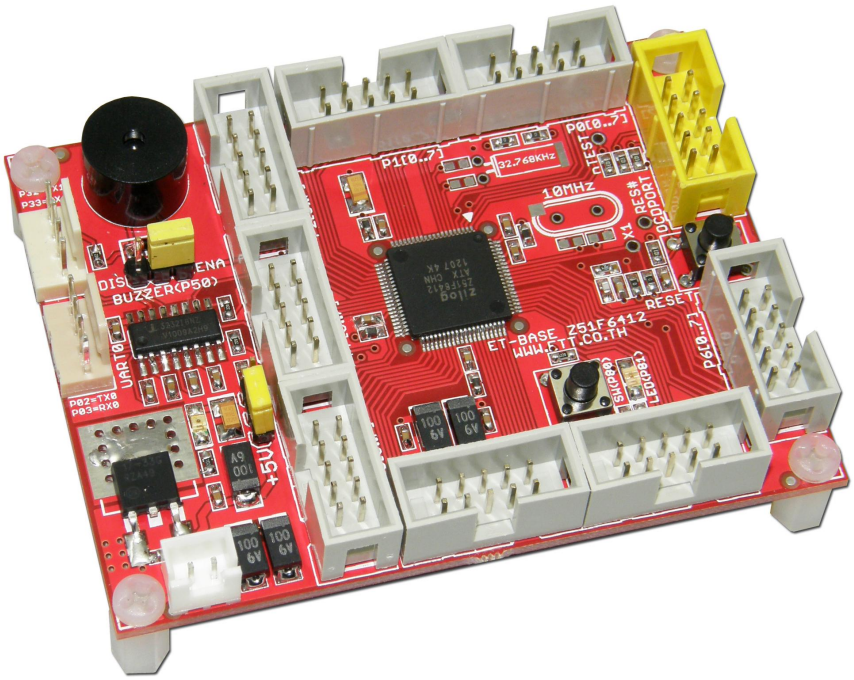
ET-BASE Z51F6412



ET-BASE Z51F6412 เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล MCS51 ใหม่ล่าสุดจาก Zilog Inc. ซึ่งเดิมเป็นผู้คิดค้นพัฒนาไมโครโปรเซสเซอร์ตระกูล Z80 อันโด่งดังและได้รับความนิยมอย่างแพร่หลายมาแล้วเมื่อหลายปีที่ผ่านมา โดยไมโครคอนโทรลเลอร์ตระกูล MCS51 ที่ทาง Zilog Inc. ทำการพัฒนาขึ้นล่าสุดนี้ ทาง Zilog ให้ชื่อว่าไมโครคอนโทรลเลอร์ตระกูล “Z8051”

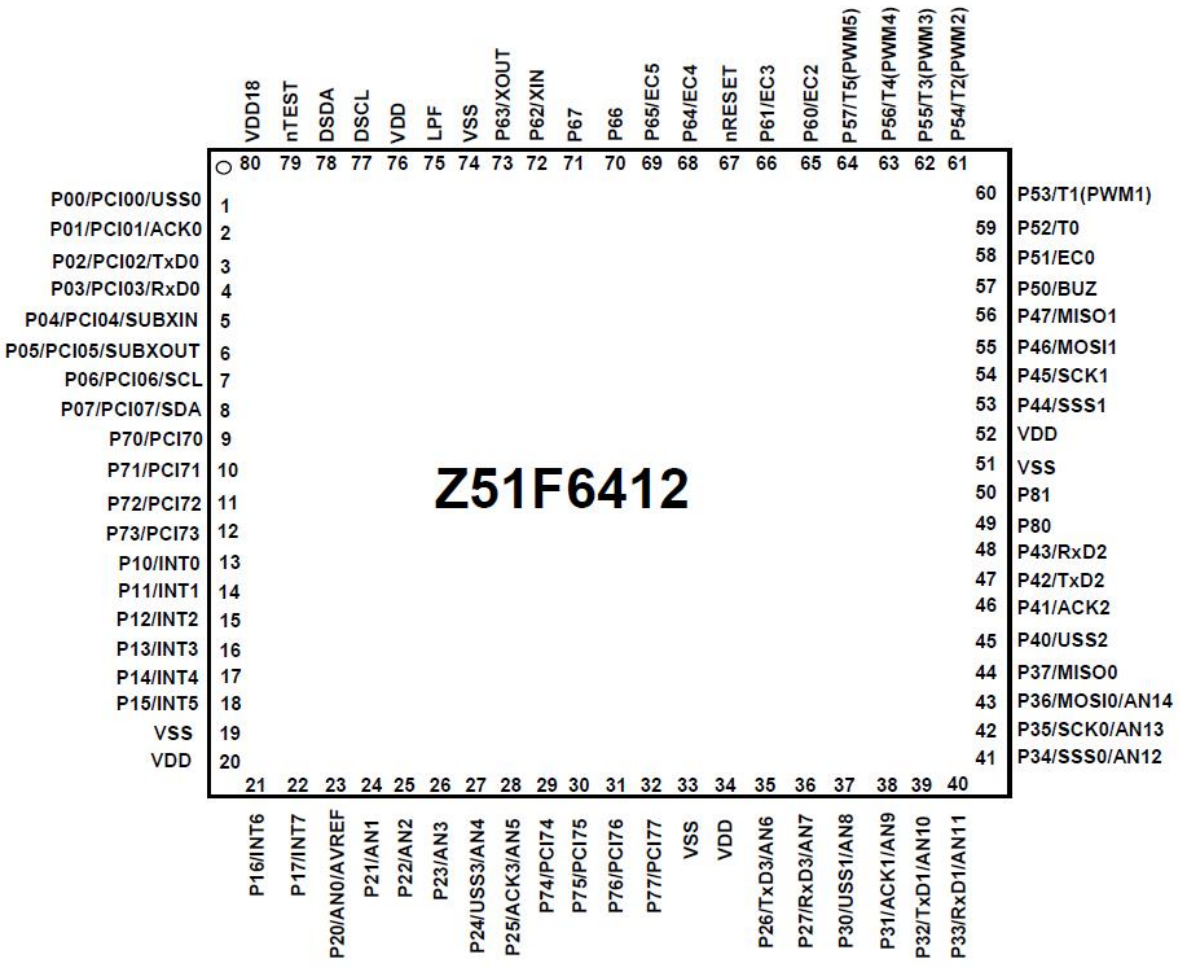
โดย Z8051 เป็นไมโครคอนโทรลเลอร์ชิพเดี่ยวแบบ CISC ขนาด 8 บิต ซึ่งใช้สถาปัตยกรรมการประมวลผลของ MCS51 ตามแบบฉบับของ Intel เหมือนเดิม แต่ทาง Zilog ได้มีการปรับเปลี่ยนระบบการทำงานของ Peripheral I/O ต่างๆใหม่หมด ทั้ง I/O Port, UART และ Timer/Counter เรียกว่า ยกเครื่อง Peripheral I/O กันใหม่หมด แต่ยังใช้วิธีการ สำหรับใช้ควบคุมและเข้าถึง Peripheral I/O ต่างๆผ่านทาง SFR (Special Function Register) เหมือน Intel เช่นเดิม ดังนั้นรูปแบบการเขียนโปรแกรมสั่งงานจึงเหมือน MCS51 เดิมจาก Intel ทุกประการ โดยทาง Zilog ได้ทำการปรับปรุงพัฒนาขีดความสามารถด้านต่างๆของ Z8051 ให้มีขีดความสามารถสูงขึ้นกว่า MCS51 มาตรฐานในหลายๆด้าน ทั้งด้านความเร็วการประมวลผล มีการเพิ่มหน่วยคำนวณคณิตศาสตร์สำหรับคูณและหารเลขจำนวนเต็ม 32 บิต และการจัดการด้านพลังงาน รวมทั้งได้บรรจุอุปกรณ์ Peripheral I/O พิเศษแบบต่างๆ เช่น GPIO Port, Pin Pull-Up, Pin Debounce, Pin Change Interrupt, USART, SPI, I2C, ADC, Timer/Counter/Capture/PWM ,Buzzer Control และความสามารถในการรองรับการ Interrupt ได้ดีกว่า MCS51 มาตรฐาน ทำให้การเขียนโปรแกรมควบคุมสั่งงานฮาร์ดแวร์ทำได้ง่ายมากขึ้น ลดความซับซ้อนในการพัฒนาโปรแกรมได้เป็นอย่างมาก

