

The MegaPIC - PIC C Briefing

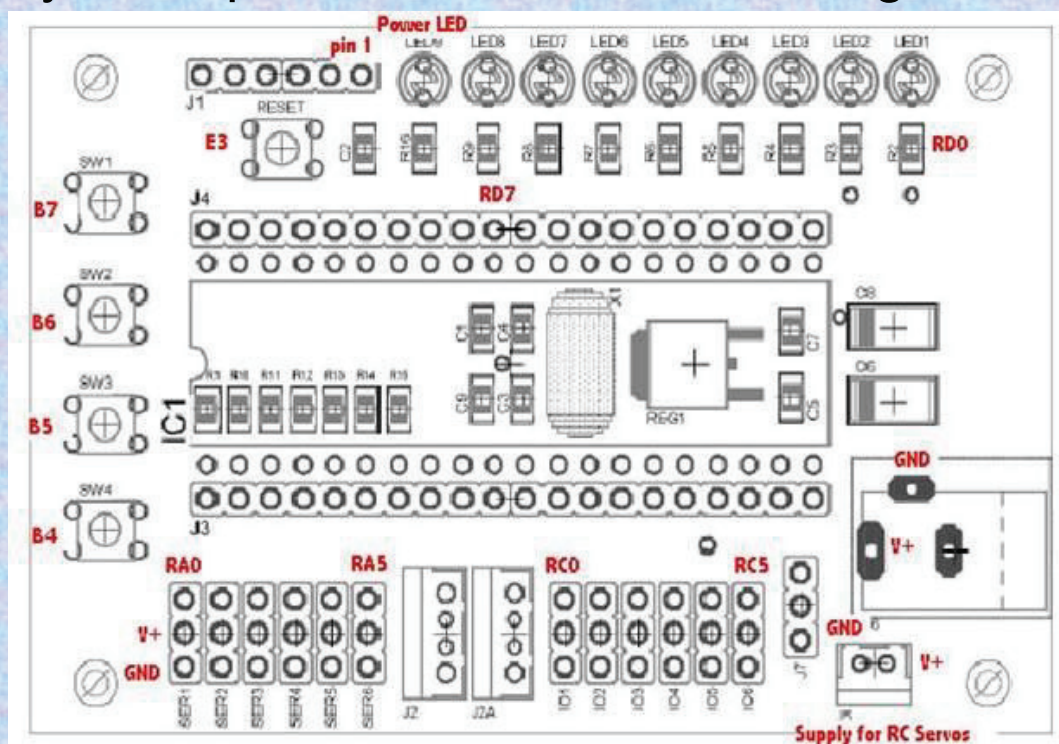
- Mr. Yee Choon Seng
- Email : engyys@nus.edu.sg or csyee@simtech.a-star.edu.sg
- Design Project resources
 - <http://guppy.mpe.nus.edu.sg/me3design.html>
- One-Stop robotics shop
 - A-Main Objectives Pte Ltd, Block 1 Rochor Road, #02-608, Rochor Centre, S180001, Tel: 63960745
 - <http://www.amianobj.com>
 - <http://shop.rotor.com.sg/>
 - <http://www.singaporehobby.com/>
 - <http://www.sgbotic.com/>
 - <http://www.roboworld.com.sg/>
 - <http://www.robot-r-us.com/>

C.S. Yee

1

The MegaPIC - PIC C Briefing

- Layout of ports & connectors on MegaPIC



C.S. Yee

2

The MegaPIC - PIC C Briefing

- MegaPIC Features

- PIC18F4520 microcontroller (40-pin DIP package)
- Input voltage from +5V to +35V
- Onboard 16MHz oscillator crystal
- 6 universal Input/Output (I/O) pins
- 6 RC Servo output ports
- 4 user programmable pushbuttons
- 8 LED outputs
- PICKIT2 programmer for in-circuit programming

The MegaPIC - PIC C Briefing

- MegaPIC - I/O configuration

Pushbuttons (Port B)

- Push Button 1 (SW1) – RB7 (Pin 40)
- Push Button 2 (SW2) – RB6 (Pin 39)
- Push Button 3 (SW3) – RB5 (Pin 38)
- Push Button 4 (SW4) – RB4 (Pin 37)

RC Servo Motor Connectors (Port A)

- Servo Connector 1 (SER1) – RA0 (Pin 02)
- Servo Connector 2 (SER2) – RA1 (Pin 03)
- Servo Connector 3 (SER3) – RA2 (Pin 04)
- Servo Connector 4 (SER4) – RA3 (Pin 05)
- Servo Connector 5 (SER5) – RA4 (Pin 06)
- Servo Connector 6 (SER6) – RA5 (Pin 07)

The MegaPIC - PIC C Briefing

- MegaPIC - I/O configuration

Universal I/O pins (Port C)

- Universal I/O 1 (IO1) – RC0 (Pin 15)
- Universal I/O 2 (IO2) – RC1 (Pin 16)
- Universal I/O 3 (IO3) – RC2 (Pin 17)
- Universal I/O 4 (IO4) – RC3 (Pin 18)
- Universal I/O 5 (IO5) – RC4 (Pin 23)
- Universal I/O 6 (IO6) – RC5 (Pin 24)

LED Display (Port D)

- ○ LED 1 – RD0 (Pin 19)
- ○ LED 2 – RD1 (Pin 20)
- ○ LED 3 – RD2 (Pin 21)
- ○ LED 4 – RD3 (Pin 22)
- ○ LED 5 – RD4 (Pin 27)
- ○ LED 6 – RD5 (Pin 28)
- ○ LED 7 – RD6 (Pin 29)
- ○ LED 8 – RD7 (Pin 30)

