UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **STAIR CLIMBING ROBOT**

SESI PENGAJIAN: 2009/2010

Saya **SYAZWAN BIN KAMARUZZAMAN (861108-38-6369)**
(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.91 TAMAN DESA AMAN
JALAN MAHARAJALELA
36000 TELUK INTAN
PERAK

MOHAMMAD FADHIL ABAS
(Nama Penyelia)

Tarikh: **23 NOVEMBER 2009**

Tarikh: **23 NOVEMBER 2009**

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).

STAIR CLIMBING ROBOT

SYAZWAN BIN KAMARUZZAMAN

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Control and Instrumentation)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2009

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

Signature : ______________________

Name : Mohammad Fadhil Abas

Date : 23 November 2009

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature :

Author : Syazwan Bin Kamaruzzaman

Date : 23 November 2009

Specially dedicated to my beloved parents
Kamaruzzaman Bin Ahmad and Sakinah Bt Yaacob,
brothers, sister and all my fellow friends
who have encouraged, guided and inspired me throughout my journey of education.

ACKNOWLEDGMENT

Alhamdulillah, the highest thanks to God because with His Willingness I try my best to complete the final year project in time.

In the process of carrying out this project, commencing from deciding the title until the completion of project, I was in contact with many people such as academicians, postgraduate researchers and lab assistants. They have rendered their assistant in various aspects and contributed towards my understanding and thoughts.

In particular, I wish to grab this opportunity to express my sincere appreciation to my project supervisor, Mr. Mohammad Fadhil Abas for his advice, guidance and encouragement that have led to the success of this project. I am also indebted to Universiti Malaysia Pahang for providing me the facilities to carry out my project. Librarians at UMP also deserve special thanks for their assistance in the process of my project information obtaining.

My fellow friends should also be recognized for their continual support and encouragement. My sincere appreciation also extends to my course mate 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

This project is to design and develop the prototype of stair climbing robot. The robot body frame will be design based on the shape of the human legs and has the abilities to climbing as the autonomous robot. There were three major part in constructing this robot; mechanical design, electronic circuit design and firmware design. In the mechanical part, robot chassis, servo bracket and servo connector of the robot will be constructed step by step. This robot has three legs which has two joint for every legs. Servo motor is place at every leg joint and used as the machine in robot movement. Microcontroller PIC18F4550 will be the main processor for the robot and PBasic language will be the programming language to communicate between microcontroller and servo motor. The microcontroller will send the instruction directly to the servo motor and the robot will start climbing the stair.

ABSTRAK

Projek ini bertujuan untuk mereka bentuk dan menghasilkan prototaip robot menaiki tangga. Rangka badan robot ini akan direkabentuk berdasarkan kaki manusia dan mempunyai kemampuan menaiki tangga sebagai robot autonomi. Projek ini terbahagi kepada tiga bahagian besar, iaitu rekabentuk mekanikal, litar elektronik dan perisian. Dalam rekabentuk mekanikal, rangka robot, pemegang servo dan penyambung pemegang akan dibina langkah demi langkah. Robot ini mempunyai tiga kaki dan mempunyai dua sendi pada setiap kaki. Servo motor akan diletakkan pada setiap sendi kaki dan berfungsi sebagai mesin dalam pergerakan robot. Micropengawal PIC18F4550 akan menjadi pemproses utama robot dan bahasa PBasic akan menjadi bahasa aturan kod untuk komunikasi antara micropengawal dan servo motor. Micropengawal menghantar isyarat aturan kod secara terus ke servo motor dan robot akan mula menaiki tangga.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	-	Analog-to-Digital Converter
CPU	-	Control Processing Unit
DAC	-	Digital-to-Analog Converter
LED	-	Light Emitting Diode
PIC	-	Peripheral Interface Controller
PVC	-	Polyvinyl Chloride
PWM	-	pulse width modulation
RAM	-	Random Access Memory
ROM	-	Read Only Memory
USB	-	Universal Serial Bus

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Robotic technologies nowadays growth rapidly by human had created the robot to implement the technology with their expert skill. One of the objectives is to create a robot as the human support equipment. The support equipment needed as the assistance in doing the work that usually done by human capability. From this situation the idea comes to design the Stair Climbing Robot.

This project is to develop the stair climbing robot. Climbing robot is the prototype that can climb as the autonomous robot. Stairs are chosen as the obstacle for the robot to climb. It is one of the most common and challenging situations of human environments. Besides, research and development in the area of stair climbing robot has become popular in recent years.

A stair climbing robot consist three elements, which is structure part, moving part and firmware part. Structure part is the main important mechanism to make robot success on climbing the stair. Moving part is the motion method of the robot movement. Firmware part is the brain of the robot where all the instruction of the robot movement is come from this part. Those three elements is the core component because it will cooperatively operate with each other while robot runs.

A stair climbing robot consist three elements, which is structure part, moving part and firmware part. Structure part is the main important mechanism to make robot success on climbing the stair. Moving part is the motion method of the robot movement. Firmware part is the brain of the robot where all the instruction of the robot movement is come from this part. Those three elements is the core component because it will cooperatively operate with each other while robot runs.

1.2 Objective

The objective of this project is to:

a) To developed climbing robot hardware
   - Structure and moving part were involve in developed the robot hardware. Structure robot were consists body part and materials that used, while moving part consist the motor which is as the muscle of the motion.

b) To developed climbing robot firmware
   - Robot will move as the autonomous system where all the movement instruction was constructed in programming language.

1.3 Scope of the Project

The project scope is:

a) Design the climbing robot structure
   - Construct the light weight, low cost and compact robot.

b) Implemented the servo motor in developing the climbing robot
   - Use the characteristic of servo motor as the muscle of the robot motion.

c) Develop the PIC18F4550 and PBasic Language as the robot brain and program language
   - Implement the PIC chip as the engine of the robot and using the PBasic Language to write the instruction program.

CHAPTER 2

LITERATURE REVIEW

2.1 Principles of Robot Locomotion

2.1.1 Reducing the Weight

In build up the robot structure a certain thing must be consider include size, weight and motion method. The weight is one of the important factors affecting the mobility of a robot. For reducing the weight it divides into 3 categories that are battery, drive motor and frame (structure). Battery is the main power supply to run the robot. Every battery has different capacity. Normally low capacity less weight and cheaper than high capacity battery. For reducing the weight, the less weight battery but middle capacity was chosen. If 10% reduces the battery weight, it will reduce 50% on overall robot weight and if 10% robot component increase were effect the overall robot increase to 50%. If the robot needs to use the long use-time battery, we need to make the replaceable battery system where use more than one short use-time battery. When the battery charge down while run the robot, replace the other fully charge battery and the empty battery is charge again. These systems were rotate so that the robot breaks the supply while running. The short use-time battery is too cheap and can probably buy two or three batteries for the price of single long-time battery.

Driver motor is chosen by availability and cost. Normally the robot is designed around the specification of driver motor. Motor were selected first and then built up the robot frame. In chosen drive motor, the several factors must be considered which are the user must avoid the obviously overpowered and grossly oversized that make unnecessary weight and require larger battery to operate.

Third category is regarding to robot frame or robot structure. Wood, metal and plastic is example of frame material. Wood frame is sufficient strength and quality but wood much heavier than others materials. Consider ways to lighten the weight but without sacrificing the strength. The frame must strength because to patch the battery, driver motor with electronic circuit board and component. The solution is select the different material for every part of body frame such as aluminum is use for make the frame crossbar and acrylic plastic is use for make surface to mount the battery and circuit board. Aluminum and acrylic plastic have different weight. Using acrylic plastic as the surface was reducing the weight compare use the aluminum as the surface.

2.1.2 Construct robot with multiple deck

By additional the multiple decks will provide the rigidity but lower possible weight. It also provides the extra space for battery and circuit board. The best less weight material for deck must be selected to avoid effect on robot weight. The column is the pole to hold the upper deck. Circular robot base has 3 columns and square robot base has 4 columns. The stability of upper deck is depending on rigidity of column.

2.1.3 Frame sagging caused by weight

The normally problem on build the frame robot is frame sag at middle because of excessive weight. The problem will cause the 'negative camber' where the robot wheel not in perpendicular to the ground. The wheels become unstable and no longer consistent and veer off to left or right while moving. Three solutions to fix this problem are reducing the weight, strengthen the frame and add crossbar. Strengthen the frame will cause the weight increase, so reduce the weight before strengthen the frame.

(Gordon McComb and Myke Predko, 2006)

2.2 Working with Servo Motor

Servo motor is closed feedback systems. In general, it can be seen in flyable airplane model and car racing model usage. Servo has four type categories and every type has a standard size while different manufacturer. Servo motor operates in constant dc voltage (4.8V & 6.0V).

Inside the servo motor has gear (to reduce motor speed), control board and potentiometer. Motor and potentiometer are connected with circuit board. Potentiometer shaft are indicates the position of output shaft. When potentiometer reaches the desired position, circuit board will shut down the motor. Servo motor been used because to achieve accurate rotational positioning.

Servo motor has limitation rotational. This is the main characteristic of servo compare with dc motor and stepper motor which has continuous rotational. Normally servo rotational limit from 0° to 180° but nowadays the manufacturer has designed 360° rotational servo. In servo has mechanism internal stop that will stop the motor when rotate over natural limit. While using servo keep avoid pushing it beyond their limit because it will cause damage.

Servo shaft positioning by receives the digital input signal. The input signal is Pulse Width Modulation (PWM). The width of the pulse will establish the positioning of shaft. Servo need continuous digital pulse every 20ms to make shaft static at reach position. If supply off, the shaft were turn to neutral position.

Four types of servo are:

1) Quarter scale : Double size from standard size, more powerful, for large model airplanes and perfect power motor.
2) Mini micro : smaller and half size from standard size, design for tight space and not too strong motor.
3) Sail winch : maximum strength servo and design for jib & mainsail sheets
4) Landing-gear retraction : design to retract landing gear of medium and large size model airplanes and exceeding 360° of motion.

(Gordon McComb and Myke Predko, 2006)

2.3 What a Servo: A Quick Tutorial

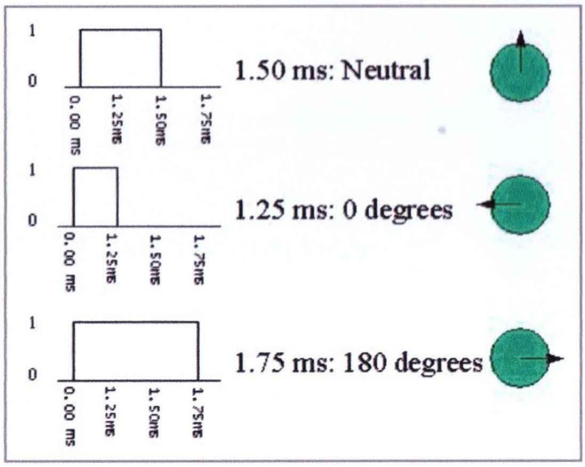


Figure 2.1: Rotational Shaft of Servo Motor

Picture above imagine the rotational of shaft by using pulse width modulation (PWM). Servo need continuous pulse every 20ms to make shaft static at reach position. For example, when supply 1.5ms pulse signal, the shaft were position at neutral position that assume as 90°. So when supply 1.25ms pulse the shaft were turn position at 0° and when supply 1.75ms pulse the shaft were turn position to 180°.The width of the pulse will determine how far the shaft to rotational. If the shaft need to turn large rotational position, the motor run at full speed and if shaft need to rotate small turn the motor run at slower speed. This is called proportional control.

(Seattle Robotics Society, 2007)

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discusses all the hardware and firmware development. For hardware development it divides into two part; mechanical part and electronic part. For firmware development, it will describe about the robot programming.

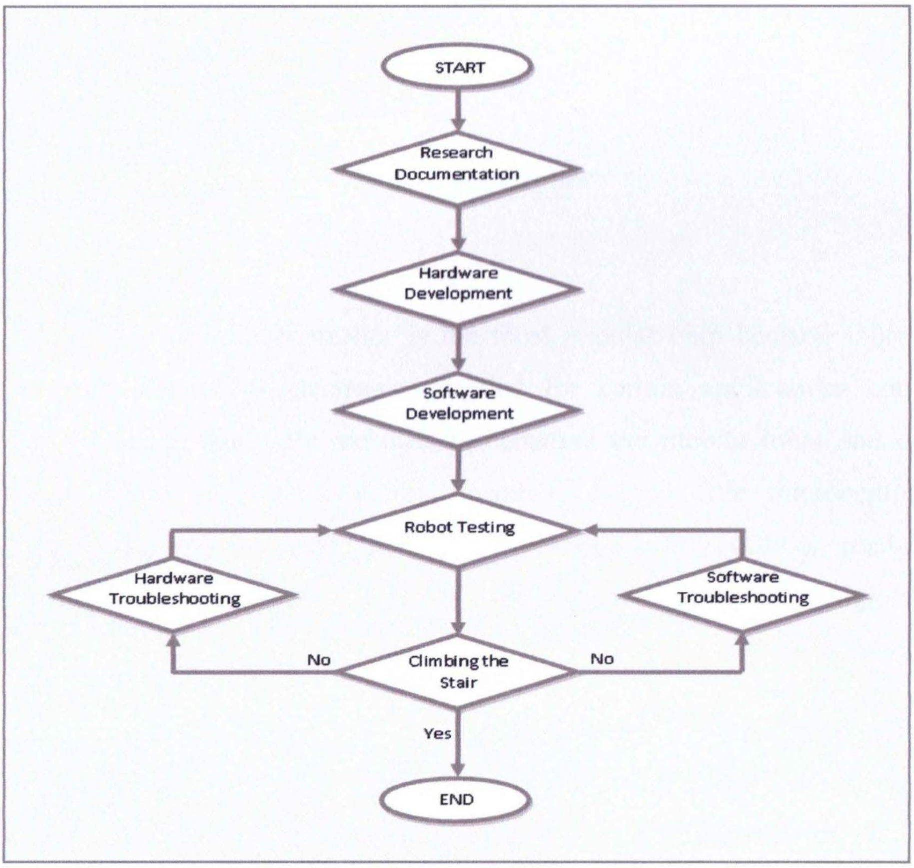


Figure 3.1: Flow Chart of Project Progress

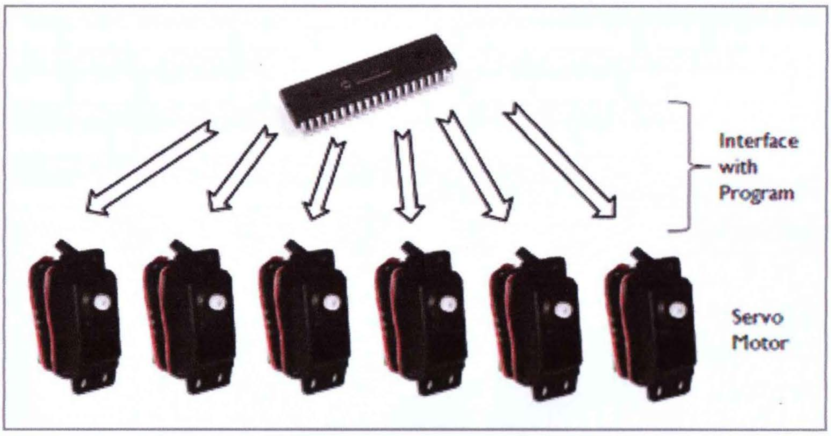


Figure 3.2: Block Diagram of Interface between Input, Processing and Output

3.2 Component Description

3.2.1 Microchip's PIC18F4550 microcontroller

The microcontroller is the most popular chip because there are large amount of electronics needed for certain applications can be eliminated. Normally are uses in construct the mobile robot and other application where computing power is needed. The microcontroller internal structure contain central processing unit (CPU), read-only memory (ROM), random-access memory (RAM),arithmetic logic unit, input and output lines, timers, serial and parallel ports, digital-to-analog converter (DAC) and analog-to-digital converters (ADC).

Peripheral Interface Controller (PIC) chip is the microcontroller family that made by Microchip Technology. PIC chip are popular because of their low cost, big memory size, 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. PIC chip model PIC18F4550 is chosen because it needs simple circuit to configured, built in ADCs, timers, and have five digital input and output.

