สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ROOM TEMPERATURE CONTROL USING AIR CONDITIONER



JAMRUS RUTTANAYUKOL

King Mongkut's Institute of Technology Ladkrabang

เลขหมู่.....
เลขทะเบียน.....76416
วัน,เดือน,ปี.....25 ส.ค. 2557

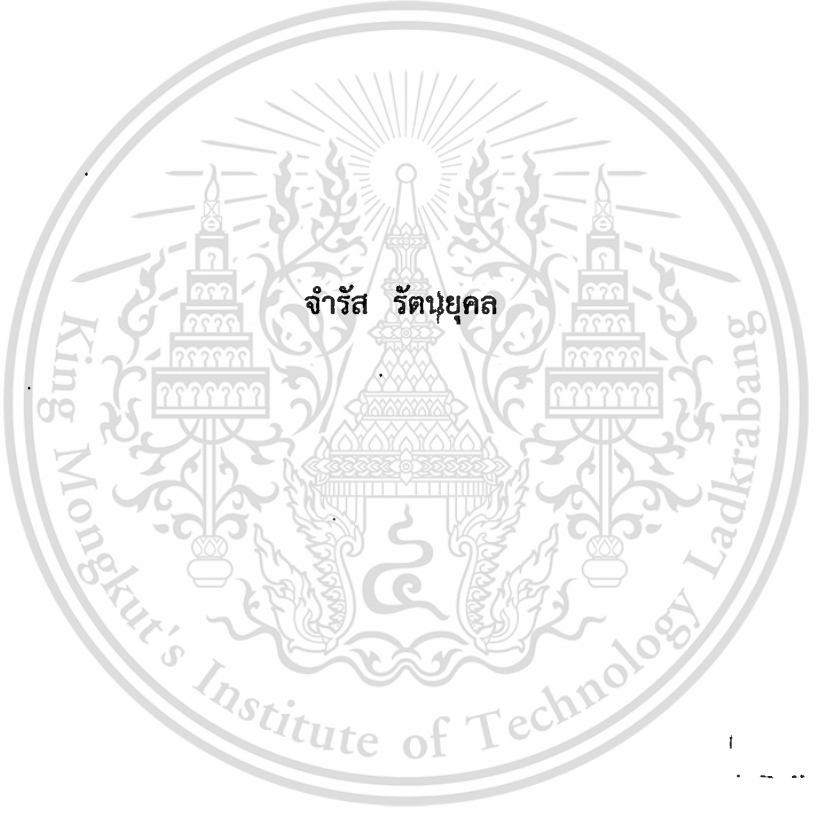
.b.....
.i.....

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT .
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2011

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การควบคุมอุณหภูมิในห้องด้วยเครื่องปรับอากาศ
ROOM TEMPERATURE CONTROL USING AIR CONDITIONER

จำรัส รัตนุยุคล



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2554

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การควบคุมอุณหภูมิในห้องด้วยเครื่องปรับอากาศ

โดย
นายจำรัส รัตนยุคล 51010153

อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร.ถาวร เบญจนราสุทธิ์

ปีการศึกษา 2554

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการออกแบบระบบควบคุมอุณหภูมิห้องอยู่อาศัยด้วยเครื่องปรับอากาศที่มีอยู่แล้ว โดยโครงสร้างของระบบประกอบด้วย ห้องที่ใช้อยู่อาศัย เครื่องปรับอากาศ และวงจรควบคุม จุดมุ่งหมายของโครงงานนี้คือการสร้างวงจรเพื่อควบคุมเครื่องปรับอากาศผ่านการติดต่อแบบไร้สาย ให้สามารถรักษาอุณหภูมิภายในห้องให้อยู่ในย่านที่ต้องการโดยใช้ตัวควบคุมแบบสัดส่วน ปริพันธ์ อนุพันธ์ที่เหมาะสมและมีตัวประมาณการอุณหภูมิสุดท้ายเพื่อประมาณเวลาปิดล่วงหน้า การออกแบบตัวควบคุมจะอาศัยแบบจำลองทางคณิตศาสตร์ที่ได้จากการทดลองเพื่อใช้ในการปรับแต่งค่าสัมประสิทธิ์ของตัวควบคุมให้เหมาะสมโดยใช้กฎการปรับแต่งของ Skogestad จากผลการเปรียบเทียบการทำงานของตัวควบคุมที่ได้นำเสนอกับตัวควบคุมแบบเปิดปิดพบว่าตัวควบคุมที่นำเสนอสามารถรักษาอุณหภูมิภายในห้องอยู่ในย่านที่ต้องการและประหยัดพลังงานลงได้มาก

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ROOM TEMPERATURE CONTROL USING AIR CONDITIONER

By
Jamrus Ruttanayukol

Advisor
Assoc. Prof. Dr.Taworn Benjanarasuth

Academic Year 2011

ABSTRACT

This thesis presents theories and implementation procedures of room temperature control using a pre-installed air conditioner as the actuator. The system composes of the room, an air conditioner, and a controller circuit. The goal is to build a circuit to control the air conditioner through wireless communication to regulate room temperature in acceptable range of desired value using an appropriated proportional-integral-derivative (PID) controller bundled with a final-value estimator to estimate earlier turn-off time. The PID controller design is based on an identified mathematical model and Skogestad's tuning rules. The experimental results show that the implemented system can regulate the room temperature as desired. Comparison from the experiments between the command inputs of the proposed control system and on-off control system also shows that the proposed controller can save considerably amount of energy.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ACKNOWLEDGEMENTS

My study and this thesis have been benefited greatly from the support of many people, some of whom I would sincerely like to thank here.

To begin with, I would like to express my sincere gratitude to my advisor Assoc. Prof. Taworn Benjanarasuth for his continuous support of this project, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not imagine having better advisor and mentor for my bachelor project research.

Beside my advisor, I would like to thank the rest of my thesis committee: Ajarn Don Issarakorn, and Ajarn Songmoung Nundrakwang, for their encouragement, insightful comments, and hard questions. I also would like to thank them as former advisors for their continuous supports and guidance during that time.

My sincere thanks also go to Ajarn Sungwan Boksuwan, for offering me the opportunities in his research group and teach me numerous valuable things. He also gave me an opportunity to conduct a paper and exhibit it oversea that leading me to obtain many distinct experiences as a bachelor student. This project is initially inspired by him during the 3rd year of my study; he also is my project advisor in the first semester.

I am deeply grateful to Asst. Prof. Porames Pranayanan, Asst. Prof. Sumitr Panaudomsab, and Asst. Prof. Nonthawat Juldecha for their motivation, vast knowledge, and opportunities that lead me experienced particular situations.

I wish to thank all other teachers and staffs at the Department of Control engineering for knowledge and supports given to me during these four years of my bachelor study.

I would also like to thank my seniors and master student for their support, not hesitate to give good advices, and their knowledge. Also I would like to thank Mr. Surawat Boksuwan for his excellent support and advices.

I thank my fellow Hazard members: Mr. Suppakit Panusumpol, Mr. Rittiporn Phukieaw, and Mr. Poramin Duangchana for offering me to join their groups during my 4th year of my bachelor study. Also I thank Mr. Charnwit Wongsuppalert for his help in time of my need. Also I thank all members of Ladkrabang Muslim Student Club for their moral lessons, fun activities, motivations, and supports during my course.

Finally, but first in my heart, my parents and the rest of my family are due to my deep gratitude for their continued moral and financial support throughout my studies, the former being of much greater importance. The broad education that I was able to enjoy while growing up has proven invaluable. I promise I will not disappoint you all.

Jamrus Ruttanayukol

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

CONTENTS

Page

Thai Abstract I
English Abstract II
Acknowledgement III
Contents IV
List of Tables VII
List of Figures VIII

Chapter 1 Introduction 1
1.1 Problem Statement and Its Significance 1
1.2 Goals and Objectives 1
1.3 Scope of Study 1
1.4 Procedure of the Study 2
1.5 Thesis Details 2

Chapter 2 Theories and Principles 3
2.1 Room Thermal Mathematical Models 3
2.1.1 Newton's Law of Cooling and Heating 3
2.1.2 Linear Least Square 4
2.1.3 Deviations 5
2.2 On-off Controller 6
2.3 PID Controllers 7
2.3.1 PID Control Structure 7
2.3.2 Skogestad's PID Tuning Method 10

Chapter 3 Design Procedures 12
3.1 Overall System Structure 12
3.2 Air Conditioning System 13
3.2.1 Components 13
3.2.2 Power Consumption 14
3.2.3 Input Definition 14
3.3 Wireless Remote Controller 15
3.3.1 Receiver Circuit 15
3.3.2 Transmitter Circuit 16
3.4 Microchip dsPIC30F 17
3.4.1 Timers 17
3.4.2 Output Compare 18
3.4.3 Analog to Digital Converter 18
3.4.4 UART 19
3.5 Temperature Measurement 19
3.5.1 Temperature Sensor TC1047A 19

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

CONTENTS (Cont.)

Page

3.5.2 Temperature Measuring Program....20
3.6 Controller Circuit....20
3.7 Remote Signal Imitation Program....23
3.8 On-off Control Program....25
3.9 PID Controller....26
3.9.1 PID Tuning....26
3.9.2 PID Control Program....28
3.10 Prediction Program....28
3.11 Overall Microcontroller Program....30

Chapter 4 Experimental Results....31
4.1 Air Conditioner Remote Signal Decoding and Imitation....31
4.1.1 Signal Decoding....31
4.1.2 Signal Imitation....35
4.2 Model Identification Process....37
4.2.1 Model Validation....37
4.2.2 Parameters Identification....42
4.3 Control Results....52
4.3.1 On-off Control Results....52
4.3.2 PID Control Results....54
4.4 PI Control with Prediction Results....57
4.5 Power Consumption Analysis....62

Chapter 5 Conclusions and Future Works....63
5.1 Conclusions....63
5.2 Future Works....63

Appendix A Device Datasheets....64
A.1 Microchip dsPIC30f6014A....65
A.2 Microchip TC1047A....67
A.3 Vishay v4838....68
A.4 Vishay TSAL7400....69
A.5 National LM2940....70
A.6 Maxim MAX3232CSE....71

Appendix B Experimental Program Code....72
B.1 Header File....73
B.2 Main Program File....73
B.3 Initializing Program File....74

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

CONTENTS (Cont.)

Page

B.4 Temperature Measuring Program File 76
B.5 Remote Signal Imitation Program File 76
B.6 Control Program File 78
B.7 Interrupt Program File 80

Bibliography 81

King Mongkut's Institute of Technology Ladkrabang

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

LIST OF TABLES

Table Page

2.1 Skogestad's formulas for PID tuning....11
3.1 Carrier's 38RC060SC specification....14
4.1 Remote data represented in examples....35
4.2 Identified parameters from experimentations and its deviations....48
4.3 Average model temperature output deviations from the experiments....52
4.4 Comparison of compressor usage....62

King Mongkut's Institute of Technology Ladkrabang

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

LIST OF FIGURES

Figure Page

2.1 Unity feedback control system....6
2.2 Laplace transform of Figure 2.1....7
2.3 Exponential response curve....11
3.1 Overall system structures....12
3.2 Photocell detection circuit....15
3.3 Vishay's v4838 block diagram....16
3.4 IR emitter circuit....16
3.5 Temperature converting program's flowchart....21
3.6 Control circuit schematics: processor part....22
3.7 Control circuit schematics: interface part....22
3.8 Control circuit schematics: power part....23
3.9 The control circuit board....23
3.10 Remote signal imitation program's flowchart....24
3.11 Data comparison program's flowchart....25
3.12 On-off control program's flowchart....25
3.13 PID control program's flowchart....28
3.14 Prediction program's flowchart....29
3.15 Overall microcontroller program's flowchart....30
4.1 IR signal period measurement....32
4.2 IR remote controller data signal (1)....32
4.3 IR remote controller data signal (2)....33
4.4 IR remote controller data signal (3)....33
4.5 IR remote controller data signal (4)....34
4.6 IR remote controller data signal (5)....34
4.7 Microcontroller generated signal for turning compressor on....36
4.8 Microcontroller generated signal for turning compressor off....37
4.9 Room temperature using air conditioner for five hours and a half....38
4.10 Temperature curve after turned the air conditioner off in the first experiment....38
4.11 Temperature curve after shutting down the air conditioner in the second experiment....39
4.12 Temperature curve after shutting down the air conditioner in the third experiment....39
4.13 Actual results from Figure 4.9 compared with its fitted curve with $\alpha = 1.7045e-4$....40
4.14 Actual results from Figure 4.10 compared with its fitted curve with $\alpha = 8.4520e-4$....41
4.15 Actual results from fig. 4.11 compared with its fitted curve with $\alpha = 0.0036$....41
4.16 Actual results from fig. 4.12 compared with its fitted curve with $\alpha = 0.0020$....42
4.17 Randomized input for the first parameters identification experiment....43
4.18 Temperature output of command signal in Figure 4.17....43
4.19 Randomized input for the second parameters identification experiment....44
4.20 Temperature output of command signal in Figure 4.19....44

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

LIST OF FIGURES (Cont.)

Figure Page

4.21 Randomized input for the third parameters identification experiment 45
4.22 Temperature output of command signal in Figure 4.21 45
4.23 Randomized input for the fourth parameters identification experiment 46
4.24 Temperature output of command signal in Figure 4.23 46
4.25 Randomized input for the fifth parameters identification experiment 47
4.26 Temperature output of command signal in Figure 4.25 47
4.27 Randomized input for the sixth parameters identification experiment 48
4.28 Temperature output of command signal in Figure 4.27 48
4.29 Temperature outputs comparison from the first experiment 49
4.30 Temperature outputs comparison from the second experiment 49
4.31 Temperature outputs comparison from the third experiment 50
4.32 Temperature outputs comparison from the fourth experiment 50
4.33 Temperature outputs comparison from the fifth experiment 51
4.34 Temperature outputs comparison from the sixth experiment 51
4.35 Command input driven by on-off controller from the first experiment 53
4.36 Temperature output by on-off controller from the first experiment 53
4.37 Command input driven by on-off controller from the second experiment 54
4.38 Temperature output by on-off controller from the second experiment 54
4.39 Input signal driven by PI controller of the first experiment 55
4.40 Command input driven by PI controller of the first experiment 55
4.41 Temperature output of PI control system of the first experiment 56
4.42 Input signal driven by PI controller of the second experiment 56
4.43 Command input driven by PI controller of the second experiment 57
4.44 Temperature output of PI control system of the second experiment 57
4.45 Input's duty cycle of the first experiment 58
4.46 Command input to the compressor of the first experiment 58
4.47 Outputs of PI control system with predictor of the first experiment 59
4.48 Input's duty cycle of the second experiment 59
4.49 Command input to the compressor of the second experiment 60
4.50 Outputs of PI control system with predictor of the second experiment 60
4.51 Input's duty cycle of the third experiment 61
4.52 Command input to the compressor of the third experiment 61
4.53 Outputs of PI control system with predictor of the third experiment 62

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 1

Introduction

1.1 Problem Statements and Its Significance

In these modern days, we use many facilities to improve our daily life. One thing that has been used in buildings and industrials is the heating, ventilation, and air conditioning (HVAC) system. In US, buildings account for 40% of the electricity consumption and 33% of them is taken into HVAC usage [1].

As a control engineer, this can be seen as a challenge to reduce these energy usages by applying control theories. Many homes use single stage heat pump that is usually implemented for two positions control. They should be beneficial if we could change the control scheme into a smarter one but still using the old equipment. This method could also be applied to other air conditioner systems with appropriate identification procedure.

This project aims to show a method to reduce energy consumption of the HVAC system in the example room while doing the best to maintain good outcome. The system should achieve the goal without modifying the installed air conditioner while using an easy way to install the control device.

This work was initially inspired by Berkeley's BRITE platform [2]. Their result in making such system to implement control theories which lead into a possibility of reducing huge amount of energy was impressive. This project will imitate BRITE platform in a less complex way in term of hardware, software, and controller.

1.2 Goals and Objectives

1. Study room temperature system affected by air conditioner.
2. Study for dsPIC30F programming applications.
3. Study for designing a PID controller.
4. Design a system to manipulate an installed air conditioner.

1.3 Scope of the Study

1. Study for a mathematical model
2. Build the system to manipulate the air conditioner.
3. Study MATLAB implementation in identification and curve fitting process.
4. Study and design PID controller using Skogestad's method.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.4 Procedure in the Study

1. Study for a mathematical model to use in the project.
2. Build a sensing system and do experiments to match the model.
3. Build the system to do both manipulating the air conditioner and sensing the temperature using dsPIC30F as the main processor.
4. Design programs used in the system.
5. Run open-loop tests for model identification in MATLAB.
6. Run simulations from those models and compare to the actual results.
7. Design a PID controller using MATLAB.
8. Implement the controller.
9. Design a prediction program.
10. Implement the actual system.
11. Summarize the project.

1.5 Thesis Details

This thesis is organized into five chapters and two appendices. Its outline is as follows.

Chapter 1 introduces foundation information of the problem and overall concept of the project. Chapter 2 introduces the reader theories involved in this work. Chapter 3 shows the overall structure of the system and presents their details, reasons, and procedure part by part. Chapter 4 unveils implementation and experimental parts of the project along with their respective results and their effect to the design. Lastly, Chapter 5 presents the conclusion of this project.

There are two sections of appendix. The first one presents important pages of datasheet for devices used in the project, and the other section presents experimental code for microcontroller in C language.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 2

Theories and Principles

2.1 Room Thermal Mathematical Model

To perform control system analysis and design, a mathematical model of the plant is needed for conventional ways. PID tuning and its description concepts use knowledge in mathematical models and their frequency responses. A structure of difference equation of the system must be specified in identification techniques for discrete-time system and prediction program design.

2.1.1 Newton's Law of Cooling and Heating

After thorough studies of acquired data from many experiments, it is shown that Newton's law of cooling and heating is the most appropriate for the task to explain the phenomena. It also needs only a simple calculation while keeping the accuracy in the proper range can be easily achieved by this model. This model was used to explain the phenomena of the similar system to the level of 0.1 C RMS errors [2], but more complex calculation and ambient temperature measurement were needed.

Newton's Law of Cooling states that the rate of change of the temperature of an object is proportional to the difference between its own temperature and the ambient temperature. From this law, we can write out the differential equation as

$$\frac{dT(t)}{dt} = \alpha(T(t) - T_a(t)) \tag{2.1}$$

where $T(t)$ is the room's temperature, $T_a(t)$ is the ambient temperature, and α is a constant parameter.

In this project, we have some other heat source beside the ambient environment. The air conditioner acts as the actuator to cool the room as instructed by the controller which will have a parameter attached to its manipulating input signal.

$$\frac{dT(t)}{dt} = \alpha(T(t) - T_a(t)) + \beta U_c(t) \tag{2.2}$$

where $U_c(t)$ is the input, and β is a constant parameter.

In this project, the main subject of interest is a room, the relationship between its temperature and its ambient temperature is through a complex structure of the wall and there are dynamics inside the room itself. Thus another equation is composed by isolating constant parameter for the ambient temperature from the room temperature as

$$\frac{dT(t)}{dt} = \alpha T(t) + \beta U_c(t) + \chi T_a(t) \tag{2.3}$$

where χ is a constant parameter.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Furthermore, although not all the time, the room will be occasionally occupied with the lab participants whose will generate heat along with equipment that is being used. This uncertainties function is added into the equation which leads to

$$\frac{dT(t)}{dt} = \alpha T(t) + \beta U_c(t) + \chi T_a(t) + q(t) \tag{2.4}$$

while this $q(t)$ represents uncertainty noises to the system.

However for the identification purpose, this model should be transformed into a discrete-time model. With forward Euler method, $dT(t)$ can be transformed into $T[n+1]-T[n]$ and dt can be referred to the sampling period. The result is the following equation

$$\frac{T[n+1]-T[n]}{T_s} = \alpha T[n] + \beta U_c[n] + \chi T_a[n] + q[n]. \tag{2.5}$$

Rearranging equation (2.5) yields

$$T[n+1] = (\alpha T_s + 1)T[n] + \beta T_s U_c[n] + \chi T_s T_a[n] + T_s q[n]. \tag{2.6}$$

Equation (2.6) can be rearranged by lumping constant parameters into the new ones as

$$\alpha_n = \alpha T_s + 1,$$
$$\beta_n = \beta T_s,$$
$$\chi_n = \chi T_s.$$

While $q[n]$ is itself the uncertainty function, multiplying T_s does not change its property. Thus, $T_s q[n]$ is still denoted as $q[n]$. The final discrete-time model is

$$T[n+1] = \alpha_n T[n] + \beta_n U_c[n] + \chi_n T_a[n] + q[n]. \tag{2.7}$$

2.1.2 Linear Least Square

After a certain model is confirmed to use in the project, identification method is the next thing to be considered. There are many identification methods that could be used to find these parameters. However, the easiest for this application would be regression methods. Regression methods can be used to analyze data from unplanned experiments and allow some faults in the experiments [3].

Equation (2.7) can be considered as a linear regression model by fusing $q[n]$ and $\chi_n T_a[n]$ then converting it into a single constant value:

$$y = \beta_0 + \beta_1 x_1 + \beta x_2 + \varepsilon. \tag{2.8}$$

With n set of data, equation (2.8) can be written in a matrix notation as

$$y = X\beta + \varepsilon \tag{2.9}$$

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

where $y=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X=\begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix}, \beta=\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}$, and $\varepsilon=\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$.

For a linear regression model, the least square estimator for the problem can be used to find a single estimated of each parameter. The least square estimator is an estimator that aim to minimize $\sum \varepsilon_i^2 = L$ and could be expressed as in equations (2.10).

$$\begin{aligned} \sum \varepsilon_i^2 &= \varepsilon'\varepsilon = (y - X\beta)'(y - X\beta), \\ \sum \varepsilon_i^2 &= y'y - \beta'X'y - y'X\beta + \beta'X'X\beta, \\ \sum \varepsilon_i^2 &= y'y - 2\beta'X'y + \beta'X'X\beta. \end{aligned} \tag{2.10}$$

The least square estimators must satisfy equation (2.11).

$$\frac{\delta L}{\delta \beta}|_{\hat{\beta}} = -2X'y + 2X'X\hat{\beta} = 0. \tag{2.11}$$

Equation (2.12) is the solution to equation (2.11).

$$X'X\hat{\beta} = X'y. \tag{2.12}$$

Equation (2.12) can be solved by multiplying $(X'X)^{-1}$ to both sides. Thus, the least squares estimator of β, $\hat{\beta}$ can be computed using equation (2.13).

$$\hat{\beta} = (X'X)^{-1}X'y \tag{2.13}$$

where $'$ denotes the transpose operator, and $^{-1}$ is the inverse operator.

2.1.3 Deviations

In the process of identification of parameters or curve fitting, measurements of accuracy are needed during analysis stage. Those widely and frequently used by experimenters are mean absolute deviation, mean squared deviation, and root-mean-squared deviation. They are used to compare differences between two set of values that may vary, neither of which is accepted as the standard.

Equations in this section will use the same set of notations where $\hat{y}$ is the estimated value, y is the actual value, and n is the number of samples used to measure the deviation value.

Mean absolute deviation (*MAD*) is described as equation (2.14).

$$MAD = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}. \tag{2.14}$$

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

This method measures the average distance of the data set from its mean during certain number of time periods.

Mean squared deviation (*MSD*) is described as equation (2.15).

$$MSD = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}. \tag{2.15}$$

This method measures the average of the squares of the errors. This method has more complexity in computation as the number of multiplication is increased along with the number of sample. The forecast accuracy of *MSD* method is higher than *MAD* method, as *MAD* does not require squaring while *MSD* does.

Root mean squared deviation (*RMSD*) is described as equation (2.16).

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}. \tag{2.16}$$

This method is another form of normal deviation with good forecast accuracy. It is frequently used to measure the difference between values predicted by a model or an estimator and the true values.

2.7 On-off Controller

On-off control, or two-position control, or bang-bang control, has been living in control field for ages as the easiest closed-loop controller which could automatically overcome many obstacles to achieve the goal under defined hysteresis loop. For the conventional air conditioner with a single stage heat pump, on-off control scheme is commonly used because the system itself has 2 states of working operation.

On-off controller is a feedback controller that controls in binary states, with rules for set point and hysteresis. The relation between the working state and the output value depends on the system it is working on. It is ought to be in the state in which will reduce the output value when it is above the set point and out of hysteresis zone, and it will switch to another state when the output value is below the set point and out of hysteresis zone.

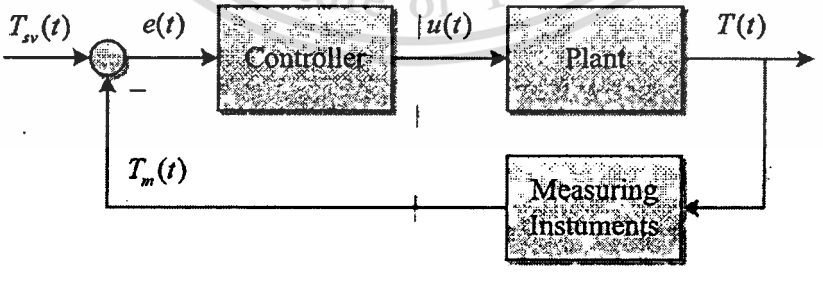


Figure 2.1 Unity feedback control system

Although the machine acts almost immediately after the command signal, its effect to the output will take some time to change the other state. Thus, the resulting output will not always be kept in the hysteresis zone. However, it will continue for a

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

period of time and its effect might stay longer. Thus, the output will always be getting out of hysteresis zone, and the designer will have to design this hysteresis zone to get output zone into acceptable value.

Figure 2.1 illustrates a close-loop control system. The controller usually has its main task to produce appropriate input signal to the plant. On-off controller used in this project can be expressed as equation (2.17).

$$\begin{aligned} &e(t) \geq 0.5 \rightarrow u(t) = 1, \\ &-0.5 \leq e(t) \leq 0.5 \rightarrow u(t) \text{ unchanged} \\ &e(t) \leq -0.5 \rightarrow u(t) = 0. \end{aligned} \tag{2.17}$$

For the first attempt to test control ability, the set point is 23C. The other test for command comparison has the set point at 25C. This controller is designed with hysteresis of ±0.5C, but the controller will not be able to keep the temperature in this range. However, the resulting temperature has its average value at the set point value and it is good enough to satisfy users.

2.8 PID Controllers

Proportional, integral, and derivative (PID) control contains three terms. They are proportional control, integral control, and derivative control. They are equations with parameters using the output value to compute the input signal at the sampled time.

PID control is widely used in the process industries due to its simplicity, robustness and wide range of applicability in the regulatory control layer. Desborough and Miller [4] reported that more than 97 percent of the regulatory controllers utilize the PID algorithm. A survey from Japan [5] showed that the ratio of applications of PID control, conventional advanced control (feed forward, override, valve position control, gain-scheduled PID, etc.) and model predictive control was about 100:10:1.

