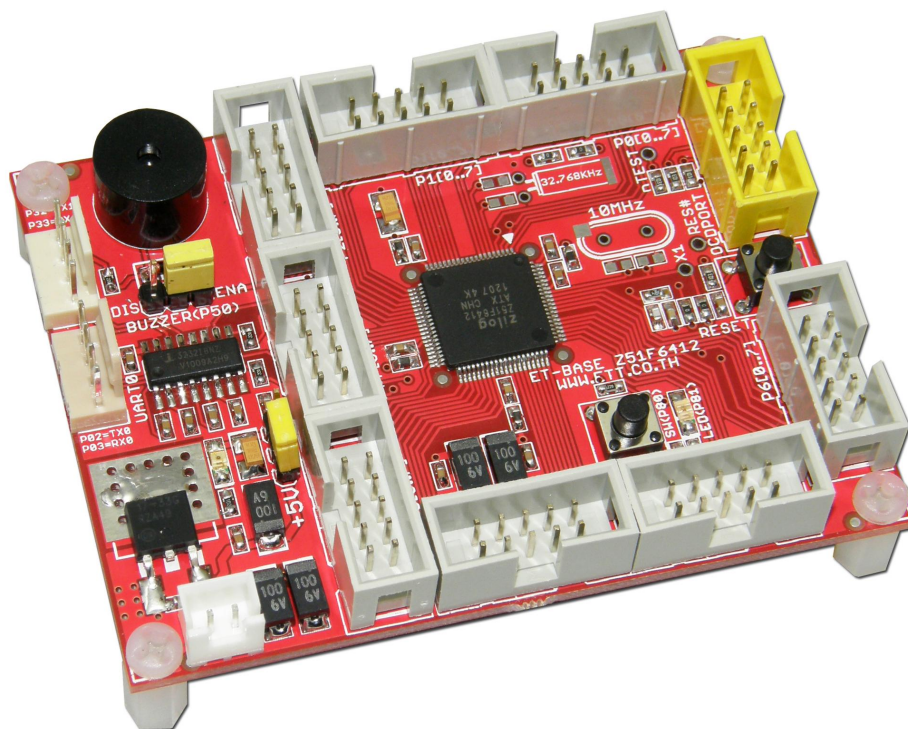


ET-BASE Z51F6412

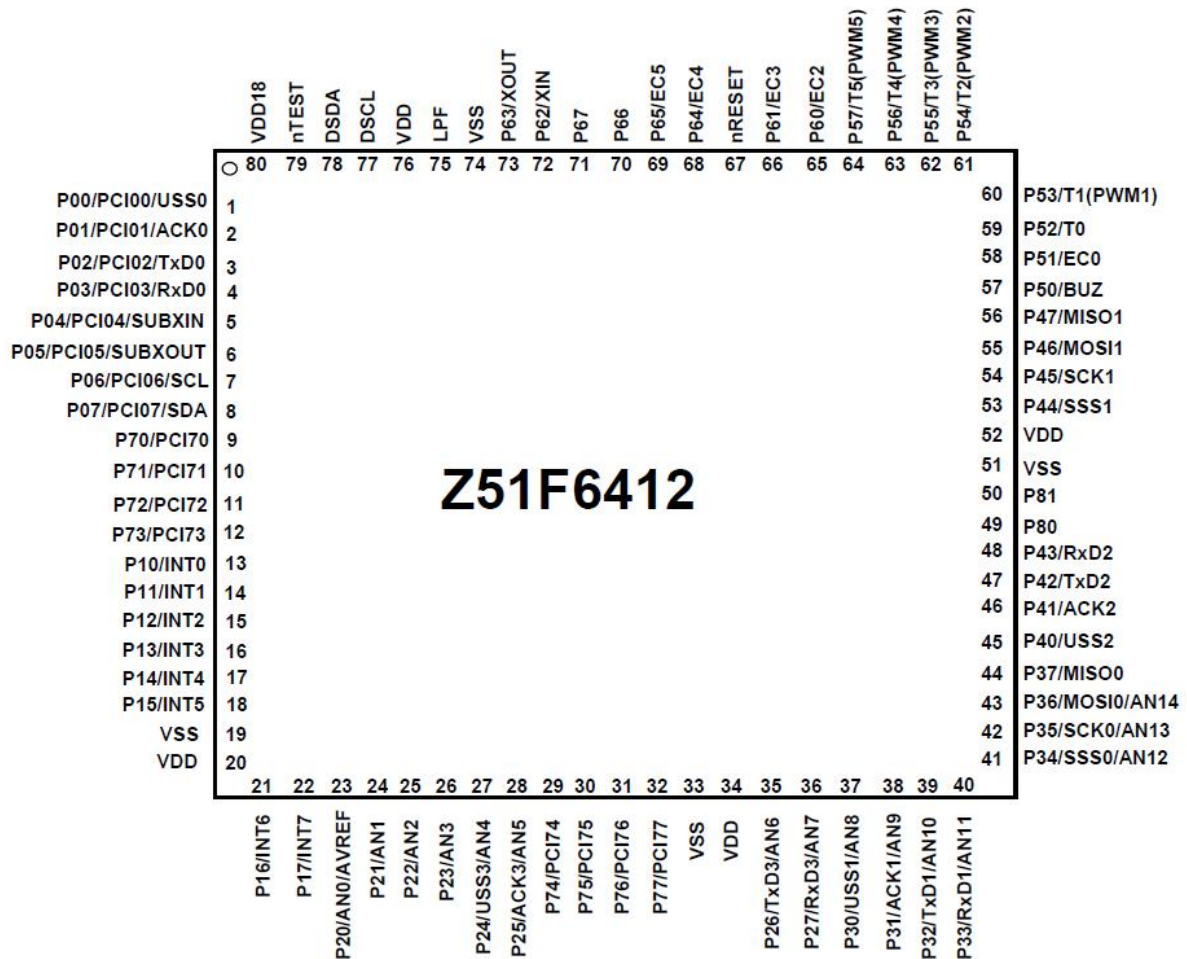
"ET-BASE Z51F6412" is Board Microcontroller in a series of MCS51 that is new release from Zilog Inc. In past years, Zilog Inc. has made and developed Microprocessor in a series of Z80 that is popular and widespread. Now, the Microcontroller in a series of MCS51 that Zilog Inc. has lately developed called Microcontroller "Z8051".

"Z8051" is 8BIT CISC Single Chip Microcontroller that uses the architectural structure for processing data of MCS51 according to the format Intel. However, Zilog has developed and modified all new operational system of Peripheral I/O such as I/O Port, UART and Timer/Counter; in this case, it still controls and accesses these Peripheral I/Os through SFR (Special Function System) as same as Intel. So, the format of writing program to command the operation is still the same as MCS51 of Intel. Moreover, Zilog also develops and improves efficiency of Z8051 higher than the standard MCS51 such as speed of processing data, increasing unit for mathematical calculation of 32Bit integral multiplication and division, energy management, and adding special Peripheral I/Os such as GPIO Port, Pin Pull-Up, Pin Debounce, Pin Change Interrupt, USART, SPI, I2C, ADC, Timer/Counter/Capture/PWM, and Buzzer Control. Furthermore, it can support Interrupt better than the standard MCS51. As mentioned above, user can write program to command Hardware easily and it reduces much complex procedures for developing program.

Specifications of Board ET-BASE Z51F6412

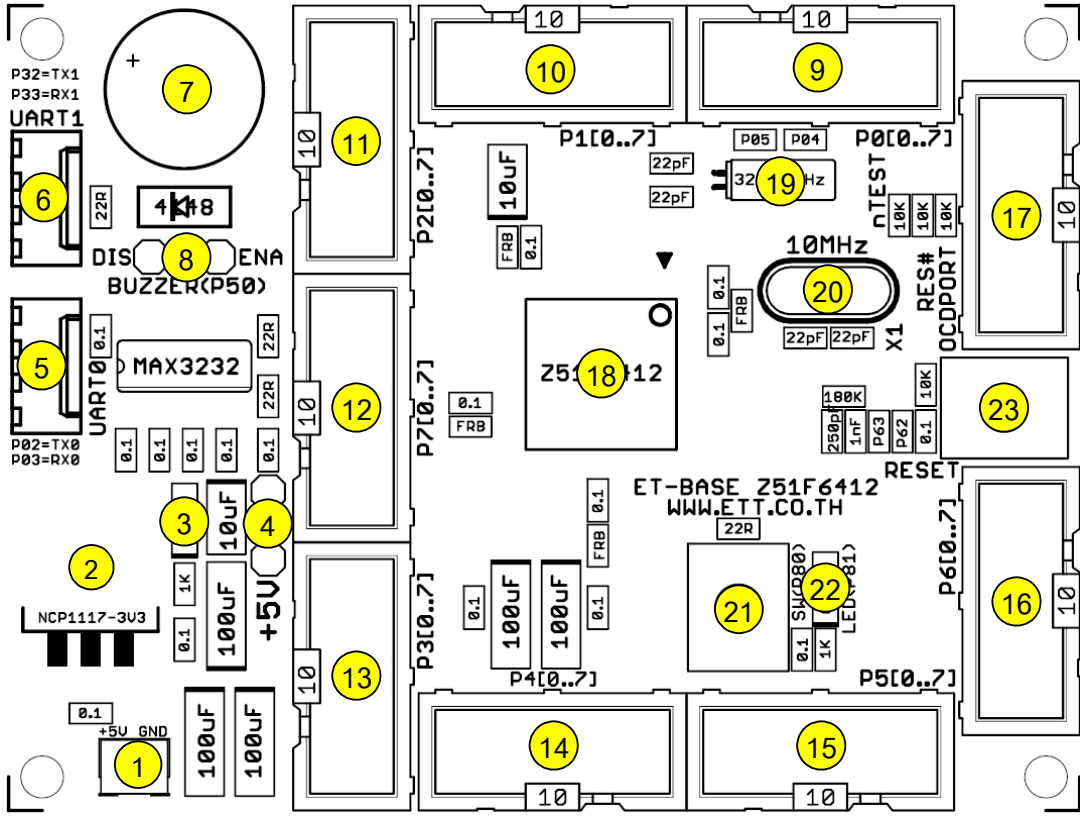
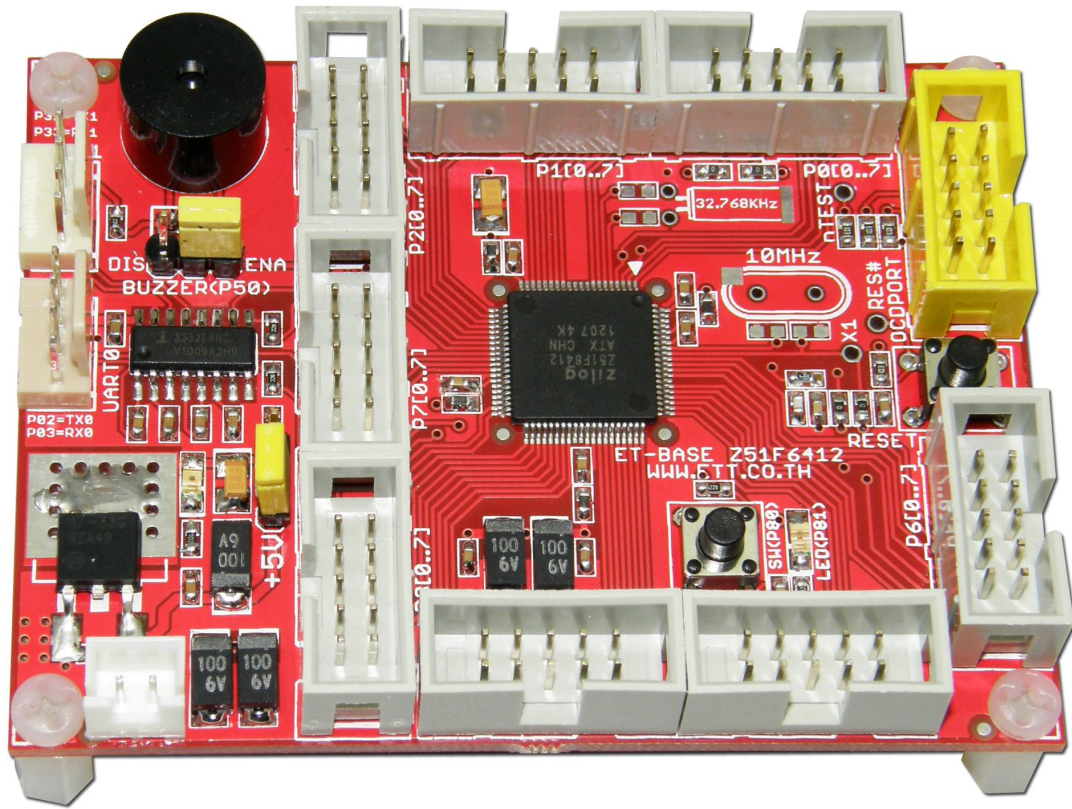
- Use MCU Z8051 No.Z51F6412 on board (MCS51 from Zilog Inc.); Run by the maximum frequency of 16MHz (processing data with 125ns/1 Machine Cycle) from Internal Oscillator
 - Has 64KByte Flash/ 3KByte XRAM/ 256 Byte IRAM
 - Use architectural structure of MCS51 for processing data (2 Clock/ 1 Machine Cycle)
 - Has Internal Oscillator 16MHz (inconsistency is not higher than +/-2%), dividend by 2,4,8,16, from program
 - Has 66Bit GPIO Port that can be programmed to be various Peripheral I/O as follows;
 - ❖ 15 Channel 12Bit ADC
 - ❖ 4 Channel UART
 - ❖ 2 Channel SPI
 - ❖ 1 Channel I2C
 - ❖ 2 Channel 8Bit Timer/Counter (T0,T1); it can be used together to be 16Bit 1 Channel
 - ❖ 4 Channel 16 Bit Timer/Counter/PWM(T2,T3,T4,T5)
 - ❖ 8 Bit External Interrupt Trigger(INT0...INT7)
 - ❖ 16 Bit Pin Change Interrupt Trigger(P0,P7)
 - ❖ Internal Pin Pull-Up; every pin can be set to Enable/Disable independently
 - ❖ Internal Pin Debounce; every pin can be set to Enable/Disable independently
 - ❖ 1 Channel Buzzer Drive
 - Support Interrupt from 32 sources devices, 32 Vector
 - Has Circuit Calculator for 32Bit integral multiplication and division
 - ❖ 16Bit x 16Bit integral multiplication, it takes 1 Cycle Clock.
 - ❖ 32Bit/ 16Bit integral division, it takes 32 Cycle Clock.
 - Watch Timer and Watch Dog Timer
 - Power-ON Reset at 1.4V
 - Programmable Brown-Out Detect(1.6V, 2.5V, 3.6V and 4.2V)
- Has Crystal Frequency 10 MHz (Option) for application that requires very high accuracy
- Has Crystal Frequency 32 KHz (Option) + PLL; it can set the frequency multiplication to be 14.75 MHz.
- Has 2 Channels Circuit Line Driver for RS232 UART Serial Port Communication by using Connector UART CPA-A 4Pin under standard of ETT
 - 1 Channel for Hardware UART0 by using Pin P0.2(TX0) and Pin P0.3(RX0)
 - 1 Channel for Hardware UART1 by using Pin P3.2(TX1) and Pin P3.3(RX1)
- Has Buzzer with Jumper to connect/disconnect Buzzer Driver (P5.0)

- Has 1 Push-Button Switch to test operation of Input by using P8.0
- Has 1 LED to display status when testing operation of Output by using P8.1
- Has SW RESET with Circuit RC-Reset
- Has 8 of Connector I/O Header 2x5(P0,P1,P2,P3,P4,P5,P6,P7)
- Has Connector OCD-PORT 10Pin IDE according the standard of Zilog; it can be used with Program Developer and Debugger under standard of Z8051 on-Chip-Debugger/Programmer (ET-Z8051 OCD)
- Has Power +5VDC Input with Regulate 3.3V/1A and LED to display operating status of Power. Has Jumper to choose Power Supply for providing MCU either to be +5VDC or 3.3VDC.
- Be Mini PCB size: 8 x 6 cm.