คุณสมบัติของบอร์ด

- ใช้ MCU ตระกูล Z8051(MCS51 จาก Zilog Inc.) เบอร์ Z51F6412 เป็น MCU ประจำบอร์ด Run ความถี่สูงสุด 16 MHz(ประมวลผล 125nS/1 Machine Cycle) จาก Internal Oscillator ภายใน
  - 64KByte Flash / 3KByte XRAM / 256 Byte IRAM
  - ใช้สถาปัตยกรรมการประมวลผลแบบ MCS51 (2 Clock / 1 Machine Cycle)
  - Internal Oscillator 16MHz(ผิดพลาดไม่เกิน +/-2%) ตั้งหาร 2,4,8,16 ได้จากโปรแกรม
  - 66 Bit GPIO Port สามารถโปรแกรมเป็น Peripheral I/O แบบต่างๆได้ เช่น
    - 15 Channel 12Bit ADC
    - 4 Channel UART
    - 2 Channel SPI
    - 1 Channel I2C
    - 2 Channel 8 Bit Timer/Counter(T0,T1) สามารถใช้รวมกันเป็น 16 Bit 1 ช่องได้
    - 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 ในทุกๆ Pin เลือก Enable/Disable ได้อิสระทุก Pin
    - Internal Pin Debounce ในทุกๆ Pin เลือก Enable/Disable ได้อิสระทุก Pin
    - 1 Channel Buzzer Drive
  - รองรับการ Interrupt จากอุปกรณ์ต่างๆ 32 แหล่ง 32 Vector
  - มีวงจร Calculator สำหรับคำนวณแบบคูณและหารเลขจำนวนเต็มขนาด 32 บิต
    - คูณเลขจำนวนเต็ม 16Bit x 16Bit โดยใช้เวลา 1 Cycle Clock
    - หารเลขจำนวนเต็ม 32Bit / 16Bit โดยใช้เวลา 32 Cycle Clock
  - Watch Timer และ Watch Dog Timer
  - Power-ON Reset ที่ 1.4V
  - Programmable Brown-Out Detect(1.6V, 2.5V, 3.6V และ 4.2V)
- มี Crystal ความถี่ 10 MHz (Option) สำหรับงานที่ต้องการความแม่นยำเที่ยงตรงสูงๆ
- มี Crystal ความถี่ 32 KHz (Option) + PLL สามารถกำหนดคูณความถี่เป็น 14.75MHz ได้
- มีวงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม UART แบบ RS232 จำนวน 2 ช่อง โดยใช้ขั้วต่อ UART แบบ CPA-4 Pin มาตรฐาน อีทีที
  - 1 ช่อง สำหรับ Hardware UART0 โดยใช้ Pin P0.2(TX0) และ P0.3(RX0)

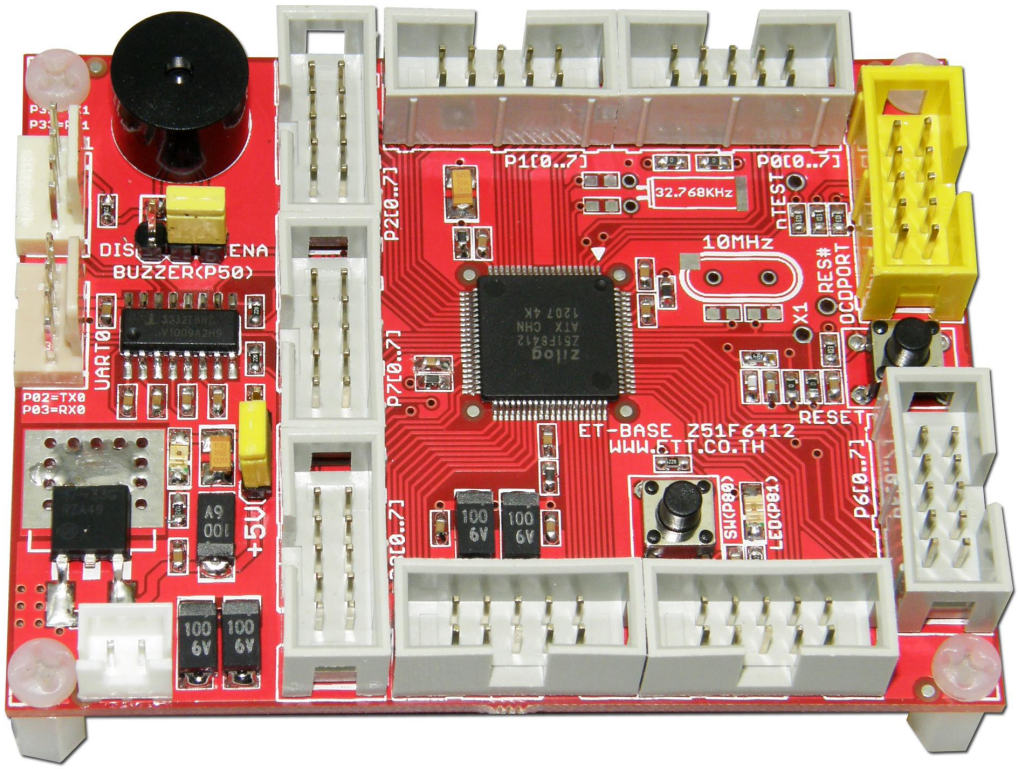
- 1 ช่อง สำหรับ Hardware UART1 โดยใช้ Pin P3.2(TX1) และ P3.3(RX1)