2.8.1 PID Control Structure

A PID controller can be adjusted to be many forms to suit its work, but every one of them can be recognized by Figure 2.1 as the controller of the control system. Laplace transform is widely used in control analysis and design for its property to convert differential equations into algebraic equations with a complex variable s. Its differences from phasors are that this method can apply to general signals and can handle transient conditions. Representation of Figure 2.1 in s-domain is shown in Figure 2.2.

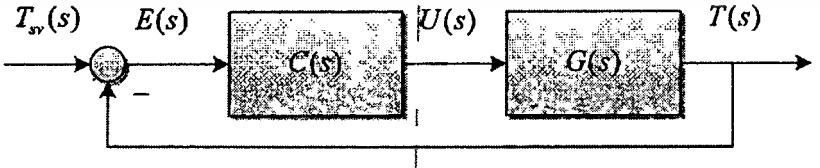


Figure 2.2 Laplace transform of Figure 2.1.

Controller's transfer function is denoted as $C(s)$, the relationship between error value $E(s)$ and the input value $U(s)$ can be presented as equation (2.18).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$C(s)=\frac{U(s)}{E(s)}. \tag{2.18}$$

The air conditioner's transfer function is denoted as $G(s)$, the relationship between input value $U(s)$ and temperature output $T(s)$ can be presented as equation (2.19).

$$G(s)=\frac{T(s)}{U(s)}. \tag{2.19}$$

As suggested in its name, PID consists of proportional, integral, and derivative statement. Their differential equations and its respective Laplace transformed equations can be expressed as follows.

Proportional control:

$$\begin{aligned} u(t)&=K_p e(t),\\ C(s)&=\frac{U(s)}{E(s)}=K_p. \end{aligned} \tag{2.20}$$

Integral control:

$$\begin{aligned} u(t)&=K_i\int_0^t e(\eta)d\eta,\\ C(s)&=\frac{U(s)}{E(s)}=\frac{K_i}{s}. \end{aligned} \tag{2.21}$$

And derivative control:

$$\begin{aligned} u(t)&=K_D\dot{e}(t),\\ C(s)&=\frac{U(s)}{E(s)}=K_D s. \end{aligned} \tag{2.22}$$

In practice, a first-order filter is usually added in the derivative term to avoid high frequency signal amplification. Because the derivative term will multiply with changes in error value and stay still when error value is constant, this inflicted a disadvantage when error value has high frequency oscillation. With that, the derivative term with a first-order filter will be expressed as equations (2.23).

$$\begin{aligned} u(t)&=K_D\dot{e}(t)-T_f\dot{u}(t),\\ C(s)&=\frac{K_D s}{T_f s+1}. \end{aligned} \tag{2.23}$$

Unlike analog controller, digital control system such as microcontroller cannot handle integral and derivative operators. Thus, above equations will need to be transformed into discrete-time form which contains only simple math operators and recomputed between periods.

The discrete from of PID controller is depending on the discrete-time integrator formulas which will affect to the integrator and derivative terms. The popular settings on

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

these are applying bilinear transform with integrator and backward Euler formula with derivative term.

Bilinear transform is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane. When the Laplace transform is performed on a discrete-time signal, the result is precisely the Z transform of the discrete-time sequence with the substitution of equation (2.24).

$$z = e^{sT_s} = \frac{e^{sT_s/2}}{e^{-sT_s/2}} \approx \frac{1+sT_s/2}{1-sT_s/2}. \tag{2.24}$$

Equation (2.24) can be solved for s or a similar approximation for $s = (1/T)\ln(z)$ can be performed. The inverse of this mapping as the first-order approximation is

$$\begin{aligned} s &= \frac{2}{T_s}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \cdots\right], \\ s &\approx \frac{2}{T_s}\frac{z-1}{z+1} = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}. \end{aligned} \tag{2.25}$$

Substituting equation (2.25) into equation (2.21) yields equation (2.26).

$$C(z) = K_i \frac{T_s}{2}\frac{1+z^{-1}}{1-z^{-1}}. \tag{2.26}$$

Performing the inverse Z transform on equation (2.26) yields equation (2.27).

$$u[n] = K_i \frac{T_s}{2}\left(e[n] + e[n-1]\right) + u[n-1]. \tag{2.27}$$

Applying the bilinear transform with the derivative term from equation (2.22) yields

$$C(z) = K_D \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}. \tag{2.28}$$

Performing the inverse Z transform on equation (2.29) yields equation (2.30).

$$u[n] = K_D \frac{2}{T_s}\left(e[n] - e[n-1]\right) - u[n-1]. \tag{2.31}$$

However for the derivative term, backward Euler method is generally used. In the time domain, the backward Euler method is presented in equation (2.32).

$$\dot{e}(t) \approx \frac{e[n]-e[n-1]}{T_s}. \tag{2.32}$$

Using equation (2.32) with equation (2.22) yields equation (2.33).

$$u[n] = \frac{K_D}{T_s}\left(e[n] - e[n-1]\right). \tag{2.33}$$

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Applying Z Transform to equation (2.33) yields

$$C(z)=\frac{K_D}{T_s}\left(1-z^{-1}\right). \tag{2.34}$$

Thus, with the bilinear transform applied to the integral term and the backward Euler method applied to the derivative term, the resulting PID controller equation is

$$C(z)=K_p+K_i\frac{T_s}{2}\frac{1+z^{-1}}{1-z^{-1}}+\frac{K_D}{T_s}\left(1-z^{-1}\right). \tag{2.35}$$

2.3.1 Skogestad's PID tuning method

This section will describe the method used in PID tuning process of this project. There is a large number of tuning methods, but the one that is model-based and recently used in the field is Skogestad's [6].

Skogestad's PID tuning method is a model-based method where the controller parameters are expressed as functions of the process model parameters. It is assumed that the control system has a transfer function block diagram as shown in Figure 2.2.

The design principle of Skogestad's method is as follows. The closed-loop transfer function $T(s)$ is specified as a first order transfer function with time delay:

$$T(s)=\frac{1}{T_C s+1}e^{-\tau s}, \tag{2.36}$$

where T_C is the time-constant of the control system which the user must specify, and τ is the process time delay which is given by the process model.

From the block diagram in Figure 2.2, the closed-loop transfer function is

$$T(s)=\frac{C(s)G(s)}{1+C(s)G(s)}. \tag{2.37}$$

Setting equation (2.37) equals to equation (2.36) yields

$$\frac{C(s)G(s)}{1+C(s)G(s)}=\frac{1}{T_C s+1}e^{-\tau s}. \tag{2.38}$$

In fundamental control theories, this closed-loop equation will have output's time response to step input signal as in Figure 2.3 [7].

Skogestad's tuning formulas for a number of processes are shown in Table 2.1. In this table, "min" means the minimum value of the two alternative values. Originally, Skogestad defined the factor c in Table 2.1 as 4. This gives good set point tracking. Reduction of c will result more overshoot in the step response, and that the stability of the control loop will be reduced. Thus the robustness against changes of process parameters will be reduced as well.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

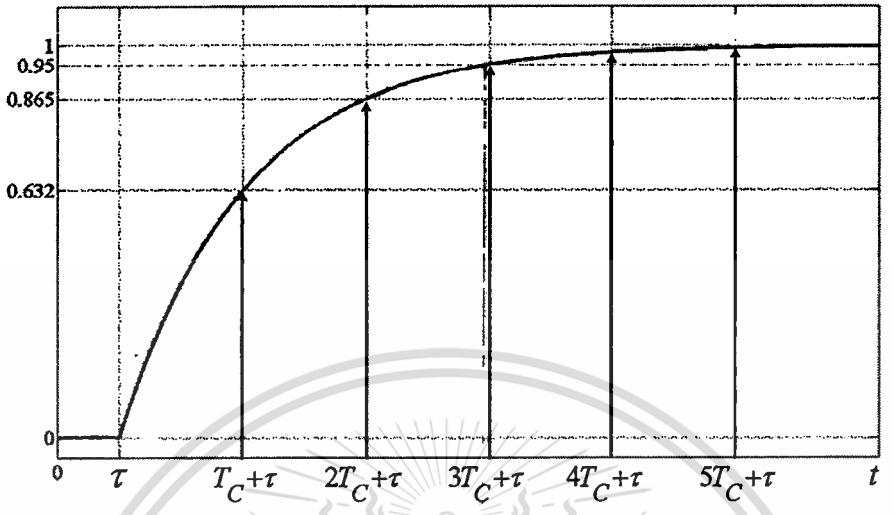


Figure 2.3 Exponential response curve.

The PID controller can be designed by using parameters from Table 2.1 to construct a controller expresses by equation (2.39). For T_c, Skogestad suggests using $T_C = \tau$ for T_C in Table 2.1 unless you have reasons for a different specification.

$$C(s) = K_p\left(1+\frac{1}{T_i s}+T_d s\right). \tag{2.39}$$

Table 2.1 Skogestad's formulas for PID tuning.

Process type	$G(s)$	K_p	T_i	T_d
Integrator + Delay	$\frac{K}{s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c(T_C+\tau)$	0
1st-order lag + Delay	$\frac{K}{Ts+1}e^{-\tau s}$	$\frac{T}{K(T_C+\tau)}$	$\min[T, c(T_C+\tau)]$	0
Integrator + 1st-order lag + Delay	$\frac{K}{(Ts+1)s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c(T_C+\tau)$	T
Two 1st-order lag + Delay	$\frac{K}{(T_1 s+1)(T_2 s+1)}e^{-\tau s}$	$\frac{T_1}{K(T_C+\tau)}$	$\min[T_1, c(T_C+\tau)]$	T_2
Double integrator + delay	$\frac{K}{s^2}e^{-\tau s}$	$\frac{1}{4K(T_C+\tau)^2}$	$4(T_C+\tau)$	$4(T_C+\tau)$

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 3

Design Procedures

3.1 Overall System Structure

The system structure depicted in Figure 3.1 is used in the project. Each part will be examined and explained in this Chapter along with its designing details.

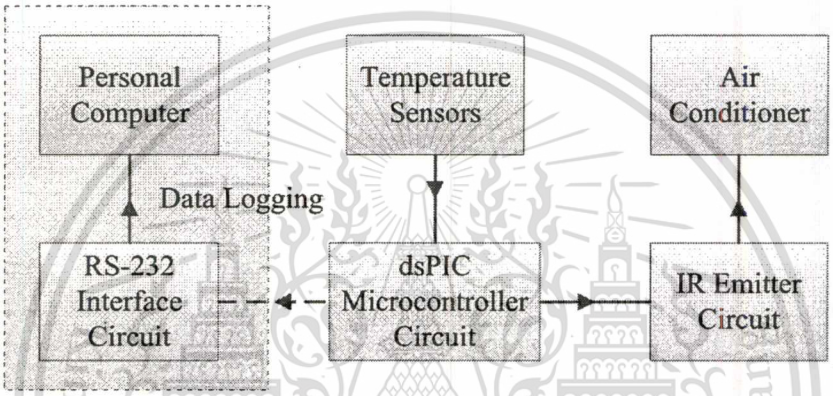


Figure 3.1 Overall system structures.

In this testbed, a dsPIC microcontroller is used for sensory data measurement, computing control signal, generating command signals, and experimental data transmission to a personal computer. IR emitter circuit is used to send the command signals to the air conditioner using a designated protocol of the infrared signals. Air conditioner is the actuator of this system and is assumed to have a working status as commanded. As data logging is necessary in experimental stage, a data logging system in this project uses a personal computer to retrieve values from the microcontroller using RS-232 protocol.

This project is designed to use in Control Theories and Computation Laboratory (ECC-211) at King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand. The room is connected to the outside environment through windows and an exhaust fan in the south wall. There is a door in the north wall connecting to the walking corridor in the building. To the west and the east, there are other rooms namely ECC-210 and ECC-212.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.2 Air Conditioning System

Air conditioning means improving the indoor environment during anytime of the year. An air conditioning system is either used to cool the environment during the warming time or to warm the environment during the cold time. To state this in another way, air conditioning refers to the control of temperature, moisture content, cleanliness, air quality, and air circulation as required by occupants, processes, or products in the space. This definition was first proposed by Willis Carrier, an early pioneer in air conditioning. Interesting biographical information on Carrier is given in his own book [8]. Carrier is credited with the first successful attempt in 1902 to reduce the humidity of air and maintain it at a specified level. This marked the birth of true environmental control as we know it today. Developments since that time have been rapid.

In these days, many great changes are taking place in the world as global warming is getting concerned in many topics leading to researches, restrictions (both voluntarily and by enforced laws), consumer product competitions, etc. Although saving energy is mostly voluntary, it should be a primary concern if the saving amount is significant.

The basic ideas of air conditioning can be obtained from the second law of thermodynamics, which states that heat flows from hot areas to cold, not the other way around. People in the ancient ages understand this principle as well and have invented many air conditioning applications [9] before today's electrical driven air conditioning system.

3.2.1 Components

This project uses a single stage air conditioner which has already been installed in the room without any intentions to modify its structure and power system. However, understanding the process of general air conditioning process can help improve procedure designs in making considerations for model, manipulating system, and control method.

Generally, there are three segments of the system—the exterior section known as a condensing unit; the interior section, comprised of the furnace assembly; and the refrigeration lines that connect the furnace with the condensing unit.

The condensing unit contains compressor, condenser coil, and fan.

- *Compressor (or heat pump)* is the electric pump, can be referred as the heart of the system, which circulates the refrigerant in a closed loop between the condenser and the evaporator coil. This is the most expensive part of the air conditioning system. Thus, we usually replace the whole system when we lose its compressor. This device also draws most of the power used in the air conditioner.
- *Condenser coil* is a network of tubes filled with refrigerant that remove heat from the heated gas refrigerant and convert the refrigerant into a liquid form again. The excess heat escapes into the outside air.
- *Fan* in the condensing unit is used to pull air through the condenser coil for heat dispersal.

The interior section contains evaporator coil, air handling unit, and air filter.

- *Evaporator coil* is a network of tubes filled with refrigerant that remove heat and moisture from the air as the refrigerant evaporates into gas again.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

- *Air handling unit* is the blower and related portion of the central air conditioning system that move air through the air ducts.
- *Air filter* is used to filter out dust, pollen, and other airborne particles as air moves through the air conditioning system.

Lastly, the refrigeration line is needed in the process to transfer refrigerant between the condenser unit and the furnace.

3.2.2 Power Consumption

An air conditioner draws a number of power consumption, which can be divided into parts for the benefit of control and researching process. The air conditioner used in this project is Carrier 38RC060SC with specifications as in Table 3.1.

Table 3.1 Carrier 38RC060SC specifications.

Name	Value	Additional details
Power supply	380 V	3PH, 50Hz
Rated load	10.35 A	
Power consumption	5,570 W	
Capacity	60,000 BTU/H	17,580 W
E.E.R.	10.45	
Refrigerant	R-22	
Fan motor	220 V	1PH, 8HP

As shown in Table 3.1, this air conditioner contains a compressor with the power of 17,580 Watts. Like every motor or pump would be, there certainly is a transient spike in power when the compressor is turned on. Due to lack of proper means to acquire the exact value of power consumption during the transient spike, this value will not be included for consideration in this project. Although the surge current has very high value, it appears only for a short time so that the power consumed during that time is very low [2]. However, the number of surge transient will be recorded for the benefit of future works.

3.2.3 Input Definition

As this project requires input signal in the form of discrete-time signal with continuous amplitude, we need to generate such signal for this system from its default binary signal. One method that is popular in control engineering field is by using pulse width modulation (PWM) signal as the input signal over the controller's sampling period. Usually this method works well with slow response system compared to frequency of the signal that the change of output value during controller's sampling period can be ignored. Although significant changes during controller's sampling period are expected, this is the easiest way to manipulate the system with the least modifying with the air conditioner hardware. Other problems still arise with this chosen conversion method such as how we define controller's sampling period and what should be constraints for input values.

Hardware limitation and safety should be prioritized for defining controller's sampling period. Usually the compressor should not be started running again before

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

fifteen minutes after its previous start [2]. This is due to the fact that there will be a huge surge of current during its transient time when it is starting. Having to face this situation repeatedly will gradually reduce the compressor's lifetime. Especially with PWM signal that the situation that occurred nearly every controller's sampling period. Built-in on-off controller have a protection for compressor, it will not drive compressor until one and a half minutes had been passed from the last turn off operation, and its sampling period is three minutes.

As the compressor will continue to cool the air for some time after it is turned off, this can be a constraint to the input system. A research shows that this value can be up to seven and a half minutes in Berkeley [2]. However, this value will be varied with the weather in Bangkok so we need to find the appropriate value by experiments. However, this project will not include the use of any constraint.

3.3 Wireless Remote Controller

Low price consumer wireless remote controllers available in the market are using infrared technology with a small range of carrier frequency. An IR control system consists of two parts, an IR transmitter with coding and driver circuit, and an IR receiver with decoding circuit. In this project, the IR transmitter is an IR driver being driven by the microcontroller. The IR detector can be built up from photocells connecting to demodulation circuit for the received signal to retrieve the data signal properly.

The other choice is to use a single chip with build-in modulator circuit, but detection of other carrier frequency is impossible. For example, v4838 is an IR detector equipped with a demodulator to demodulate IR signal at 38 kHz. The IR signal generated by simply pouring power directly into IR transmitter will not be detected by this chip; it has to be PWM signal at 38 kHz. If other frequency is applied to the IR drive circuit, the output of this device is appeared as if nothing is detected. Thus, the drive system will have to drive IR LEDs with the frequency of 38 kHz to contact with the air conditioner's wireless panel as well.

3.3.1 Receiver Circuit

To imitate the remote controller signal, its data signal must be recognized. IR detector module IC will be used to detect the data signal. However, though IR wavelength itself is always the same, its data carrier frequency is not.

An IR photocell can be used to unveil remote controller's carrier frequency by connecting the photocell in series with a resistor and supply, then measure the voltage across its pins. Photocell circuit is depicted in Figure 3.2.

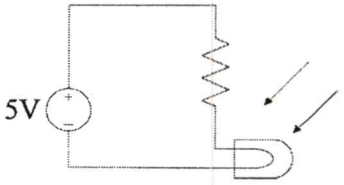


Figure 3.2 IR signal detection circuits.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Experimental results will be presented in the next Chapter. An experimental result in Figure 4.1 shows that the carrier frequency is 38 kHz. Once the carrier frequency is recognized, IR detector IC can be used to measure the data signal. Vishay's v4838 is chosen to do the job. This device is needed to confirm correctness of the generated signal.

Vishay's v4838 is a miniaturized receiver for infrared remote control systems, the epoxy package acts as an IR filter. Its output is demodulated and can be directly decoded by a microprocessor. Its block diagram is depicted in Figure 3.3. Experiments will be conducted by supplying direct current voltage to pin 3 and connecting pin 2 with common ground, and measure pin 1 using oscilloscope.

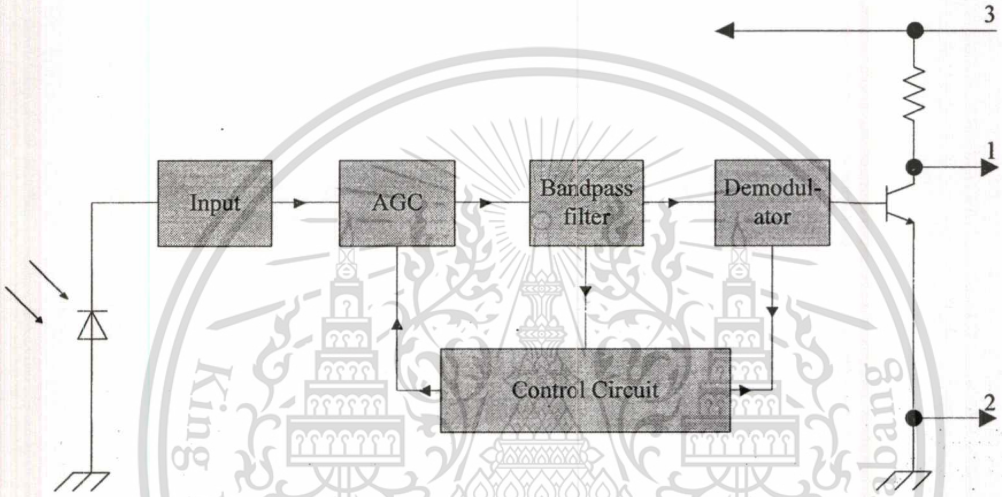


Figure 3.3 Vishay's v4838 block diagram.

3.3.2 Transmitter Circuit

To imitate infrared remote control system, IR emitter circuit is needed to create infrared signal to the AC control panel. Figure 3.4 shows the circuit diagram used for driving IR LEDs, IRx denotes microcontroller's pin. This circuit is designed to draw huge current from its supply to enhance IR LEDs performance and more mobility of the workstation. With that, the power supply for this circuit has to be separated from the control board's power supply to ensure power stability of the microcontroller when the circuit is activating.

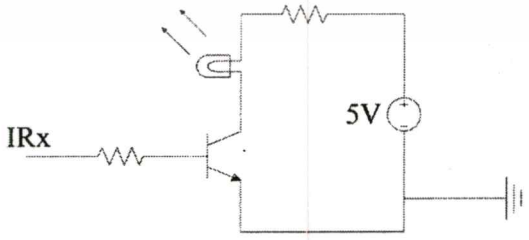


Figure 3.4 IR emitter circuit.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.4 Microchip dsPIC30F

As one of this project's objectives stated: build the least complex control system that can do the entire tasks. The system centered by a microcontroller is a good choice, since the system will contain very few of circuit boards and the imitation of the system can easily be done with a glimpse.

A microcontroller unit (MCU) is a small computer on a single integrated circuit, usually containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are embedded in many control, monitoring, and processing systems. They are usually used in specialized systems such as washing machines, telephones, microwave ovens, and automobiles. The main reason for using Microchip's microcontroller in this project is the availability of development tools and demonstration boards in the lab. They can also be used as reference design of a custom board later on.

Microchip offers two families of 16-bit Digital Signal Controller (DSC) devices, dsPIC30 and dsPIC33, with wide selection of price, performance, and feature sets. Microchip dsPIC30F microcontrollers have limitation of computation power at 30 MIPS (mega instructions per seconds). However, when the DSC is running at 30 MIPS, there will be considerable heat dissipation although it does not exceed its working zone as stated in the datasheet. The control circuit will be designed using Microchip dsPIC30f6014A as the main microcontroller because Microchip dsPIC30f6014A has most pin with the fullest capacities in peripheral of the family. Attaching more sensor and equipment in future work will not require any redesign of the control circuit. If this device cannot provide enough peripheral and compute system, then they will need to upgrade the processor to dsPIC33 series.

Microchip dsPICdem1.1 is used during earlier stage of the project. It is a demonstration circuit premade with peripherals connected, and its stability is ensured. The circuit uses dsPIC30F6014 as the main processor, dsPIC30F6014A is the improved version of dsPIC30F6014. They're identical to each other.

This family has integrated ADC module with 12-bit wide data that can be sampling with high speed rate. General microprocessors will have 10-bit ADC module. It is not enough for converting sensor data to use in control.

3.4.1 Timers

Depending on the specific variation, the dsPIC30F devices offer several 16-bit timers. Some of them can be combined to form a 32-bit timer. However, all have only 2 control registers:

- TxCON
- PRx

Timers can be used to measure time to make decision or doing some tasks by using interrupt events that will force the controller to go to the interrupt program when the counting register's value is matched with *PRx* value. The counting rate can be calculated from the system frequency and the prescaler register. Interrupt period can be determined by assigning *PRx* value as

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content and cite the document when use.
76416

$$PRx = \frac{T_{\text{TMR interrupt period}}}{T_{CY}(\text{TMRx Prescale Value})} - 1 \quad (3.1)$$

where T_{CY} is microcontroller's oscillator period, and $T_{\text{TMR interrupt period}}$ is timer's interrupt period.

Timer modules are used to perform many tasks for the project: schedule the tasks, specify sampling period, specify time base for output compare module, and specify input period.

3.4.2 Output Compare

The Output Compare (OC) module has the ability to compare the value of a selected time base with the value of one or two compare registers (depending on the operation mode selected). On a compare match event, resulting actions are depending on the operation mode selected, including interrupt events.

Each OC channel has the following registers:

- OCxCON
- OCxR
- OCxRS

This module is used to generating pulses during IR remote signal transmission stages. It is easy to configure. After properly assigned OCxR and OCxRS values, the program can start generate pulse train signal from OCx pin by turning on the module to PWM mode and stop generate pulse train by shutting down the module.

3.4.3 Analog to Digital Converter

As a control system includes a sensing system, Analog to digital converter (ADC) offers wide and easy way to communicate with the sensor directly. The ADC module in dsPIC30f6014A can communicate with up to 16 devices in the speed of 200ksps (kilo samplings per seconds) for 12-bit conversion. With this speed, the sensing program of 16 devices can be done in milliseconds, which is more than enough when the system is working in period of minutes.

The ADC module has six control and status registers.

- ADCON1
- ADCON2
- ADCON3
- ADCHS
- ADPCFG
- ADCSSL

The ADC initialization program starts by setting up pins for analog function, choose voltage reference, convert sequence, and then turn the module on. For the conversion process, the sampling will start by setting the SAMP bit. Then wait until DONE bit is set by hardware, and then read the value from ADCBUF0. For this project, the value from ADCBUF0 will be read as temperature by using equation (3.3).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.4.4 UART

This project uses this module to communicate with a personal computer to log data during experimentation stages. The Universal Asynchronous Receiver Transmitter (UART) module in dsPIC30F device family is a full-duplex asynchronous system that can communicate with peripheral device, which in this project, is a personal computer through RS-232 interface.

A UART module consists of the following key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

There are 5 control registers involved in UART module:

- UxMODE: UARTx Mode Register
- UxSTA: UARTx Status and Control Register
- UxRXREG: UARTx Receive Register
- UxTXREG: UARTx Transmit Register
- UxBRG: UARTx Baud Rate Register

As the only role of UART module in this project is to send data to log into computer for further investigation, receiving part of this module is not used. Thus only UxMODE, UxSTA, and UxBRG are used. For the use of UART module, baud rate is needed to be properly calculated and assigned the value needed into UxBRG register as in the following equation.

$$UxBRG = \frac{F_{CY}}{16*(\text{Baud Rate})} - 1. \tag{3.2}$$

Once the module is enabled, the UxTX and UxRX pins are configured as an output and an input, respectively, overriding the TRIS and PORT register. Also, for the use of this module in the project, a premade function *printf* is used for transmitting string data. Thus, after a proper setup of UxMODE, UxSTA, and UxBRG, there is no need to meddle with UART registers again. Except for switching frequency case, we'll have to configure UxBRG to the right value and enabling the module again.

3.5 Temperature Measurement

One of the main components of a control system is its sensors. For this system, the controller needs temperature value to compute its control act for the control period.

