

ET-PIC 24 WEB-V1

ET-PIC 24 WEB-V1 is PIC Board Microcontroller from Microchip that uses 16 Bit No.PIC24FJ128GA008 Microcontroller for processing data and develops board. The remarkable specification of PIC24FJ128GA008 is high speed and there is following resources as below;

○ **Central Processing Unit (CPU)**

- 16 MIPS speed to process data
- Hardware supports 16 x 16 Bit multiplication that takes only 1 Instruction Cycle
- Hardware supports 32-Bit x 16 Bit division
- C Compile is designed to Optimized Instruction Set

○ **System**

- Internal Clock Generator 31kHz to 8MHz and maximum 32MHz by using 4X PLL
- Internal LDO Voltage Regulator Circuit
- Support JTAG Boundary Scan and Flash Memory Program Support
- Fail-Safe Clock Monitor-allows safe shutdown if clock fails
- Watchdog Timer System uses Signal Clock as RC Oscillator that is separated from other parts
- Voltage 2.0- 3.6 Volt

○ **nanoWatt Power Managed Modes**

- Support RUN Mode, IDLE Mode, and SLEEP Node
- Can adjust various operation Modes of Signal Clock to be more efficient and it corresponds with power management

○ **Analog Features**

- 10-bit 16 Channel Module Analog to Digital Converter with 500 Kbit/second for speed of Sampling signal
- 2 Channel Module Analog Comparators

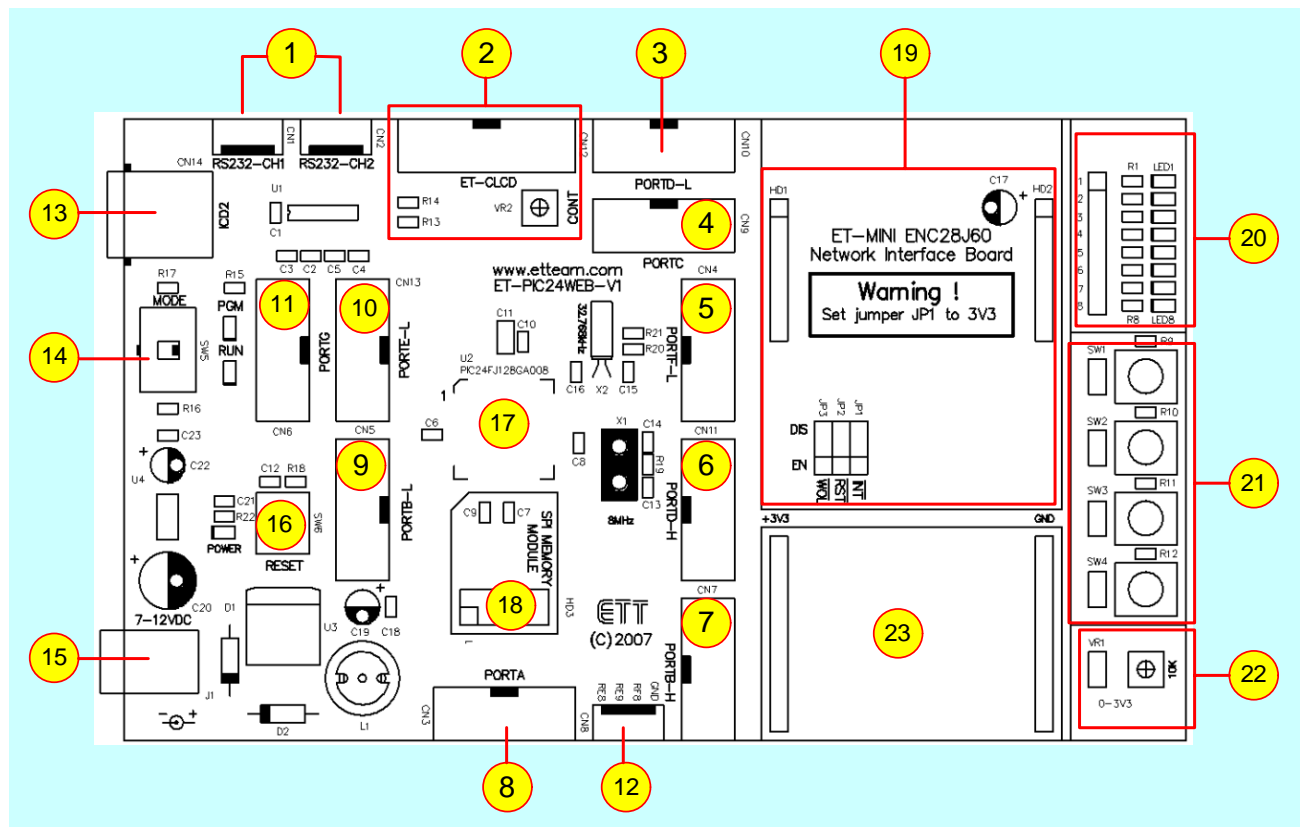
○ **Other Specifications of PIC24FJ128GA008**

- 2 Channel Module UART Communication with 4-storeyed depth of FIFO Buffer

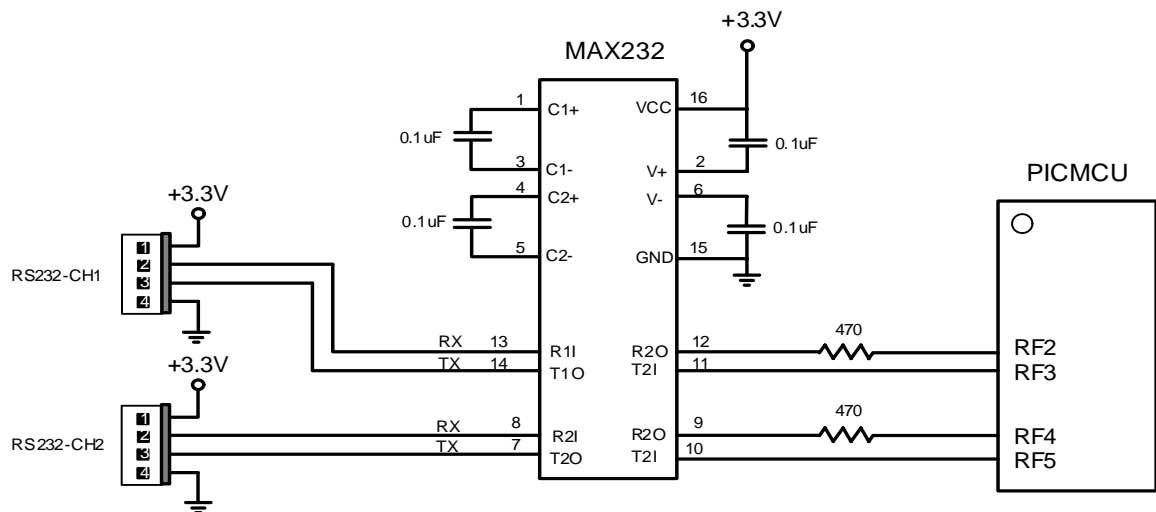
- 2 Channel Module SPI Communication with 8-storeyed dept of FIFO Buffer
- 2 Channel Module I2C™ Communication supports both Master Mode and Slave Mode
- 16-Bit 5 Module Timer
- 5 Module Capture, Compare/PWM
- Internal Hardware System as RTCC, Real-Time Clock Calendar with Alarms
- Communication System as PWP, Parallel Master Port, with 16 Address Lines, and 8/16 Data
- 128KByte Flash Memory
- 8KByte SRAM
- 70 Bit I/O Ports

▪ General Specifications of Board ET-PIC 24 WEB

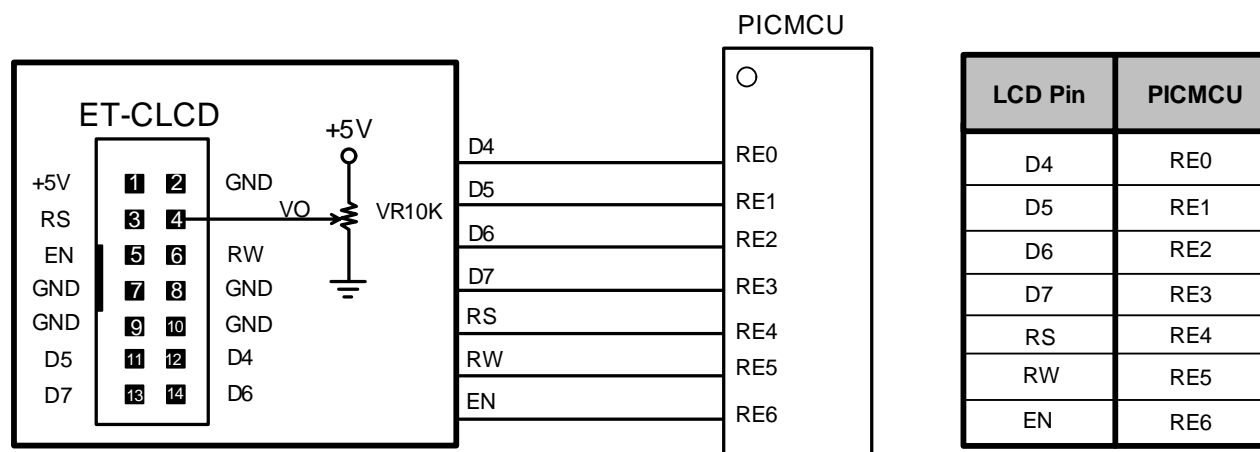
- 80 PIN PIC24FJ128GA008 Microcontroller
- 8 MHz Signal Clock Oscillator (can use x4 PLL to run up to 32 MHz)
- 9 Port 10 PIN I/O Port (ETT standard arrangement)
- 2 Port RS232 Driver Circuit
- 1 Port to connect LCD (ET-CLCD) under ETT standard arrangement
- Connect ICD2 to download program and Switch to on/off signal Run/Program
- 8 Channel LED Circuit to test Output
- 4 Channel Switch BUTTON Circuit to test Input
- 1 Channel 0-3.3V Voltage Generator Circuit from adjustable Resistor to test Module A/D
- 1 Port to connect with EEPROM 25LCxxx
- Port to connect with Module Internet ET-MINI ENC28J60
- Switching Regulator to convert signal DC Input to 3.3V
- Connector VCC and GND

Structure of Board ET-PIC 24 WEB-V1Details

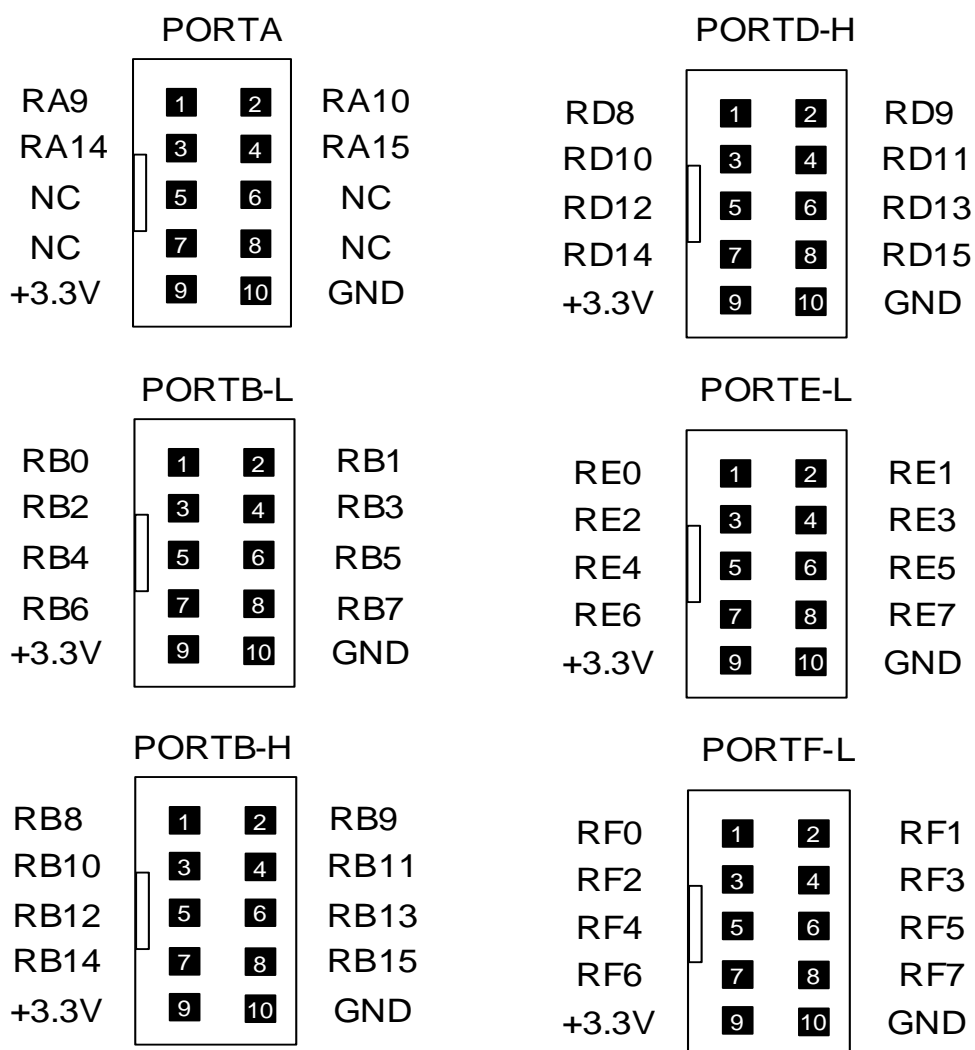
- **No.1** is 2 Port to connect Signal RS-232 and its circuit connection is shown below.