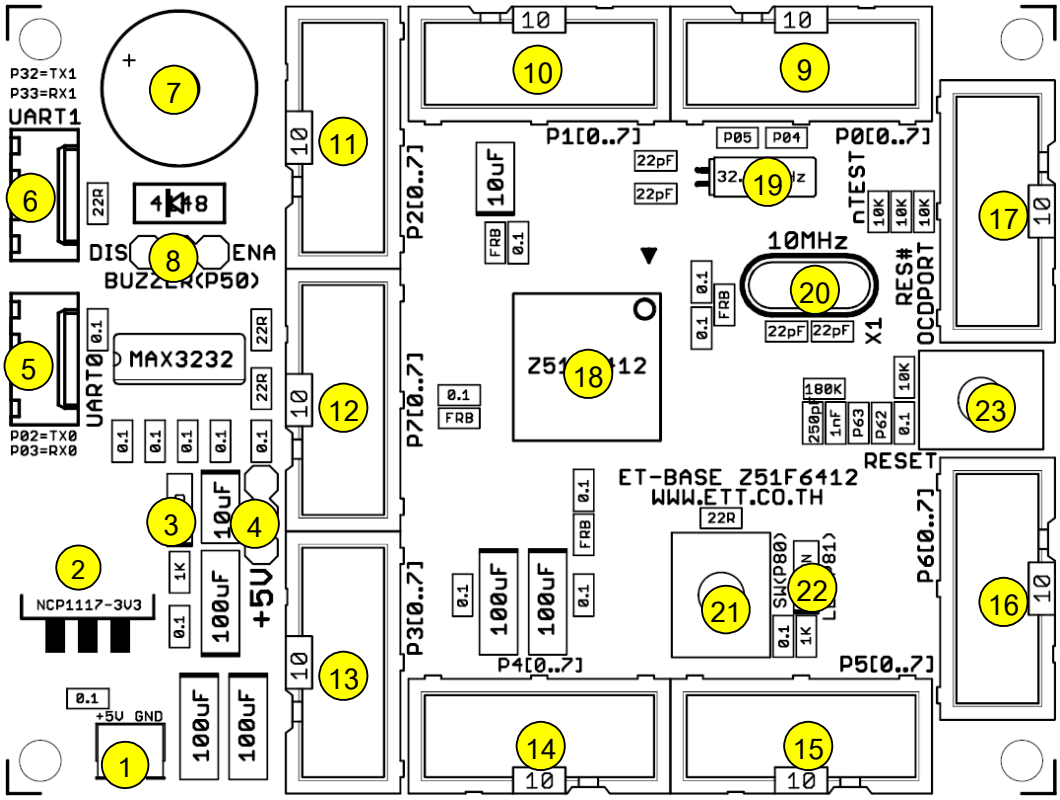
- มี Buzzer พร้อม Jumper ตัดต่อสัญญาณ Buzzer Drive(P5.0)
- มี SW แบบกดติดปล่อยดับ สำหรับทดสอบการทำงาน Input จำนวน 1 SW โดยใช้ P8.0
- มี LED แสดงสถานะ สำหรับทดสอบการทำงาน Output จำนวน 1 ดวง โดยใช้ P8.1
- มี SW Reset พร้อมวงจร RC-Reset
- มีขั้วต่อสัญญาณ I/O แบบ Header ขนาด 2x5 จำนวน 8 ชุด(P0,P1,P2,P3,P4,P5,P6,P7)
- มีขั้ว OCD-PORT แบบ 10Pin IDE มาตรฐาน Zilog สำหรับใช้ร่วมกับชุดพัฒนาโปรแกรมและ Debugger มาตรฐาน Z8051 On-Chip-Debugger / Programmer(ET-Z8051 OCD)
- Power +5VDC Input พร้อม Regulate แบบ 3.3V/1A และ LED แสดงสถานะแหล่งจ่าย Power Jumper สำหรับ เลือกแหล่งจ่ายไฟเลี้ยงให้ MCU ว่าจะใช้เป็น +5VDC หรือ 3.3VDC
- ขนาด PCB Size เล็กเพียง 8 x 6 cm.



โครงสร้างการจัด Pin ของ Z51F6412 แบบ 80-LQFP

โครงสร้างบอร์ด ET-BASE Z51F6412



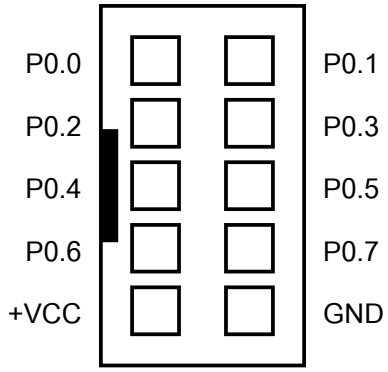


รูปแสดง โครงสร้างของบอร์ด ET-BASE Z51F6412

- **หมายเลข 1** คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟ +5VDC
- **หมายเลข 2** คือ IC Regulate ขนาด 3.3VDC/1A
- **หมายเลข 3** คือ LED แสดงสถานะของแหล่งจ่ายไฟ +VCC
- **หมายเลข 4** คือ Jumper สำหรับเลือกขนาดแรงดันไฟเลี้ยง MCU(+VCC) ระหว่าง 3.3V หรือ 5V
- **หมายเลข 5** คือ ขั้วต่อ UART0 โดยเป็นสัญญาณแบบ RS232 รองรับ Hardware UART0 ซึ่งใช้ Pin ของ P0.2(TX0) และ P0.3(RX0) เป็นสัญญาณเชื่อมต่อ
- **หมายเลข 6** คือ ขั้วต่อ UART1 โดยเป็นสัญญาณแบบ RS232 รองรับ Hardware UART1 ซึ่งใช้ Pin ของ P3.2(TX1),P3.3(RX1) เป็นสัญญาณเชื่อมต่อ
- **หมายเลข 7** คือ Buzzer กำเนิดเสียง
- **หมายเลข 8** คือ Jumper สำหรับเลือกตัดต่อสัญญาณ P5.0 กับ Buzzer
- **หมายเลข 9** คือ ขั้วต่อ IDE10Pin ของ P0[0..7]
- **หมายเลข 10** คือ ขั้วต่อ IDE10Pin ของ P1[0..7]
- **หมายเลข 11** คือ ขั้วต่อ IDE10Pin ของ P2[0..7]
- **หมายเลข 12** คือ ขั้วต่อ IDE10Pin ของ P7[0..7]
- **หมายเลข 13** คือ ขั้วต่อ IDE10Pin ของ P3[0..7]
- **หมายเลข 14** คือ ขั้วต่อ IDE10Pin ของ P4[0..7]
- **หมายเลข 15** คือ ขั้วต่อ IDE10Pin ของ P5[0..7]
- **หมายเลข 16** คือ ขั้วต่อ IDE10Pin ของ P6[0..7]
- **หมายเลข 17** คือ ขั้วต่อ ของ OCD-PORT(On-Chip-Debug) สำหรับเชื่อมต่อกับเครื่องมือ OCD
- **หมายเลข 18** คือ MCU ประจำบอร์ด เบอร์ Z51F6412 (80Pin LQFP Package)
- **หมายเลข 19** คือ Sub Crystal ค่าความถี่ 32.768KHz(Option) ซึ่งถ้าต้องการติดตั้งใช้งาน Crystal ชุดนี้จะต้องสงวน Pin P0.4(SUBXIN) และ P0.5(SUBXOUT) สำหรับใช้เชื่อมต่อด้วย
- **หมายเลข 20** คือ Main Crystal ค่าความถี่ 10.00MHz(Option) ซึ่งถ้าต้องการติดตั้งใช้งาน Crystal ชุดนี้จะต้องสงวน Pin P6.2(XIN) และ P6.3(XOUT) สำหรับใช้เชื่อมต่อด้วย
- **หมายเลข 21** คือ Switch(SW1) กดติดปล่อยดับ สำหรับใช้ทดสอบการทำงานของ Digital Input โดย SW นี้จะเชื่อมต่อกับขาสัญญาณจาก Port P8.0
- **หมายเลข 22** คือ LED สำหรับใช้ทดสอบการทำงานของ Digital Output โดย LED นี้จะเชื่อมต่อกับขาสัญญาณจาก Port P8.1
- **หมายเลข 23** คือ สวิตช์ Reset สำหรับ Reset การทำงานของ MCU