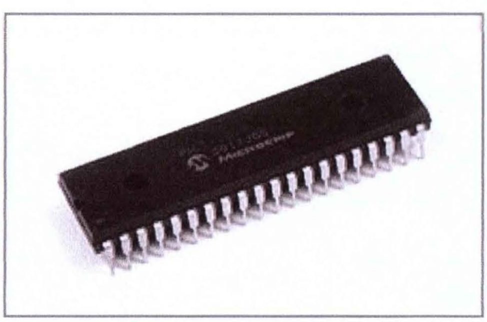


Figure 3.3: PIC18F4550

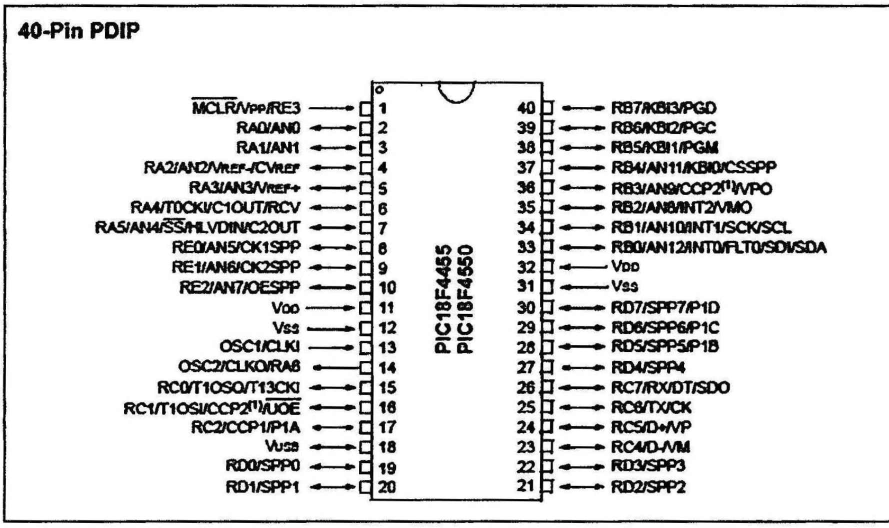


Figure 3.4: Block Diagram of PIC18F4550

3.2.2 Servo Motor

This robot needs the motor that can stop at the certain position with its own capability. Servo motor is the best way solution because of it muscle of motion. The characteristic of servo motor with rotate in angle is needed in robot legs motion. The main characteristic of servo motor is limitation rotational. Servo motor that been used have limit range from 0° to 180°. The torque and speed of it can rotate with weight 7.0 kg/cm and speed 0.16 s/60° respectively. Besides that, it can support the hold robot weight to climb the stair. The driver for servo motor is using the integrated drivers that already build in the servo motor.

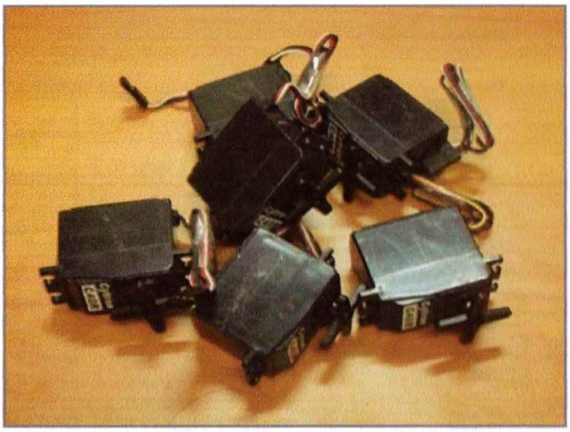


Figure 3.5: Servo Motor

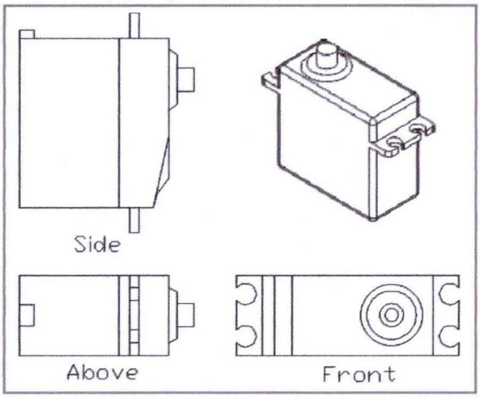


Figure 3.6: Schematic Diagram of Servo Motor

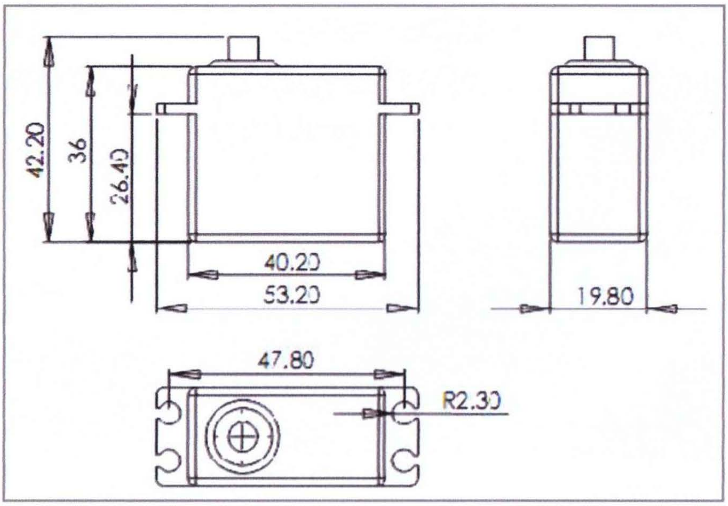


Figure 3.7: Dimension of Servo Motor

Specification		Servo Motor Model
		C40R
4.8V	Speed (s/60°)	0.19
	Torque (Kg.cm)	6.00
6.0V	Speed (s/60°)	0.16
	Torque (Kg.cm)	7.00
7.0V	Speed (s/60°)	-
	Torque (Kg.cm)	-
Signal To Control Angle		TTL PWM
PWM At Min Angle (ms)		0.54
PWM At Max Angle (ms)		2.40
Operating Voltage (VDC)		4.8-6.0
Operating Frequency (Hz)		50.0
Moving Range(degree)		0-180
Wiring (Black/Brown Wire)		Ground
Wiring (Red Wire)		4.8-6.0
Wiring (Orange/Other Wire)		PWM signal
Dimension (mm)		~ 40.2x19.8x36
Weight (g)		38.0
Gear material		Plastic Gear
Servo type		Standard

Figure 3.8: Specification of Servo Motor

3.3 Hardware Development

3.3.1 Mechanical Construction

In mechanical construction this robot was divided into two parts that is chassis and legs. It was constructed like a human body but it only has waist and three legs. Decision to build the robot with three legs is because to make it stable while climbing the stair.

3.3.1.1 Construction of Legs

First step in mechanical construction is creating the robot legs. After done with research, construct the robot with three legs is more advantage than robot with has two legs like human. Robot with Three legs are more stable and steady while climbing the stair and don't need the accelerator sensor as the stabilizer. Servo motor was chosen as the muscle of the motion where it was place at every joint of the robot legs. Every servo motor was mounting at the servo bracket and it becomes the housing of the servo motor. One robot legs have two servo motor, four servo bracket and one foot. Six servo motor was use for this three robot legs. Two types of servo bracket were made by using light weight steel plate. Both it have different shape and measurement. One is a place for mounting the servo motor and the other one is as the connecter between two servo motor. Robot foot is made by straight plate steel that have dimension 3.5 cm x 7.5 cm. It will function as human foot.

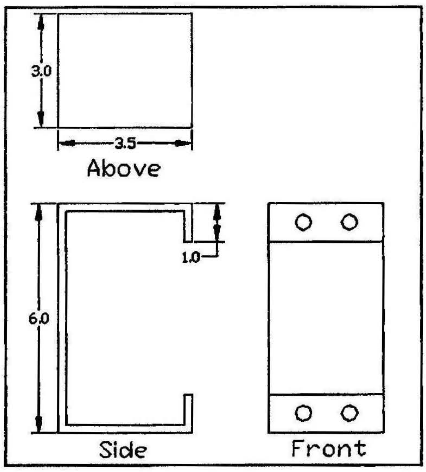


Figure 3.9: Schematic of Servo Bracket

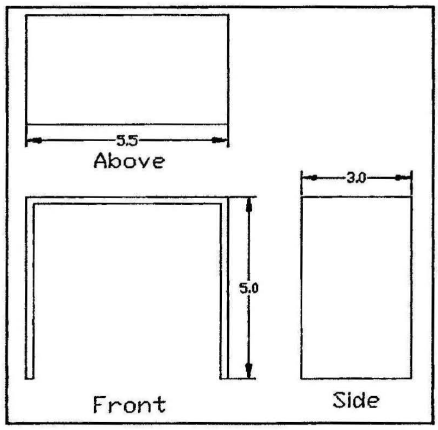


Figure 3.10: Schematic of Connector Bracket

3.3.1.2 Construction of Chassis

Construct the robot chassis is the second step after done with robot legs. Light weight robot is considered to equal with servo motor strength. Simple robot chassis was created with place the straight plate steel as the waist of this robot with dimension 25.5 cm x 3.5 cm which is same material with the robot foot. It will be the place to mounting the robot legs.

Battery was place at the middle of the chassis. The weights of the battery were function as the robot balancer.

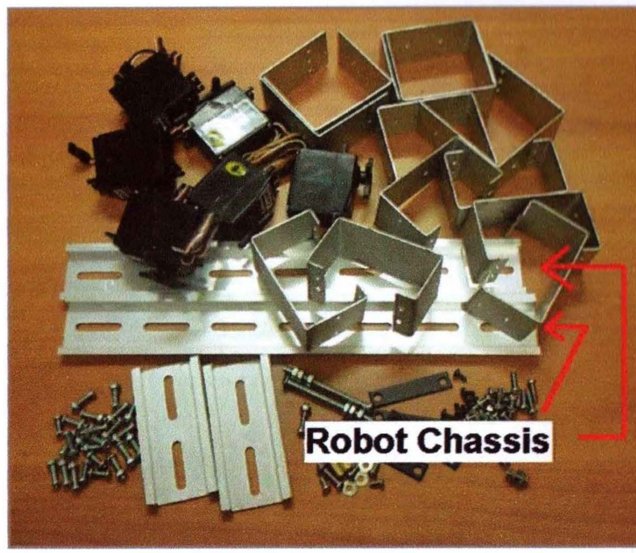


Figure 3.11: Robot Chassis

3.3.1.3 Assembling of Robot

The assembling is begun with mounting the servo motor at servo bracket. Every bracket have same dimension which is created by follow the servo motor dimension. Then screw the connecter bracket at motor gear. This part will be the leg joint. Only one connecter bracket have different dimension while the other five have same dimension. After that servo motor were couples for every leg with mounting the below connector bracket with above servo bracket. Leg foot is mounting at below servo bracket. The both site legs foot position is forward which is like human foot. The foot becomes the robot supporter to avoid the robot falls to front. The middle foot position is horizontal with function as main base to lift the both site leg to next stair. The last process of robot assembling is mounting every leg to robot chassis. Robot chassis were place horizontal to make every leg in straight line position. Second floor of the robot chassis is as the electronic board base. This part is the heart of the robot and it become easy to troubleshoot when place at the above position. The middle between first floor and second floor robot chassis is the place for power source. 12V Lithium polymer battery with dimension 2.2 cm x 3.3 cm x 10.3 cm is chosen as the robot source.

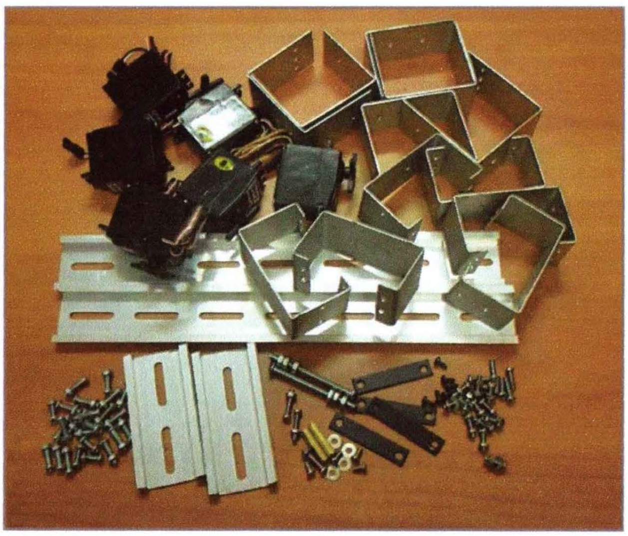


Figure 3.12: Material for Mechanical Construction

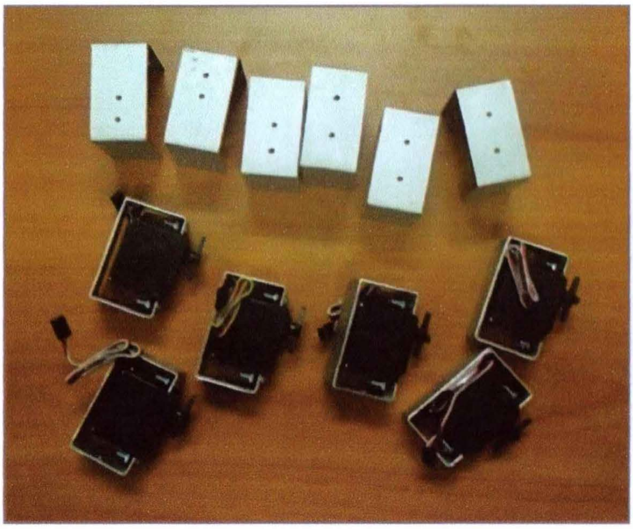


Figure 3.13: Mounting Servo Motor at the Servo Bracket

Figure 3.14: Connect the Connector Bracket with Servo Bracket

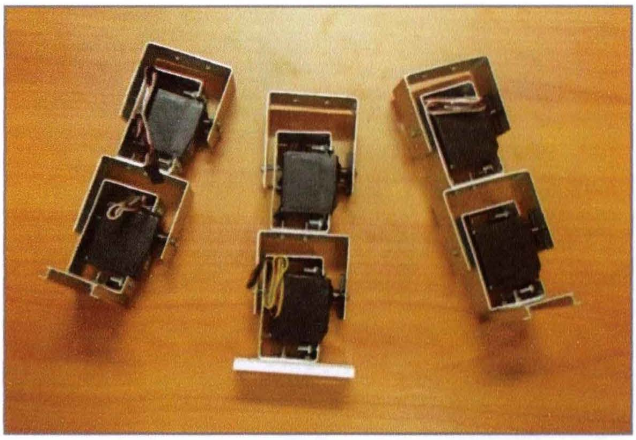


Figure 3.15: Construct the Robot Legs

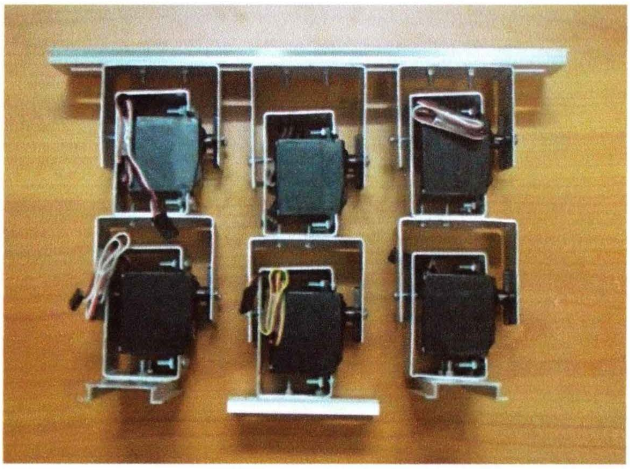


Figure 3.16: Mounting Robot Legs with Chassis

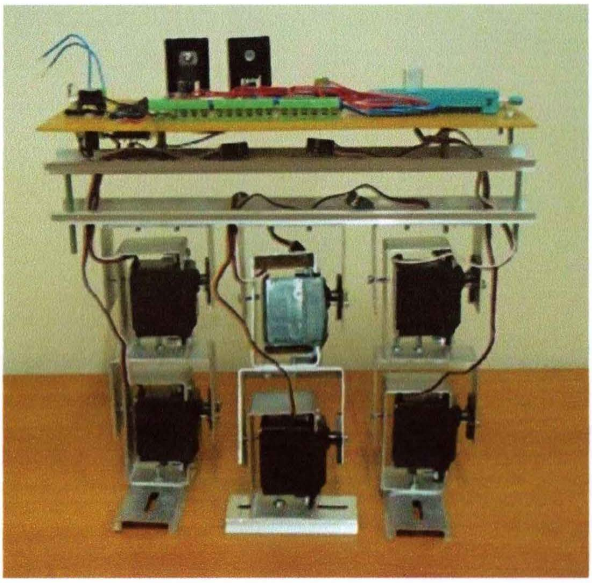


Figure 3.17: Prototype of Stair Climbing Robot

3.3.2 Electronic Development

This part is the heart of the robot. The electronic board will trigger the output signal and send to the servo motor as the output component. Microchip's PIC18F4550 microcontroller is chosen as the heart of this robot. The simple circuit and modern technology is the reason why it been chosen.