This structure shows Pin arrangement of Z51F6412 80-LQFP.

Structure of Board ET-BASE Z51F6412

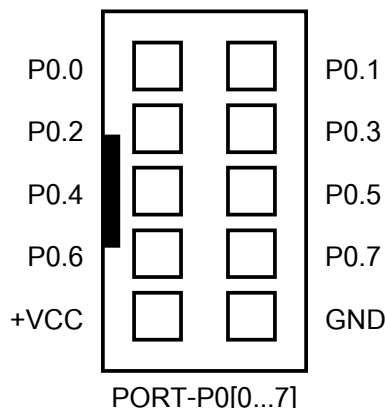


Picture shows structure of Board ET-BASE Z51F6412.

- **No.1:** It is Connector +5VDC Power Supply of Board.
- **No.2:** It is IC Regulate 3.3VDC/1A.
- **No.3:** It is LED to display status of +VCC Power Supply.
- **No.4:** It is Jumper to choose voltage level of MCU(+VCC) between 3.3V and 5V.
- **No.5:** It is Connector UART0. It is Signal RS232 that supports Hardware UART0; it uses Pin of P0.2(TX0) and P0.3(RX0) for connection.
- **No.6:** It is Connector UART1. It is Signal RS232 that supports Hardware UART1; it uses Pin of P3.2(TX1) and P3.3(RX1) for connection.
- **No.7:** It is Buzzer to generate sound.
- **No.8:** It is Jumper to choose Connect/Disconnect signal between P5.0 and Buzzer.
- **No.9:** It is Connector IDE 10PIN of P0[0..7].
- **No.10:** It is Connector IDE 10PIN of P1[0..7].
- **No.11:** It is Connector IDE 10PIN of P2[0..7].
- **No.12:** It is Connector IDE 10PIN of P7[0..7].
- **No.13:** It is Connector IDE 10PIN of P3[0..7].
- **No.14:** It is Connector IDE 10PIN of P4[0..7].
- **No.15:** It is Connector IDE 10PIN of P5[0..7].
- **No.16:** It is Connector IDE 10PIN of P6[0..7].
- **No.17:** It is Connector of OCD-PORT (On-Chip-Debug) to connect with OCD Device.
- **No.18:** It is MCU No.Z51F6412 on board (80Pin LQFP Package).
- **No.19:** It is Sub Crystal Frequency 32.768KHz (Option). If user requires installing and using this Crystal, it has to reserve Pin P0.4(SUBXIN) and P0.5(SUBXOUT) for connection.
- **No.20:** It is Main Crystal Frequency 10.00MHz (Option). If user requires installing and using this Crystal, it has to reserve Pin P6.2(XIN) and P6.3(XOUT) for connection.
- **No.21:** It is Push-Button Switch (SW1) to test the operation of Digital Input; this SW is interfaced with signal pin from Port P8.0.
- **No.22:** It is LED to test the operation of Digital Output; this LED is interfaced with signal pin from Port P8.1.
- **No.23:** It is Switch RESET to reset the operation of MCU.

CONNECTORS

PORT P0[0...7]



PORT-P0[0...7]: There is 8Bit Signal entirely from Port P0 of MCU; P0.2 and P0.3 is connected with Circuit Line Driver of RS232 (UART0) completely. If it is also connected with RS232 of UART0, it cannot use Pin P0.2 and P0.3 to be I/O any more.

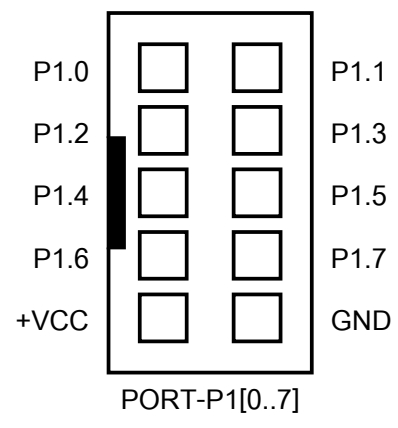
Signal from Port P0 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port (P0[7...0]).
- It is used to be Input Pin Change Interrupt:PCI0[7...0]).
- It is used to be UART0 Function (P0[3...0]).
- It is interfaced with Sub Crystal Oscillator 32.768 KHZ.
- It is used to be I2C BUS Function (P0[7...6]).

Port	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
GPIO	I/OP0.7	I/OP0.6	I/OP0.5	I/O P0.4	I/OP0.3	I/O P0.2	I/O P0.1	I/O P0.0
PCIO	PCI0.7	PCI0.6	PCI0.5	PCI0.4	PCI0.3	PCI0.2	PCI0.1	PCI0.0
UART0	-	-	-	-	RXD0	TXD0	ACK0	USS0
XTAL	-	-	XOUT	XIN	-	-	-	-
I2C	SDA	SCL	-	-	-	-	-	-

Table shows functions for using Pin Ports of Port P0.

PORT P1[0..7]



PORT-P1[0..7]: There is 8Bit Signal entirely from Port P1 of MCU. Pin of this P1 is floated independently; so, user can choose and use all 8Bit as preferred.

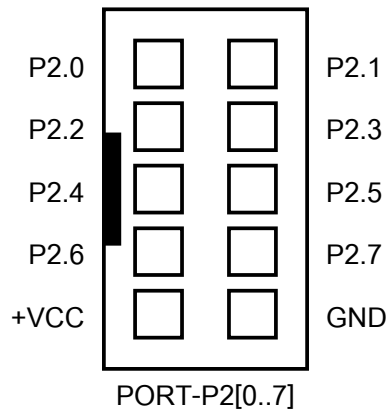
Signal from Port P1 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port (P1[7..0]).
- It is used to be Pin External Interrupt: INT[7..0].

Port	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
GPIO	I/O P1.7	I/O P1.6	I/O P1.5	I/O P1.4	I/O P1.3	I/O P1.2	I/O P1.1	I/O P1.0
INT	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0

Table shows functions for using Pin Ports of Port P1.

PORT-P2[0...7]



PORT-P2 [0...7]: There is 8Bit Signal entirely from Port P2 of MCU. Pin of this P2 is floated independently; so, user can choose and use all 8Bit as preferred.

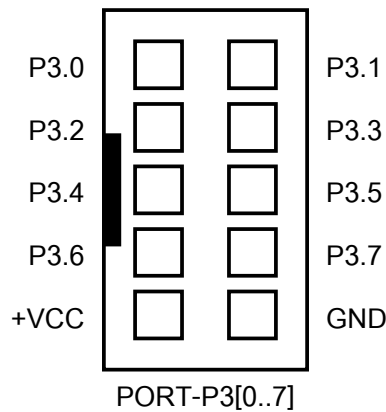
Signal from Port P2 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port (P2[7...0]).
- It is used to be Pin Input Analog ADC with 12BIT Resolution (AN[7...0]).
- It is used to be UART3 Function (P2[7...4]).

Port	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
GPIO	I/O P2.7	I/O P2.6	I/O P2.5	I/O P2.4	I/O P2.3	I/O P2.2	I/O P2.1	I/O P2.0
ADC	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
UART3	RXD3	TXD3	ACK3	USS3	-	-	-	-

Table shows functions for using Pin Ports of Port P2.

PORT P3[0..7]



PORT-P3[0..7]: There is 8Bit Signal entirely from Port P3 of MCU; P3.2 and P3.3 is connected with Circuit Line Driver of RS232 (UART1) completely. If it is also connected with RS232 of UART1, it cannot use Pin P3.2 and P3.3 to be I/O any more.

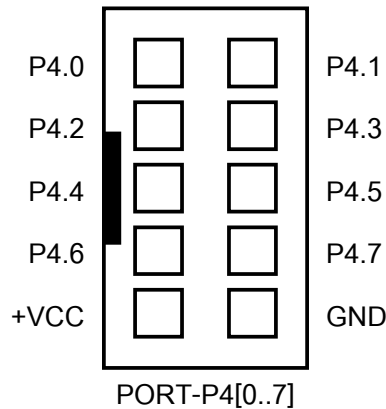
Signal from Port P3 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port(P3[7..0]).
- It is used to be Pin Input Analog ADC with 12Bit Resolution (AN[14..8]).
- It is used to be UART1 Function (P3[3..0]).
- It is used to be SPI0 Function (P3[7..4]).

Port	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
GPIO	I/O P3.7	I/O P3.6	I/O P3.5	I/O P3.4	I/O P3.3	I/O P3.2	I/O P3.1	I/O P3.0
ADC	-	AN14	AN13	AN12	AN11	AN10	AN9	AN8
UART1	-	-	-	-	RXD1	TXD1	ACK1	USS1
SPI0	MISO0	MOSIO	SCK0	SSS0	-	-	-	-

Table shows functions for using Pin Ports of Port P3.

PORT P4[0..7]



PORT-P4[0..7]: There is 8Bit Signal entirely from Port P4 of MCU. Pin of this P4 is floated independently; so, user can choose and use all 8Bit as preferred.

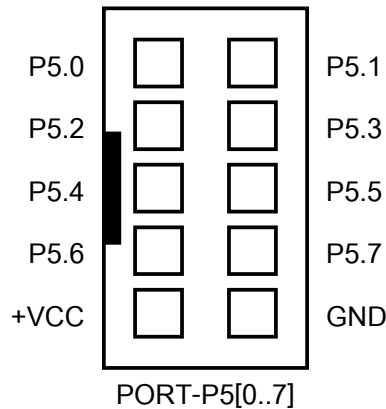
Signal from Port P4 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port(P4[7..0])
- It is used to be UART2 Function(P4[3..0]).
- It is used to be SPI1 Function(P4[7..4]).

Port	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
GPIO	I/O P4.7	I/O P4.6	I/O P4.5	I/O P4.4	I/O P4.3	I/O P4.2	I/O P4.1	I/O P4.0
UART2	-	-	-	-	RXD2	TXD2	ACK2	USS2
SPI1	MISO1	MOSI1	SCK1	SSS1	-	-	-	-

Table shows functions for using Pin Ports of Port P4.

PORT P5[0..7]



PORT-P5[0..7]: There is 8Bit Signal entirely from Port P5 of MCU. Pin of this P5 is floated independently; so, user can choose and use all 8Bit as preferred.

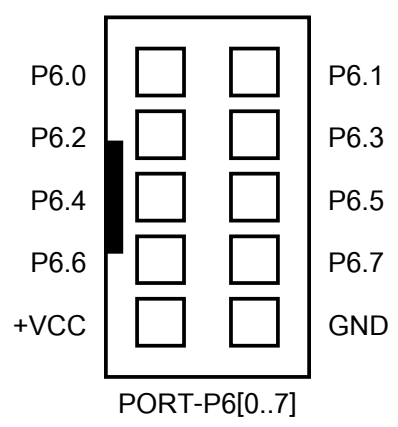
Signal from Port P5 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port(P5[7..0]).
- It is used to be Buzzer Drive(P5[0]).
- It is used to be PWM Function(P5[7..3]).
- It is used to be Timer Input/Output Function(P5[7..1]).

Port	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
GPIO	I/O P5.7	I/O P5.6	I/O P5.5	I/O P5.4	I/O P5.3	I/O P5.2	I/O P5.1	I/O P5.0
BUZZER	-	-	-	-	-	-	-	BUZZER
PWM	PWM5	PWM4	PWM3	PWM2	PWM1	-	-	-
Timer	T5	T4	T3	T2	T1	T0	EC0	-

Table shows functions for using Pin Ports of Port P5.

PORT P6[0..7]



PORT-P6[0..7]: There is 8Bit Signal entirely from Port P6 of MCU. Pin of this P6 is floated independently; so, user can choose and use all 8Bit as preferred.

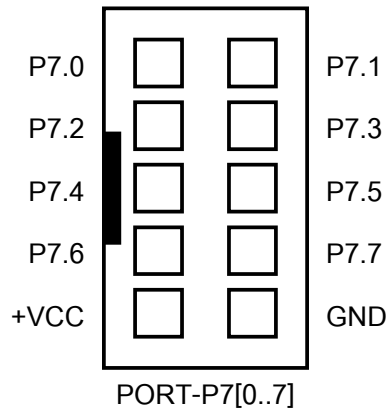
Signal from Port P6 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port(P6[7..0]).
- It is used to be Main Crystal Oscillator(P6[3..2]).
- It is used to be Timer Input Function(P6[5,4,1,0]).

Port	P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0
GPIO	I/O P6.7	I/O P6.6	I/O P6.5	I/O P6.4	I/O P6.3	I/O P6.2	I/O P6.1	I/O P6.0
XTAL	-	-	-	-	XOUT	XIN	-	-
Timer	-	-	EC5	EC4	-	-	EC3	EC2

Table shows functions for using Pin Ports of Port P6.

PORT P7[0..7]



PORT-P7[0..7]: There is 8Bit Signal entirely from Port P7 of MCU. Pin of this P7 is floated independently; so, user can choose and use all 8Bit as preferred.

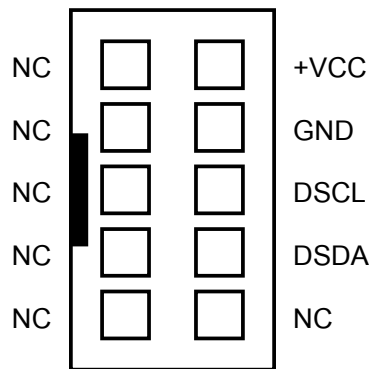
Signal from Port P7 of Z51F6412 can be programmed to be various functions; it can be normal GPIO Input/Output or it is used with special functions as follows;

- It is used to be GPIO Input/Output Port (P7[7..0]).
- It is used to be Input Pin Change Interrupt : PCI7[7..0].

Port	P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0
GPIO	I/O P7.7	I/O P7.6	I/O P7.5	I/O P7.4	I/O P7.3	I/O P7.2	I/O P7.1	I/O P7.0
PCI7	PCI7.7	PCI7.6	PCI7.5	PCI7.4	PCI7.3	PCI7.2	PCI7.1	PCI7.0

Table shows functions for using Pin Ports of Port P7.