ขั้วต่อสัญญาณต่าง ๆ

PORT P0[0..7]



PORT-P0[0...7]

PORT-P0[0..7] เป็นสัญญาณจาก Port P0 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยจะมี P0.2 และ P0.3 ซึ่งถูกเชื่อมต่อร่วมกับวงจร Line Driver ของ RS232(UART0) ด้วยแล้ว ซึ่งถ้ามีการใช้งานเชื่อมต่อกับ RS232 ของ UART0 ด้วยจะไม่สามารถใช้งานขาสัญญาณ P0.2 และ P0.3 เป็น I/O ได้อีก

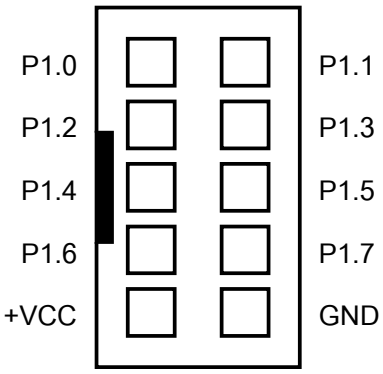
โดยสัญญาณจาก Port P0 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P0[7..0])
- ใช้งานเป็นขาตรวจจับการเปลี่ยนแปลง Input (Input Pin Change Interrupt : PCI0[7..0])
- ใช้งานเป็น UART0 Function(P0[3..0])
- ใช้เชื่อมต่อกับ Sub Crystal Oscillator ค่า 32.768KHz
- ใช้งานเป็น 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	-	-	-	-	-	-

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่าง ๆของ Port P0

PORT P1[0…7]



PORT-P1[0..7]

PORT-P1[0..7] เป็นสัญญาณจาก Port P1 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P1 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

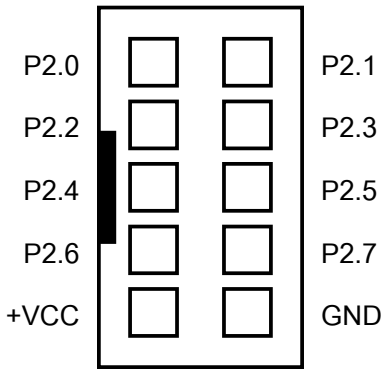
โดยสัญญาณจาก Port P1 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P1[7..0])
- ใช้งานเป็นขาตรวจจับการ Interrupt จากภายนอก (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

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P1

PORT-R2[0...7]



PORT-P2[0..7]

PORT-P2[0..7] เป็นสัญญาณจาก Port P2 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P2 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

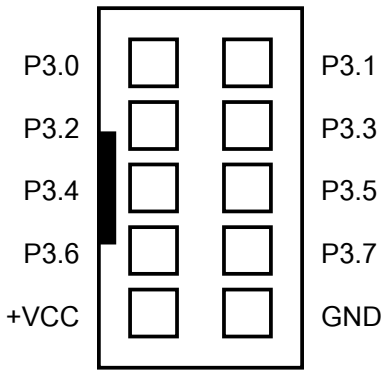
โดยสัญญาณจาก Port P2 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P2[7..0])
- ใช้งานเป็นขา Input Analog ADC ขนาดความละเอียด 12บิต (AN[7..0])
- ใช้งานเป็น 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	-	-	-	-

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P2

PORT P3[0…7]



PORT-P3[0..7] เป็นสัญญาณจาก Port P3 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยจะมี P3.2 และ P3.3 ซึ่งถูกเชื่อมต่อร่วมกับวงจร Line Driver ของ RS232(UART1) ด้วยแล้ว ซึ่งถ้ามีการใช้งานเชื่อมต่อกับ RS232 ของ UART1 ด้วยจะไม่สามารถใช้งานขาสัญญาณ P3.2 และ P3.3 เป็น I/O ได้อีก

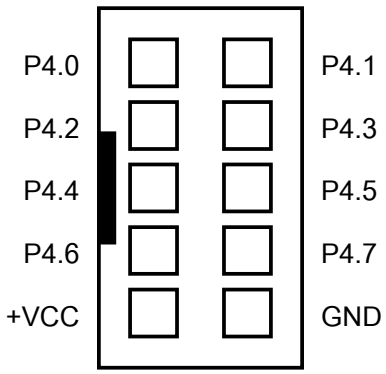
โดยสัญญาณจาก Port P3 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P3[7..0])
- ใช้งานเป็นขา Input Analog ADC ขนาดความละเอียด 12บิต (AN[14..8])
- ใช้งานเป็น UART1 Function(P3[3..0])
- ใช้งานเป็น 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	MOSI0	SCK0	SSS0	-	-	-	-

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P3

PORT P4[0…7]



PORT-P4[0..7]

PORT-P4[0..7] เป็นสัญญาณจาก Port P4 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P4 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

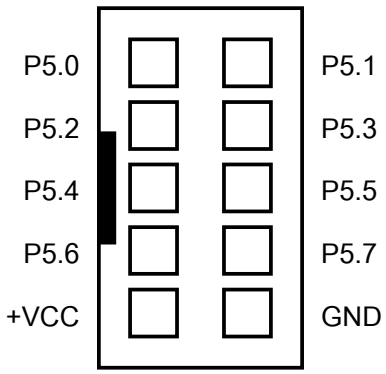
โดยสัญญาณจาก Port P4 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P4[7..0])
- ใช้งานเป็น UART2 Function(P4[3..0])
- ใช้งานเป็น 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	-	-	-	-

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P4

PORT P5[0…7]



PORT-P5[0..7]

PORT-P5[0..7] เป็นสัญญาณจาก Port P5 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P5 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

