# DEVELOPMENT OF SMART EGG INCUBATOR SYSTEM FOR VARIOUS TYPES OF EGG (SEIS)

ABU MUSA BIN MOHD ADID

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

## BORANG PENGESAHAN STATUS TESIS♦

**JUDUL:** **SMART EGG INCUBATOR SYSTEM FOR VARIOUS TYPES OF EGG (SEIS)**

**SESI PENGAJIAN:** **2008/2009**

Saya **ABU MUSA BIN MOHD ADID (850720-03-6265)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan ( √ )

| | | |
|---|---|---|
| [ ] | **SULIT** | (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) |
| [ ] | **TERHAD** | (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan) |
| [ √ ] | **TIDAK TERHAD** | |

Disahkan oleh:

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap:

**NO. 1, JLN PERSIARAN 9,**
**TAMAN BUNGA RAYA, GEMEREH,**
**85000 SEGAMAT,**
**JOHOR**

**ABDUL HALIM BIN MOHD HANAFI**
( Nama Penyelia )

Tarikh: **12 NOVEMBER 2008**

Tarikh: : **14 NOVEMBER 2008**

CATATAN:
- * Potong yang tidak berkenaan.
- ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
- ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

# DEVELOPMENT OF SMART EGG INCUBATOR SYSTEM FOR VARIOUS TYPES OF EGG (SEIS)

ABU MUSA BIN MOHD ADID

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

12 NOVEMBER, 2008

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : ____________________________

Name : ABDUL HALIM BIN MOHD HANAFI

Date : 12 NOVEMBER 2008

I declare that this thesis entitled "Smart Egg Incubator System for Various Types of Egg" is the result of my own research except as cited in the references. The thesis is not been accepted for any degree is not concurrently submitted in candidature of any other degree.

Signature : ________________________________

Author : <u>ABU MUSA BIN MOHD ADID</u>

Date : <u>12 NOVEMBER 2008</u>

To my beloved father, mother, brothers and sisters.

## ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Mr. Abdul Halim Bin Mohd Hanafi, for encouragement, guidance, critics and friendship, advices and motivation. Without his continued support and interest, this thesis would not have been the same as presented here.

My fellow postgraduate students should also be recognized for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members.

## ABSTRACT

The purpose of this project is to design and develop the system of an egg incubator that able to incubate various type of egg named as Smart Egg Incubator System for Various Types of Egg (SEIS). The SEIS will fill up with the temperature and humidity sensor that can measure the condition of the incubator and automatically change to the suitable condition for the egg. The health of egg is very important for the development of embryo within the egg. Improper control means that the temperature or humidity is too high or too low. In this project, the light heater is use to give the suitable temperature to the egg. By using the water and controlling fan, it is can make sure the humidity and ventilation in good condition. The status condition in the SIES will appear on the LCD screen display. To make sure all part of egg was heated by lamp, DC motor is very useful to rotate iron rode at the bottom side and automatically change the position of egg. The entire element will be controlled using programmable integrated circuit (PIC). The PIC is a type of microcontroller that can process a data from sensor and will execute the control element to change the condition of SEIS. This project will be a user friendly product since the SEIS can move to other place. It will secure by user must enter the password before activate the system.

# ABSTRAK

Tujuan utama projek ini adalah untuk membangun dan mencipta suatu sistem mesin pengeram telur yang mana berkebolehan untuk digunakan terhadap pelbagai jenis telur yand dipanggil *Smart Egg Incubator System for Various Types of Egg (SEIS)*. SIES ini dilengkapi dengan pengesan haba dan pengesan kelembapan yang akan memantau keadaan semasa SIES dan akan berfungsi secara automatik untuk memberikan keadaan yang sesuai terhadap telur. Kesihatan telur merupakan perkara yang penting terhadap pembesaran embrio. Ketidaksempurnaan dalam pengawalan bererti keadaan suhu dan kelembapan yang terlalu tinggi atau terlalu rendah. Di dalam projek ini, haba yang terkeluar daripada lampu akan memberikan suhu yang sesuai kepada telur. Dengan menggunakan air dan pengawalan kipas, ia akan memastikan kelembapan dan pengudaraan berada dalam keadaan yang sesuai. Status semasa keadaan SEIS akan dipaparkan pada LCD. Untuk memberikan haba yang menyeluruh pada semua bahagian telur, motor DC digunakan untuk memutarkan rod besi dan secara automatiknya akan mengubah kedudukan telur. Semua bahagian akan dikawal dengan mengunakan *programmable integrated circuit (PIC).* PIC merupakan sejenis kawalan mikro yang akan memproses data daripada alat pengesan dan akan menggerakkan bahagian kawalan untuk mengubah keadaan SEIS. Alat ini merupakan mesra pengguna kerana ia mudah dialih ke tempat lain. Ianya terdapat sistem kawalan keselamatan di mana pengguna perlu memasukkan kata laluan yang betul sebelum mengaktifkan sistem kawalan.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| SEIS | - | Smart Egg Incubator System for Various Types of Egg |
| DC | - | Direct Current |
| AC | - | Alternating Current |
| PIC | - | Programmable Interface Controller |
| LCD | - | Liquid Crystal Display |
| CPU | - | Central Processing Unit |
| RAM | - | Random-Access Memory |
| EEPROM | - | Electrically Erasable Programmable Read-Only Memory |
| PWM | - | Pulse Width Modulation |
| ADC | - | analogue to digital converter |
| ASCII | - | American Standard Code for Information Interchange |
| GUI | - | Graphic User Interface |

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Industrial rising of farm animals indoors under conditions of extremely restricted mobility is commonly known as factory farming. It is done as part of industrial agriculture which is a set of methods that changes as laws and technology change known as industrial agriculture which is designed to produce the highest output at the lowest cost, using economies of scale, modern machinery, modern medicine, and global trade for financing, purchases and sales.

Egg incubator is one of the inventions that provide opportunity especially for who want to be excellent farmer. There is one of easy and fastest way that can make a product. This invention will upgrade the egg incubator that has already in market today. The systems will automatically controlling the temperature and humidity of the incubator for various types of egg. The function of egg incubator is to take over the animal job to incubate an egg until hatching.

## 1.2 Objectives

- To built the incubator with automatic controlling the humidity and temperature. Improper control means that the temperature or humidity is too high or too low for a sufficient length of time that it interferes with the normal growth and development of the embryo. Poor results also occur from improper ventilation, egg turning and sanitation of the machines or eggs.

- To built the incubator that able to incubate various types of egg. Making an egg incubator that user friendly will produce more valuable production and available for many types of egg.

## 1.3 Scopes

Design a system and hardware for egg incubator with automatic controlled the temperature and humidity using programmable interface controller (PIC) according to types of egg and parameters given by user.

## 1.4 Project Goal

With a bit of research we determined it would be the best course of action for several reasons:

- The incubator will help farmer produce product in a short time with large amount of eggs.
- An egg incubator can be considered a replacement for incubate session of animal.
- The incubator will be large enough to avoid problems of less production.
- Ideas to incubate many types of egg in one time can be tested.

## 1.5 Thesis Organization

This thesis is combination of b chapters that contain the Introduction, Literature Review, Methodology, Result and Discussion and the last chapter is a Conclusion and Recommendation of the project.

Chapter 1 is an introduction of the project. In this chapter, we will explain the background and objectives of the project. The concept of the project and the overall overview of the project also will be discussed in this chapter.

Chapter 2 focuses on the literature review and the project flow for the development of the Smart Egg Incubator System for Various Types of Egg.

Chapter 3 will explain about the project methodologies of the project. The project development consists of three parts which is mechanical design, hardware design and software design. In mechanical design, it is about the development of the egg roller, chassis and body. This incubator body was 70% built by using wood and 30% for egg chassis. By using this kind of material, we can reduce the cost to build this incubator. In this section also, we will discuss about the dimension, and how to construct and assemble the chassis and roller. In the hardware design, they are containing of circuit design of the controller system. That will include P-spice layout and information about components usage. The layout consists of four circuit such as master controller circuit, slave controller circuit, lamp controller circuit and power supply circuit. The last part is a software development of the controller. In this chapter we will expose on how to compile and program the PIC18F4550. The program has been built by step which is to function all of electronic device. Lastly all sub programs will organize to execute the flow function.

Chapter 4 discusses all the results obtained and the discussion of the project. During do the project, many problems have accorded. Besides that, by doing some inspection and troubleshooting, the solution has been found and all devices may function properly.

Chapter 5 discusses the conclusion and recommendation of the project. Finally the project has been done with the expected result. This project can be testing for the long time period but some recommendation must be included to make sure the system running properly without any violation.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

Incubation is the process by which birds hatch their eggs, and to the development of the embryo within the egg. The most vital factor of incubation is the constant temperature required for its development over a specific period. Especially in domestic fowl, the act of sitting on eggs to incubate them is called brooding. The action or behavioral tendency to sit on a clutch of eggs is also called broody, and most egg laying breeds of chicken have had this behavior selectively bred out of them to increase production.

In most species, body heat from the brooding parent provides the constant temperature, though several groups, notably the Megapodes, instead use geothermal heat or the heat generated from rotting vegetable material, effectively creating a giant compost heap. The humidity is also critical, and if the air is too dry the egg will lose too much water to the atmosphere, which can make hatching difficult or impossible. As incubation proceeds, an egg will normally become lighter, and the air space within the egg will normally become larger, owing to evaporation from the egg. [1]

## 2.2 Embryonic Development

Embryonic development is a continuous process that can roughly be divided into three different phases. They are differentiation, growth and the maturation.

Typically, differentiation of organs occurs in the first days of incubation. The growth and the maturation of the organs occur in the later phases of development. Each of these phases requires specific incubator conditions. As the embryo grows, its metabolic rate increases and this is accompanied by increased heat production. Consequently, the natural pattern of the embryo and eggshell temperature shows an increase towards the end of incubation. In the incubator we must differentiate between the temperature set point at which the incubator operates and the temperature of the air at the level of the eggs, which determines the temperature of the egg and embryo. [1]

At the start of incubation the embryo produces little heat and eggs must be warmed. This means that the air temperature must be higher than the egg temperature. As the embryo grows, metabolic heat production increases and to prevent overheating the air surrounding the eggs must be cooled such that heat is removed from the eggs.

## 2.3 Egg Incubator

Novice poultry producers usually become interested in artificial incubation of their own chicks. The success of this type project depends on proper care and incubation of the hatching eggs so healthy, vigorous chicks are produced. Many times a producer carefully attends to the incubation process but disregards the care of the eggs before they are placed in the incubator. Even before incubation starts the embryo is developing and needs proper care. Hatching eggs suffer from reduced hatchability if the eggs are not cared for properly.

The size and type of incubator selected depends on the needs and future plans of each producer. Many different models are available. For continuous settings, separate incubator and hatcher units are recommended. If all eggs in the unit are at the same stage of incubation, a single unit can be used.

Locate the incubator and hatcher units indoors to protect them from major weather changes. It is essential that the room has a good ventilation system to supply plenty of fresh air. Keeping the units indoors makes it easier to maintain uniform temperature and humidity.

There are basically two types of incubators available, *forced-air* and *still-air* incubators. Forced-air incubators have fans that provide internal air circulation. The capacity of these units may be very large. The still-air incubators are usually small without fans for air circulation. Air exchange is attained by the rise and escape of warm, stale air and the entry of cooler fresh air near the base of the incubator. Recommended temperatures vary between the two incubators, so follow the manufacturer's recommendation that accompany the units. [3]

## 2.4 Incubating Conditions

The condition of incubator is very importance element. Poor results are most commonly produced with improper control of temperature and/or humidity. Improper control means that the temperature or humidity is too high or too low for a sufficient length of time that it interferes with the normal growth and development of the embryo. Poor results also occur from improper ventilation, egg turning and sanitation of the machines or eggs.

The components than must be control in the incubator are temperature, humidity and ventilation. The temperature must depend on the types of egg. In order to hatch a good percentage of fertile eggs, an incubator must be able to maintain a constant temperature. Though different sorts of eggs require different heat levels, most will grow and hatch well at 99 to 101°F. Sure, that does sound imposingly precise, but such accuracy isn't all that difficult to achieve. [2]

Rarely is the humidity too high in properly ventilated still-air incubators. The water pan area should be equivalent to one-half the floor surface area or more. Increased ventilation during the last few days of incubation and hatching may necessitate the addition of another pan of water or a wet sponge. Humidity is maintained by increasing the exposed water surface area.

Ventilation is very important during the incubation process. While the embryo is developing, oxygen enters the egg through the shell and carbon dioxide escapes in the same manner. As the chicks hatch, they require an increased supply of fresh oxygen. As embryos grow, the air vent openings are gradually opened to satisfy increased embryonic oxygen demand. Care must be taken to maintain humidity during the hatching period. Unobstructed ventilation holes, both above and below the eggs, are essential for proper air exchange. [3]

**Table 2.1:** Condition of the incubator for various types of egg

| Species | Incub. Period (days) | Temp (F.)[1] | Humidity (F.)[2] | Do not turn after | Humidity Last 3 days[2] | Open vent more |
|---|---|---|---|---|---|---|
| Chicken | 21 | 100 | 85-87 | 18th day | 90 | 18th day |
| Turkey | 28 | 99 | 84-86 | 25th day | 90 | 25th day |
| Duck | 28 | 100 | 85-86 | 25th day | 90 | 25th day |
| Muscovy Duck | 35-37 | 100 | 85-86 | 31st day | 90 | 30th day |
| Goose | 28-34 | 99 | 86-88 | 25th day | 90 | 25th day |
| Guinea Fowl | 28 | 100 | 85-87 | 25th day | 90 | 24th day |
| Pheasant | 23-28 | 100 | 86-88 | 21st day | 92 | 20th day |
| Peafowl | 28-30 | 99 | 84-86 | 25th day | 90 | 25th day |
| Bobwhite Quail | 23-24 | 100 | 84-87 | 20th day | 90 | 20th day |

The Table 2.1 shows the condition of the incubator for various type of egg which is consists of suitable temperature and humidity for every type of eggs. The Fahrenheit unit can be converting to Celsius by using the Fahrenheit formula, F = [32 + (9/5) a'C].

# CHAPTER 3

# METHODOLOGY

## 3.1 Project Development

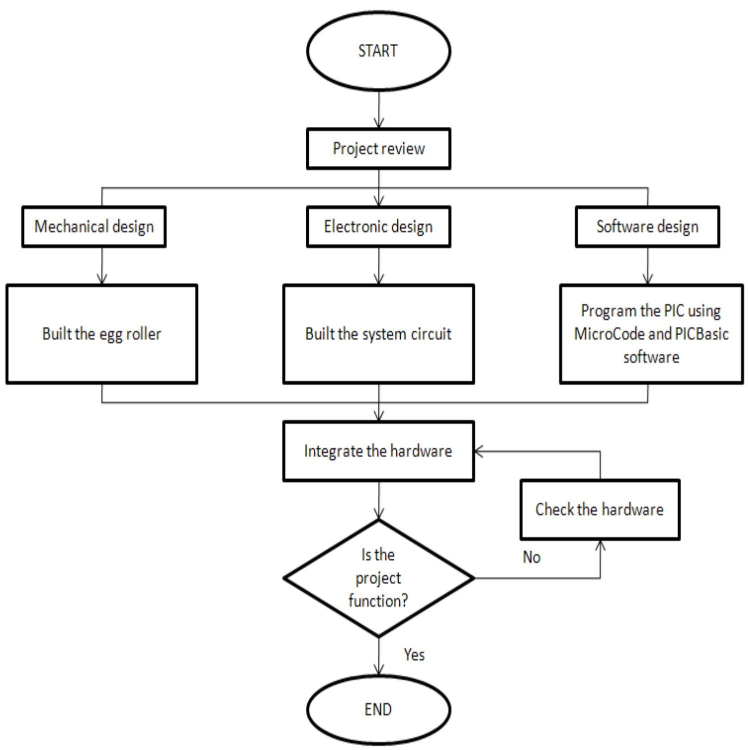



**Figure 3.1:** Flowchart of the project development

By refer to the Figure 3.1, project development was divided into three main sections. There are mechanical design, electronic design and software design. Then this part will combined together to perform the Smart Egg Incubator System for Various Types of Egg (SEIS).

The smart egg incubator is able to incubate various type of egg. The incubator will fill up with the temperature and humidity sensor that can measure the condition of the incubator and automatically change to the suitable condition for the egg.

Locate the incubator and hatchers units indoors to protect them from major weather changes. It is essential that the room has a good ventilation system to supply plenty of fresh air. Keeping the units indoors makes it easier to maintain uniform temperature and humidity.

Actually, this project consists of three elements that should be controlled. There are movement, temperature and humidity. In this project, the light is use to give the suitable temperature to the egg. The percentage of the humidity in the incubator need to consistent by control the fan and water through in the incubator, it is can make sure the humidity and ventilation in good condition. The status condition in the incubator will appear on the LCD screen display. To make sure all part of egg was heated by lamp, stepper motor is very useful to rotate iron rod at the bottom side and automatically change the position of egg.