PORT OCD (On-Chip Debug)

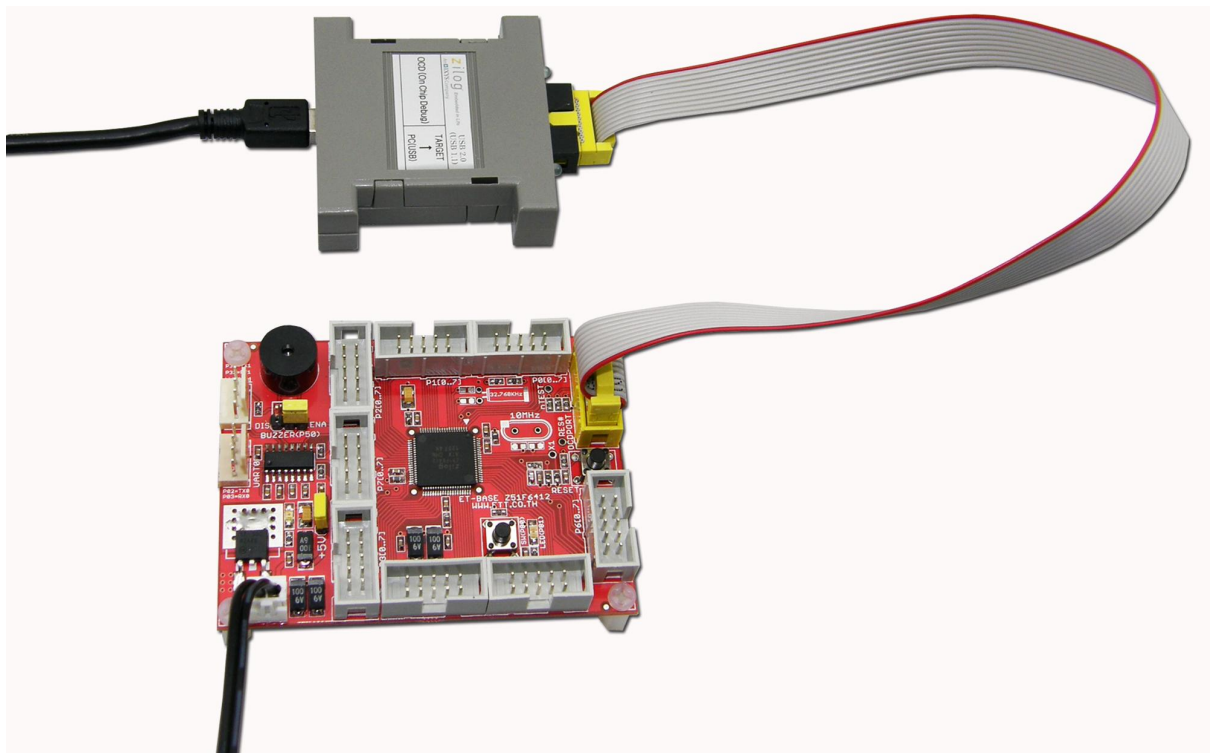


PORT-OCD(DEBUG)

PORT-OCD: It is Connector to connect with OCD/ISP Device to program and control Debugging for MCU on board. It supports the connection with device that has been designed for the ISP Application (In-System Programming) and Debug of MCU Z8051 according to the standard restriction of Zilog as follows;

- Zilog Z8051 OCD of Zilog Inc.
- ET-Z8051 OCD of ETT

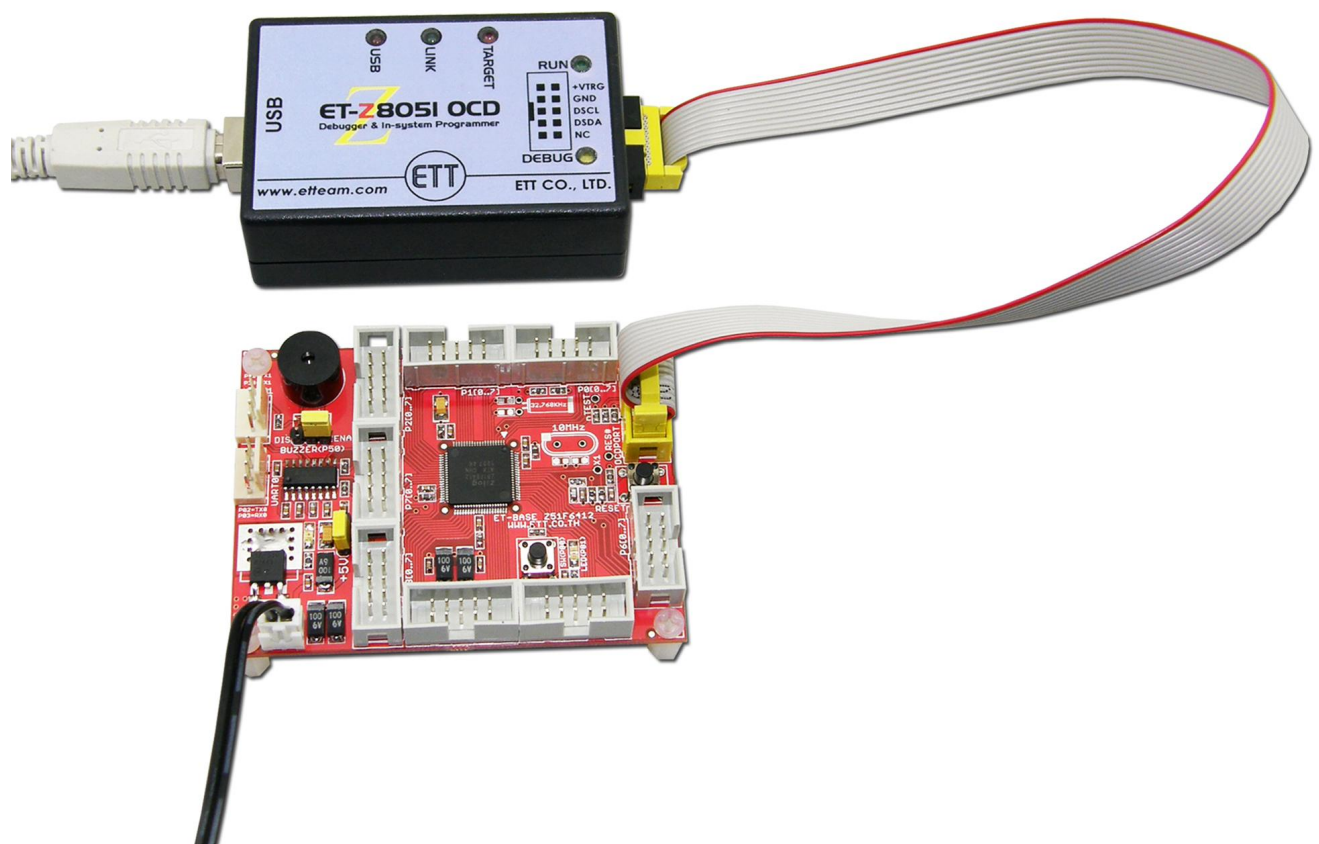
The connection between Board ET-BASE Z51F6412 and OCD Device is performed through 10Pin Pair Cable as shown in the example below;



Picture illustrates the connection between Board ET-BASE Z51F6412 and Zilog Z8051 OCD.



Picture shows the feature of ET-Z8051 OCD from ETT.



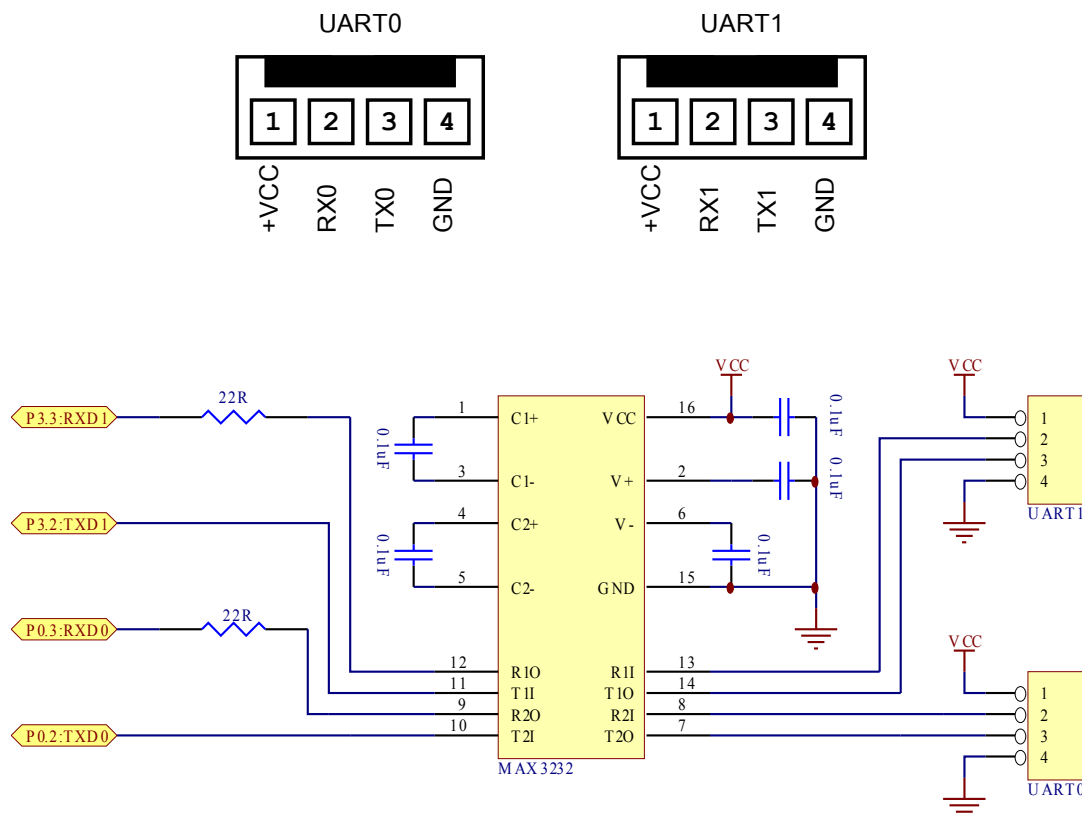
Picture illustrates the connection between Board ET-BASE Z51F6412 and ET-Z8051 OCD.

How to use RS232

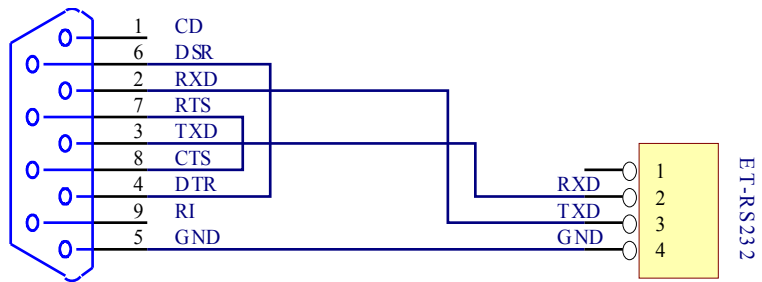
Port RS232 is Signal RS232 that has converted signal level from MAX3232 completely. In case of Z51F6412, it normally has 4 Channel Hardware UART; it uses Pins as follows;

- UART0 uses P0.2(TX0) and P0.3(RX0).
- UART1 uses P3.2(TX1) and P3.3(RX1).
- UART2 uses P4.2(TX2) and P4.3(RX2).
- UART3 uses P2.6(TX3) and P2.7(RX3).

It designs the Circuit of board to use 2 Channels of Hardware UART; it interfaces with Circuit RS232 Line Driver that is UART0 and UART1. For UART2 and UART3, it is floated independently, so user can design and use this signal for suitable applications; for example, it is used to be RS422/RS485; or, it is used to be UART Communication with other devices directly in the format of TTL Level. Signal of RS232 in each channel is arranged in the format of Connector CPA-4PIN(RS232) as shown in the picture below;

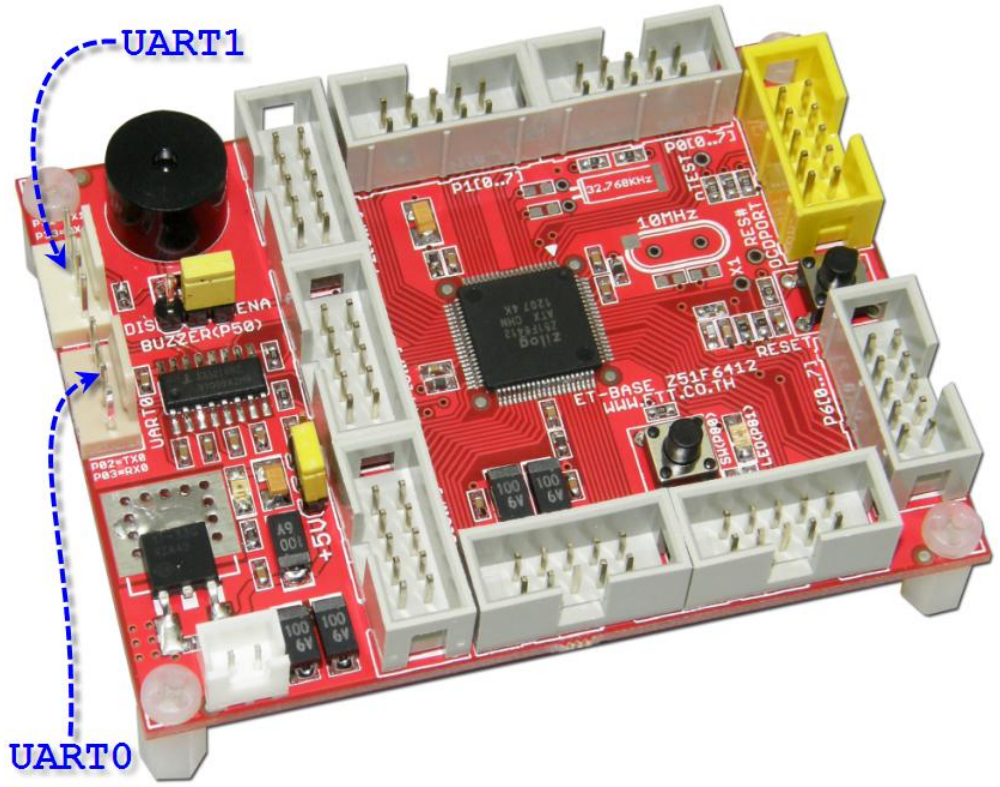


The Cable that is used to connect Signal RS232 between ComPort of computer PC and Connector RS232 of Board ET-BASE Z51F6412 is shown in the picture below;



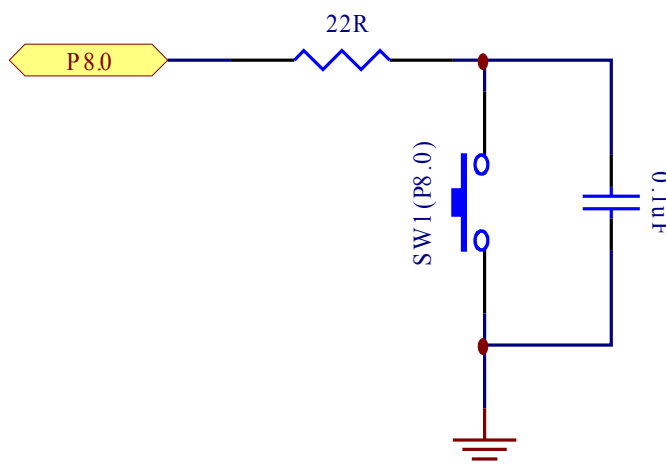
DB9(FEMALE)
PC-RS232

Picture shows Circuit of Cable RS232.



How to use Circuit SW1

SW1 is Circuit Push-Button Switch to create Signal Logic "0" and "1" to test the operation of Logic Input. For example, it tests to detect the status of pressing Switch; when the Switch is not pressed, the Logic is "1"; or, when the Switch is pressed, the Logic is "0". Signal Logic from this Circuit is connected to Pin P8.0 of MCU as shown in the picture.