3.5.1 Temperature Sensor TC1047A

Microchip TC1047A is chosen in this project as the suitable temperature-to-voltage converter available in the market for this project. It can be supplied by the main processor's power supply and its accuracy is $\pm 2^{\circ}$C at 25 C with linear temperature slope at 10mV/C.

From the given relations between the temperature and its outputs at some specific points in the datasheet, the conversion equation from the 12-bit register of ADC module is made as follow.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$T = \frac{\frac{5.0}{4096.0} ADCBUF0 - 0.5}{0.01}. \quad (3.3)$$

For the proper usage of this device, Microchip provides an application note [10] to further explain the use of TC1047A. Along with PCB layout design techniques and an example firmware implementation for using the device with PIC microcontroller, there also are facts about the mechanism of the device for the reasons behind those techniques.

3.5.2 Temperature Measuring Program

As mentioned in section 3.4.3, the Analog to Digital Converter module on dsPIC30F will start sampling when the SAMP bit is set. Module's buffer will become readable once the sampling is completed and DONE bit is cleared automatically by hardware. The last task for the program will be reading the module's buffer and convert 12-bit integer data into temperature value using equation (3.3). Flowchart for temperature measuring program is depicted in Figure 3.5.

3.6 Controller Circuit

Although dsPICdem1.1 is used during experimental stage, a design of custom circuit board for controller is required. The circuit is designed to contain all required parts for this project and some other parts from dsPICdem1.1 are added to use the circuit for other applications. The circuit is kept as simple as possible. Sensors and IR emitters are not installed in the board in order to allow mobility of the system, and the future works might attach more devices. The circuit structure composes of three parts; processing part, interface part, and power part.

Processing part includes connection of the microcontroller to peripherals and connectors. Every microcontroller pins are connected to connectors for ease of other uses. Two capacitors are connected between two power lines (5V and ground) near the microcontroller to improve its stability. The microcontroller's reset button is a tact switch (S1). Remaining tact switches are implemented for demonstration purposes or other uses.

Pull-up resistors that are connected between VDD and pins are required to limit current that will flow into the microcontroller to its range as stated in datasheet. Also resistors that are connected between logic pins and ground are there to provide further safety for other uses and unexpected events. Capacitors that are added between the pins and ground are designed to soften the voltage to the pin; this is implemented to make sure that only one step event is occurred per tact. Processing part's schematic is shown in Figure 3.6.

Interface part includes circuit interface connections. The microcontroller can connect to MPLAB's external programmers using either RJ11 or 6 pins connectors. A 3x2 pins connector is designed for the project, interfacing to sensor and IR driver circuit. Another 3x2 pins connector is designed to provide different supply voltage to the ADC module. Also, there is MAX3232CSE RS-232 transceiver connected to microcontroller pins and its required components. Interface part's schematic is shown in Figure 3.7.

Power part includes the power regulator circuit. This circuit uses LM2940S-5.0 as a 5V regulator to provide stable 5V to the control circuit. Two capacitors are added to

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

improve stability to the regulator. There also is a diode to protect the circuit from reversed connection, and a LED to indicate the working status of the power part. Power part's schematic is shown in Figure 3.8.

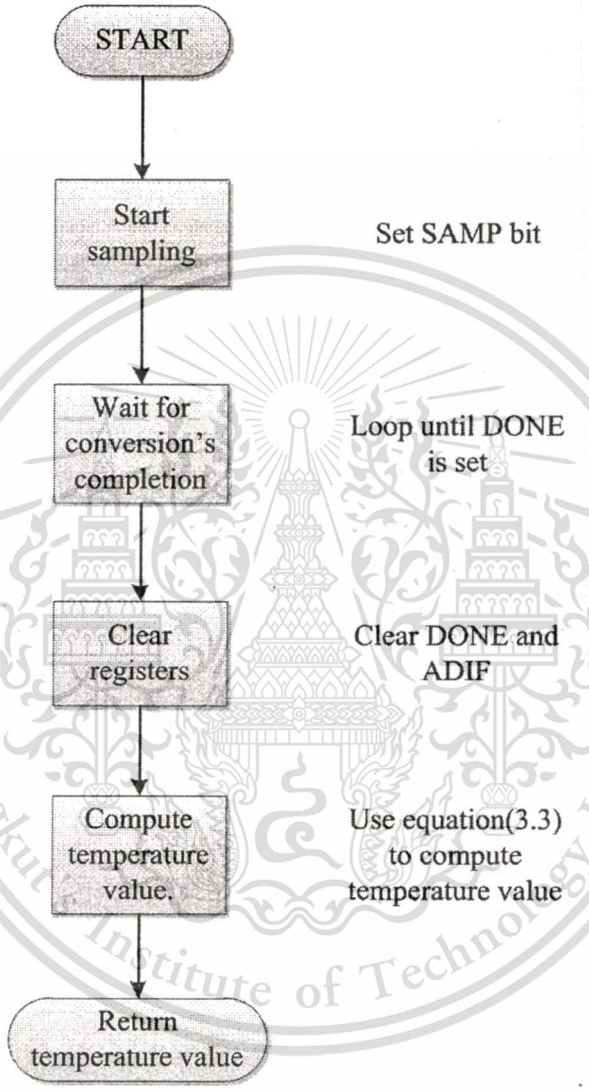


Figure 3.5 Temperature measuring program's flowchart.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

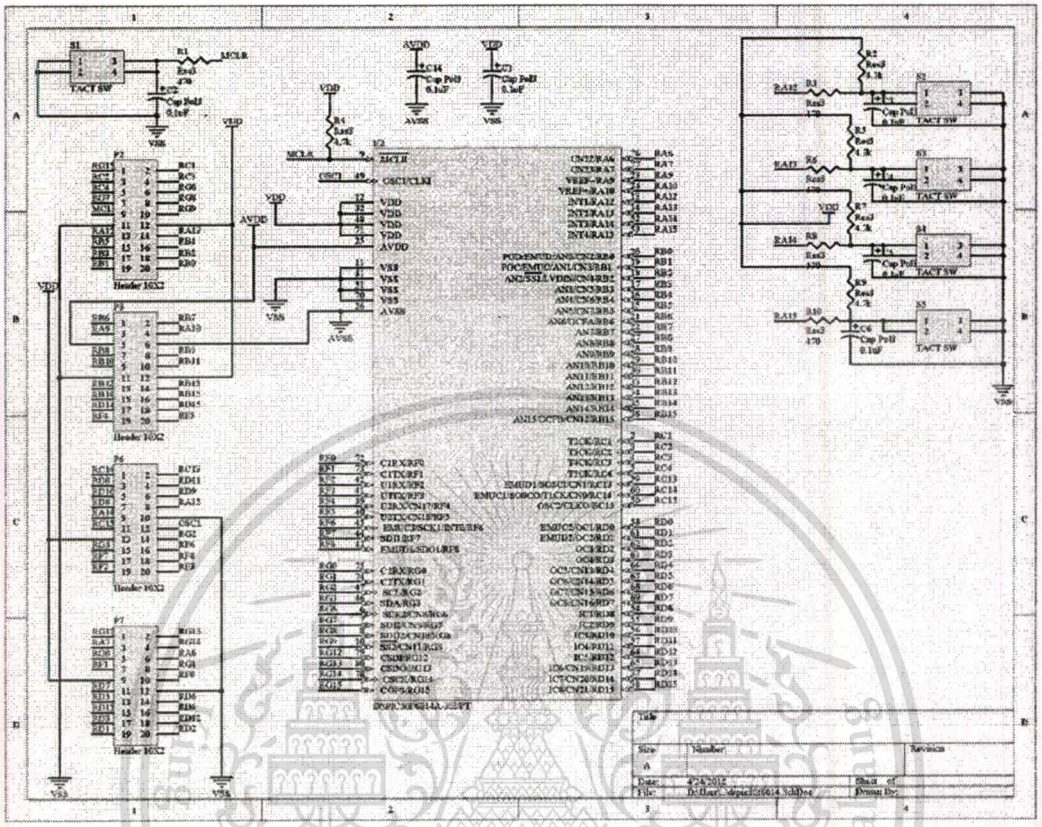


Figure 3.6 Control circuit schematics: processing part.

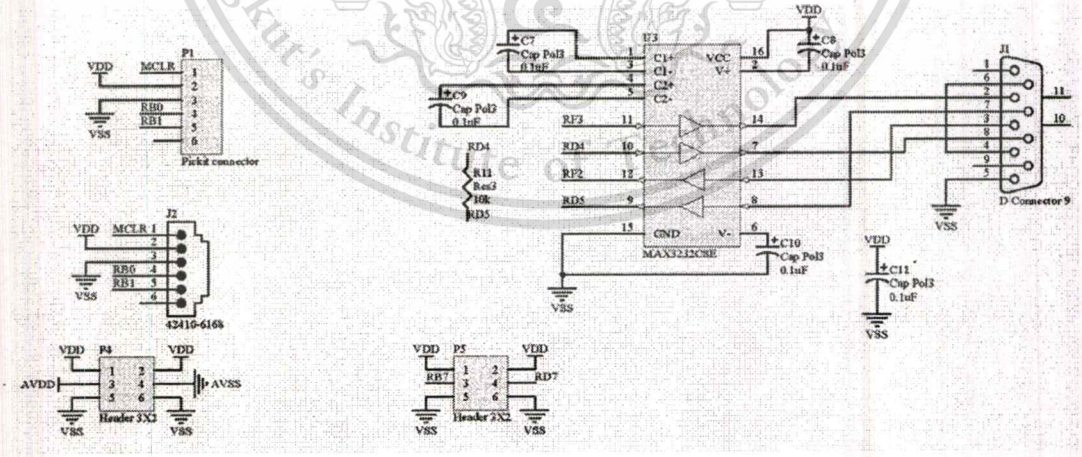


Figure 3.7 Control circuit schematics: interface part.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

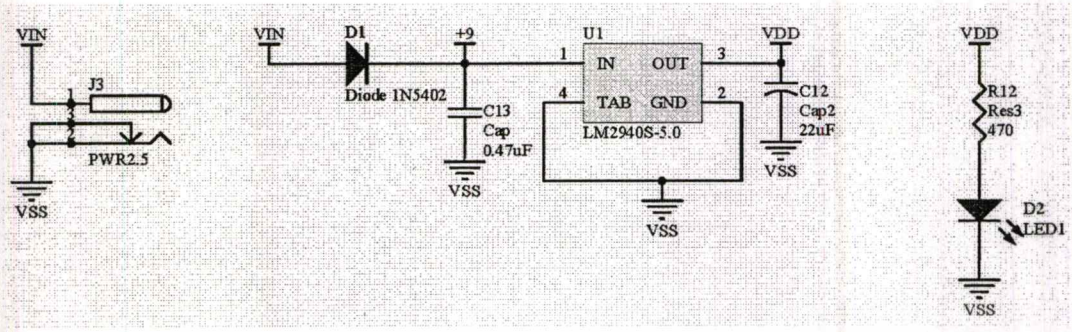


Figure 3.8 Control circuit schematics: power part.

The printed circuit board is manufactured as designed and working well for the project application. The picture of the completed control circuit is shown Figure 3.9.

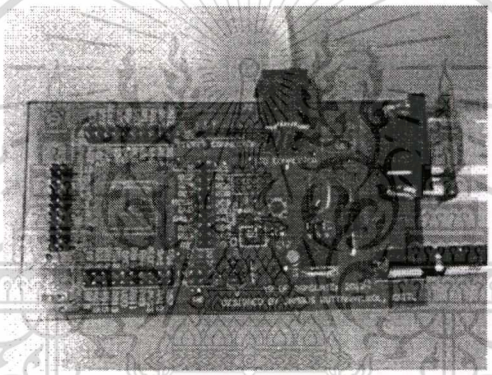


Figure 3.9 The control circuit board.

3.7 Remote Signal Imitation Program

With the knowledge of binary patterns generated by the air conditioner remote controller from section 4.1, the remote signal imitation program can be easily designed. As of the program designed in this project, the program can be divided into two parts; the main program, and the data comparison program.

The main program's flowchart is shown in Figure 3.10. The program starts by getting the desired compressor mode, and then initializes variables and starts producing pulse train by starting the OC module, and then just quits the program. This is the first stage of the program which is dictated by a variable named 'k'. To enter this stage, k has to be assigned with 0 and it will be assigned as 1 at the end of this stage to move on.

In the second stage, the program runs directly to program codes under k = 1. Its first task is to wait for 4 ms to let the OC module producing the pulse train. After that it will change the output status by stopping the OC module and move the program onto the next stage by assigning 2 to k. The microcontroller is not actually just waiting; it quits the program and repeatedly enters this program to check the clock variable, if 4 ms had passed, the microcontroller will move on to the remaining statements which is to increase k, reset clock variable, and stop pulse train. This procedure of watching the time is

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

applied to every waiting block in this program. It allows the microcontroller to do other jobs during the waiting time. The third stage is similar to the second stage but waiting for 2 ms instead to let the OC module inactivate. After that the program will assign 3 to k and start the OC module to produce the pulse train again.

The fourth stage of the program is the gap between data and getting the next data to be sent. It starts by waiting for 0.6 ms, and then stop OC module and reset the clock variable. Then the microcontroller will check if the data signal is completely sent, if so, 0 will be assigned to k and the program will not be called again until the desired mode is changed. If not, the program will move onto the data comparison sub-program.

The data comparison sub-program flowchart is shown in Figure 3.11. Its structure is not complicated. The program makes use of bit-wise operations such as comparing and shifting. It starts by comparing the current bit of the data with a comparing variable to see if it is clear or set. If the current bit is clear, 4 will be assigned to k. If not, 5 will be assigned to k instead. Then the comparing variable will be shifted to the left one bit.

Afterwards, the program will use variable k to decide how long it will wait until producing the pulse train again as data gap. If k is 4, the program will wait for 0.4 ms. And if k is 5, the program will wait for 1 ms. For both cases, the OC module is inactivate during the waiting time. After that, the program will start producing the pulse train and assign 3 to k so the program will repeat by returning to the fourth stage until the data signal is completely sent.

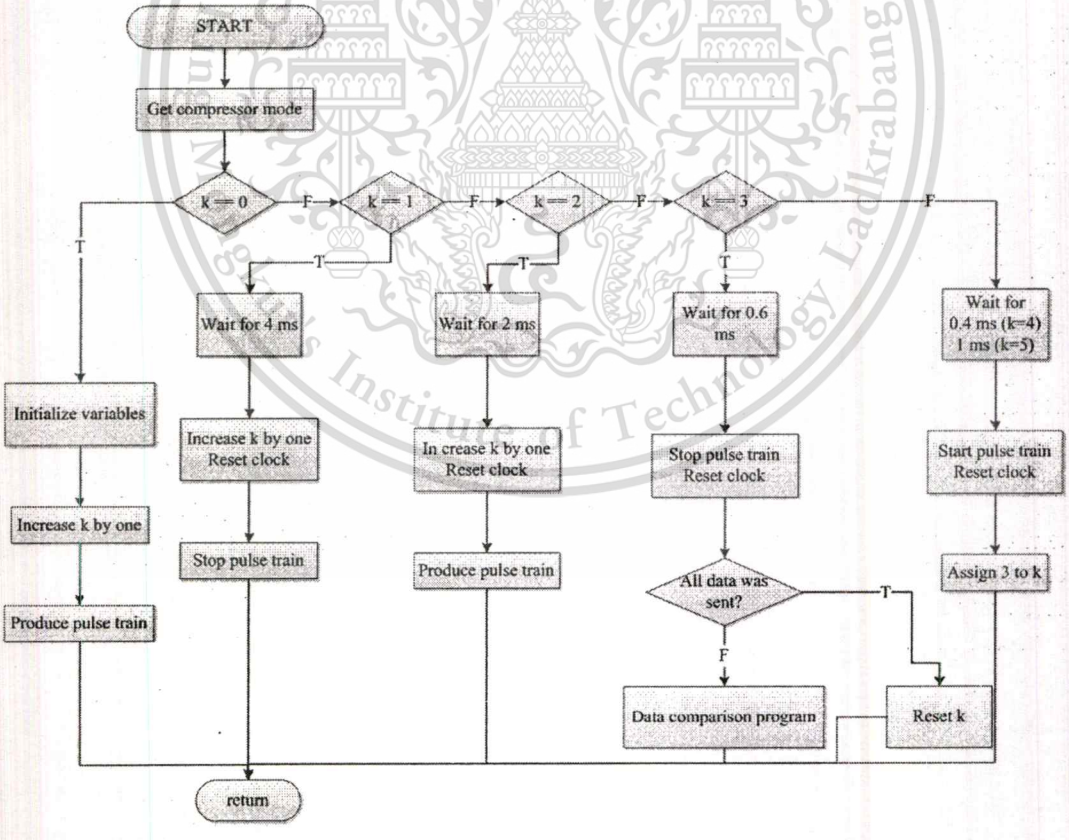


Figure 3.10 Remote signal imitation program's flowchart.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

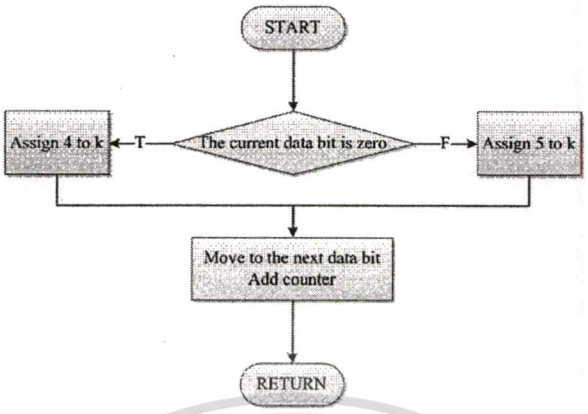


Figure 3.11 Data comparison program's flowchart.

3.8 On-off Control Program

In order to verify the effectiveness of the proposed control scheme, the on-off control program will be implemented to compare the results with the proposed PID control. The program with on-off controller is designed with hysteresis of 0.5 C and sampling time of 1.5 minutes. The reason for using this sampling time is that the thermostat does not respond for 1.5 minutes even if the temperature is high enough to turn the compressor on. Flowchart for the on-off controller program used in this project is shown in Figure 3.11.

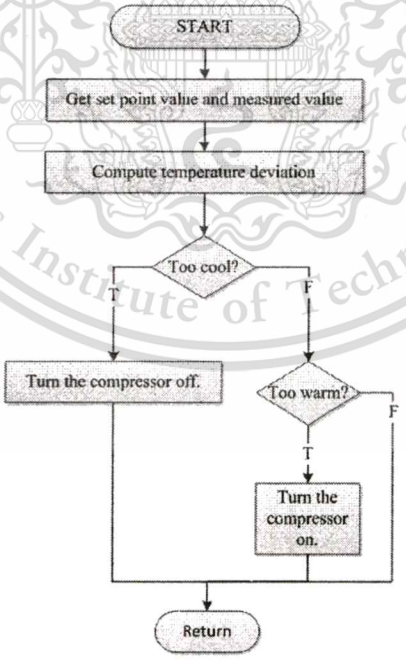


Figure 3.12 On-off control program's flowchart.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.9 PID Controller

After applying on-off control to confirm command ability and keep the data to compare its command with PID controller in term of approximated power consumed, and identifying parameters using random input, PID controller's parameter can be designed using Skogestad's tuning method as described in section 2.3.2.

3.9.1 PID Tuning

The main analysis tool is MATLAB. From plotting logged values into visual form, do linear least square with a single tap, and build model objects to do simulation or design control parameters and analyses.

MATLAB is a numerical programming environment for algorithm development, data analysis, visualization, and numerical computation, Developed by MathWorks. Its name is abbreviated from matrix laboratory for that most of operations in the environment are matrix manipulations.

PID tuning in MATLAB is convenient because of many pre-made analysis is included in the environment. Designing programs with graphical user interface (GUI) are available along with command line functions. However, in this study we will use Skogestad's rules of PID tuning to study PID tuning procedures.

As described in section 2.3.1 of this report, this method can be easily applied to any first order system either with or without time delay. However, models we have from identification process are discrete-time models. Transformation is needed to apply Skogestad's rules.

Equation (2.7) is modified to use in the system without installing sensors outside the room, ambient temperature and heat noise are lumped together and assumed to be a constant value to apply linear regression model in equation (2.8). The equation (2.7) becomes

$$T[n+1] = \alpha_n T[n] + \beta_n U_c[n] + Q, \tag{3.4}$$

where Q is the constant heat noise value. This model can easily be identified by linear least square method as in equation (2.13).

In discrete-time analysis, Z-transform is used instead of Laplace-transform. Equation (3.4) with Z-transformed yields

$$G(z) = \frac{\beta_n}{z - \alpha_n}, \tag{3.5}$$

where $G(z)$ is the system's transfer function in z domain. From experiments, the average values of β_n, α_n and Q are recognized as

$$\begin{aligned} \beta_n &= -2.9150, \\ \alpha_n &= 0.4497, \\ Q &= 15.8730. \end{aligned} \tag{3.6}$$

Substituting values from equations (3.6) to equation (3.4) and equation (3.5) yields its difference equation as

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$T[n+1]=0.4497T[n]-2.9150U_c[n]+15.8730, \tag{3.7}$$

and its transfer function as

$$G(z)=\frac{-2.9150}{z-0.4497}. \tag{3.8}$$

MATLAB offers transformation of discrete-time domain to continuous-time domain in many methods. As the system is sampled using zero-order hold method, this conversion method will be used. Applying zero-order hold method to equation (3.8) yields

$$G(s)=\frac{-0.004288}{s+0.0008222} \tag{3.9}$$

as its representation in continuous-time domain. Rearranging equation (3.9) to its relative form in Table 2.1 yields

$$G(s)=\frac{-5.21528}{1216.249s+1}. \tag{3.10}$$

PID parameters can be directly calculated from Table 2.1 and their values are as follows:

$$\begin{aligned} K_p &= -0.12956, \\ T_i &= 1216.249, \\ T_D &= 0. \end{aligned} \tag{3.11}$$

Constructing a PID controller using values from the above equations (3.11) yields

$$C(s)=-0.12956\left(1+\frac{1}{1216.249s}\right). \tag{3.12}$$

Equation (3.12) is a continuous-time transfer function. To use this equation in the microcontroller, it needs to be converted to discrete-time form. Bilinear method will be used due to its accuracy to the approximation. Using bilinear method to equation (3.12) yields

$$C(z)=\frac{-0.1775z+0.08167}{z-1}. \tag{3.13}$$

To apply equation (3.13) to the microcontroller, the controller equation needs to be in causal form. Computing equation (3.13) requires knowledge of future values of the output and input because z can be inverse z-transformed to the next values of the variable. Multiplying $\frac{z^{-1}}{z^{-1}}$ to equation (3.13) yields equation (3.14).

$$C(z)=\frac{-0.1775+0.08167z^{-1}}{1-z^{-1}}. \tag{3.14}$$

Take equation (3.14) into inverse Z-transform process yields equation (3.15).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$u[n]=0.08167e[n-1]-0.1775e[n]+u[n-1]. \quad (3.15)$$

Equation (3.15) will be used to computing input signal in PID control program in the next section.

3.9.2 PID Control Program

Program for a PID controller will be easier to construct than on-off controller with hysteresis. Its program will not contain more than ten statements because the heart of PID control is in its parameters tuning. Its easiness lies in the overall structure. That is one of the reasons PID is widely used. The program's flowchart is illustrated in Figure 3.12.

This program is designed to run as sub-routine in main once every 15 minutes. It starts by computing set point error value, then computes input value using difference equation of PID controller designed as in equation (3.15), then constraints input value between 0 and 1 since PWM input cannot provide larger input than 1 and lower than 0, and then return input value multiplied with number of counts per cycle of 15 minutes. This program works well in implementations to regulate room temperature at 25 C during daytime.

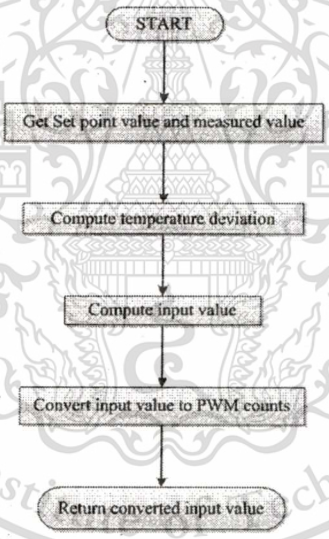


Figure 3.13 PID control program's flowchart.

3.10 Prediction Program

As mentioned in the earlier sections of this report that this project is aimed to reveal methods to reduce power used in air conditioning process. Prediction of turning off point before the terminated time is a way to go. There is social promotion to encourage users to turn the air conditioning system off for 30 minutes before the end of usage. However, this method does not seem to be effective to the whole because there are many conditions that affect temperature convergence. Such of them are the room structure, nonlinearity in the relationship of room temperature and ambient temperature, time-

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

varying ambient temperature acts as a noise to the system, uncertainties due to heat generated by the occupiers. Furthermore, the users themselves sometimes forget to do so.

The prediction program is designed by using fixed parameters from equation (3.7) to predict temperature in the next sampling period. Taking the input variable out from equation (3.7) yields

$$T[n+1] = 0.4771T[n] + 15.1806. \tag{3.16}$$

This autonomous model will gradually converge to 31.8185C. This value is the default temperature of the room without using air conditioning system when the outside temperature is about 35C.

The prediction program's flowchart is depicted in Figure 3.13. The user must specify the acceptable deviation from the set point temperature that the measured temperature should not exceed at the terminated time in the program. The program receives current temperature, set point temperature, and the remaining sampling period. It will run incurrence loop in the number of remaining period. Then the last period predicted value will be decided in a statement if it is too hot or not. If the predicted value is too hot, the microcontroller will continue to perform control action. If it is not, the microcontroller will stop performing control action and leave the compressor off to the end of usage. However, the user can override the shutdown command to perform control actions again to by pressing the reset button on the control circuit board.