The entire element will be controlled using programmable integrated circuit (PIC). The PIC is a type of microcontroller that can process a data from sensor and will execute the control element to change the condition of incubator. This project will be a user friendly product since the incubator can move to other place.

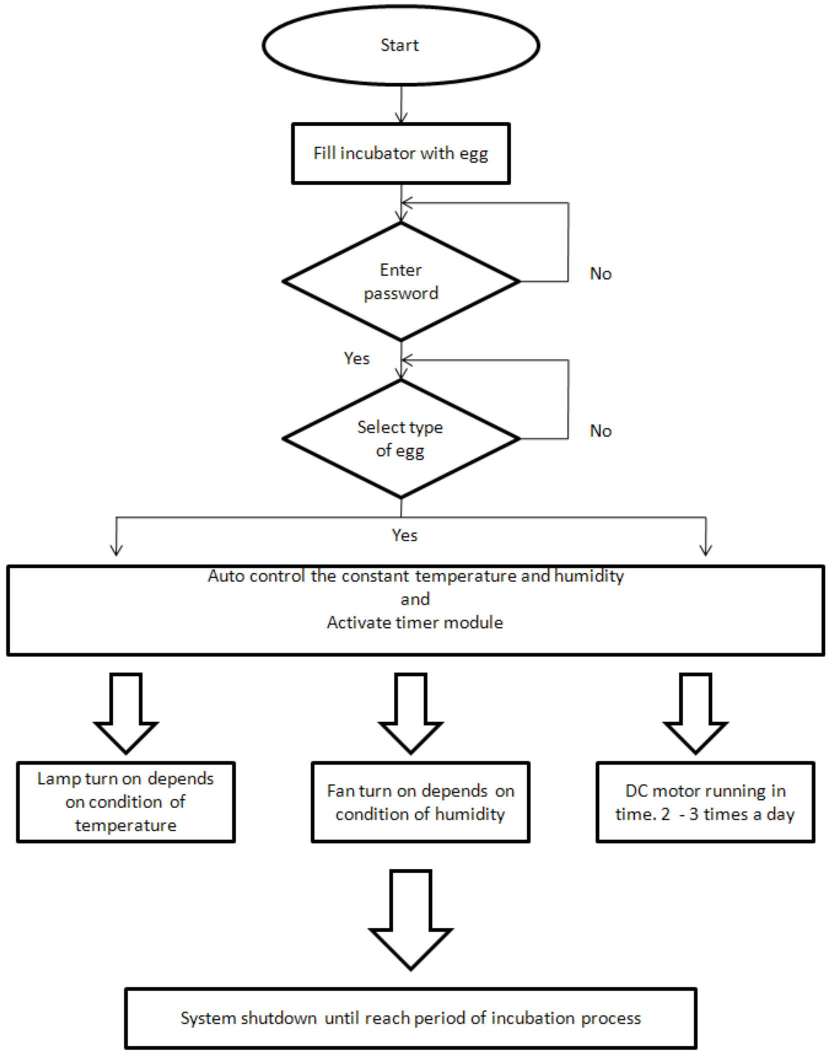



**Figure 3.2:** Flow chart of the project

From the Figure 3.2 shows the system of the incubator working. Firstly user needs to insert the correct password. That is for system protection from other disturbance and makes the system functioning constantly. After that users must select the type of egg. The system will operate under the type selected. The data of various types of egg was inserted during system development. For all various type of egg, there have difference incubation period until hatching. For this project development, 3 types of egg were selected. There are chicken egg's, duck egg's and quail egg's. The condition of egg has explained in table 1.1. In order to give good heating on an egg, the egg should change their position 2 or 3 times per day. This important to make sure all position of egg heated by lamp. Humidity of the incubator was control by water container to give some fresh air flow and fan controller to make the suitable humidity in incubator.

## 3.2 Mechanical Design

The project development was started with the mechanical design. It is consists the development of incubator casing and built an egg roller.

### 3.2.1 Incubator casing

To build the egg incubator system involves lots of concern in terms of the temperature, humidity and movement in order to care the health of the egg. The important thing that should know is how to change the position of the egg during the period of incubation. It is should get along with a mechanical egg-shifting system. The eggs in the incubator need to be shifted slowly and smoothly, since jostling would disturb the development of the chicks. By using the DC motor, the PIC should be programmed as long as stepper motor running in shifted slowly.

The construction of the incubator will begin with the built the casing of the incubator. A good quality material was used such as hardwood. The hardwood was choose because is preferable rather than softer wood. The softer wood will warp during the incubation process. That will affect the humidity of egg. This incubator can fill up to 20-25 eggs. Dimensional measuring of the incubator is 39cm long x 48.4cm wide x 48.4cm tall. 4 bulbs were placed around the wall inside of the incubator. The bulb usage is 5 Watt's that supply heat to the egg.

There are two stages for the incubator that call first drover and second drover. First drover is where amount of egg were placed. It fills with the egg roller, bulbs and fan. For the second drover, it is for water that to make fresh air flow in the incubator. This incubator becomes user friendly because it can move to other place. By use the board transparent plastic make easier to owner look inside the incubator.

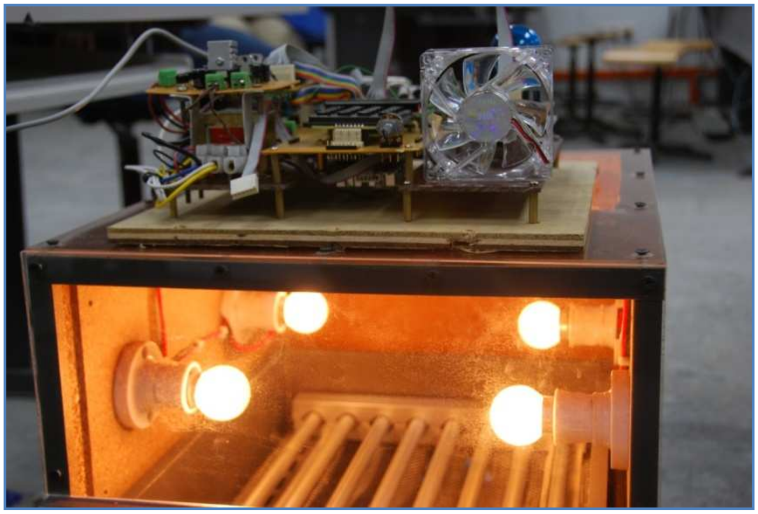


**Figure 3.3:** Casing of incubator

### 3.2.2 An egg roller

The egg roller is very important in order to change the position of eggs. The position of egg must be change two times every day until hatching. An idea to designing the egg roller such as conveyer which is needs some bearing and belting. It needs to change the position of egg in smoothly. The material used to build the egg roller is an aluminum plat and aluminum rode. It will reduce cost of construction because about 60% the material usage from cycle material.

Before construct the egg roller, all measurement was record since to make sure it really fix when put inside the first drover.

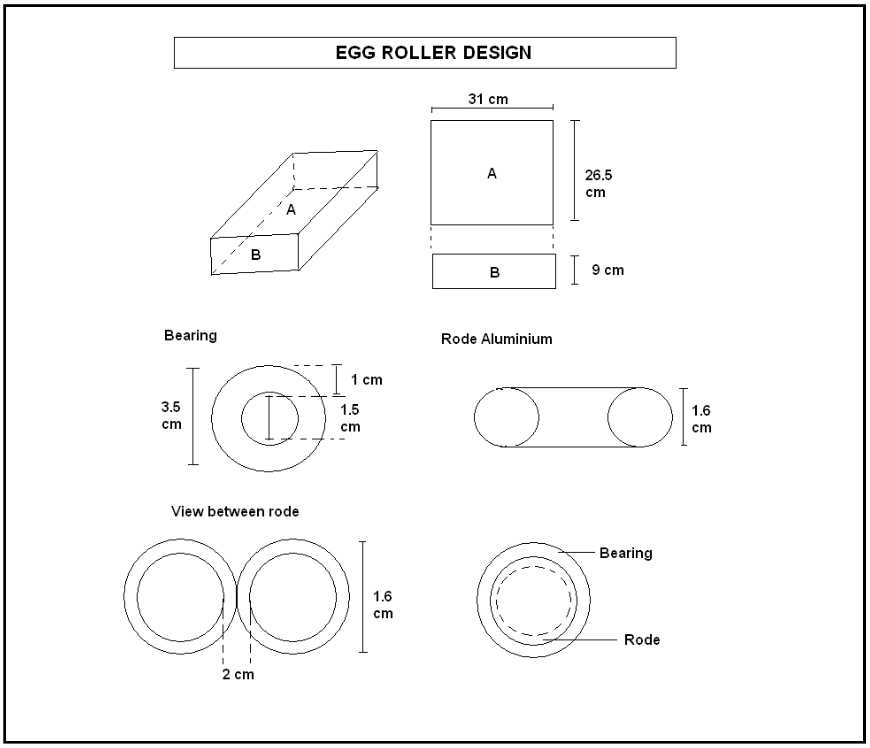



**Figure 3.4:** An egg roller design

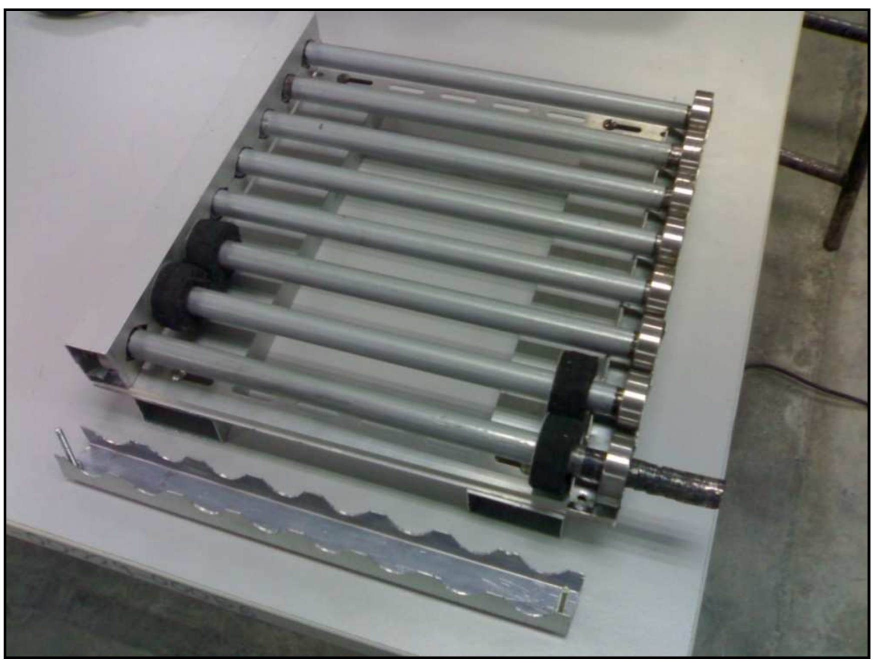


**Figure 3.5:** An egg roller

## 3.3 Hardware Design

In this chapter, the circuit explanation consists four major parts. There are master controller circuit, slave controller circuit, lamp controller circuit and power supply circuit. The master controller circuit was interface to the temperature and humidity circuit, Liquid Crystal Display (LCD), Keypad, lamp circuit and fan. For the slave controller circuit, it interface with DC motor circuit that running the egg roller.

Block diagram figure 3.6 shows the connection all part of system. There are consisting of input, CPU and output. The inputs of the system are keypad, temperature and humidity sensor and also power supply. The PIC18F4550 will operate as a CPU which is the main controlling system. For the output element consist of LCD screen display, DC motor, fan and lamp.

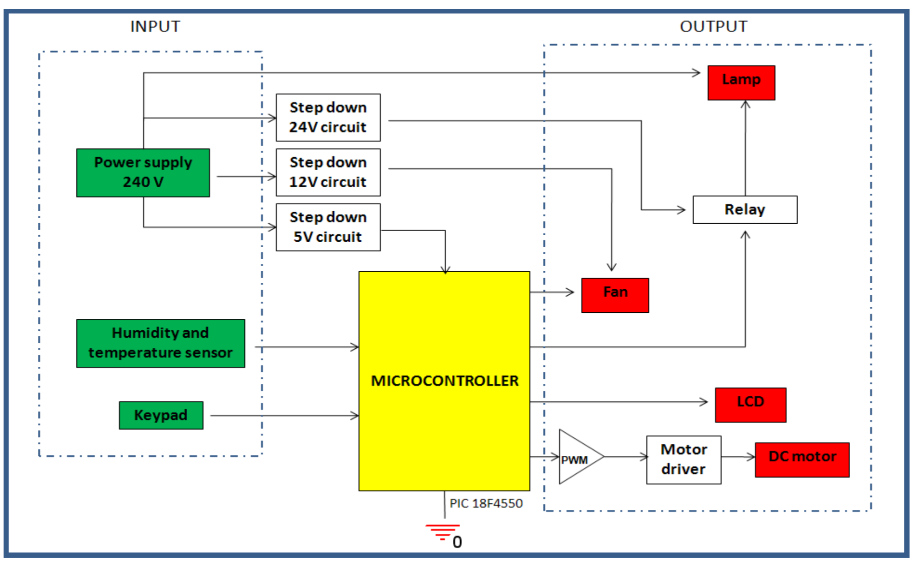



**Figure 3.6:** Block diagram of the hardware design

The operation of the system will start while user selects the type of egg by press the code on the keypad. After that controller which is PIC 18F4550 will set up the range of the temperature and humidity of the incubator. If any changes for the ventilation of the incubator, the temperature and humidity sensor will measure and send data to the controller as a feedback. LCD board will display any changes that might occur. The controller will control the speed of fan, light and flow of water until the ventilation of the incubator back to the condition needed. The controller also controls the shift rotation of the DC motor. This allows the egg change their position to care the growth of the embryo.

This system will run until the egg hatching. To make sure the system always running, the battery charge is needed. The battery as a backup power supply will supply their voltage whiles the system going breakdown. The breakdown might be occurring if main power supply was blackout.

### 3.3.1 Master Controller circuit

The master controller circuit schematic is shown in appendix A. The function of master circuit is to controller the input data from keypad and display on the LCD screen. It is also read the input data from temperature and humidity sensor and store in variable before display on the LCD screen. The master circuit also controls the output fan. When the CPU detect the value of temperature is high than suitable value, the lamp will off and when the CPU detect the value of humidity is high than suitable value, the fan will on. It can make the suitable condition in good status for egg incubation process.

#### 3.3.1.1 CPU microcontroller PIC18F4550

The microcontroller is an entire computer on a single chip. The advantage of designing around microcontroller is that a large amount of electronics needed for certain applications can be eliminated. This makes it the ideal device for use with large system and other applications where computing power is needed. The microcontroller is popular because the chip can be programmed easily to perform different functions and is very inexpensive. The microcontroller contains all the basic components that make up a computer. It contain a central processing unit (CPU), read-only memory, random-access memory (RAM), arithmetic logic unit, input and output lines, timers, serial and parallel ports, digital-to-analog converter, and analog-to-digital converters. There many kind of microcontroller in market such as Motorola, PIC, Basic Stamp and etc. But, in this project, the PIC18F4550 is chosen because of the functionality, lower in cost, robustness, easy to program and troubleshoot.

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1640 originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to "Programmable Interface Controller", but shortly thereafter was renamed "Programmable Intelligent Computer".

PICs are popular with developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

Harvard architecture is computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters (23 digits wide). These early machines had limited data storage, entirely contained within the data processing unit, and provided no access to the instruction storage as data, making loading and modifying programs an entirely offline process. The PIC18F4550 have characteristic such as below:

- Harvard architecture (enhanced flash & data EEPROM) – 40 pin configuration
- Cristal 4 MHz – 48 MHz (to provide pulse for the PIC) – internal 8Mhz
- Have 5 ports (A-E) that can make it as input or output
- Provide internal Analog to digital converter
- Flash Memory – 32 Kbyte, Data memory – 2 K byte Provide internal PWM.