Because it designs Port structure of Z51F6412 to be flexible and there are various circuits insides that assist user to choose and apply each part of the circuit as preferred. In the part of Port Input that is used to read status from TOUCH Switch, Z51F6412 has 4 Registers that are related to this Port Input; so, user can choose and setup the operation of Port suitably as follows;

- P8 is Port P8 Data Register of Port P8; it is used to read status of Pin Port.
- P8IO is Port P8 Direction of Port P8; it is used to set function of Port to be IN or OUT.
- P8PU is Port P8 Pull-Up; it is used to open Pull-Up internal Chip or not.
- P8DB is Port P8 Debounce; it is used to open Circuit Debounce internal Chip or not.

Each Register is 8Bit; it separates function of each bit to control Pin Port of each Pin independently. In case of SW1 that is connected with P8.0, it only uses data in Bit 0 of each Register to control the operation of Pin P8.0 as shown in the example below;

```

#include <z51f6412.h> // Z51F6412 Register

/*****
/* ET-BASE Z51F6412 Hardware SW Pin */
*****/
#define SW_PIN          (1 << 0) // P8[0] = SW Pin
#define SW_PORT_DIR     P8IO    // Port P8 Direction
#define SW_PORT_DATA    P8      // Port P8 Data
#define SW_PORT_PULLUP  P8PU    // Port P8 Pull-Up
#define SW_PORT_DEBOUNCE P8DB  // Port P8 Debounce

// Bit Variable
static bit this_sw;
static bit last_sw;
sbit SW_READ = P8^0; // Pin Read SW = P8.0

void main(void)
{
    .
    .
    .
    SW_PORT_DIR      &= ~(SW_PIN); // Config SW Pin = Input
    SW_PORT_DEBOUNCE |= (SW_PIN); // Enable Debounce For SW Pin
    SW_PORT_PULLUP   |= (SW_PIN); // Enable Pull-Up For SW Pin
    SW_PORT_DATA     |= (SW_PIN); // Default Logic "1"

    last_sw = 1; // Default SW Status = Release

    while(1)
    {
        this_sw = SW_READ; // Read Bit SW
        if(this_sw != last_sw) // If SW Status Change
        {
            if((last_sw==1)&&(this_sw==0)) //Verify SW Press & Service
            {
                .
                .
                .
            }

            if((last_sw==0)&&(this_sw==1)) //Verify SW Release & Service
            {
                .
                .
                .
            }

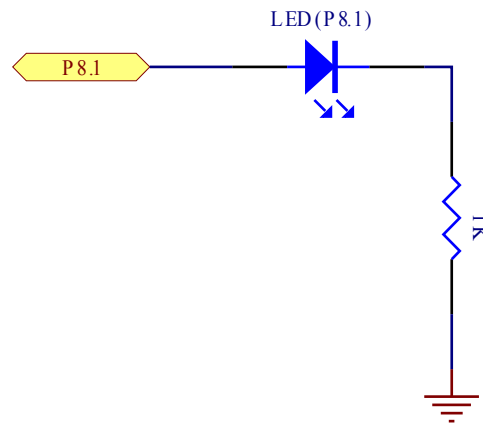
            last_sw = this_sw; //Update SW Reference Status
        }
    }
}

```

It shows an example Code for Initial SW1.

How to use Circuit LED

LED is circuit that displays the logical result (LOGIC) to user; it is used with Signal Logic Output. If it receives Logic "1", LED is ON; but, it receives Logic "0", LED is OFF. The Signal Logic that drives the LED display in this Circuit is connected from Pin P8.1 of MCU as shown in the picture below;



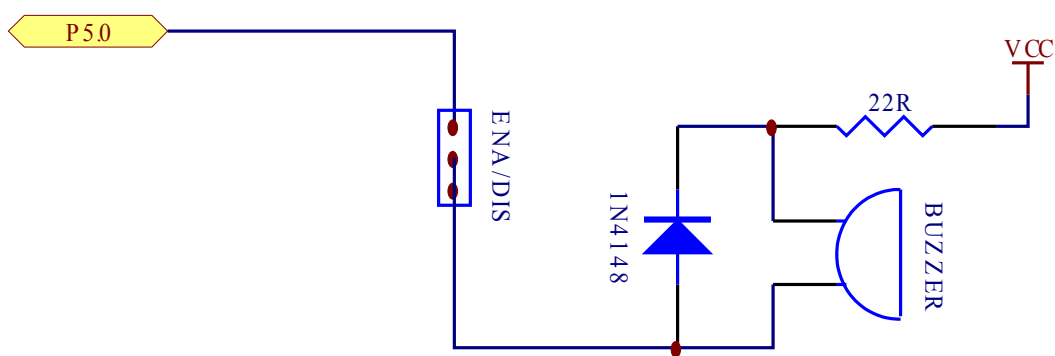
```
#include <z51f6412.h> // Z8051 : Z51F6412

/*****
/* ET-BASE Z51F6412 Hardware LED Pin */
*****/
#define LED_PIN          (1 << 1) // P8[1] = LED
#define LED_PORT_DIR     P8IO     // Port P8 Direction
#define LED_PORT_DATA    P8       // Port P8 Data
.
.
.
void main(void)
{
.
.
.
LED_PORT_DIR |= (LED_PIN); // Configure LED = Output
...
LED_PORT_DATA |= (LED_PIN) // LED Pin = 1 (ON LED)
...
LED_PORT_DATA &= ~(LED_PIN) // LED Pin = 0 (OFF LED)
...
LED_PORT_DATA ^= (LED_PIN) // LED Pin = Toggle
.
.
.
}
```

It shows an example Code for Initial LED.

How to use Circuit Buzzer

Buzzer is Circuit to generate various sounds by different frequencies. The Z51F6412 has included this circuit to control sound generator of Buzzer internal structure of Chip; it uses Pin P5.0 to control sound generating of Buzzer. The special Circuit in this part reduces many complications about writing program to create frequency to control sound generating of Buzzer; in this case, user only sets value to setup the operation of Buzzer, especially the related Register, internal circuit sends Signal to control Buzzer automatically; so, it is easier to write program. Signal Pin Port that is used to drive Buzzer is to use Pin P5.0; moreover, there is Jumper to connect/disconnect signal as preferred as shown in the picture below;



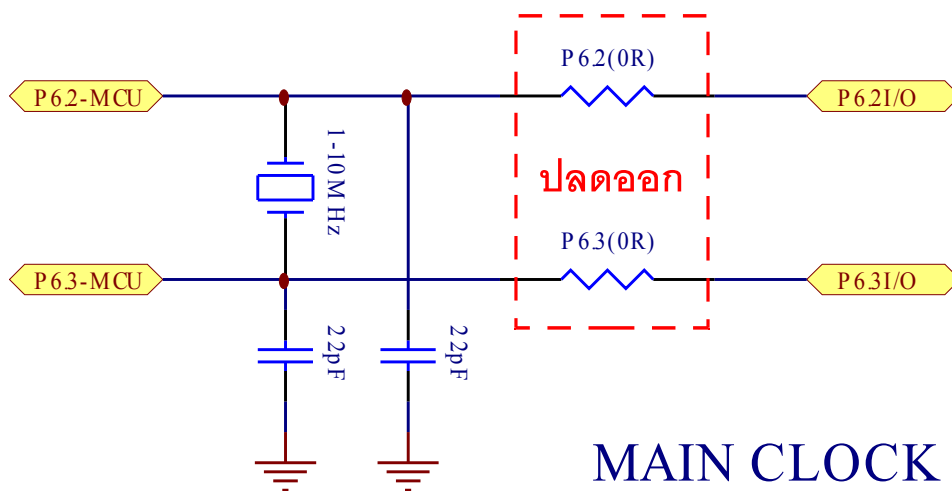
```
#include <z51f6412.h> // z8051 : Z51F6412
.
.
void main(void)
{
.
.
// Config Buzzer
// BUZCR = 00000,11,0
// 00000xxx : Default Reserve Bit
// xxxxx11x : Buzzer Clock Source = Fx/256
// xxxxxxx0 : Buzzer Disable
BUZCR |= 0x06;
BUZDR = 0x1F; //Buzzer Frequency

while(1)
{
BUZCR |= 0x01; //ON Buzzer (xxxxxxx1:Enable Buzzer)
.
.
BUZCR &= ~0x01; //OFF Buzzer (xxxxxxx0:Disable Buzzer)
}
}
```

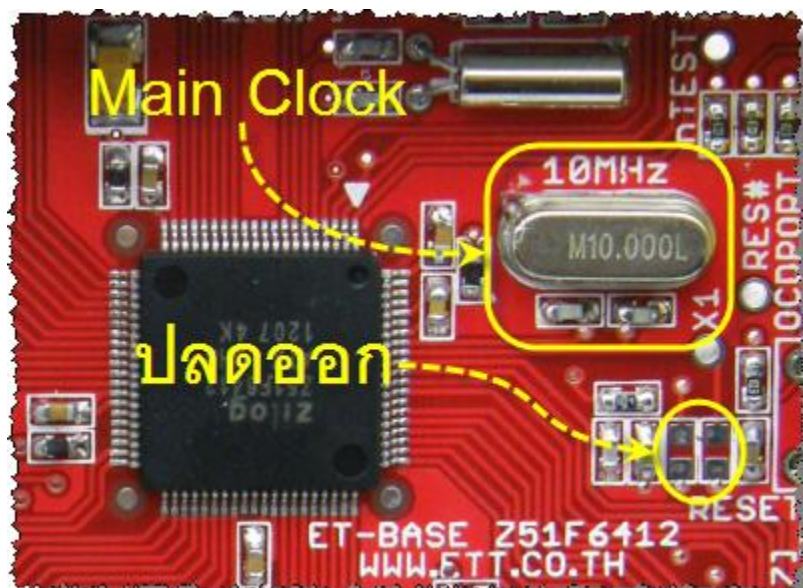
It shows an example Code for Initial Buzzer.

How to use Circuit Main Clock Oscillator

Main Clock Oscillator is external Signal Clock that uses Crystal Oscillator in the range of 1 MHz to 10 MHz to generate frequency; the Signal Clock from this circuit is high accuracy, so it is suitable for the application that requires very high accuracy. When setting MCU to run from the Signal Clock of this Main Clock Oscillator, user has to install the Crystal in the range of 1-10 MHz and 22pF Capacitor into the board; however, it has to lose 2 Pin I/Os that are P6.2 and P6.3 because it has to use both pins to interface with Module Crystal Oscillator. Moreover, it should remove 0-OHM Resistor from board to remove the connection between Signal P6.2 and P6.3 and Connector IDE 10 Pin of Port P6[0..7]; it protects this Signal Clock from loading by other external signal.

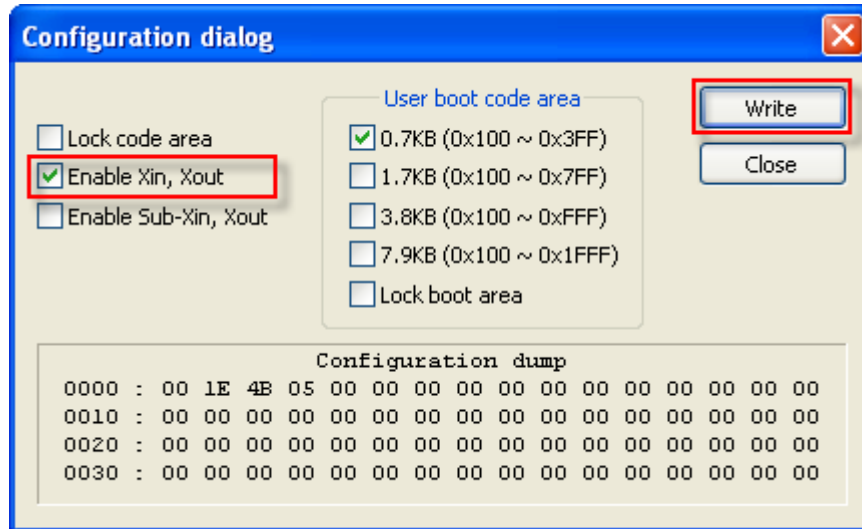


It shows how to install Crystal 1-10 MHz and Capacitor for Circuit Main Clock Oscillator.



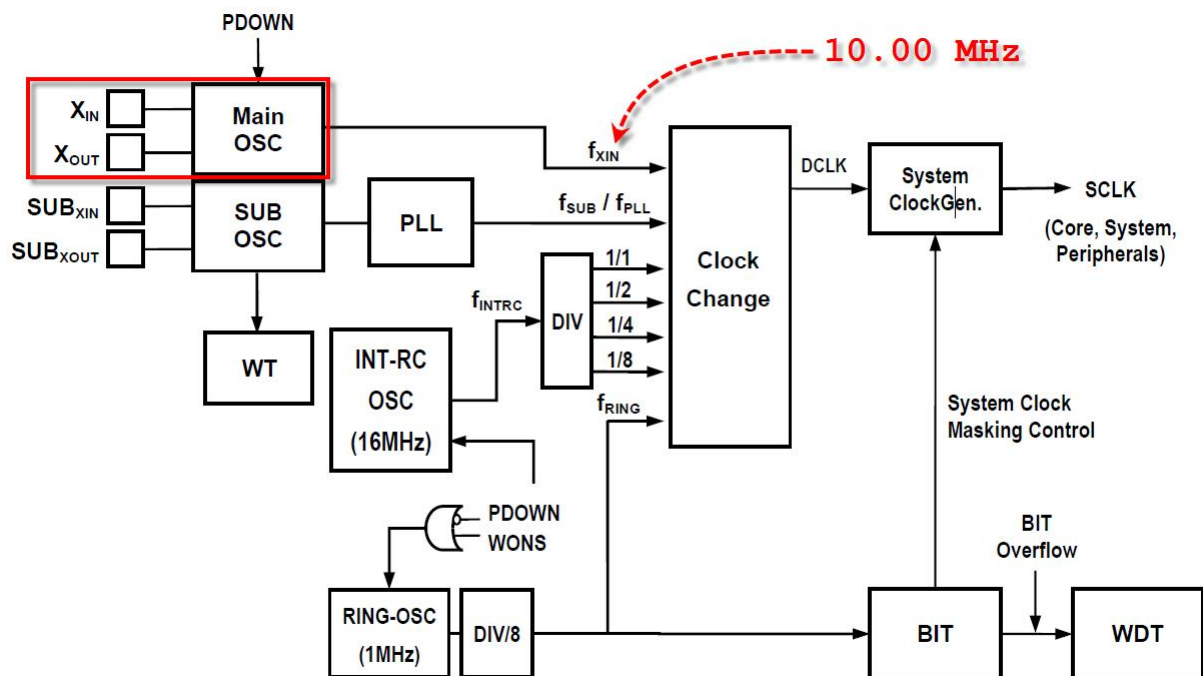
It shows how to install Main Clock and remove P6.2/P6.3 from Connector Port P6[7..0].

User can set this Circuit Main Clock Oscillator to run when MCU starts running by setting the value in Configuration Bit in the step of programming MCU. Moreover, there is another one choice that user can enable/disable the operation of Circuit Frequency Generator is to use commands in the program.



It shows how to set Configuration Bit to enable the operation of Circuit Main Clock Oscillator.

User can set MCU to run from source of this Signal Main Clock Oscillator by using commands in the program; moreover, user can switch to other sources of Signal Clock as preferable.