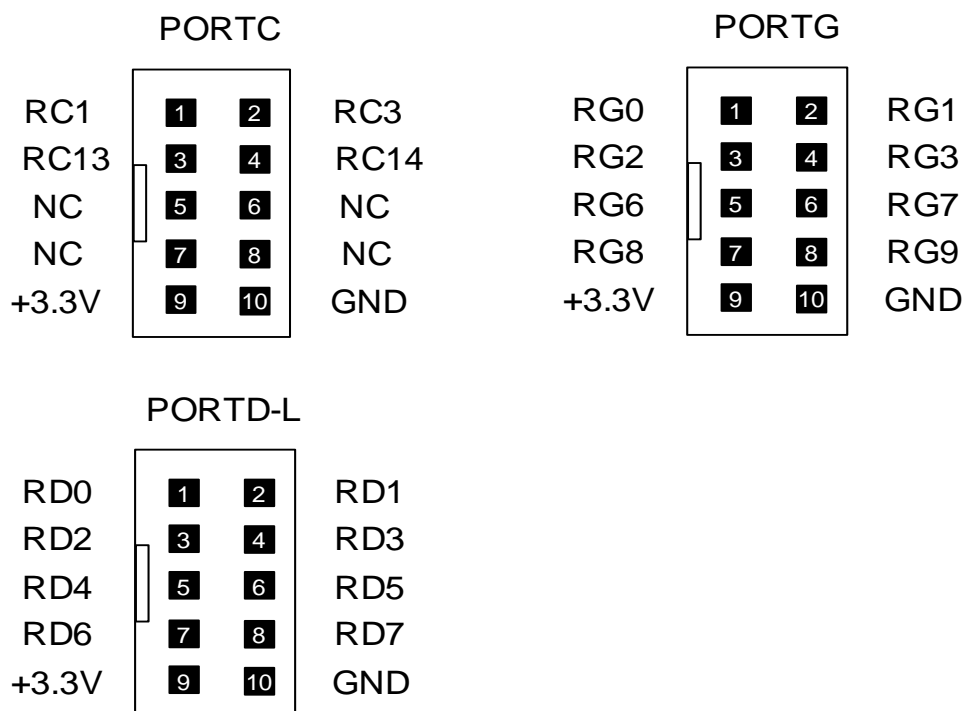


- **No.2** is Port ET-LCD to connect with Character LED Display and its pin arrangement is shown below;

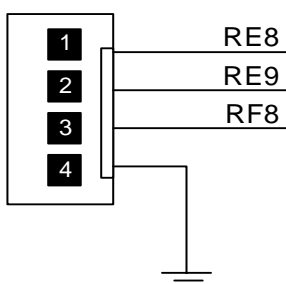


- No.3, 4, 5, 6, 7, 8, 9, 10, and 11 is Port I/O of Microcontroller that is designed to be the format of the standard 10-PIN ETT Port and each port is arranged as below;

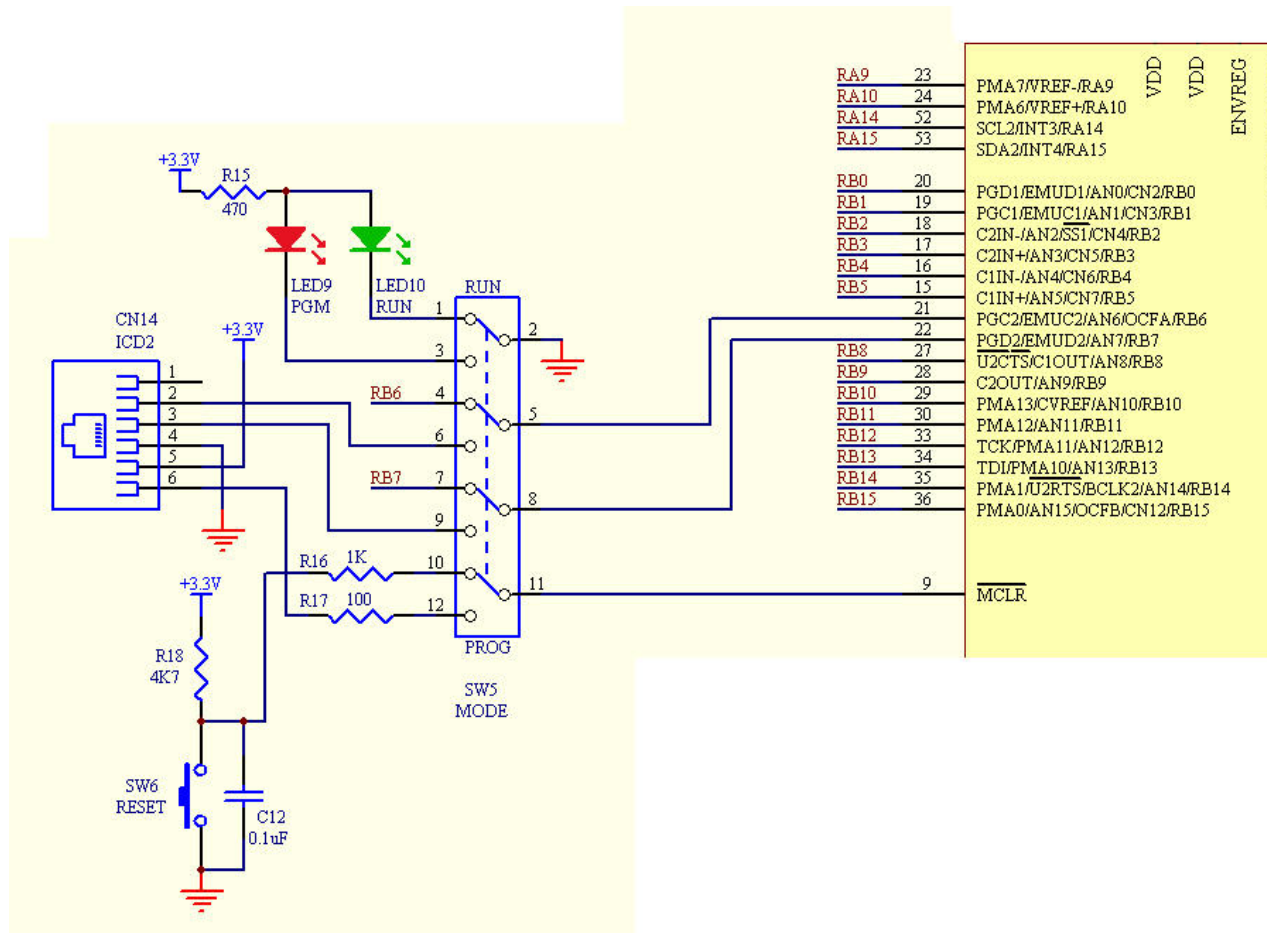




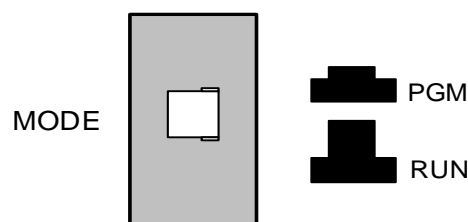
- **No.12** is 4 PIN I/O Port that consists of Signal RE8, RE9, RF8 and GND as below;



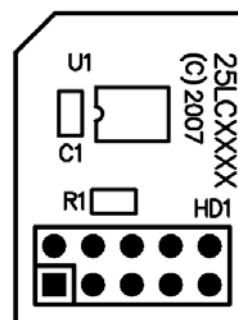
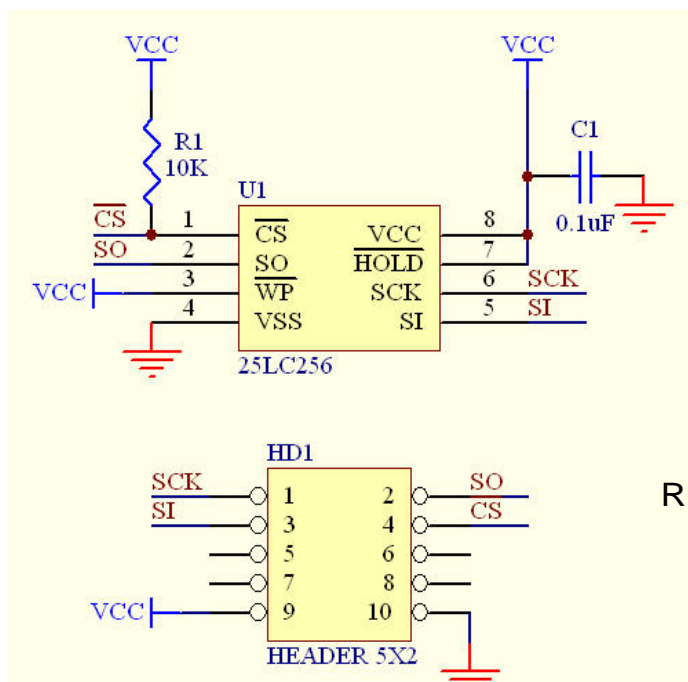
- **No.13** is Connector to download program that is arranged under the standard of ICD2 and supports Programmer that is standard ICD2 Interface such as PICKit2, ICD2 and ETT Programmer "ET-PGMPIC USB". User must always press Switch MODE to PGM position before programming to on/off signal pin with external programmer.



- **No.14** is Switch to select Mode RUN and PGM. When this Switch is shifted to PGM position, it will on/off signal pin that is used to program data code into programmer for programming the designed data program. When Switch is shifted to RUN position, it make signal pins general I/O.



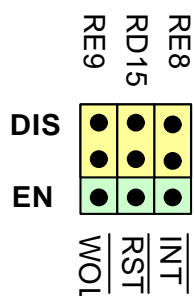
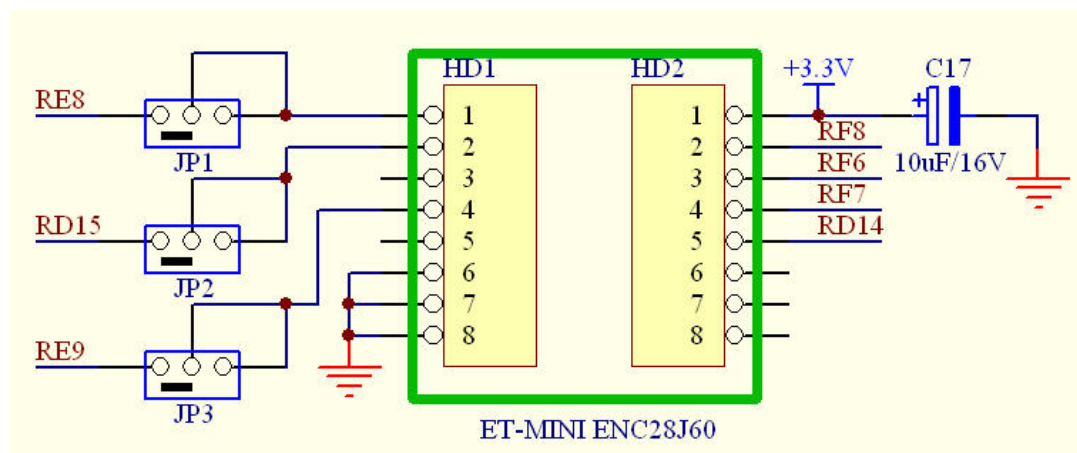
- **No.15** is Connector DC-JACK to supply power into board and supports external Voltage 9-12 VDC.
- **No.16** is Switch RESET.
- **No.17** is 80-Pin IC PIC24FJ128GA008 Microcontroller.
- **No.18** is EEPROM No.25LCxxx from Microchip that is SPI Interface.



SPI MEMORY

RG6 (SCK)	1	2	RG7(SO)
RG8 (SI)	3	4	RD12 (CS)
NC	5	6	NC
NC	7	8	NC
+3.3V	9	10	GND

- **No.19** is Connector to connect with Module Internet ET-MINI ENC28J60.

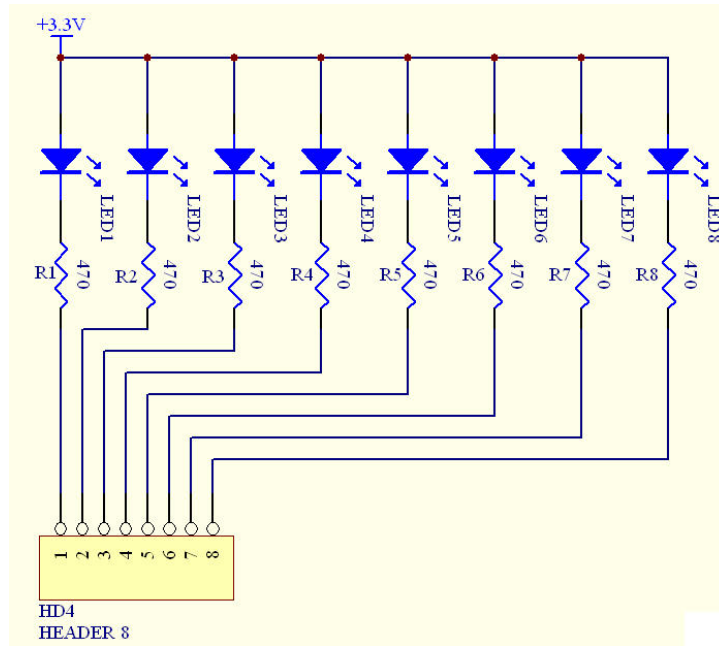


It is Jumper to select the connection for signal RE8, RD15 and RE9 of Microcontroller with signal Pin INT, RST and WOL of Module ET-MINI ENC28J60.

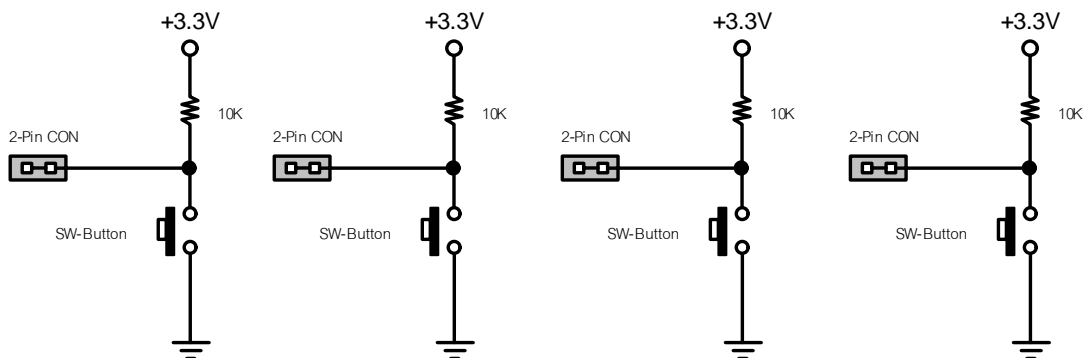
DIS= Disable: Not Connect

EN= Enable: Connect

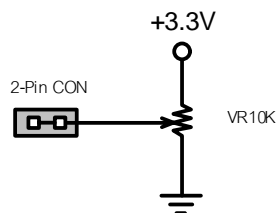
No.20 is Test I/O LED that consists of 8 LED and its circuit is shown below;



- **No.21** is device to test signal Input from 4 Switches that can create signal Logic 0 (0 Volt) and Logic 1 (+3.3 Volt) as shown in the circuit below;



- **No.22** is 4 device sets to test Analog Voltage and it can adjust Voltage level in the range of 0-3.3 Volt and its circuit is shown below;



- **No.23** is available space to interface further circuit.