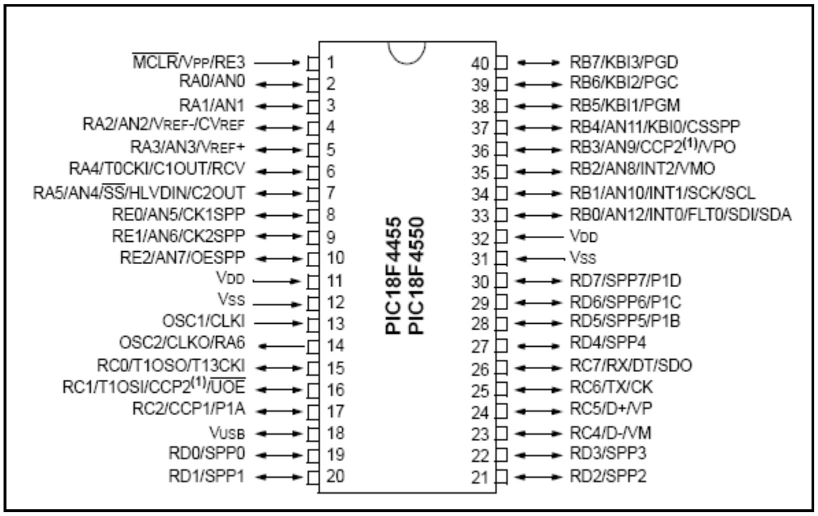



**Figure 3.7:** PIC18F4550 pin configuration

### 3.3.1.2 Liquid Crystal Display (LCD)

LCD screen needs to display the current value of temperature and humidity. Based on LCD module of the industry standard Hitachi HD44780, it connects to 7 pins of one port, and operates in 4 bit 'nibble' mode to save I/O pins. For this project, the LCD is connected to Port B. By connecting to Port B we have to use a pull-up resistor (R1) on RB4, and are unable to use RB5 (which is only an input), however this frees all of Port B which will allow us to use some of the extra hardware available on Port B, along with the LCD. The potentiometer P1 is for adjusting the contrast of the display, and if incorrectly adjusted can cause the display to be invisible. By using 4 bit mode we can connect the entire LCD module to one port, it uses exactly 10 pins (just right for our Molex connectors). In 4 bit mode we don't use pins 7-10, which are used as the lower 4 data bits in 8 bit mode, instead we write (or read) to the upper 4 pins twice, transferring half of the data each time.

#### 3.3.1.3 Keypad

In order to enter the password code, keypad 4x4 should used by connected to PIC. This is how the HEX keypad is connected, each square with a number or letter in it is a push to make switch, which connects the horizontal wires (rows) with the vertical wires (columns). As the switches are all interconnected, we need a way to differentiate between the different ones - the four resistors on the interface board pull lines X1 to X4 (column) high, these four lines are the ones which are read in the program. So in the absence of any switch been pressed these lines will all read high. The lines Y1 to Y4 (ROW) connections are connected to output pins, and if these are set high the switches will effectively do nothing - connecting a high level to a high level, results in a high level. Then the PIC will compare with the reference value to display on the LCD.

#### 3.3.1.4 Fan

To execute the 12VDC fan, the transistor is needed. The function of transistor is such as a switch and amplifier. The low signals from PIC will accepted by transistor and amplify it to execute the DC 12V fan.

#### 3.3.1.5 Temperature and humidity circuit

HSM-20G is a temperature and humidity sensor that built together functioning to measure the current condition of the incubator. With the ADC (analogue to digital converter) facilities internally in the PIC will read the analogue signal from sensor and convert to digital signal. Table 3.1 shows the characteristic of HSM-20G.

**Table 3.1:** Characteristic of HSM-20G

| Characteristics | | HSM-20G |
|---|---|---|
| Input voltage range | | DC 5.0±0.2V |
| Output voltage range | | DC 1.0—3.0 V |
| Measurement Accuracy | | ±5% RH |
| Operating Current (Maximum) | | 2mA |
| Storage RH Range | | 0 to 99% RH |
| Operating RH Range | | 20 to 95% (100% RH intermittent) |
| Transient Condensation | | ＜3%RH |
| Temperature Range | Storage | -20℃ to 70℃ |
| | Operating | 0℃ to 50℃ |
| Hysteresis (RH @ 25℃) | | MAX 2%RH |
| Long Term Stability(typical drift per year) | | ±1.5% |
| Linearity | | Linearity |
| Time Response(63% step change) | | 1 min |
| Dimensions(L*W) | | 34mm*22mm |

### 3.3.2 Slave controller circuit

The slave controller circuit schematic is shown in appendix B. The function of slave circuit is to controller DC motor and executes the hatching period depends on egg type.

#### 3.3.2.1 DC motor circuit

In this project, dc motor is needed since it attached with egg roller to change the position of egg. The dc motor need 12v power supply and must give PWM (pulse width modulation) to control their speed. By using cytron MD30A DC motor driver, we can supply the PWM and also easy to control the speed. For the timer, special function in PIC can make the timer function depends on egg period. To setting up the timer it is done by programmed PIC by using the compiler software.

MD30A is designed to drive high current brush motor or application. It is a full bridge motor driver intended for wide range of robotics and automotive applications. The board incorporates most of the components of the typical applications. With minimum interface, the board is ready to be plugged and play. It even includes 2 push buttons for fast test run. Simply add in power, this driver is ready to drive high current motor. Additionally, MD30A have protection of reverse polarity on battery input. It has been designed with capabilities and features of:

- Industrial grade PCB with heavy copper material for high current applications
- Each component is soldered properly and tested
- Support up to 30A maximum
- Protection against wrong polarity of Vin
- 5V logic level inputs
- 5V to 12V compatible for Vcc
- PWM speed control up to 10 KHz
- Bi-directional control for 1 motor
- Over voltage clamp
- Thermal Shut Down
- Cross-Conduction protection
- Linear current limiter
- Very low standby power consumption
- Protection against: Loss of ground and loss of Vin
- on-board push buttons for fast testing
- Fan heat sink for fast thermal release

The characteristic of MD30A DC motor driver shows at Table 3.2:

**Table 3.2:** Characteristic of MD30A DC motor driver

Electrical Characteristics

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{in}$* | Motor supply voltage | | 5.5 | | 24* | V |
| $V_{cc}$ | Operating supply voltage | | 5 | | 12 | V |
| f | PWM frequency | | 0 | | 10 | KHz |
| $V_{ov}$ | Overvoltage shut-down | | 36 | 43 | | V |
| $I_{LIM}$ | Current Limitation | | 30 | 45 | | A |
| $T_{TSD}$ | Thermal shut-down temperature | | 150 | 170 | 220 | °C |
| $T_{TR}$ | Thermal Reset temperature | | 135 | | | °C |
| $V_{pwl}$ | PWM low level voltage | | | | 1.5 | V |
| $I_{pwl}$ | Low level PWM pin current | Vpw = 1.5V | 1 | | | uA |
| $V_{pwh}$ | PWM high level voltage | | 3.25 | | | V |
| $I_{pwh}$ | High level PWM pin current | Vpw = 3.25V | | | 10 | uA |
| $V_{CW/CCWL}$ | CW input low level voltage | | | | 1.5 | V |
| $I_{CW/CCWL}$ | CW input low current | $V_{CW/CCW}$ = 1.5V | 1 | | | uA |
| $V_{CW/CCWH}$ | CW input high level voltage | | 3.25 | | 5.1 | V |
| $I_{CW/CCWH}$ | CW input high current | $V_{CW/CCW}$ = 3.25V | | | 10 | uA |

* Although $V_{in}$ can support up to 30V, it is limited to 24V because the maximum voltage of on board fan is 24V.

### 3.3.3 Lamp controller circuit

The lamp controller circuit consists of 4 relays that connected used to supply 240Vac and to switch on the lamp. To activate the relay, we need 24Vdc supply voltage with separately voltage regulator. In this circuit, 6-pin DIP optoisolators transistor (4N25) was used to energize the coil of the relay. The 4N25 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector. It is most economical optoisolator choice for medium speed and switching applications. Generally it is purpose for switching circuits and also for interfacing and coupling systems of different potentials and impedances. The lamp control circuit schematic is shown in appendix C.

### 3.3.4 Power supply circuit

In the power supply circuit, it is dividing into two supplies. First, it is a supply voltage for controller and second for DC motor. It is must be separated because DC motor used high current to run. That must be less function on controller while we put voltage supply together with DC motor.

The power supply circuit schematic is shown in appendix D. This power supply will provide 5v, 12v, and 24v. It is used step down method that regulated supply from 240VAC. Transformer center tap 240VAC to 12VAC was used to provide the voltage needed. The voltage regulator that used is 7805, 7812 and 7824. The function of voltage regulator is to give constant voltage to the circuit. The 5V power supply may used to supply voltage for operation of microcontroller, DC motor driver, Liquid crystal device (LCD) and keypad. For 12V is use to supply voltage to fan and DC motor. And for 24V power supply is use to energize the coil of relay. 240VAC may supply to bulb.
Below is the summary the function of the device usage:

## 3.4 Software Design

In the software design, mikroElektronika Basic compiler for Microchip PIC microcontrollers Version 6.0.0.0 and microcode studio compiler 2.46 was used to program the PIC. In this project, the program was write by part to make sure all devices can function properly.

### 3.4.1 Master control system

In the master code, program was write using mikroElektronika Basic compiler software. High language was used in this compiler that make easy to understand and to write the program. Basically, main source file has two sections: declaration and program body. Declarations should be in their proper place in the code, organized in an orderly manner. Otherwise, compiler may not be able to comprehend the program correctly.

For the beginning of this program, all variable was declared to make sure all variable can be used.

```
program password
include kp_conv

symbol lamp_1 = portb.0
symbol lamp_2 = portb.1
symbol lamp_3 = portc.6
symbol lamp_4 = portc.7
symbol timer_off = portc.4            'input from slave
symbol timer_on = portc.2             'output to slave
dim cnt as byte
dim txt as string[5]
dim digit as byte[4]
dim i as byte
dim pass_ok as boolean ' will hold TRUE is password is OK, otherwise FALSE
dim temp_res as word
dim t_old,t1,t2,t3,t4,t,ch,ch1,ch2_,ch3,ch4,ch5,ch6,ch7,ch8 as word
dim t_old2,t5,t6,t7,t8,t_2,ch2 as word
dim Text as char[17]
dim Text_2 as char [17]
dim tlong, tlong2 as longint
const password as byte[4] = (1,2,3,5)

```

**Figure 3.8:** Variable declaration

From this coding, port b0, b1, c6 and c7 was declared as a lamp_1, lamp_2, lamp_3 and lamp_4. And constant password was declared as “1, 2, 3, 4” on keypad press.

### 3.4.1.1 Initialize LCD and keypad

```
Keypad_Init(PORTD)
Lcd_Init(PORTB)          ' Initialize LCD on PORTB
Lcd_Cmd(LCD_CLEAR)       ' Clear display
Lcd_Cmd(LCD_CURSOR_OFF)  ' Cursor off
Lcd_Out(1,1,"Insert Password")
```

**Figure 3.9:** Initialize LCD and keypad

From this command, LCD was initialized at port B and keypad at port D. since the port has been declared, the LCD screen shows ‘Insert password’ that will remind the user to enter the password.

### 3.4.1.2 Password program

```
next i

    pass_ok = FALSE ' assume that the passwords don't match
    if digit[0] = password[0] then
       if digit[1] = password[1] then
          if digit[2] = password[2] then
             if digit[3] = password[3] then
                pass_ok = TRUE ' passwords match
             end if
          end if
       end if
     end if

    ' output text on LCD
    if pass_ok=TRUE then
        lcd_cmd(lcd_clear)
        Lcd_Out(1,1,"ACCESS GRANTED")
        delay_ms(1500)
        lcd_cmd(lcd_clear)
        goto select_type
    else
        Lcd_out(1,1,"ACCESS DENIED")
        delay_ms(1500)
        lcd_cmd(lcd_clear)
        goto main
    end if
wend
```

**Figure 3.10:** Password coding

The password program contains 4 digits which mean that user must enter 4 numbers before login the system. If the numbers were selected are correct then the program will go to the select type of egg. That mean user must select egg type that want to incubate. If the password selected are wrong, so the system will back to top that mean user must enter the password again.

### 3.4.1.3 Sensor program

```
t1  = ADC_read(0)                ' get ADC value from channel 0
delay_ms(15)                         ' wait for a new value
t2  = ADC_read(0)
delay_ms(15)
t3  = ADC_read(0)
delay_ms(15)
t4  = ADC_read(0)
t = (t1+t2+t3+t4)/4              ' make a average of the 4 reading

   if t_old <> t then              ' if value are different then...
       t_old = t
       tlong = t*5000                  ' check Vref setting 1.2v better
       t     = tlong >> 10
       ch    = t div 1000            ' prepare value for diplay

       ch1    = (t div 1000) mod 10
       lcd_chr(1,7,48+ch1)

       ch2_   = (t div 100) mod 10
       Lcd_Chr(1,8,48+ch2_)

       ch3    = (t div 10) mod 10
       Lcd_Chr(1,9,48+ch3)

       ch4    = t mod 10
       Lcd_Chr(1,11,48+ch4)
       delay_ms(500)
    end if
```

**Figure 3.11:** Sensor program

In this system, sensor was connected at port A0 and A1 that allow the analogue signal into the PIC. The PIC will convert the analogue signal to digital signal since the PIC has ADC internally. According to this program, the PIC will receive 4 analogue data and store it into the variable which is t1, t2, t3 and t4. After that the program will make an average data and store it into t variable. This program will display 4 digit numbers on LCD screen which mean shows the value of current temperature and humidity in Fahrenheit.

```
if ch8 >= 1 then
   if ch5 >= 2 then
      if ch6 >= 0 then
         lamp_1 = 0
         lamp_2 = 0
         lamp_3 = 0
         lamp_4 = 0
      end if
   end if
end if
if ch8 = 1 then
   if ch5 <= 0 then
      if ch6 = 8 then
      lamp_1 = 1
      lamp_2 = 1
      lamp_3 = 1
      lamp_4 = 1
      end if
   end if
end if
```

**Figure 3.12:** Comparing with current value

The value of current temperature and humidity that display on the LCD screen will be compared. If the value equal and above 120F the lamp will off and when the value equal to 108F then the lamp will on this process will always running since it must make sure the condition of the incubator in good condition.

### 3.4.1.4 Lamp program

```
lampON:
trisb.0 = 0
trisb.1 = 0
trisc.6 = 0
trisc.7 = 0

lamp_1 = 0
lamp_2 = 0
lamp_3 = 0
lamp_4 = 0
delay_ms(1000)
lamp_1 = 1
delay_ms(1000)
lamp_2 = 1
delay_ms(1000)
lamp_3 = 1
delay_ms(1000)
lamp_4 = 1
delay_ms(1000)
return
end.
```

**Figure 3.13:** Lamp program

After the egg type was selected, the system will run automatically. The bulbs will on one by one with delay in one second each.

### 3.4.2 Slave control system

In the master code, program was write using microCode studio compiler software. High language was used in this compiler that make easy to understand and to write the program. The code explorer allows you to automatically jump to include files, defines, constants, variables, aliases and modifiers, symbols and labels that are contained within your source code. It's easy to set up the compiler, assembler and programmer options or let MicroCode Studio built in auto search feature. Compilation and assembler errors can easily be identified and corrected using the error results window. MicroCode Studio even comes with a serial communications window.

```
DEFINE OSC 20

DEFINE ccp1_reg portc
DEFINE ccp1_bit 2
DEFINE ccp2_reg portc   'for pwm motor2
DEFINE ccp2_bit 1       'for pwm motor2

ADCON1 = %00001111        'convert analog to digital
led VAR porte.0
control VAR porta.0
B0 VAR BYTE
cw VAR portd.6
ccw VAR portd.7
trisd=0
trisa=1
trise=0
```

**Figure 3.14:** Initialize program

Firstly the program needs to declare the value of crystal usage. In this project, the controller was used 20 kHz crystal which is making the CPU execute the program in frequency 20 kHz.

To control the motor speed, Pulse Width Modulation (PWM) must supply to the DC motor. One of the specialties using PIC, there has provided a pin that supplies the PWM. But need to define first where the pin needs to use. In this program, the PWM was generated at port C since we define it at port C.

All variables must be declared by simple instruction. Variables are where temporary data is stored in a PicBasic Pro program. They are created using the VAR keyword. Variables may be bits, bytes or words. Space for each variable is automatically allocated in the microcontrollers RAM by PBP.

### 3.4.2.1 Main program

```
'MAIN PORGRAM STARTS HERE
main:
    GOSUB TIMERON
    LOW cw
    LOW ccw
mx:
IF TIME[2]=10 THEN
     HIGH cw
     LOW ccw
     HPWM 1,255,1221
     PAUSE 5000
     LOW cw
     LOW ccw
ENDIF
IF TIME[1]=1 AND TIME[2]=20 THEN
    LOW cw
    LOW ccw
    GOSUB TIMEROFF
ENDIF
GOTO mx
```

**Figure 3.15:** Main program

In the main program, the CPU will on the timer since it execute the instruction "TIMERON". The CPU will compare the current timer with reference timer before execute the next instruction. When TIME[2] equal to 10 then DC motor will running since PWM was generated and supply to the DC motor. The motor will run in 5 second with 100% duty cycle and after that the motor will off. The timer will off when the CPU executes the instruction "TIMEROFF".