It shows how to choose and use Signal Clock from Circuit Main Clock Oscillator.

```

#include <z51f6412.h> // Z8051 : Z51F6412

void main(void)
{
    char ch;
    int i;

    /* Config System Clock = External XTAL 10.00 MHz */
    // PLLCR = 0,0,0,00,00,0
    // 0xxxxxxx : PLL Output Status
    // x0xxxxxx : PLL Output Bypass
    // xx0xxxxxx : Power PLL = Default
    // xxx00xxx : FBDiv = Default
    // xxxxx00x : PLL M = Default
    // xxxxxxx0 : PLL Disable
    PLLCR = 0x00; // Disable PLL

    // SCCR = 0,00,1,1,0,01
    // 0xxxxxxx : Stop Mode = Mode 2
    // x00xxxxxx : Clock Divide 1
    // xxx1xxxxx : Clock Change By Software
    // xxxx1xxx : RC Oscillator Disable
    // xxxxx0xx : XTAL Oscillator Enable
    // xxxxxx01 : System Clock Source = Main Clock(1~10 MHz)
    SCCR = 0x10; // Enable Main XTAL-10MHz

    for(i=0; i<32000; i++); // Delay for XTAL Stabilization
    SCCR = 0x19; // Select Clock = XTAL-10MHz &
                // Stop Internal RC

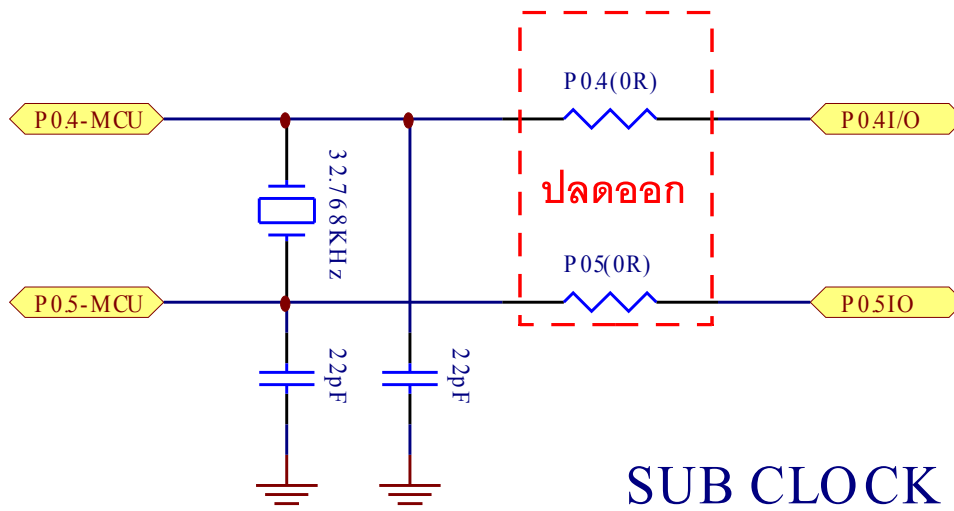
    /* Now System Clock = 10.00MHz */
    .
    .
    .
}

```

It shows an example program for setting MCU to run from Main Clock Oscillator.

How to use Circuit Sub Clock Oscillator

Sub Clock Oscillator is Clock System from external MCU that uses low Oscillator Frequency 32.768 KHz to reduce noise from Circuit Frequency Generator; this frequency is compatible with Circuit Phase-Lock-Loop to increase the frequency to 17.45 MHz. When setting MCU to run from the Signal Clock of this Circuit Sub Clock Oscillator, user has to install Crystal 32.768 KHz and 22pF Capacitor into the board; however, it has to lose 2 Pin I/Os that are P0.4 and P0.5 because it has to use both pins to interface with Module Crystal Oscillator 32.768 KHz. Moreover, it should remove 0-OHM Resistor from board to remove the connection between Signal P0.4 and P0.5 and Connector IDE 10 Pin of Port P0[0..7]; it protects this Signal Clock from loading by other external signal.

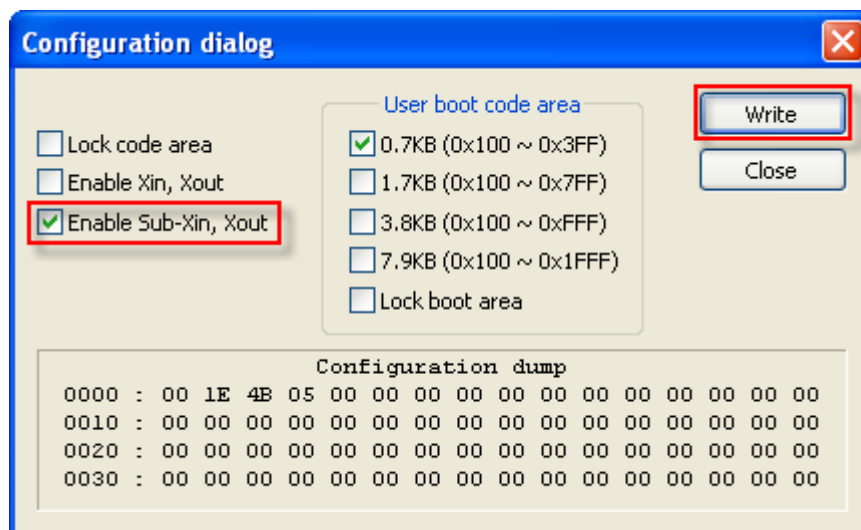


It shows how to install Crystal 32.768 KHz and Capacitor for Circuit Sub Clock Oscillator.



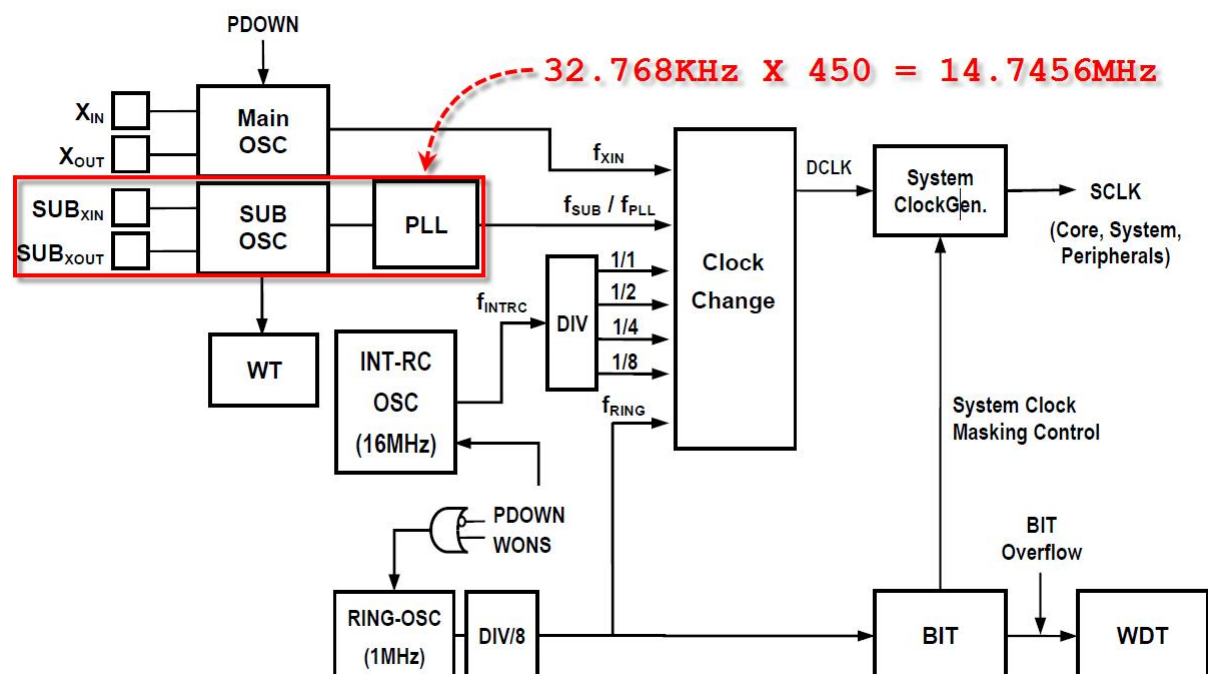
It shows how to install Circuit Sub Clock and remove P0.4/P0.5 from Connector Port P0[7...0].

User can set this Circuit Sub Clock Oscillator to run when MCU starts running by setting the value in Configuration Bit in the step of programming MCU. Moreover, there is another one choice that user can enable/disable the operation of Circuit Frequency Generator is to use commands in the program.



It shows how to setup Configuration Bit to enable the operation of Circuit Sub Clock Oscillator.

User can set MCU to run from source of this Signal Sub Clock Oscillator by using commands in the program. Moreover, it can enable internal Phase-Lock-Loop to set the frequency for running MCU at 11.0755 MHz (multiplied by 338) or running at the maximum speed of 14.7456 MHz (multiplies by 450) as preferred.



It shows how to choose and use Signal Clock from Circuit Sub Clock Oscillator + PLL.

```
#include <z51f6412.h> // Z8051 : Z51F6412

void main(void)
{
    char ch;
    //int i;

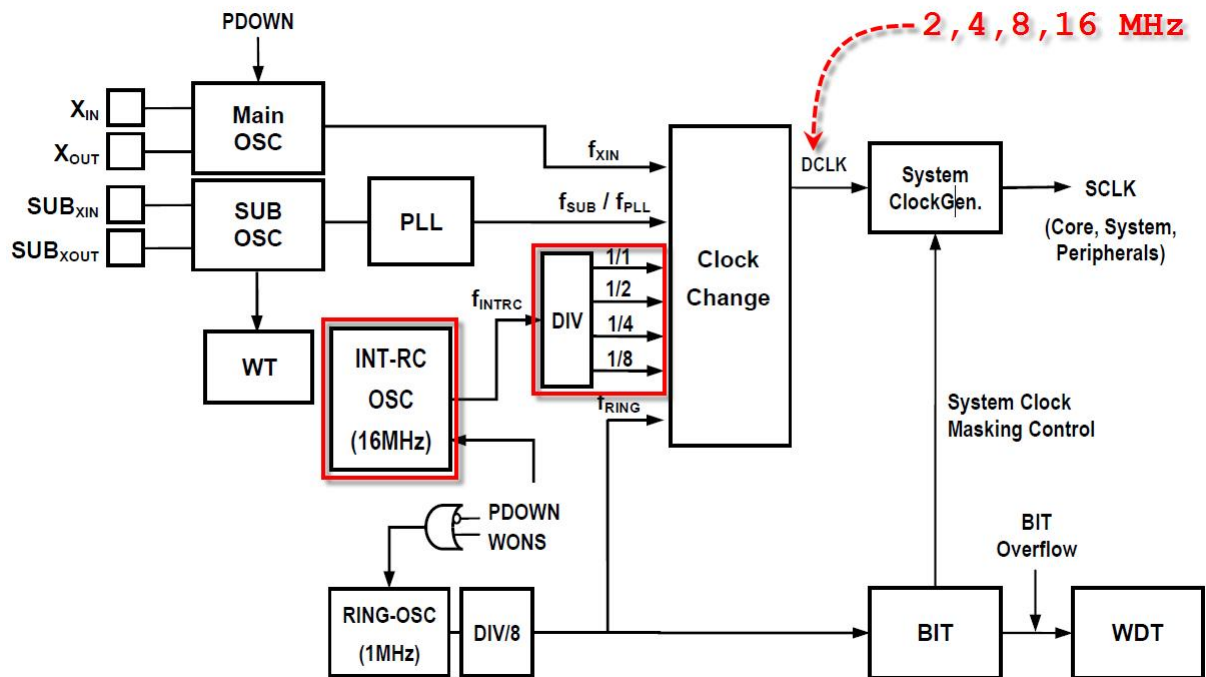
    /* Config System Clock = 32.768KHz+PLL(14.7456MHz) */
    //PLLCR = 0,1,0,10,00,1
    // 0xxxxxxx : PLL Status Read Only
    // x1xxxxxx : PLL Output Enable
    // xx0xxxxx : Power PLL = Default
    // xxx10xxx : FBDiv = 450
    // xxxxx00x : PLL M = 1
    // xxxxxxx1 : PLL Enable
    PLLCR = 0x51; // Enable PLL

    // SCCR = 0,00,1,1,0,10
    // 0xxxxxxx : Stop Mode = Mode 2
    // x00xxxxx : Clock Divide 1
    // xxx1xxxx : Clock Change By Software
    // xxxx1xxx : RC Oscillator Disable
    // xxxxx0xx : XTAL Oscillator Enable
    // xxxxxx10 : System Clock Source = 32.768KHz
    SCCR = 0x1A; // Run XTAL-32.768KHz+PLL
    while((PLLCR & 0x80) != 0x80); // Wait PLL Output Lock
    /* Now System Clock = 14.7456MHz */
    .
    .
    .
}
```

It shows an example program for setting MCU to run from Circuit Sub Clock Oscillator + PLL.

How to use Internal RC Oscillator

It is good alternative to choose and use Signal Clock Generator from Internal RC Oscillator. There are several strong points when setting MCU to run by this Signal Clock from Internal RC Oscillator; for example, it saves energy and devices; so, user gets more 4 Pins because it does not lose any Pin I/O for interfacing with external Module Crystal Oscillator. This internal Signal Clock is quite accurate because the inconsistency is only +/-2%; so, it is enough and suitable for general applications. User can choose to set the suitable divider to find out the frequency from this Circuit Frequency Generator of Internal RC Oscillator; and finally, user will get the preferable Frequency as 2 MHz, 4MHz, 8MHz or 16MHz from the commands in the program.



It shows how to choose and use Signal Clock from Circuit Internal RC Oscillator.

```
#include <z51f6412.h> // Z8051 : Z51F6412

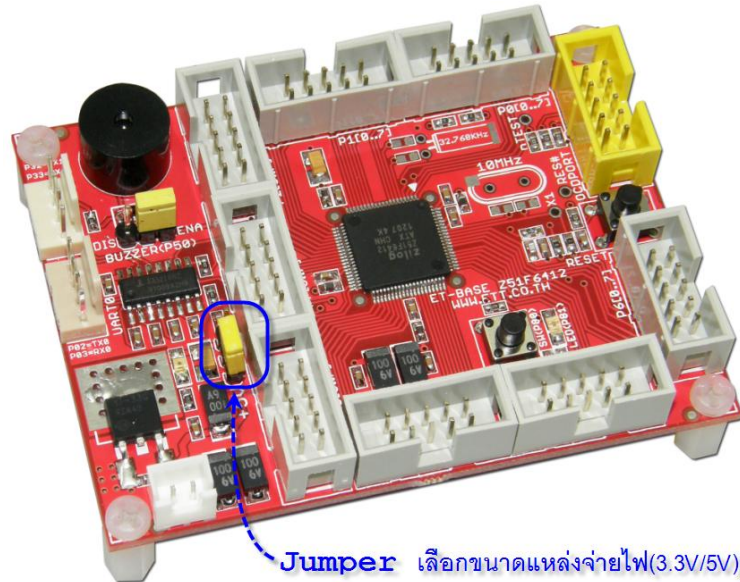
void main(void)
{
    /* Config System Clock = Internal RC 8.00 MHz */
    // PLLCR = 0,0,0,00,00,0
    // 0xxxxxxx : PLL Output Status
    // x0xxxxxxx : PLL Output Bypass
    // xx0xxxxxx : Power PLL = Default
    // xxx00xxx : FBDiv = Default
    // xxxxx00x : PLL M = Default
    // xxxxxxx0 : PLL Disable
    PLLCR = 0x00; // Disable PLL

    // SCCR = 0,01,0,0,1,00
    // 0xxxxxxx : Stop Mode = Mode 2
    // x01xxxxxx : INTRC Clock Divide = INTRC(16MHz)/2 = 8MHz
    // xxx0xxxxx : Clock Change By Hardware
    // xxxx0xxx : RC Oscillator Enable
    // xxxxx1xx : XTAL Oscillator Disable
    // xxxxxx00 : System Clock Source = INTRC(16MHz)
    SCCR = 0x24; // INT-RC 8MHz
    /* Now System Clock = 8.00MHz */
    .
    .
    .
}
```

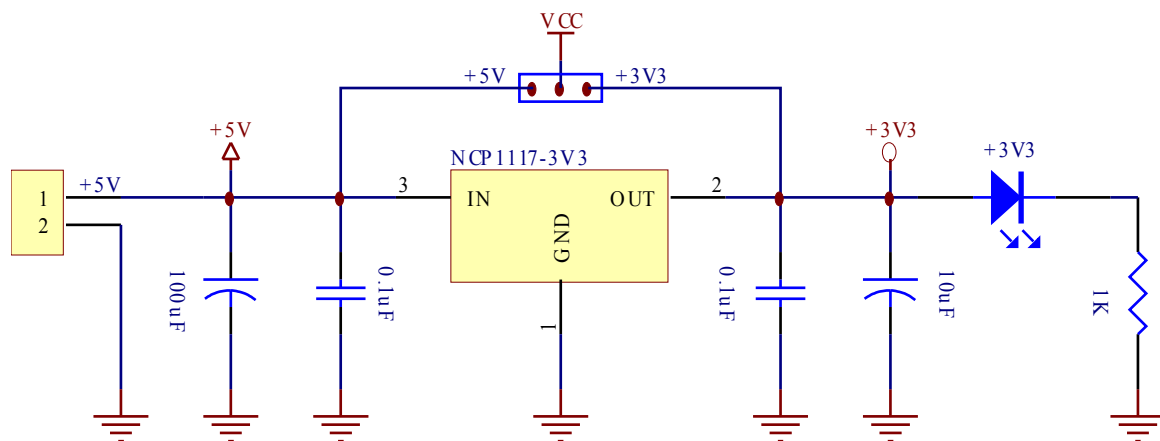
It shows an example program for setting MCU to run by Internal RC with 8.00 MHz.