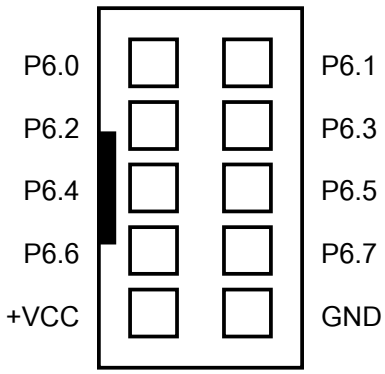
โดยสัญญาณจาก Port P5 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P5[7..0])
- ใช้งานเป็น Buzzer Drive(P5[0])
- ใช้งานเป็น PWM Function(P5[7..3])
- ใช้งานเป็น 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	-

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P5

PORT P6[0…7]



PORT-P6[0..7]

PORT-P6[0..7] เป็นสัญญาณจาก Port P6 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P6 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

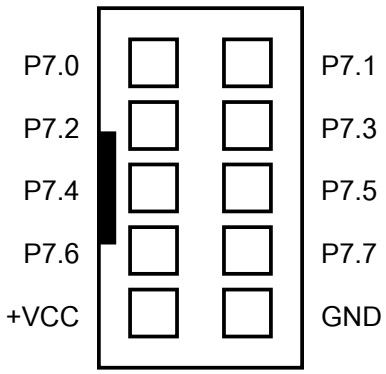
โดยสัญญาณจาก Port P6 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้ง GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P6[7..0])
- ใช้งานเป็น Main Crystal Oscillator(P6[3..2])
- ใช้งานเป็น 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

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P6

PORT P7[0…7]



PORT-P7[0..7]

PORT-P7[0..7] เป็นสัญญาณจาก Port P7 ของ MCU ซึ่งมีทั้งหมด 8 บิต โดยขาสัญญาณของ P7 นี้จะปล่อยว่างอิสระไว้สำหรับให้ผู้ใช้งานเลือกใช้ได้โดยอิสระทั้ง 8 บิต

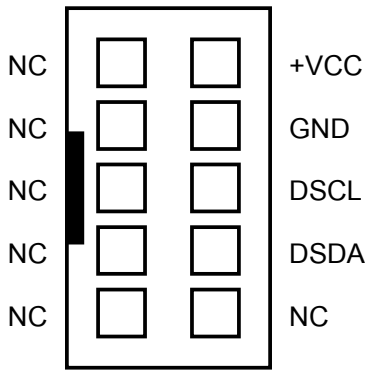
โดยสัญญาณจาก Port P7 ของ Z51F6412 สามารถโปรแกรมหน้าที่การใช้งานได้หลายหน้าที่ทั้งแบบ GPIO Input/Output ปรกติ และใช้งานร่วมกับฟังก์ชั่นพิเศษต่างๆ ดังนี้

- ใช้งานเป็น GPIO Input/Output Port(P7[7..0])
- ใช้งานเป็นขาตรวจจับการเปลี่ยนแปลง Input (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

ตาราง แสดงหน้าที่การใช้งาน Pin Port ต่างๆของ Port P7

PORT OCD(On-Chip Debug)

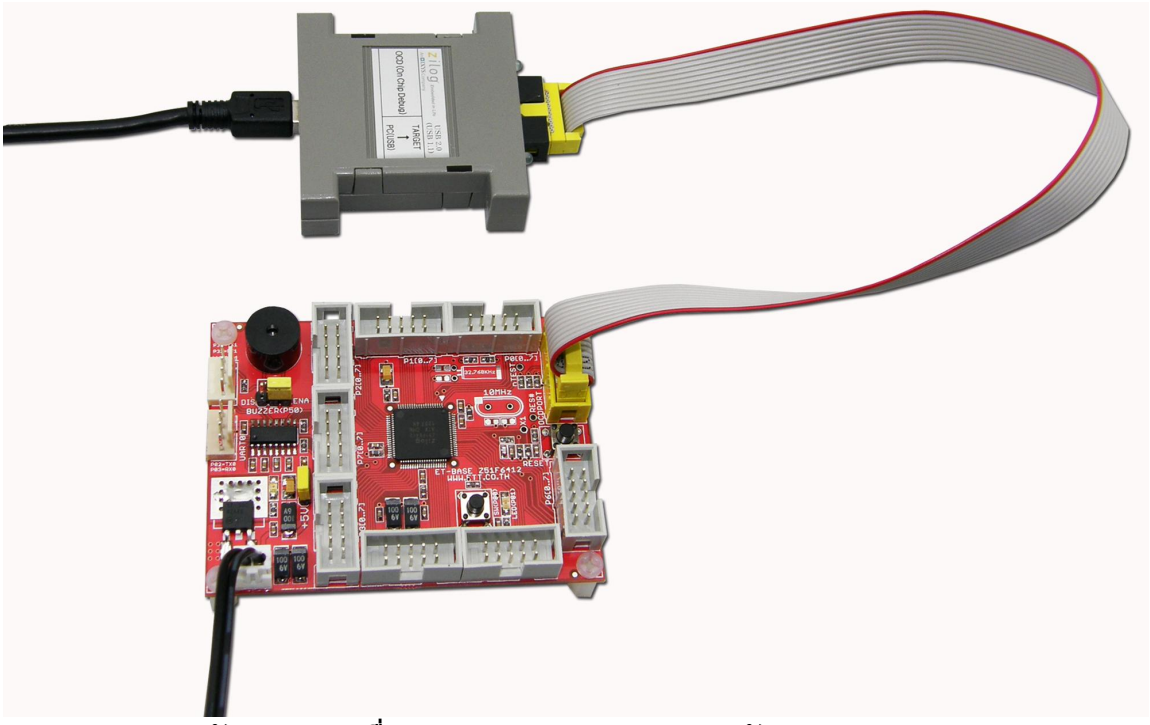


PORT-OCD(DEBUG)

PORT-OCD เป็นขั้วสัญญาณสำหรับเชื่อมต่อกับเครื่องมือ OCD/ISP สำหรับใช้สั่ง โปรแกรม และควบคุมการ Debug กับ MCU ในบอร์ด โดยรองรับการเชื่อมต่อกับเครื่องมือที่ได้รับการออกแบบไว้รองรับการ โปรแกรมแบบ ISP (In System Programming) และ Debug ของ MCU ตระกูล Z8051 ตามข้อกำหนดมาตรฐานของ Zilog ซึ่งได้แก่