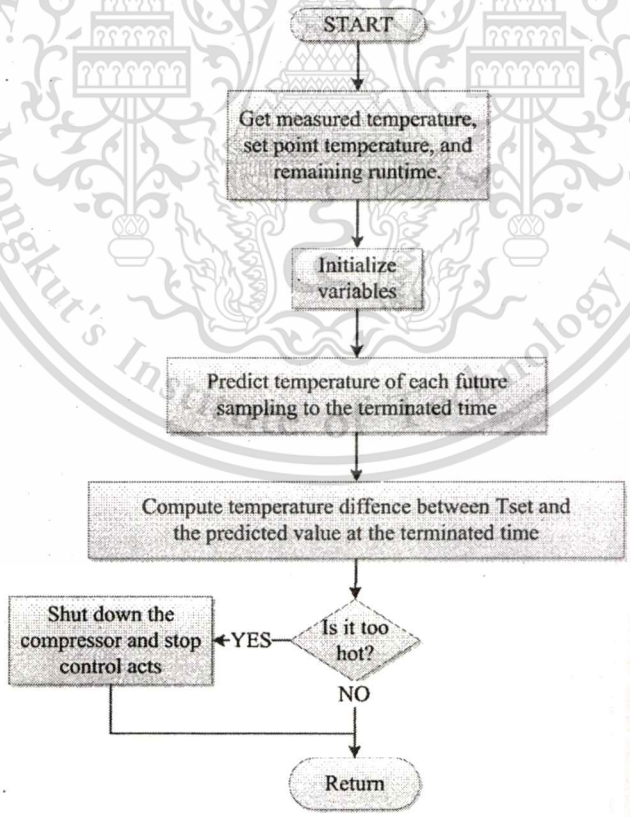


Figure 3.14 Prediction program's flowchart.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.11 Overall Microcontroller Program

Main program starts by declaring its variable and initializing modules using its initializing function. A control action requires the measuring temperature to compute the command signal for the compressor as the PWM duty cycle of the compressor's working time and then runs the prediction program. If the working time is not zero, the status variables for remote imitation program is set to "turn compressor on" state. Then temperature value, compressor state, PWM duty cycle, and prediction value are printed to the logging computer using UART module.

After initializing functions and running the first control action, the program enters the main iterative loop. The loop starts by checking whether the compressor's working time is reached. If so, the status variables are set to the "turn compressor off" state. If not, the status variables are unchanged. Then the program checks if a sampling period has passed. If not, the remote data signal imitation program is activated. But if sampling counter checking condition is met, sampling counter is cleared. Then the microcontroller measures the temperature and computes the new command signal for the compressor based on the PID control program and the prediction program and repeats the loop. For data logging purpose, the program records temperature value, compressor state, PWM duty cycle, and prediction value every minute triggered by the same sampling counter used in the iterative control loop.

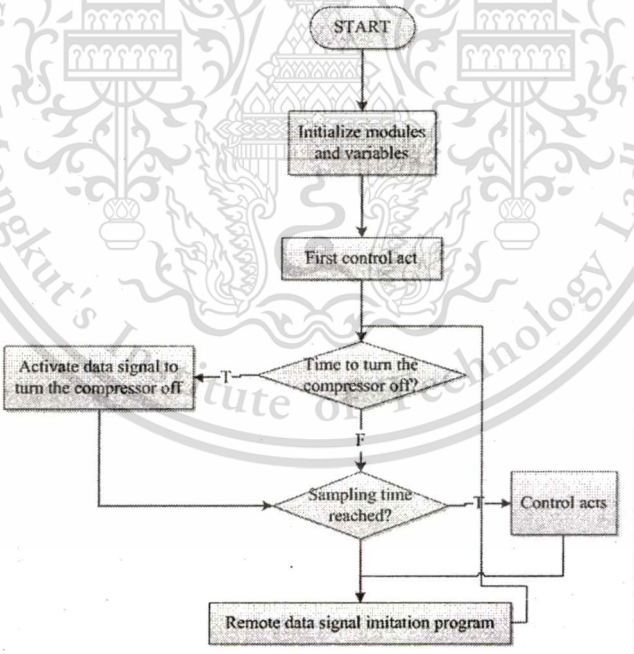


Figure 3.15 Overall microcontroller program's flowchart.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 4

Experimental Results

This Chapter shows the results from the experiments conducted in this project along with analytical results. This Chapter is roughly organized into 3 parts; remote signal decoding and imitation, model identification, and control.

4.1 Air Conditioner Remote Signal Decoding and Imitation

One of project's objectives is to build an air conditioner control system which is easy to use anywhere with the minimal modification. Therefore, the proposed system should simply use the same remote signal as generated by the air conditioner remote controller that comes with the air conditioner. This system has to neither modify working structure of the air conditioner nor wire any lines to it. To achieve this goal, this section presents some of the experimental results conducted during the program design procedure to retrieve and generate the air conditioner remote signal. It is divided into two parts. The first part is to show the results in measuring the signal from the remote controller. The other part is to show the imitation signal generated by the designed microcontroller program.

4.1.1 Signal Decoding

This section shows experimental results in decoding the air conditioner remote signal and their usage in the remote imitation program design procedure. The experiments in this section are conducted in the following order. First, an experiment to find the remote's carrier frequency is conducted. After that, the set of experiments using v4838 to capture the pattern of remote signal are conducted.

As described in Chapter 3, the remote carrier frequency has to be recognized. An experiment using the circuit in Figure 3.2 is conducted to measure the remote controller carrier frequency. The measured result is illustrated in Figure 4.1. It is shown that this signal contains 23 pulses in around 0.6 milliseconds. Thus, its period and frequency can be approximated to be about 26 microseconds and 38 kHz respectively.

Once the carrier frequency is recognized, the following experiments using Vishay's v4838 are conducted to measure data signal of some certain states that are indicated on the remote's screen when varying on/off compressor mode, on/off air conditioner state, and the desired temperature. Figures 4.2 through 4.6 shows the modulated data signal measured from v4838 measuring by an oscilloscope from these experiments. From these figures, it can be clearly seen that the data signal contains starting bit in the form of 4 milliseconds grounded and 2 milliseconds high voltage.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

According to Figure 3.3, this signal means that the header contains 4 milliseconds of firing IR signal at 38 kHz and stop firing for 2 milliseconds. After that, 44-pulses train is followed. This part can be considered as binary data because their high voltage pulse width varies in two values; 0.4 milliseconds and 1 millisecond. Then the signal is grounded for 0.6 milliseconds. The corresponding logic when the signal is high for 0.4 milliseconds then grounded is "0" and the other case is "1." From these patterns for logic "0" and "1", the data implied from Figures 4.2 through 4.6 can be summarized as in Table 4.1.

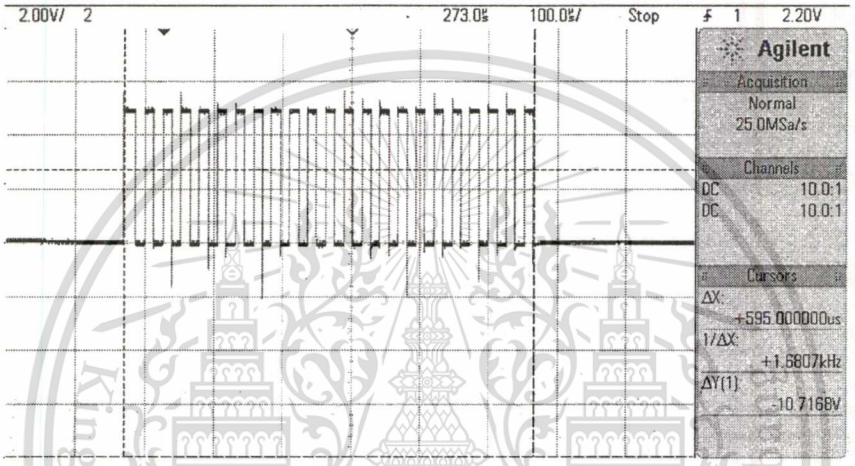


Figure 4.1 IR signal period measurement.

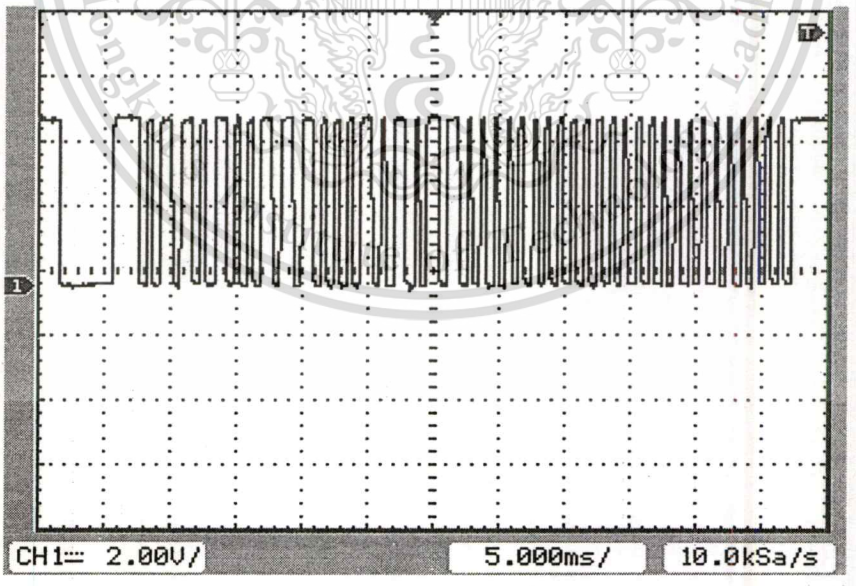


Figure 4.2 IR remote controller data signal (1).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

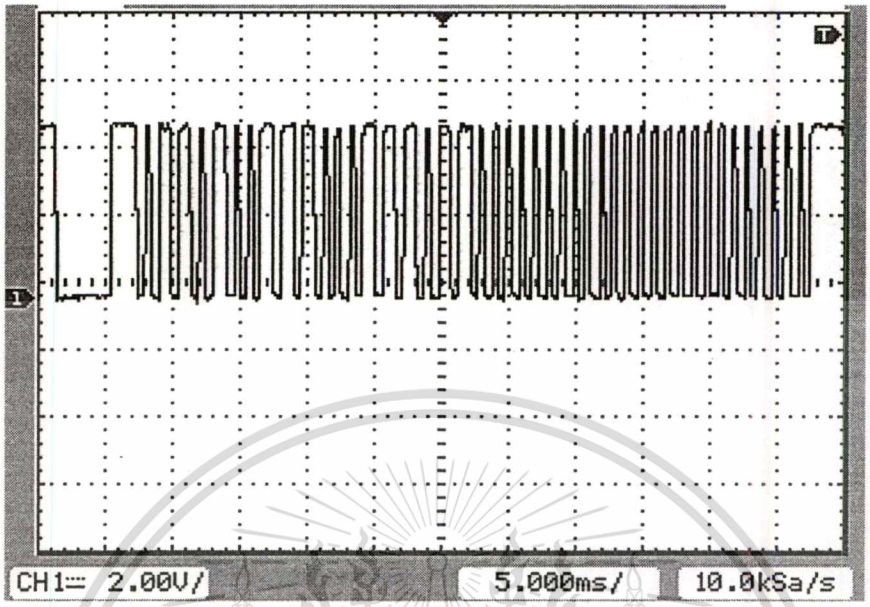


Figure 4.3 IR remote controller data signal example (2).

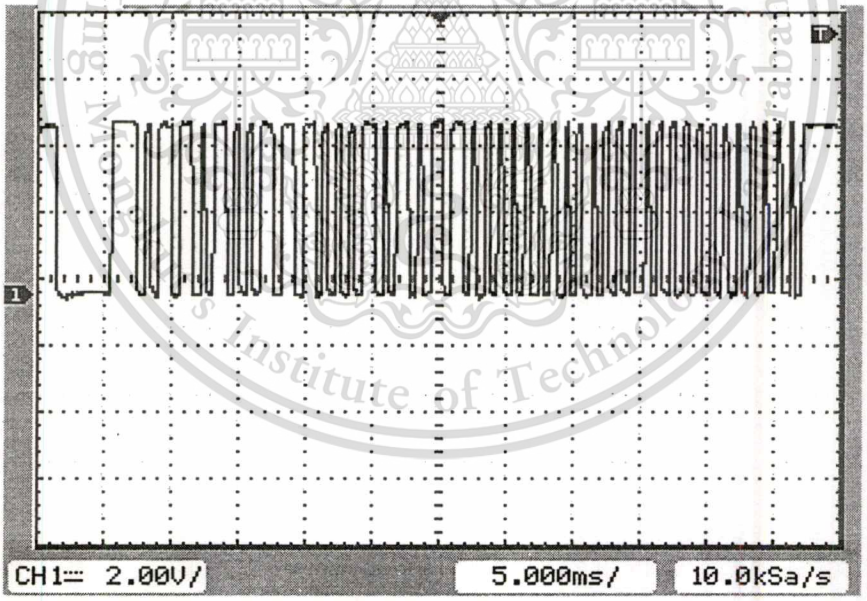


Figure 4.4 IR remote controller data signal example (3).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

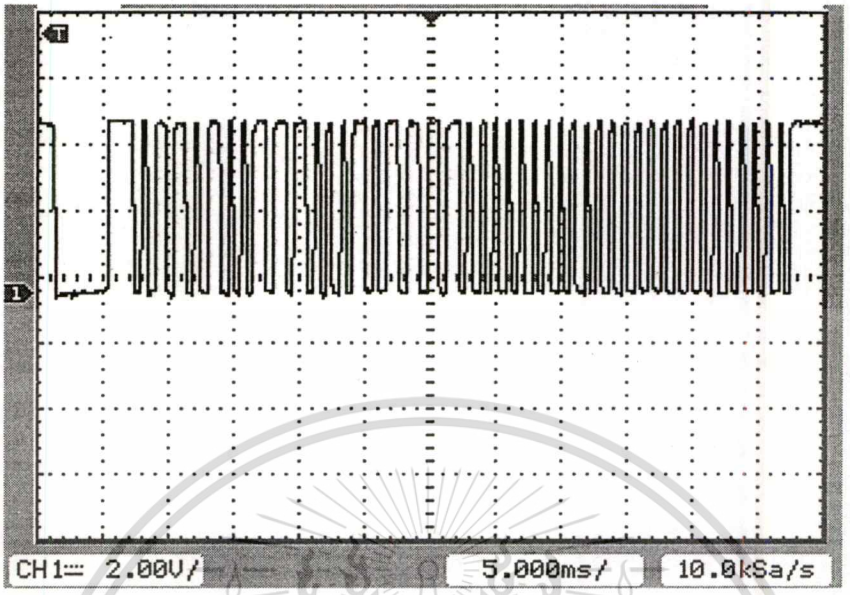


Figure 4.5 IR remote controller data signal example (4).

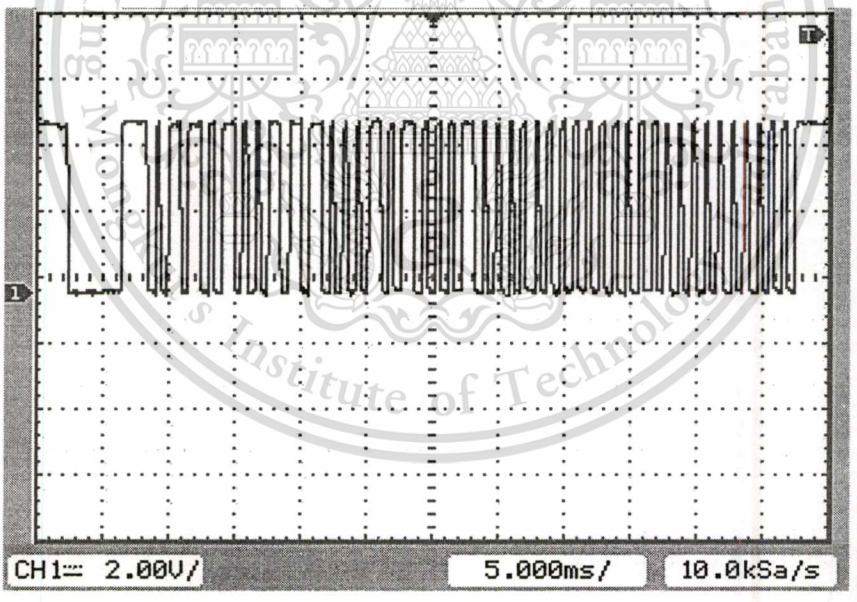


Figure 4.6 IR remote controller data signal example (5).

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Table 4.1 Remote data presented in examples.

Data	Corresponding binary data
Figure 4.2	01101001110001010110000000000000000000000000
Figure 4.3	01101001110001110110000000000000000000000000
Figure 4.4	01101001110001010110000000000000000000000000
Figure 4.5	01101001110001011110000000000000000000000000
Figure 4.6	01101001110001010001000000000000000000000000

From the data in Table 4.1, all of them can be separated into two groups; the first 20 bits, which are not all zero, and other 24 bits, which contain only zero. When controlling only on/off compressor mode, on/off air conditioner state, and the desired temperature as needed for this work, it can be noticed that the first 20 bits can also be divided into two parts. The first 14 bits are unchanged with the data 01101001110001 and the last 6 bits which are changed in these experiments. From the experiments shown in this thesis and several other omitted here, these last 6 bits can be interpreted to consist of three parts: MODE bit, ON-OFF bit, and 4-bit temperature value bits respectively. Thus, in the program design, the data is kept in the program using a 16-bit variable for the first 14 bits and two 8-bit variables for another 6 bits as described.

As this project requires two control acts, which are to turn the compressor on and to turn the compressor off. To turn the compressor on, the corresponding signal is 01101001110001010111000000000000000000000000. To the compressor off, the corresponding signal is 01101001110001110110000000000000000000000000. Once the data pattern and its binary data are recognized, they can be written into C code so that the microcontroller can generate them online as described in section 3.7.

4.1.2 Signal Imitation

As mentioned in Chapter 3, the microcontroller has to make its interrupts collaborate with its hardware modules. Thus software simulation results cannot assure the working state of the written codes. Programming to let each of them work is not a problem, but to let them collaborate with each other requires experiences. Thus experiments to confirm their collaboration are conducted during the program design stage. However, most of them are omitted in this thesis which only the final results are shown in this section.

Once microcontroller modules and collaborations are worked as designed, the remote data signal generator program as designed in Figure 3.10 is tested. Data signals to turn the compressor on and off as mentioned in the previous sub-section are generated by the microcontroller and transmitted using the circuit in Figure 3.4. An IR demodulator module v4838 is used to receive the data signal to verify the correctness instead of the air conditioner. Figure 4.7 and 4.8 show the microcontroller's generated signals of IR

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

commands used in this project for turning the compressor on and off respectively. The first channel or the upper signal in the figures is measured from v4838 while the second channel or the lower signal is the signal generated by the microcontroller measured at its pin. It is shown that the program can generate the signal as designed. Figure 4.7 shows that the turn compressor on command air conditioner remote data signals of 01101001110001010111000000000000000000000000 can be generated correctly. Similarly, Figure 4.8 depicts that the turn compressor off command data signals of 01101001110001110110000000000000000000000000 can also be properly generated. After that, both signals are tested to control the air conditioner which can order the air conditioner to work as desired. However, the air conditioner's wireless controller's states as appeared on the panel are not presented in this report.

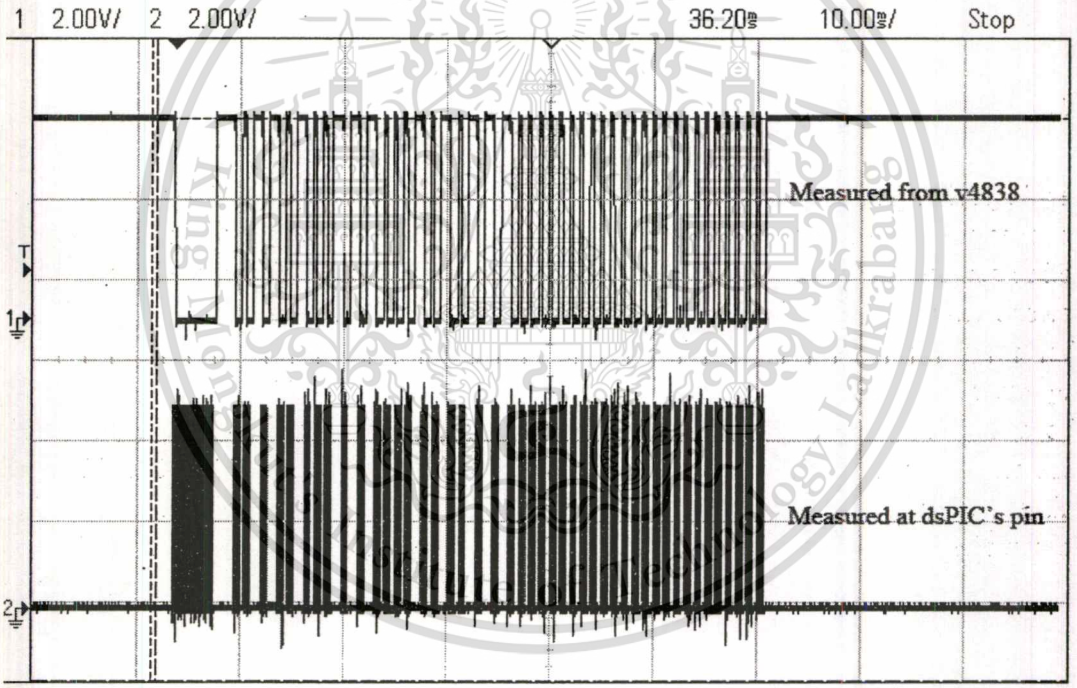


Figure 4.7 Microcontroller generated signal for turning compressor on.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

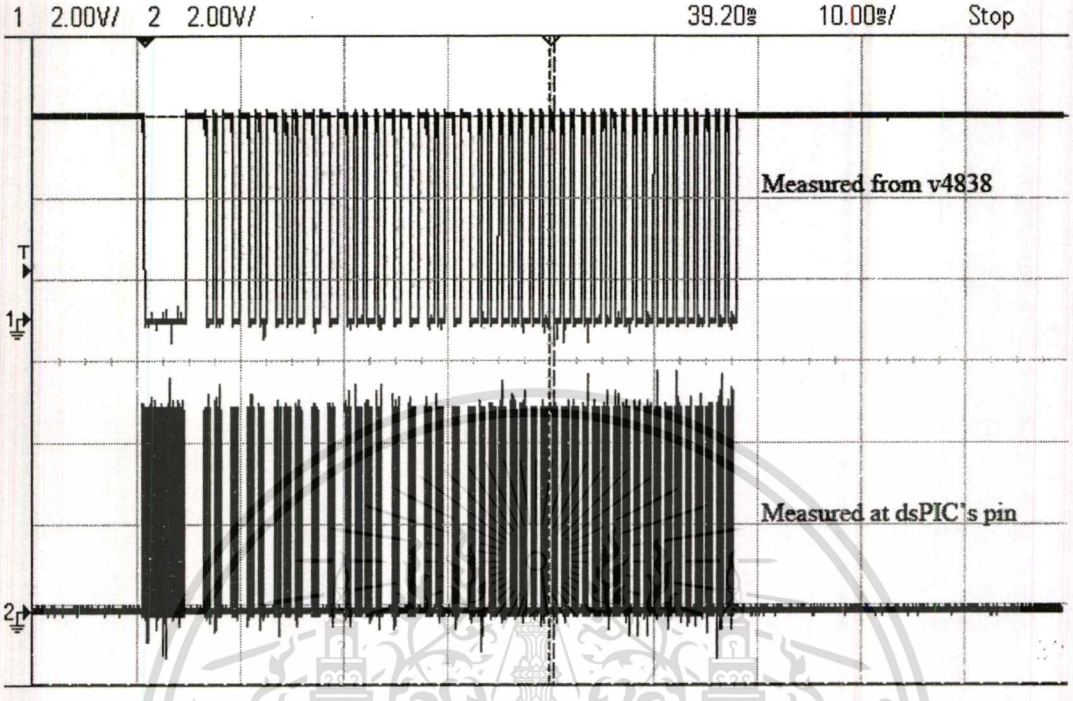


Figure 4.8 Microcontroller generated signal for turning compressor off.

4.2 Model Identification Process

As described in Chapter 2, a mathematical model of a system is needed in order to design a control system or a prediction system. This section presents experimental results in model identification process for the room thermal system. This section is organized into two parts; model validation, and parameters identification. Model Validation section shows the characteristic of the system through experimental results. Once a model is verified to match with the actual result, its parameters have to be identified. Parameters Identification section shows experimental results in parameters identification and its resulting final numerical model which is used in Chapter 3 to design the control system and prediction program.

4.2.1 Model Validation

This section shows the experiments to observe the dynamic of the system that is required for the study of a suitable model to be used for the controller's parameters tuning and prediction program design. Although complex models can give more detailed explanations, simpler models are easier for parameters identification. Also, using a too complex model in this project is not suit for practical ways to apply this work to other air conditioning systems. Simpler models with a lot of approximations will just take the user

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

to implement the same identification method again to apply this work with a new room or a new system. While the same design procedure and programs can be used in general.

Before explaining more details about the selection of the model, experiments from turning the air conditioner on and off are conducted to observe the dynamic of the system. The results are illustrated in Figure 4.9 through 4.12.

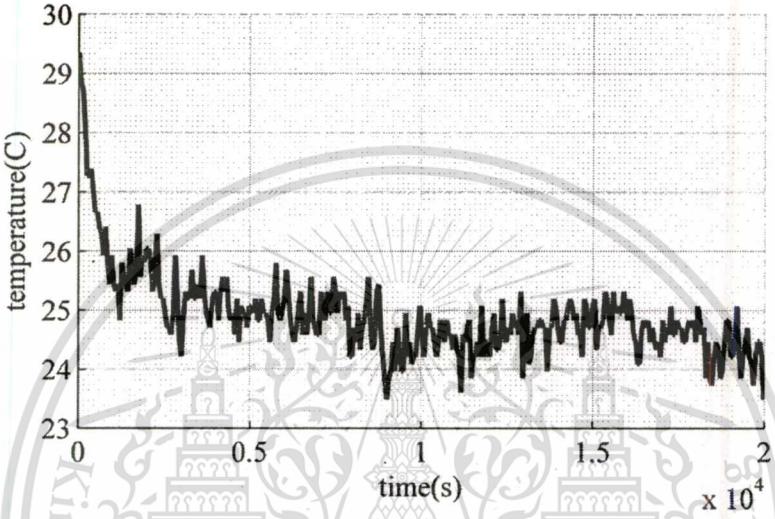


Figure 4.9 Room temperature using air conditioner for five hours and a half.