Module ET-MINI ENC28J60

ET-MINI ENC28J60 is a module that is designed to be a medium of the connections between Microcontroller and Ethernet Network. It supports operation of Protocol TCP/IP by using IC ENC28J60 that is IC Ethernet Controller to support IEEE 802.3 standard communication and connect signal controller through SPI Bus with maximum high speed 10 Mb/s. Microchip supports Microchip TCP/IP Stack to develop program and user can download free without any charge from Microchip website (www.microchip.com). The briefly specifications of ENC28J60 is described below;

Specifications of IC ENC28J60

General:

- IEEE 802.3 compatible Ethernet Controller
- Integrated MAC and 10BASE-T PHY
- 8Kbyte Transmit/Receive Packet Dual Port Buffer SRAM
- Programmable Automatic Retransmit on Collision
- Programmable Padding and CRC Generation
- Programmable Automatic Rejection of Erroneous Packets
- SPI™ Interface with speeds up to 10 Mb/s
- Supports Full and Half-Duplex modes

Buffer:

- Configurable transmit/receive buffer size
- Hardware managed circular receive FIFO
- Byte-wide random and sequential access
- Internal DMA for fast memory copying
- Hardware assisted IP checksum calculation

PHY:

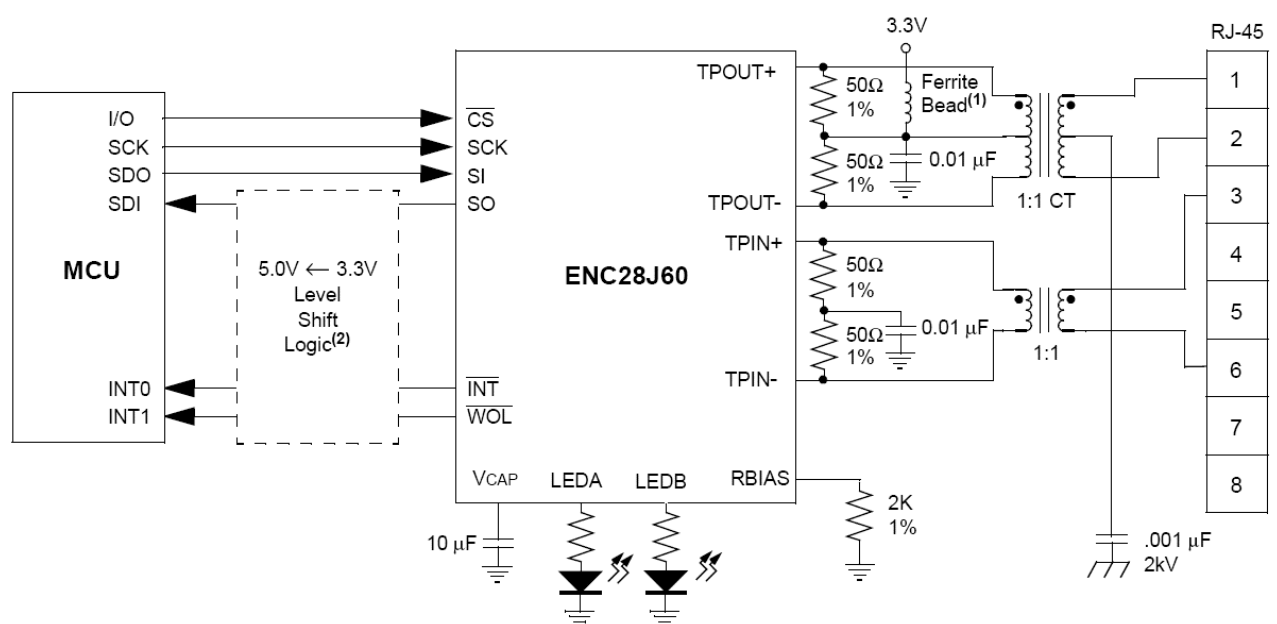
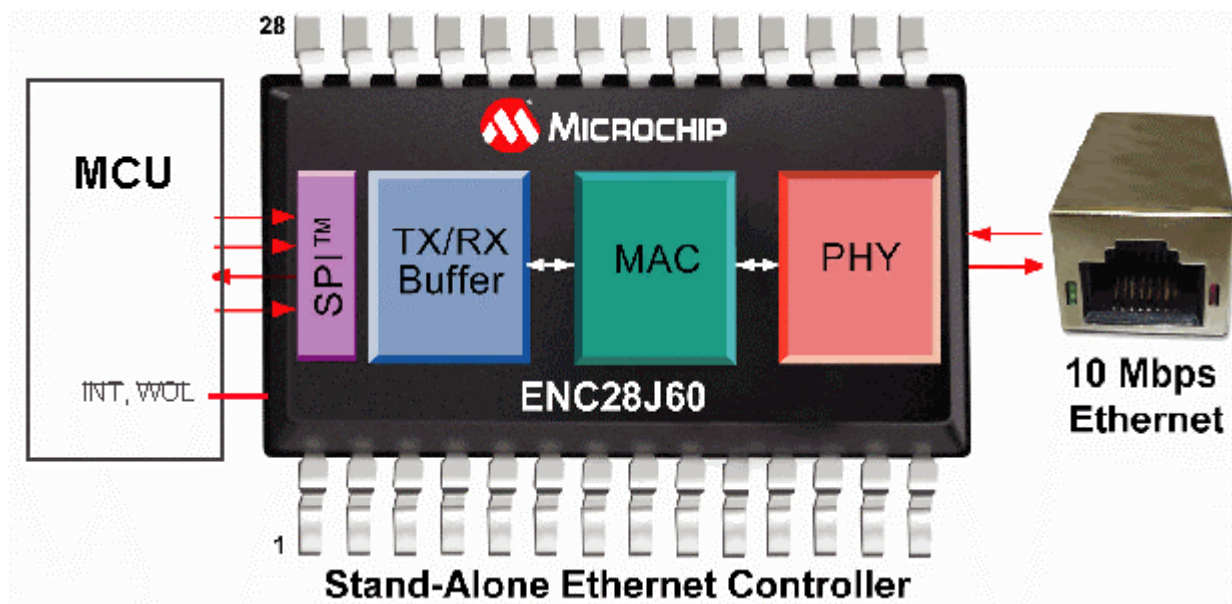
- Wave shaping output filter
- Loopback mode

MAC:

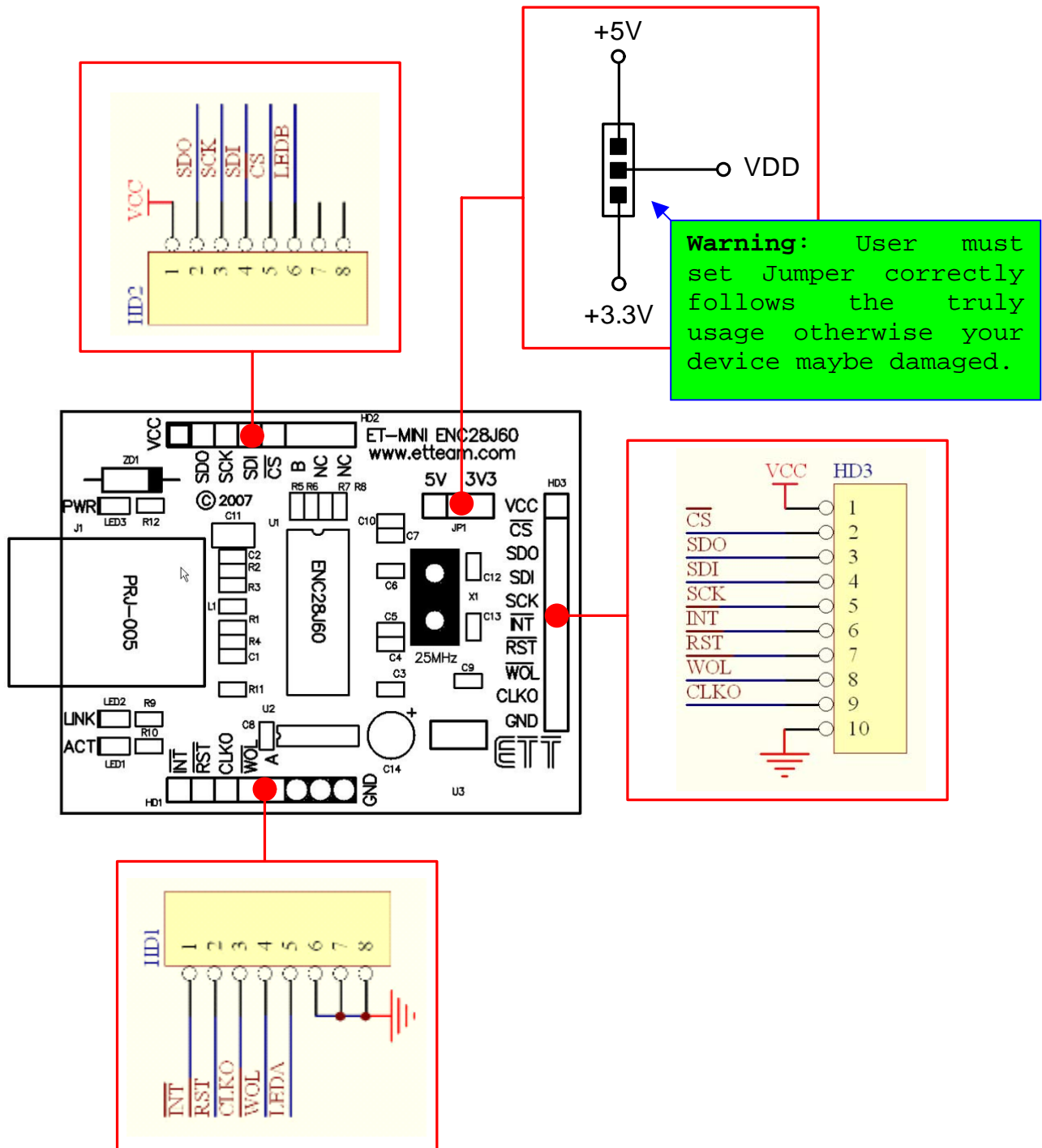
- Support for Unicast, Multicast and Broadcast packets
- Programmable pattern matching of up to 64 bytes within packet at user defined offset

- Programmable wake-up on multiple packet formats, including Magic Packet®, Unicast, Multicast, Broadcast, specific packet match or any packet

The method to connect device with Microcontroller is quite easy because it is SPI Bus Interface that uses fewer pins. For power system of ENC28J60 is 3 Volt IC if user wants to run with 5 Volt Microcontroller, user must Buffer Circuit that is designed by ETT. It supports the connection of power system between 3 Volt and 5 Volt internal Board ET-MINI ENC28LJ60; in this case, user can select power system by setting Jumper 5V/3V3.



Picture displays block diagram of the connection signals between ENC28J60 and Microcontroller.



Picture displays structure of Board ET-MINI ENC28J60.

From picture above, HD1 and HD2 is designed to connect with Board ET-PIC 24 WEB; for HD3 is designed to connect with other Microcontrollers and it is designed in the feature of ET-MINI from ETT.

Table shows Name and Function of Signal Pins of ENC28J60

Pin Name	Pin Type	Function
$\overline{\text{CS}}$	INPUT	Signal Enable/Disable the SPI Bus Interface of ENC28J60 CS = 0 is Enable the SPI Bus Interface of ENC28J60 CS = 1 is Disable the SPI Bus Interface of ENC28J60
SDO	OUTPUT	Signal Serial Data Output
SDI	INPUT	Signal Serial Data Input
SCK	INPUT	Signal Serial Clock
$\overline{\text{INT}}$	OUTPUT	Signal Interrupt Active Logic 0
$\overline{\text{RST}}$	INPUT	Signal RESET Active Logic 0
$\overline{\text{WOL}}$	OUTPUT	Signal Wake-up on LAN interrupt Active Logic 0
CLKO	OUTPUT	Signal Programmable clock output
LEDA	OUTPUT	Display status of Signal LINK
LEDB	OUTPUT	Display status of Signal ACT

Table shows the signal connection between Board ENC28J60 and ET-PIC24 WEB

ENC28J60	ET-PIC24WEB (PIC24FJ128GA008)
$\overline{\text{CS}}$	RD14
SDO	RF8
SDI	RF7
SCK	RF6
$\overline{\text{INT}}$	RE8 (Set Jumper)
$\overline{\text{RST}}$	RD15 (Set Jumper)
$\overline{\text{WOL}}$	RE9 (Set Jumper)
CLKO	-
LEDA	-
LEDB	-

Summarize Allocation and Use Resources of Board ET-PIC24 WEB V1.0

Normally, MCU NO.PIC24FJ128GA008 that is used with Board ET-PIC24 WEB V1.0 has independent signal pin up to 69 I/O but some signal pin is designed and connected with particular I/O device, so user can use these signal pins to be general I/O and we can summarize these signal pins as below;

MCU Oscillator

- **RC12** is used to be **OSC1** and connects with Crystal 8.00MHz to be Signal Clock of MCU.
- **RC15** is used to be **OSC2** and connects with Crystal 8.00MHz to be Signal Clock of MCU.

RTC Oscillator

- **RC13** is used to be **OSC1** and connects with Crystal 32.768KHz to be Signal Clock of RTC.
- **RC14** is used to be **OSC2** and connects with Crystal 32.768KHz to be Signal Clock of RTC.

RS232-CH1 Serial Port Communication (UART)

- **RF2** is used to be Pin **RXD** to receive data from RS232 Channel 1.
- **RF3** is used to be Pin **TXD** to transmit data to RS232 Channel 1.

RS232-CH2 Serial Port Communication (UART)

- **RF4** is used to be Pin **RXD** to receive data from RS232 Channel 2.
- **RF5** is used to be Pin **TXD** to transmit data to RS232 Channel 2.

Ethernet Module (SPI-1)

- **RF6** is used to be **SCK** to connect with Ethernet Module (ET-MINI ENC28J60).
- **RF7** is used to be **SDI** to connect with Ethernet Module (ET-MINI ENC28J60).
- **RF8** is used to be **SDO** to connect with Ethernet Module (ET-MINI ENC28J60).

- **RD14** is used to be **CS** to connect with Ethernet Module (ET-MINI ENC28J60).
- **RE8** is used to be **INT1** to connect with Ethernet Module (ET-MINI ENC28J60). In this case, user can select to use or not use as desired by setting **Jumper INT (EN/DS)**; normally, Jumper is set to be **DS** (Disable: Not use).
- **RE9** is used to be **WOL** to connect with Ethernet Module (ET-MINI ENC28J60). In this case, user can select to use or not use as desired by setting **Jumper WOL (EN/DS)**; normally, Jumper is set to be **DS** (Disable = Not use).
- **RD15** is used to be **RST** to connect with Ethernet Module (ET-MINI ENC28J60). In this case, user can select to use or not use as desired by setting **Jumper RST (EN/DS)**; normally, Jumper is set to be **DS** (Disable = Not use).

SPI Memory Module (SPI-2)

- **RG6** is used to be **SCK** to connect with SPI Memory.
- **RG7** is used to be **SDI** to connect with SPI Memory.
- **RG8** is used to be **SDO** to connect with SPI Memory.
- **RD12** is used to be **CS** to connect with SPI Memory.

Character LCD Display

- **RE0** is used to be **LCD D4** to connect with 4 Bit Character LCD.
- **RE1** is used to be **LCD D5** to connect with 4 Bit Character LCD.
- **RE2** is used to be **LCD D6** to connect with 4 Bit Character LCD.
- **RE3** is used to be **LCD D7** to connect with 4 Bit Character LCD.
- **RE4** is used to be **LCD RS** to connect with 4 Bit Character LCD.
- **RE5** is used to be **LCD RW** to connect with 4 Bit Character LCD.
- **RE6** is used to be **LCD EN** to connect with 4 Bit Character LCD.

LED[1..8] to test Web Server Control

- **RD0..RD7** is used to be **Digital Output** and connects with **LED [1..8]**.