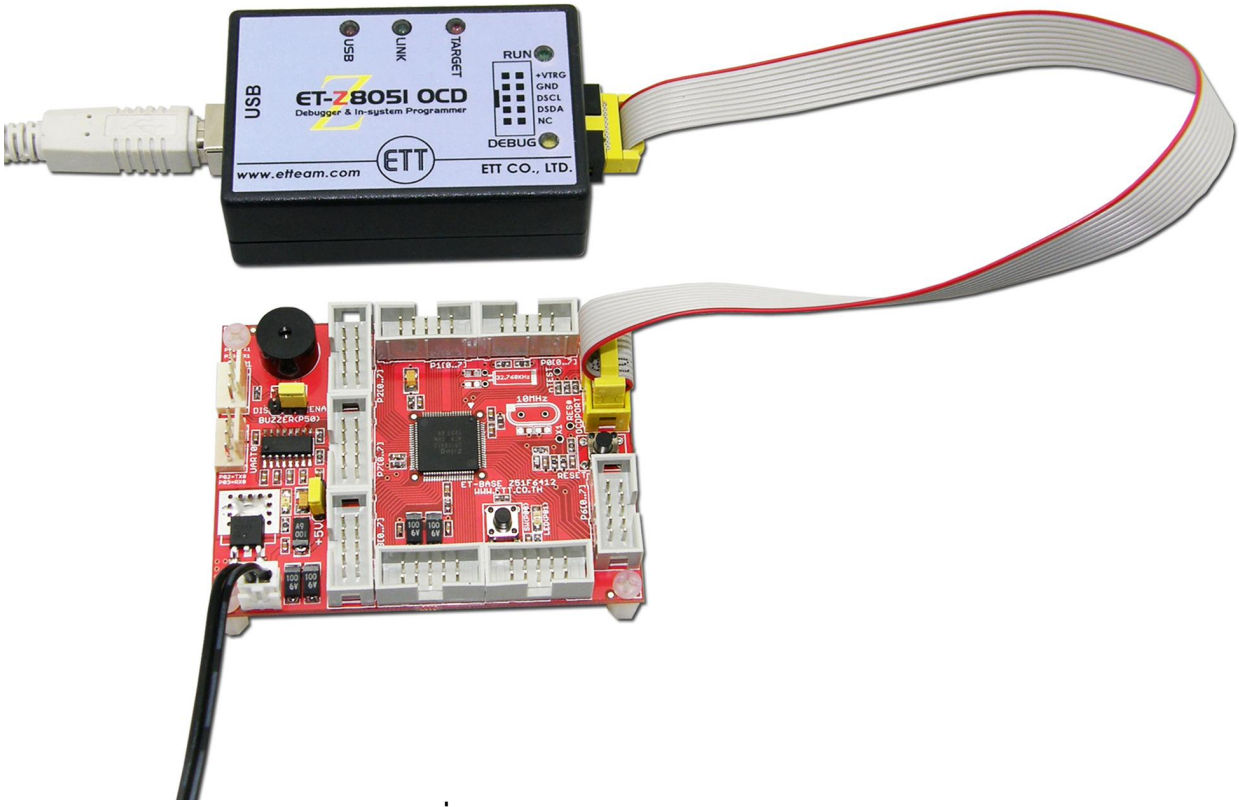
- Zilog Z8051 OCD ของ Zilog Inc.
- ET-Z8051 OCD ของ อีทีที

โดยการเชื่อมต่อสัญญาณระหว่างบอร์ด ET-BASE Z51F6412 กับเครื่องมือ OCD จะกระทำผ่านทางสายแพร์ 10 Pin ดังตัวอย่าง



รูปแสดงตัวอย่างการเชื่อมต่อ ET-BASE Z51F6412 **กับ** Zilog Z8051 OCD

รูปแสดง ลักษณะเครื่อง ET-Z8051 OCD **ของ อีทีที**



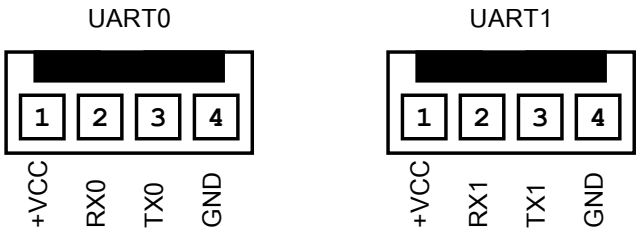
รูปแสดงตัวอย่างการเชื่อมต่อ ET-BASE Z51F6412 **กับ** ET-Z8051 OCD

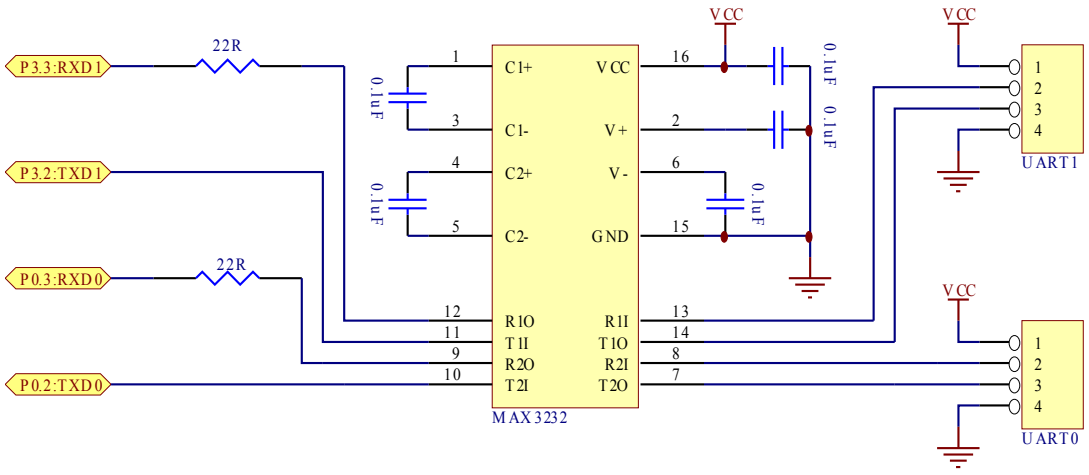
การใช้งาน RS232

พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณจาก MAX3232 เรียบร้อยแล้ว โดยในกรณีของ Z51F6412 นั้นตามปรกติแล้วจะมี Hardware UART จำนวน 4 ช่อง โดยใช้ขาสัญญาณต่างๆดังนี้

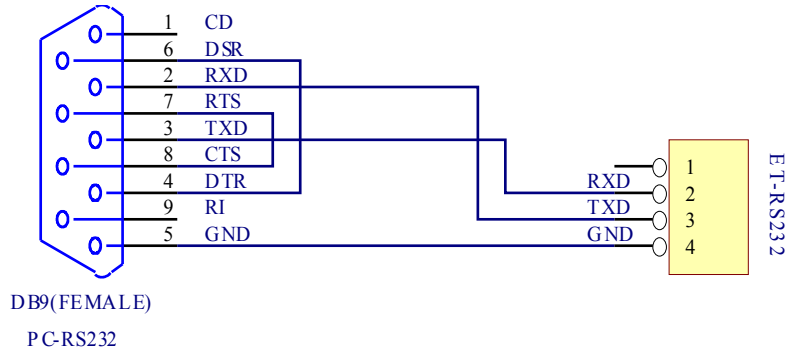
- UART0 จะใช้ P0.2(TX0) และ P0.3(RX0)
- UART1 จะใช้ P3.2(TX1) และ P3.3(RX1)
- UART2 จะใช้ P4.2(TX2) และ P4.3(RX2)
- UART3 จะใช้ P2.6(TX3) และ P2.7(RX3)