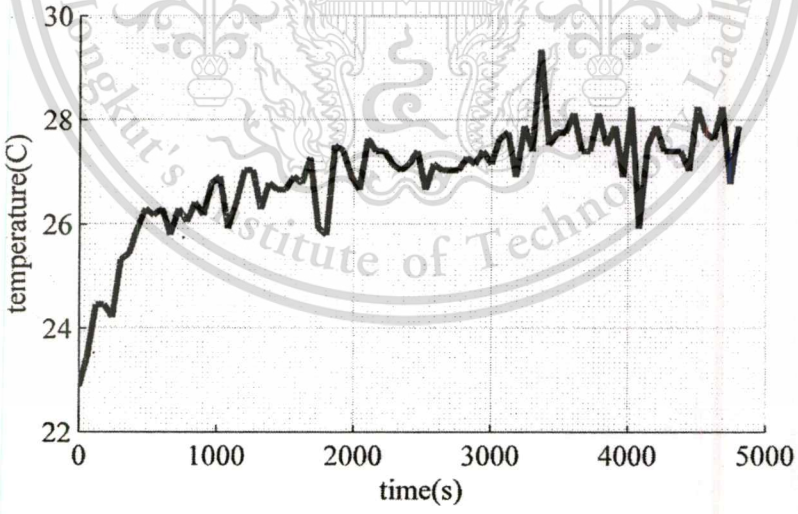


Figure 4.10 Temperature curve after turned the air conditioner off in the first experiment.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

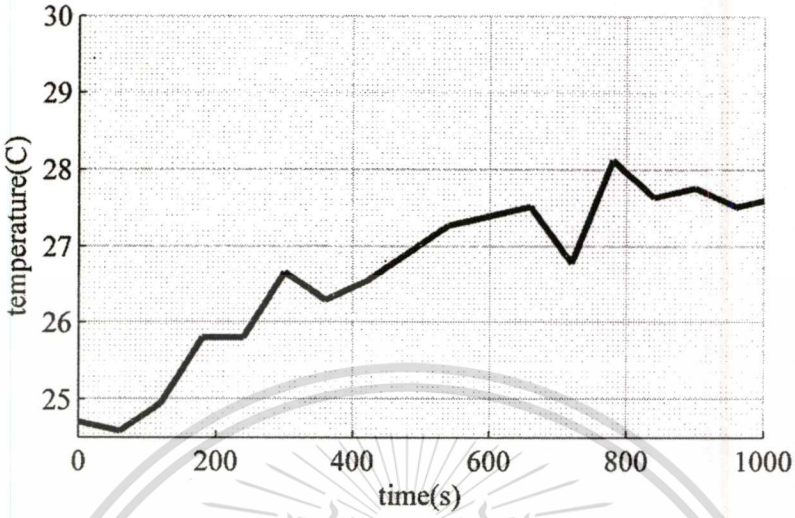


Figure 4.11 Temperature curve after shutting down the air conditioner in the second experiment.

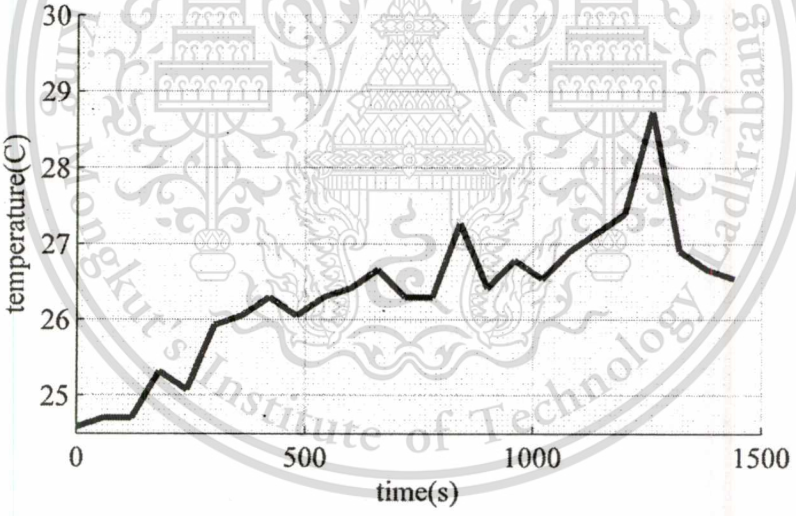


Figure 4.12 Temperature curve after shutting down the air conditioner in the third experiment.

Figure 4.9 shows the temperature curve from constantly turning the air conditioner on five hours and a half with its compressor turned on for the whole time. Figure 4.10, 4.11, and 4.12 show the room temperature after turning air conditioner off in different time length and situations. It can be noticed that the room temperature is converged to some point that is not the ambient (outside) temperature. This is due to the effect of sheltering. The building itself provides heat resistance and capacitance. Due to

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

their curve trends of exponential, it can clearly be noticed from these figures that a simple model based on Newton's law of cooling and heating as described in section 2.1.1 can crudely describe their characteristics.

From the Newton's law of cooling and heating, equation (4.1), modified from equation (2.1), is used to find the constant parameter α of each experiment by using linear least square method. The ambient temperature T_a in equation (2.1) is replaced by the approximated final value T_∞ from each experiment.

$$\begin{aligned}
&(T-T_\infty)=(T_0-T_\infty)e^{-\alpha t},\\
&\alpha t=\ln(|T_0-T_\infty|)-\ln(|T-T_\infty|),\\
&(t^T t)\alpha=t^T(\ln(|T_0-T_\infty|)-\ln(|T-T_\infty|)),\\
&\alpha=(t^T t)^{-1}t^T(\ln(|T_0-T_\infty|)-\ln(|T-T_\infty|)).
\end{aligned} \tag{4.1}$$

Figure 4.13 through 4.16 show the comparisons between the actual experimental results and the simulation results from its paired model in equation (2.1).

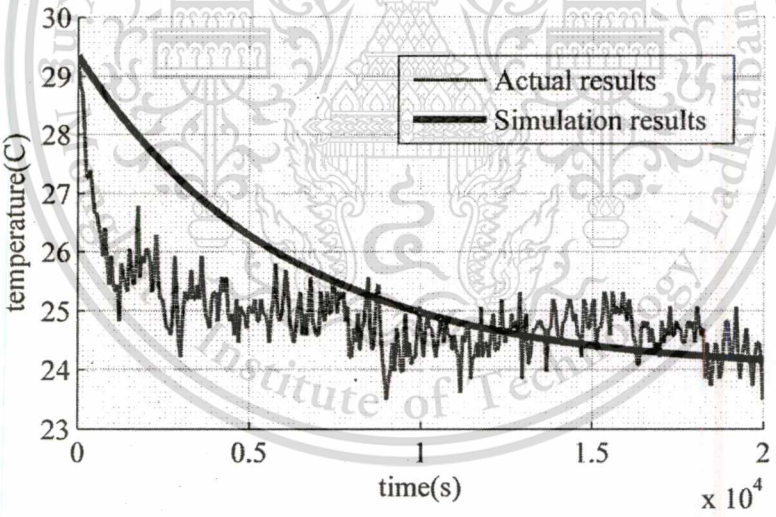


Figure 4.13 Actual results from Figure 4.9 compare with its fitted curve with $\alpha=1.7045e-4$.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

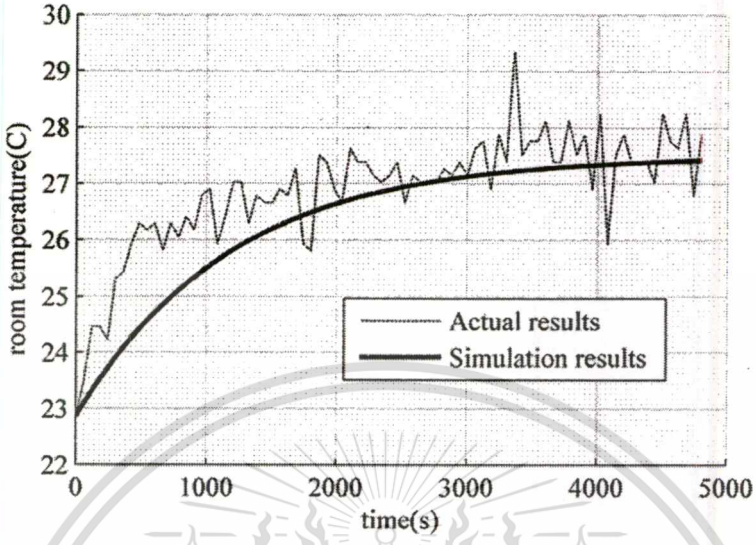


Figure 4.14 Actual results from Figure 4.10 compare with its fitted curve with $\alpha = 8.4520e-4$.

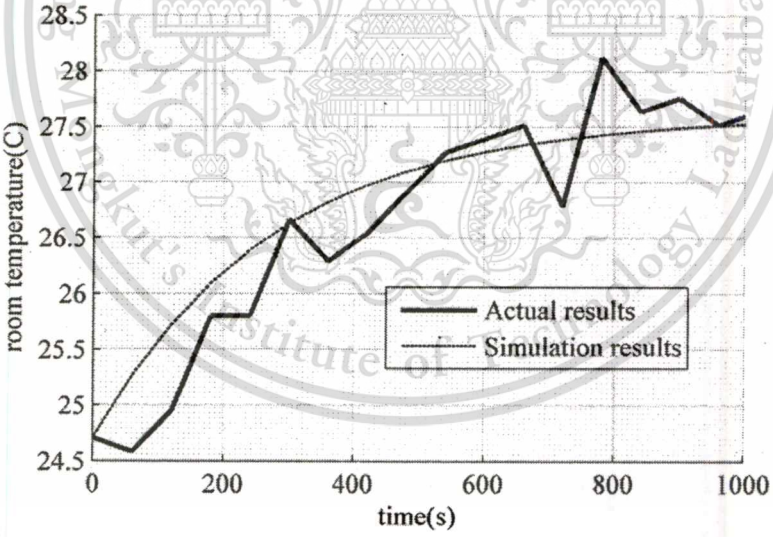


Figure 4.15 Actual results from Figure 4.11 compare with its fitted curve with $\alpha = 0.0036$.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

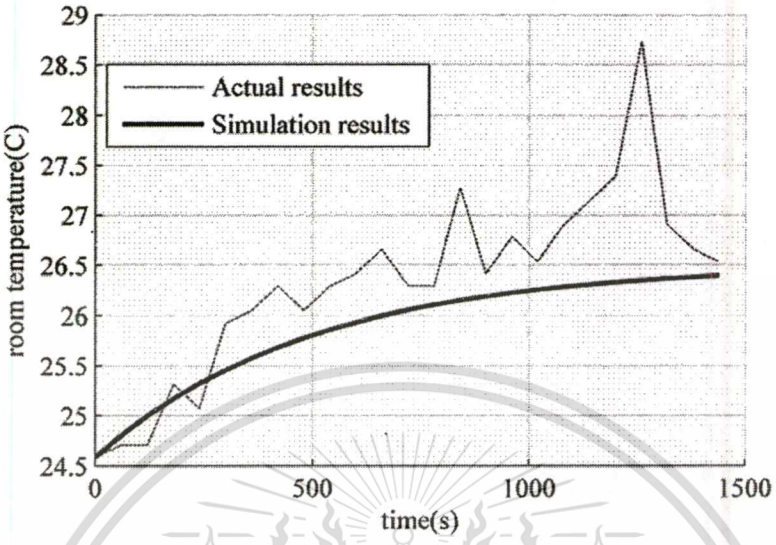


Figure 4.16 Actual results from Figure 4.12 compare with its fitted curve with $\alpha = 0.0020$.

4.2.2 Parameters Identification

Once a model is verified that it can be used to describe the system's characteristics, its parameters have to be identified. This subsection presents results from parameters identification process along with its analysis using theories from Chapter 2. The resulting numerical model in this subsection is used to design a PID control parameters and a prediction program as stated in Chapter 3.

Six experiments are conducted to identify the model parameters. Each of them is conducted in different time and situations. Each experiment's input in this subsection is randomized using MATLAB function with uniform distribution. The first four experiments' inputs are randomized between 10% and 50% duty cycle. The other two inputs are randomized between 10% and 80% duty cycle. Figure 4.17 through 4.24 show the first four experimental results with their inputs and Figure 4.25 through 4.28 show the other two experimental results and their inputs.

In Figure 4.18, some significant drop in temperature value can be noticed during the time 1.7e4 through 1.8e4 seconds and 3.3e4 through 3.4e4 seconds, this is due to the command error problem. Sometimes the command signal cannot reach the air conditioner and it is hard to notice. However, the command can reach the air conditioner for most of the time in the experiments and the mentioned problem occurs no longer than 2 periods of signal. Other experiments in this section do not face this problem as obvious as this experiment. As linear least square is not sensitive in experimental faults, this problem is ignored throughout this thesis.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

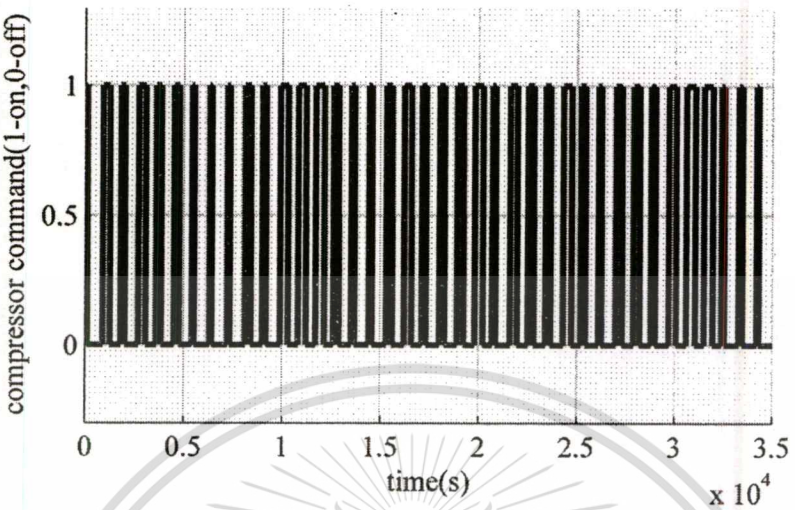


Figure 4.17 Randomized input for the first parameters identification experiment.

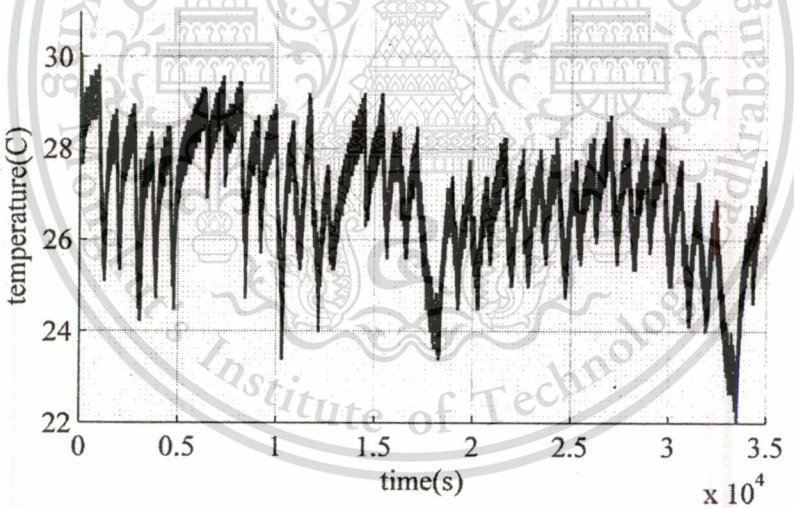


Figure 4.18 Temperature output of command signal in Figure 4.17.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

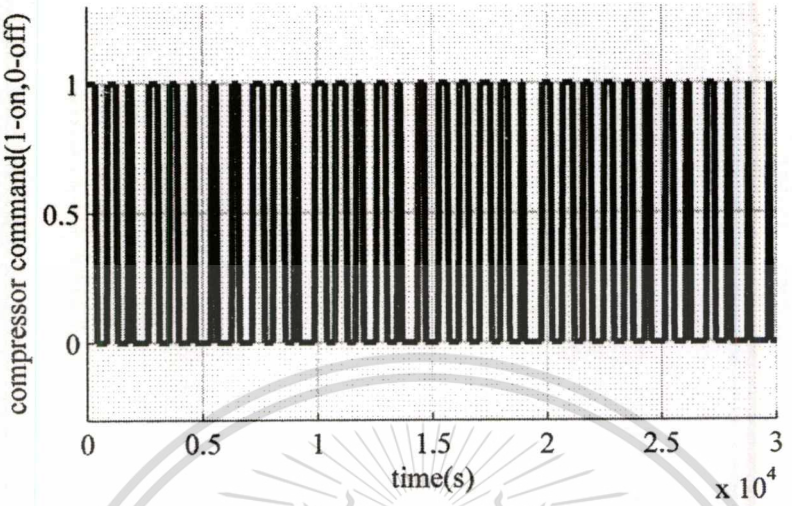


Figure 4.19 Randomized input for the second parameters identification experiment.

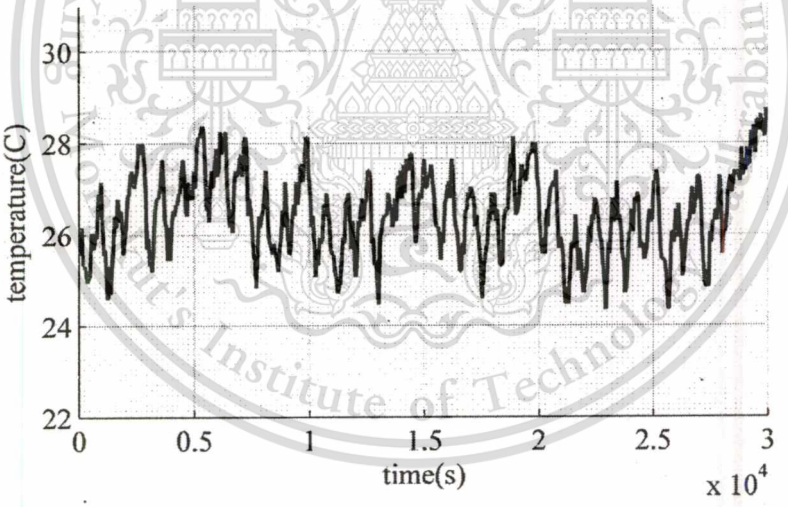


Figure 4.20 Temperature output of command signal in Figure 4.19.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

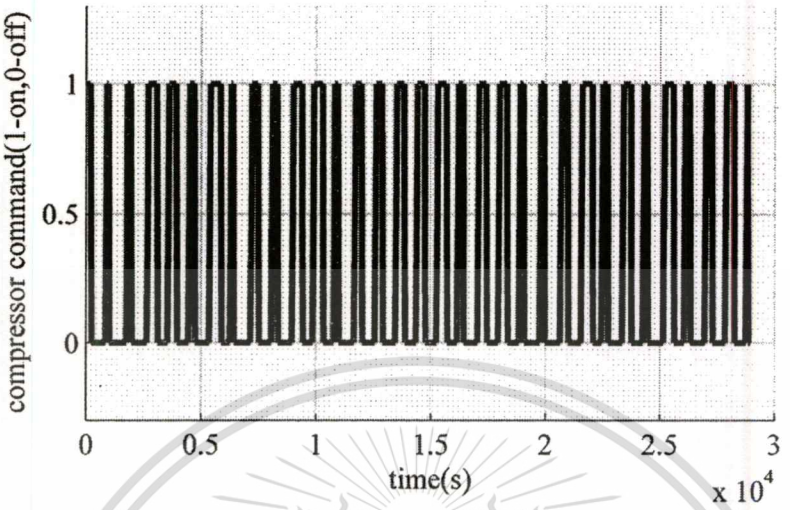


Figure 4.21 Randomized input for the third parameters identification experiment.

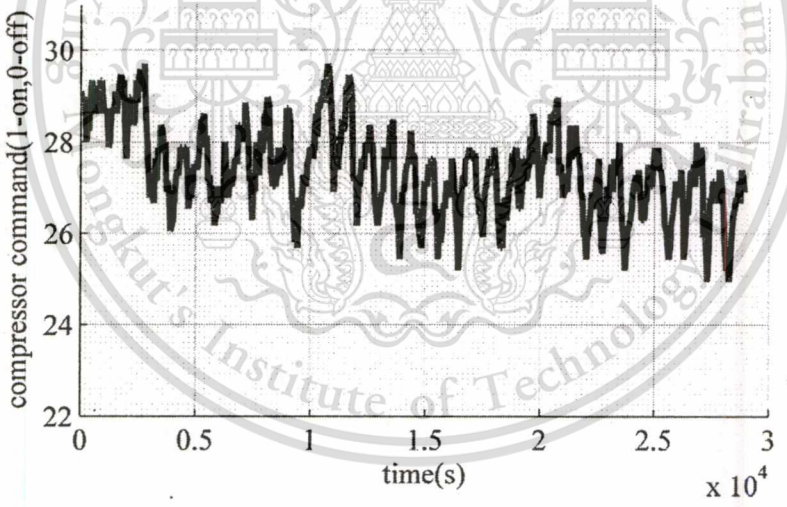


Figure 4.22 Temperature output of command signal in Figure 4.21.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

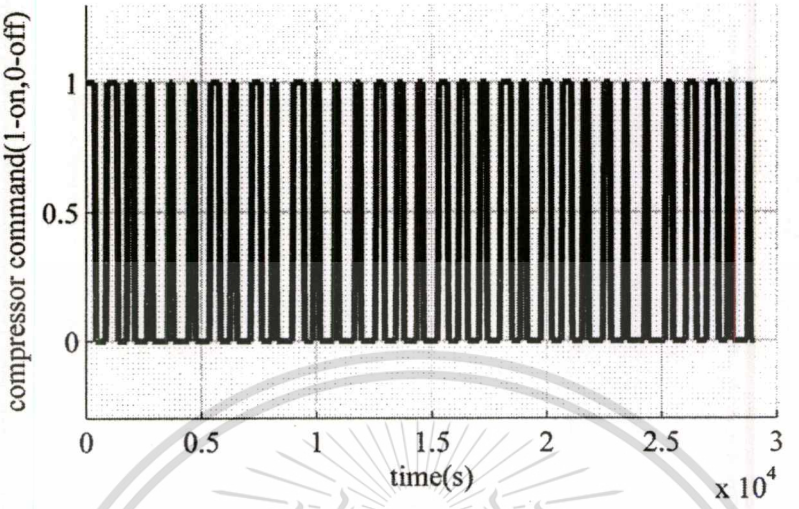


Figure 4.23 Randomized input for the fourth parameters identification experiment.

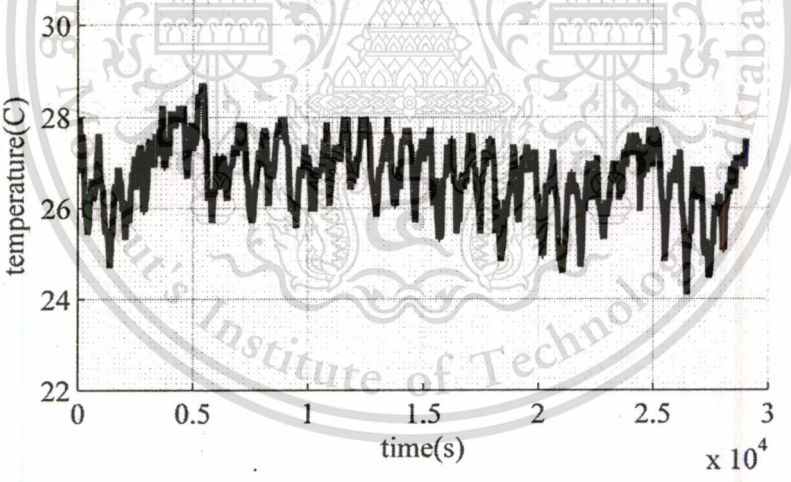


Figure 4.24 Temperature output of command signal in Figure 4.23.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

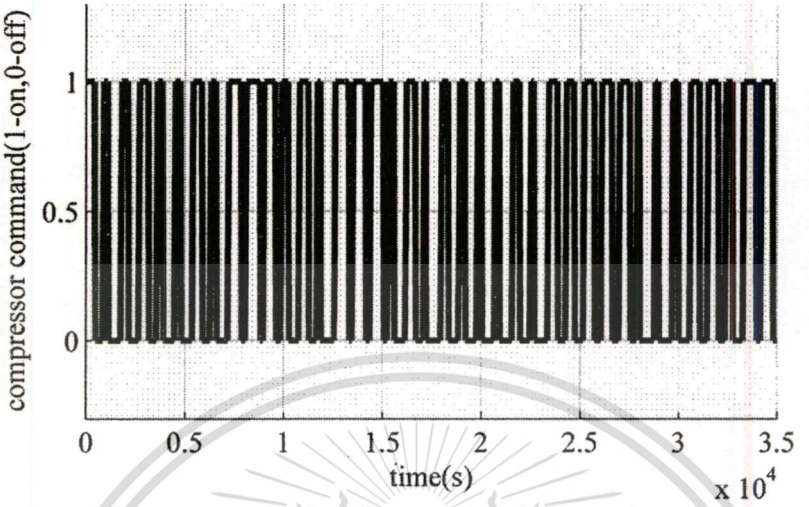


Figure 4.25 Randomized input for the fifth parameters identification experiment.

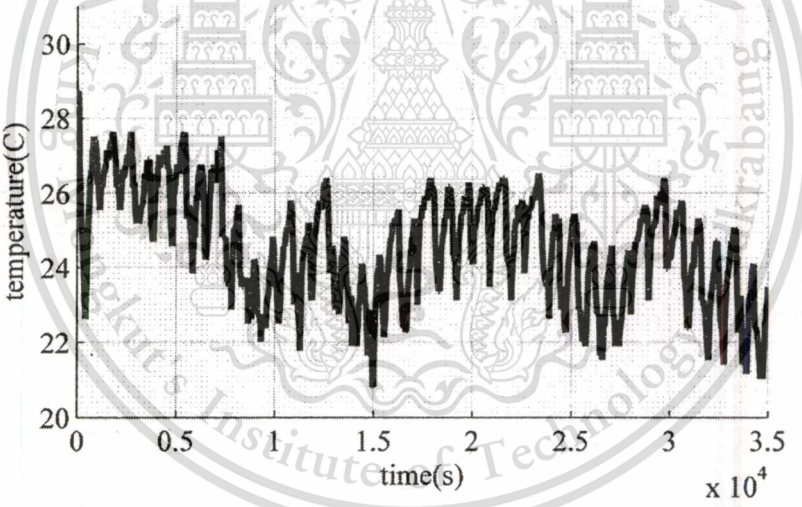


Figure 4.26 Temperature output of command signal in Figure 4.25.

With experimental data from the experiments in this subsection, parameters from each experiment can be approximated using linear least square described in section 2.1.2. Table 4.2 shows parameters from each experiment along with their deviations from simulations. Lastly, the numerical models from all of these experiments are averaged and the average model is used in the design procedure. Figure 4.29 through 4.34 show each experimental value at sampling period of 15 minutes and its respective simulation results from its fitted model and average model.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

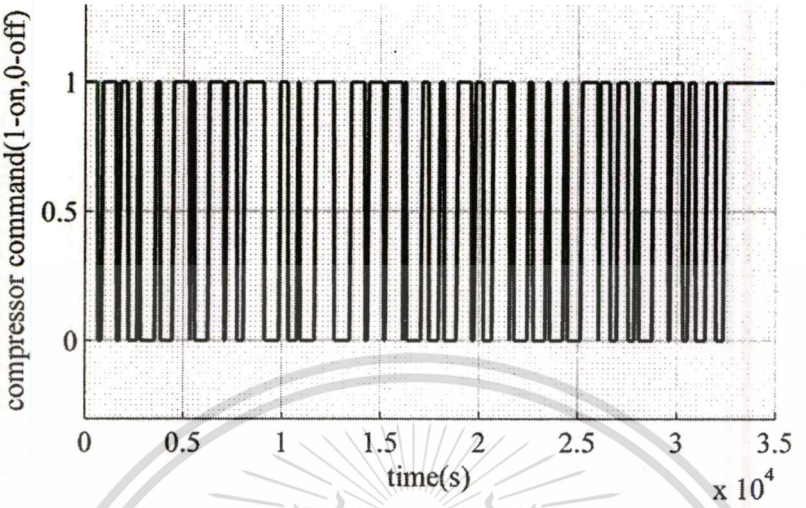


Figure 4.27 Randomized input for the sixth parameters identification experiment.