Circuit Power Supply

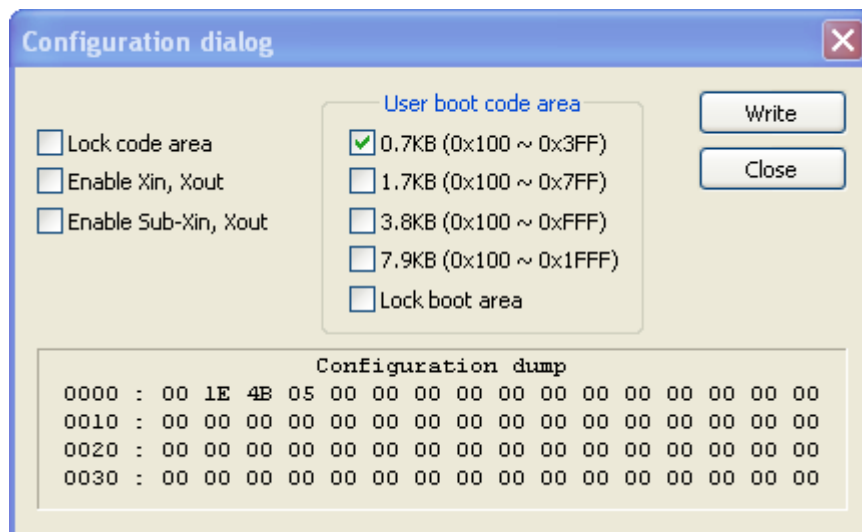
Circuit Power Supply of this board is compatible with +5VDC from external board. There is Circuit Regulate 3.3V/1A internal board with Jumper to set and choose the preferable voltage levels between +5VDC or +3V3; it is used to be Voltage of MCU (+VCC) and Circuit I/O internal board.



The method to choose and set the voltage level +VCC for MCU internal board is to set Jumper 3V3/+5V; in this case, user should consider the main objective and the connected device, and then user needs to choose the suitable +VCC. This MCU No.Z51F6412 can operate well with the voltage in the range of 2.0V to 5.5V; there are 2 voltage levels that can be choose on board that are +5V and +3.3V. Remember, when user has chosen any voltage level in any range of Power Supply for board, the level of Signal Logic that is used to connect with external device should not be higher than voltage level of Power Supply. For example, if using Power Supply 3.3V, the signal that is connected must be 3.3V; but, if it is connected with Signal 5V, the MCU maybe damaged. So, user should be careful and choose the correct voltage level for MCU.



Fuse Configuration of Z51F6412



Z51F6412 has special Register called "**FUSE_CONF**"; it is to set the operation of MCU. It sets the value of this Fuse in the process of programming Code by OCD Device that is Debug and ISP and Parallel Programming. This Register is 8BIT; the initial value internal MCU that is standard value from factory is 0x00. The meanings of Register "**FUSE_CONF**" are listed below;

D7	D6	D5	D4	D3	D2	D1	D0
BSIZE1	BSIZE0	SXIENA	XINENA	-	OCDSEL	LOCKB	LOCKF

- **BZSIE1:BSIZE0**: It is used together to setup Boot Code Size; in this case, there are 4 sizes as follows;
 - **[0:0]** = 768 Byte (1KByte - 256Byte : Boot Code Address 0x00FF...0x03FF)
 - **[0:1]** = 1792 Byte (2KByte - 256Byte : Boot Code Address 0x00FF...0x07FF)
 - **[1:0]** = 3840 Byte (4KByte - 256Byte : Boot Code Address 0x00FF...0x0FFF)
 - **[1:1]** = 7936 Byte (8KByte - 256Byte : Boot Code Address 0x00FF...0x1FFF)
- **SXIENA**: It is used to setup the operation of Sub Clock Oscillator that is Circuit Clock Generator 32.768KHz from external; it is interfaced with P0.4 and P0.5. The Default value from factory is set as **Disable**.
 - **0** : Sub Clock Oscillator Disable
 - **1** : Sub Clock Oscillator Enable
- **XINENA**: It is used to setup the operation of Main Clock Oscillator that is Circuit Clock Generator from external; it is in the range of 1 MHz - 10 MHz. It is interfaced with P6.2

and P6.3; in this case, the Default value from factory is set as **Disable**.

- **0** : Main Clock Oscillator Disable
- **1** : Main Clock Oscillator Enable
- **OCDSSEL**: It is used to setup the operation of circuit to eliminate noise at Pin OCD (Select Noise Cancelling of OCD Pin).
 - **0**: Signal OCD is filtered by Circuit that filters noise 10nS.
 - **1**: Signal OCD Sync and Signal Clock is internal INTRC Clock.
- **LOCKE**: It is used to setup value to protect the memory in the part of Boot Code from erasing, it accords to size that is set by BSIZE[1:0].
 - **0** : Boot Lock Disable
 - **1**: Boot Lock Enable; it protects the memory in the part of Boot Code from erasing.
- **LOCKF**: It is used to setup format to protect the Memory Code. When Enable, this Fuse Bit is "1" and it cannot read any data from the Memory Code of MCU by external Programmer. This Fuse Bit can be cancelled by erasing all data in the memory only.
 - **0** : Lock Disable
 - **1** : Lock Enable(Protect device from reading)

*****NOTE*****

For Debugger Mode, user can edit value of this Register temporarily, without actually programming **Config**; it temporarily tests the operational result of MCU. Except, the value of Bit "**LOCKF**", it cannot be temporarily edited or changed by Debugger device. When MCU exits from this Debugger Mode, the specifications of MCU are normal **Config**.

How to connect MCU with ET-Z8051 OCD in Debug Mode

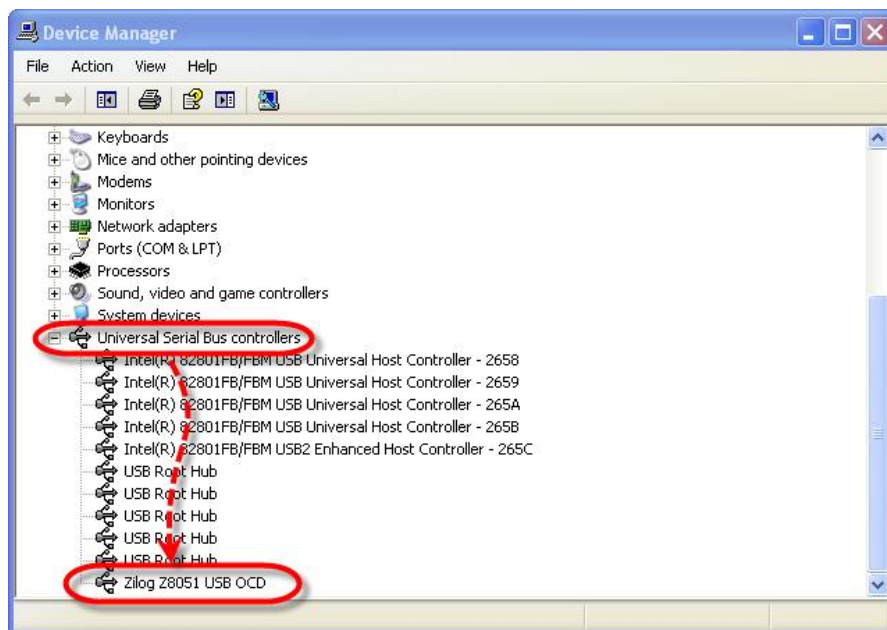


Picture shows how to interface ET-BASE Z51F6412 with ET-Z8051 OCD.

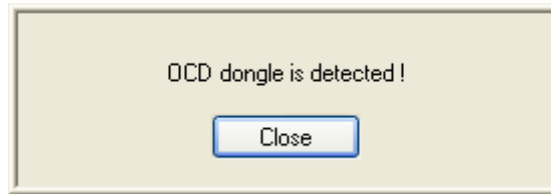
This operation mode is suitable for doing in the process of developing program because user can program, control, test, and stop the operation of MCU independently; moreover, user can check and modify Registers while MCU is running, without erasing and programming MCU frequently. This operation mode is compatible with Program Zilog Z8051 OCD. Nowadays (August, 2012), this program is updated to version Zilog Z8051 OCD Version 1.147. This is the procedure for using board briefly;

1. Install Program **Zilog Z8051 OCD** completely, and then run File name "**Z8051_1.1.exe**".
2. Interface USB Cable of ET-Z8051 OCD with computer PC and then install Driver of the device completely; user does this step only one time. The device "ET-Z8051 OCD of ETT uses the same Driver as Zilog Z8051 OCD of Zilog Inc. Normally, this Driver has already been installed completely with Program "**Z8051_1.1.exe**" in step No.1; in this case, if it installs the program according to the standard value, the Driver is in the location "**\device drivers\OCD USB**" under the Directory that installs the program. Nowadays, the Driver supports the Operating System of Windows for computer that is both 32BIT and 64BIT as follows;
 - a. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x32" for 32Bit
 - b. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x64" for 64Bit

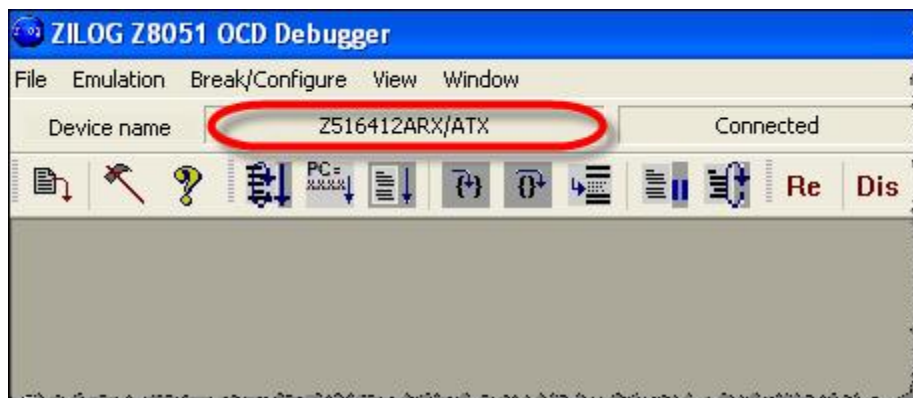
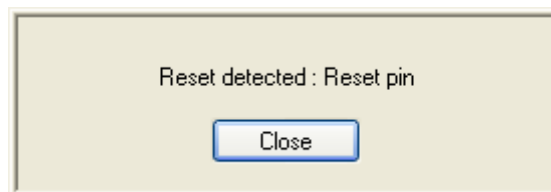
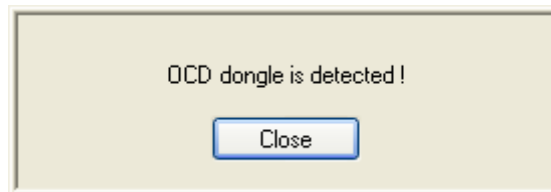
If everything is correct, red LED USB on ET-Z8051 OCD is ON; moreover, when user enters **Device Manager**, user can see **USB Device** in Tab **Universal Serial Bus Controller**; in this case, it found the device called "**Zilog Z8051 USB OCD**" as shown in the example below;



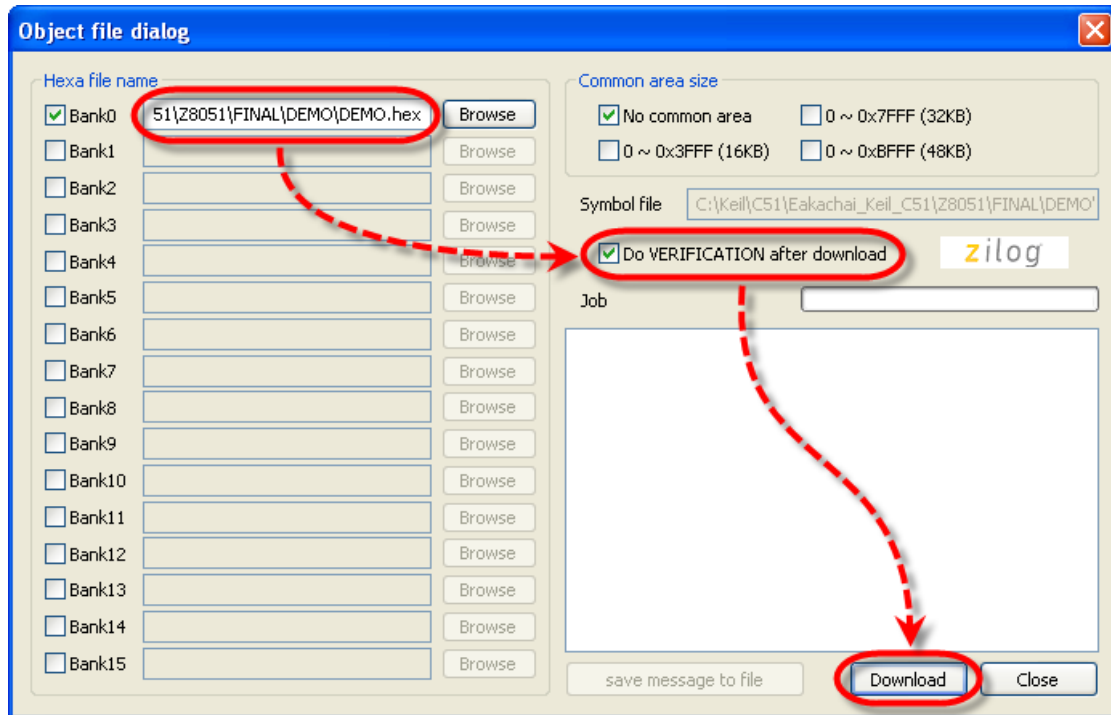
3. Run Program **Zilog Z8051 OCD V1.147**. When this program first starts running, it displays message Dialog to notify user to know that it found the Z8051 OCD Device as shown in the example below. Moreover, the OCD Device of ET-Z8051 OCD shows the green LED LINK that is lit up; it means that the OCD Device is ready to run.