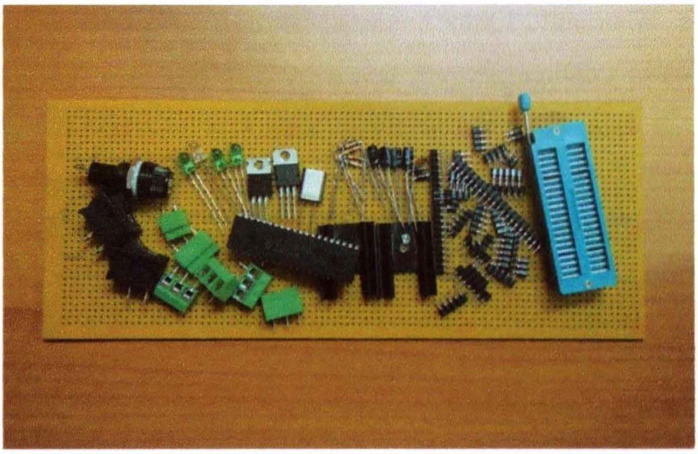


Figure 3.18: Components for Circuit Development

3.3.2.1 Regulator Circuit

Two type of regulator circuit was developed for this robot which is 5 volts output regulator and 6 volts output regulator. 5 volts regulator circuit as the source for PIC microcontroller circuit and 6 volts regulator circuit as the source servo motor.

First circuit part is the protection circuit. The 12 volts source from battery will through the diode 1N4001, 2A fuse and light emitting diode (LED) as the source input indicator light. Both diode and fuse is to avoid the high-current spikes and backward source produce by the battery source.

Second part is regulator switch circuit. The 12 volts source will separate into two paths and has own on-off switch. One path will send source to 5 volts regulator circuit and other one to 6 volts regulator circuit.

In regulator circuits, diode 1N4001 functions as the protection component. 6 volts and 5 volts regulator circuit use LM7806 and LM7805 as a regulator which is produce the 6 volts and 5 volts output source respectively. Two LED has connected to both regulator circuits as the indicator light.

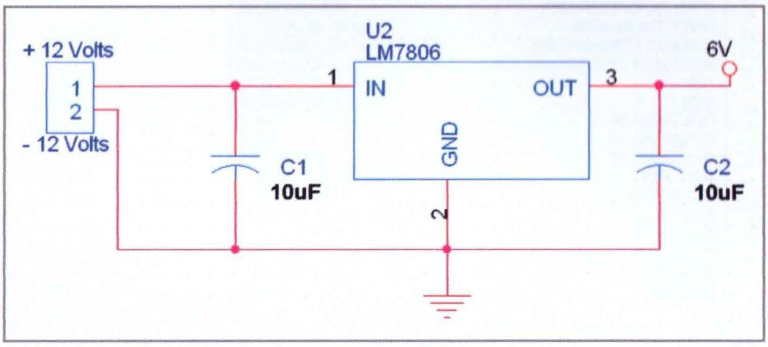


Figure 3.19: 6 Volts Voltage Regulator Circuit

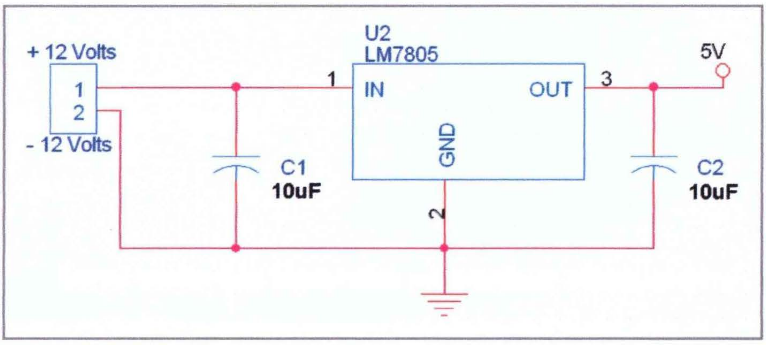


Figure 3.20: 5 Volts Voltage Regulator Circuit

3.3.2.2 PIC Microcontroller Circuit

The circuit is designed around Microchip's PIC18F4550 microcontroller. 5 volts source is needed to run this circuit. Incoming source will through the diode 1N4148 as the voltage guard before enter the PIC chip. The 20.00 MHz crystal oscillator is use to provide a stable clock signal for PIC chip.

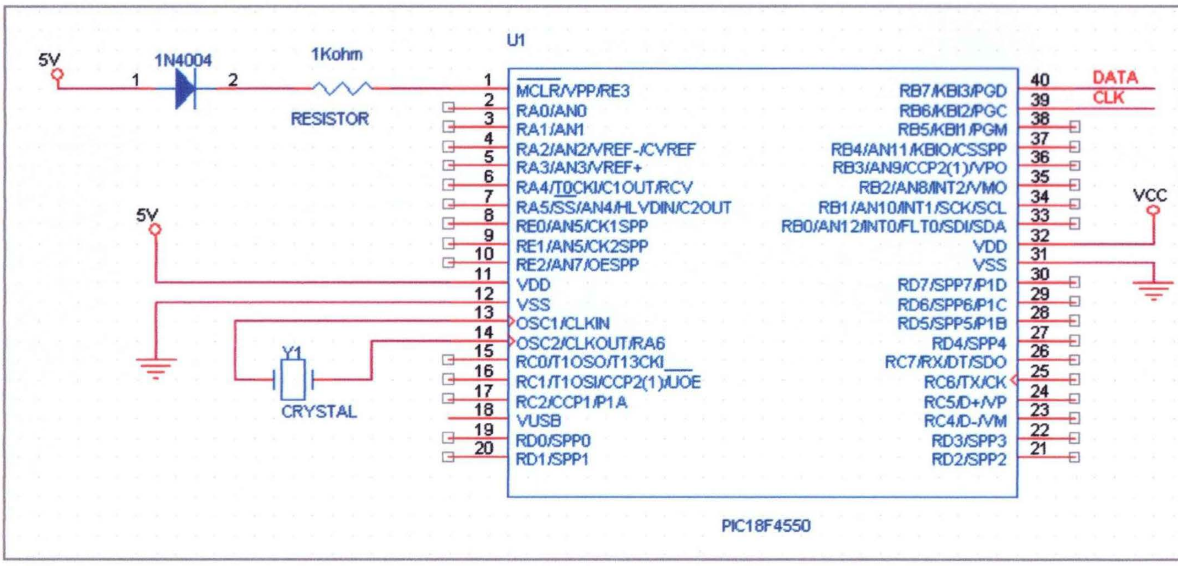


Figure 3.21: Circuit Diagram of PIC18F4550

3.3.2.3 Servo Motor Circuit

Servo motor has three wire connections which is positive source (6 volts), negative source (ground) and input signal came from PIC chip. Six ports at PIC chip is needed to produce the output signal to servo motor. Port D was chosen as the output port source. RD2(pin 21), RD3(pin 22), RD4(pin 27), RD5(pin 28), RD6(pin 29) and RD7(pin 30) was connected to servo motor 1,2,3,4,5 and 6 respectively.

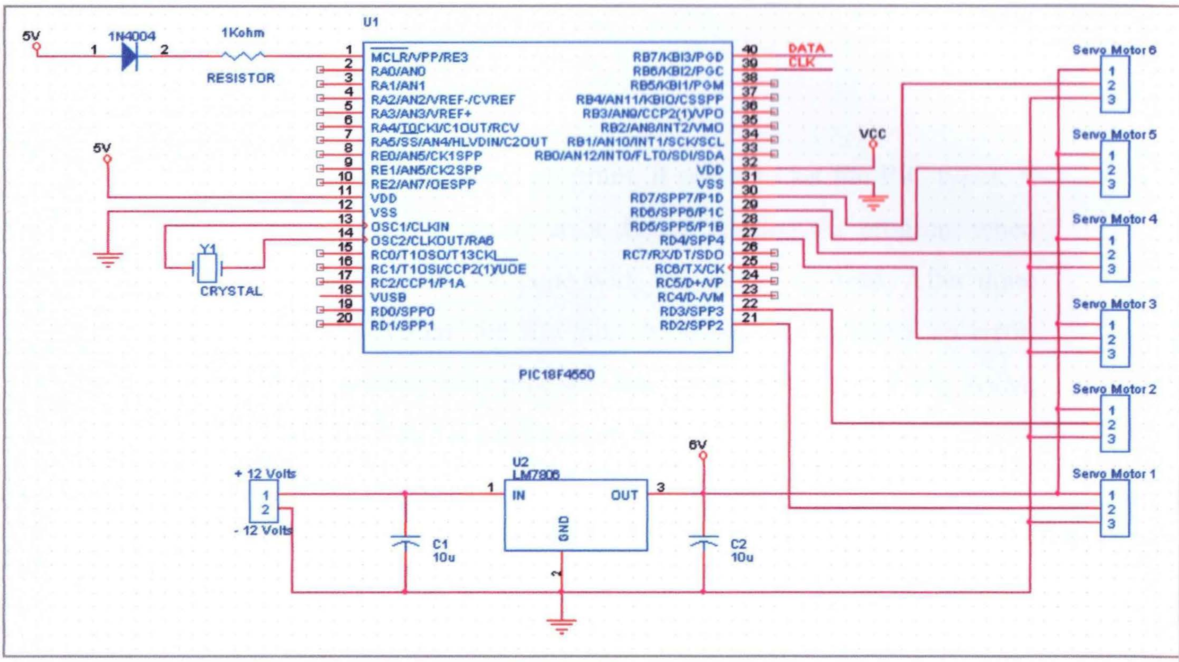


Figure 3.22: Connection between Servo Motor and PIC18F4550 Circuit Diagram

3.4 Firmware Development

Two types software application was used in writing and burning the program to PIC chip. MicroCode Studio is software for writing the programming. PBasic Language is chosen because it easier programming language to write and suitable for beginner. PICkit is used for burning the program into PIC chip.

3.4.1 MicroCode Studio

Before start writing the program, it needs to set the PIC types. It important because there was an error during burning the program when the setting PIC types are not same with PIC chip that used. After done with writing the program, the file must be assembled to check an error. The assembled process will produce four format type file; LST, ASM, HEX and PBP. HEX file is used to burn into PIC chip.

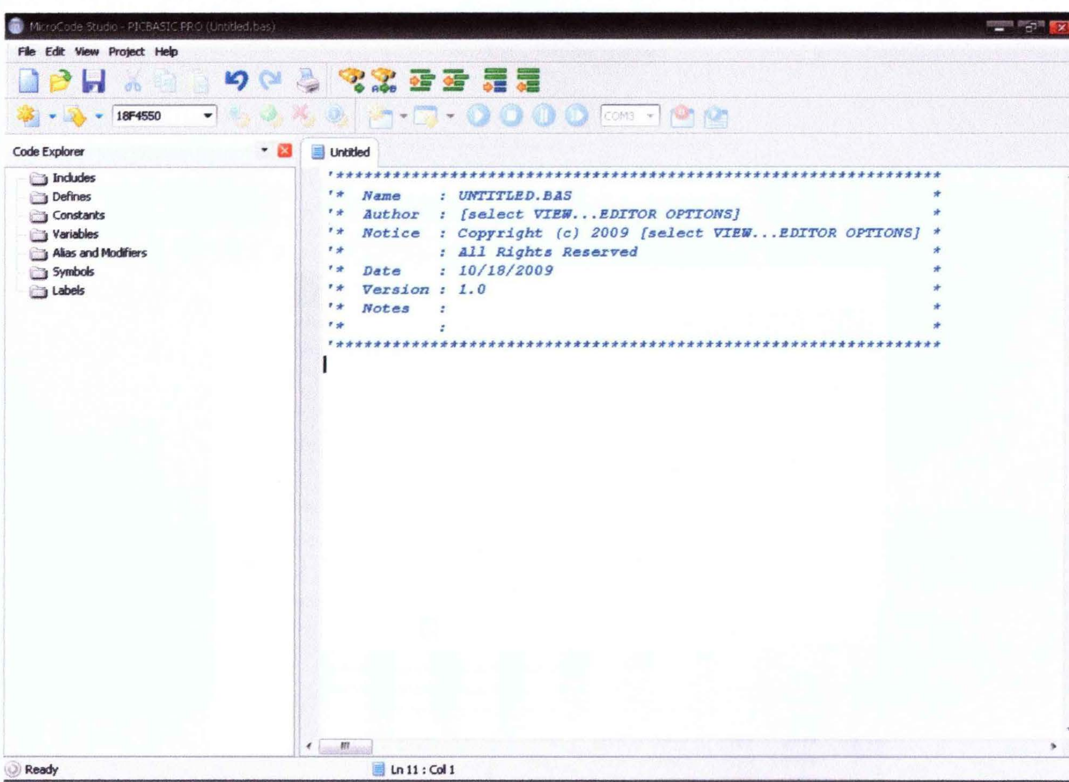


Figure 3.23: MicroCode Studio

3.4.2 PICkit

Cytron USB PIC Programmer is the burner program. Software that used is PICkit. When plug the burner with PIC chip to computer, the PICkit will detect it. Then import the HEX file from MicroCode Studio into PICkit and the PICkit will burn the HEX file into PIC chip.

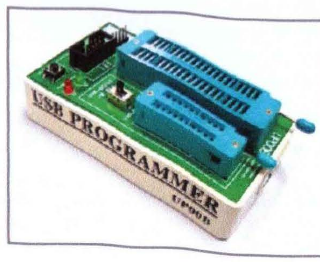


Figure 3.24: USB Programmer

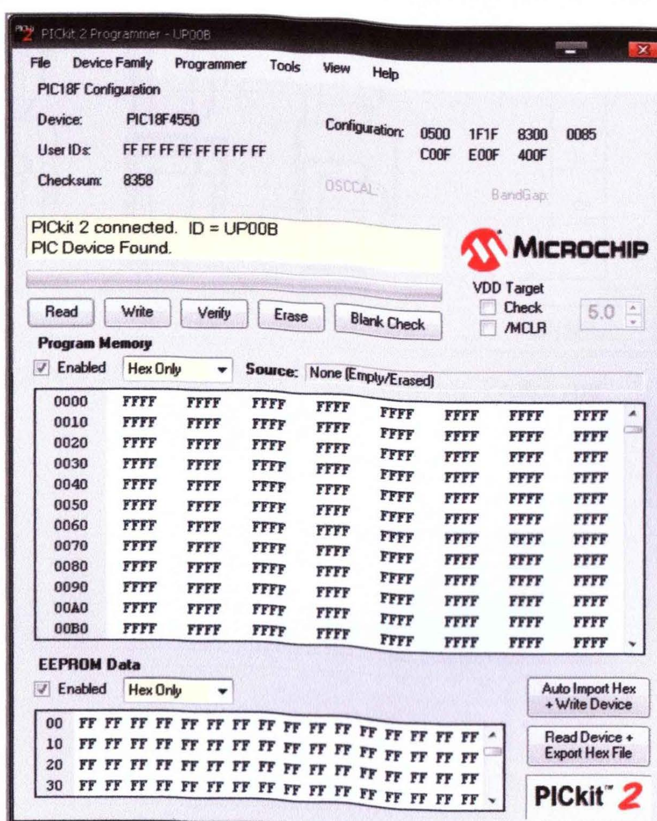


Figure 3.25: PICkit Programmer

3.5 Simulation and Diagram

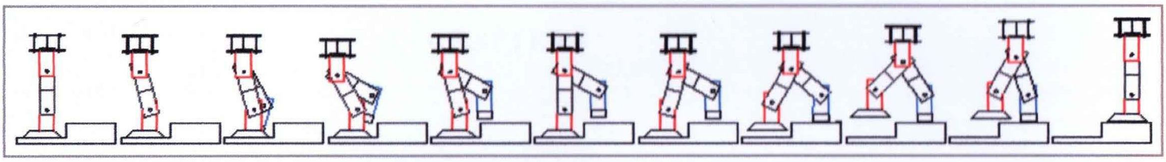


Figure 3.26: Simulation of Robot Movement

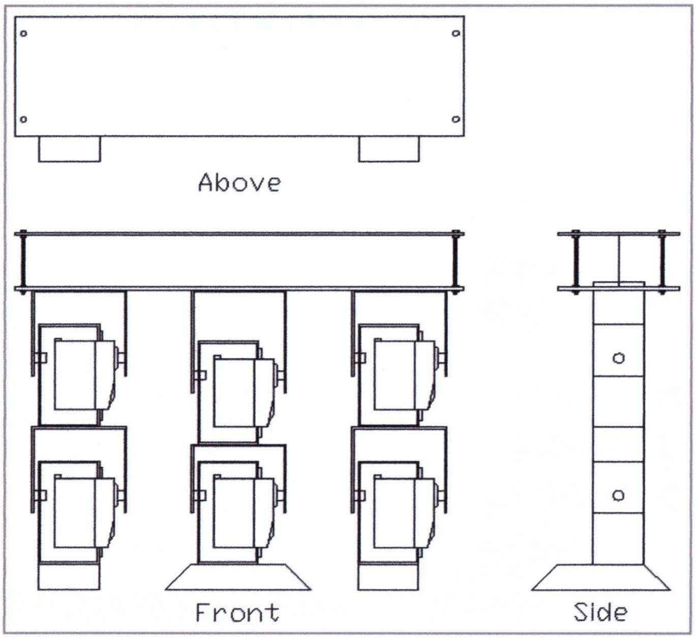


Figure 3.27: Diagram of Stair Climbing Prototype

CHAPTER 4

RESULT & DISCUSSION

4.1 Introduction

This chapter discusses all about the results and discussion obtained from the project. After development the prototype of this project, the strength and weaknesses of the robot was identified. The result will analyzed to ensure the objective achieve the goal.