#### 3.4.2.2 Timer

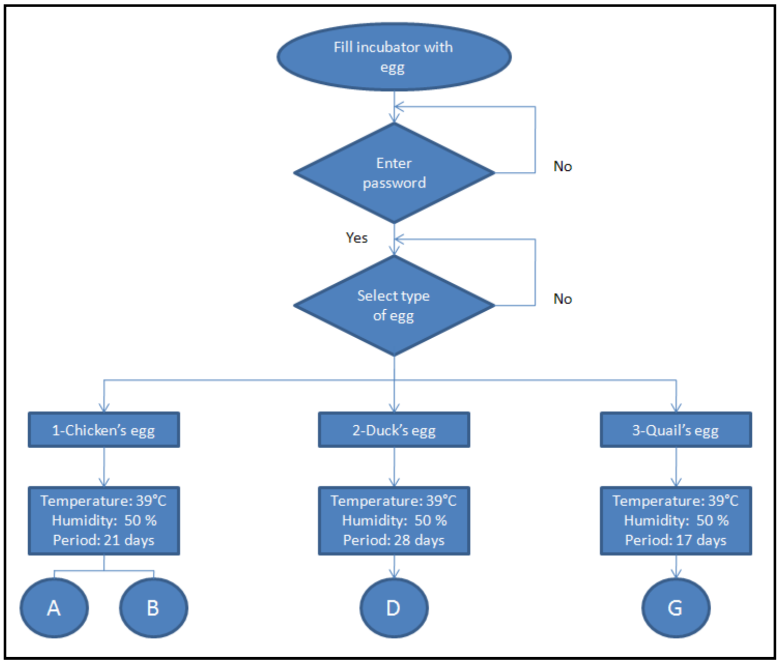
```
InitData:
    DATE[0]=19   'DAY
    DATE[1]=10   'MONTH
    DATE[2]=2006'YEAR
    TIME[0]=0    'HOUR
    TIME[1]=0    'MINUTE
    TIME[2]=0    'SECOND
    TIMEMS=0
RETURN

TimeChg:
    TIMEMS=TIMEMS+1
    IF TIMEMS=100 THEN
        IF TIME[2]=59 THEN
            TIME[2]=0
            IF TIME[1]=59 THEN
                TIME[1]=0
                IF TIME[0]=23 THEN
                    TIME[0]=0
                    GOSUB   TimeChgDate
                ELSE
                    TIME[0]=TIME[0]+1
                ENDIF
            ELSE
            TIME[1]=TIME[1]+1
            ENDIF
        ELSE
            TIME[2]=TIME[2]+1
        ENDIF
        TIMEMS=0
    ENDIF
RETURN
```

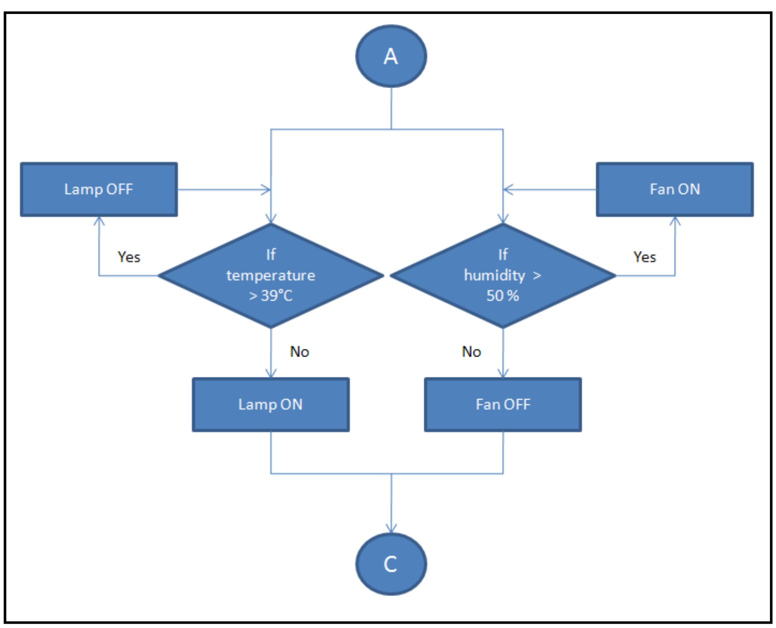
**Figure 3.16:** Timer

The timer program was declared with the special protocol that may the timer function in real time. The timer will change in second, minute, hour, day, month and year. That has been declared such as TIME[2] = second, TIME[1] = minute, TIME[0] = hour, DATE[2] = day, DATE[1] = month and DATE[0] = year.

### 3.4.3 System execution

The entire program has been combining together to execute all the system. The system will run with the following flow chat shows in Figure 3.17:



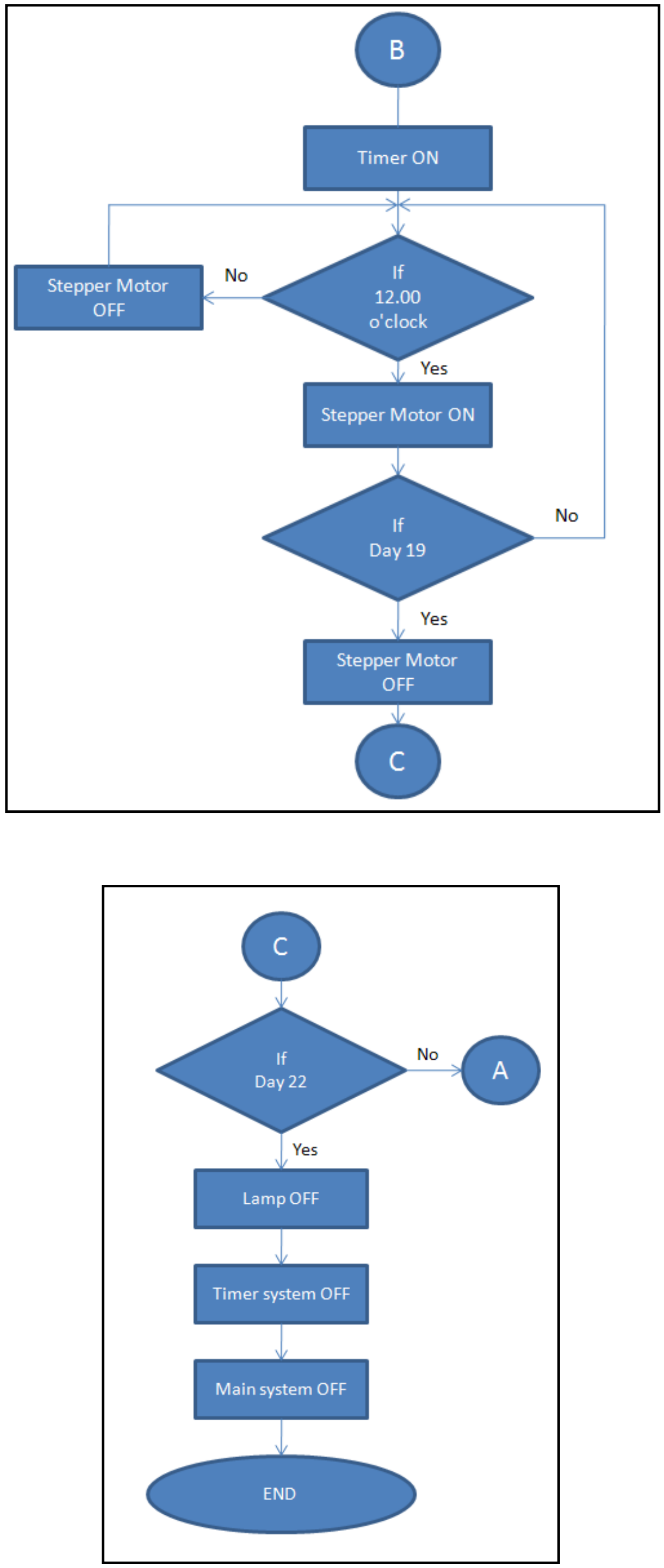



**Figure 3.17:** Flow system execution

Firstly the user must enter the correct password [1, 2, 3, 4]. After that, user must select the type of egg. And then the system will run automatically. The characteristic for all egg type is difference depends on the table that shows in table 1.1. Besides, the timer will on and running depend on period that has been set up in the system. The sensor will always measure the current temperature and humidity that produce from bulb and water. If the temperature over the reference value, all bulb will off until the temperature in incubator was stable. Then if the humidity over the reference value, the fan will on to keep the incubator in good air flow and good ventilation. When the timer reaches the hatching period, the system will automatically off. Note: for the D and G, the process is same with A and B.

# CHAPTER 4

# PROJECT RESULT AND DISCUSSION

## 4.1 Result

By the explanation in the chapter 3, the system has been running properly but not be testing with the actual period. The result has summarized such as below:

- System will activate after user enter the security password
- All 4 lamps [240v] will on after user select the specific type off egg. It will heat the egg.
- The timer of the system will activate the egg period depends on types of egg were selected.
- Temperature and humidity sensor will measure the condition off incubator and display the value on LCD screen.
- If temperature value over the requirement value, all lamp will off. If temperature value below than requirement value, all lamp will on.
- If humidity value over the requirement value, the fan will on. If humidity value below than requirement value, the fan will off.
- The DC motors will running every 12.00 o'clock every day until end of the period. Motor will run in 5 second to change the position of egg.

## 4.2 Problem and troubleshooting

In this project, there are some problems that faced. It may categorize in three parts. There are hardware part, software part and mechanical part. To solve the problem have four steps used. Firstly the entire problem must be list down on the paper which is containing where the problem occurred. Secondly its need to do list downs all assumption about the sources that related to the problem. Than do an investigation and experiment depends on the assumption that has been listed. Finally, the sources have found and fix it with the best solution. Than the output must be control and maintain to make sure all the system function smoothly.

### 4.2.1 Hardware part

Three main problems occurred during assemble the hardware part shows in the Table 4.1 below. For the first problem is the stepper motor that used before does not function properly and low torque. It is not suitable to rolling the egg roller. Then to solve the problem by change the stepper motor with the DC motor that has high torque and constant speed. Secondly the problem that found is the output of the master controller is not stable. It is occurred because the main supply was shared with the dc motor power supply. From an inspection, the DC motor will use high current to energize the internal coil and need the high torque to run the motor. It will affect the current that supply to the master control circuit. Then its need to separate the power supply between master control circuit and DC motor. And the last thing is about the relay does not function properly to on the lamp. This is because of one of voltage regulator cannot support for more than two relays. So its need to use two voltage regulators to support four relays. Finally the entire problems have been fixed and the system work properly.

**Table 4.1:** Problem and troubleshooting in hardware part

| **Problem** | **Troubleshooting** |
|---|---|
| Stepper motor does not function smoothly | Replace the stepper motor with DC motor and use the DC motor driver. |
| Voltage does not enough to execute the main system | Used separate power supply between master controller board and DC motor |
| Relays does not function | Use separate two voltage regulator for four relays |

### 4.2.2 Software part

In the software part, mostly its not have much difficult to troubleshoot the flow system. Basically the software that used will shows where the wrong instruction usage after compile the program. Firstly it's a troubling to display the word at LCD screen. Will review the information at internet, to show the word at LCD screen must use American Standard Code for Information Interchange (ASCII code). The appendix E will show the character and code. The second thing is difficult to combine all program with a timer program. The system does not run as expected. To solve this problem must used two controlled named as master and slave controller circuit.

**Table 4.2:** Problem and troubleshooting in software part

| Problem | Troubleshooting |
|---|---|
| Coding that write using microBasic does not functioning the LCD display | Correctly coding by change their ACSII code. |
| Timer problem | Using another PIC 18F4550 to controller the timer and stepper motor. That such a slave controller to a master controller. |

### 4.2.3 Mechanical part

There are three main difficulties in mechanical part. Firstly it's about egg roller did not rolling smoothly. Before this its try to use a wood for a roller case but it is not suitable because the iron rode rotate smoothly. So its need a bearing to connect with the iron rode. And then gear and small belting are use to roll the iron rode. It is difficult to find the gear and belting because they only can be found by customized at factory.

**Table 4.3:** Problem and troubleshooting in mechanical part

| Problem | Troubleshooting |
|---|---|
| An egg roller does not rolling smoothly | Change the material by using iron rode and bearing to built egg roller |
| Wood is not suitable to built the casing of egg roller | Change the wood with aluminum |

# CHAPTER 5

# CONCLUSION AND RECOMMENDATION

## 5.1 Conclusion

The system of smart egg incubator will incubate the various types of egg. The system will automatically control the temperature and humidity since to maintain the ventilation of incubator. The system was added with the security system from other outside disturbance. This is because, if the system was disturbed it will affect the growth of egg. This system will make the user easily friendly since it can move to other place. However, the system still need to upgrade little bit to more precise and need to upgrade their facilities.

## 5.2 Recommendation

There are few recommendations to improve this project:

i. The system needs backup power supply. It is for prevent the blackout or shot circuit.
ii. It can be added with touch screen since to make the incubator in user friendly with the GUI system (Graphic User Interface).
iii. In this project, the system has been secured but the security system can be added at drover to prevent egg from stolen.
iv. The system can be monitor with the visual basic system.
v. The system need to upgrade to allow much type of egg can be incubated.

## 5.3 Costing & Commercialization

Since this project can help the farmer to gain their product, it can be patterned and commercialize but this project must be upgrade to make more user friendly and the system must be more accurate and efficiency. SEIS are purposely designed for poultry farm industries, educational and for new comer businessman in poultry farm field. Overall costing of this project is RM 450.00.

# REFERENCES


[1] 14th March 2008, Citing Internet Sources, Avian Incubation, URL http://en.wikipedia.org/wiki/Avian_incubation

[2] 22th April 2008, Citing Internet Sources, Build Your Own Incubator, URL http://www.motherearthnews.com/DIY/1982-03-01/Build-Your-Own-Incubator.aspx

[3] 3rd May 2008, Citing Internet Sources, Care and Incubation of Hatching Eggs, URL http://www.msstate.edu/dept/poultry/hatch.htm

[4] 18th May2008, Citing Internet Sources, ASCIII Table and Description, URL http://www.asciitable.com/

[5] 30th March2008, Citing Internet Sources, PIC Tutorial Hardware, URL http://www.winpicprog.co.uk/pic_tutorial_extras.htm

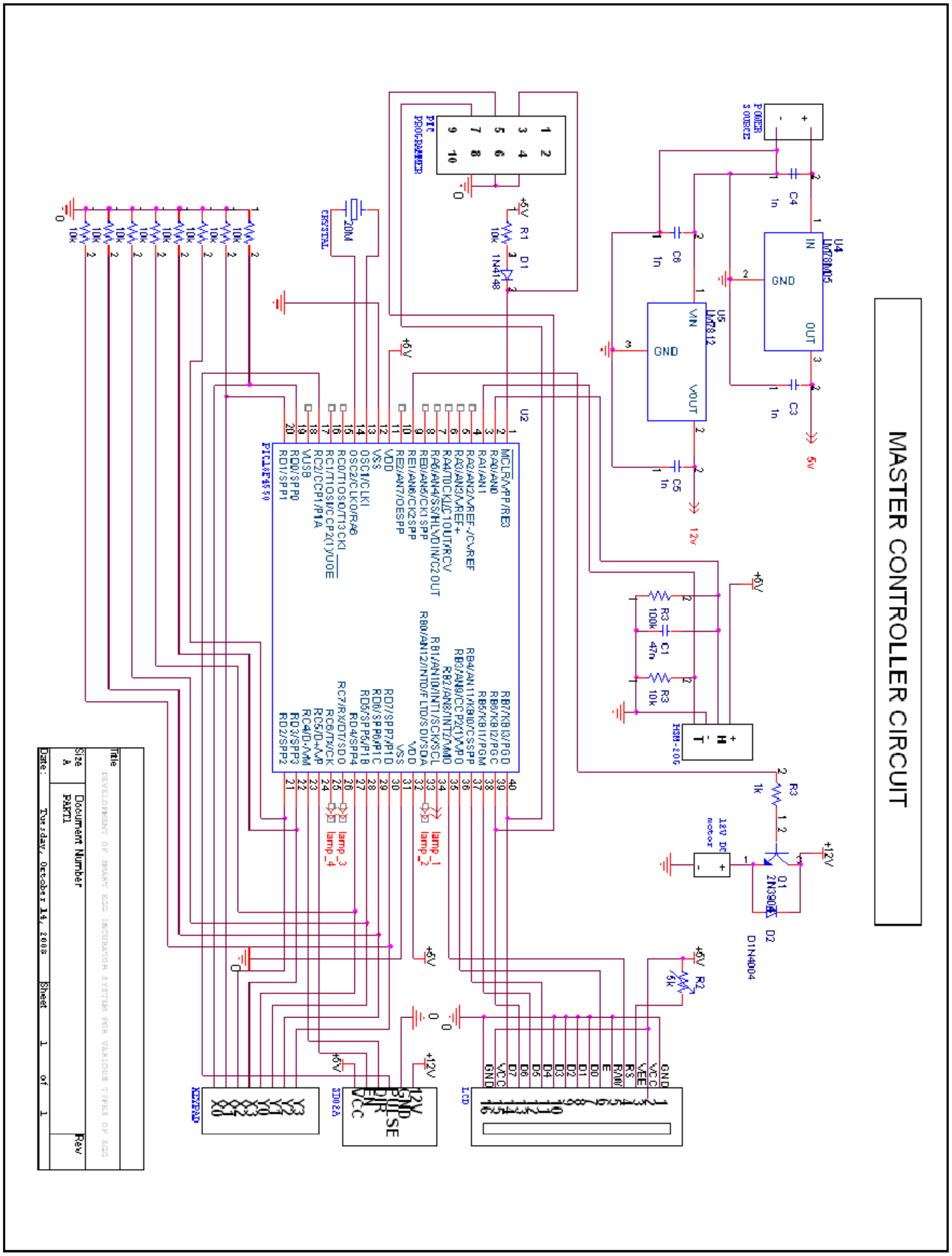
[6] 7th July 2008, Citing Internet Sources, MEL PICBASIC Forum**,** URL http://www.picbasic.co.uk/forum/