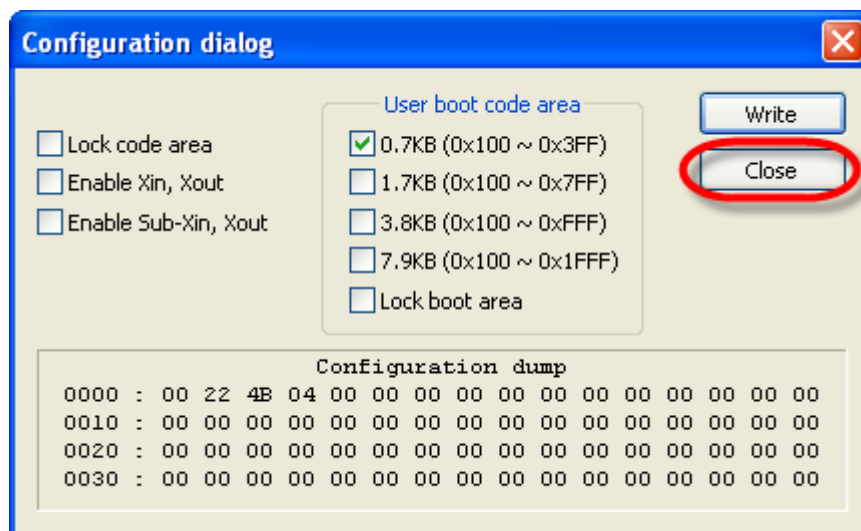
4. If it has provided Power Supply for Board ET-BASE Z51F6412 completely, it should remove Cable Power Supply of Board ET-BASE Z51F6412 from board first. Next, it should interface 10Pin Pair Cable between ET-Z8051 OCD with yellow Connector IDE 10Pin (PORT-OCD) of Board ET-BASE Z51F6412 completely; and finally, it should provide Power Supply into Board ET-BASE Z51F6412. If everything is correct, all 3 LEDs of ET-Z8051 OCD are lit up (ON) that are USB LINK and TARGET; moreover, it also displays 2 message Dialogs respectively and it shows number "**Z516412ARX/ATX**" in the **Device name** as shown in the example below;



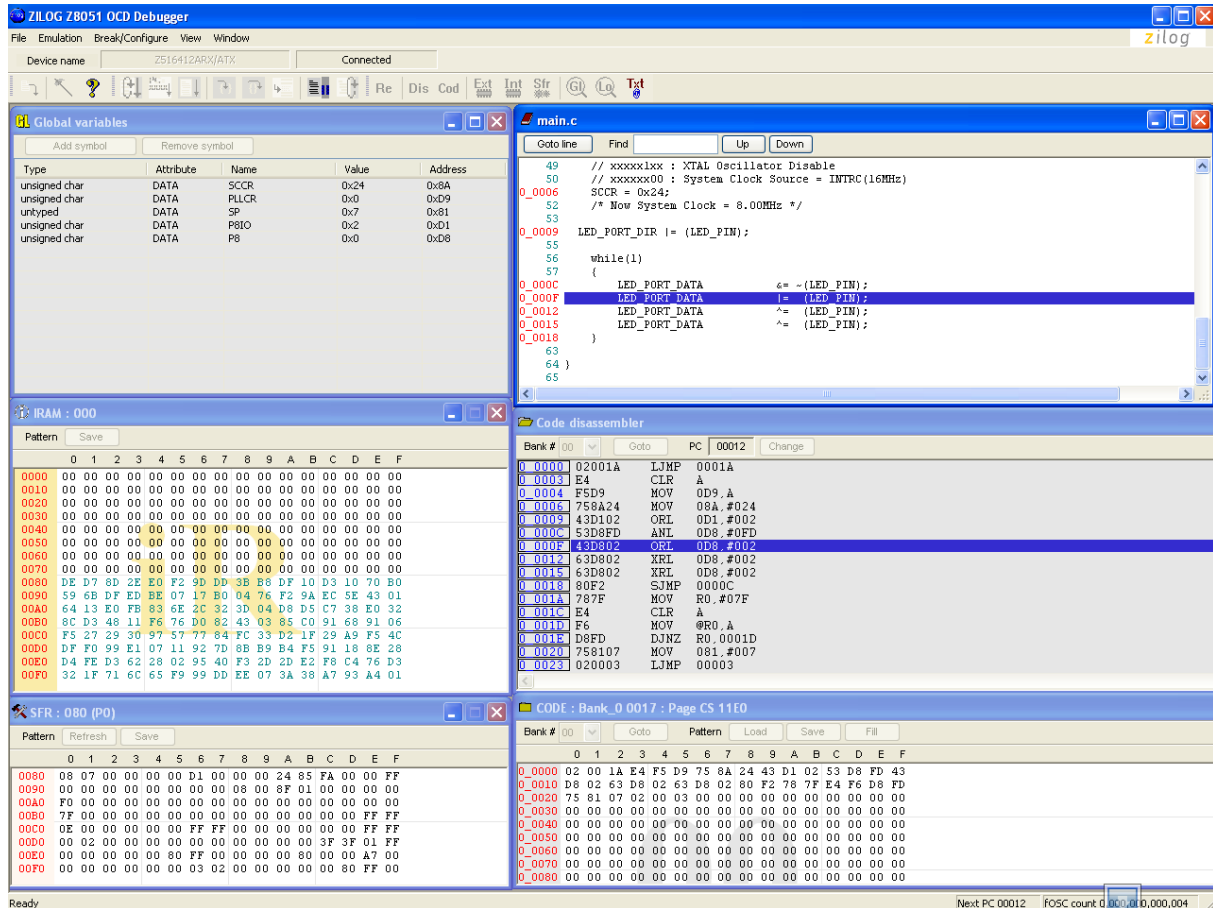
5. Now, it shows the connection between MCU on Board ET-BASE Z51F6412 and OCD Device of ET-Z8051 OCD is correct; so, user can command Program Zilog Z8051 OCD to run as preferred. Next, user has to program the preferable HEX File to download into MCU on board; click Menu **File** → **Load HEX**, specifying the preferable file name and then download as shown in the example below;



In the part of **Configuration**, user can click Button **Close** instantly if there is no any change; however, if user has edited or changed any data in this **Configuration**, it has to setup value as desired and then choose **Write**. In this case, it chooses Button **Close** to setup **Configuration** values according to the Default value from factory as shown in the example.



6. After downloaded HEX File into MCU completely, user can command MCU to run as preferred; in this case, it may command MCU to RUN(GO) to see the actual operation; or it may set position to pause (BREAK); or, it may run step by step (STEP) to check if the operation of the designed program is correct. User can read further details in this part from User's Manual of ET-Z8051 OCD and Program Zilog Z8051 OCD by self.



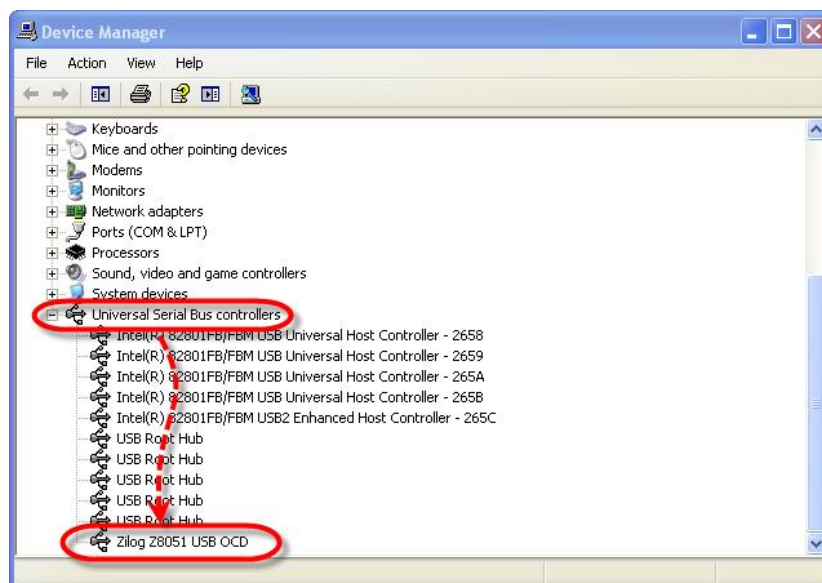
Picture shows an example of Debug the operation of MCU through Program Zilog Z8051 OCD v1.147.

How to connect MCU with ET-Z8051 OCD in ISP Programming Mode

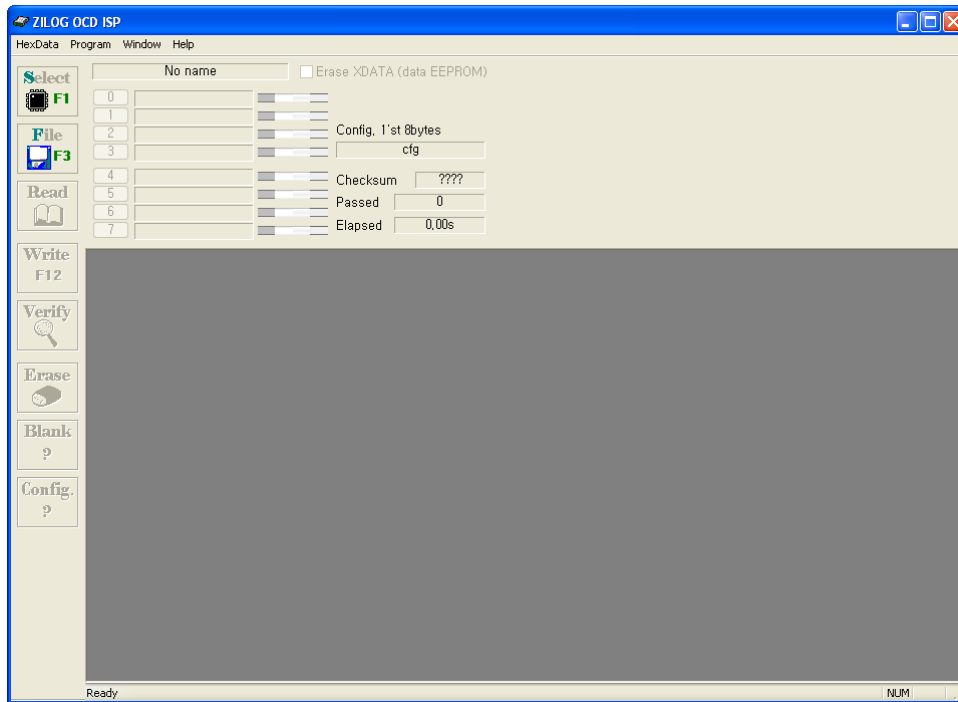
This operation mode is suitable for the operational format after developed the program completely. Function of this operation mode is only to Program Hex Code into MCU; it is suitable for programming HEX Code when manufacturing a lot of products because it can do quickly. This operation mode is compatible with Program "Zilog Z8051 ISP". Nowadays (August, 2012), this program is updated to version "Zilog Z8051 ISP Version 1.147". This is the procedure for using board briefly;

1. Install Program Zilog Z8051 ISP and run File name "Z8051_1.1.exe".
2. Interface USB Cable of ET-Z8051 OCD with computer PC and then install Driver of the device completely; user does this step only one time. The device "ET-Z8051 OCD of ETT uses the same Driver as Zilog Z8051 OCD of Zilog Inc. Normally, this Driver has already been installed completely with Program "Z8051_1.1.exe" in step No.1; in this case, if it installs the program according to the standard value, the Driver is in the location "..\device drivers\OCD USB\" under the Directory that installs the program. Nowadays, the Driver supports the Operating System of Windows for computer that is both 32BIT and 64BIT as follows;
 - a. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x32" for 32Bit
 - b. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x64" for 64Bit

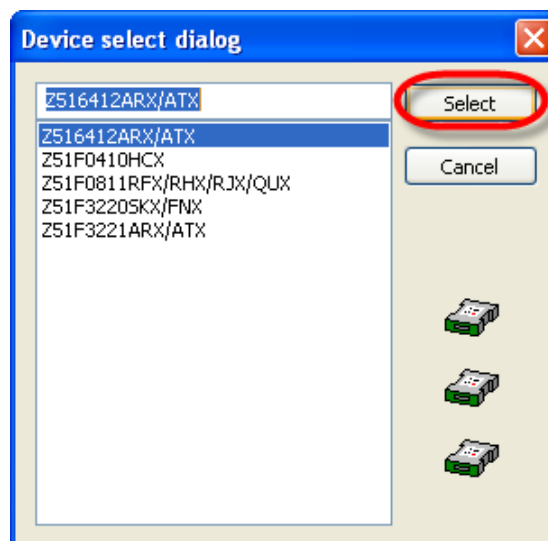
If everything is correct, red LED USB on ET-Z8051 OCD is lit up (ON); moreover, when user enters **Device manager**, it found **USB Device** in Tab **Universal Serial Bus Controller** called "Zilog Z8051 USB OCD" as shown in the example below;



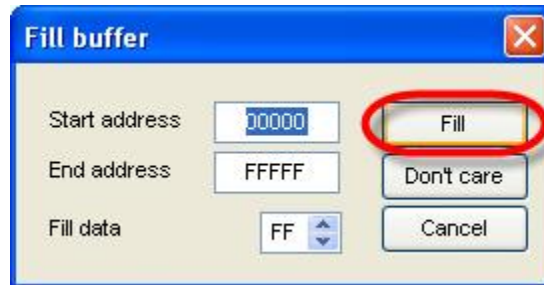
- Run Program "Zilog Z8051 ISP V1.147" completely. When the program first starts running, there is nothing on the screen of program as shown in the example below;