4.2 Robot Assemble

This is the result for the hardware development. There is the assembled of the robot structure consists robot body (chassis) and robot legs.

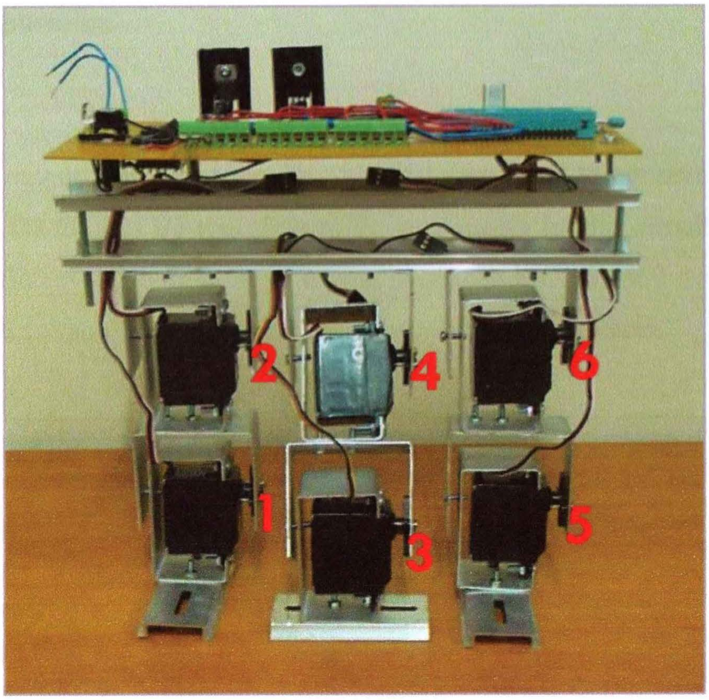


Figure 4.1: A prototype of Stair Climbing Robot

4.3 The Movement of the Robot

The first step in the analysis process is to identify the functionality of the robot. The analysis is conducted based on the objectives of the project which is the ability to climbing the stair. Servo motor was programmed by periodic programming because it needs to move simultaneously and slowly. The speed of the servo motor affect the robot movement and it responsibility to make sure the robot move smoothly.

4.4 Circuit Application

Six output ports from the PIC18F4550 were used to connect with servo motor. Port D2 till port D7 was chosen. Voltage supply for this circuit is separate into two different value of voltage. It is to avoid the PIC18F4550 from received the overload voltage because the servo motor needs 6V to operate.

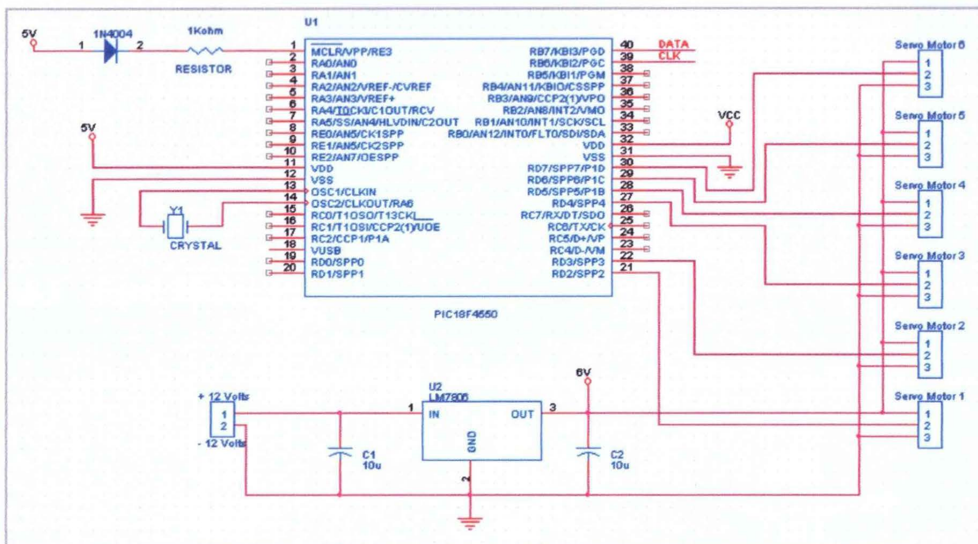


Figure 4.2: Circuit Connection between PIC18F4550 with Servo Motor

4.5 Servo Motor

Servo motor are using Pulse Width Modulation (PWM) signal to move. It will expect to receive signal every 20ms. The length of the pulse will indicate how far the motor turn. The standard length pulse that servo motor can receive is from 0.5ms to 2.5ms which is 0.5ms, 1.5ms and 2.5ms will make the motor turn

0°, 90° and 180° position respectively. However, every servo motor has different range value after doing the calibration.

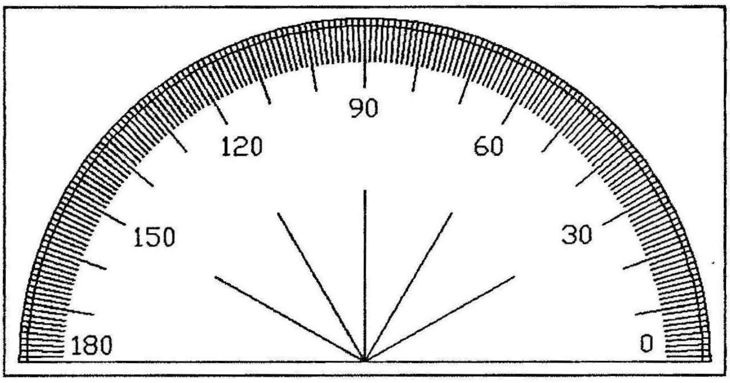


Figure 4.3: Protractor (Angle Indicator)

4.5.1 Servo Motor 1

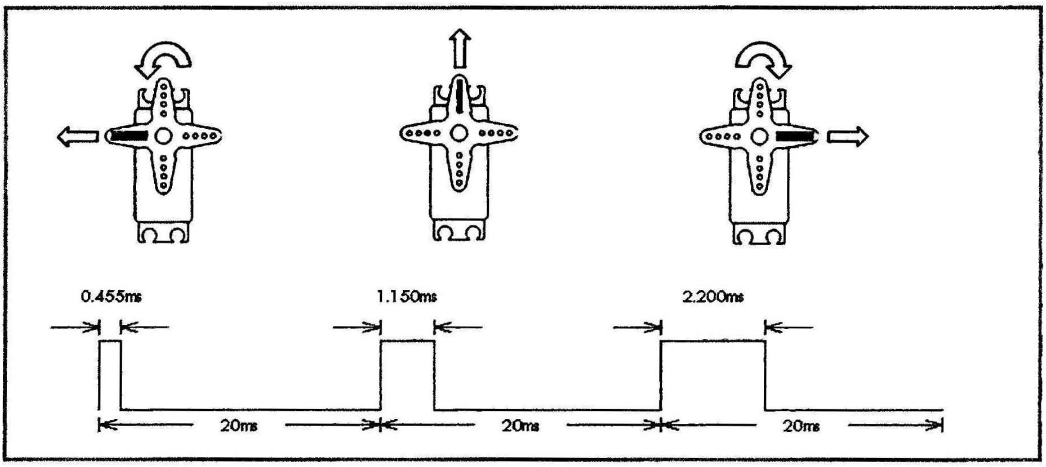


Figure 4.4: Motor Position for Servo Motor 1

Table 4.1: Calibration for Servo Motor 1

Angle	0°	90°	180°
Pulse	0.455ms	1.150ms	2.200ms

4.5.2 Servo Motor 2

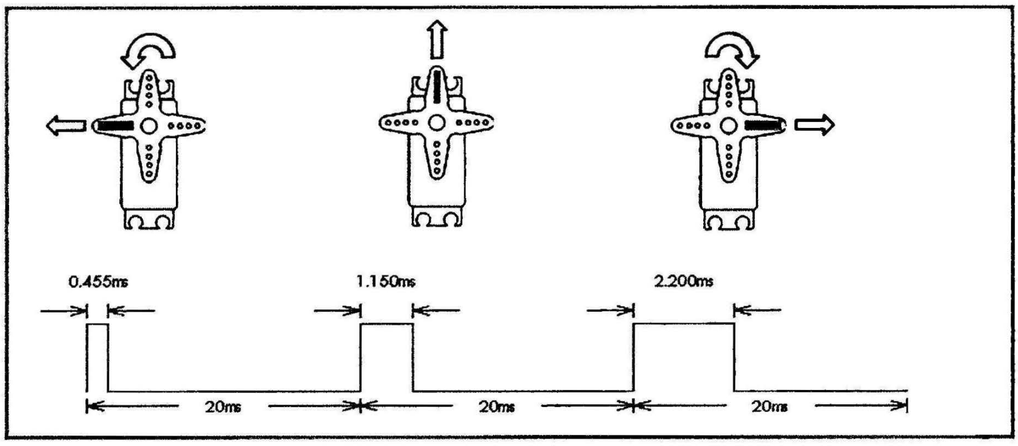


Figure 4.5: Motor Position for Servo Motor 2

Table 4.2: Calibration for Servo Motor 2

Angle	0°	90°	180°
Pulse	0.455ms	1.150ms	2.200ms

4.5.3 Servo Motor 3

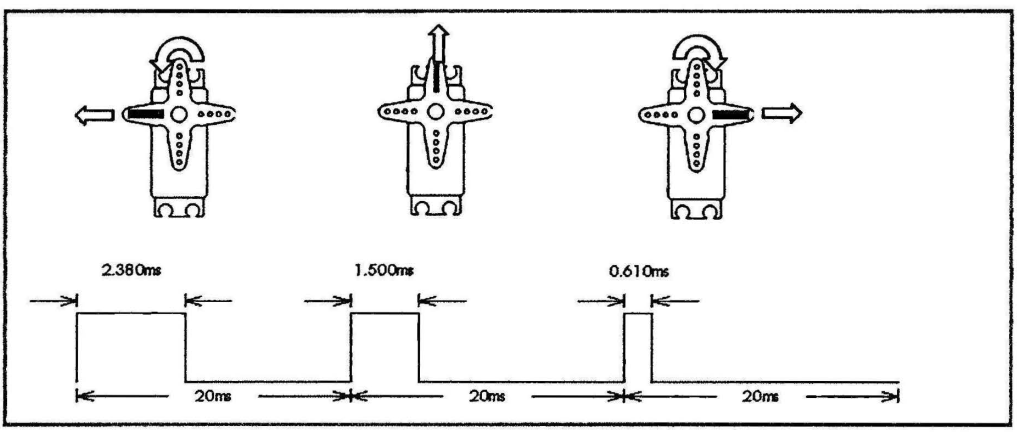


Figure 4.6: Motor Position for Servo Motor 3

Table 4.3: Calibration for Servo Motor 3

Angle	0°	90°	180°
Pulse	2.380ms	1.500ms	0.610ms

4.5.4 Servo Motor 4

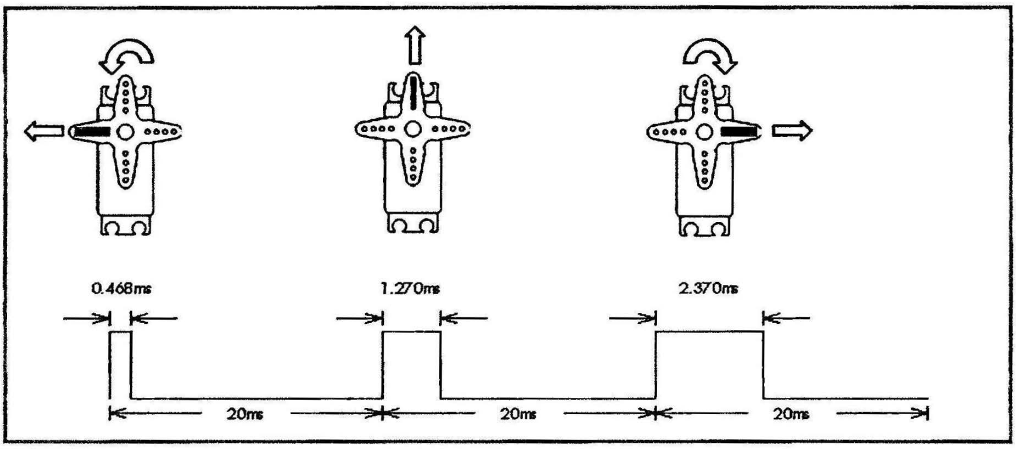


Figure 4.7: Motor Position for Servo Motor 4

Table 4.4: Calibration for Servo Motor 4

Angle	0°	90°	180°
Pulse	0.468ms	1.270ms	2.370ms

4.5.5 Servo Motor 5

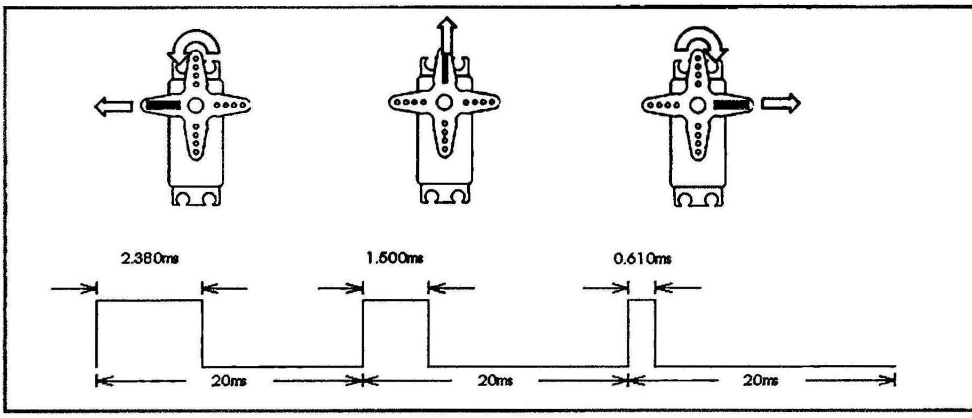


Figure 4.8: Motor Position for Servo Motor 5

Table 4.5: Calibration for Servo Motor 5

Angle	0°	90°	180°
Pulse	2.380ms	1.500ms	0.610ms

4.5.6 Servo Motor 6

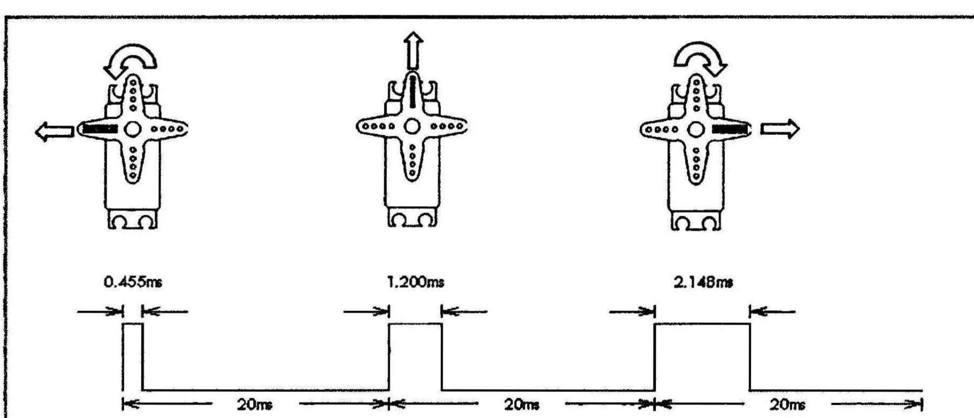


Figure 4.9: Motor Position for Servo Motor 6

Table 4.6: Calibration for Servo Motor 6

Angle	0°	90°	180°
Pulse	0.455ms	1.200ms	2.148ms

4.6 Robot Abilities

Once the robot characteristics have been identified, the next step is to analysis and to identify the strengths and weaknesses of robot. Robot is tested to climb by using wood stair. From the experiments conducted, robot was able to climb the stair. The robot movement must move slowly to avoid robot from fall. However, the problem occurred when the robot climb the robot body have unsteady stand. It was inhibit the robot to climb faster.

4.7 Project Evaluation

Throughout the development of this robot, some problems and obstacles are encountered involving electronic and mechanical parts. The problems are:

4.7.1 Mechanical Structure

a) It took a long time to decide the structure of the robot body. The weight of body robot must be considered to so that the robot body becomes small and light weight.

b) Where to place the servo motor was the second problem. In reduced the material for the robot body, decision was make to design the bracket for servo motor and the bracket become the legs of the robot.

c) The gear of servo motor was broke while doing the calibration because the gear material is plastic. Too easy to broke if the calibration is out of range.