[7] 20th September 2008, Citing Internet Sources, mikroElektronika Forum**,** URL http://www.mikroe.com/forum/

[8] 4th May 2008, Citing Internet Sources, Duck Egg Incubator Facts, URL http://www.duckeggs.com/duck-egg-incubator.html

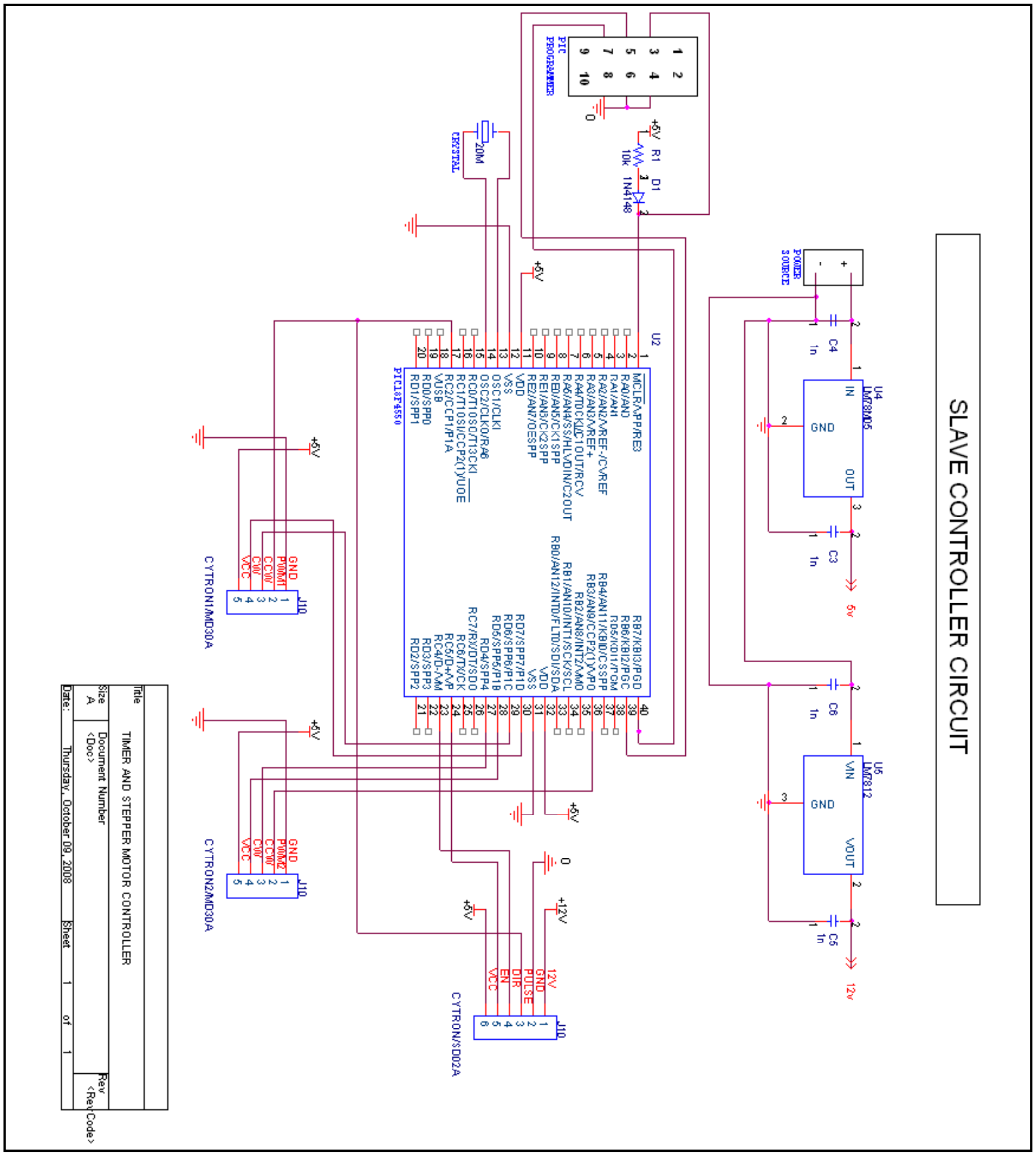
# APPENDIX A

## Master Controller Circuit



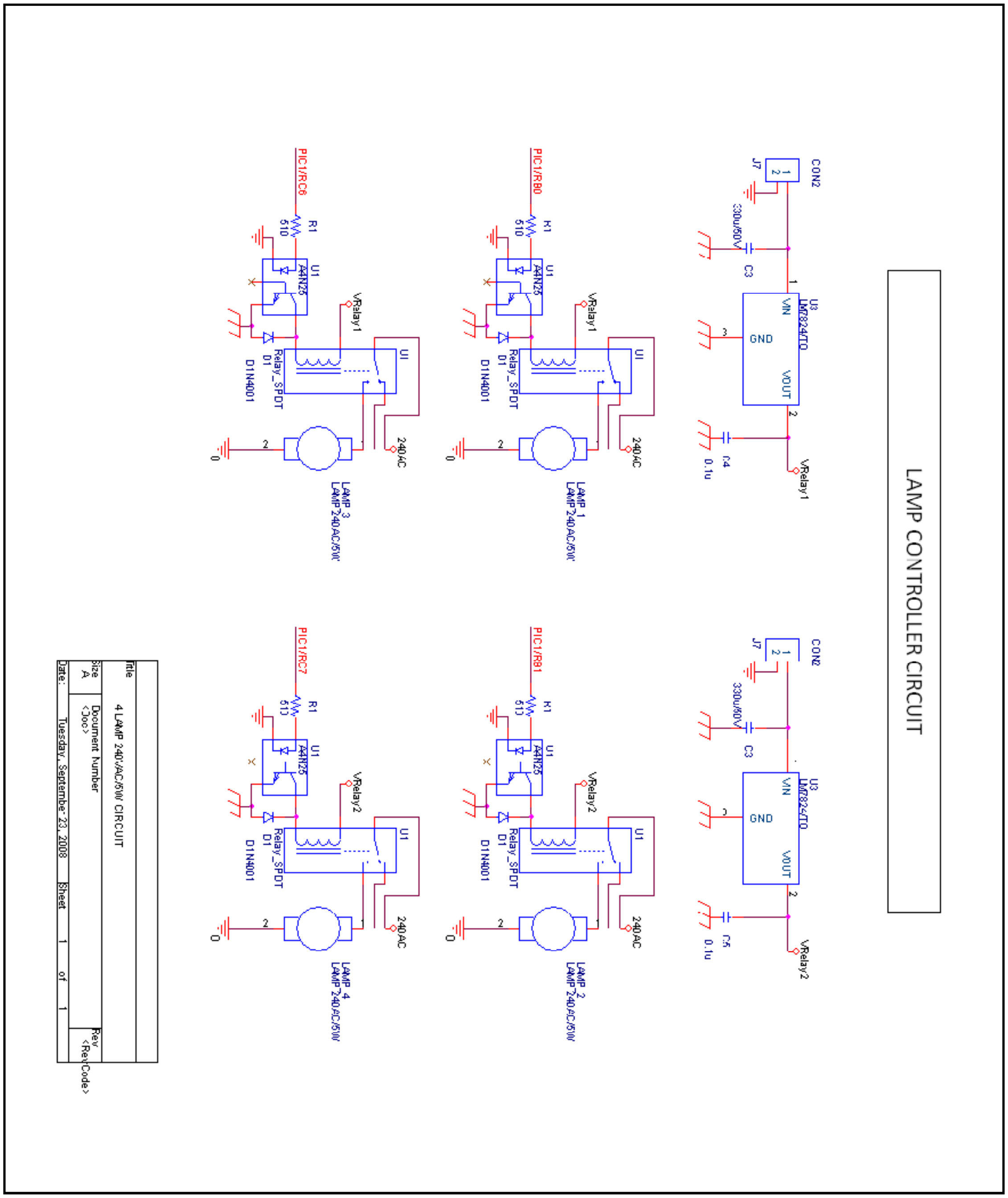
# APPENDIX B

## Slave Controller Circuit



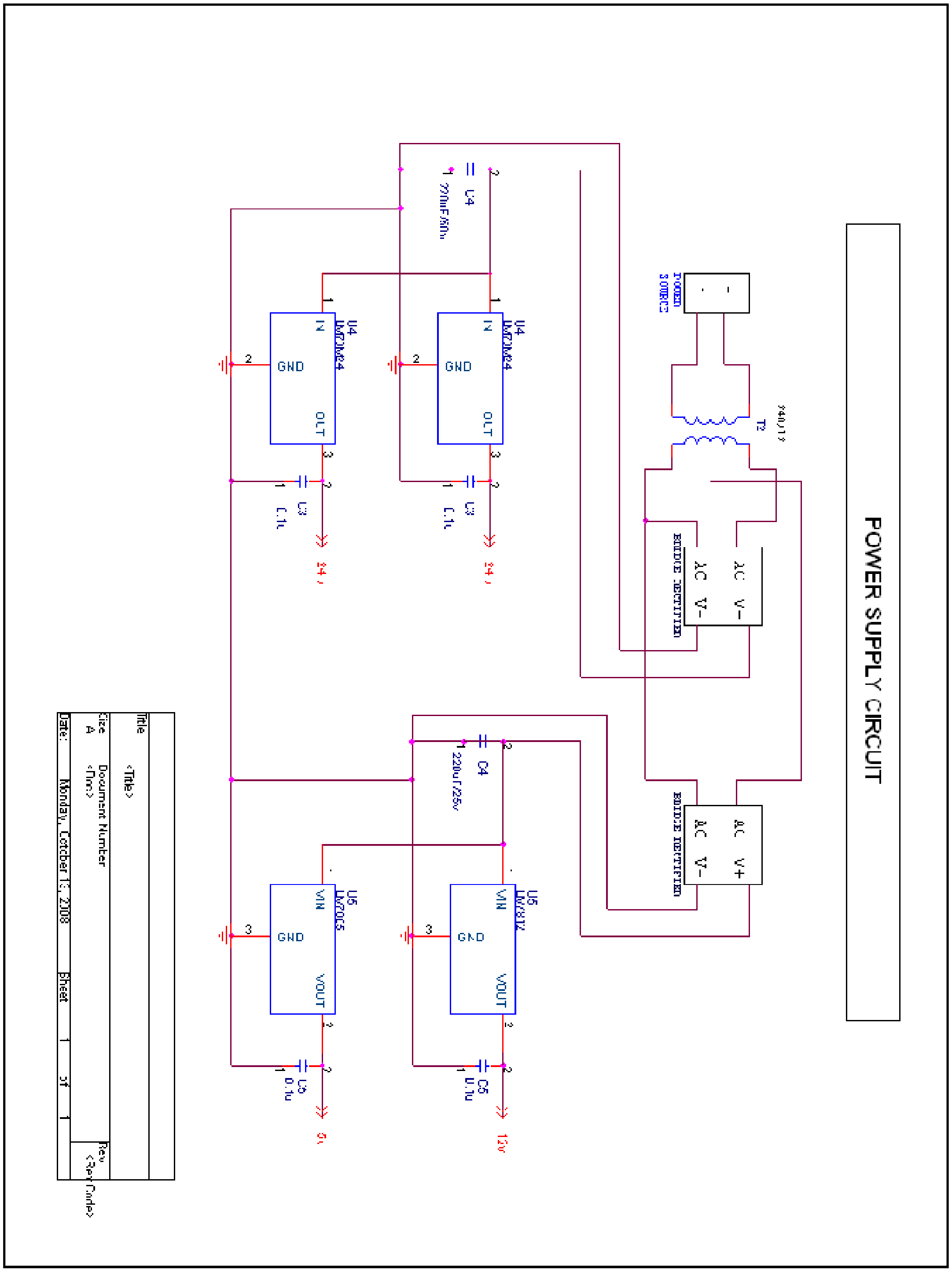
# APPENDIX C

## Lamp Controller Circuit



# APPENDIX D

## Power Supply Circuit



# APPENDIX E

## ASCIII Table and Description

### ASCII Table and Description

ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort. ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose. Below is the ASCII character table and this includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

| Dec | Hx | Oct | Char | | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL | (null) | 32 | 20 | 040 |   | Space | 64 | 40 | 100 | @ | @ | 96 | 60 | 140 | ` | ` |
| 1 | 1 | 001 | SOH | (start of heading) | 33 | 21 | 041 | ! | ! | 65 | 41 | 101 | A | A | 97 | 61 | 141 | a | a |
| 2 | 2 | 002 | STX | (start of text) | 34 | 22 | 042 | " | " | 66 | 42 | 102 | B | B | 98 | 62 | 142 | b | b |
| 3 | 3 | 003 | ETX | (end of text) | 35 | 23 | 043 | # | # | 67 | 43 | 103 | C | C | 99 | 63 | 143 | c | c |
| 4 | 4 | 004 | EOT | (end of transmission) | 36 | 24 | 044 | $ | $ | 68 | 44 | 104 | D | D | 100 | 64 | 144 | d | d |
| 5 | 5 | 005 | ENQ | (enquiry) | 37 | 25 | 045 | % | % | 69 | 45 | 105 | E | E | 101 | 65 | 145 | e | e |
| 6 | 6 | 006 | ACK | (acknowledge) | 38 | 26 | 046 | & | & | 70 | 46 | 106 | F | F | 102 | 66 | 146 | f | f |
| 7 | 7 | 007 | BEL | (bell) | 39 | 27 | 047 | ' | ' | 71 | 47 | 107 | G | G | 103 | 67 | 147 | g | g |
| 8 | 8 | 010 | BS | (backspace) | 40 | 28 | 050 | ( | ( | 72 | 48 | 110 | H | H | 104 | 68 | 150 | h | h |
| 9 | 9 | 011 | TAB | (horizontal tab) | 41 | 29 | 051 | ) | ) | 73 | 49 | 111 | I | I | 105 | 69 | 151 | i | i |
| 10 | A | 012 | LF | (NL line feed, new line) | 42 | 2A | 052 | * | * | 74 | 4A | 112 | J | J | 106 | 6A | 152 | j | j |
| 11 | B | 013 | VT | (vertical tab) | 43 | 2B | 053 | + | + | 75 | 4B | 113 | K | K | 107 | 6B | 153 | k | k |
| 12 | C | 014 | FF | (NP form feed, new page) | 44 | 2C | 054 | , | , | 76 | 4C | 114 | L | L | 108 | 6C | 154 | l | l |
| 13 | D | 015 | CR | (carriage return) | 45 | 2D | 055 | - | - | 77 | 4D | 115 | M | M | 109 | 6D | 155 | m | m |
| 14 | E | 016 | SO | (shift out) | 46 | 2E | 056 | . | . | 78 | 4E | 116 | N | N | 110 | 6E | 156 | n | n |
| 15 | F | 017 | SI | (shift in) | 47 | 2F | 057 | / | / | 79 | 4F | 117 | O | O | 111 | 6F | 157 | o | o |
| 16 | 10 | 020 | DLE | (data link escape) | 48 | 30 | 060 | 0 | 0 | 80 | 50 | 120 | P | P | 112 | 70 | 160 | p | p |
| 17 | 11 | 021 | DC1 | (device control 1) | 49 | 31 | 061 | 1 | 1 | 81 | 51 | 121 | Q | Q | 113 | 71 | 161 | q | q |
| 18 | 12 | 022 | DC2 | (device control 2) | 50 | 32 | 062 | 2 | 2 | 82 | 52 | 122 | R | R | 114 | 72 | 162 | r | r |
| 19 | 13 | 023 | DC3 | (device control 3) | 51 | 33 | 063 | 3 | 3 | 83 | 53 | 123 | S | S | 115 | 73 | 163 | s | s |
| 20 | 14 | 024 | DC4 | (device control 4) | 52 | 34 | 064 | 4 | 4 | 84 | 54 | 124 | T | T | 116 | 74 | 164 | t | t |
| 21 | 15 | 025 | NAK | (negative acknowledge) | 53 | 35 | 065 | 5 | 5 | 85 | 55 | 125 | U | U | 117 | 75 | 165 | u | u |
| 22 | 16 | 026 | SYN | (synchronous idle) | 54 | 36 | 066 | 6 | 6 | 86 | 56 | 126 | V | V | 118 | 76 | 166 | v | v |
| 23 | 17 | 027 | ETB | (end of trans. block) | 55 | 37 | 067 | 7 | 7 | 87 | 57 | 127 | W | W | 119 | 77 | 167 | w | w |
| 24 | 18 | 030 | CAN | (cancel) | 56 | 38 | 070 | 8 | 8 | 88 | 58 | 130 | X | X | 120 | 78 | 170 | x | x |
| 25 | 19 | 031 | EM | (end of medium) | 57 | 39 | 071 | 9 | 9 | 89 | 59 | 131 | Y | Y | 121 | 79 | 171 | y | y |
| 26 | 1A | 032 | SUB | (substitute) | 58 | 3A | 072 | : | : | 90 | 5A | 132 | Z | Z | 122 | 7A | 172 | z | z |
| 27 | 1B | 033 | ESC | (escape) | 59 | 3B | 073 | ; | ; | 91 | 5B | 133 | [ | [ | 123 | 7B | 173 | { | { |
| 28 | 1C | 034 | FS | (file separator) | 60 | 3C | 074 | < | < | 92 | 5C | 134 | \ | \ | 124 | 7C | 174 | | | \| |
| 29 | 1D | 035 | GS | (group separator) | 61 | 3D | 075 | = | = | 93 | 5D | 135 | ] | ] | 125 | 7D | 175 | } | } |
| 30 | 1E | 036 | RS | (record separator) | 62 | 3E | 076 | > | > | 94 | 5E | 136 | ^ | ^ | 126 | 7E | 176 | ~ | ~ |
| 31 | 1F | 037 | US | (unit separator) | 63 | 3F | 077 | ? | ? | 95 | 5F | 137 | _ | _ | 127 | 7F | 177 | &#127; | DEL |