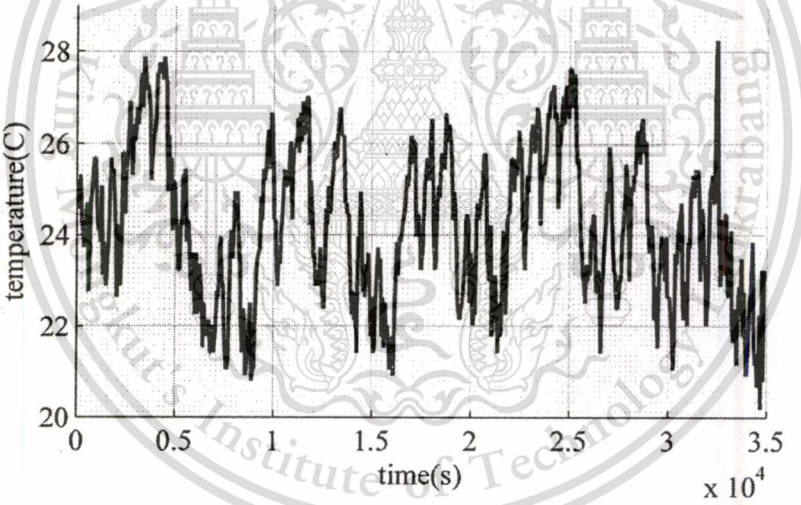


Figure 4.28 Temperature output of command signal in Figure 4.28.

Table 4.2 Identified parameters from experimentations and its deviations.

Experiment no.	α_n	β_n	Q	MAD	MSD	RMSD
1	0.5128	-2.1361	14.0566	0.7201	1.3881	1.1782
2	0.2597	-2.8735	21.1715	0.3208	0.1575	0.3968
3	0.4749	-2.6971	15.4864	0.3935	0.3134	0.5598
4	0.4641	-2.4542	15.4201	0.3604	0.1885	0.4342
5	0.6742	-3.4729	9.7682	0.5698	0.4145	0.6438
6	0.3125	-3.8562	19.3355	0.6278	0.8792	0.9377
Average	0.4497	-2.9150	15.8730	-	-	-

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

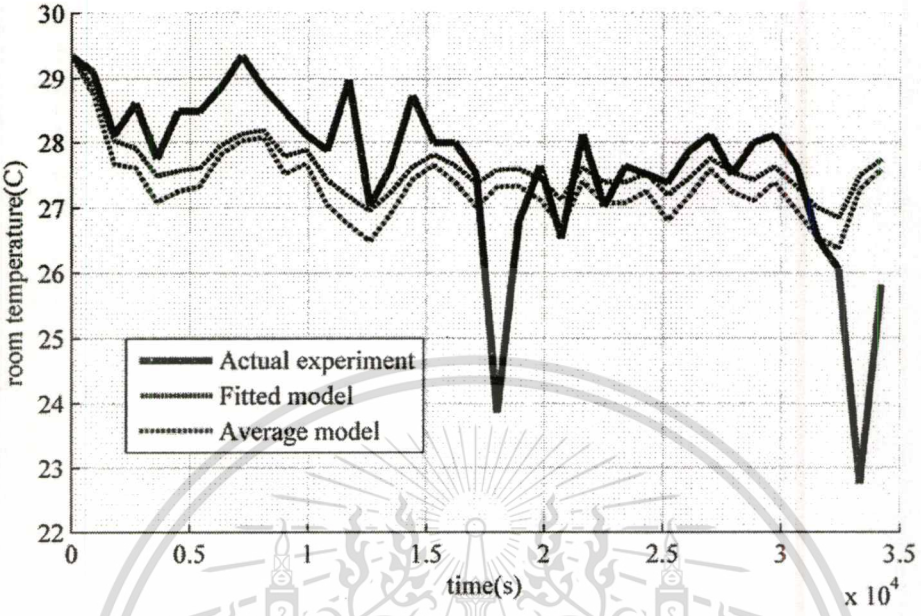


Figure 4.29 Temperature outputs comparison from the first experiment.

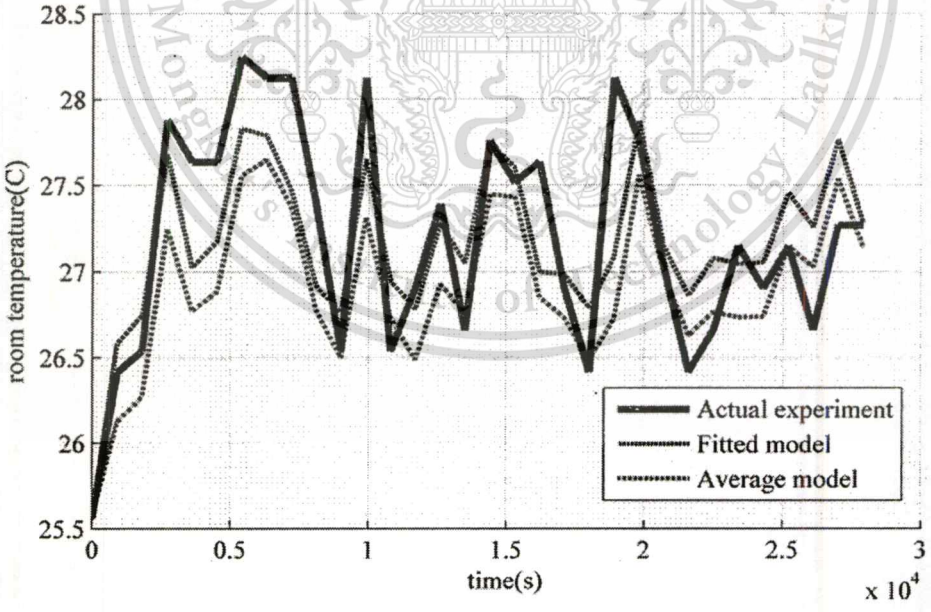


Figure 4.30 Comparing simulation results with its true values using PWM input from Figure 4.19.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

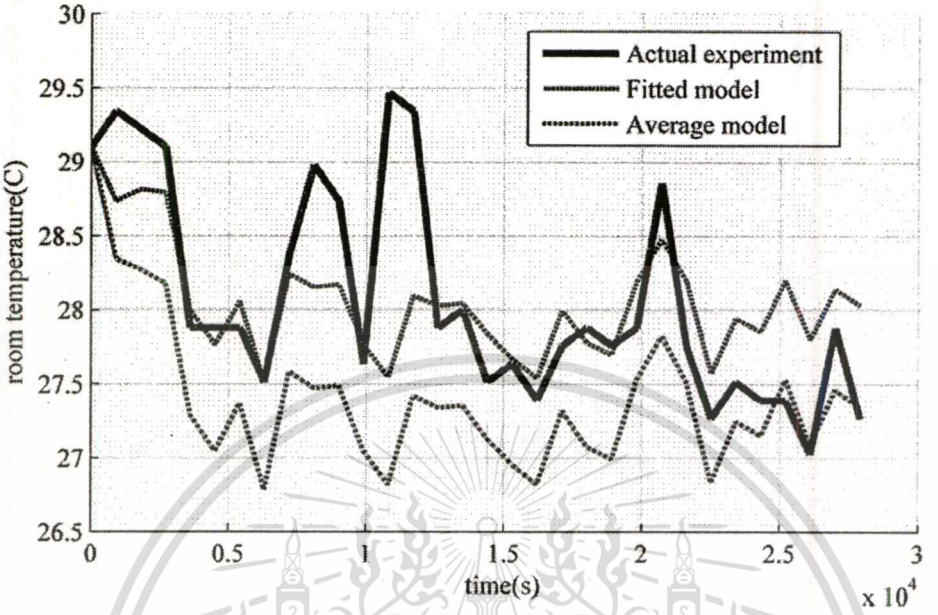


Figure 4.31 Comparing simulation results with its true values using PWM input from Figure 4.21.

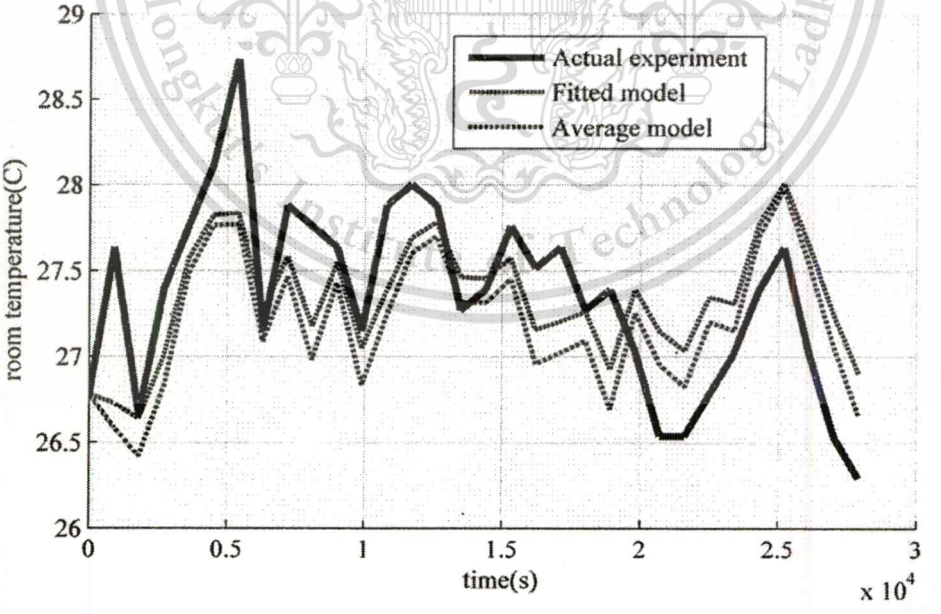


Figure 4.32 Comparing simulation results with its true values using PWM input from Figure 4.23.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

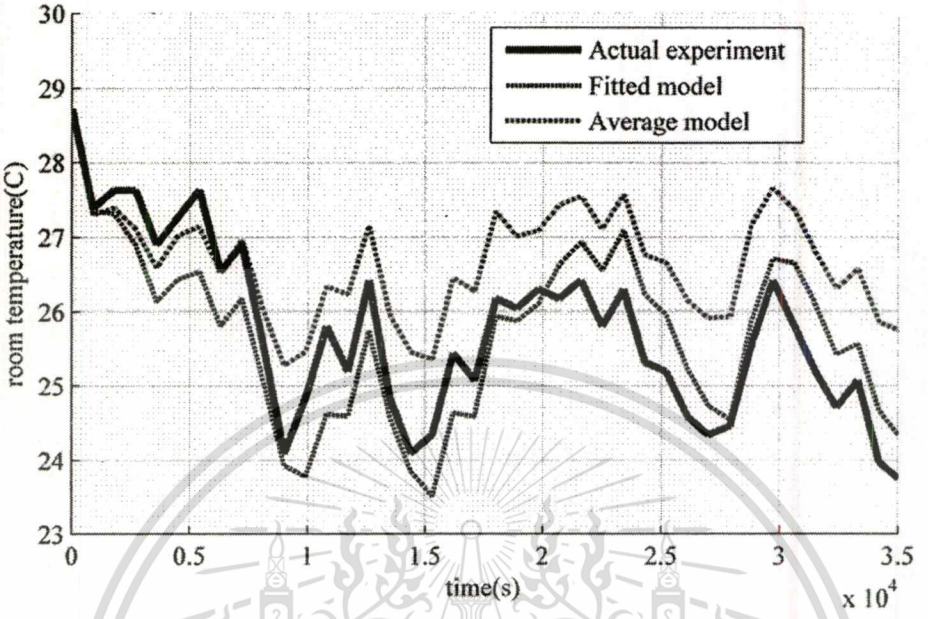


Figure 4.33 Comparing simulation results with its true values using PWM input from Figure 4.25.

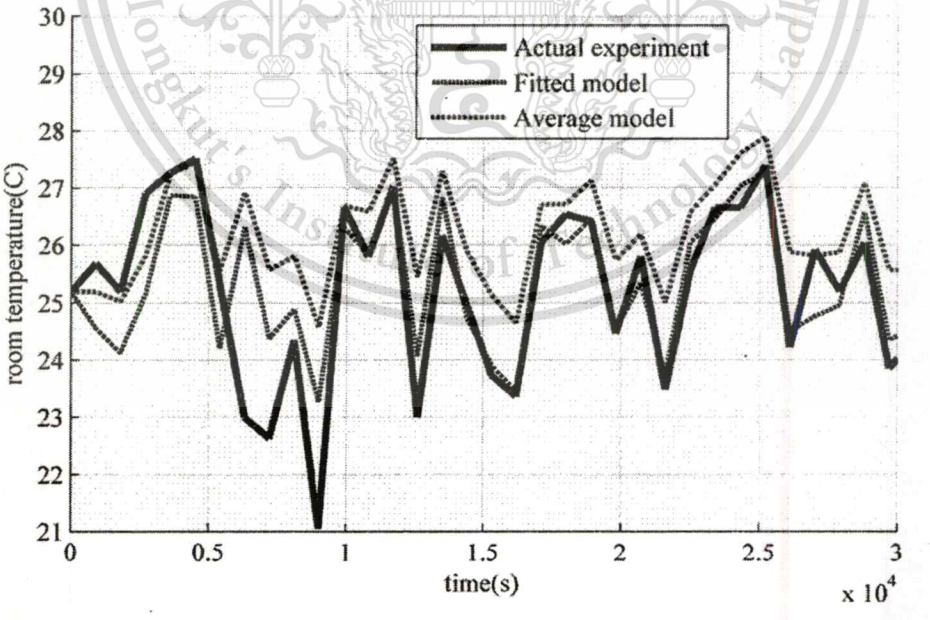


Figure 4.34 Comparing simulation results with its true values using PWM input from Figure 4.27.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Table 4.3 Average model temperature output deviations from the experiments.

Deviation type	Experiment no.					
	1	2	3	4	5	6
MAD	0.8673	0.3776	0.6969	0.3893	1.0179	1.0682
MSD	1.5015	0.2452	0.7780	0.2136	1.3320	1.9922
RMSD	1.2253	0.4952	0.8821	0.4622	1.1541	1.4114

Table 4.3 shows that the average model simulated deviations are not much differed from that of the approximated models from each experiment in Table 4.2. Thus this model can be used as the representative of the system for the design procedures in Chapter 3.

4.3 Control Results

This section presents the experimental results of the room thermal control systems using 2 control schemes, which are on-off control and PI control.

4.3.1 On-off Control Results

This section shows experimental results using the on-off controller to regulate room temperature at a desired value. These experiments are conducted to validate microcontroller interaction with its actuator, the compressor. Its command inputs and temperature results are to be compared with the results of PI control in section 4.5.

Figure 4.35 and Figure 4.36 show the first experimental results from applying the on-off controller to perform temperature regulation on the system with the set point temperature at 23C. Figure 4.36 shows that the microcontroller can command the air conditioner to work properly. The temperature is lowering when the controller turns the compressor on and vice versa. This regulation result also shows that even though the hardware of the system can handle a regulation at 23C, the compressor have to work constantly for the most time which will obviously reduce its lifespan.

The desired temperature in the second experiment is programmed to be adjustable between two levels. The controller in the first 3000 seconds regulates the temperature with a desired temperature at 23C. Then the controller regulates the temperature at 25C, which is also used in the proposed PID control system in the next subsection.

The other experiment to have the system work on two different set point temperature values is conducted. In this experiment the desired temperature value is 23C until around 3000 seconds, and it is changed to 25C afterwards. Figure 4.37 shows that the hardware can regulate the system with its set point changed to 25C. Figure 4.38 illustrates the resulting temperature of this second experiment. It can be summarized that this control scheme cannot keep the temperature near the hysteresis range as designed overtime. This is due to the hardware mechanics that the effect of cooling and heating are

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

not stopped immediately after turning the compressor off or on, but they will keep continuing on the same effect until the effect run out and going to the opposite direction.

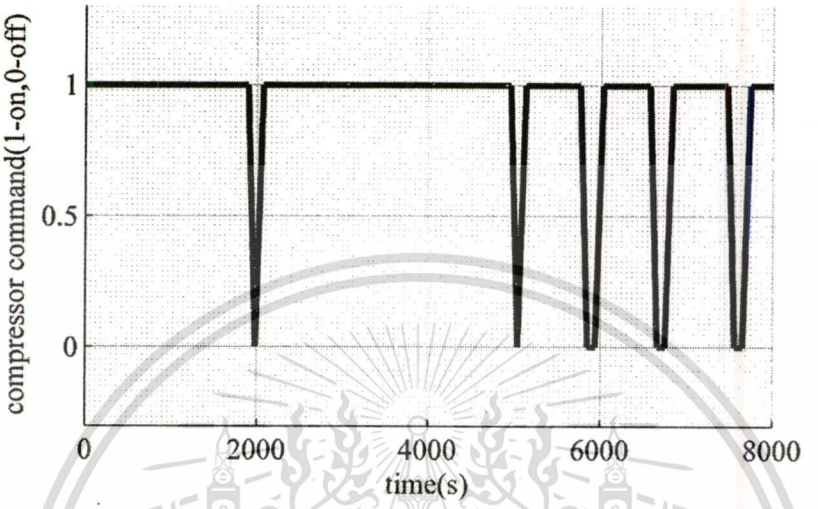


Figure 4.35 Command input driven by on-off controller from the first experiment.

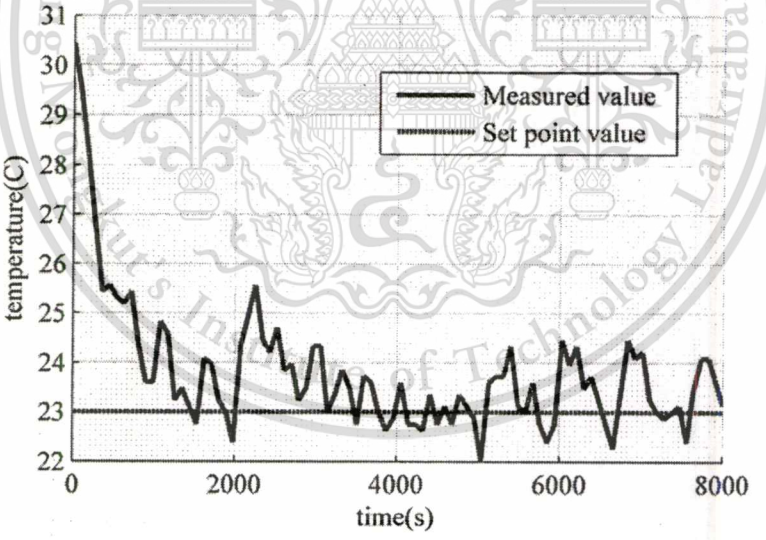


Figure 4.36 Temperature output by on-off controller from the first experiment.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

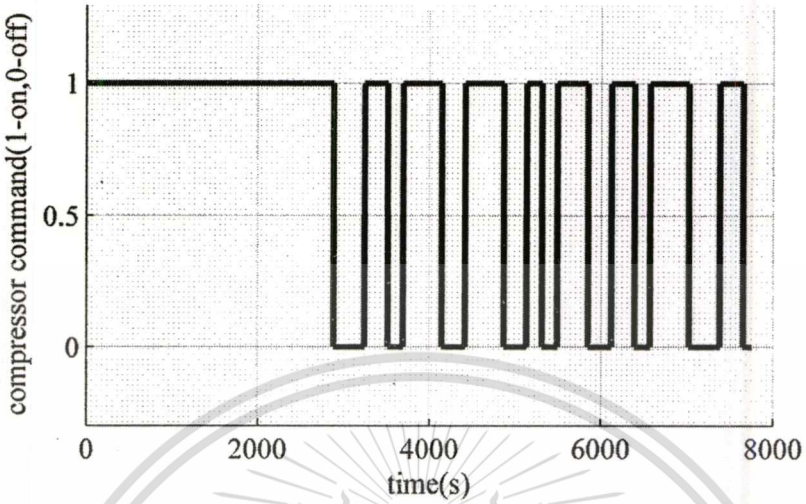


Figure 4.37 Command input driven by on-off controller from the second experiment.

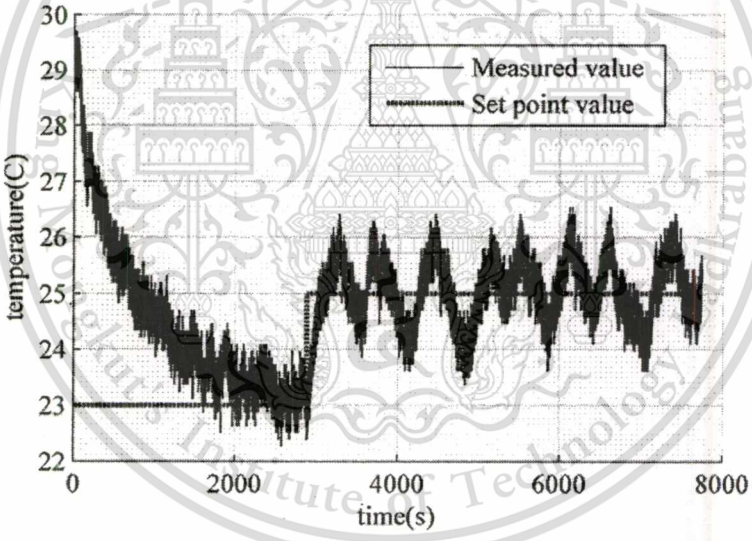


Figure 4.38 Temperature output by on-off controller from the second experiment.

4.3.2 PI Control Results

From PID tuning section 3.9.1 that uses averaged parameters from Table 4.2 and is instructed by Skogestad's rule in Table 2.1 that only proportional and integral controller is required by the approximated model, so the proposed controller in this project is PI controller. This subsection shows the proposed PI control implementation results on the system. The experiments are conducted in the same posture but not the exact situations. Their desired temperature value is 25C as the observation result from the previous subsection that the hardware cannot handle regulation at lower temperature and

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

this is the value that is widely recommended to use to reserve the power consumption. The microcontroller is programmed to perform temperature regulation as illustrated in Figure 3.1. The observed data is the temperature sampled at the sampling rate of a minute, the controller's output, and the compressor working state. Thus, each experiment contains three figures; its input PWM, the compressor working state, and the temperature.

Figure 4.39 through 4.41 show the first experimental results as described. The feedback value is the value sampled by the controller and the measured value is the value logged to see the temperature curve during the sampling period.

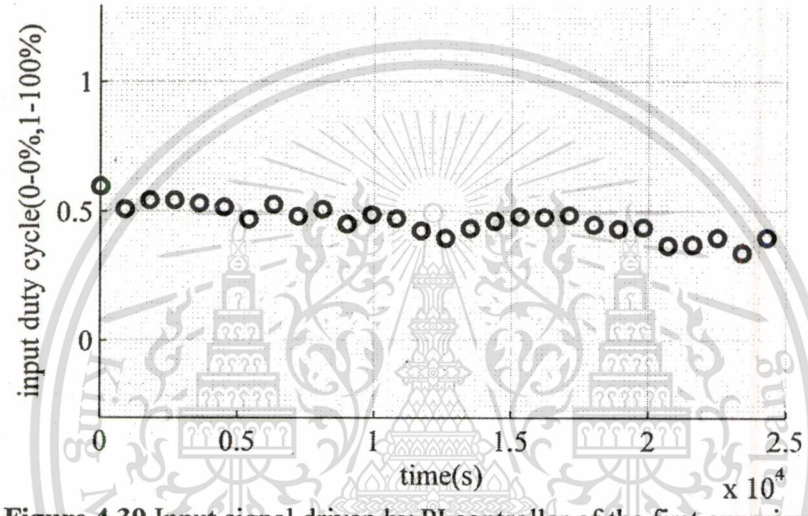


Figure 4.39 Input signal driven by PI controller of the first experiment.

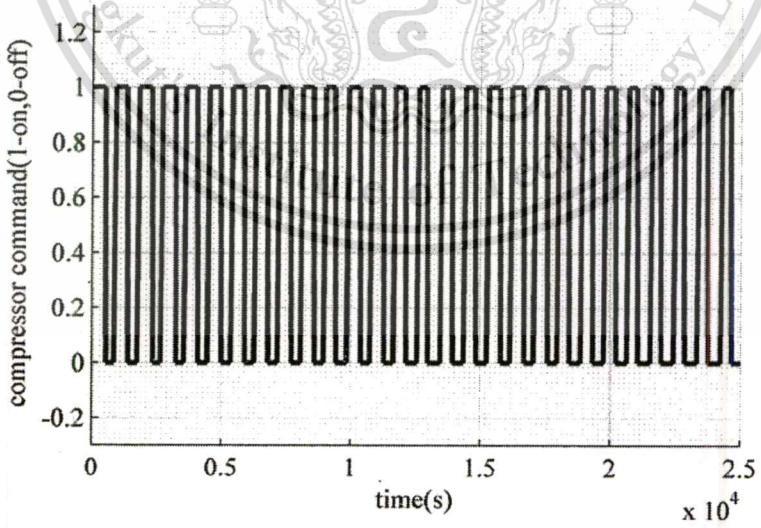


Figure 4.40 Command input driven by PI controller of the first experiment.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

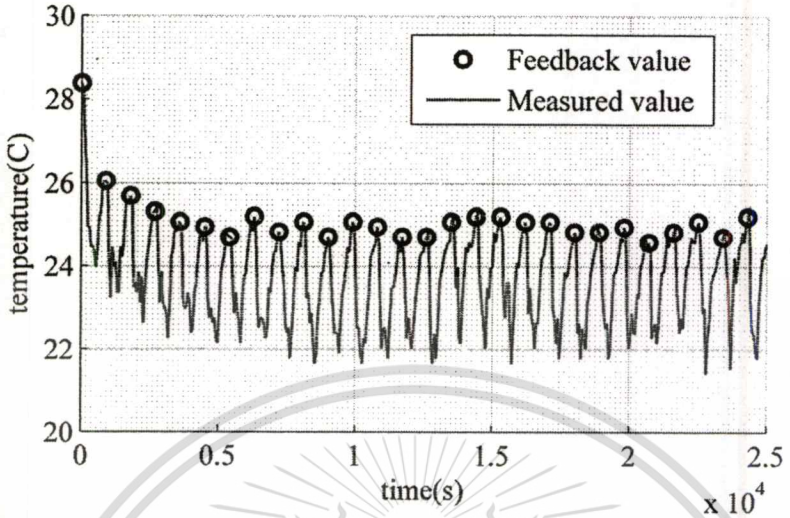


Figure 4.41 Temperature output of PI control system of the first experiment.