4.7.2 Firmware Development

a) Periodic and repeated method programming was using as the instruction to move the servo motor. Need to adjust the speed of servo motor by programming because servo motor speed is constant.

4.8 Summary

In this chapter, it is obvious that the ability of the robot can be added by doing some modifications. Even the project was completed successfully and the objective achieved, there are several problems, which should be improved in the future. The improvement needed to make sure the weaknesses can be overcome.

CHAPTER 5

CONCLUSION

5.1 Conclusion

This thesis discusses about the development of Stair Climbing Robot actuated by six servo motor. This project is implemented using PIC18F4550 which was programmed using the PBasic Language to control the robot. The robot was successfully built and tested as specified by the objective.

Through the development of the project, many skills have been acquired. The skills are designing a mechanical structure, circuit design and interfacing hardware and firmware in computer.

The ability of Stair Climbing Robot to climb the stair shows the successful of the project. It should be noted that the project is carried out with limited resources and funding. Therefore, this robot is not appropriate to be marketed or commercialized. It is a hope that the next robot development, resources could be added so that the robot's ability can be improved.

As a conclusion, this project has been successfully designed, implemented and tested. It is hoped this robot can be redeveloped and modified to improve the abilities and to provide benefits in future.

5.2 Recommendation

Although Stair Climbing Robot was successfully developed and met the objectives, however it is found that this robot still can be improved by implementing some modifications.

One of it is to use Polyvinyl Chloride (PVC) as the main material to build the body frame. PVC is cheaper and light weight compare with steel.

Another one is use servo motor where the gear made from metal. This will resolve the problem of broken gear. By using the metal gear the torque of the motor will increase.

Lastly is adding the sensor at robot leg to detect the stair. The robot movement will upgrade where it can walk normally as mobile robot and climbing when meet the stair.

5.3 Costing and Commercialization

This project is developed involving the cost about RM493.74. This total cost is including the cost of hardware components such as purchasing the servos and others. Therefore, this project is designed to be very affordable and can function properly.

Furthermore, these projects have the value of commercialization because it using the few interesting elements likes PIC microcontroller and using the servos. Moreover, when this project through a few modifications, this project can extend it capabilities and of course it becomes more valuable on the market.

Table 5.1: Components Price List for Commercialization Purpose

NO	COMPONENTS	PRICE/UNIT (RM)	QUANTITY	PRICE (RM)
PIC Circuit				
1	PIC18F4550	45.00	1	45.00
2	Oscillator Crystal (20Mhz)	1.50	1	1.50
3	Diode (D1N4148)	0.10	1	0.10
4	Resistor (10kΩ)	0.04	1	0.04
5	Zif Socket	20.00	1	20.00
6	Strip Board	3.00	1	3.00
7	Headers	0.50	4	2.00
8	Battery (12V)	90.00	1	90.00
Voltage Regulator Circuit				
9	Voltage Regulator (LM7805)	1.00	1	1.00
10	Voltage Regulator (LM7806)	1.00	1	1.00
11	Capacitor (10uF)	0.10	4	0.10
Servo Motor				
12	Servo Motor	55.00	6	330.00
			TOTAL	**493.74**

REFERENCES

[1] Pushkin Kachroo & Patricia Mellodge (2005).*Mobile Robotic Car Design: RC Servo Motor.*

[2] Gordon McComb and Myke Predko (2006).*Principles of Robot Locomotion.*

[3] Gordon McComb and Myke Predko (2006).*Working with Servo Motor.*

[4] Seattle Robotics Society (2007).*What a Servo?* available at http://www.seattlerobotics.org/.

[5] G.Wendel (2004).*The stairBOT*, available at http://www.stairbot.de/.

[6] Said M. Megahed (2003).*Efficient computation algorithm for dynamic modelling of tree structure robot arms*, available at http://www.sciencedirect.com/

[7] Connectable Software Systems.*USB Project Development Using PIC18F4550*, available at http://www.connectable.org.uk/.

[8] The ROVer Ranch.*Types of Robots*, available at http://prime.jsc.nasa.gov/.

[9] Cytron Technologies. *Servo Motor*, available at http://www.cytron.com.my/.

[10] Microchip Technology Inc.*PIC18F4550*, available at http://www.microchip.com/

[11] MicroEngineering Labs, Inc (2004).*PicBasic Pro Compiler*, available at http://www.melabs.com.

APPENDIX A

SOURCE CODE

'Stair Climbing Robot'

```
'****************************************************************
'* Name    : Stair Climbing Robot                               *
'* Author  : Syazwan Kamaruzzaman                               *
'* Notice  : Copyright (c) 2009 [select VIEW...EDITOR OPTIONS]  *
'*         : All Rights Reserved                                *
'* Date    : 11/15/2009                                         *
'* Version : 1.0                                                *
'* Notes   :                                                    *
'*         :                                                    *
'****************************************************************

define OSC 20          ; oscillator 20mhz

SER1 var PORTD.2       ; port D2 as servo motor 1
SER2 var PORTD.3       ; port D3 as servo motor 2
SER3 var PORTD.4       ; port D4 as servo motor 3
SER4 var PORTD.5       ; port D5 as servo motor 4
SER5 var PORTD.6       ; port D6 as servo motor 5
SER6 var PORTD.7       ; port D7 as servo motor 6

I VAR WORD             ; I as word [0-255]
J var word

TRISD.2=0             ; port D2 as output
TRISD.3=0             ; port D3 as output
TRISD.4=0             ; port D4 as output
TRISD.5=0             ; port D5 as output
TRISD.6=0             ; port D6 as output
TRISD.7=0             ; port D7 as output

MAIN:

  FOR I=0 TO 149       ; initial

    high ser1                ; high servo 1
    pauseus 1150             ; pause 1.15ms
    low ser1                 ; low servo 1
    pauseus 1350             ; pause 1.35ms (1.15ms+1.35ms=2.5ms)

    high ser2                ; high servo 2
    pauseus 1150             ; pause 1.15ms
    low ser2                 ; low servo 2
    pauseus 1350             ; pause 1.35ms (1.15ms+1.35ms=2.5ms)

    high ser3                ; high servo 3
    pauseus 1770             ; pause 1.77ms
    low ser3                 ; low servo 3
    pauseus 730              ; pause 0.73ms (1.77ms+0.73ms=2.5ms)

    high ser4                ; high servo 4
    pauseus 1638             ; pause 1.638ms
    low ser4                 ; low servo 4
    pauseus 862              ; pause 0.862ms (1.638ms+0.862ms=2.5ms)

    high ser5                ; high servo 5
    pauseus 1500             ; pause 1.5ms
    low ser5                 ; low servo 5
    pauseus 1000             ; pause 1.0ms (1.5ms+1.0ms=2.5ms)

    high ser6                ; high servo 6
    pauseus 1200             ; pause 1.2ms
    low ser6                 ; low servo 6
    pauseus 1300             ; pause 1.3ms (1.2ms+1.3ms=2.5ms)

    pauseus 5000             ; pause 5.0ms ([2.5msx6=15ms]+5ms=20ms)

  next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1752
    low ser3
    pauseus 748

    high ser4
    pauseus 1638
    low ser4
    pauseus 862

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1716
    low ser3
    pauseus 784

    high ser4
    pauseus 1638
    low ser4
    pauseus 862

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1615
    low ser4
    pauseus 885

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1644
    low ser3
    pauseus 856

    high ser4
    pauseus 1569
    low ser4
    pauseus 931

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1734
    low ser3
    pauseus 766

    high ser4
    pauseus 1638
    low ser4
    pauseus 862

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1698
    low ser3
    pauseus 802

    high ser4
    pauseus 1638
    low ser4
    pauseus 862

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1662
    low ser3
    pauseus 838

    high ser4
    pauseus 1592
    low ser4
    pauseus 908

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1626
    low ser3
    pauseus 874

    high ser4
    pauseus 1546
    low ser4
    pauseus 954

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1608
    low ser3
    pauseus 892

    high ser4
    pauseus 1523
    low ser4
    pauseus 977

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1477
    low ser4
    pauseus 1023

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1431
    low ser4
    pauseus 1069

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1385
    low ser4
    pauseus 1115

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1590
    low ser3
    pauseus 910

    high ser4
    pauseus 1500
    low ser4
    pauseus 1000

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1454
    low ser4
    pauseus 1046

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1408
    low ser4
    pauseus 1092

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4    ; 1st

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1572
    low ser3
    pauseus 928

    high ser4
    pauseus 1362
    low ser4
    pauseus 1138

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1554
   low ser3
   pauseus 946

   high ser4
   pauseus 1339
   low ser4
   pauseus 1161

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1518
   low ser3
   pauseus 982

   high ser4
   pauseus 1293
   low ser4
   pauseus 1207

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 2nd

   high ser1
   pauseus 1020
   low ser1
   pauseus 1480

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1500
   low ser3
   pauseus 1000

   high ser4
   pauseus 1270
   low ser4
   pauseus 1230

   high ser5
   pauseus 1640
   low ser5
   pauseus 860

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 2nd

   high ser1
   pauseus 760
   low ser1
   pauseus 1740

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1500
   low ser3
   pauseus 1000

   high ser4
   pauseus 1270
   low ser4
   pauseus 1230

   high ser5
   pauseus 1920
   low ser5
   pauseus 580

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 1st

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1536
   low ser3
   pauseus 946

   high ser4
   pauseus 1316
   low ser4
   pauseus 1184

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 1st

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1500
   low ser3
   pauseus 1000

   high ser4
   pauseus 1270
   low ser4
   pauseus 1230

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 2nd

   high ser1
   pauseus 890
   low ser1
   pauseus 1610

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1500
   low ser3
   pauseus 1000

   high ser4
   pauseus 1270
   low ser4
   pauseus 1230

   high ser5
   pauseus 1780
   low ser5
   pauseus 720

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 2nd

   high ser1
   pauseus 630
   low ser1
   pauseus 1870

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1500
   low ser3
   pauseus 1000

   high ser4
   pauseus 1270
   low ser4
   pauseus 1230

   high ser5
   pauseus 2060
   low ser5
   pauseus 440

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 2nd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1500
    low ser3
    pauseus 1000

    high ser4
    pauseus 1270
    low ser4
    pauseus 1230

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1560
    low ser3
    pauseus 850

    high ser4
    pauseus 1402
    low ser4
    pauseus 900

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1620
    low ser3
    pauseus 850

    high ser4
    pauseus 1534
    low ser4
    pauseus 900

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1530
    low ser3
    pauseus 850

    high ser4
    pauseus 1336
    low ser4
    pauseus 900

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1590
    low ser3
    pauseus 850

    high ser4
    pauseus 1468
    low ser4
    pauseus 900

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 3rd

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 1650
    low ser3
    pauseus 850

    high ser4
    pauseus 1600
    low ser4
    pauseus 900

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1225
    low ser2
    pauseus 1275

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1260
    low ser6
    pauseus 1240

    pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1300
    low ser2
    pauseus 1200

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1320
    low ser6
    pauseus 1180

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1375
    low ser2
    pauseus 1125

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1380
    low ser6
    pauseus 1120

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1450
    low ser2
    pauseus 1050

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1440
    low ser6
    pauseus 1060

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1525
    low ser2
    pauseus 975

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1500
    low ser6
    pauseus 1000

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1600
    low ser2
    pauseus 900

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1560
    low ser6
    pauseus 940

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1675
    low ser2
    pauseus 825

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1620
    low ser6
    pauseus 880

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1750
    low ser2
    pauseus 750

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1680
    low ser6
    pauseus 820

    pauseus 5000

next I

FOR I=0 TO 4      ; 4th

    high ser1
    pauseus 500
    low ser1
    pauseus 2000

    high ser2
    pauseus 1825
    low ser2
    pauseus 675

    high ser3
    pauseus 1680
    low ser3
    pauseus 820

    high ser4
    pauseus 1660
    low ser4
    pauseus 840

    high ser5
    pauseus 2200
    low ser5
    pauseus 300

    high ser6
    pauseus 1740
    low ser6
    pauseus 760

    pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 4th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1900
  low ser2
  pauseus 600

  high ser3
  pauseus 1680
  low ser3
  pauseus 820

  high ser4
  pauseus 1660
  low ser4
  pauseus 840

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1800
  low ser6
  pauseus 700

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1820
  low ser2
  pauseus 680

  high ser3
  pauseus 1644
  low ser3
  pauseus 856

  high ser4
  pauseus 1516
  low ser4
  pauseus 984

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1730
  low ser6
  pauseus 770

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1740
  low ser2
  pauseus 760

  high ser3
  pauseus 1608
  low ser3
  pauseus 892

  high ser4
  pauseus 1372
  low ser4
  pauseus 1128

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1660
  low ser6
  pauseus 840

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1660
  low ser2
  pauseus 840

  high ser3
  pauseus 1572
  low ser3
  pauseus 928

  high ser4
  pauseus 1228
  low ser4
  pauseus 1272

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1590
  low ser6
  pauseus 910

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1860
  low ser2
  pauseus 640

  high ser3
  pauseus 1662
  low ser3
  pauseus 838

  high ser4
  pauseus 1588
  low ser4
  pauseus 912

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1765
  low ser6
  pauseus 735

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1780
  low ser2
  pauseus 720

  high ser3
  pauseus 1626
  low ser3
  pauseus 874

  high ser4
  pauseus 1444
  low ser4
  pauseus 1056

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1695
  low ser6
  pauseus 805

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1590
  low ser3
  pauseus 910

  high ser4
  pauseus 1300
  low ser4
  pauseus 1200

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1625
  low ser6
  pauseus 875

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1620
  low ser2
  pauseus 880

  high ser3
  pauseus 1554
  low ser3
  pauseus 946

  high ser4
  pauseus 1156
  low ser4
  pauseus 1344

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1555
  low ser6
  pauseus 945

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1580
  low ser2
  pauseus 920

  high ser3
  pauseus 1536
  low ser3
  pauseus 964

  high ser4
  pauseus 1084
  low ser4
  pauseus 1416

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1520
  low ser6
  pauseus 980

  pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 940
  low ser4
  pauseus 1560

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 560
  low ser1
  pauseus 1940

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1006
  low ser4
  pauseus 1494

  high ser5
  pauseus 2116
  low ser5
  pauseus 384

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 620
  low ser1
  pauseus 1880

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1072
  low ser4
  pauseus 1428

  high ser5
  pauseus 2032
  low ser5
  pauseus 468

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 5th

  high ser1
  pauseus 500
  low ser1
  pauseus 2000

  high ser2
  pauseus 1540
  low ser2
  pauseus 960

  high ser3
  pauseus 1518
  low ser3
  pauseus 982

  high ser4
  pauseus 1012
  low ser4
  pauseus 1488

  high ser5
  pauseus 2200
  low ser5
  pauseus 300

  high ser6
  pauseus 1485
  low ser6
  pauseus 1015

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 530
  low ser1
  pauseus 1970

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 973
  low ser4
  pauseus 1527

  high ser5
  pauseus 2158
  low ser5
  pauseus 342

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 590
  low ser1
  pauseus 1910

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1039
  low ser4
  pauseus 1461

  high ser5
  pauseus 2074
  low ser5
  pauseus 426

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 650
  low ser1
  pauseus 1850

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1105
  low ser4
  pauseus 1395

  high ser5
  pauseus 1990
  low ser5
  pauseus 510

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 680
  low ser1
  pauseus 1820

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1138
  low ser4
  pauseus 1362

  high ser5
  pauseus 1948
  low ser5
  pauseus 552

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I

FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 740
  low ser1
  pauseus 1760

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1204
  low ser4
  pauseus 1296

  high ser5
  pauseus 1864
  low ser5
  pauseus 636

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I

FOR I=0 TO 49     ; 6th

  high ser1
  pauseus 800
  low ser1
  pauseus 1700

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1270
  low ser4
  pauseus 1230

  high ser5
  pauseus 1780
  low ser5
  pauseus 720

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I

FOR I=0 TO 4     ; 7th

  high ser1
  pauseus 800
  low ser1
  pauseus 1700

  high ser2
  pauseus 1660
  low ser2
  pauseus 840

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1430
  low ser4
  pauseus 1070

  high ser5
  pauseus 1780
  low ser5
  pauseus 720

  high ser6
  pauseus 1610
  low ser6
  pauseus 890

  pauseus 5000

next I

FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 710
  low ser1
  pauseus 1790

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1171
  low ser4
  pauseus 1329

  high ser5
  pauseus 1906
  low ser5
  pauseus 594

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I

FOR I=0 TO 4     ; 6th

  high ser1
  pauseus 770
  low ser1
  pauseus 1730

  high ser2
  pauseus 1500
  low ser2
  pauseus 1000

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1237
  low ser4
  pauseus 1263

  high ser5
  pauseus 1822
  low ser5
  pauseus 678

  high ser6
  pauseus 1450
  low ser6
  pauseus 1050

  pauseus 5000

next I

FOR I=0 TO 4     ; 7th

  high ser1
  pauseus 800
  low ser1
  pauseus 1700

  high ser2
  pauseus 1580
  low ser2
  pauseus 920

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1350
  low ser4
  pauseus 1150

  high ser5
  pauseus 1780
  low ser5
  pauseus 720

  high ser6
  pauseus 1530
  low ser6
  pauseus 970

  pauseus 5000

next I

FOR I=0 TO 4     ; 7th

  high ser1
  pauseus 800
  low ser1
  pauseus 1700

  high ser2
  pauseus 1740
  low ser2
  pauseus 760

  high ser3
  pauseus 1500
  low ser3
  pauseus 1000

  high ser4
  pauseus 1510
  low ser4
  pauseus 990

  high ser5
  pauseus 1780
  low ser5
  pauseus 720

  high ser6
  pauseus 1690
  low ser6
  pauseus 810

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 7th

    high ser1
    pauseus 800
    low ser1
    pauseus 1700

    high ser2
    pauseus 1820
    low ser2
    pauseus 680

    high ser3
    pauseus 1500
    low ser3
    pauseus 1000

    high ser4
    pauseus 1590
    low ser4
    pauseus 910

    high ser5
    pauseus 1780
    low ser5
    pauseus 720

    high ser6
    pauseus 1770
    low ser6
    pauseus 730

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 872
    low ser1
    pauseus 1628

    high ser2
    pauseus 1900
    low ser2
    pauseus 600

    high ser3
    pauseus 1588
    low ser3
    pauseus 912

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1724
    low ser5
    pauseus 776

    high ser6
    pauseus 1850
    low ser6
    pauseus 650

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 1014
    low ser1
    pauseus 1486

    high ser2
    pauseus 1820
    low ser2
    pauseus 680

    high ser3
    pauseus 1764
    low ser3
    pauseus 736

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1612
    low ser5
    pauseus 888

    high ser6
    pauseus 1770
    low ser6
    pauseus 730

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1700
    low ser2
    pauseus 800

    high ser3
    pauseus 1940
    low ser3
    pauseus 560

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1650
    low ser6
    pauseus 850

    pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 7th

    high ser1
    pauseus 800
    low ser1
    pauseus 1700

    high ser2
    pauseus 1900
    low ser2
    pauseus 600

    high ser3
    pauseus 1500
    low ser3
    pauseus 1000

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1780
    low ser5
    pauseus 720

    high ser6
    pauseus 1850
    low ser6
    pauseus 650

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 944
    low ser1
    pauseus 1556

    high ser2
    pauseus 1860
    low ser2
    pauseus 640

    high ser3
    pauseus 1676
    low ser3
    pauseus 824

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1668
    low ser5
    pauseus 832

    high ser6
    pauseus 1810
    low ser6
    pauseus 690

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 1082
    low ser1
    pauseus 1418

    high ser2
    pauseus 1740
    low ser2
    pauseus 760

    high ser3
    pauseus 1852
    low ser3
    pauseus 648

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1556
    low ser5
    pauseus 944

    high ser6
    pauseus 1690
    low ser6
    pauseus 810

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

    high ser1
    pauseus 1014
    low ser1
    pauseus 1486

    high ser2
    pauseus 1700
    low ser2
    pauseus 800

    high ser3
    pauseus 1940
    low ser3
    pauseus 560

    high ser4
    pauseus 1670
    low ser4
    pauseus 830

    high ser5
    pauseus 1612
    low ser5
    pauseus 888

    high ser6
    pauseus 1650
    low ser6
    pauseus 850

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

  high ser1
  pauseus 1048
  low ser1
  pauseus 1452

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1940
  low ser3
  pauseus 560

  high ser4
  pauseus 1670
  low ser4
  pauseus 830

  high ser5
  pauseus 1584
  low ser5
  pauseus 916

  high ser6
  pauseus 1650
  low ser6
  pauseus 850

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

  high ser1
  pauseus 1116
  low ser1
  pauseus 1384

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1940
  low ser3
  pauseus 560

  high ser4
  pauseus 1670
  low ser4
  pauseus 830

  high ser5
  pauseus 1528
  low ser5
  pauseus 972

  high ser6
  pauseus 1650
  low ser6
  pauseus 850

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1940
  low ser3
  pauseus 560

  high ser4
  pauseus 1670
  low ser4
  pauseus 830

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1650
  low ser6
  pauseus 850

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1535
  low ser2
  pauseus 965

  high ser3
  pauseus 2072
  low ser3
  pauseus 428

  high ser4
  pauseus 1619
  low ser4
  pauseus 881

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1499
  low ser6
  pauseus 1001

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 8th

  high ser1
  pauseus 1082
  low ser1
  pauseus 1418

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1940
  low ser3
  pauseus 560

  high ser4
  pauseus 1670
  low ser4
  pauseus 830

  high ser5
  pauseus 1556
  low ser5
  pauseus 944

  high ser6
  pauseus 1650
  low ser6
  pauseus 850

  pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 8th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1700
  low ser2
  pauseus 800

  high ser3
  pauseus 1940
  low ser3
  pauseus 560

  high ser4
  pauseus 1670
  low ser4
  pauseus 830

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1650
  low ser6
  pauseus 850

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1590
  low ser2
  pauseus 910

  high ser3
  pauseus 2028
  low ser3
  pauseus 384

  high ser4
  pauseus 1636
  low ser4
  pauseus 864

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1541
  low ser6
  pauseus 959

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1480
  low ser2
  pauseus 1020

  high ser3
  pauseus 2116
  low ser3
  pauseus 384

  high ser4
  pauseus 1602
  low ser4
  pauseus 898

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1457
  low ser6
  pauseus 1043

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1425
  low ser2
  pauseus 1075

  high ser3
  pauseus 2160
  low ser3
  pauseus 340

  high ser4
  pauseus 1585
  low ser4
  pauseus 915

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1413
  low ser6
  pauseus 1087

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1315
  low ser2
  pauseus 1185

  high ser3
  pauseus 2248
  low ser3
  pauseus 252

  high ser4
  pauseus 1551
  low ser4
  pauseus 949

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1329
  low ser6
  pauseus 1171

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1205
  low ser2
  pauseus 1295

  high ser3
  pauseus 2338
  low ser3
  pauseus 162

  high ser4
  pauseus 1517
  low ser4
  pauseus 983

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1245
  low ser6
  pauseus 1255

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 10th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 2304
  low ser3
  pauseus 196

  high ser4
  pauseus 1564
  low ser4
  pauseus 936

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1370
  low ser2
  pauseus 1130

  high ser3
  pauseus 2204
  low ser3
  pauseus 296

  high ser4
  pauseus 1568
  low ser4
  pauseus 932

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1371
  low ser6
  pauseus 1129

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1260
  low ser2
  pauseus 1240

  high ser3
  pauseus 2292
  low ser3
  pauseus 208

  high ser4
  pauseus 1534
  low ser4
  pauseus 966

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1287
  low ser6
  pauseus 1213

  pauseus 5000

next I
```

```
FOR I=0 TO 49      ; 9th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 2380
  low ser3
  pauseus 120

  high ser4
  pauseus 1500
  low ser4
  pauseus 1000

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 10th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 2228
  low ser3
  pauseus 272

  high ser4
  pauseus 1628
  low ser4
  pauseus 872

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2152
    low ser3
    pauseus 348

    high ser4
    pauseus 1692
    low ser4
    pauseus 808

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 1820
    low ser4
    pauseus 680

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 1948
    low ser4
    pauseus 552

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 2076
    low ser4
    pauseus 424

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2076
    low ser3
    pauseus 424

    high ser4
    pauseus 1756
    low ser4
    pauseus 744

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 1884
    low ser4
    pauseus 616

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 2012
    low ser4
    pauseus 488

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 49     ; 10th

    high ser1
    pauseus 1150
    low ser1
    pauseus 1350

    high ser2
    pauseus 1150
    low ser2
    pauseus 1350

    high ser3
    pauseus 2000
    low ser3
    pauseus 500

    high ser4
    pauseus 2140
    low ser4
    pauseus 360

    high ser5
    pauseus 1500
    low ser5
    pauseus 1000

    high ser6
    pauseus 1200
    low ser6
    pauseus 1300

    pauseus 5000

next I
```

```
FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1977
  low ser3
  pauseus 523

  high ser4
  pauseus 2090
  low ser4
  pauseus 698

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1931
  low ser3
  pauseus 569

  high ser4
  pauseus 1990
  low ser4
  pauseus 734

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1885
  low ser3
  pauseus 615

  high ser4
  pauseus 1890
  low ser4
  pauseus 770

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1839
  low ser3
  pauseus 661

  high ser4
  pauseus 1790
  low ser4
  pauseus 806

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1954
  low ser3
  pauseus 546

  high ser4
  pauseus 2040
  low ser4
  pauseus 716

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1908
  low ser3
  pauseus 592

  high ser4
  pauseus 1940
  low ser4
  pauseus 752

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1862
  low ser3
  pauseus 638

  high ser4
  pauseus 1840
  low ser4
  pauseus 788

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I

FOR I=0 TO 4     ; 11th

  high ser1
  pauseus 1150
  low ser1
  pauseus 1350

  high ser2
  pauseus 1150
  low ser2
  pauseus 1350

  high ser3
  pauseus 1816
  low ser3
  pauseus 684

  high ser4
  pauseus 1740
  low ser4
  pauseus 824

  high ser5
  pauseus 1500
  low ser5
  pauseus 1000

  high ser6
  pauseus 1200
  low ser6
  pauseus 1300

  pauseus 5000

next I
```

```
FOR I=0 TO 4      ; 11th

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1793
   low ser3
   pauseus 707

   high ser4
   pauseus 1690
   low ser4
   pauseus 849

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I

FOR I=0 TO 49      ; 11th

   high ser1
   pauseus 1150
   low ser1
   pauseus 1350

   high ser2
   pauseus 1150
   low ser2
   pauseus 1350

   high ser3
   pauseus 1770
   low ser3
   pauseus 730

   high ser4
   pauseus 1638
   low ser4
   pauseus 862

   high ser5
   pauseus 1500
   low ser5
   pauseus 1000

   high ser6
   pauseus 1200
   low ser6
   pauseus 1300

   pauseus 5000

next I

Goto Main
```