Source: www.LookupTables.com

**APPENDIX F**

**Datasheet of PIC18F4550**

MICROCHIP

# PIC18F2455/2550/4455/4550 Data Sheet

## 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology

# MICROCHIP PIC18F2455/2550/4455/4550

## 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology

### Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 endpoints (16 bidirectional)
- 1-Kbyte dual access RAM for USB
- On-chip USB transceiver with on-chip voltage regulator
- Interface for off-chip USB transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 µA typical
- Sleep mode currents down to 0.1 µA typical
- Timer1 oscillator: 1.1 µA typical, 32 kHz, 2V
- Watchdog Timer: 2.1 µA typical
- Two-Speed Oscillator Start-up

### Flexible Oscillator Structure:

- Four Crystal modes including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal oscillator block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Dual oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three external interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 6.25 ns ($T_{CY}/16$)
  - Compare is 16-bit, max. resolution 100 ns ($T_{CY}$)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-Shutdown and Auto-Restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I$^2$C™ Master and Slave modes
- 10-bit, up to 13-channels Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators with input multiplexing

### Special Microcontroller Features:

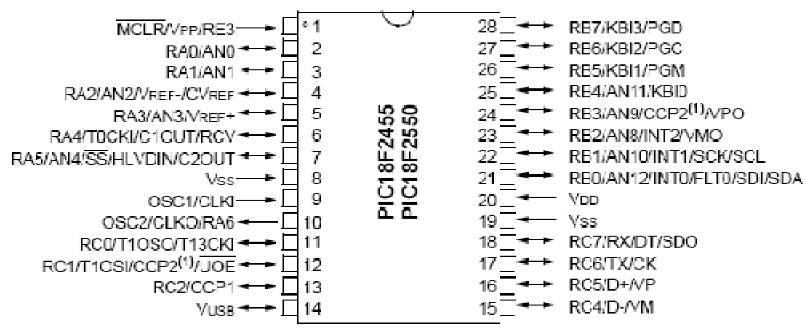
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide operating voltage range (2.0V to 5.5V)

| Device | Program Memory | | Data Memory | | I/O | 10-bit A/D (ch) | CCP/ECCP (PWM) | SPP | MSSP | | EAUSART | Comparators | Timers 8/16-bit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Flash (bytes) | # Single-Word Instructions | SRAM (bytes) | EEPROM (bytes) | | | | | SPI™ | Master I$^2$C™ | | | |
| PIC18F2455 | 24K | 12288 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F2550 | 32K | 16384 | 2048 | 256 | 24 | 10 | 2/0 | No | Y | Y | 1 | 2 | 1/3 |
| PIC18F4455 | 24K | 12288 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |
| PIC18F4550 | 32K | 16384 | 2048 | 256 | 35 | 13 | 1/1 | Yes | Y | Y | 1 | 2 | 1/3 |

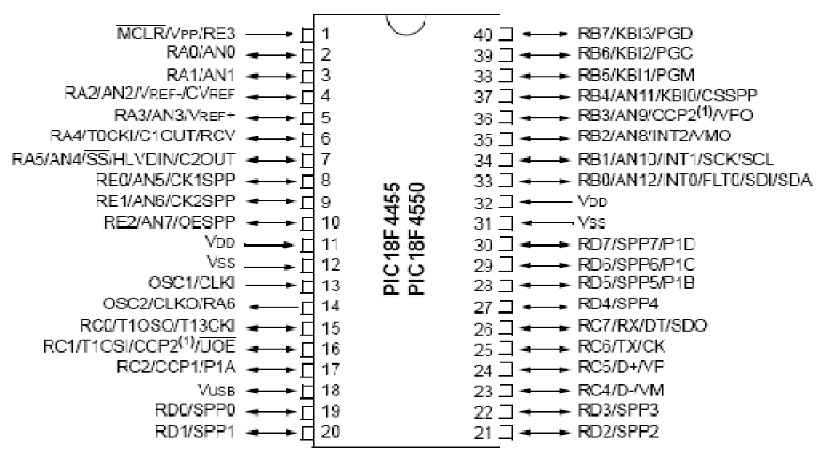
# PIC18F2455/2550/4455/4550

## Pin Diagrams

### 28-Pin PDIP, SOIC




### 40-Pin PDIP



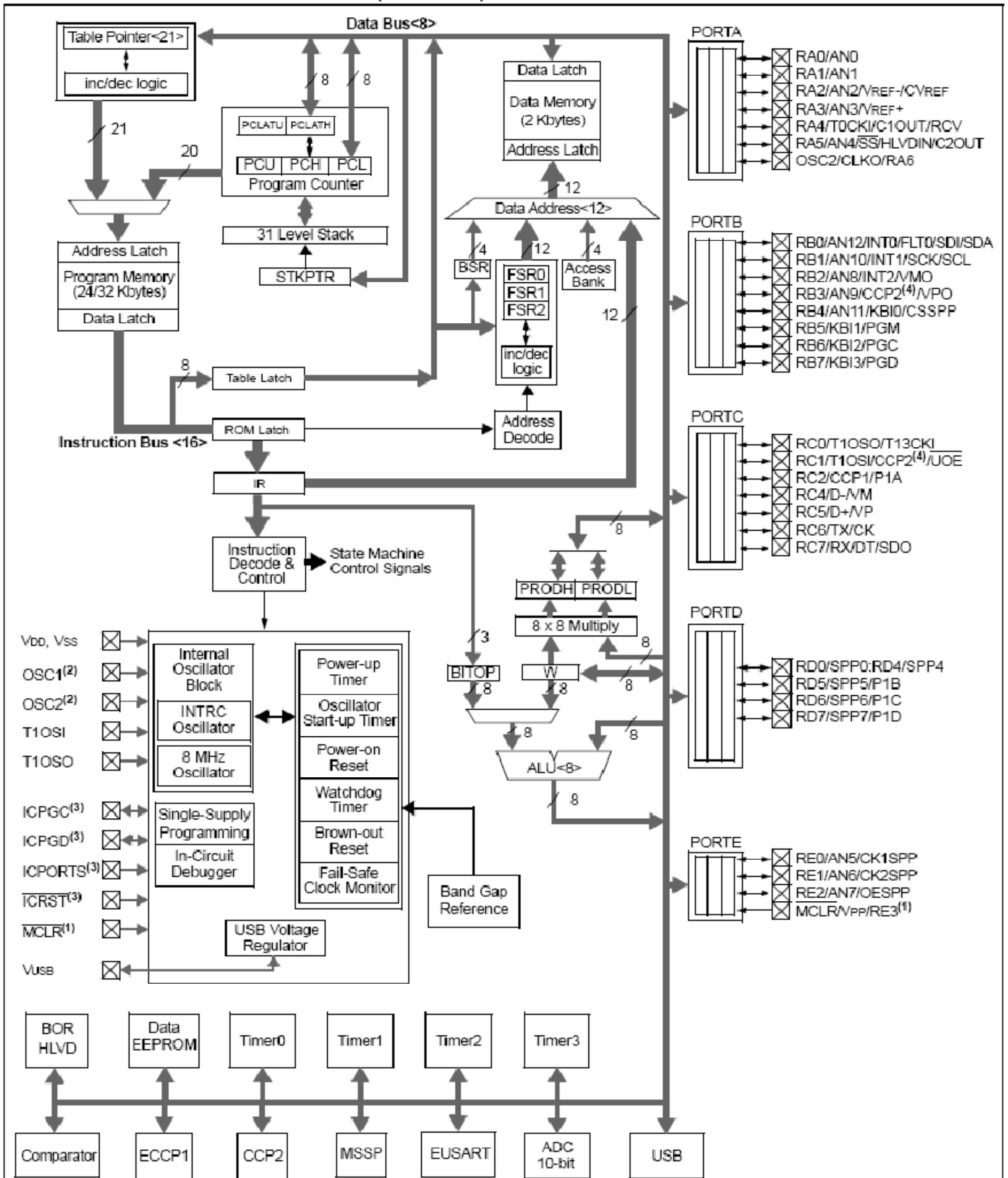

**Note 1:** RB3 is the alternate pin for CCP2 multiplexing.

TABLE 1-1: DEVICE FEATURES

| Features | PIC18F2455 | PIC18F2550 | PIC18F4455 | PIC18F4550 |
|---|---|---|---|---|
| Operating Frequency | DC – 48 MHz | DC – 48 MHz | DC – 48 MHz | DC – 48 MHz |
| Program Memory (Bytes) | 24576 | 32768 | 24576 | 32768 |
| Program Memory (Instructions) | 12288 | 16384 | 12288 | 16384 |
| Data Memory (Bytes) | 2048 | 2048 | 2048 | 2048 |
| Data EEPROM Memory (Bytes) | 256 | 256 | 256 | 256 |
| Interrupt Sources | 19 | 19 | 20 | 20 |
| I/O Ports | Ports A, B, C, (E) | Ports A, B, C, (E) | Ports A, B, C, D, E | Ports A, B, C, D, E |
| Timers | 4 | 4 | 4 | 4 |
| Capture/Compare/PWM Modules | 2 | 2 | 1 | 1 |
| Enhanced Capture/ Compare/PWM Modules | 0 | 0 | 1 | 1 |
| Serial Communications | MSSP, Enhanced USART | MSSP, Enhanced USART | MSSP, Enhanced USART | MSSP, Enhanced USART |
| Universal Serial Bus (USB) Module | 1 | 1 | 1 | 1 |
| Streaming Parallel Port (SPP) | No | No | Yes | Yes |
| 10-bit Analog-to-Digital Module | 10 Input Channels | 10 Input Channels | 13 Input Channels | 13 Input Channels |
| Comparators | 2 | 2 | 2 | 2 |
| Resets (and Delays) | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT | POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), $\overline{\text{MCLR}}$ (optional), WDT |
| Programmable Low-Voltage Detect | Yes | Yes | Yes | Yes |
| Programmable Brown-out Reset | Yes | Yes | Yes | Yes |
| Instruction Set | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled | 75 Instructions; 83 with Extended Instruction Set enabled |
| Packages | 28-pin PDIP 28-pin SOIC | 28-pin PDIP 28-pin SOIC | 40-pin PDIP 44-pin QFN 44-pin TQFP | 40-pin PDIP 44-pin QFN 44-pin TQFP |

**FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM**




**Note 1:** RE3 is multiplexed with MCLR and is only available when the MCLR Resets are disabled.
**2:** OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Configurations"** for additional information.
**3:** These pins are only available on 44-pin TQFP under certain conditions. Refer to **Section 25.9 "Special ICPORT Features (Designated Packages Only)"** for additional information.
**4:** RB3 is the alternate pin for CCP2 multiplexing.

## 11.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

The T0CON register (Register 11-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 11-1. Figure 11-2 shows a simplified block diagram of the timer module in 16-bit mode.

### REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-bit/16-bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

## 11.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable; its value is set by the PSA and T0PS2:T0PS0 bits (T0CON<3:0>) which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256, in power-of-2 increments, are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF TMR0`, `MOVWF TMR0`, `BSF TMR0`, etc.) clear the prescaler count.

> **Note:** Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

### 11.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed "on-the-fly" during program execution.

## 11.4 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

**TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on page |
|---|---|---|---|---|---|---|---|---|---|
| TMR0L | Timer0 Module Low Byte Register | | | | | | | | 52 |
| TMR0H | Timer0 Module High Byte Register | | | | | | | | 52 |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 51 |
| T0CON | TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS2 | T0PS1 | T0PS0 | 52 |
| TRISA | — | TRISA6[(1)] | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 54 |

**Legend:** — = unimplemented locations, read as '0'. Shaded cells are not used by Timer0.

**Note 1:** RA6 is configured as a port pin based on various primary oscillator modes. When the port pin is disabled, all of the associated bits read '0'.

## 21.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has 10 inputs for the 28-pin devices and 13 for the 40/44-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 21-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 21-2, configures the functions of the port pins. The ADCON2 register, shown in Register 21-3, configures the A/D clock source, programmed acquisition time and justification.

**REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | CHS3 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits

`0000` = Channel 0 (AN0)
`0001` = Channel 1 (AN1)
`0010` = Channel 2 (AN2)
`0011` = Channel 3 (AN3)
`0100` = Channel 4 (AN4)
`0101` = Channel 5 (AN5)(1,2)
`0110` = Channel 6 (AN6)(1,2)
`0111` = Channel 7 (AN7)(1,2)
`1000` = Channel 8 (AN8)
`1001` = Channel 9 (AN9)
`1010` = Channel 10 (AN10)
`1011` = Channel 11 (AN11)
`1100` = Channel 12 (AN12
`1101` = Unimplemented(2)
`1110` = Unimplemented(2)
`1111` = Unimplemented(2)

**Note 1:** These channels are not implemented on 28-pin devices.

**2:** Performing a conversion on unimplemented channels will return a floating input measurement.

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

When ADON = `1`:
`1` = A/D conversion in progress
`0` = A/D Idle

bit 0 **ADON:** A/D On bit

`1` = A/D converter module is enabled
`0` = A/D converter module is disabled

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

**REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0[1] | R/W[1] | R/W[1] | R/W[1] |
|---|---|---|---|---|---|---|---|
| — | — | VCFG1 | VCFG0 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

| PCFG3: PCFG0 | AN12 | AN11 | AN10 | AN9 | AN8 | AN7[2] | AN6[2] | AN5[2] | AN4 | AN3 | AN2 | AN1 | AN0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000[1] | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0001 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0010 | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0011 | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0100 | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0101 | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0110 | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0111[1] | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 1000 | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 1001 | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 1010 | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 1011 | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 1100 | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 1101 | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 1110 | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 1111 | D | D | D | D | D | D | D | D | D | D | D | D | D |

A = Analog input D = Digital I/O

**Note 1:** The POR value of the PCFG bits depends on the value of the PBADEN configuration bit. When PBADEN = 1, PCFG<3:0> = 0000; when PBADEN = 0, PCFG<3:0> = 0111.

**2:** AN5 through AN7 are available only on 40/44-pin devices.

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

## 15.4 PWM Mode

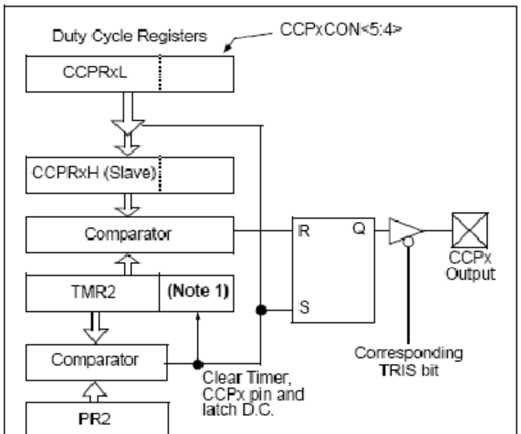
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB or PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

> **Note:** Clearing the CCP2CON register will force the RB3 or RC1 output latch (depending on device configuration) to the default low level. This is not the PORTB or PORTC I/O data latch.

Figure 15-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.4 "Setup for PWM Operation"**.