SW[1..4] to test Web Server Control

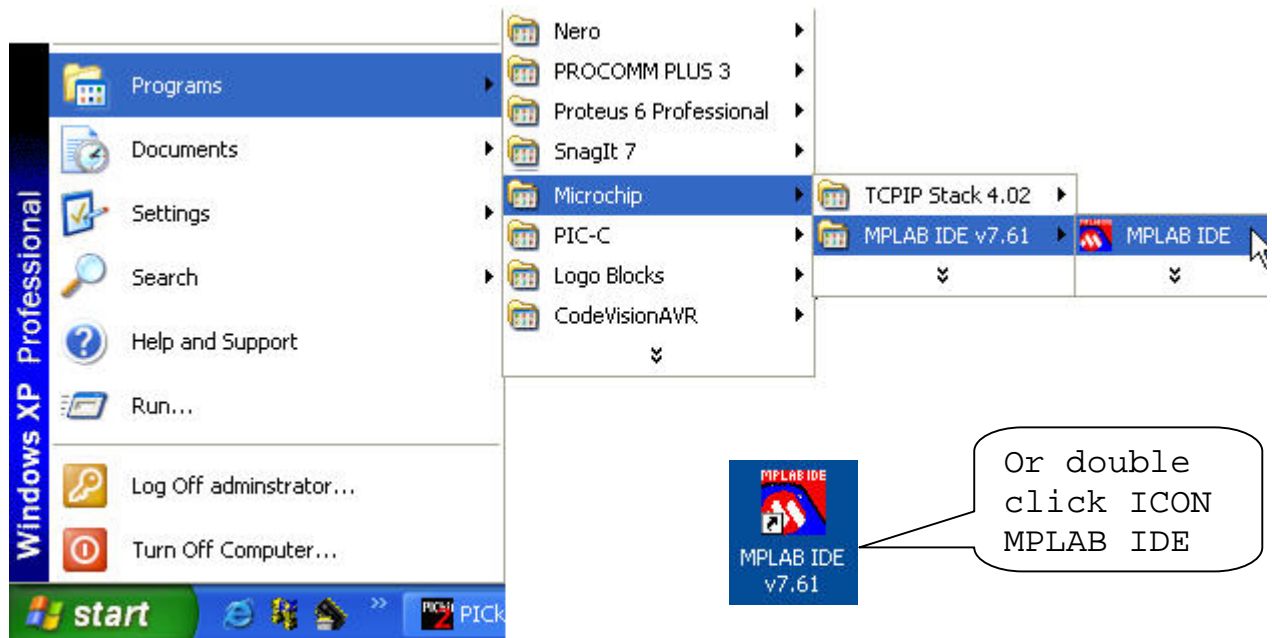
- RD8..RD11 is used to be **Digital Input** and connects with Switch[1..4].

ADC Input to test Web Server Control

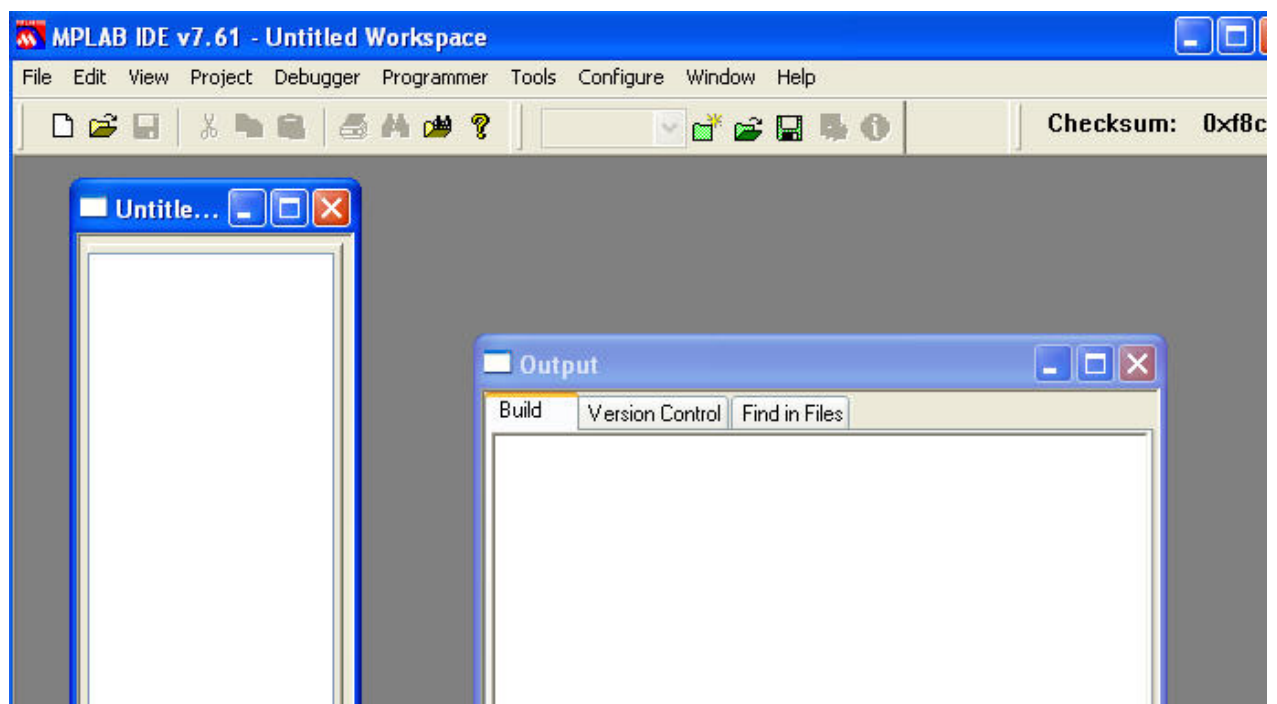
- RB5 (AN5) is used to be **ADC0** to read Input value of 10 Bit Analog to Digital and connects with VR1.

Initial Recommendation of MPLAB C30

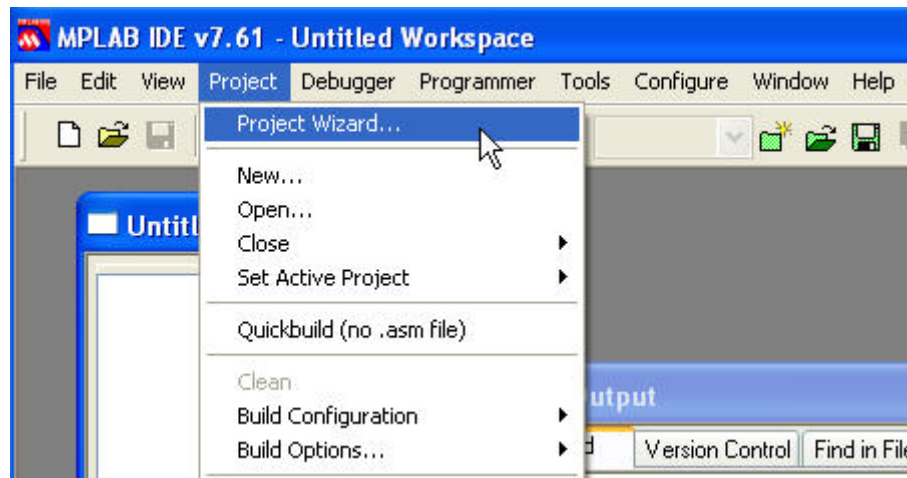
Open **Program MPLAB IDE**, click **Start -> Microchip -> MPLAB IDE v7.61 -> MPLAB IDE** or double click **Icon MPLAB IDE** on Desktop as shown below.



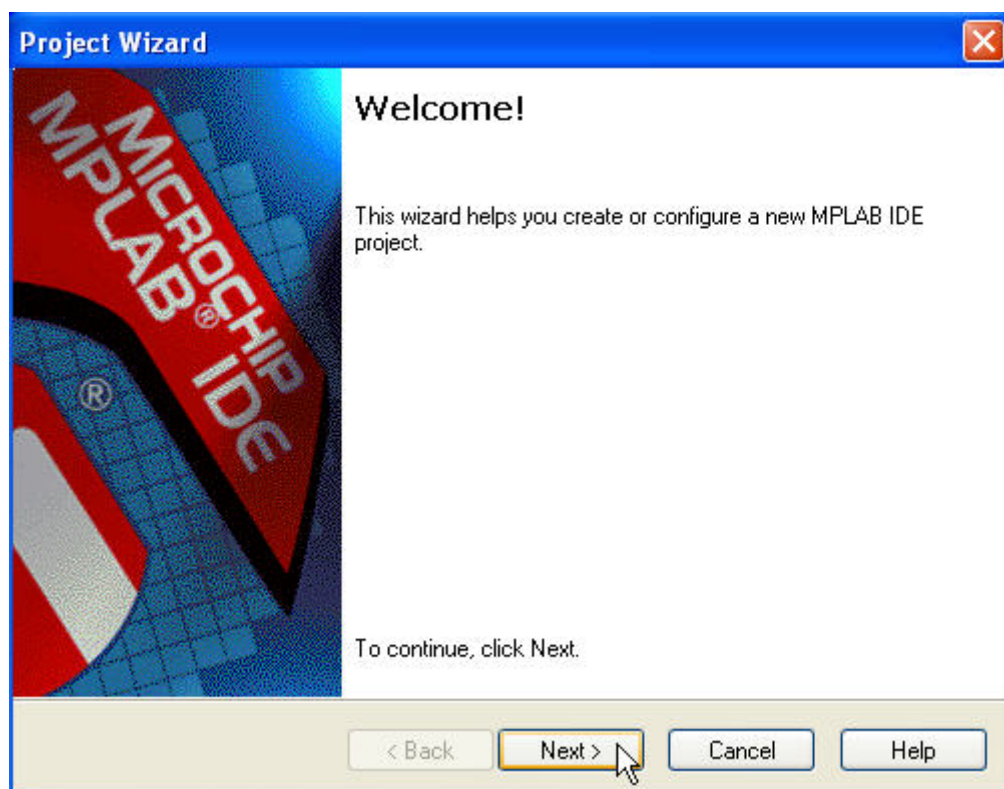
2. It will display window of Program MPLAB IDE v7.61 as shown below.



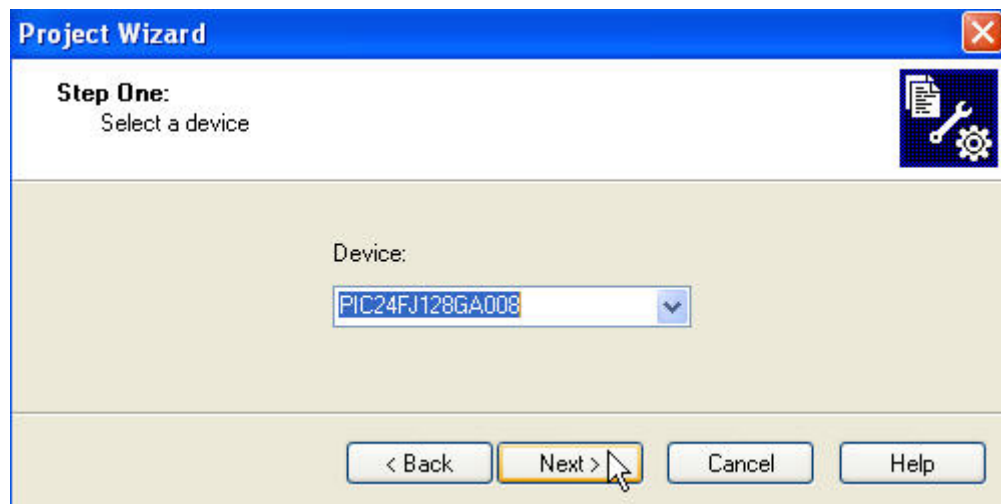
- Create Project, click **Project -> Project Wizard..**



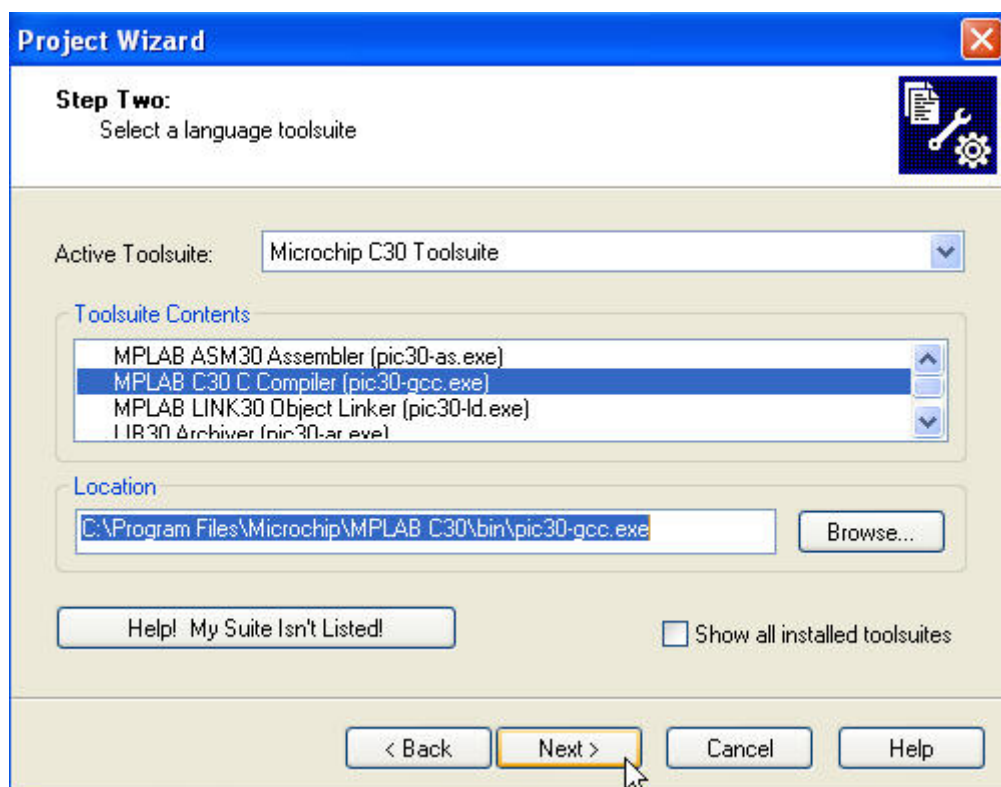
It will display Window Project Wizard; click **Next >** to do the next step.