APPENDIX B

PIC18F4550 Datasheet

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 (TCY/16)
  - Compare is 16-bit, max. resolution 100 ns (TCY)
  - 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 I²C™ 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

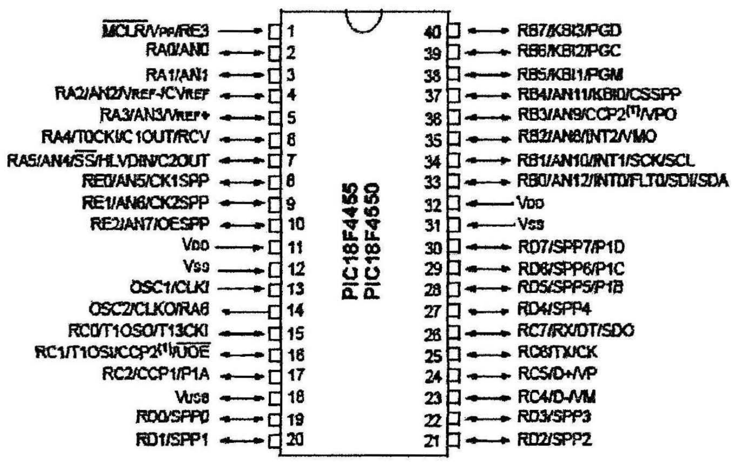
Special Microcontroller Features:

- 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)

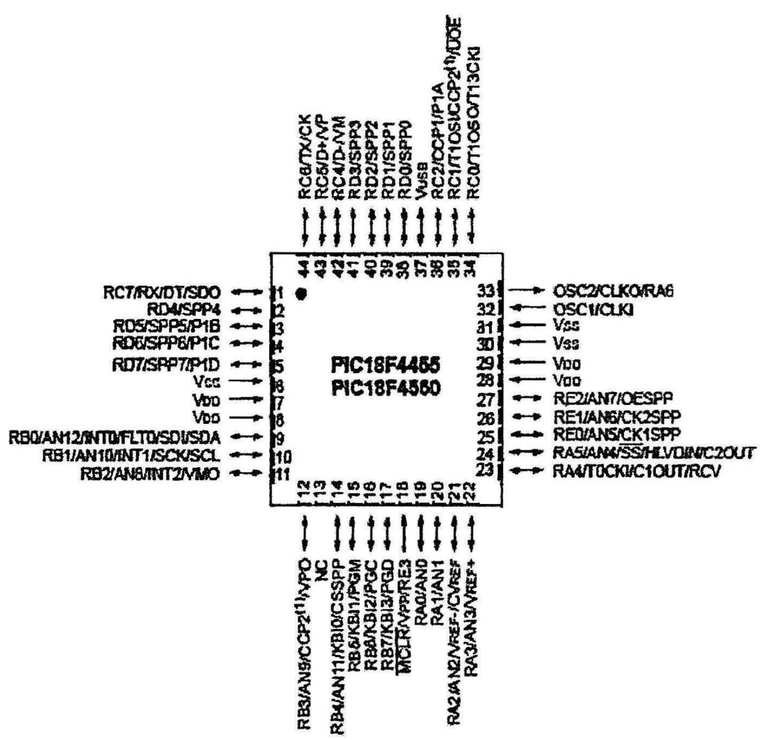
Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI™	Master I²C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	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

Pin Diagrams

40-Pin PDIP



44-Pin QFN



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

2: Special ICPORTS features available in select circumstances. See Section 25.9 "Special ICPORT Features (Designated Packages Only)" for more information.

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F2455
- PIC18F2550
- PIC18F4455
- PIC18F4550
- PIC18LF2455
- PIC18LF2550
- PIC18LF4455
- PIC18LF4550

This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high endurance, Enhanced Flash program memory. In addition to these features, the PIC18F2455/2550/4455/4550 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F2455/2550/4455/4550 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized. See **Section 28.0 "Electrical Characteristics"** for values.

1.1.2 UNIVERSAL SERIAL BUS (USB)

Devices in the PIC18F2455/2550/4455/4550 family incorporate a fully featured Universal Serial Bus communications module that is compliant with the USB Specification Revision 2.0. The module supports both low-speed and full speed communication for all supported data transfer types. It also incorporates its own on-chip transceiver and 3.3V regulator and supports the use of external transceivers and voltage regulators.

1.1.3 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F2455/2550/4455/4550 family offer twelve different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes using crystals or ceramic resonators.
- Four External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O).
- An internal oscillator block which provides an 8 MHz clock (±2% accuracy) and an INTRC source (approximately 31 kHz, stable over temperature and VDD), as well as a range of 6 user selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the high-speed crystal and external oscillator modes, which allows a wide range of clock speeds from 4 MHz to 48 MHz.
- Asynchronous dual clock operation, allowing the USB module to run from a high-frequency oscillator while the rest of the microcontroller is clocked from an internal low-power oscillator.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

1.2 Other Special Features

- **Memory Endurance:** The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-Programmability:** These devices can write to their own program memory spaces under internal software control. By using a bootloader routine, located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.
- **Extended Instruction Set:** The PIC18F2455/2550/4455/4550 family introduces an optional extension to the PIC18 instruction set, which adds 8 new instructions and an Indexed Literal Offset Addressing mode. This extension, enabled as a device configuration option, has been specifically designed to optimize re-entrant application code originally developed in high-level languages such as C.
- **Enhanced CCP Module:** In PWM mode, this module provides 1, 2 or 4 modulated outputs for controlling half-bridge and full-bridge drivers. Other features include auto-shutdown for disabling PWM outputs on interrupt or other select conditions and auto-restart to reactivate outputs once the condition has cleared.
- **Enhanced Addressable USART:** This serial communication module is capable of standard RS-232 operation and provides support for the LIN bus protocol. Other enhancements include Automatic Baud Rate Detection and a 16-bit Baud Rate Generator for improved resolution. When the microcontroller is using the internal oscillator block, the EUSART provides stable operation for applications that talk to the outside world without using an external crystal (or its accompanying power requirement).
- **10-bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated, without waiting for a sampling period and thus, reducing code overhead.
- **Dedicated ICD/ICSP Port:** These devices introduce the use of debugger and programming pins that are not multiplexed with other microcontroller features. Offered as an option in select packages, this feature allows users to develop I/O intensive applications while retaining the ability to program and debug in the circuit.

1.3 Details on Individual Family Members

Devices in the PIC18F2455/2550/4455/4550 family are available in 28-pin and 40/44-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2.