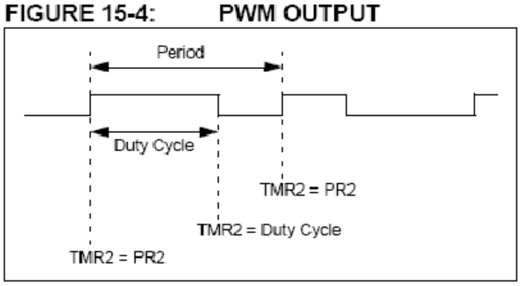
**FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM**




**Note 1:** The 8-bit timer TMR2 value is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create the 10-bit time base.

A PWM output (Figure 15-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 15-4: PWM OUTPUT**




### 15.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

**EQUATION 15-1:**

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH

> **Note:** The Timer2 postscalers (see **Section 13.0 "Timer2 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

### 15.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSbs and the CCPxCON<5:4> contains the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

**EQUATION 15-2:**

$$\text{PWM Duty Cycle} = (\text{CCPRXL:CCPXCON}<5:4>) \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPRxH until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPRxH is a read-only register.

The CCPRxH register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPRxH and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCPx pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

**EQUATION 15-3:**

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)} \text{bits}$$

| **Note:** | If the PWM duty cycle value is longer than the PWM period, the CCPx pin will not be cleared. |
|---|---|

**TABLE 15-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz**

| PWM Frequency | 2.44 kHz | 9.77 kHz | 39.06 kHz | 156.25 kHz | 312.50 kHz | 416.67 kHz |
|---|---|---|---|---|---|---|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | FFh | FFh | FFh | 3Fh | 1Fh | 17h |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.58 |

### 15.4.3 PWM AUTO-SHUTDOWN (CCP1 ONLY)

The PWM auto-shutdown features of the Enhanced CCP module are also available to CCP1 in 28-pin devices. The operation of this feature is discussed in detail in **Section 16.4.7 "Enhanced PWM Auto-Shutdown"**.

Auto-shutdown features are not available for CCP2.

### 15.4.4 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPRxL register and CCPxCON<5:4> bits.
3. Make the CCPx pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCPx module for PWM operation.

# APPENDIX G

## Datasheet of 7805, 7812, 7824

Order this document by MC7800/D

MOTOROLA

# MC7800, MC7800A, LM340, LM340A Series

## Three-Terminal Positive Voltage Regulators

These voltage regulators are monolithic integrated circuits designed as fixed–voltage regulators for a wide variety of applications including local, on–card regulation. These regulators employ internal current limiting, thermal shutdown, and safe–area compensation. With adequate heatsinking they can deliver output currents in excess of 1.0 A. Although designed primarily as a fixed voltage regulator, these devices can be used with external components to obtain adjustable voltages and currents.

- Output Current in Excess of 1.0 A
- No External Components Required
- Internal Thermal Overload Protection
- Internal Short Circuit Current Limiting
- Output Transistor Safe–Area Compensation
- Output Voltage Offered in 2% and 4% Tolerance
- Available in Surface Mount $D^2PAK$ and Standard 3–Lead Transistor Packages
- Previous Commercial Temperature Range has been Extended to a Junction Temperature Range of –40°C to +125°C

### DEVICE TYPE/NOMINAL OUTPUT VOLTAGE

| | | | |
|---|---|---|---|
| MC7805AC<br>LM340AT–5<br>MC7805C<br>LM340T–5 | 5.0 V | MC7812C<br>LM340T–12 | 12 V |
| | | MC7815AC<br>LM340AT–15<br>MC7815C<br>LM340T–15 | 15 V |
| MC7806AC<br>MC7806C | 6.0 V | | |
| MC7808AC<br>MC7808C | 8.0 V | MC7818AC<br>MC7818C | 18 V |
| MC7809C | 9.0 V | MC7824AC<br>MC7824C | 24 V |
| MC7812AC<br>LM340AT–12 | 12 V | | |

### ORDERING INFORMATION

| Device | Output Voltage Tolerance | Operating Temperature Range | Package |
|---|---|---|---|
| MC78XXACT | 2% | $T_J$ = –40° to +125°C | Insertion Mount |
| LM340AT–XX | 2% | | Insertion Mount |
| MC78XXACD2T | 2% | | Surface Mount |
| MC78XXCT | 4% | | Insertion Mount |
| LM340T–XX | 4% | | Insertion Mount |
| MC78XXCD2T | 4% | | Surface Mount |

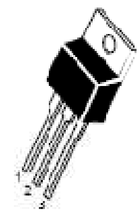
XX indicates nominal voltage.

### THREE–TERMINAL POSITIVE FIXED VOLTAGE REGULATORS

SEMICONDUCTOR TECHNICAL DATA

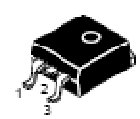
T SUFFIX
PLASTIC PACKAGE
CASE 221A

Heatsink surface connected to Pin 2.



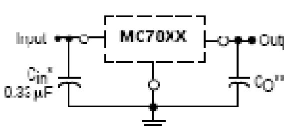

Pin 1. Input
2. Ground
3. Output

D2T SUFFIX
PLASTIC PACKAGE
CASE 936
($D^2PAK$)




Heatsink surface (shown as terminal 4 in case outline drawing) is connected to Pin 2.

### STANDARD APPLICATION




A common ground is required between the input and the output voltages. The input voltage must remain typically 2.0 V above the output voltage even during the low point on the input ripple voltage.

XX, These two digits of the type number indicate nominal voltage.

* $C_{in}$ is required if regulator is located an appreciable distance from power supply filter.

** $C_O$ is not needed for stability; however, it does improve transient response. Values of less than 0.1 µF could cause instability.

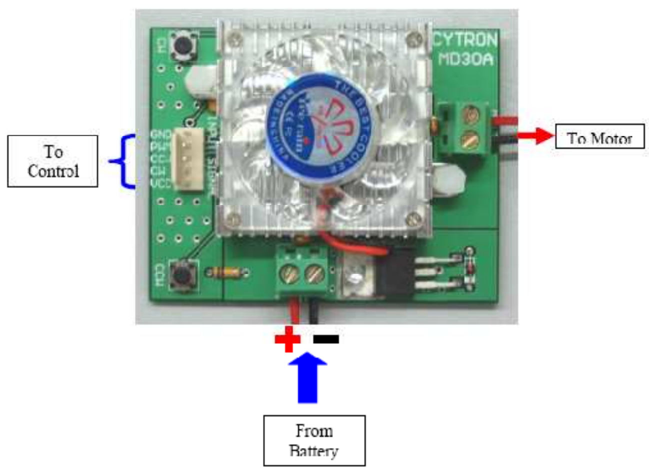
 Rev 5

# APPENDIX H

## Datasheet of MD30A DC motor driver

Cytron Technologies

# CYTRON MD30A
# 30A Motor Driver




### 1. INTRODUCTION AND OVERVIEW

MD30A is designed to drive high current brush motor or application. It is a full bridge motor driver intended for wide range of robotics and automotive applications. The board incorporates most of the components of the typical applications. With minimum interface, the board is ready to be plugged and play. It even includes 2 push buttons for fast test run. Simply add in power, this driver is ready to drive high current motor. Additionally, MD30A have protection of reverse polarity on battery input. It has been designed with capabilities and features of:

- Industrial grade PCB with heavy copper material for **high current applications**
- Each component is soldered properly and tested
- Support up to **30A maximum**
- **Protection against wrong polarity** of $V_{in}$
- 5V logic level inputs
- 5V to 12V compatible for $V_{cc}$
- PWM speed control up to **10KHz**
- Bi-directional control for 1 motor
- Over voltage clamp
- Thermal Shut Down
- Cross-Conduction protection
- Linear current limiter
- Very low standby power consumption
- **Protection against**: Loss of ground and loss of $V_{in}$
- 2 on-board push buttons for fast testing
- Fan heat sink for fast thermal release

## 3. SPECIFICATION

Pin Function Description

| Label | Definition | Function |
|---|---|---|
| $V_{in}$* | Battery Input | Power source for motor. It can be as low as 5.5V and as high as 30V. The driver provides protection against wrong polarity on this input. The PCB pad is designed for terminal block. User may solder terminal block or solder the battery cable directly. |
| Motor | Motor Terminal | Terminal for motor connection. The PCB pad is designed for terminal block. User may solder terminal block or solder the battery cable directly. |
| $V_{cc}$ | Operating supply | Input for driver logic operation. The range is from 5V to 12V |
| CW | Clock Wise | Voltage controller input pin with hysteresis, CMOS compatible. These two pins control the state of the bridge in normal operation according to the truth table 4.1 (stop, brake, clockwise and counterclockwise). Both these pins are pull up internally to $V_{cc}$. Thus by default the motor is brake to $V_{in}$. |
| CCW | Counter Clock Wise | |
| PWM | Pulse Width Modulation | Voltage controlled input pin with hysteresis, CMOS compatible. Gates of Low-Side FETS get modulated by the PWM signal during their ON phase allowing speed control of the motor. |
| Gnd | Ground | Logic ground signal. Internally connected together with $V_{in}$'s ground |

* Although $V_{in}$ can support up to 30V, it is limited to 24V because the maximum voltage of on board fan is 24V.

Absolute Maximum Rating

| Symbol | Parameter | Value | Unit |
|---|---|---|---|
| $V_{in}$ | Motor supply voltage | 24 | V |
| $V_{cc}$ | Operating voltage | 12 | V |
| $I_{max}$ | Maximum Output Current (continuos) | 30 | A |
| $I_R$ | Reserve Output Current (continuos) | 30 | A |
| $I_{in}$ | Logic Input current (CW/CCW) | +/- 10 | mA |
| $I_{pw}$ | PWM Input Current | +/- 10 | mA |
| $T_j$ | Junction Operating Temperature | Internally Limited | °C |
| $T_c$ | Case Operating Temperature | -40 to 150 | °C |
| $T_{STG}$ | Storage Temperature | -55 to 150 | °C |

# APPENDIX I

## Datasheet of HSM-20G Temperature and Humidity sensor



### 2. PRODUCT SPECIFICATION

#### 2.1 The specifications of humidity sensor

| No. | Specification | Humidity Sensor |
|---|---|---|
| 1. | Input voltage range | DC 5.0±0.2 V |
| 2. | Output voltage range | DC 1.0—3.0 V |
| 3. | Measurement Accuracy | ±5% RH |
| 4. | Operating Current (Maximum) | 2mA |
| 5. | Storage RH Range | 0 to 99% RH |
| 6. | Operating RH Range | 20 to 95% (100% RH intermittent) |
| 7. | Transient Condensation | < 3%RH |
| 8. | Temperature Range:<br>- Storage<br>- Operating | <br>20℃ to 70℃<br>0℃ to 50℃ |
| 9. | Hysteresis (RH @ 25℃) | MAX 2%RH |
| 10. | Long Term Stability (typical drift per year) | ±1.5% |
| 11. | Linearity | Linearity |
| 12. | Time Response (63% step change) | 1 min |
| 13. | Dimensions (L*W) | 30mm*22mm |

Table 2.1

#### 2.2 Pin Definitions and Ratings

| Pin | Name | Function |
|---|---|---|
| - | GND | Connects to Ground |
| H | Humudity Output | Voltage analog output. |
| + | Vcc | Connects to Vcc (+5V ) |
| T | Temperature Output | Voltage analog output. |

Note: Please refer Getting Started for the pin connection

Table 2.2

## 3. PRODUCT LAYOUT

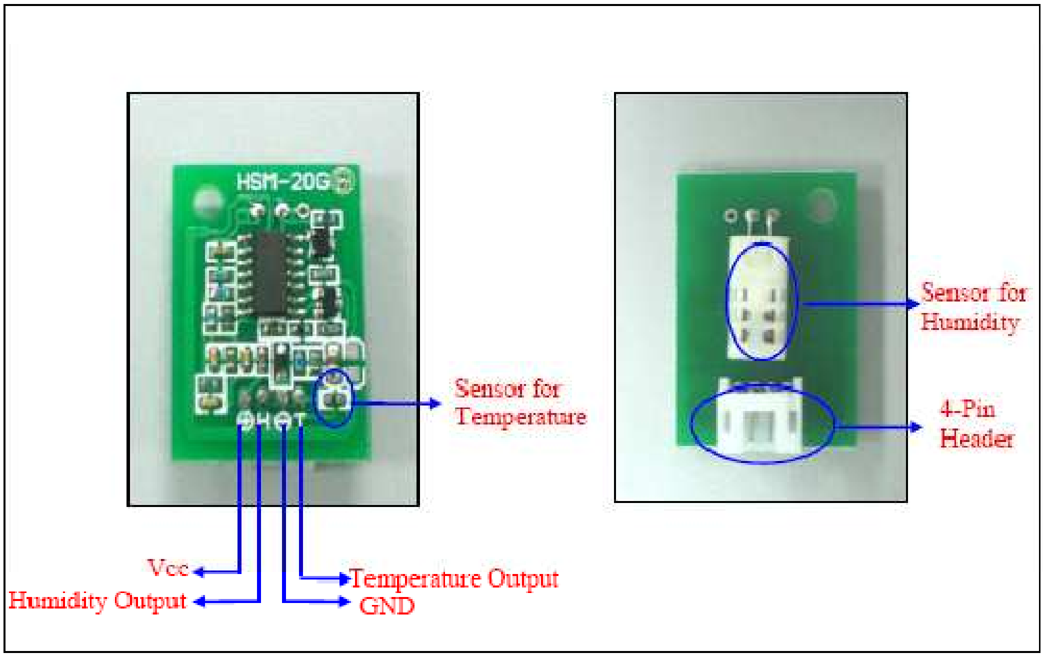



**Figure 3.1**

| Label | Description |
|---|---|
| Vcc | The power supply to the sensor. |
| Humidity Output | The analog output of Humidity for the sensor. |
| GND | The Ground of the sensor. |
| Temperature Output | The analog output of Temperature for the sensor. |
| Sensor for Temperature | The sensor to sense the temperature. |
| Sensor for Humidity | The sensor to sense the humidity. |
| 4-Pin Header | The connector to the cable which connect to testing/microcontroller circuit. |

**Table 3.1**

# APPENDIX J

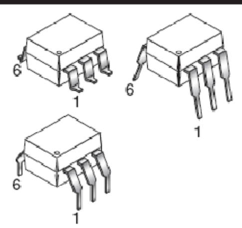
## Datasheet of 4N25 Phototransistor Optocouplers

FAIRCHILD SEMICONDUCTOR®

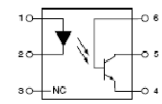
# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

| 4N25 | 4N26 | 4N27 | 4N28 | 4N35 | 4N36 |
|---|---|---|---|---|---|
| 4N37 | H11A1 | H11A2 | H11A3 | H11A4 | H11A5 |

**WHITE PACKAGE (-M SUFFIX)**

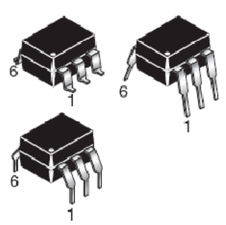


**SCHEMATIC**




PIN 1. ANODE
2. CATHODE
3. NO CONNECTION
4. EMITTER
5. COLLECTOR
6. BASE

**BLACK PACKAGE (NO -M SUFFIX)**



## DESCRIPTION

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a 6-pin dual in-line package.

## FEATURES

- Also available in white package by specifying -M suffix, eg. 4N25-M
- UL recognized (File # E90700)
- VDE recognized (File # 94766)
  - Add option V for white package (e.g., 4N25V-M)
  - Add option 300 for black package (e.g., 4N25.300)

## APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs

FAIRCHILD SEMICONDUCTOR®

# GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

| 4N25 | 4N26 | 4N27 | 4N28 | 4N35 | 4N36 |
|---|---|---|---|---|---|
| 4N37 | H11A1 | H11A2 | H11A3 | H11A4 | H11A5 |

**ABSOLUTE MAXIMUM RATINGS** ($T_A$ = 25°C unless otherwise specified)

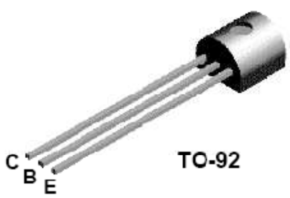
| Parameter | Symbol | Value | Units |
|---|---|---|---|
| **TOTAL DEVICE** | | | |
| Storage Temperature | $T_{STG}$ | -55 to +150 | °C |
| Operating Temperature | $T_{OPR}$ | -55 to +100 | °C |
| Lead Solder Temperature | $T_{SOL}$ | 260 for 10 sec | °C |
| Total Device Power Dissipation @ $T_A$ = 25°C | $P_D$ | 250 | mW |
| Derate above 25°C | | 3.3 (non-M), 2.94 (-M) | |
| **EMITTER** | | | |
| DC/Average Forward Input Current | $I_F$ | 100 (non-M), 60 (-M) | mA |
| Reverse Input Voltage | $V_R$ | 6 | V |
| Forward Current - Peak (300µs, 2% Duty Cycle) | $I_F$(pk) | 3 | A |
| LED Power Dissipation @ $T_A$ = 25°C | $P_D$ | 150 (non-M), 120 (-M) | mW |
| Derate above 25°C | | 2.0 (non-M), 1.41 (-M) | mW/°C |
| **DETECTOR** | | | |
| Collector-Emitter Voltage | $V_{CEO}$ | 30 | V |
| Collector-Base Voltage | $V_{CBO}$ | 70 | V |
| Emitter-Collector Voltage | $V_{ECO}$ | 7 | V |
| Detector Power Dissipation @ $T_A$ = 25°C | $P_D$ | 150 | mW |
| Derate above 25°C | | 2.0 (non-M), 1.76 (-M) | mW/°C |

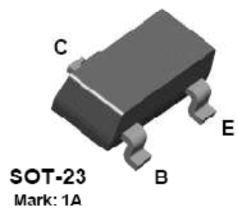
# APPENDIX K

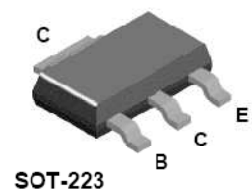
## Datasheet of 2N3904 NPN Amplifier

FAIRCHILD SEMICONDUCTOR ™

**2N3904** **MMBT3904** **PZT3904**




## NPN General Purpose Amplifier

This device is designed as a general purpose amplifier and switch. The useful dynamic range extends to 100 mA as a switch and to 100 MHz as an amplifier.