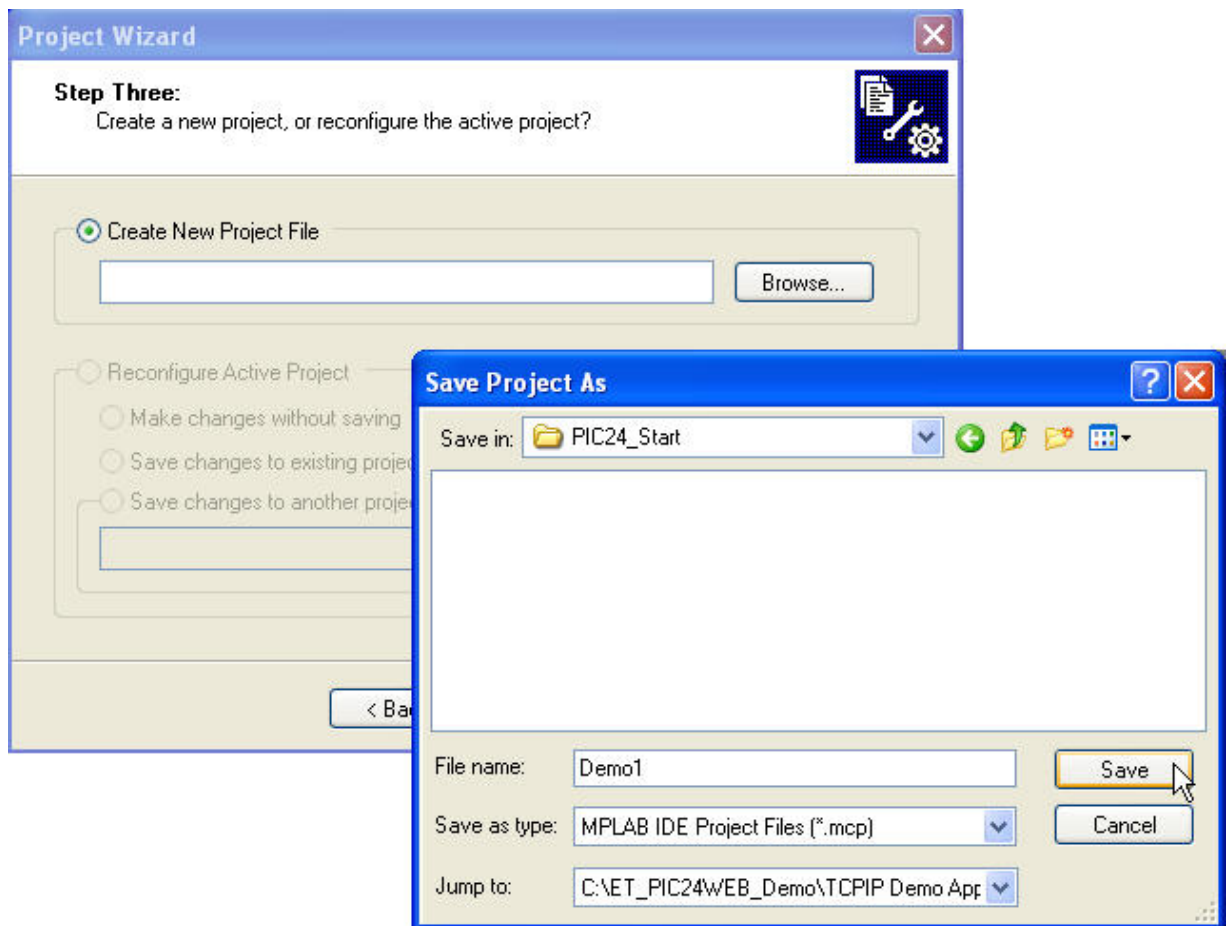
- Select device in the blank of **Device:** and it must be corresponding with Board. In this case, it is **PIC24FJ128GA008** and then click **Next >**.



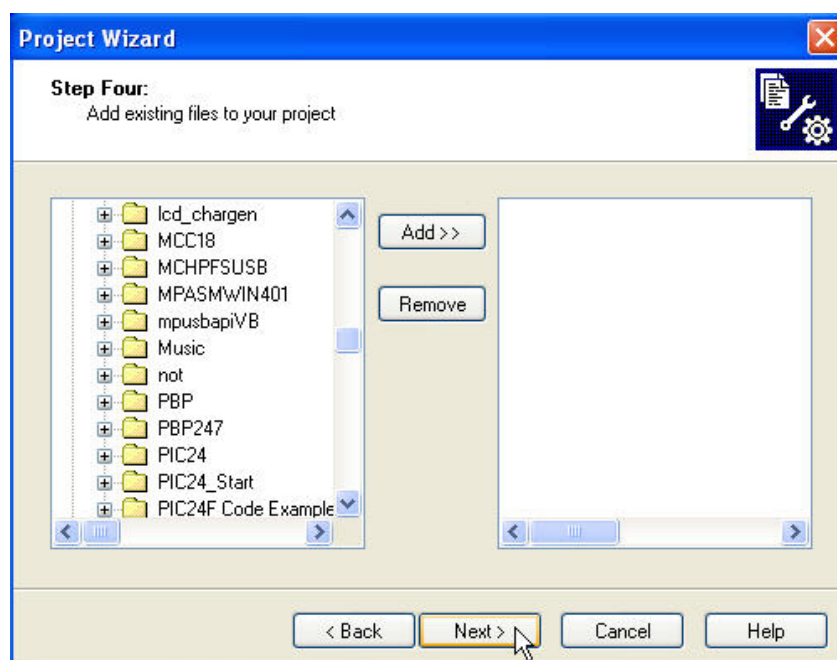
- Select the used Compiler; in this case, it is MPLAB C30 C Compiler (pic30-gcc.exe) and then click **Next >**.



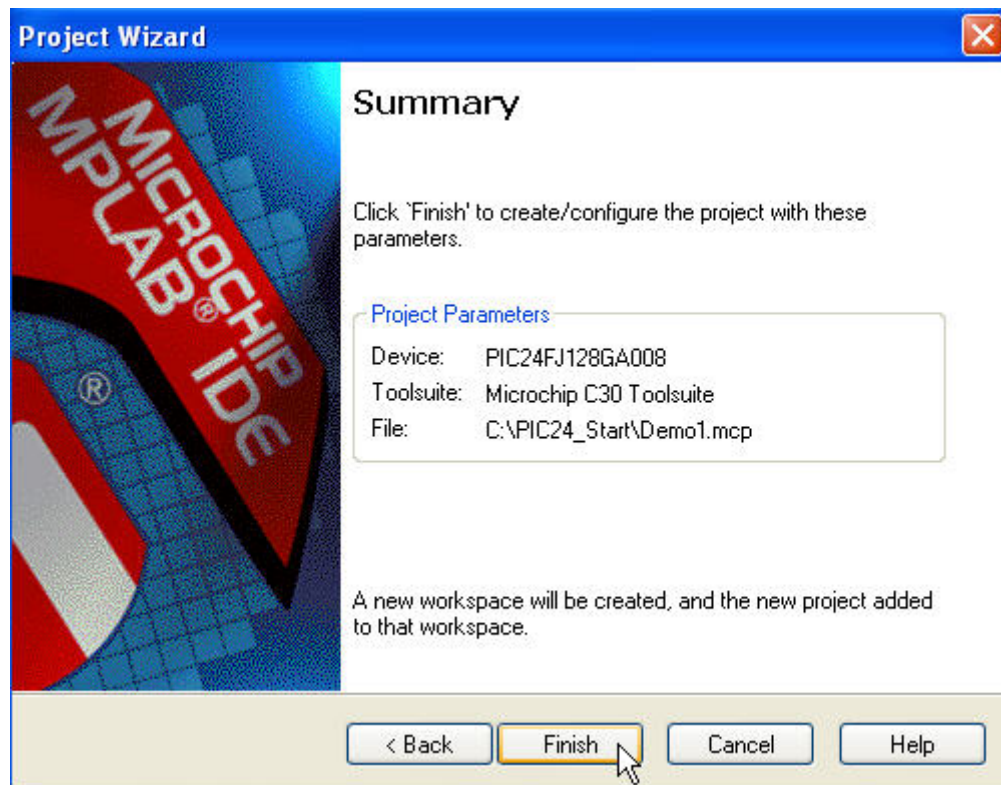
- Click **Browse** to the desired Folder to store Project; for example, create Folder name PIC24_Start follows the example and then Browse.. the Folder, specify Project name; in this example, it is Demo1 and then **Save**.



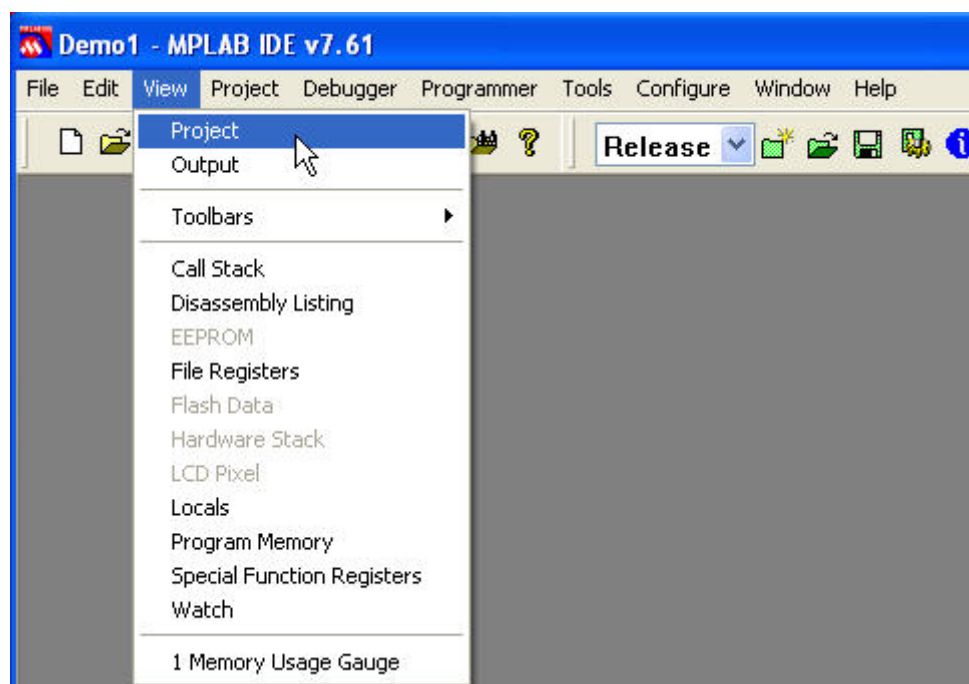
- It will display window; in this case, user can **Add** or **Remove** File Codes in project if there is File Code in project, but if there is no any File Code, click **Next >** to go to the next step.



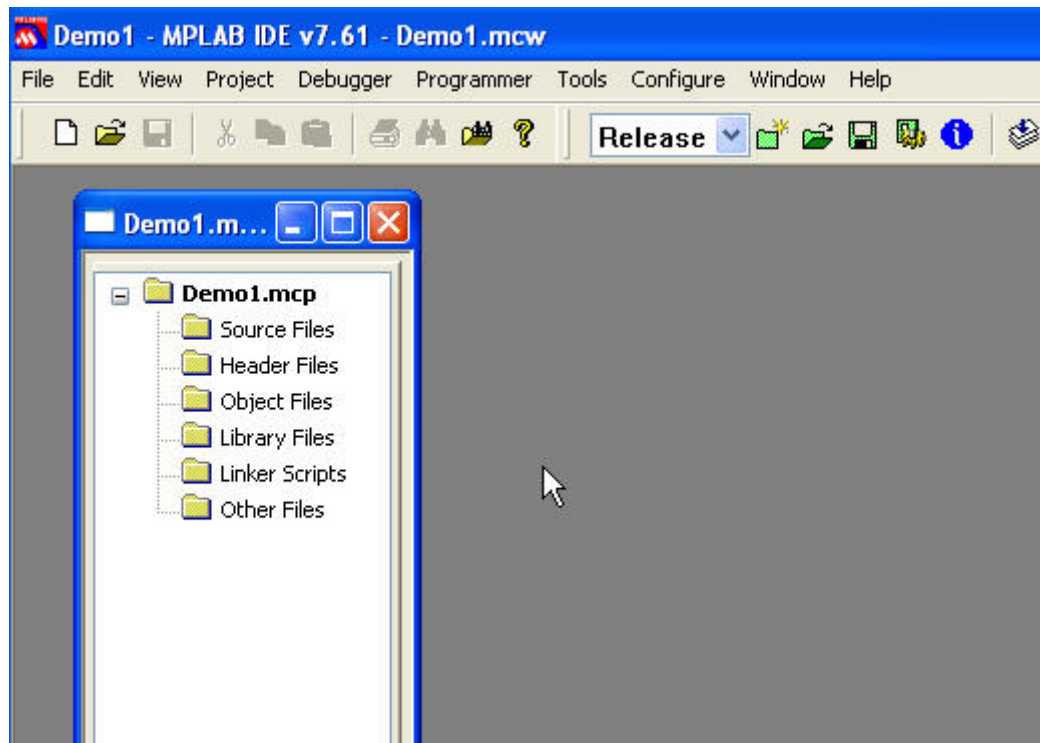
- It will display window that summarizes Parameter values of the created Project, verify values correctly, and then click **Finish** to end the step of Project Wizard.



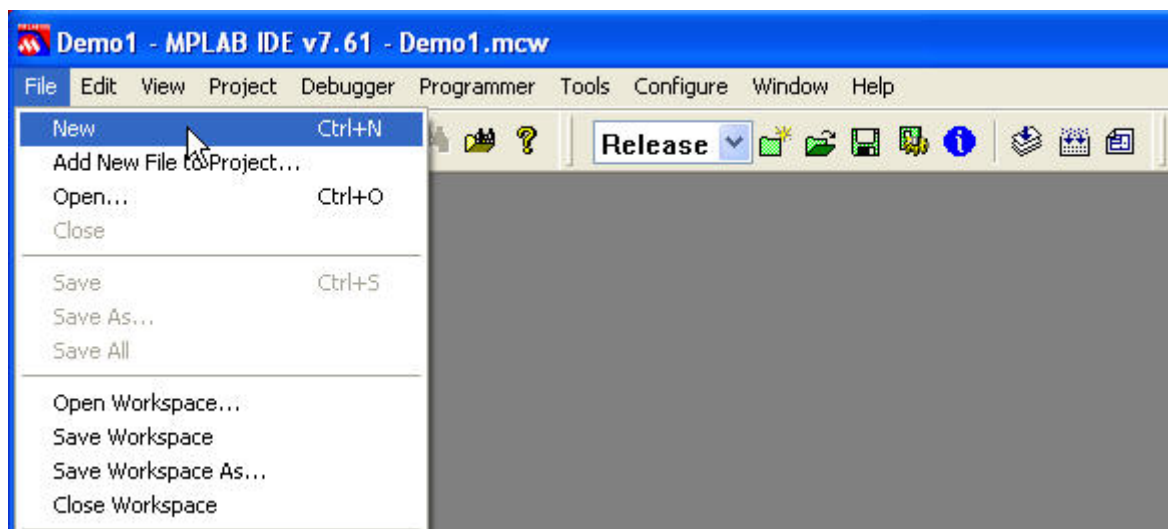
- It enters the main window of MPLAB, click **View -> Project**



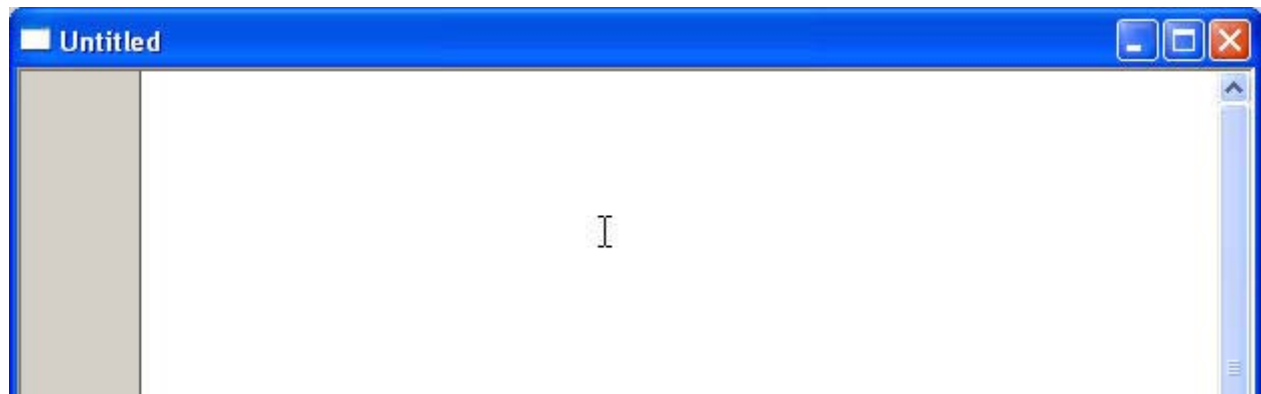
- It will display window that shows structure of Project as shown below.