The devices are differentiated from each other in six ways:

1. Flash program memory (24 Kbytes for PIC18FX455 devices, 32 Kbytes for PIC18FX550).
2. A/D channels (10 for 28-pin devices, 13 for 40/44-pin devices).
3. I/O ports (3 bidirectional ports and 1 input only port on 28-pin devices, 5 bidirectional ports on 40/44-pin devices).
4. CCP and Enhanced CCP implementation (28-pin devices have 2 standard CCP modules, 40/44-pin devices have one standard CCP module and one ECCP module).
5. Streaming Parallel Port (present only on 40/44-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1.

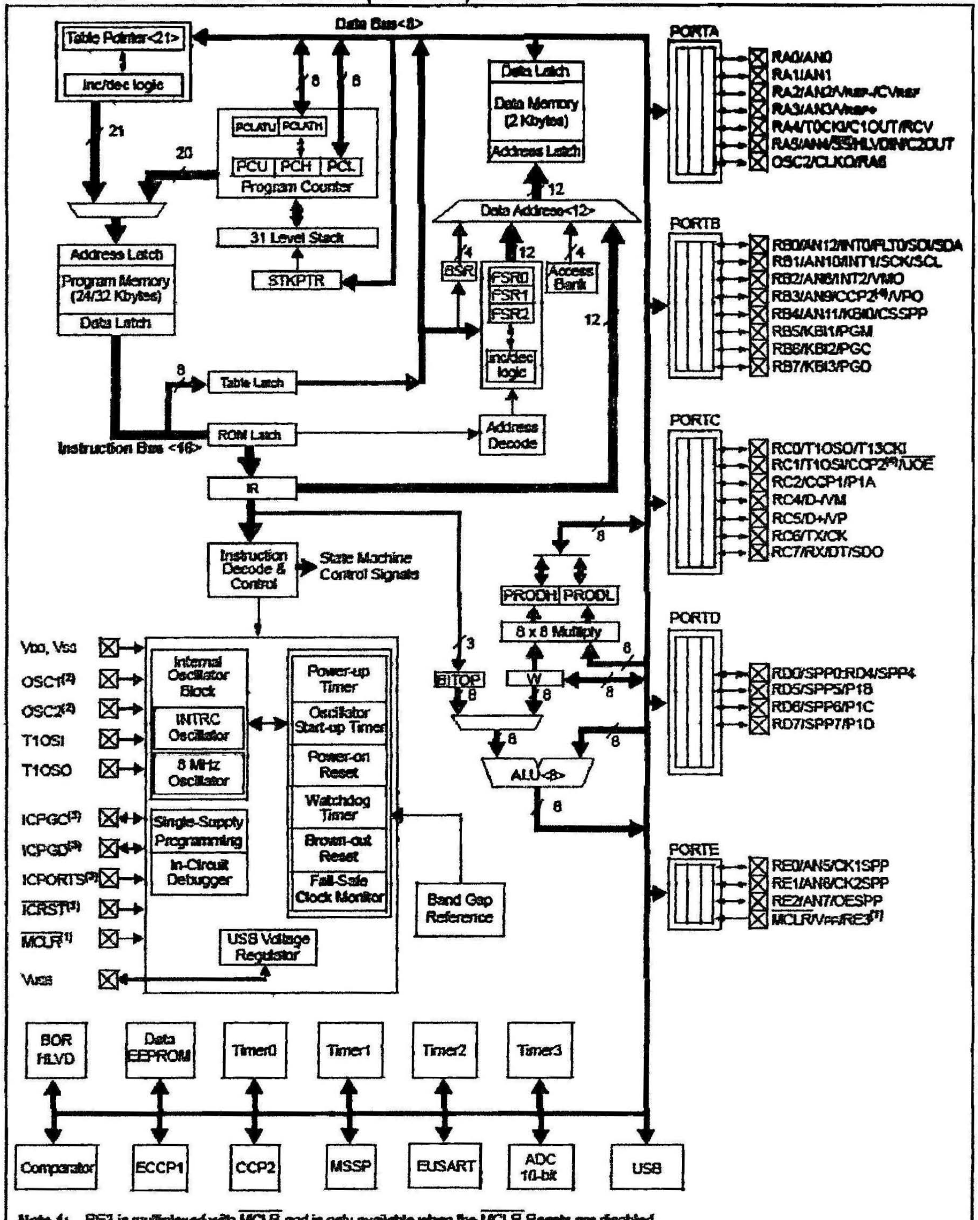
The pinouts for all devices are listed in Table 1-2 and Table 1-3.

Like all Microchip PIC18 devices, members of the PIC18F2455/2550/4455/4550 family are available as both standard and low-voltage devices. Standard devices with Enhanced Flash memory, designated with an "F" in the part number (such as PIC18F2550), accommodate an operating VDD range of 4.2V to 5.5V. Low-voltage parts, designated by "LF" (such as PIC18LF2550), function over an extended VDD range of 2.0V to 5.5V.

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), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), 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 $\overline{MCLR}$ and is only available when the $\overline{MCLR}$ 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.

PIC18F2455/2550/4455/4550

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
MCLR/VPP/RE3	1	18	18			Master Clear (input) or programming voltage (input).
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P		Programming voltage input.
RE3				I	ST	Digital input.
OSC1/CLKI	13	32	30			Oscillator crystal or external clock input.
OSC1				I	Analog	Oscillator crystal input or external clock source input.
CLKI				I	Analog	External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pins.)
OSC2/CLKO/RA6	14	33	31			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6				I/O	TTL	General purpose I/O pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX configuration bit is cleared.
2: Default assignment for CCP2 when CCP2MX configuration bit is set.
3: These pins are No Connect unless the ICPRT configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTA is a bidirectional I/O port.
RA0/AN0	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
CVREF				O	Analog	Analog comparator reference output.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT/RCV	6	23	23			
RA4				I/O	ST	Digital I/O.
T0CKI				I	ST	Timer0 external clock input.
C1OUT				O	—	Comparator 1 output.
RCV				I	TTL	External USB transceiver RCV input.
RA5/AN4/SS/HLVDIN/C2OUT	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
SS				I	TTL	SPI™ slave select input.
HLVDIN				I	Analog	High/Low-Voltage Detect input.
C2OUT				O	—	Comparator 2 output.
RA6	—	—	—	—	—	See the OSC2/CLKO/RA6 pin.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number PDIP	Pin Number QFN	Pin Number TQFP	Pin Type	Buffer Type	Description
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/FLT0/SDI/SDA	33	9	8			
RB0				I/O	TTL	Digital I/O.
AN12				I	Analog	Analog input 12.
INT0				I	ST	External interrupt 0.
FLT0				I	ST	Enhanced PWM Fault input (ECCP1 module).
SDI				I	ST	SPI™ data in.
SDA				I/O	ST	I²C™ data I/O.
RB1/AN10/INT1/SCK/SCL	34	10	9			
RB1				I/O	TTL	Digital I/O.
AN10				I	Analog	Analog input 10.
INT1				I	ST	External interrupt 1.
SCK				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL				I/O	ST	Synchronous serial clock input/output for I²C mode.
RB2/AN8/INT2/VMO	35	11	10			
RB2				I/O	TTL	Digital I/O.
AN8				I	Analog	Analog input 8.
INT2				I	ST	External interrupt 2.
VMO				O	—	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	36	12	11			
RB3				I/O	TTL	Digital I/O.
AN9				I	Analog	Analog input 9.
CCP2(1)				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
VPO				O	—	External USB transceiver VPO output.
RB4/AN11/KBI0/CSSPP	37	14	14			
RB4				I/O	TTL	Digital I/O.
AN11				I	Analog	Analog input 11.
KBI0				I	TTL	Interrupt-on-change pin.
CSSPP				O	—	SPP chip select control output.
RB5/KBI1/PGM	38	15	15			
RB5				I/O	TTL	Digital I/O.
KBI1				I	TTL	Interrupt-on-change pin.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	Digital I/O.
KBI2				I	TTL	Interrupt-on-change pin.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	Digital I/O.
KBI3				I	TTL	Interrupt-on-change pin.
PGD				I/O	ST	In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX configuration bit is set.

3: These pins are No Connect unless the ICPRT configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	15	34	32			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/UOE	16	35	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2[2]				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
UOE				O	—	External USB transceiver OE output.
RC2/CCP1/P1A	17	36	36			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
P1A				O	TTL	Enhanced CCP1 PWM output, channel A.
RC4/D-/VM	23	42	42			
RC4				I	TTL	Digital input.
D-				I/O	—	USB differential minus line (input/output).
VM				I	TTL	External USB transceiver VM input.
RC5/D+/VP	24	43	43			
RC5				I	TTL	Digital input.
D+				I/O	—	USB differential plus line (input/output).
VP				I	TTL	External USB transceiver VP input.
RC6/TX/CK	25	44	44			
RC6				I/O	ST	Digital I/O.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO	26	1	1			
RC7				I/O	ST	Digital I/O.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see TX/CK).
SDO				O	—	SPI™ data out.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTD is a bidirectional I/O port or a Streaming Parallel Port (SPP). These pins have TTL input buffers when the SPP module is enabled.
RD0/SPP0	19	38	38			
RD0				I/O	ST	Digital I/O.
SPP0				I/O	TTL	Streaming Parallel Port data.
RD1/SPP1	20	39	39			
RD1				I/O	ST	Digital I/O.
SPP1				I/O	TTL	Streaming Parallel Port data.
RD2/SPP2	21	40	40			
RD2				I/O	ST	Digital I/O.
SPP2				I/O	TTL	Streaming Parallel Port data.
RD3/SPP3	22	41	41			
RD3				I/O	ST	Digital I/O.
SPP3				I/O	TTL	Streaming Parallel Port data.
RD4/SPP4	27	2	2			
RD4				I/O	ST	Digital I/O.
SPP4				I/O	TTL	Streaming Parallel Port data.
RD5/SPP5/P1B	28	3	3			
RD5				I/O	ST	Digital I/O.
SPP5				I/O	TTL	Streaming Parallel Port data.
P1B				O	—	Enhanced CCP1 PWM output, channel B.
RD6/SPP6/P1C	29	4	4			
RD6				I/O	ST	Digital I/O.
SPP6				I/O	TTL	Streaming Parallel Port data.
P1C				O	—	Enhanced CCP1 PWM output, channel C.
RD7/SPP7/P1D	30	5	5			
RD7				I/O	ST	Digital I/O.
SPP7				I/O	TTL	Streaming Parallel Port data.
P1D				O	—	Enhanced CCP1 PWM output, channel D.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTE is a bidirectional I/O port.
RE0/AN5/CK1SPP	8	25	25			
RE0				I/O	ST	Digital I/O.
AN5				I	Analog	Analog input 5.
CK1SPP				O	—	SPP clock 1 output.
RE1/AN6/CK2SPP	9	26	26			
RE1				I/O	ST	Digital I/O.
AN6				I	Analog	Analog input 6.
CK2SPP				O	—	SPP clock 2 output.
RE2/AN7/OESPP	10	27	27			
RE2				I/O	ST	Digital I/O.
AN7				I	Analog	Analog input 7.
OESPP				O	—	SPP output enable output.
RE3	—	—	—	—	—	See $\overline{MCLR}$/VPP/RE3 pin.
VSS	12, 31	6, 30, 31	6, 29	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	7, 8, 28, 29	7, 28	P	—	Positive supply for logic and I/O pins.
VUSB	18	37	37	O	—	Internal USB 3.3V voltage regulator output.
NC/ICCK/ICPGC	—	—	12			No Connect or dedicated ICD/ICSP™ port clock.[3]
ICCK				I/O	ST	In-Circuit Debugger clock.
ICPGC				I/O	ST	ICSP programming clock.
NC/ICDT/ICPGD	—	—	13			No Connect or dedicated ICD/ICSP port clock.[3]
ICDT				I/O	ST	In-Circuit Debugger data.
ICPGD				I/O	ST	ICSP programming data.
NC/$\overline{ICRST}$/ICVPP	—	—	33			No Connect or dedicated ICD/ICSP port Reset.[3]
$\overline{ICRST}$				I	—	Master Clear (Reset) input.
ICVPP				P	—	Programming voltage input.
NC/ICPORTS	—	—	34	P	—	No Connect or 28-pin device emulation.[3]
ICPORTS						Enable 28-pin device emulation when connected to Vss.
NC	—	13	—	—	—	No Connect.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX configuration bit is set.

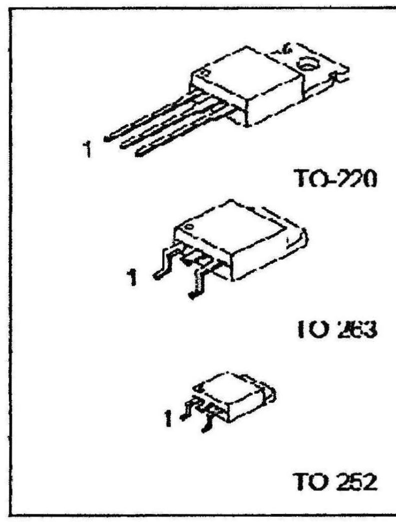
3: These pins are No Connect unless the ICPRT configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the $\overline{DEBUG}$ configuration bit is cleared.

APPENDIX C

REGULATOR LM7805 & LM7806 Datasheet

UTC LM78XX LINEAR INTEGRATED CIRCUIT

3-TERMINAL 1A POSITIVE VOLTAGE REGULATOR



1: Input 2: GND 3: Output

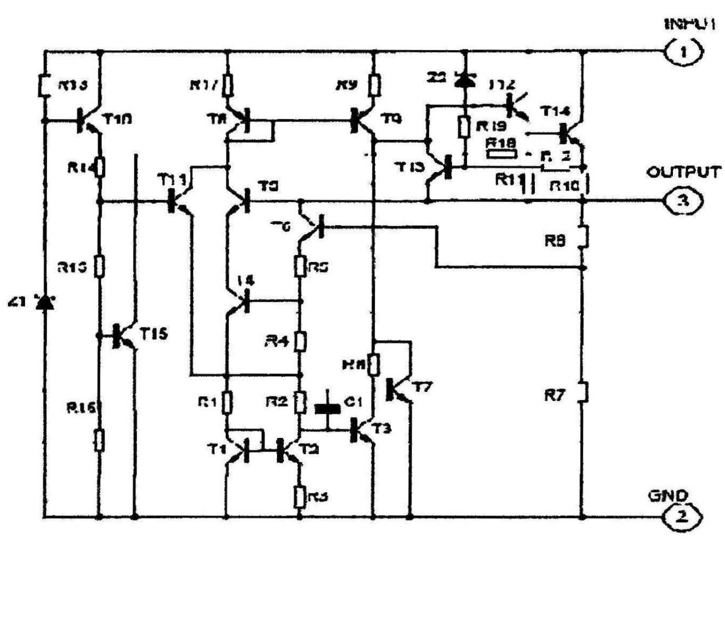
DESCRIPTION

The UTC 78XX family is monolithic fixed voltage regulator integrated circuit. They are suitable for applications that required supply current up to 1 A.

FEATURES

*Output current up to 1.5 A
*Fixed output voltage of 5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V and 24V available
*Thermal overload shutdown protection
*Short circuit current limiting
*Output transistor SOA protection

TEST CIRCUIT



UTC LM78XX LINEAR INTEGRATED CIRCUIT

ABSOLUTE MAXIMUM RATINGS

(Operating temperature range applies unless otherwise specified)

PARAMETER	SYMBOL	RATING		UNIT
Input voltage(for Vo=5~18V)			35	V
(for Vo=24V)	Vi		40	V
Output Current	Io		1	A
Power Dissipation	PD	Internally	Limited	W
Operating Junction Temperature Range	TOPR	-20	+150	°C
Storage Temperature Range	TSTG	-55	+150	°C

UTC LM7805 ELECTRICAL CHARACTERISTICS

(Vi=10V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

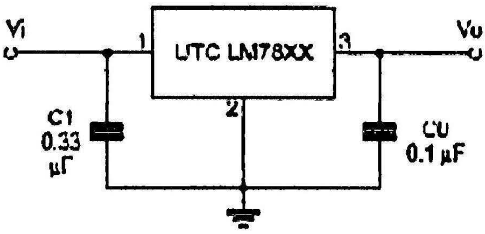
PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	4.80	5.0	5.20	V
		Vi =7.5V to 20V, Io=5mA - 1.0A,PD<15W	4.75		5.25	V
Load Regulation	ΔVo	Tj=25°C,Io=5mA - 1.5A			50	mV
		Tj=25°C,Io=0.25A - 0.75A			25	mV
Line regulation	ΔVo	Vi =7V to 25V,Tj=25°C			50	mV
		Vi =7.5V to 20V,Tj=25°C,Io=1A			50	mV
Quiescent Current	Iq	Tj=25°C, Io=<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =7.5V to 20V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		40		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-0.6		mV/°C
Ripple Rejection	RR	Vi =8V - 18V,f=120Hz,Tj=25°C	62	80		dB
Peak Output Current	IPK	Tj=25°C		1.8		A
Short-Circuit Current	Isc	Vi=35V, Tj=25°C		250		mA
Dropout Voltage	Vd	Tj=25°C		2.0		V

UTC LM7808 ELECTRICAL CHARACTERISTICS

(Vi=11V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	5.76	6.0	6.24	V
		Vi =8.5V to 21V, Io=5mA - 1.0A, PD<15W	5.70		6.30	V
Load Regulation	ΔVo	Tj=25°C,Io=5mA - 1.5A			60	mV
		Tj=25°C,Io=0.25A - 0.75A			30	mV
Line regulation	ΔVo	Vi =8V to 25V,Tj=25°C			60	mV
		Vi =8.5V to 21V,Tj=25°C,Io=1A			60	mV
Quiescent Current	Iq	Tj=25°C, Io=<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =8.5V to 21V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		45		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-0.7		mV/°C
Ripple Rejection	RR	Vi =9V - 19V,f=120Hz,Tj=25°C	59	75		dB

APPLICATION CIRCUIT



Note 1: To specify an output voltage, substitute voltage value for "XX".
Note 2: Bypass capacitors are recommended for optimum stability and transient response and should be located as close as possible to the regulators.