In the controller's perspective, this control scheme is better in regulating the temperature because the temperature keeps going on around 25C at the sampling time instant. It is not expected to have a smooth curve because there are too many uncertainties to the system. However, if the measured value every one minute is included in the consideration, it is not as good as it look from the controller's perspective.

In Figure 4.44, the controller faces a command error problem in the experiment. Although the controller can get the temperature back to around the set point values, six periods to get the temperature on track again is too long time for the users and the temperature varied during that time contains is high enough that the user can be inconvenient. This shows that the controller's regulation ability is vulnerable to the command error problem.

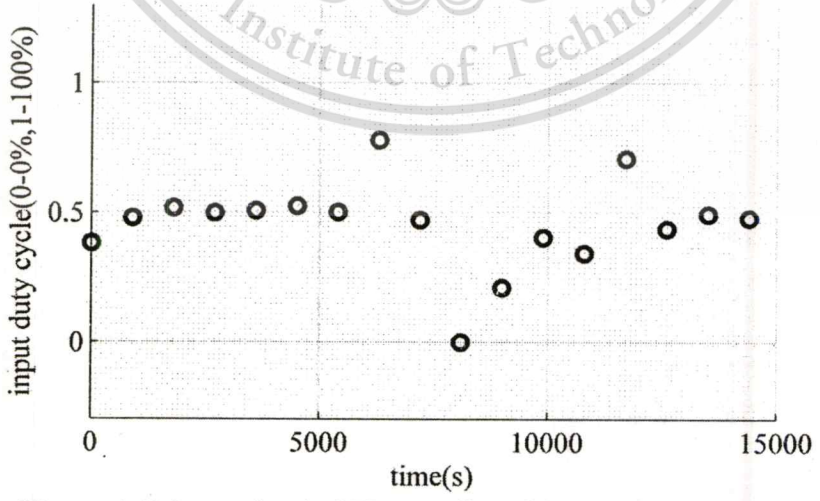


Figure 4.42 Input signal of PI controller of the second experiment.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

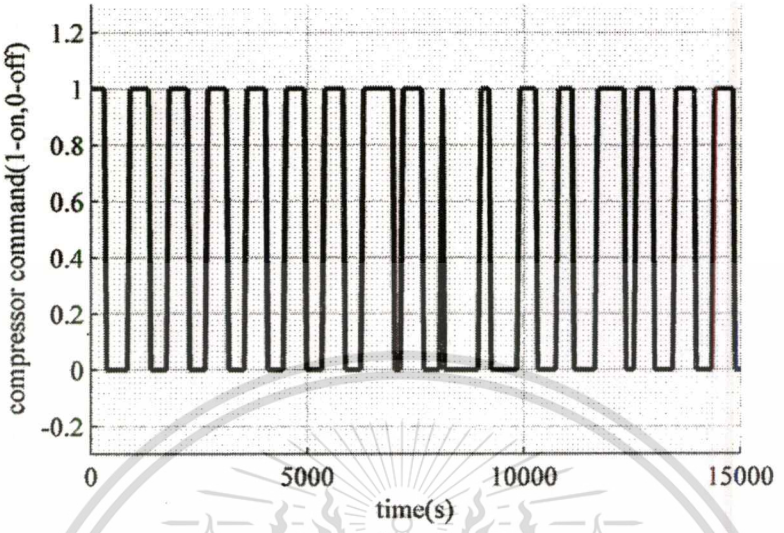


Figure 4.43 Command input driven by PI controller of the second experiment.

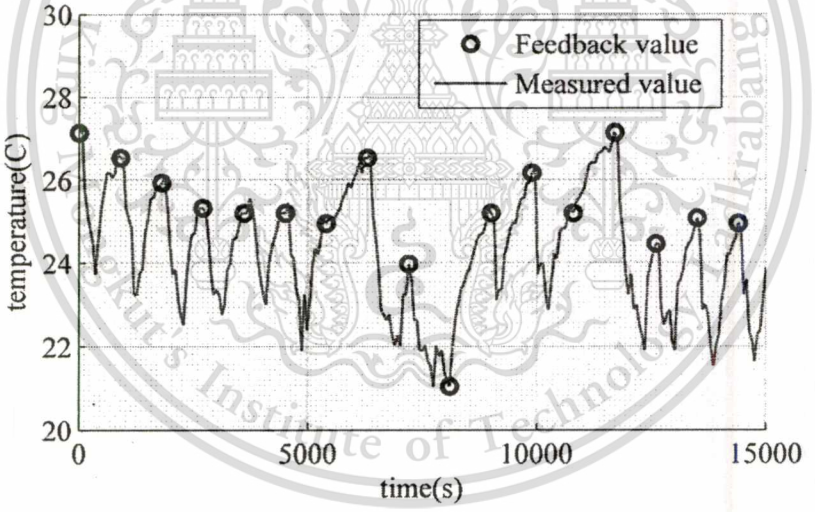


Figure 4.44 Temperature output of PI control system of the second experiment.

4.4 PI Control with Prediction Results

This section presents application of the PI control bundled with the final value predictor program as designed in Chapter 3. Three experiments are conducted in this section with similar situations. However, the first experiment is differed in its acceptable variance; it is set to ±2C from the set point value. The microcontroller is set up in the same posture as in the previous section. It is programmed to send the values of four variables for every minute, which are input's duty cycle, compressor working state,

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

measured temperature value, and predicted final value. A temperature figure in this section contains three lines. Controller's value indicates the controller feedback value for each period. Actual value is the temperature measured using sampling period of a minute. And Predicted value is the value predicted as the temperature value at the end of usage if the program decides to stop control acts at that time.

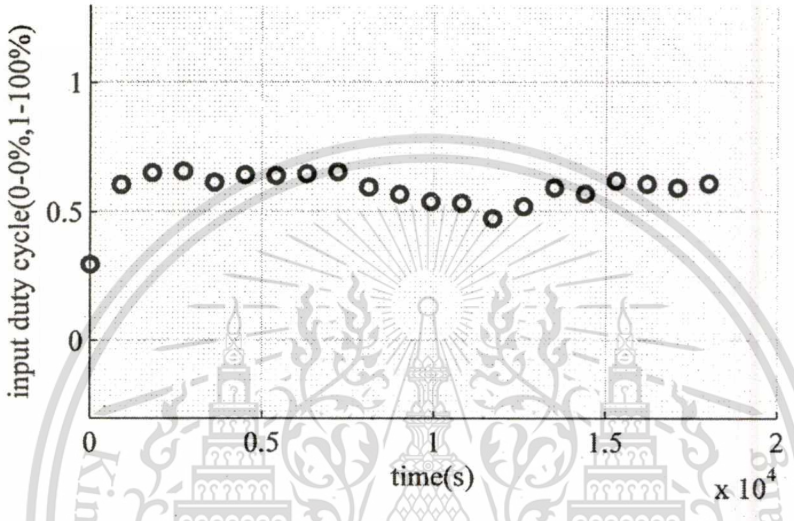


Figure 4.45 Input's duty cycle of the first experiment.

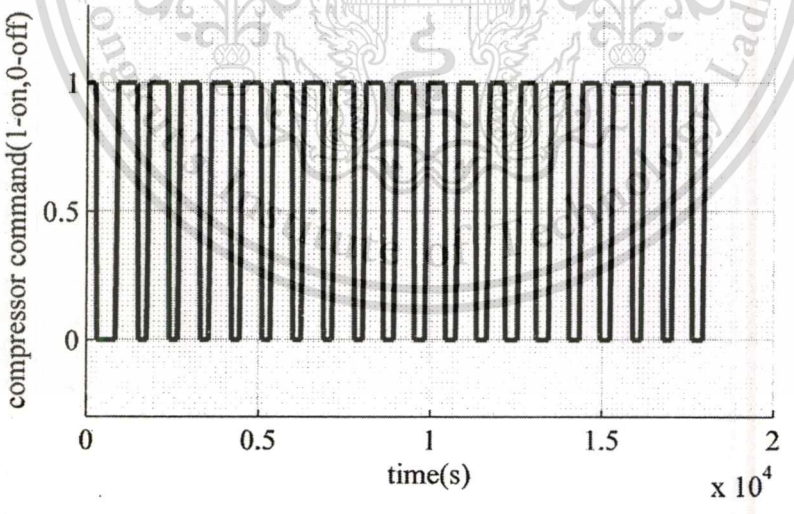


Figure 4.46 Command input to the compressor of the first experiment.

Figure 4.45 through 4.47 show the first attempt results. In this experiment, the controller performs regulation around 25C very well in its perspective, but final value condition is not met. Thus the system is not shut down before end of usage. This

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

experimentation shows that it is possible to predict earlier turn off time with increased final acceptance value. As the main goal in experiments in this section is to measure the prediction ability of the designed prediction program, the other experiments are conducted with the increased final acceptance value of ±2.5C.

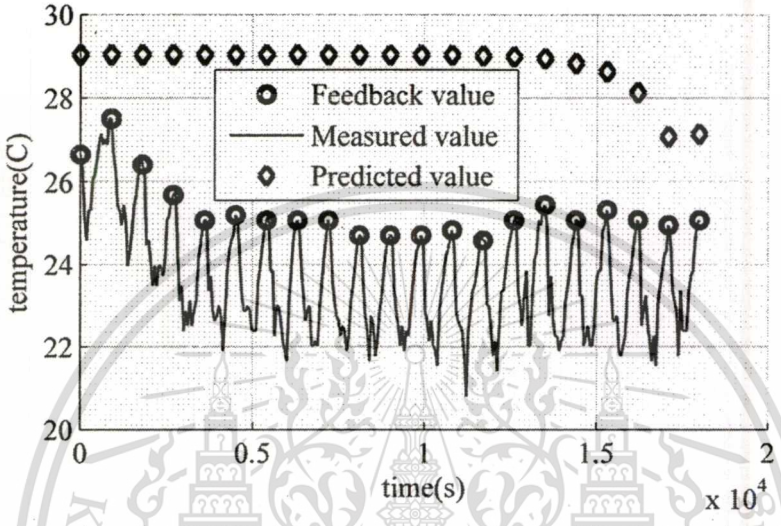


Figure 4.47 Outputs of PI control system with predictor of the first experiment.

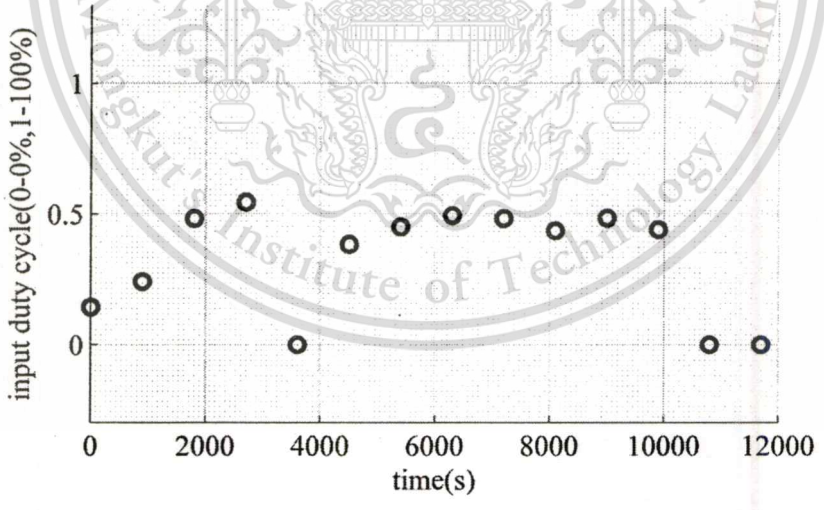


Figure 4.48 Input's duty cycle of the second experiment.

Figure 4.53 through 4.55 show good results in the second experiment. With a more relaxed setting than the first attempt. The system is turned off two periods earlier with temperature value converges to the predicted value. Also, Figure 4.50 shows that there is a command error occurred but the predictor works well along with its existence.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

The third experiment also shows this problem, which is not able to turn on the compressor at a sampling period of 10900s, so the temperature rises for that period but its overall results are good and the problem disappeared once the temperature can be regulated in the next period.

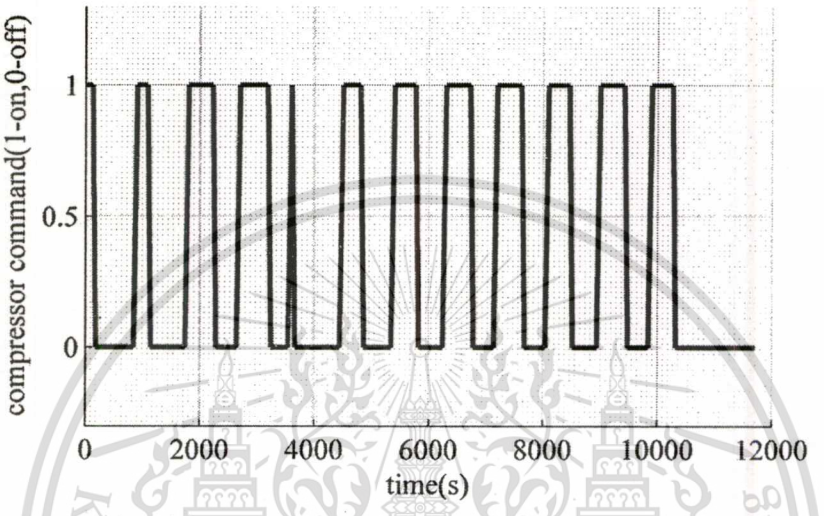


Figure 4.49 Command input to the compressor of the second experiment.

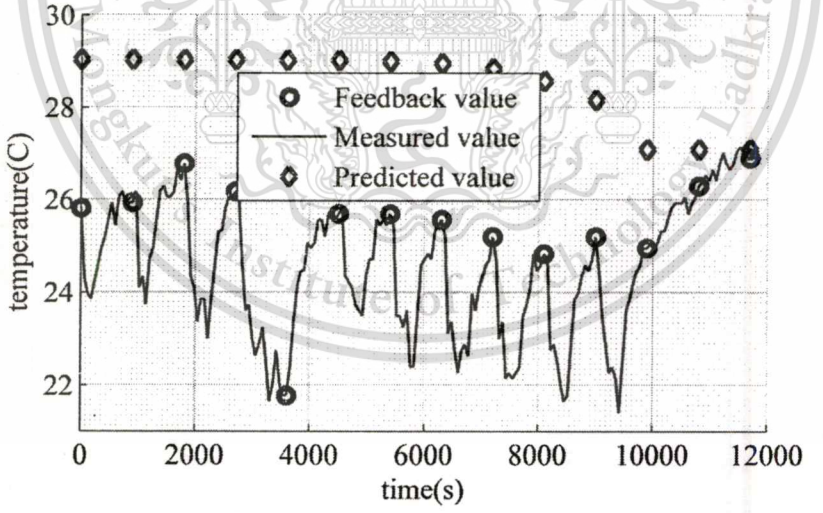


Figure 4.50 Outputs of PI control with predictor of the second experiment.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

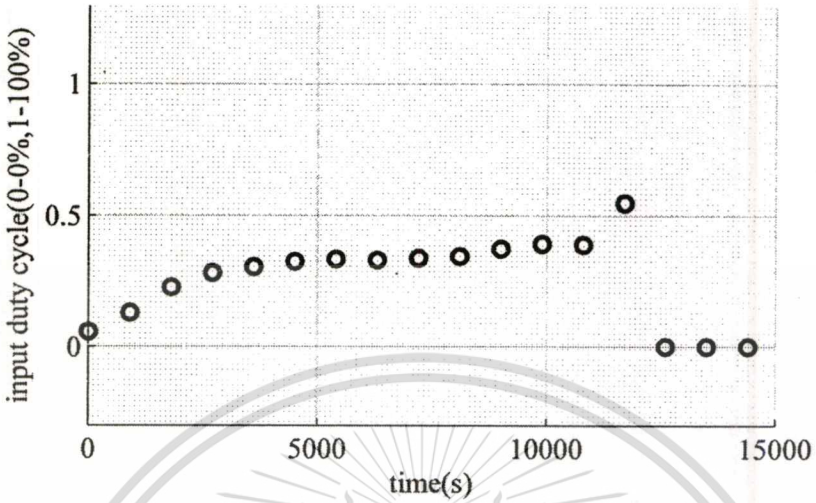


Figure 4.51 Input's duty cycle of the third experiment.

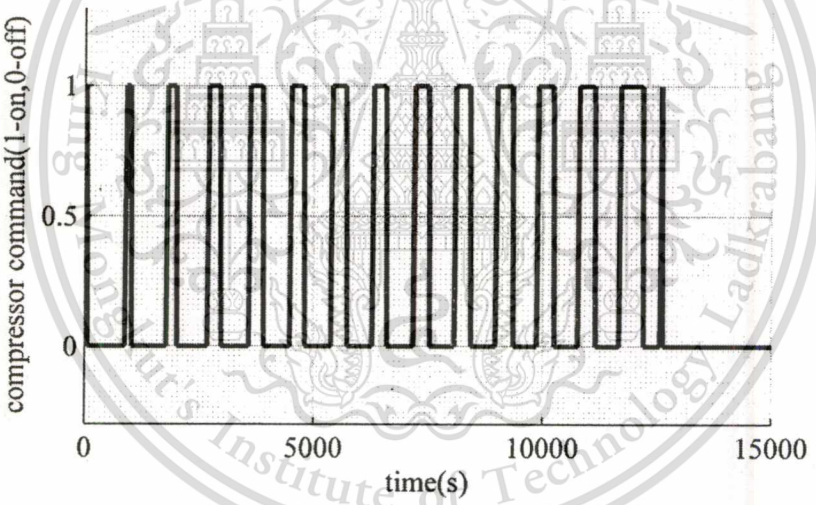


Figure 4.52 Command input to the compressor of the third experiment.

Although the existence of a command error, the predictor still works well and orders the system to shut down controller earlier for two periods, 30 minutes, before the designated end of usage. This experiment is conducted in a cloudy day in which the overall ambient temperature and its effect are lower than the other experiments. Thus the predictor is not as accurate as in the second experiment. Because the model identification experiments are conducted in hotter days and the main problem is the extraction of heat noise and ambient temperature effect, which is extracted as a constant value in this project.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

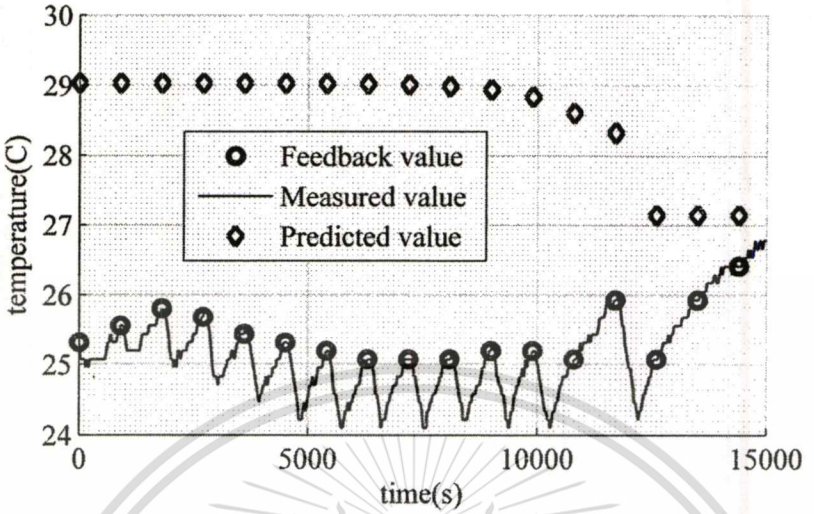


Figure 4.53 Outputs of PI control with predictor of the third experiment.

4.5 Power Consumption Analysis

The input definition for the proposed PI controller is activating compressor every 15 minutes. Thus, there are surges due to compressor's transient characteristics 4 times in an hour. This value from using PI controller is certain, but the number of starting compressor in the on-off control system is not certainly defined. It is plainly due to the system's nature.

The comparison is based on the experimental results under similar situations. The testing period is from noon to 4PM and the ambient temperature is nearly equal around 35C. Comparison starts when the room temperature is around 25C by waiting an hour after starting the experiments and uses the records for three hours afterward. Table 4.3 shows the compressor usage of the on-off controller and PI controller.

Table 4.4 Comparison of compressor usage.

Controllers	Compressor working time (minutes)	Surge (times)
On-off controller	113.4	21
PI controller	87.5275	12

Table 4.3 shows that PI controller can reduce compressor working time by 23.8725 minutes, or 21.0516 percent, and avoid current surges by 9 times, or 42.8571 percent.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Chapter 5

Conclusion

5.1 Conclusion

This thesis presents the application of microcontroller to use PI controller on air conditioner with two positions compressor. The goal is to reduce power consumption of air conditioner while room temperature is maintained in desired value. The final value predictor was added in this project to reduce more energy.

The project's goal was achieved. PI control implementation saved more than 20 percent of compressor's working time in comparison with on-off control while measured temperature was regulated at desired value for three hours. This result conformed to BRITE's [2] that used learning-based MPC instead.

There are drawbacks in implementation of PWM input at 15 minutes period with PI controller as seen in experimental results. First, the controller cannot handle disturbance well. Second, although feedback temperature did stay at desired value, its value during that sampling period did not. These are due to long sampling period of the PWM input.

The final value predictor had proved its ability to save more energy without user concern for half an hour while the measured temperature is still in accepted range.

5.2 Future Works

Although conformed to BRITE's experimental results in a way that saving energy by applying PWM definition and a more complex controller to such system is possible, the percentage of reduced energy is still considerably lower. This is due to the controller's limitation which can be upgraded.

In this project there is a problem prior to manipulating system. Command lost error randomly occurred at times. This must not happen in the real use and should be fixed.

This project cannot show its exact power consumption in comparison because there is no measurement of power consumption during experiments. To appeal users to install this system, the number for power consumption should be added.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Appendix A

Device Datasheets

King Mongkut's Institute of Technology Ladkrabang

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

A.1 Microchip dsPIC30f6014A

MICROCHIP dsPIC30F6011A/6012A/6013A/6014A

dsPIC30F6011A/6012A/6013A/6014A High-Performance Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *"dsPIC30F Family Reference Manual"* (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- Flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- Up to 144 Kbytes on-chip Flash program space
- Up to 48K instruction words
- Up to 8 Kbytes of on-chip data RAM
- Up to 4 Kbytes of nonvolatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPS operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- Up to 41 interrupt sources:
  - 8 user-selectable priority levels
  - 5 external interrupt sources
  - 4 processor traps

DSP Features:

- Dual data fetch
- Modulo and Bit-Reversed modes
- Two 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle:
  - Multiply-Accumulate (MAC) operation
- Single-cycle ±16 shift

Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Five 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions:
- Data Converter Interface (DCI) supports common audio Codec protocols, including I^2S and AC'97
- 3-wire SPI modules (supports 4 Frame modes)
- I^2C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Two addressable UART modules with FIFO buffers
- Two CAN bus modules compliant with CAN 2.0B standard

Analog Features:

- 12-bit Analog-to-Digital Converter (ADC) with:
  - 200 Ksps conversion rate
  - Up to 16 input channels
  - Conversion available during Sleep and Idle
- Programmable Low-Voltage Detection (PLVD)
- Programmable Brown-out Reset

Special Microcontroller Features:

- Enhanced Flash program memory:
  - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
  - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with on-chip low-power RC oscillator for reliable operation

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

dsPIC30F6011A/6012A/6013A/6014A

Special Microcontroller Features (Cont.):

- Fail-Safe Clock Monitor operation:
  - Detects clock failure and switches to on-chip low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
  - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption

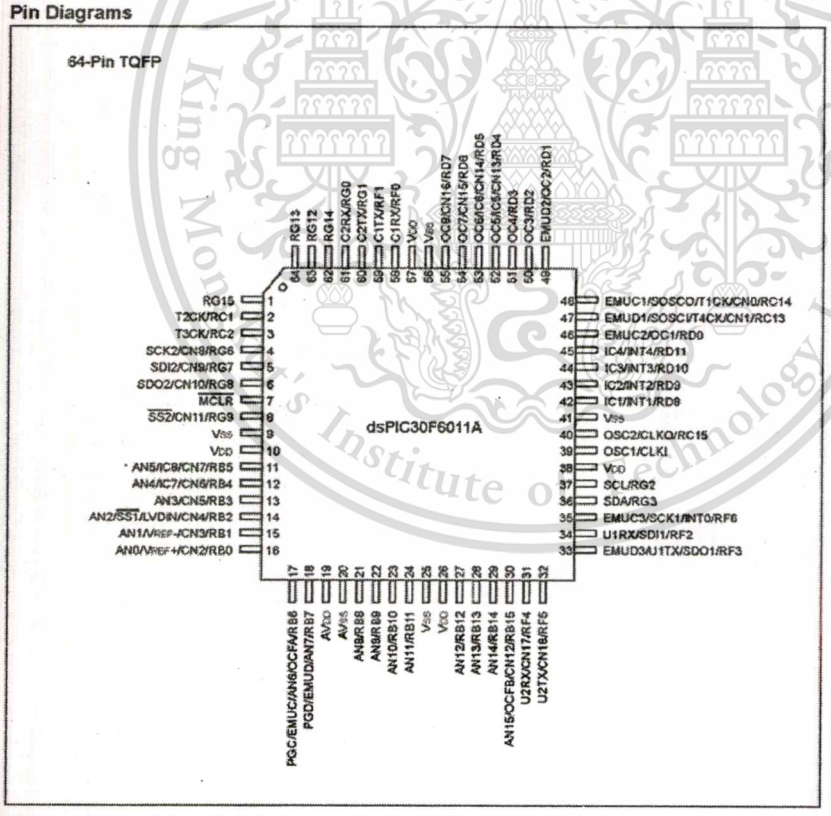
dsPIC30F6011A/6012A/6013A/6014A Controller Families

Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Codec Interface	ADC 12-bit 100 Ksps	UART	SPI	I^2C^{TM}	CAN
		Bytes	Instructions											
dsPIC30F6011A	64	132K	44K	6144	2048	5	8	8	—	16 ch	2	2	1	2
dsPIC30F6012A	64	144K	48K	8192	4096	5	8	8	AC'97, I^2S	16 ch	2	2	1	2
dsPIC30F6013A	80	132K	44K	6144	2048	5	8	8	—	16 ch	2	2	1	2
dsPIC30F6014A	80	144K	48K	8192	4096	5	8	8	AC'97, I^2S	16 ch	2	2	1	2

dsPIC30F6011A/6012A/6013A/6014A

Pin Diagrams

64-Pin TQFP



Note: For descriptions of individual pins, see **Section 1.0 "Device Overview"**.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

A.2 Microchip TC1047A

MICROCHIP

TC1047/TC1047A

Precision Temperature-to-Voltage Converter

Features

- Supply Voltage Range:
  - TC1047: 2.7V to 4.4V
  - TC1047A: 2.5V to 5.5V
- Wide Temperature Measurement Range: -40°C to +125°C
- High Temperature Converter Accuracy: ± 2°C, Max, at 25°C
- Linear Temperature Slope 10mV/°C (typ.)
- Available in 3-Pin SOT-23B Package
- Very Low Supply Current:
  - 35µA Typical

Applications

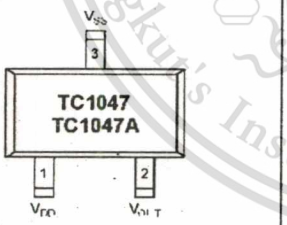
- Cellular Phones
- Power Supply Thermal Shutdown
- Temperature Controlled Fans
- Temperature Measurement/Instrumentation
- Temperature Regulators
- Consumer Electronics
- Portable Battery Powered Equipment

Device Selection Table

Part Number	Package	Temp. Range
TC1047VNB	3-Pin SOT-23B	-40°C to +125°C
TC1047AVNB	3-Pin SOT-23B	-40°C to +125°C

Package Type

3-Pin SOT-23B*



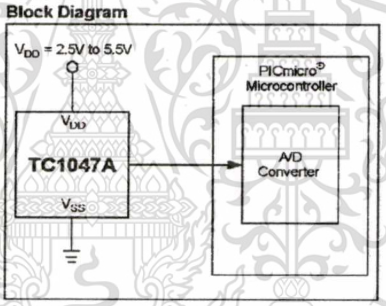
Note: * 3-Pin SOT-23B is equivalent to JEDEC TO-236

General Description

The TC1047 and TC1047A are linear voltage output temperature sensors whose output voltage is directly proportional to the measured temperature. The TC1047 and TC1047A can accurately measure temperature from -40°C to +125°C. With the TC1047, the supply voltage can vary between 2.7V and 4.4V. The power supply range of the TC1047A is from 2.5V to 5.5V.