The MegaPIC - PIC C Briefing

- MegaPIC - I/O configuration

Jumper & Power Connectors Setting

- J1 is meant for PICKit2 programmer connection. Pin 1 is next to the power LED, i.e. LED9.
- J5 is meant for +5V supply input, which can be used to power the Controller board as well as the RC Servo Motors. You have to ensure the current from this power source is enough to power all RC Servo Motors as well as Controller Board.
- J6 is the DC Jack for power between +7V to +35V. Please ensure the polarity of the DC Jack is correct, i.e. Core of Jack is +5V and Shell of Jack is Ground. If reversed, you will burn the power regulator on the Controller Board.
- J7 is a power source selector for the inputs to the Micro-Controller. Shorting Pins 1 and 2 will route power from J5 to the Micro-Controller. When Pins 2 and 3 are shorted, power from the DC Jack will be used to power the Micro-Controller.

The MegaPIC - PIC C Briefing

- How to start using PIC C ?

Header files according to Compilers

IAR EWPIC18: *#include <io18f4520.h>*

MPLAB C18: *#include <p18f4520.h>*

main() function

```
void main ( )        // Declare the function  
{  
    // Do something simple.  
}
```

The MegaPIC - PIC C Briefing

- Declaring functions

```
int square (int n)  
{  
    return (n * n);  
    // return the value of n * n.  
}
```

The MegaPIC - PIC C Briefing

- Data Objects
 - int : 16-bit integer number, -32,768 to +32,767
 - char: 8-bit character or value, 0 to 255, eg 'x'
 - Decimal number : 1234
 - Hexadecimal : 0x1F (which is decimal 31)
 - Binary : 0b00011111 (decimal 31)
- Expansion of data objects:
 - Arrays, homogeneous list of any type
 - Structure (struct), non-homogeneous group of related data.

The MegaPIC - PIC C Briefing

- Operators (Integers only)
 - Arithmetic +, -, *, /
 - Comparison >, <, ==, >=, <=
 - Bitwise arithmetic | (OR), & (AND), ^ (XOR), ~ (NOT)
 - Boolean Arithmetic || (OR), && (AND), ! (NOT)
- Operators for Long Integers
 - Limited to +, -, *, >, <, ==, >= and <=
- Operators for float
 - Maths & compare +, -, *, /, >, <, ==, >= and <=
 - Some trigonometric, logarithmic and exponential functions (try not to use them.)

The MegaPIC - PIC C Briefing

- Expressions

Simple maths expression:

`a = a + 3;`

is the same as:

`a += 3;`

`a++;` is the same as `a += 1;`

How about `++a` ?

The MegaPIC - PIC C Briefing

- Control Flow 1 : if – else

Syntax:

```
if (expression or condition)
    <statement or list of statements in { }>;
else
    <alternative statement(s) in { }>;
```

Example 1:

```
if (i < 10)
    j = j + i;
else
    PORTA = j;
```

Example 2:

```
if (i < 10)    i++;
```

The MegaPIC - PIC C Briefing

- Control Flow 2 : while

Syntax:

```
while ( expression or condition )  
{  
    <statement(s)>;  
}
```

Example:

```
while ( i < 10)  
{  
    i = i + 1;  
}
```

or

```
while ( i < 10)    i++;
```

The MegaPIC - PIC C Briefing

- Control Flow 3 : for loop

Syntax:

```
for ( <expr-1>, <expr-2>, <expr-3> )  
{  
    <statement(s)>;  
}
```

<expr-1> : assignment / initial condition

<expr-2> : relational expression

<expr-3> : increment / decrement steps

Example:

```
for (i=0; i<10; i++)  
{  
    PORTA = i;    }
```

or for (i=0; i<10; i++) PORTA = i;

- Control Flow 4 : break

The **break** statement is used to get out of the **while** or **for** loop without finishing the full number of cycles.

- Preprocessor

Pre-conditions for the compiler before creating executable program for the MicroController.

Declaring constants:

```
#define PI 3.14159  
#define FORWARD 0  
#define BACKWARD 1
```

Adding header files:

```
#include <timers.h>  
#include <p18f4520.h>
```

Control how the MicroController should behave:

```
#pragma vector=0x08  
__interrupt void my_handler(void);
```

The MegaPIC - PIC C Briefing

- **Standard PIC functions**

Configuring Ports for Input or Output: (Both IAR and C18)

TRISA = 0xff;

Sets the whole PortA as input port, i.e. for sensors.

Setting to 0 for Output and 1 for Input.

Can configure pins within same port individually, i.e.

TRISB = 0x0f; Sets bit 0 to 3 as inputs and bits 4 to 7 as outputs.

Sending output through a port:

PORTA = 0xAA; Output the binary 10101010 to PortA.

RA0 = 0; Output the value 0 or Low to bit 0 of PortA. (IAR)

PORTAbits.RA0=0; (For C18)

Reading input from a bit of a port in IAR:

Var1 = RC0; Read bit 0 from PortC & store state in variable Var1.

if (RB7) Button1=1; If Switch 1 is pressed, set Button1 to 1/TRUE.

The MegaPIC - PIC C Briefing

- **Standard PIC functions**

Control of LED's:

Output 0 to the respective bits of PortD to turn ON the LED. Output 1 will turn the bit OFF.

PORTD=0b10101010; (Turns alternate LED's ON/OFF)

Recommended Compiler:

IAR Embedded Workbench

<http://www.iar.com/website1/1.0.1.0/214/1/>

Use one of your email and register for a 30-day free trial license.

You may download the installation file from

<http://guppy.mpe.nus.edu.sg/me3design.html>

From 11 Feb 2010 onwards.

.