แต่ในส่วนวงจรของบอร์ด จะออกแบบให้ใช้ Hardware UART จำนวน 2 ช่อง สำหรับเชื่อมต่อกับวงจร Line Driver แบบ RS232 คือ UART0 และ UART1 ส่วน UART2 และ UART3 จะปล่อยว่างเป็นอิสระไว้สำหรับให้ผู้ใช้นำสัญญาณดังกล่าวไปออกแบบใช้งานต่างๆได้โดยอิสระตามความต้องการ เช่น ใช้เป็น RS422/RS485 หรือ ใช้งานสื่อสารแบบ UART กับอุปกรณ์อื่นๆโดยตรงแบบ TTL Level โดยสัญญาณของ RS232 แต่ละช่อง จะจัดขั้วเป็นแบบ CPA-4PIN (RS232) ดังรูป

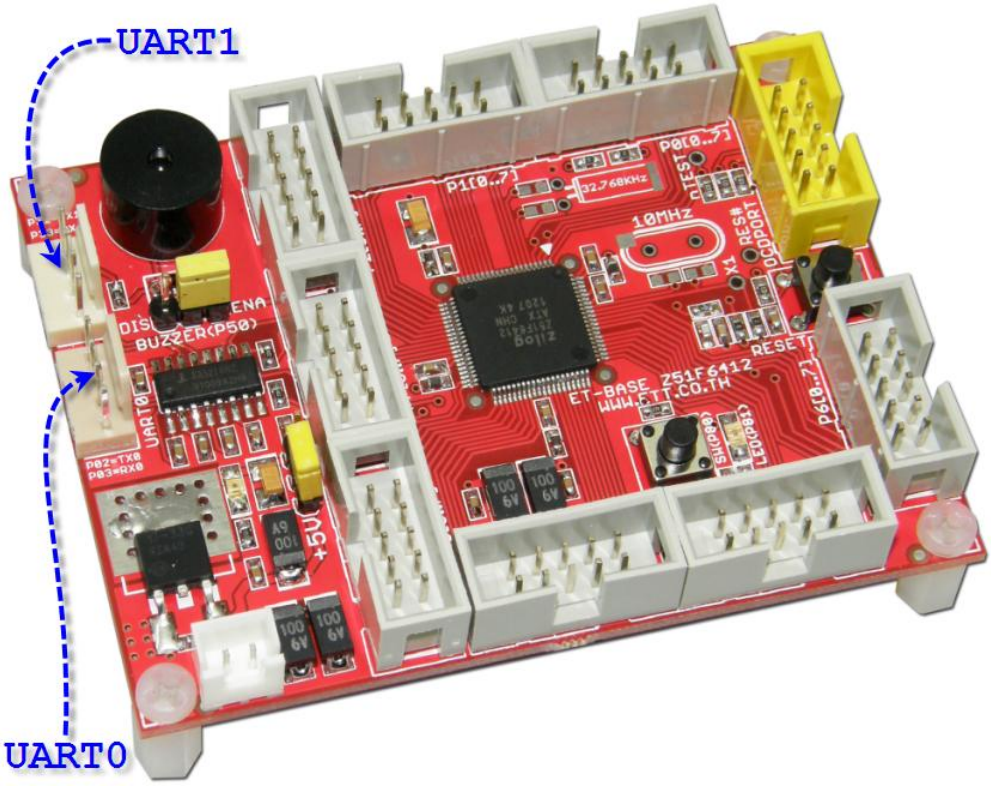




สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้วต่อ RS232 ของบอร์ด ET-BASE Z51F6412 นั้น เป็นดังนี้

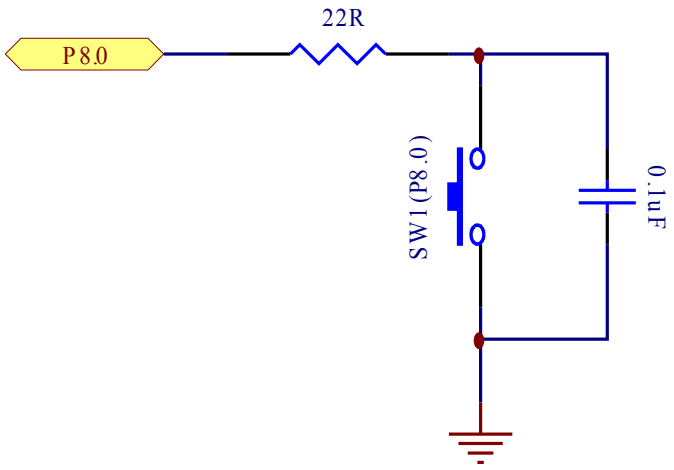


รูป แสดงวงจรสาย Cable สำหรับ RS232



การใช้วงจร SW1

SW1 เป็นวงจรสวิตช์ กดติด-ปล่อยดับ ใช้สำหรับสร้างสัญญาณโลจิก “0” และ “1” เพื่อทดสอบการทำงานของ Input แบบ Logic เช่น ทดลองการตรวจจับค่าการกดสวิตช์ โดยเมื่อสวิตช์ไม่ถูกกดจะได้ค่าสถานะทางโลจิกเป็น “1” แต่ถ้าสวิตช์ถูกกดจะได้ค่าสถานะทางโลจิกเป็น “0” โดยสัญญาณโลจิกที่ได้จากวงจรนี้ จะถูกเชื่อมต่อไปยัง ขาสัญญาณ P8.0 ของ MCU ดังรูป



เนื่องจากโครงสร้าง Port ของ Z51F6412 ได้รับการพัฒนาให้มีความอ่อนตัวในการใช้งาน และมีวงจรภายในหลายๆอย่างที่เอื้ออำนวยต่อการใช้งานให้ผู้ใช้สามารถเลือกว่าจะใช้งานหรือไม่ใช้งานของวงจรแต่ละส่วนได้ตามต้องการ ซึ่งในส่วนของการใช้งาน Port เป็น Input เพื่ออ่านค่าสถานะจากสวิตช์แบบหน้าสัมผัสนั้น Z51F6412 จะมีรีจิสเตอร์ ที่เกี่ยวข้องกับการใช้งาน Port เป็น Input จำนวน 4 ชุดเพื่อให้ผู้ใช้เลือกกำหนดการทำงานของ Port ตามความเหมาะสมในการใช้งานคือ

- P8 เป็น Port P8 Data Register ของ Port P8 ใช้สำหรับอ่านค่าสถานะของ Pin Port
- P8IO เป็น Port P8 Direction ของ Port P8 ใช้กำหนดหน้าที่ของ Port ว่าจะเป็น In หรือ Out
- P8PU เป็น Port P8 Pull-Up ใช้กำหนดว่าจะเปิดใช้ Pull-Up ภายในชิพหรือไม่
- P8DB เป็น Port P8 Debounce ใช้กำหนดว่าจะเปิดใช้งานวงจร Debounce ภายในชิพหรือไม่