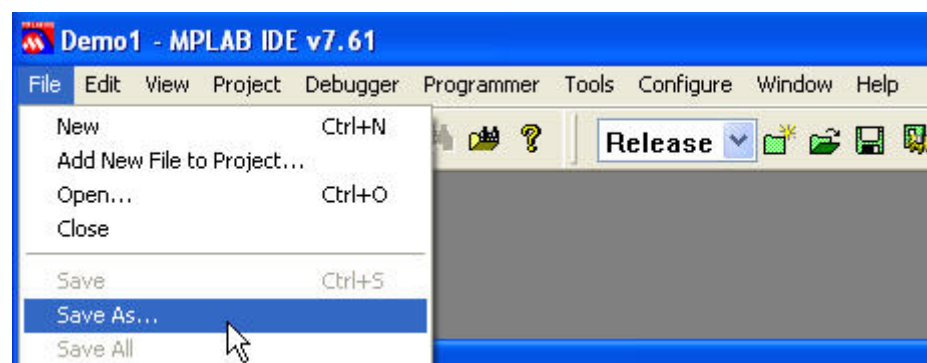
- Click **File** -> **New** to create File Source Code.



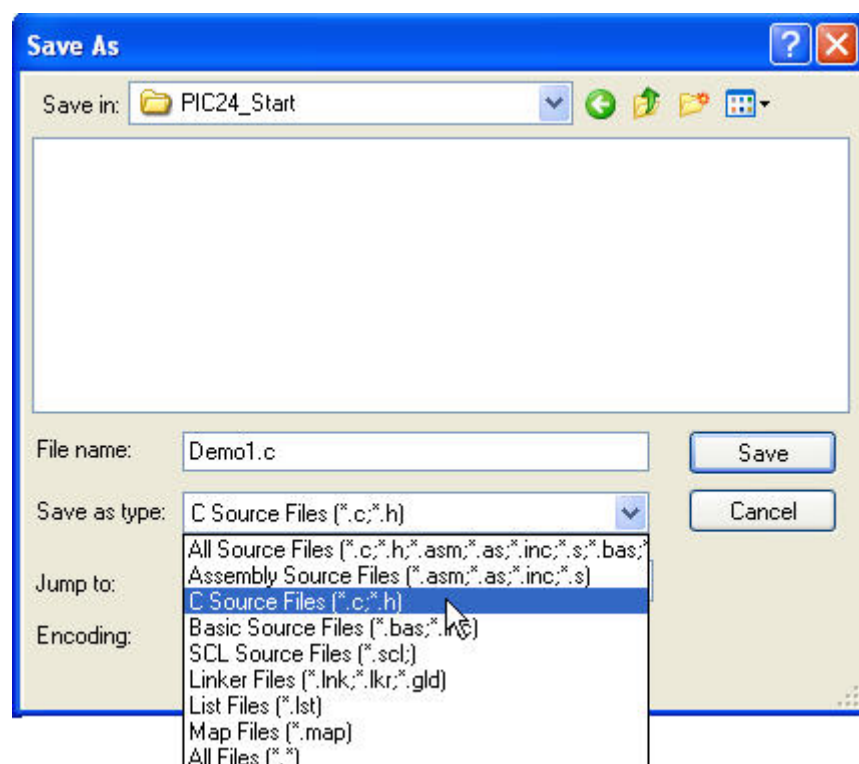
- It will display the blank screen window that is Untitled as shown below.



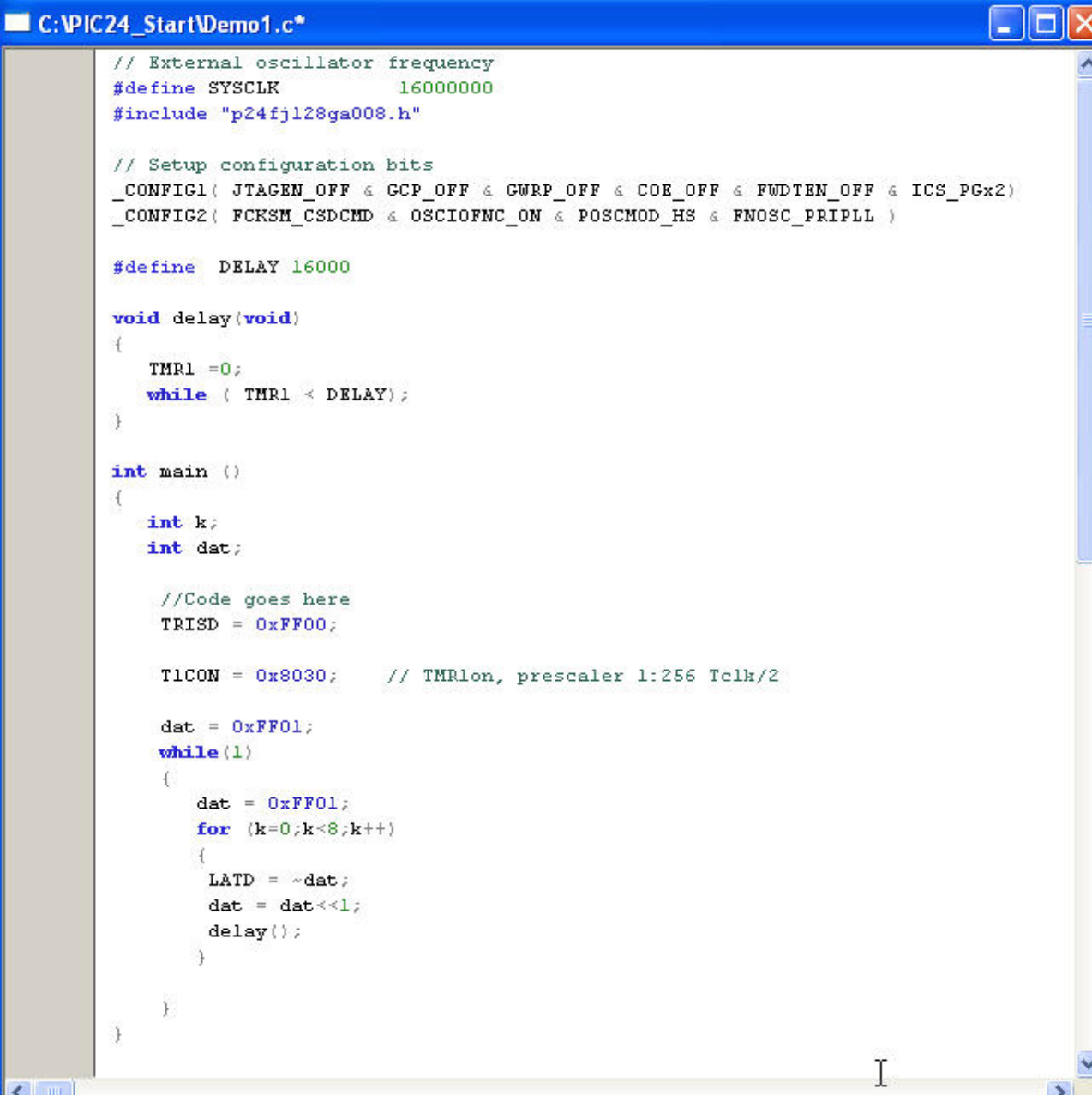
- Select **File -> Save As...** to save as File. C



- Specify filename and follow by surname .C; in this example, it is Demo1.C and then selects **Save as type:** to be C Source Files (*.c, *.h) as shown below.



- File Demo1.c will be appeared and then type code program into the file as shown in the example below.



```
// External oscillator frequency
#define SYSCLK      16000000
#include "p24fjl28ga008.h"

// Setup configuration bits
_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDIEN_OFF & ICS_PGx2)
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_ON & POSCMOD_HS & FNOSC_PRIPLL )

#define DELAY 16000

void delay(void)
{
    TMR1 = 0;
    while ( TMR1 < DELAY);
}

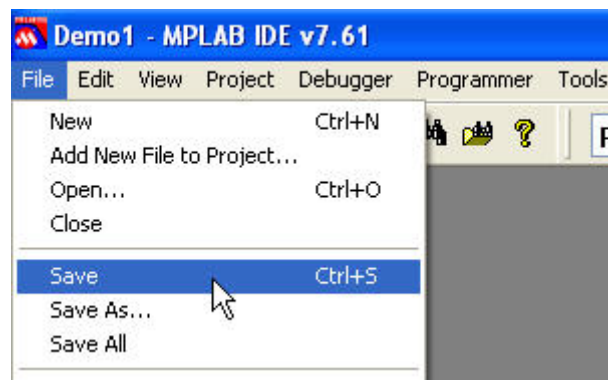
int main ()
{
    int k;
    int dat;

    //Code goes here
    TRISD = 0xFF00;

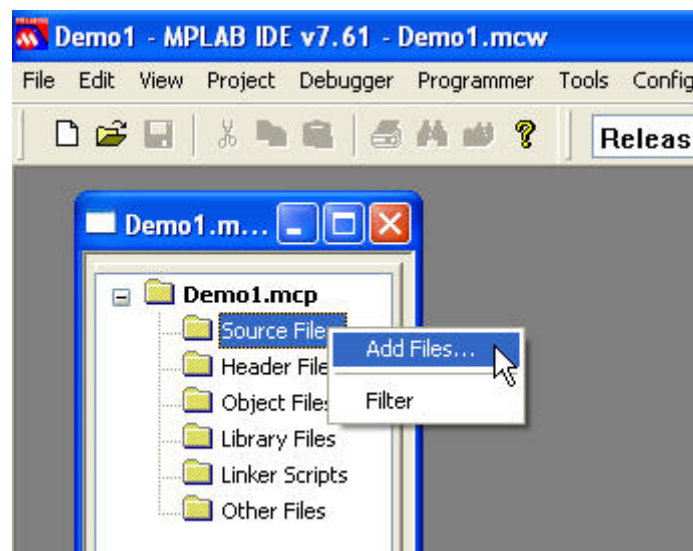
    T1CON = 0x8030;    // TMR1on, prescaler 1:256 Tclk/2

    dat = 0xFF01;
    while(1)
    {
        dat = 0xFF01;
        for (k=0;k<8;k++)
        {
            LATD = ~dat;
            dat = dat<<1;
            delay();
        }
    }
}
```

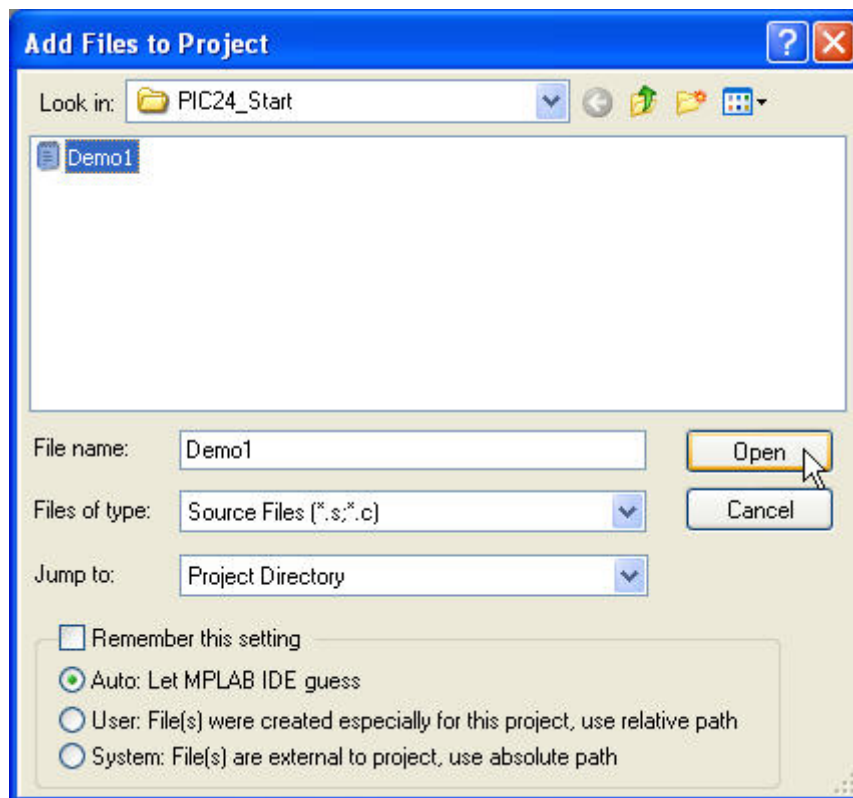
- Save file, click **File -> Save**



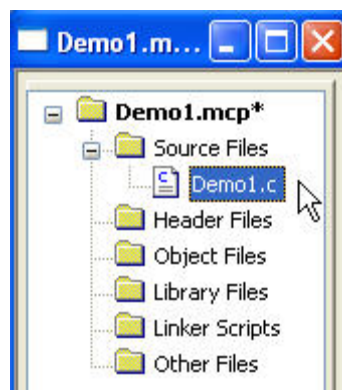
- Add the created File Source Code into Project, click right on **Source File** and then select **Add Files..**



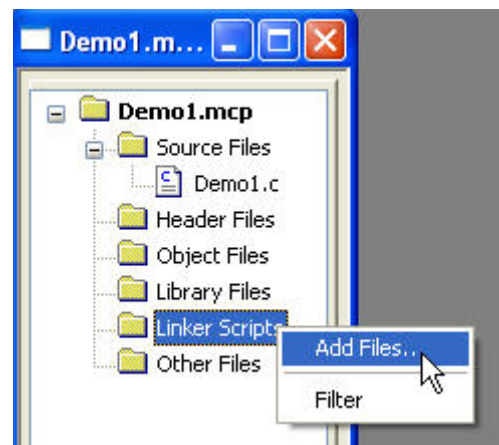
- It will display window **Add Files to Project**, click left on **File Source Code**; in this example, it is Demo1 and then click **Open**.



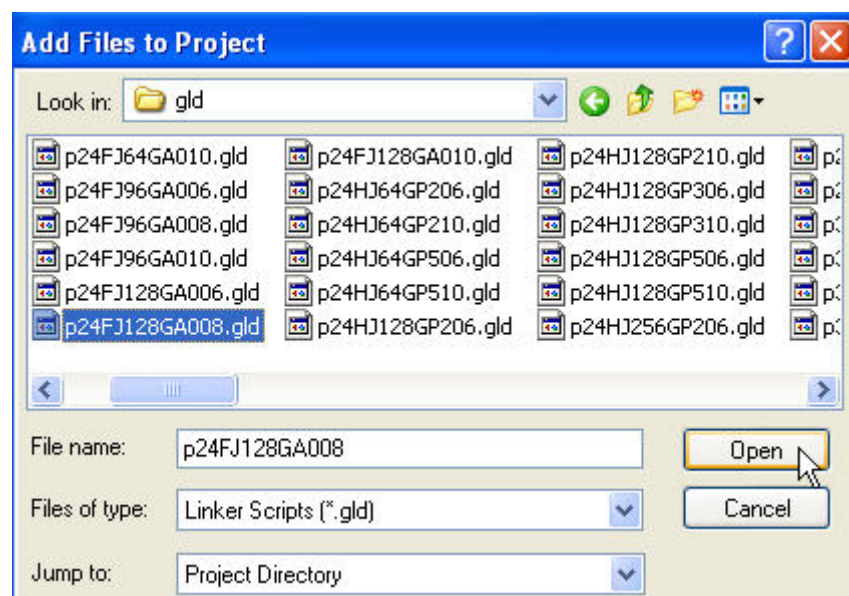
- Filename Demo1.c will appear in Source Files.