APPENDIX D

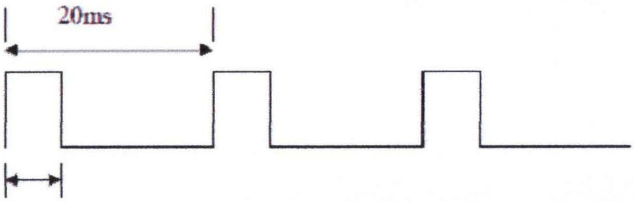
SERVO MOTOR Datasheet

Cytron
Technologies

RC Servo
C36R, C40R, C55R

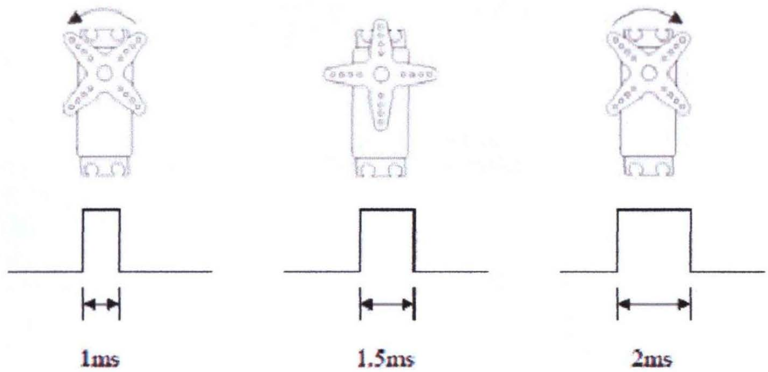
2. HOW RC SERVO MOTOR WORKS

Servos are controlled by sending them a pulse of variable width. The signal wire is used to send this pulse. The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction. It is important to note that different servos will have different constraints on their rotation.



The angle is determined by the duration of a pulse that is applied to the signal wire. This is called Pulse Width Modulation. The servo expects to see a pulse every 20 ms. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position). However, the **exact correspondence between pulse width and servo varies from one servo manufacturer to another.1.5ms is not necessarily neutral or middle position.**

The position pulse must be repeated to instruct the servo to stay in position. When a pulse is sent to a servo that is less than 1.5 ms the servo rotates to a position and holds its output shaft some number of degrees counterclockwise from the neutral point. When the pulse is wider than 1.5 ms the opposite occurs. The minimal width and the maximum width of pulse that will command the servo to turn to a valid position are functions of each servo. Different brands, and even different servos of the same brand, will have different maximum and minimums. Generally the minimum pulse will be about 1 ms wide (some servo is 0.5ms) and the maximum pulse will be 2 ms wide (some servo is 2.5ms).



Another parameter that varies from servo to servo is the turn rate. This is the time it takes from the servo to change from one position to another. The turning rate and torque value can be check at product specification section.

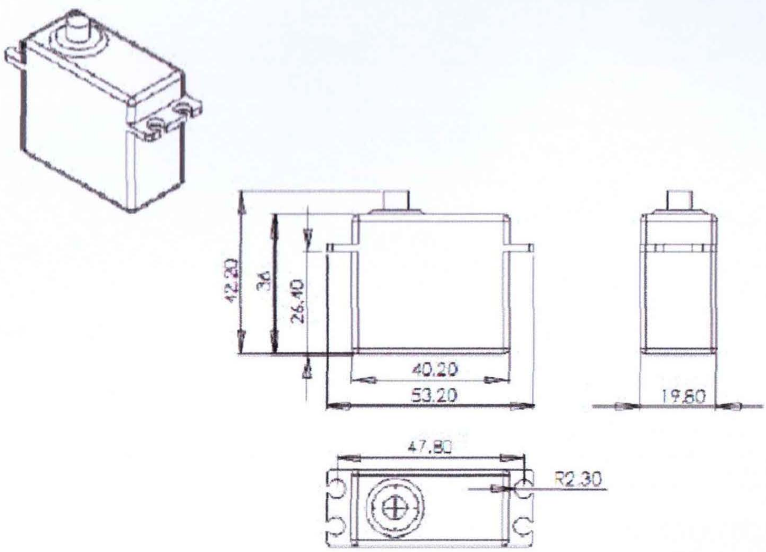
3. PRODUCT SPECIFICATION

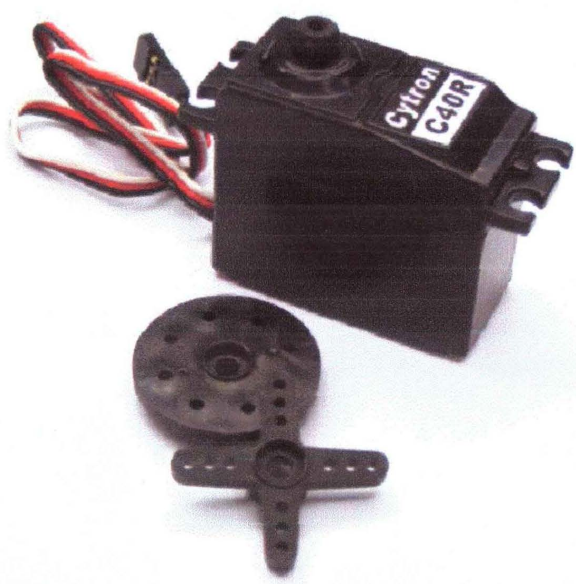
Cytron Technologies offer great range of RC servo motor. With the combination of various gear type, speed, torque and voltage, users are free to choose the suitable RC servo for project development. Of course, it can also be used for RC application. Below is product specification for Cytron RC Servo motor.

Specification		Servo Motor Model		
		C36R	C40R	C55R
4.8V	Speed (s/60°)	0.16	0.19	0.22
	Torque (Kg.cm)	3.5	6.00	9.00
6.0V	Speed (s/60°)	0.14	0.16	0.20
	Torque (Kg.cm)	4.50	7.00	11.0
7.0V	Speed (s/60°)	-	-	0.17
	Torque (Kg.cm)	-	-	13.00
Signal To Control Angle		TTL PWM	TTL PWM	TTL PWM
PWM At Min Angle (ms)		0.5	0.54	0.582
PWM At Max Angle (ms)		2.35	2.40	2.50
Operating Voltage (VDC)		4.8-6.0	4.8-6.0	4.8-7.0
Operating Frequency (Hz)		50.0	50.0	50.0
Moving Range(degree)		0-180	0-180	0-180
Wiring (Black/Brown Wire)		Ground	Ground	Ground
Wiring (Red Wire)		4.8-6.0	4.8-6.0	4.8-7.0
Wiring (Orange/Other Wire)		PWM signal	PWM signal	PWM signal
Dimension (mm)		~ 40.2x19.8x36	~ 40.2x19.8x36	~ 41x20x37
Weight (g)		36.0	38.0	55.0
Gear material		Plastic Gear	Plastic Gear	Metal Gear
Servo type		Standard	Standard	Standard

* **Pulse width range is for reference only; please start the servo calibration at 1.5ms.**

C40R RC Servo

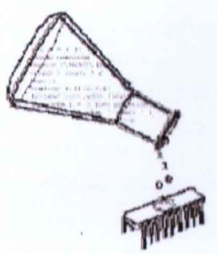




APPENDIX E

PIC BASIC PRO Compiler

PicBasic Pro Compiler



microEngineering Labs, Inc.

COPYRIGHT NOTICE

Copyright © 2004 microEngineering Labs, Inc.
All rights reserved.

This manual describes the use and operation of the PicBasic Pro Compiler from microEngineering Labs, Inc. Use of the PicBasic Pro Compiler without first obtaining a license is a violation of law. To obtain a license, along with the latest version of the product and documentation, contact microEngineering Labs, Inc.

Publication and redistribution of this manual over the Internet or in any other medium without prior written consent is expressly forbidden. In all cases this copyright notice must remain intact and unchanged.

microEngineering Labs, Inc.
Box 60039
Colorado Springs CO 80960-0039
(719) 520-5323
(719) 520-1867 fax
email: support@melabs.com
web: www.melabs.com

TRADEMARKS

EPIC and PicBasic Pro are trademarks of microEngineering Labs, Inc.
BASIC Stamp is a trademark of Parallax, Inc.
PICmicro is a registered trademark of Microchip Technology Inc.

Any PICmicro MCU with analog inputs, such as the PIC16C7xx, PIC16F87x and PIC12C67x series devices, will come up in analog mode. You must set them to digital if that is how you intend to use them:

```
ADCON1 = 7
```

For the PIC12F675 and 16F676, a different register must be set instead:

```
ANSEL = 0
```

While these settings work for many devices, you will need to check the data sheet for the specific device to verify the exact settings.

Another example of potential disaster is that PORTA, pin 4 exhibits unusual behavior when used as an output. This is because the pin has an open drain output rather than the usual bipolar stage of the rest of the output pins. This means it can pull to ground when set to 0, but it will simply float when set to a 1, instead of going high. To make this pin act in the expected manner, add a pull-up resistor between the pin and 5 volts. The value of the resistor may be between 1K and 33K, depending on the drive necessary for the connected input. This pin acts as any other pin when used as an input.

Some PICmicro MCUs, such as the PIC16F62x, 87x(A) and PIC18Fxxx allow low-voltage programming. This function takes over one of the PORTB pins and can cause the device to act erratically if this pin is not pulled low. It is best to make sure that low-voltage programming is not enabled at the time the PICmicro MCU is programmed.

All of the PICmicro MCU pins are set to inputs on power-up. If you need a pin to be an output, set it to an output before you use it, or use a PicBasic Pro command that does it for you. Once again, review the PICmicro MCU data sheets to become familiar with the idiosyncrasies of a particular part.

There is no data direction (TRIS) register for PORTA on PIC17Cxxx devices. Therefore, commands that rely on the TRIS register for their operation, such as **I2CREAD** and **I2CWRITE**, may not be used on PORTA.

The name of the port pins on the PIC12C5xx, 12CE5xx, 12C67x, 12CE67x and 12F6xx devices is GPIO. The name for the TRIS register is TRISIO.

```
GPIO.0 = 1
TRISIO = %101010
```

On the PIC12C5xx and 12CE5xx devices, pin GPIO.2 is forced to an input regardless of the setting of the TRIS register. To allow this pin to be used as a standard I/O pin, add the following line to the beginning of the program:

```
OPTION_REG.5 = 0
```

As hinted at above, the name of the OPTION register on all PICmicro MCUs is OPTION_REG.

Certain PICmicro MCUs have on-chip non-volatile data storage implemented like an I2C interfaced serial EEPROM. **READ** and **WRITE** will not work on devices with on-chip I2C interfaced serial EEPROM like the PIC12CE51x, 12CE67x and 16CE62x parts. Use the **I2CREAD** and **I2CWRITE** instructions instead.

Some PICmicro MCUs, such as the PIC12C67x, 12CE67x and 12F6xx, have on-chip RC oscillators. These devices contain an oscillator calibration factor in the last location of code space. The on-chip oscillator may be fine-tuned by acquiring the data from this location and moving it into the OSCCAL register. Two **DEFINE**s have been created to perform this task automatically each time the program starts:

```
Define OSCCAL_1K 1        ' Set OSCCAL for 1K
                          device
Define OSCCAL_2K 1        ' Set OSCCAL for 2K
                          device
```

Add one of these 2 **DEFINE**s near the beginning of the PicBasic Pro program to perform the setting of OSCCAL.

If a UV erasable device has been erased, the value cannot be read from memory. If one of these **DEFINE**s is used on an erased part, it will cause

2.6.2. Pin and Variable Names

Make the name of a pin or variable something more coherent than **Pin0** or **B1**. In addition to the liberal use of comments, descriptive pin and variable names can greatly enhance readability. The following code fragment demonstrates:

```
BattLED var PORTB.0        ' Low battery LED
level var   byte           ' Variable will contain the
                           battery level

       If level < 10 Then        ' If batt level is low
              High BattLED       ' Turn on the LED
       Endif
```

2.6.3. Labels

Labels should also be more meaningful than "label1:" or "here:". Even a label like "loop:" is more descriptive (though only slightly). Usually the line or routine you are jumping to does something unique. Try and give at least a hint of its function with the label, and then follow up with a comment.

2.6.4. GOTO

Try not to use too many **GOTO**s. While **GOTO**s may be a necessary evil, try to minimize their use as much as possible. Try to write your code in logical sections and not jump around too much. **GOSUB**s can be helpful in achieving this.

If a port does not have 8 pins, such as PORTA, only the pin numbers that exist may be used, i.e. 8 - 12. Using pin numbers 13 - 15 will have no discernable effect.

This pin number, 0 - 15, has nothing to do with the physical pin number of a PICmicro MCU. Depending on the particular PICmicro MCU, pin number 0 could be physical pin 6, 21 or 33, but in each case it maps to PORTB.0 (or GPIO.0 for 8-pin devices, or PORTC.0 for a PIC14000).

```
        High 0        ' Set PORTB.0 (or GPIO.0) high

        B0 = 9        ' Select PORTC.1 (or PORTA.1)
        Toggle B0     ' Toggle PORTC.1 (or PORTA.1)
```

Pins may be referenced by number (0 - 15), name (e.g. **Pin0**, if *BS1DEFS.BAS or BS2DEFS.BAS is included or you have defined them yourself*), or full bit name (e.g. PORTA.1). Any pin or bit of the microcontroller can be accessed using the latter method.

The pin names (i.e. **Pin0**) are not automatically included in your program. In most cases, you would define pin names as you see fit using the **VAR** command:

```
        led   Var   PORTA.3
```

However, two definition files have been provided to enhance BASIC Stamp compatibility. The files BS1DEFS.BAS or BS2DEFS.BAS may be included in the PicBasic Pro program to provide pin and bit names that match the BASIC Stamp names.

```
        Include "bs1defs.bas"
```

or

```
        Include "bs2defs.bas"
```

BS1DEFS.BAS defines **Pins**, **B0-B13**, **W0-W6** and most of the other BS1 pin and variable names.

BS2DEFS.BAS defines **Ins**, **Outs**, **InL**, **InH**, **OutL**, **OutH**, **B0-B25**, **W0-W12** and most of the other BS2 pin and variable names.

PORTL and **PORTH** are also defined in PBP. **PORTL** encompasses pin 0 - 7 and **PORTH** encompasses pins 8 – 15.

When a PICmicro MCU powers-up, all of the pins are set to input. To use a pin as an output, the pin or port must be set to an output or a command must be used that automatically sets a pin to an output.

To set a pin or port to an output (or input), set its TRIS register. Setting a TRIS bit to 0 makes its corresponding port pin an output. Setting a TRIS bit to 1 makes its corresponding port pin an input. For example:

```
        TRISA = %00000000         ' Or TRISA = 0
```

sets all the PORTA pins to outputs.

```
        TRISB = %11111111         ' Or TRISB = 255
```

sets all the PORTB pins to inputs.

```
        TRISC = %10101010
```

Sets all the even pins on PORTC to outputs, and the odd pins to inputs. Individual bit directions may be set in the same manner.

```
        TRISA.0 = 0
```

sets PORTA, pin 0 to an output. All of the other pin directions on PORTA are unchanged.

The BASIC Stamp variable names **Dirs**, **Dirh**, **Dirl** and **Dir0-Dir15** are not defined and should not be used with the PicBasic Pro Compiler. TRIS should be used instead, but has the opposite state of **Dirs**.

This **does not** work in PicBasic Pro:

```
        Dir0 = 1      ' Doesn't set pin PORTB.0 to output
```

Do this instead:

```
        TRISB.0 = 0 ' Set pin PORTB.0 to output
```