The output voltage range for these devices is typically 100mV at -40°C, 500mV at 0°C, 750mV at +25°C, and 1.75V at +125°C. A 10mV/°C voltage slope output response allows for a predictable temperature measurement over a wide temperature range. The TC1047 and TC1047A are packaged in 3-Pin SOT-23B packages, making them ideal for space critical applications.

Block Diagram



This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

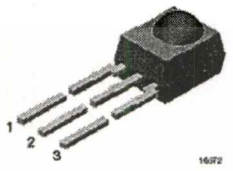
A.3 Vishay v4838

VISHAY.

TSOP22.., TSOP24.., TSOP48.., TSOP44..

Vishay Semiconductors

IR Receiver Modules for Remote Control Systems



FEATURES

- Low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against EMI
- Supply voltage: 2.7 V to 5.5 V
- Improved immunity against ambient light
- Insensitive to supply voltage ripple and noise
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC

MECHANICAL DATA

Pinning for TSOP44.., TSOP48..:
1 = OUT, 2 = GND, 3 = V_S
Pinning for TSOP22.., TSOP24..:
1 = OUT, 2 = V_S, 3 = GND

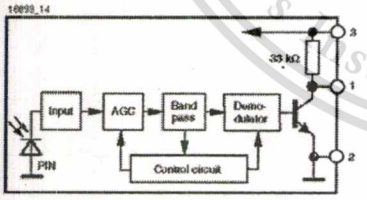
DESCRIPTION

These products are miniaturized receivers for infrared remote control systems. A PIN diode and a preamplifier are assembled on a lead frame, the epoxy package acts as an IR filter.
The demodulated output signal can be directly decoded by a microprocessor. The TSOP22.., TSOP48.. are compatible with all common IR remote control data formats. The TSOP24.., TSOP44.. are optimized to suppress almost all spurious pulses from energy saving fluorescent lamps but will also suppress some data signals.
This component has not been qualified according to automotive specifications.

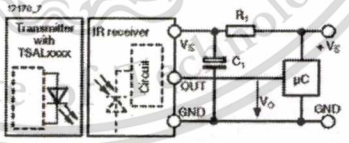
PARTS TABLE

CARRIER FREQUENCY	STANDARD APPLICATIONS (AGC2)		VERY NOISY ENVIRONMENTS (AGC4)	
	PINNING			
	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S, 3 = GND	1 = OUT, 2 = GND, 3 = V_S	1 = OUT, 2 = V_S, 3 = GND
30 kHz	TSOP4830	TSOP2230	TSOP4430	TSOP2430
33 kHz	TSOP4833	TSOP2233	TSOP4433	TSOP2433
36 kHz	TSOP4836	TSOP2236	TSOP4436	TSOP2436
38 kHz	TSOP4838	TSOP2238	TSOP4438	TSOP2438
40 kHz	TSOP4840	TSOP2240	TSOP4440	TSOP2440
56 kHz	TSOP4856	TSOP2256	TSOP4456	TSOP2456

BLOCK DIAGRAM



APPLICATION CIRCUIT



The external components R_1 and C_1 are optional to improve the robustness against electrical overstress (typical values are R_1 = 100 Ω, C_1 = 0.1 µF).
The output voltage V_O should not be pulled down to a level below 1 V by the external circuit.
The capacitive load at the output should be less than 2 nF.

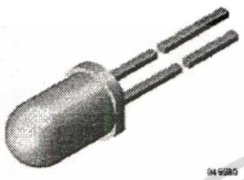
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

A.4 Vishay TSAL7400

VISHAY.
www.vishay.com

TSAL7400
Vishay Semiconductors

High Power Infrared Emitting Diode, 940 nm, GaAlAs/GaAs



FEATURES

- Package type: leaded
- Package form: T-1¾
- Dimensions (in mm): Ø 5
- Peak wavelength: λ_p = 940 nm
- High reliability
- High radiant power
- High radiant intensity
- Angle of half intensity: $\varphi = \pm 25°$
- Low forward voltage
- Suitable for high pulse current operation
- Good spectral matching with Si photodetectors
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC

Note
** Please see document "Vishay Material Category Policy": www.vishay.com/doc?99902

DESCRIPTION

TSAL7400 is an infrared, 940 nm emitting diode in GaAlAs/GaAs technology with high radiant power molded in a clear, untinted plastic package.

APPLICATIONS

- Infrared remote control units with high power requirements
- Free air transmission systems
- Infrared source for optical counters and card readers

PRODUCT SUMMARY				
COMPONENT	I_e (mW/sr)	φ (deg)	λ_p (nm)	t_r (ns)
TSAL7400	40	± 25	940	800

Note
- Test conditions see table "Basic Characteristics"

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	REMARKS	PACKAGE FORM
TSAL7400	Bulk	MOQ: 4000 pcs, 4000 pcs/bulk	T-1¾

Note
- MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS (T_{amb} = 25 °C, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Reverse voltage		V_R	5	V
Forward current		I_F	100	mA
Peak forward current	t_p/T = 0.5, t_p = 100 µs	I_{FM}	200	mA
Surge forward current	t_p = 100 µs	I_{FSM}	1.5	A
Power dissipation		P_V	160	mW
Junction temperature		T_j	100	°C
Operating temperature range		T_{amb}	- 40 to + 85	°C
Storage temperature range		T_{stg}	- 40 to + 100	°C
Soldering temperature	t ≤ 5 s, 2 mm from case	T_{sd}	260	°C
Thermal resistance junction/ambient	J-STD-051, leads 7 mm soldered on PCB	R_{thJA}	230	K/W

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

A.5 National LM2940

National Semiconductor

October 2006

LM2940/LM2940C
1A Low Dropout Regulator

General Description

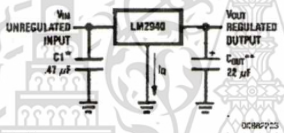
The LM2940/LM2940C positive voltage regulator features the ability to source 1A of output current with a dropout voltage of typically 0.5V and a maximum of 1V over the entire temperature range. Furthermore, a quiescent current reduction circuit has been included which reduces the ground current when the differential between the input voltage and the output voltage exceeds approximately 3V. The quiescent current with 1A of output current and an input-output differential of 5V is therefore only 30 mA. Higher quiescent currents only exist when the regulator is in the dropout mode ($V_{IN} - V_{OUT} < 3V$).

Designed also for vehicular applications, the LM2940/LM2940C and all regulated circuitry are protected from reverse battery installations or 2-battery jumps. During line transients, such as load dump when the input voltage can momentarily exceed the specified maximum operating voltage, the regulator will automatically shut down to protect both the internal circuits and the load. The LM2940/LM2940C cannot be harmed by temporary mirror-image insertion. Familiar regulator features such as short circuit and thermal overload protection are also provided.

Features

- Dropout voltage typically 0.5V @ I_O = 1A
- Output current in excess of 1A
- Output voltage trimmed before assembly
- Reverse battery protection
- Internal short circuit current limit
- Mirror image insertion protection
- P+ Product Enhancement tested

Typical Application



00867223

*Required if regulator is located far from power supply filter.

**C_{OUT} must be at least 22 µF to maintain stability. May be increased without bound to maintain regulation during transients. Locate as close as possible to the regulator. This capacitor must be rated over the same operating temperature range as the regulator and the ESR is critical; see curve.

Ordering Information

Temperature Range	Output Voltage						Package
	5.0	8.0	9.0	10	12	15	
0°C < T_J < 125°C	LM2940CT-5.0	–	LM2940CT-9.0	–	LM2940CT-12	LM2940CT-15	TO-220
	LM2940CS-5.0	–	LM2940CS-9.0	–	LM2940CS-12	LM2940CS-15	TO-263
	LM2940CSX-5.0	–	LM2940CSX-9.0	–	LM2940CSX-12	LM2940CSX-15	
−40°C < T_J < 125°C	LM2940LD-5.0	LM2940LD-8.0	LM2940LD-9.0	LM2940LD-10	LM2940LD-12	LM2940LD-15	LLP 1k Units Tape and Reel
	LM2940LDX-5.0	LM2940LDX-8.0	LM2940LDX-9.0	LM2940LDX-10	LM2940LDX-12	LM2940LDX-15	LLP 4.5k Units Tape and Reel
−40°C < T_J < 125°C	LM2940T-5.0	LM2940T-8.0	LM2940T-9.0	LM2940T-10	LM2940T-12		TO-220
	LM2940S-5.0	LM2940S-8.0	LM2940S-9.0	LM2940S-10	LM2940S-12	–	TO-263
	LM2940SX-5.0	LM2940SX-8.0	LM2940SX-9.0	LM2940SX-10	LM2940SX-12	–	
−40°C < T_A < 85°C	LM2940IMP-5.0	LM2940IMP-8.0	LM2940IMP-9.0	LM2940IMP-10	LM2940IMP-12	LM2940IMP-15	SOT-223
	LM2940IMPX-5.0	LM2940IMPX-8.0	LM2940IMPX-9.0	LM2940IMPX-10	LM2940IMPX-12	LM2940IMPX-15	SOT-223 in Tape and Reel
Marking	L53B	L54B	L0EB	L55B	L56B	L70B	

The physical size of the SOT-223 is too small to contain the full device part number. The package markings indicated are what will appear on the actual device.

© 2006 National Semiconductor Corporation

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

A.6 Maxim MAX3232CSE

19-0273; Rev 7; 1/97

MAXIM

3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors

General Description

The MAX3222/MAX3232/MAX3237/MAX3241 transceivers have a proprietary low-dropout transmitter output stage enabling true RS-232 performance from a 3.0V to 5.5V supply with a dual charge pump. The devices require only four small 0.1µF external charge-pump capacitors. The MAX3222, MAX3232, and MAX3241 are guaranteed to run at data rates of 120kbps while maintaining RS-232 output levels. The MAX3237 is guaranteed to run at data rates of 250kbps in the normal operating mode and 1Mbps in the MegaBaud™ operating mode, while maintaining RS-232 output levels.

The MAX3222/MAX3232 have 2 receivers and 2 drivers. The MAX3222 features a 1µA shutdown mode that reduces power consumption and extends battery life in portable systems. Its receivers remain active in shutdown mode, allowing external devices such as modems to be monitored using only 1µA supply current. The MAX3222 and MAX3232 are pin, package, and functionally compatible with the industry-standard MAX242 and MAX232, respectively.

The MAX3241 is a complete serial port (3 drivers/5 receivers) designed for notebook and subnotebook computers. The MAX3237 (5 drivers/3 receivers) is ideal for fast modem applications. Both these devices feature a shutdown mode in which all receivers can remain active while using only 1µA supply current. Receivers R1 (MAX3237/MAX3241) and R2 (MAX3241) have extra outputs in addition to their standard outputs. These extra outputs are always active, allowing external devices such as a modem to be monitored without forward biasing the protection diodes in circuitry that may have Vcc completely removed.

The MAX3222, MAX3237, and MAX3241 are available in space-saving TSSOP and SSOP packages.

Applications

Notebook, Subnotebook, and Palmtop Computers

High-Speed Modems

Battery-Powered Equipment

Hand-Held Equipment

Peripherals

Printers

Typical Operating Circuits appear at end of data sheet.

MegaBaud and UCSP are trademarks of Maxim Integrated Products, Inc.

Next Generation Device Features

- **For Smaller Packaging: MAX3228E/MAX3229E: +2.5V to +5.5V RS-232 Transceivers in UCSP™**
- **For Integrated ESD Protection: MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E: ±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers**
- **For Low-Voltage or Data Cable Applications: MAX3380E/MAX3381E: +2.35V to +5.5V, 1µA, 2 Tx/2 Rx RS-232 Transceivers with ±15kV ESD-Protected I/O and Logic Pins**

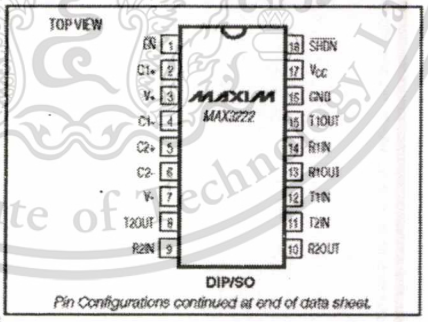
Ordering Information

PART	TEMP RANGE	PIN-PACKAGE	PKG CODE
MAX3222CUP+	0°C to +70°C	20 TSSOP	U20+2
MAX3222CAP+	0°C to +70°C	20 SSOP	A20+1
MAX3222CWN+	0°C to +70°C	18 SO	W18+1
MAX3222CPN+	0°C to +70°C	18 Plastic Dip	P18+5

+Denotes lead-free package.

Ordering information continued at end of data sheet.

Pin Configurations



Pin Configurations continued at end of data sheet.

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Appendix B

Experimental Program Code

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

B.1 Header File

//File head.h

```
#include <p30f6014.h>
#include <dsp.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
extern unsigned int RTC_DATA;
extern unsigned int i,k,l,c;
extern unsigned int RTC_STR;
extern unsigned int com16;
extern unsigned char com8;;
extern const unsigned int first14;
extern const unsigned char data[2];
extern unsigned int RTC_SAM;
extern unsigned int cmod[4];
extern float u_c;
```

B.2 Main Program File

//File main.c

```
#include "head.h"
#define SIGL OC8CONbits.OCM = 0b110
#define TUSAGE 12
#define TSET 25
_FOSC(CSW_FSCM_OFF & XT_PLL8);
_FWDT(WDT_OFF);
_FBORPOR(PBOR_OFF & BORV_27 & PWRT_16 & MCLR_EN);
_FGS(CODE_PROT_OFF);
extern float Tlast;
unsigned int N = 0;
void _ISR _T3Interrupt(void);
void UARTinits(void);
void OC6inits(void);
void Timersinits(void);
void INT1inits(void);
void _ISR _INT1Interrupt(void);
void sendDATA(unsigned int mode);
void sendDATA2(unsigned int mode);
void PORTsinits(void);
double TempConv(void);
void ADCinits(void);
void onoffcontrl(float T, float Tset);
unsigned int PIDcontrl(float T, float Tset);
void predictor(float T, float Tset, unsigned int Rs);
int main(void)
{
        float T;
        int RST = TUSAGE;
        float Tset = TSET;
        UARTinits();
        INT1inits();
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
        Timersinits();
        OC6inits();
        ADCinits();
        PORTsinits();
        T = TempConv();
        N = PIDcontrl(T, Tset);
        predictor(T,Tset,RST);
        RST--;
        cmod[1] = 1;
        cmod[0] = 0;
        cmod[2] = 1;
        printf("\r %0.4f \t %d \t %0.4f \t %d \t %0.4f \t %d \n",(double)T, cmod[2], (double)u_c, cmod[3],
(double)Tlast, RST);
        while(1)
        {
                if(RTC_STR == N)
                {
                        cmod[1] = 1;
                        cmod[0] = 1;
                        cmod[2] = 0;
                }
                if(RTC_SAM >= 300)
                {
                        RTC_SAM = 0;
                        T = TempConv();
                        if(RTC_STR >= 9000)
                                if(RST >= 0)
                                        if(cmod[3] == 0)
                                        {
                                                N = PIDcontrl(T, Tset);
                                                cmod[1] = 1;
                                                cmod[0] = 0;
                                                RTC_STR = 0;
                                                cmod[2] = 1;
                                                RST--;
                                                predictor(T,Tset,RST);
                                        }
                        printf("\r %0.4f \t %d \t %0.4f \t %d \t %.4f \t %d \n",(double)T, cmod[2],
(double)u_c, cmod[3], (double)Tlast, RST);
                }
                if(cmod[1] == 1)
                        if(cmod[3] == 0)
                                sendDATA2(cmod[0]);
        }
}
```

B.3 Initializing Program File

```
//File inits.c

#include "head.h"
#define SIGH OC6CONbits.OCM = 0b000
#define SIGL OC6CONbits.OCM = 0b110
void UARTinits(void);
void OC6inits(void);
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
void Timersinits(void);
void INT1inits(void);
void PORTsinits(void);
void ADCinits(void);
void UARTinits(void)
{
        U1MODEbits.PDSEL = 0b00;
        U1MODEbits.STSEL = 0;
        U1BRG = 7;
        U1MODEbits.UARTEN = 1;
        U1STAbits.UTXEN = 1;
}
void OC6inits(void)
{
        OC6RS = 194;
        OC6R = 194;
        SIGH;
}
void Timersinits(void)
{
        PR2 = 387;
        PR3 = 22;
        PR4 = 5759;
        T4CONbits.TCKPS = 0b11;
        T3CONbits.TCKPS = 0b10;
        _T4IE = 1;
        _T3IE = 1;
        T2CONbits.TON = 1;
        T3CONbits.TON = 1;
        T4CONbits.TON = 1;
}
void INT1inits(void)
{
        _TRISA12 = 1;
        _INT1IE = 1;
}
void PORTsinits(void)
{
        _TRISD0 = 0;
        _LATD0 = 0;
        _TRISD5 = 0;
        _LATD5 = 0;
        _TRISD7 = 0;
        _LATD7 = 0;
        _TRISD1 = 0;
        _LATD1 = 0;
        _TRISB7 = 1;
}
void ADCinits(void)
{
        ADPCFG = 0b1111111101111111;
        ADCHSbits.CH0SA = 0b0111;
        //ADCON2bits.CHPS = 0b00;
        ADCON1bits.SAMP = 0;
        ADCON2bits.VCFG = 0b000;
        ADCON3bits.ADCS = 0b111100;
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
        ADCON3bits.ADRC = 0;
        ADCON2bits.CSCNA = 0;
        ADCON1bits.SSRC = 0b111;
        ADCON1bits.ASAM = 0;
        ADCON3bits.SAMC = 0b11111;
        ADCON1bits.FORM = 0b00;
        ADCON2bits.SMPI = 0b0000;
        ADCON1bits.ADON = 1;
}
```

B.4 Temperature Measuring Program File

//File tempconv.c

```
#include "head.h"
double TempConv(void);
double TempConv(void)
{
        ADCON1bits.SAMP = 1;
        while(ADCON1bits.DONE == 0);
        ADCON1bits.DONE = 0;
        IFS0bits.ADIF = 0;
        return (((5.0/4096.0)*ADCBUF0-0.5)/0.01);
}
```

B.5 Remote Signal Imitation Program File

//File sendDATA2.c

```
#include "head.h"
#define STARTL 40
#define STARTH 20
#define DATAN 20
#define ENDN 44
#define FIRSTD 14
#define ZEROL 4
#define ONEL 10
#define GAP 6
#define INIT16 0x2000
#define INIT8 0x20
#define SIGH OC6CONbits.OCM = 0b000
#define SIGL OC6CONbits.OCM = 0b110
const unsigned int first14 = 0b0001101001110001;
const unsigned char data[2] = {0b00010111, 0b00110110};
unsigned int com16 = INIT16;
unsigned char com8 = INIT8;
unsigned int c = 0;
void sendDATA2(unsigned int mode);
void sendDATA2(unsigned int mode)
{
        switch(k)
        {
                case 0:
                {
                        SIGL;
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
            com16 = INIT16;
            com8 = INIT8;
            c = 0;
            RTC_DATA = 0;
            k = 1;
            _LATD1 = 0;
            break;
}
case 1:
{
            if(RTC_DATA >= STARTL)
            {
                        RTC_DATA = 0;
                        SIGH;
                        k = 2;
                        _LATD1 = !_LATD1;
            }
            break;
}
case 2:
{
            if(RTC_DATA >= STARTH)
            {
                        RTC_DATA = 0;
                        SIGL;
                        k++;
                        _LATD1 = !_LATD1;
            }
            break;
}
case 3:
{
            if(RTC_DATA >= GAP)
            {
                        RTC_DATA = 0;
                        SIGH;
                        _LATD1 = !_LATD1;
                        if(c<ENDN)
                        {
                                    if(c<FIRSTD)
                                    {
                                                if((com16&first14)==0)
                                                k = 4;
                                                else
                                                k = 5;
                                                com16 = com16 >> 1;
                                    }
                                    else if(c<DATAN)
                                    {
                                                if((com8&data[mode])==0)
                                                            k = 4;
                                                else
                                                            k = 5;
                                                com8 = com8 >> 1;
                                    }
                                    else
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
                                        k = 4;
                                c++;
                        }
                        else
                        {
                                k = 0;
                                i = 0;
                                cmod[1] = 0;
                        }
                }
                break;
        }
        case 4:
        {
                if(RTC_DATA >= ZEROL)
                {
                        RTC_DATA = 0;
                        SIGL;
                        k = 3;
                        _LATD1 = !_LATD1;
                }
                break;
        }
        case 5:
        {
                if(RTC_DATA >= ONEL)
                {
                        RTC_DATA = 0;
                        SIGL;
                        k = 3;
                        _LATD1 = !_LATD1;
                }
                break;
        }
        default:
        {
                SIGH;
                k = 0;
                break;
        }
    }
}
```

B.6 Control Program File

```
//File controller.c

#include "head.h"
#define A 0.1775
#define B 0.08167
#define Al 0.4771
#define Q 15.1806
#define SE 3
//Global variables
float e = 0;
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
float e_1 = 0;
float u_c = 0;
float Tlast = 29.0;
unsigned int cmod[4] = {0,0,0,0};
unsigned int st = 0;
extern double RTC_min;
extern unsigned int N;
//Function's declaration
void onoffcontrl(float T, float Tset);
double TempConv(void);
void sendDATA2(unsigned int mode);
unsigned int PIDcontrl(float T, float Tset);
void predictor(float T, float Tset, unsigned int Rs);
//Function's lines
void onoffcontrl(float T, float Tset)
{
        e = Tset - T;
        if(e >= 0.5)
        {
                if((cmod[0] == 0) || (st == 0))
                {
                        st = 1;
                        cmod[0] = 1;
                        cmod[1] = 1;
                        cmod[2] = 0;
                }
        }
        if(e <= -0.5)
        {
                if((cmod[0] == 1) || (st == 0))
                {
                        st = 1;
                        cmod[0] = 0;
                        cmod[1] = 1;
                        cmod[2] = 1;
                }
        }
}
unsigned int PIDcontrl(float T, float Tset)
{
        e_1 = e;
        e = Tset - T;
        u_c = B*e_1 - A*e + u_c;
        if(u_c > 1)
        u_c = 1;
        if(u_c < 0)
        u_c = 0;
        return u_c*9000;
}
void predictor(float T, float Tset, unsigned int Rs)
{
        float T_1;
        unsigned int w;
        T_1 = Al*T + Q;
        for (w=1;w<Rs;w++)
        T_1 = Al*T_1 + Q;
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
        if((T_1 - Tset) <= SE)
        {
                cmod[3] = 1;
                u_c = 0;
                N = 0;
        }
        Tlast = T_1;
}
```

B.7 Interrupt Program File

//File interrupts.c

```
#include "head.h"
unsigned int i = 0;
unsigned int k = 0;
unsigned int RTC_DATA = 0;
unsigned int RTC_STR = 0;
unsigned int RTC_SAM = 0;
void _ISR _T3Interrupt(void);
void _ISR _INT1Interrupt(void);
void _ISR _T4Interrupt(void);
void _ISR _T3Interrupt(void)
{
        RTC_DATA++;
        _T3IF = 0;
}
void _ISR _T4Interrupt(void)
{
        _T4IF = 0;
        RTC_STR++;
        RTC_SAM++;
}
void _ISR _INT1Interrupt(void)
{
        i++;
        _INT1IF = 0;
}
```

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Bibliography

[1] U.S. Dept. Energy. **Buildings Energy Data Book.** [Online]. Available: http://buildingsdatabook.eren.doe.gov/. 2009.

[2] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin. "Reducing Transient and Steady State Electricity Consumption in HVAC using Learning-Based Model-Predictive Control" **Proceedings of the IEEE,** Vol. 100, No. 1, January 2012. pp. 240-253.

[3] D.C. Montgomery. **Design and Analysis of Experiments.** 6th ed. John Wiley & Sons, Inc. 2005.

[4] L.D. Desborough, R.M. Miller. "Increasing Customer Value of Industrial Control Performance Monitoring—Honeywell's Experience" **AIChE Symposium Series,** Vol. 98, No. 326, January, 2001. pp. 172-192.

[5] M. Kano and M. Ogawa. "The State of the Art in Advanced Process Control in Japan" **Proceeding of IFAC Symposium on Advanced Control of Chemical Process 2009,** July 2009, Istanbul, Turkey.

[6] S. Skogestad. "Simply Analytic Rules for Model Reduction and PID Controller Tuning" **Journal of Process Control,** Vol. 13, No. 4, June 2003. pp. 291-309.

[7] K. Ogata. **Modern Control Engineering.** 4th ed. Prentice Hall, Inc. 2002.

[8] W.H. Carrier. **Modern Air Conditioning, Heating and Ventilating.** Pitman Publishing. 1941.

[9] A. Badawy. "Achitechtural Provision against Heat in the Orient" **Journal of Near Eastern Studies,** Vol. 17, No. 2, April 1958. pp. 122-128.

[10] J. Lepkowsky and S. Bible. "**Interfacing the TC1047A Analog Output Temperature Sensor to a PICmicro® Microcontroller.**" [Online]. Available: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en022405. 2005.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.