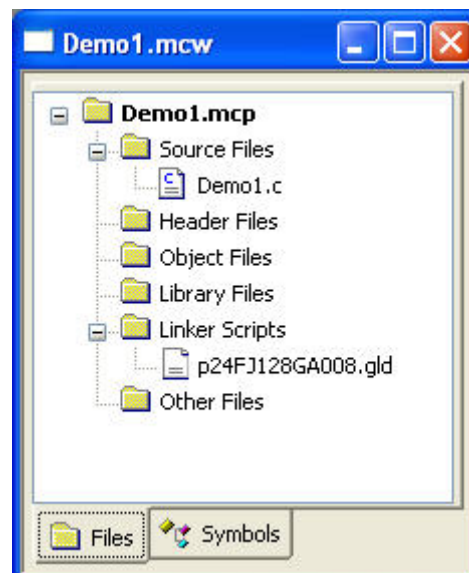
- Add **File Linker Scripts**, click right on **Linker Scripts** and then select **Add Files..**



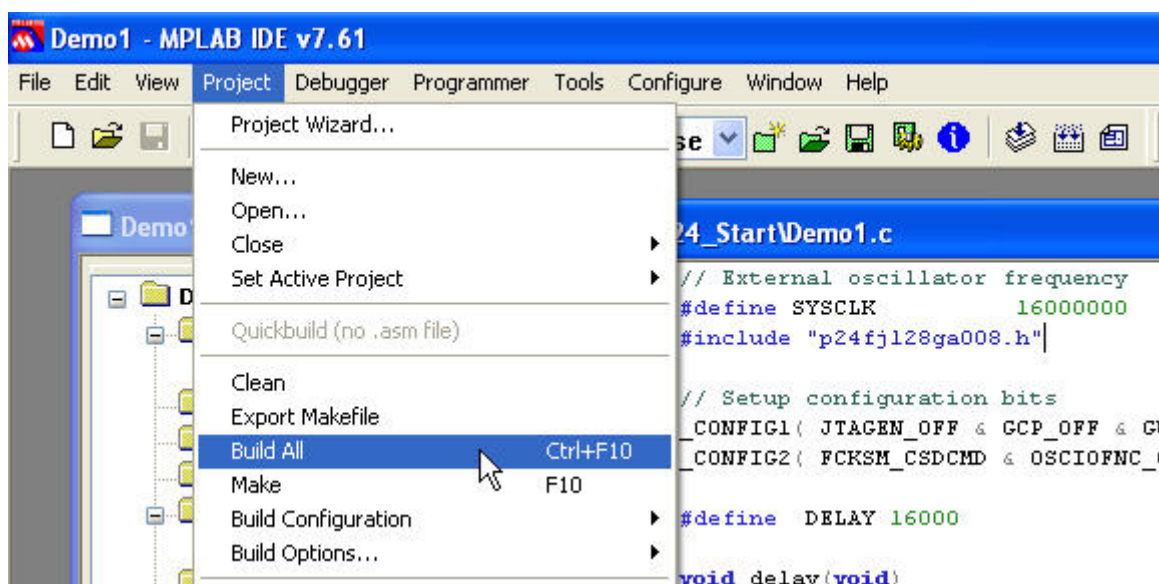
- It will display window **Add Files to Projects**, click file **p24FJ128GA008.gld** that is in the following folder;
C:\Program Files\Microchip\MPLAB C30\support\gld



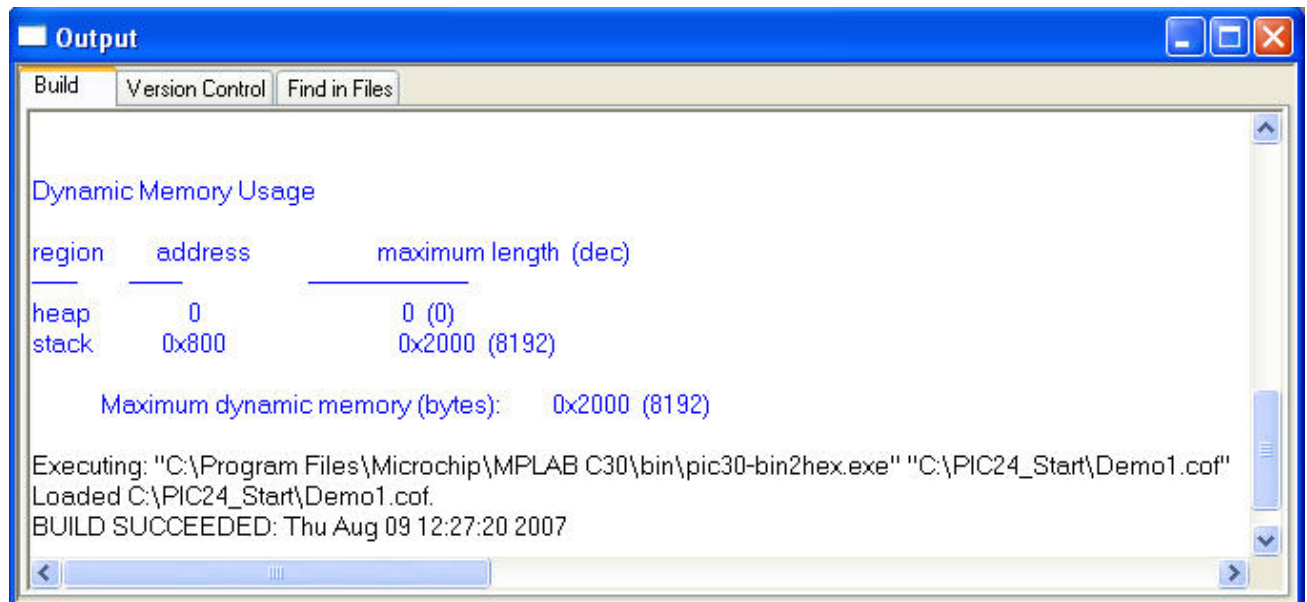
- When everything is complete, user can see file **p24FJ128GA008.gld** that has already added into Linker Scripts completely as shown below;



- Next, compile project; click **Project -> Build All** as shown below.



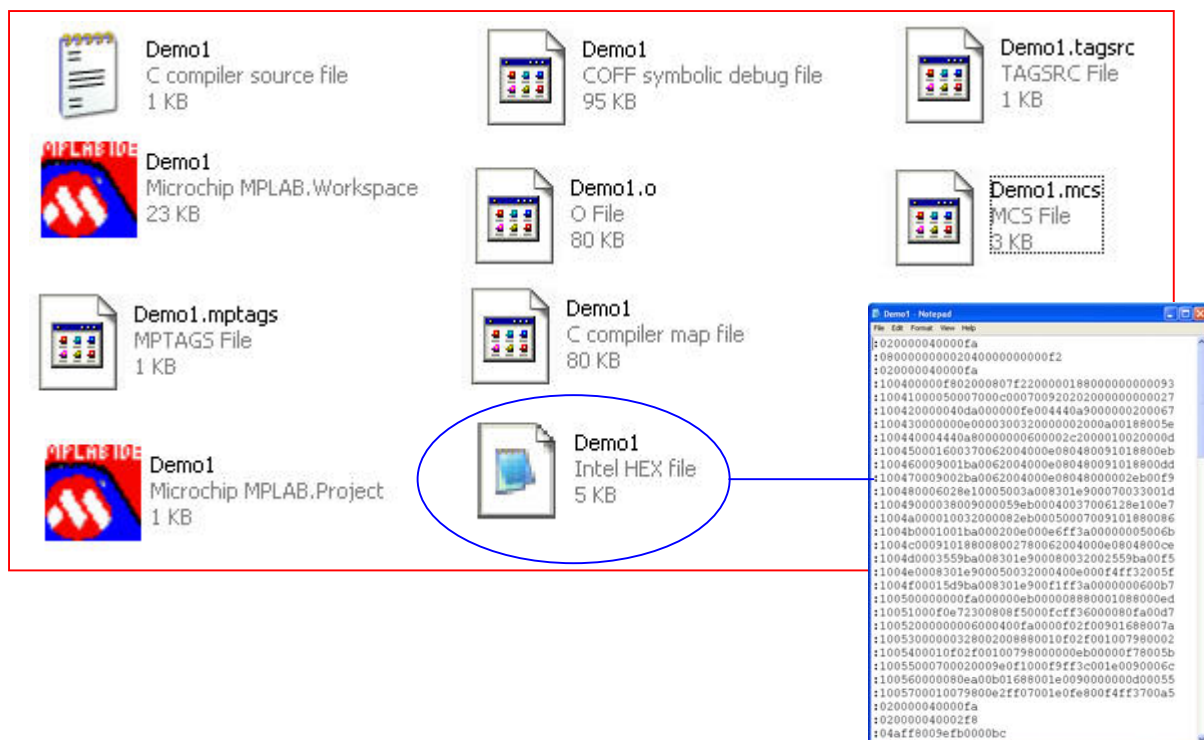
- It will report result of compiling in the window Output -> Build; if compiling is complete, it will display message **BUILD SUCCEEDED** as shown below.



- It will display File Demo1.Hex in the same folder as the created file as shown below;

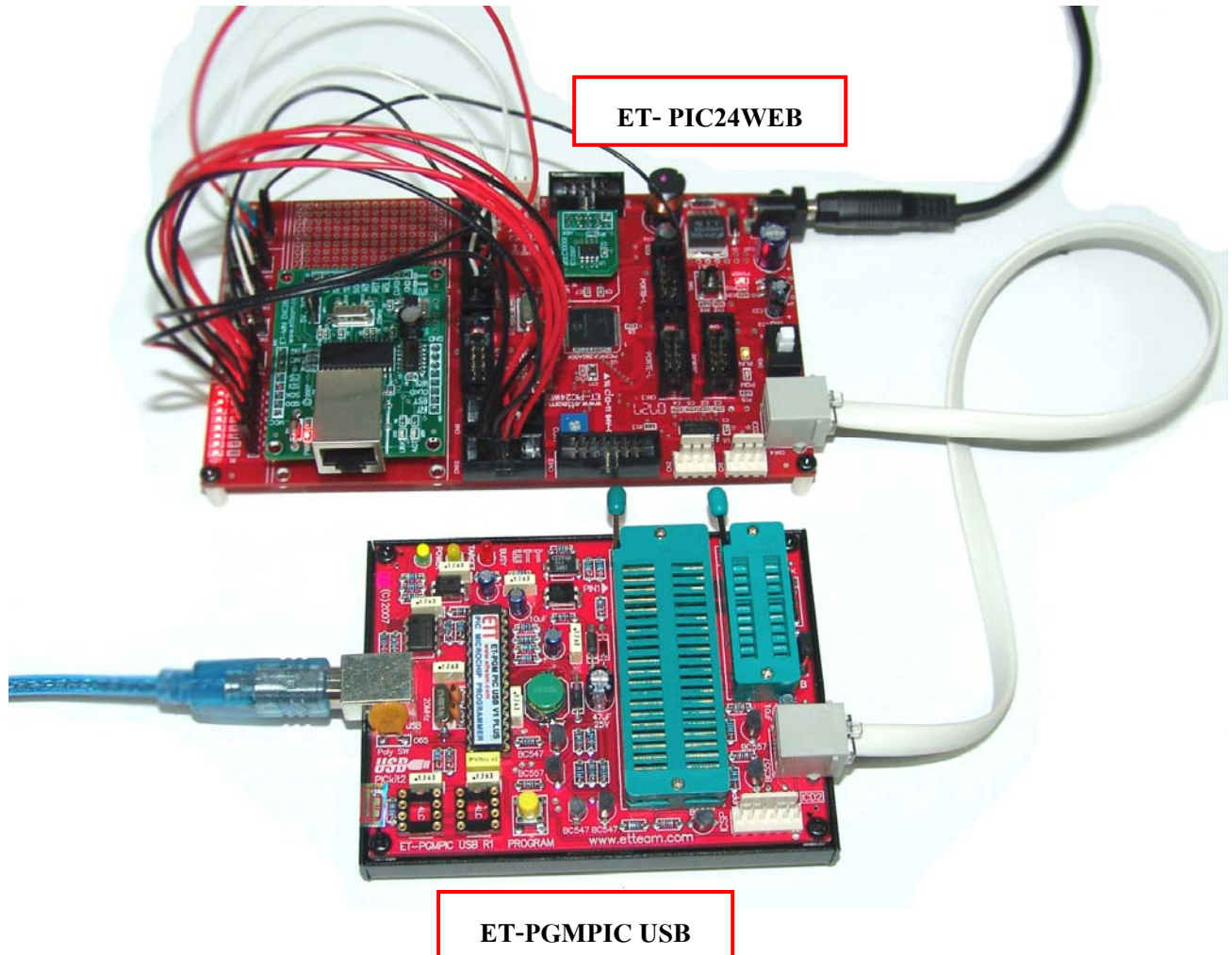


PIC24_Start

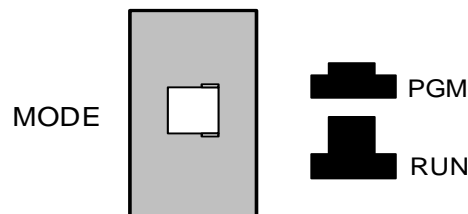


■ Code Programming

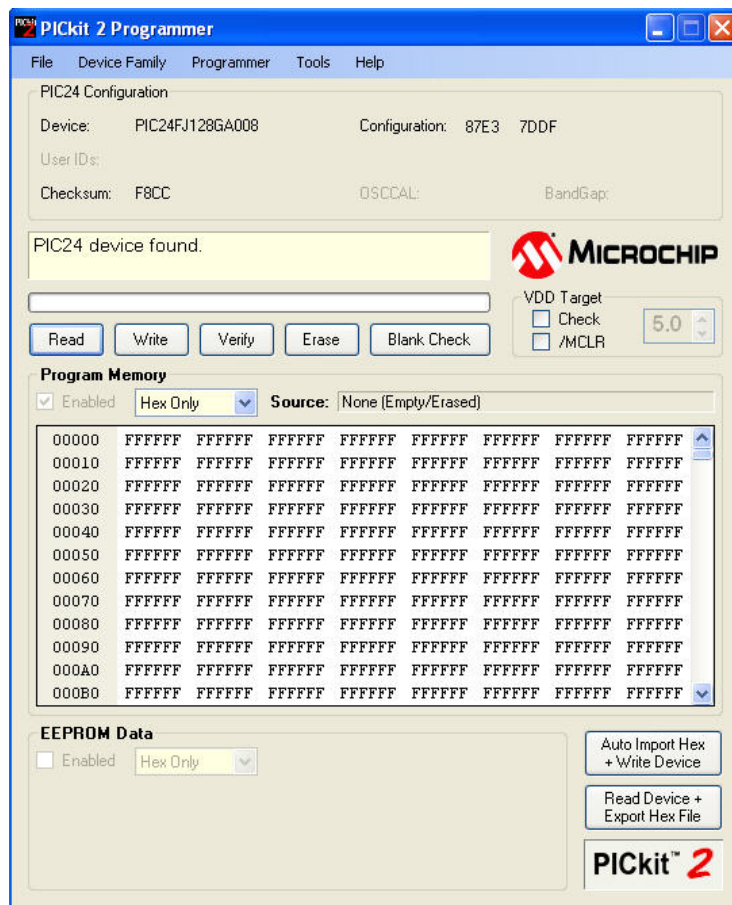
After user gets Hex File; next, it is programming code into Microcontroller of Board ET-PIC 24 WEB; in this case, it must use external programmer such as ICD2, PICKit2 or ETT Programmer version ET-PGM USB. User can interface Cable for programming at Connector ICD2 as shown in the picture below.