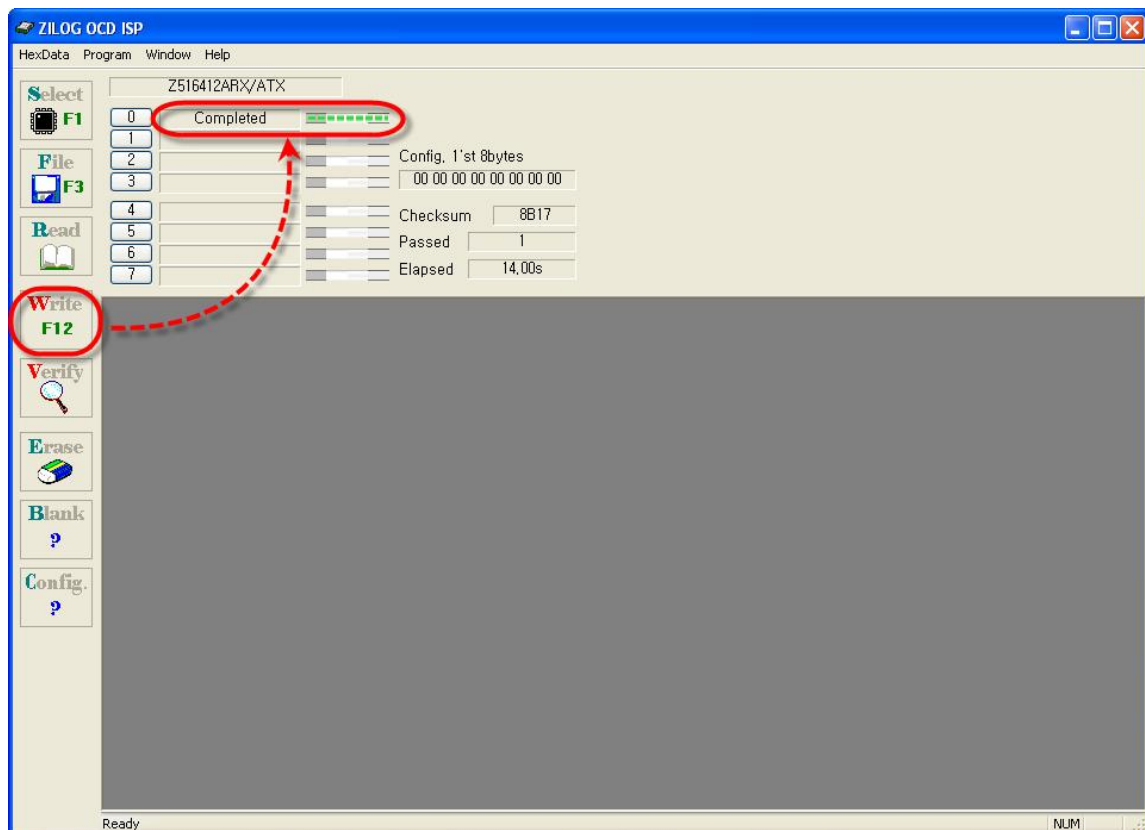
- If it has provided Power Supply for Board ET-BASE Z51F6412 completely, it should remove Cable Power Supply of Board ET-BASE Z51F6412 from board first. Next, it should interface 10Pin Pair Cable between ET-Z8051 OCD with yellow Connector IDE 10Pin (PORT-OCD) of Board ET-BASE Z51F6412 completely; and finally, it should provide Power Supply into Board ET-BASE Z51F6412. Next, it has to set MCU number; click "Select Device" to set the MCU number to be "Z51F6412ARX/ATX" as shown in the example below;



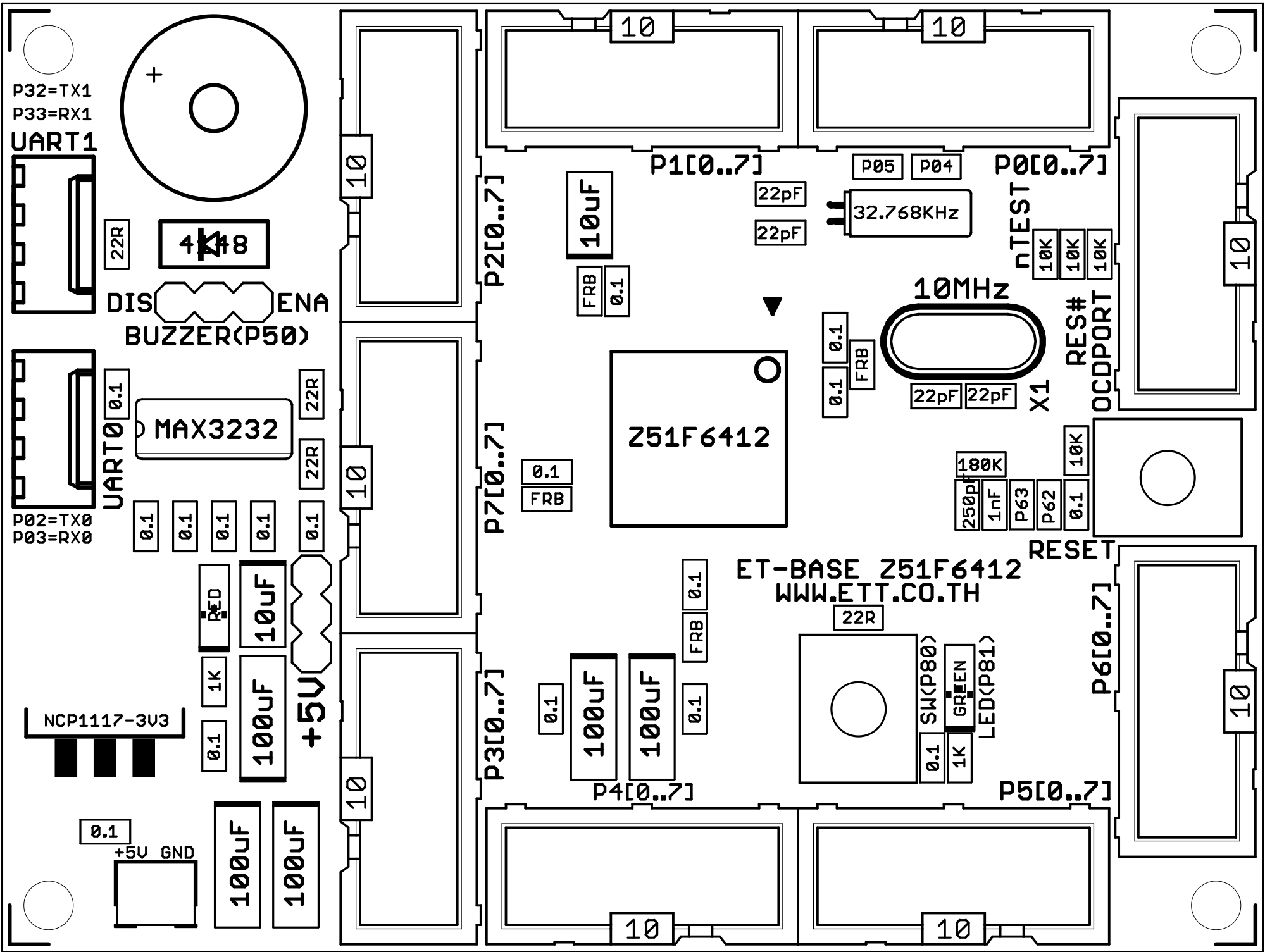
- When setup the MCU number completely, open the preferable HEX File to program into MCU. Choose "Load Code HEX File", the Dialog Box appears and user needs to setup the format of **Fill Buffer**; in this case, it sets **Fill** as **FF** as shown in the example. Next, it has to set name and location of the preferable HEX file as shown in the example;



- After opened the HEX File and kept it in Buffer of the program completely, choose "Write" to program Code in the Buffer into the memory internal MCU; in this case, user has to wait until the operation of program is complete. Next, it has to remove Cable OCD PORT from board and then remove Power Supply from board as well; it has to wait for a moment for 2-3 seconds, then user interface Power Supply to provide power into board again. Finally, MCU internal MCU starts running according to the designed program instantly.



It shows an example of Program MCU by OCD through Program Zilog Z8051 ISP v1.147.



ET-BASE Z51F6412
WWW.ETT.CO.TH

P32=TX1
P33=RX1
UART1

P02=TX0
P03=RX0
UART0

P05 P04 P0[0..7]
nTEST
RES#
OCDPORT

RESET

Z51F6412

DIS ENA
BUZZER(P50)

MAX3232

NCP1117-3V3

+5V GND

+5V

SW(P80)
LED(P81)

10

10

10

10

10

P3[0..7]

P7[0..7]

P2[0..7]

P4[0..7]

P1[0..7]

P5[0..7]

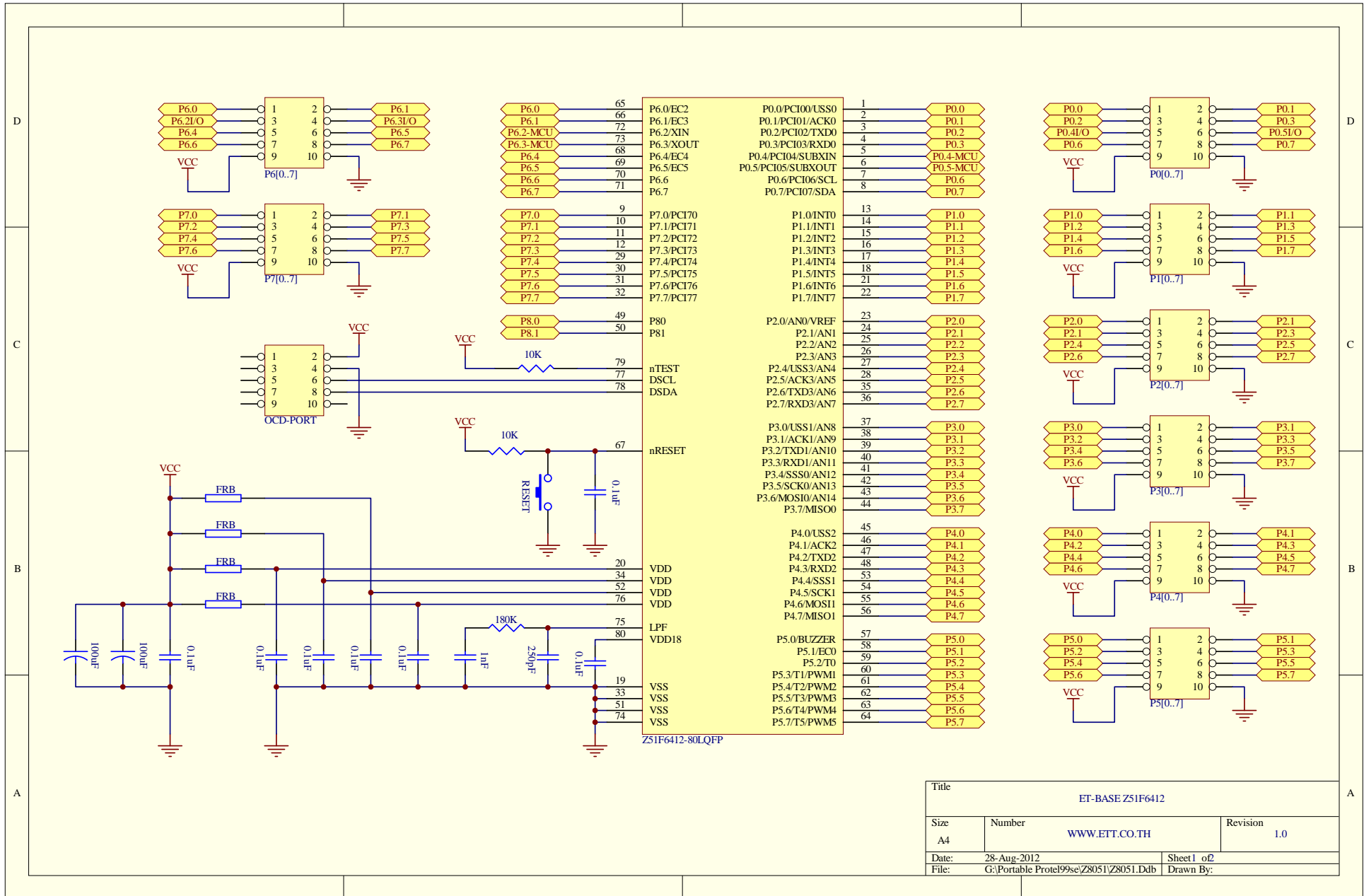
P6[0..7]

10

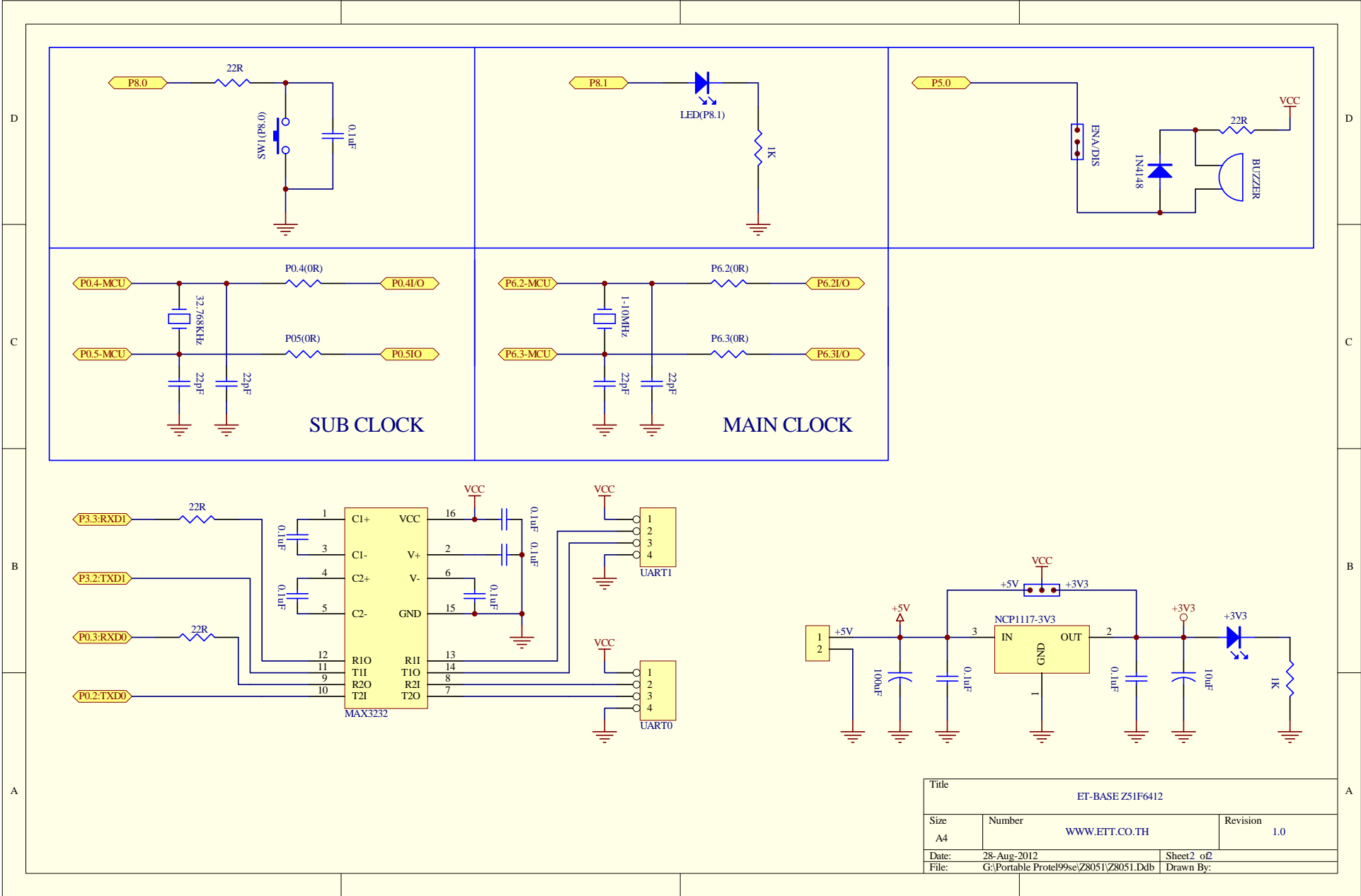
10

10

10



Title			
ET-BASE Z51F6412			
Size	Number	Revision	
A4	WWW.ETT.CO.TH	1.0	
Date:	28-Aug-2012	Sheet 1 of 2	
File:	G:\Portable Protel99se\Z8051\Z8051.Dcb	Drawn By:	



Title			
ET-BASE Z51F6412			
Size	Number	WWW.ETT.CO.TH	Revision
A4			1.0
Date:	28-Aug-2012	Sheet2 of2	
File:	G:\Portable Protel99se\Z8051\Z8051.Ddb	Drawn By:	