PIN_D7);
   delay_us(1);
   output_low(PIN_B3);
   output_low(PIN_D7);
   delay_us(1);
   }
  }
 }


output_C(0x17);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
  for(k=0;k<255;k++)
  {
  output_high(PIN_B4);
  output_high(PIN_D5);
```

```
 delay_us(1);
 output_low(PIN_B4);
 output_low(PIN_D5);
 delay_us(1);
 }
}
}


output_C(0x18);

for(i=0;i<15;i++)
{
for(j=0;j<255;j++)
{
 for(k=0;k<255;k++)
 {
 output_high(PIN_B4);
 output_high(PIN_D6);
 delay_us(1);
 output_low(PIN_B4);
 output_low(PIN_D6);
 delay_us(1);
 }
}
}


output_C(0x19);

for(i=0;i<15;i++)
{
for(j=0;j<255;j++)
{
```

```
 for(k=0;k<255;k++)
 {
 output_high(PIN_B4);
 output_high(PIN_D7);
 delay_us(1);
 output_low(PIN_B4);
 output_low(PIN_D7);
 delay_us(1);
 }
}
}


output_C(0x1A);

for(i=0;i<15;i++)
{
 for(j=0;j<255;j++)
 {
 for(k=0;k<255;k++)
 {
 output_high(PIN_B5);
 output_high(PIN_D6);
 delay_us(1);
 output_low(PIN_B5);
 output_low(PIN_D6);
 delay_us(1);
 }
 }
}
```

```
    output_C(0x1B);

    for(i=0;i<15;i++)
    {
     for(j=0;j<255;j++)
     {
      for(k=0;k<255;k++)
      {
      output_high(PIN_B5);
      output_high(PIN_D7);
      delay_us(1);
      output_low(PIN_B5);
      output_low(PIN_D7);
      delay_us(1);
      }
     }
    }


    output_C(0x1C);

    for(i=0;i<15;i++)
    {
     for(j=0;j<255;j++)
     {
      for(k=0;k<255;k++)
      {
      output_high(PIN_B6);
      output_high(PIN_D7);
      delay_us(1);
      output_low(PIN_B6);
      output_low(PIN_D7);
      delay_us(1);
      }
     }
    }



     }
}
```

APPENDIX B

C code for PIC18F452 (analog to digital conversion)

```
#include <18F452.h>
#device ADC=10
#include <stdio.h>
#include<math.h>
#fuses HS,NOWDT,NOPROTECT, NOPUT, NOBROWNOUT, NOLVP
#use delay(clock = 10000000)
#use rs232(baud=9600, parity=N, xmit=PIN_C6, rcv=PIN_C7, bits=8)
#include <string.h>


//Declare variable

unsigned int16 adcValue;
float voltage;
void main()
{

    set_tris_b(0xFF); //Set port B as input

    //adc
    setup_adc_ports(ANALOG_RA3_REF);
    setup_adc(ADC_CLOCK_INTERNAL);       // Use internal ADC
clock.
    set_adc_channel(0);
```

```
while(1)

{
 printf("\033[2J");                //clear hyperterminal screen

 while (input_b()==0x01)

 {
 delay_us(50);           // Delay for sampling cap to charge
 adcValue = read_adc();    // Get ADC reading
 voltage = 1.000 * adcValue / 1023.000;

 printf("\033[2J");              //clear hyperterminal screen
 printf("Pair 12: %f V\n",voltage);
 delay_ms(574);
 }

 while (input_b()==0x02)
 {
 delay_us(50);           // Delay for sampling cap to charge
 adcValue = read_adc();    // Get ADC reading
 voltage = 1.000 * adcValue / 1023.000;

 printf("\033[2J");              //clear hyperterminal screen
 printf("Pair 13: %f V\n",voltage);
 delay_ms(574);
 }

 while (input_b()==0x03)
```

```
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();  // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 14: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x04)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();  // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 15: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x05)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();  // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 16: %f V\n",voltage);
```

```
delay_ms(574);
}

while (input_b()==0x06)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 17: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x07)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 18: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x08)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
```

```
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 23: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x09)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 24: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x0A)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");            //clear hyperterminal screen
printf("Pair 25: %f V\n",voltage);
delay_ms(574);
}
```

```
while (input_b()==0x0B)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 26: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x0C)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 27: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x0D)
{

delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;
```

```
printf("\033[2J");              //clear hyperterminal screen
printf("Pair 28: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x0E)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 34: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x0F)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 35: %f V\n",voltage);
delay_ms(574);
}


while (input_b()==0x10)
```

```
{
delay_us(50);          // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");          //clear hyperterminal screen
printf("Pair 36: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x11)
{
delay_us(50);          // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");          //clear hyperterminal screen
printf("Pair 37: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x12)
{
delay_us(50);          // Delay for sampling cap to charge
adcValue = read_adc();   // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");          //clear hyperterminal screen
printf("Pair 38: %f V\n",voltage);
```

```
delay_ms(574);
}

while (input_b()==0x13)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 45: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x14)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 46: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x15)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
```

```
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 47: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x16)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 48: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x17)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 56: %f V\n",voltage);
delay_ms(574);
}
```

```
while (input_b()==0x18)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 57: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x19)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
printf("Pair 58: %f V\n",voltage);
delay_ms(574);
}

while (input_b()==0x1A)
{
delay_us(50);           // Delay for sampling cap to charge
adcValue = read_adc();    // Get ADC reading
voltage = 1.000 * adcValue / 1023.000;

printf("\033[2J");              //clear hyperterminal screen
```

```
    printf("Pair 67: %f V\n",voltage);
    delay_ms(574);
    }

    while (input_b()==0x1B)
    {
    delay_us(50);            // Delay for sampling cap to charge
    adcValue = read_adc();   // Get ADC reading
    voltage = 1.000 * adcValue / 1023.000;

    printf("\033[2J");              //clear hyperterminal screen
    printf("Pair 68: %f V\n",voltage);
    delay_ms(574);
    }

    while (input_b()==0x1C)
    {
    delay_us(50);            // Delay for sampling cap to charge
    adcValue = read_adc();   // Get ADC reading
    voltage = 1.000 * adcValue / 1023.000;

    printf("\033[2J");              //clear hyperterminal screen
    printf("Pair 78: %f V\n",voltage);
    delay_ms(574);
    }

  }
}
```

APPENDIX C

(Schematic Diagram of Complete ECT circuit)