### Absolute Maximum Ratings*

$T_A$ = 25°C unless otherwise noted

| Symbol | Parameter | Value | Units |
|---|---|---|---|
| $V_{CEO}$ | Collector-Emitter Voltage | 40 | V |
| $V_{CBO}$ | Collector-Base Voltage | 60 | V |
| $V_{EBO}$ | Emitter-Base Voltage | 6.0 | V |
| $I_C$ | Collector Current - Continuous | 200 | mA |
| $T_J$, $T_{stg}$ | Operating and Storage Junction Temperature Range | -55 to +150 | °C |

*These ratings are limting values above which the serviceability of any semiconductor device may be impaired.

NOTES:
1) These ratings are based on a maximum junction temperature of 150 degrees C.
2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

### Thermal Characteristics

$T_A$ = 25°C unless otherwise noted

| Symbol | Characteristic | Max | | | Units |
|---|---|---|---|---|---|
| | | 2N3904 | *MMBT3904 | **PZT3904 | |
| $P_D$ | Total Device Dissipation | 625 | 350 | 1,000 | mW |
| | Derate above 25°C | 5.0 | 2.8 | 8.0 | mW/°C |
| $R_{\theta JC}$ | Thermal Resistance, Junction to Case | 83.3 | | | °C/W |
| $R_{\theta JA}$ | Thermal Resistance, Junction to Ambient | 200 | 357 | 125 | °C/W |

*Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

**Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 $cm^2$.

**APPENDIX L**

**Datasheet of G5LA 24VDC PCB Relay**

OMRON

# PCB Relay | G5LA

**A Cubic, Single-pole 10A Power Relay**

- Small size and light weight
  - 19.6 × 15.6 × 15.6 mm, 7.5g
- High Insulation
  - Dielectric strength 2,000V
  - Withstand impulse voltage 4,500V
- High heat resistance and tracking performance
  - UL class-F available (-CF model)
  - IEC60335 GWT compliant
  - Tracking resistance CTI>250
- Environmental friendly (RoHS compliant)

## Ordering Information

| Contact form | Switching capacity | Model number | |
|---|---|---|---|
| | | Flux protection | Fully sealed |
| SPDT | Standard | G5LA-1 | G5LA-14 |
| | | G5LA-1-CF | G5LA-14-CF |
| | High capacity (NC side) | G5LA-1-E | G5LA-14-E |
| | | G5LA-1-E-CF | G5LA-14-E-CF |
| SPST-NO | | G5LA-1A | G5LA-1A4 |
| | | G5LA-1A-CF | G5LA-1A4-CF |

**Note** : When ordering, add the rated coil voltage to the model number.
Example: G5LA-1 12VDC
Rated coil voltage

### Model Number Legend

G5LA −□□□-□-□ □VDC

**1. Number of Poles**
1: 1pole••

**2. Contact Form**
None: SPDT
A: SPST-NO

**3. Enclosure Ratings**
None: Flux protection
4: Fully sealed

**4. Switching capacity**
None: Standard
E: High capacity (NC side)

**5. UL Insulation System**
None: Standard
CF: Class F

**6. Rated Coil Voltage**

OMRON

# Specifications

## ■ Coil Ratings

| Rated voltage | 5 VDC | 9 VDC | 12 VDC | 24 VDC | 48 VDC |
|---|---|---|---|---|---|
| Rated current | 72 mA | 40 mA | 30 mA | 15 mA | 10 mA |
| Coil resistance | 69.4 Ω | 225 Ω | 400 Ω | 1600 Ω | 4800 Ω |
| Must operate voltage | 75% max. of rated voltage | | | | |
| Must release voltage | 10% min. of rated voltage | | | | |
| Max voltage | 130% of rated voltage at 85• •170% of rated voltage at 23• • | | | | |
| Power consumption (Approx.) | 360 mW | | | | 480 mW |

**Note:** The rated current and coil resistance are measured at a coil temperature of 23°C with a tolerance of ±10%.

## ■ Contact Ratings

| Item | | Standard model | High capacity (-E) model |
|---|---|---|---|
| Contact material | | $AgSnO_2$ | |
| Load | | Resistive load (cosΦ=1) | |
| Rated load | NO | 10A at 250VAC<br>10A at 24VDC | |
| | NO/NC | 5A/5A at 125VAC<br>5A/5A at 24VDC | 5A/5A at 250VAC<br>5A/5A at 24VDC |
| Rated carry current | | 10A(NO), 5A(NC) | 10A |
| Max. switching voltage | | 250VAC, 24VDC | |
| Max. switching current | | 10A(NO), 5A(NC) | 10A |
| Max. switching power | NO | AC2,500VA, DC240W | |
| | NO/NC | AC625VA, DC120W | AC1,250VAC, DC120W |
| Failure rate (reference value) | | 100mA at DC5V | |

**Note:** P level:• $\lambda_{60}$ = 0.1• •$10^{-5}$/operation

## ■ Characteristics

| | |
|---|---|
| Contact resistance | 100 m• •max. |
| Operation time | 10 ms max. |
| Release time | 5 ms max. |
| Max. operating frequency | Mechanical: 18,000 operations/hr<br>Electrical: 1,800 operations/hr (under rated load) |
| Insulation resistance | 1,000 M• •min. (at 500 VDC) |
| Dielectric strength | 2,000 VAC. 1mA 50/60Hz for 1 min between coil and contacts<br>750 VAC 1mA 50/60Hz for 1 min between contacts of same polarity |
| Vibration resistance | Destruction: 10 to 55 Hz, 1.5-mm double amplitude<br>Malfunction: 10 to 55 Hz, 1.5-mm double amplitude |
| Shock resistance | Destruction: 1,000 $m/s^2$ (approx.100G)<br>Malfunction: 100 $m/s^2$ when energized ; 100 $m/s^2$ when no energized |
| Endurance | Mechanical: 10,000,000 operations min.<br>Electrical: 100,000 operations typical |
| Ambient temperature | Operating: -40• •to 85• •(with no icing)<br>Storage: -40• •to 85• •(with no icing) |
| Ambient humidity | Operating: 35% to 85%<br>Storage: 35% to 85% |
| Weight | Approx. 7.5g |

**Note:** Values in the above table are the initial values.

# APPENDIX M

## Datasheet of 16-Button 4 x 4 Matrix Keypad

ELECTRONIX EXPRESS
Tel. (732) 381-8020 • Fax (732) 381-1006
RSR Electronics, Inc.
Tel. (732) 381-8777 • Fax (732) 381-1572

365 Blair Road
Avenel, NJ 07001, U.S.A.
http://www.elexp.com (email: electron@elexp.com)

# Keypad – 16 Button

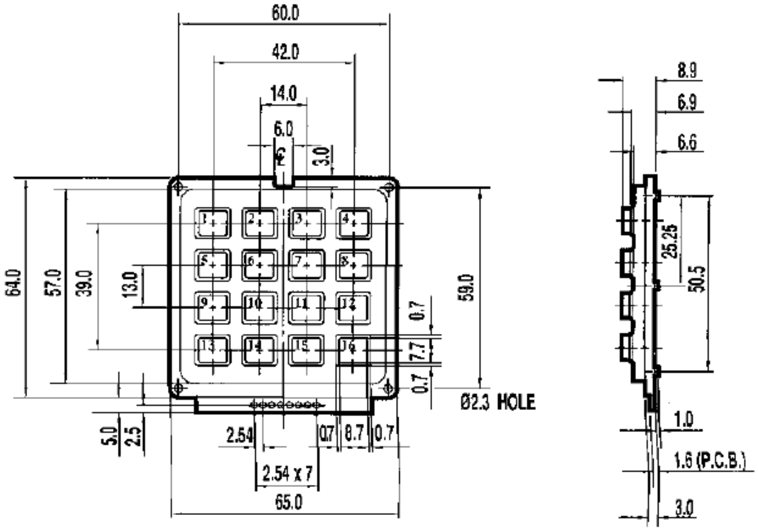
## 4 x 4 Matrix

*Part No. 17KP1604*

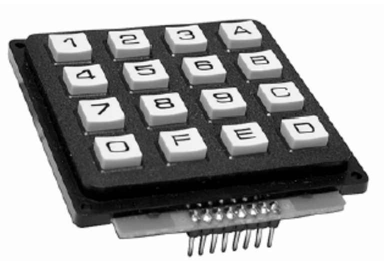
*Back to* Keypad Model KP1604 *Main Page*

**GENERAL SPECIFICATION**

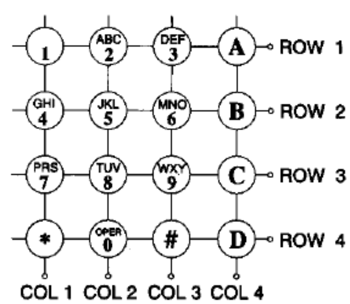
- **Contact Rating:**
  20mA, 24VDC
- **Contact Resistance:**
  200 ohm max
- **Life:**
  1,000,000 cycles per key
- **Operating Temperature:**
  –20°C to +60°C
- **Storage Temperature:**
  –40°C to +65°C




ACTUATING FORCE: 100 ± 30 GRAMS.
STROKE: 1.5 ± 0.5mm.



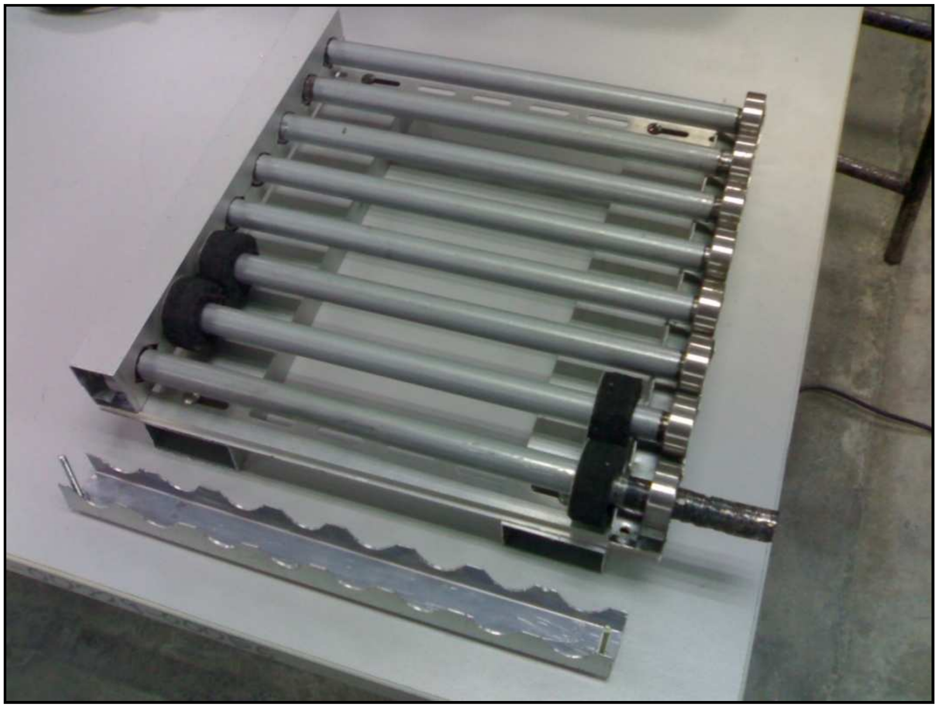
**CIRCUIT DIAGRAM**

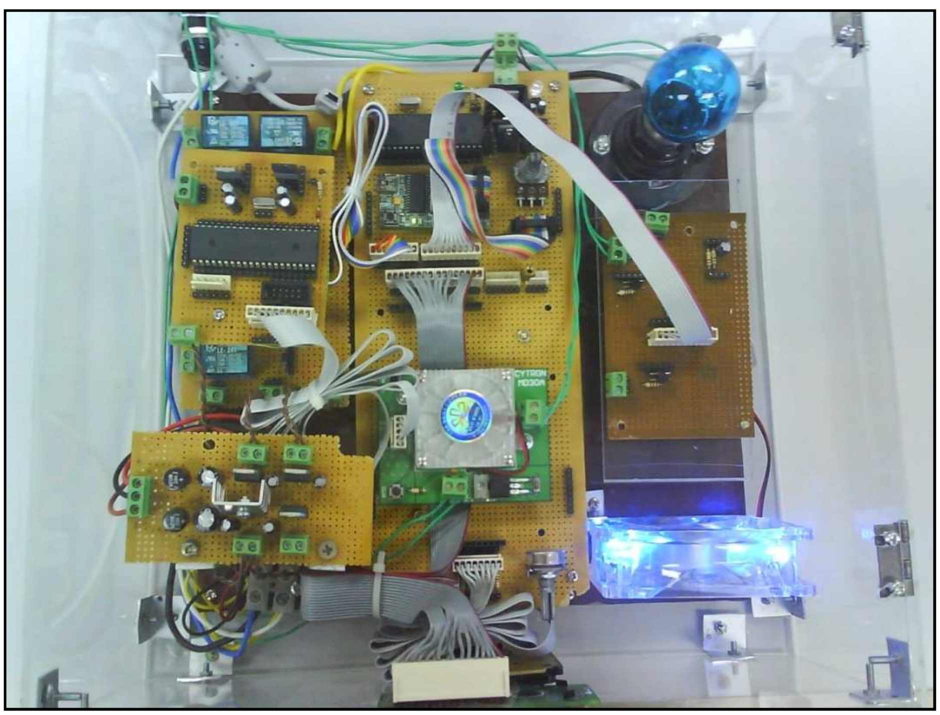


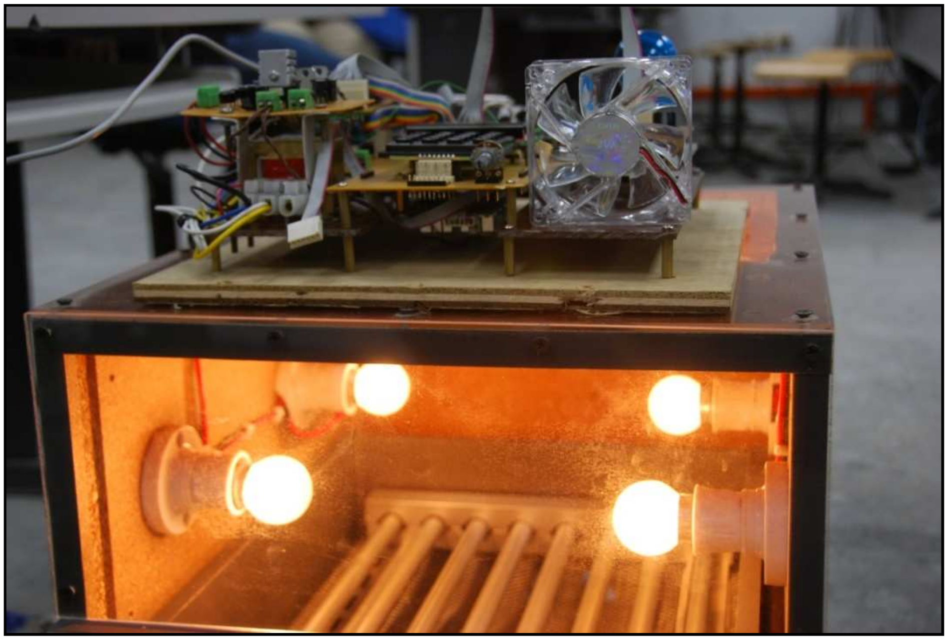

| OUTPUT ARRANGEMENT | |
|---|---|
| **Output Pin No** | **Symbol** |
| 1 | COL 1 |
| 2 | COL 2 |
| 3 | COL 3 |
| 4 | COL 4 |
| 5 | ROW 1 |
| 6 | ROW 2 |
| 7 | ROW 3 |
| 8 | ROW 4 |

## APPENDIX N

## Smart Egg Incubator System for Various Types of Egg








Insert Password
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D