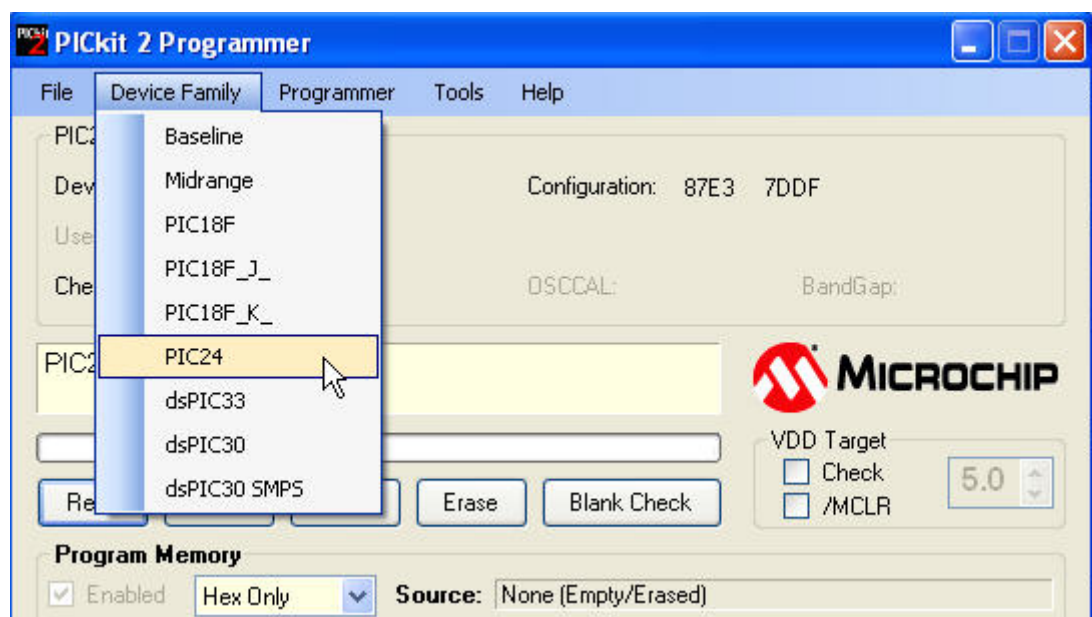
- Press **Switch MODE** to **PGM** position.



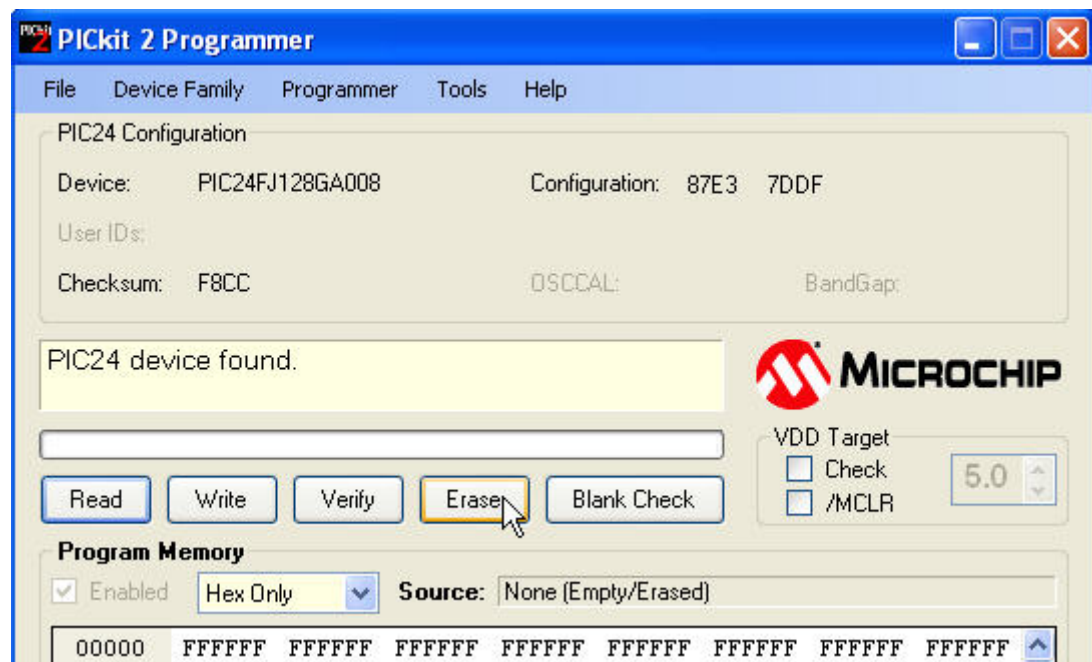
- Open **Program PICKit2 Programmer**.



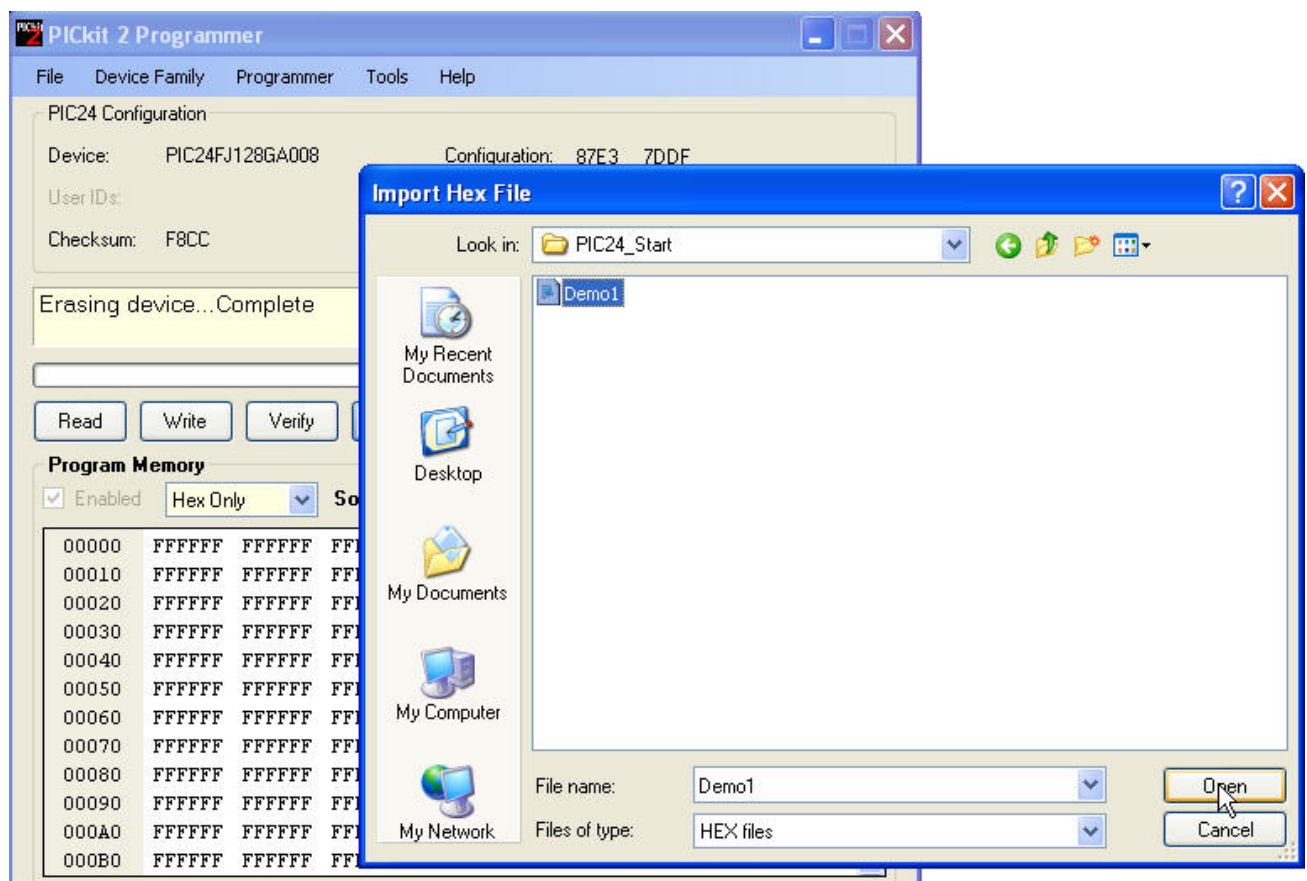
- Select **Device Family** -> **PIC24**



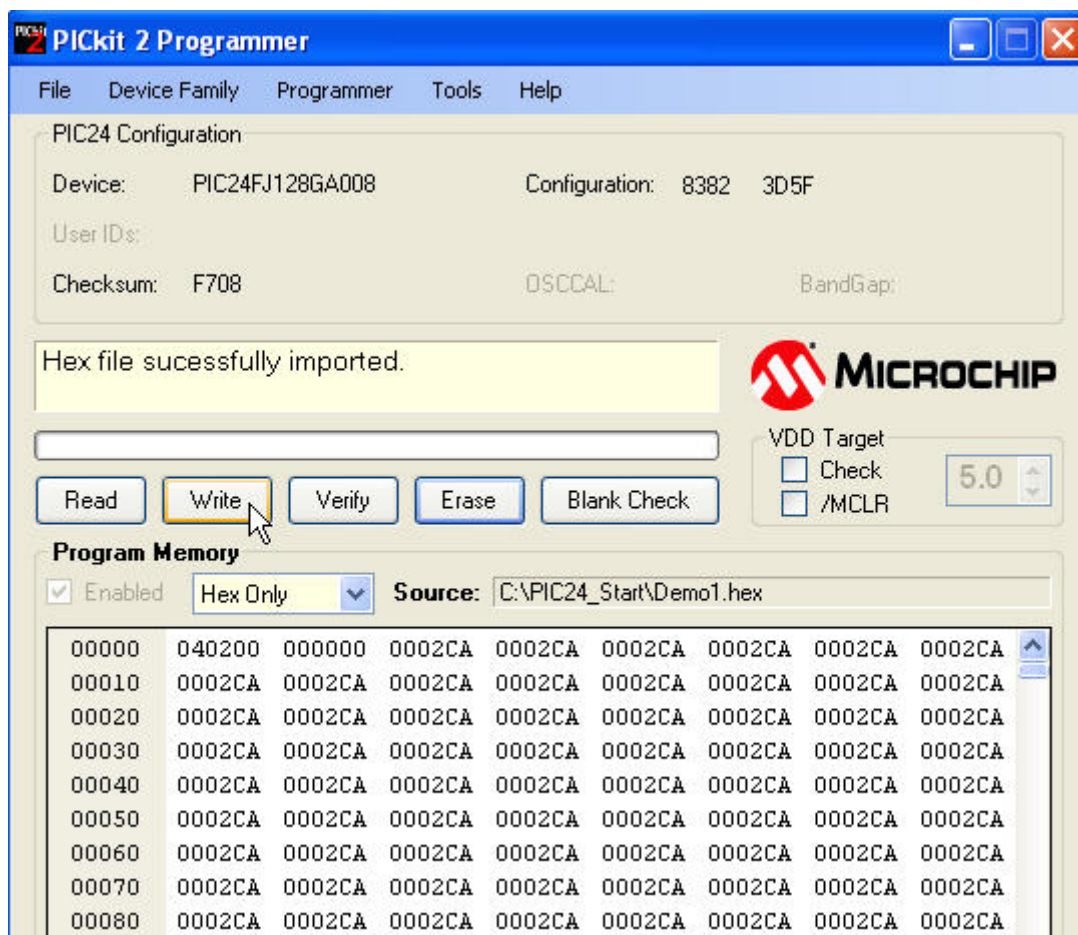
- If the connection is complete, it can find Device number to be PIC24FJ128GA008 as shown below. In the first step, user must select **Erase** to erase the old program Memory.



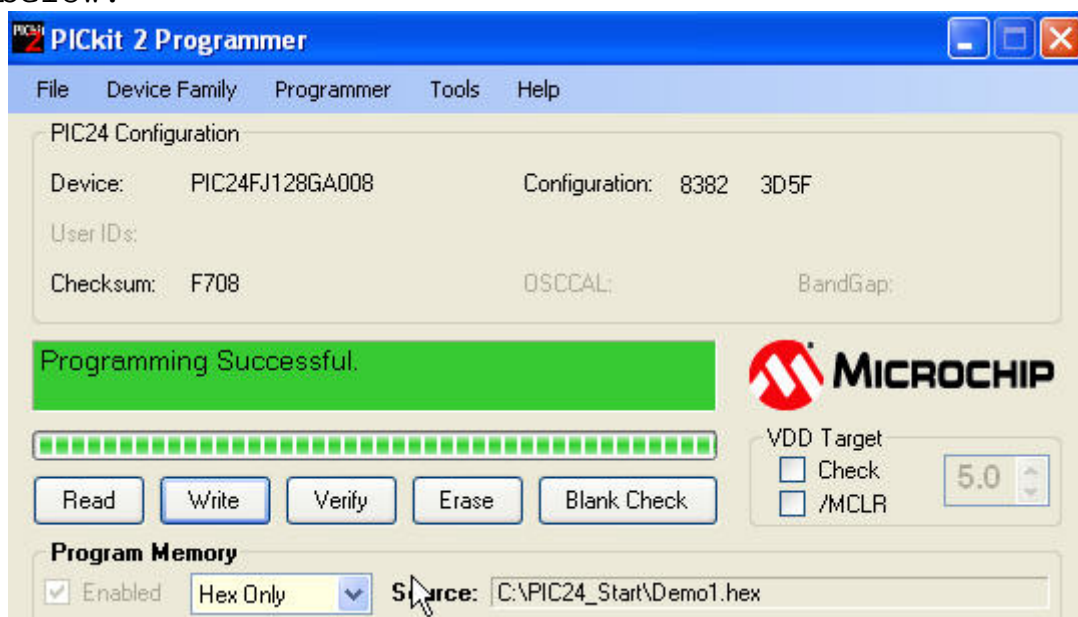
- Import File.HEX, click **File -> Import** and enter **Folder PIC24_Start** to select **file Demo1.Hex** and then **Open**.



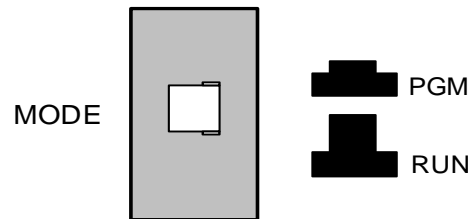
- It will display message **"Hex File successfully imported"**; click **Write** to program Hex code into memory of Microcontroller.



- When it has already programmed completely, it will display message **"Programming Successful"** as shown below.



- Shift **Switch MODE** to **RUN** position.



- Connect Hardware to test operation of program as follows;

PIC24FJ128GA008	LED
RD0	LED1
RD1	LED2
RD2	LED3
RD3	LED4
RD4	LED5
RD5	LED6
RD6	LED7
RD7	LED8

- Each LED from LED 1 to LED 8 will be ON one by one.