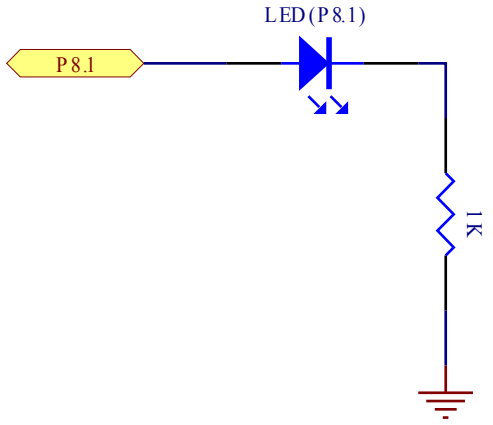
โดยรีจิสเตอร์แต่ละชุดจะมีขนาด 8 บิต โดยแยกหน้าที่แต่ละบิตเพื่อใช้สำหรับควบคุม Pin Port แต่ละ Pin ได้โดยอิสระ โดยในกรณีของ SW1 ซึ่งเชื่อมต่อไว้กับ P8.0 ก็จะใช้เฉพาะข้อมูลใน บิต0 ของแต่ละรีจิสเตอร์ เพื่อควบคุมการทำงานของ Pin P8.0 ดังตัวอย่าง

```
#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);        // Defualt 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
    }
  }
}
```

แสดง ตัวอย่าง Code สำหรับ Initial SW1

การใช้งานวงจร LED

LED เป็นวงจรแสดงผลทางโลจิก โดยใช้สำหรับแสดงผลทางโลจิกเพื่อให้ผู้ใช้ได้รับรู้ โดยใช้กับสัญญาณ Output แบบ Logic โดยถ้าได้รับโลจิก “1” จะทำให้ LED ติดสว่าง และ ถ้าได้รับโลจิก “0” จะทำให้ LED ดับ โดยสัญญาณโลจิกที่จะใช้ขับการแสดงผลของ LED ในวงจรนี้ จะถูกเชื่อมต่อมาจาก ขาสัญญาณ P8.1 ของ MCU ดังรูป

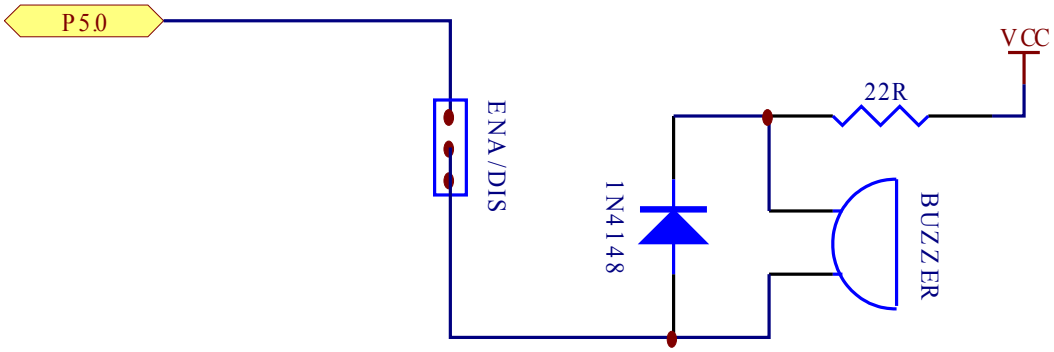



```
#include <z51f6412.h>                          // Z8051 : Z51F6412

/*************************************/
/* 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
  .
  .
  .
}
```

แสดง ตัวอย่าง Code สำหรับ Initial LED

การใช้งานวงจร Buzzer

Buzzer เป็นวงจรกำเนิดเสียงด้วยค่าความถี่ต่างๆ ซึ่งใน Z51F6412 จะมีวงจรควบคุมการกำเนิดเสียงของ Buzzer บรรจุไว้ภายในโครงสร้างของตัวชิพด้วย ใช้ใช้ขาสัญญาณ P5.0 เป็นขาสัญญาณในการควบคุมการกำเนิดเสียงของ Buzzer ซึ่งวงจรพิเศษส่วนนี้สามารถลดความยุ่งยากซับซ้อนในการเขียนโปรแกรมสร้างความถี่เพื่อควบคุมการกำเนิดเสียงของ Buzzer ได้อย่างมาก ผู้ใช้งานเพียงแต่กำหนดค่าสำหรับกำหนดการทำงานของ Buzzer ให้กับรีจิสเตอร์ที่เกี่ยวข้อง วงจรภายในก็จะส่งสัญญาณควบคุม Buzzer ให้เองโดยอัตโนมัติทำให้การเขียนโปรแกรมเป็นเรื่องง่ายมากขึ้น โดยสัญญาณ Pin Port ที่จะใช้ขับBuzzer จะใช้ Pin P5.0 โดยมี Jumper สำหรับตัดต่อสัญญาณให้ผู้ใช้งานเลือกใช้งานหรือไม่ได้ตามต้องการดังรูป

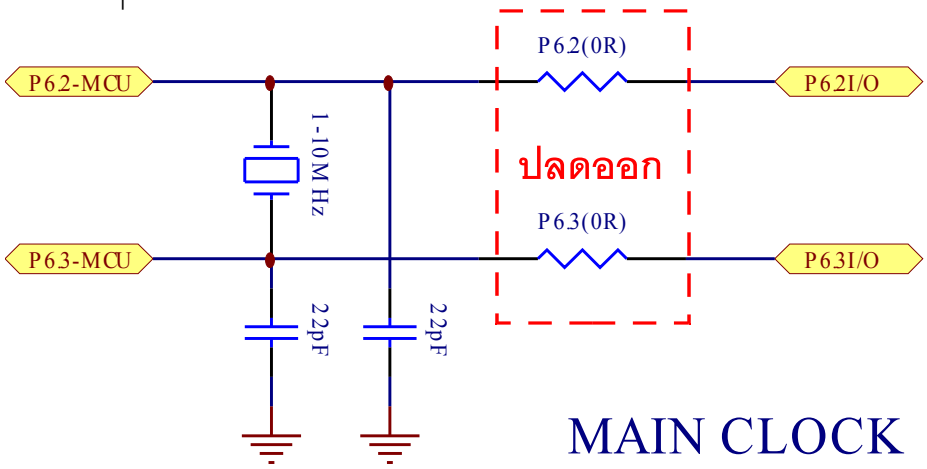



```
#include <z51f6412.h>                // Z8051 : Z51F6412
.
.
void main(void)
{
  .
  .
  // Config Buzzer
  // BUZCR = 00000,11,0
  // 00000xxx : Default Reserve Bit
  // xxxxx11x : Buzzer Clock Source = Fx/256
  // xxxxxxx0 : Buzzer Disable
  BUZCR |= 0x06;
  BUZDR =  0x1F; //Buzzer Frequency

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

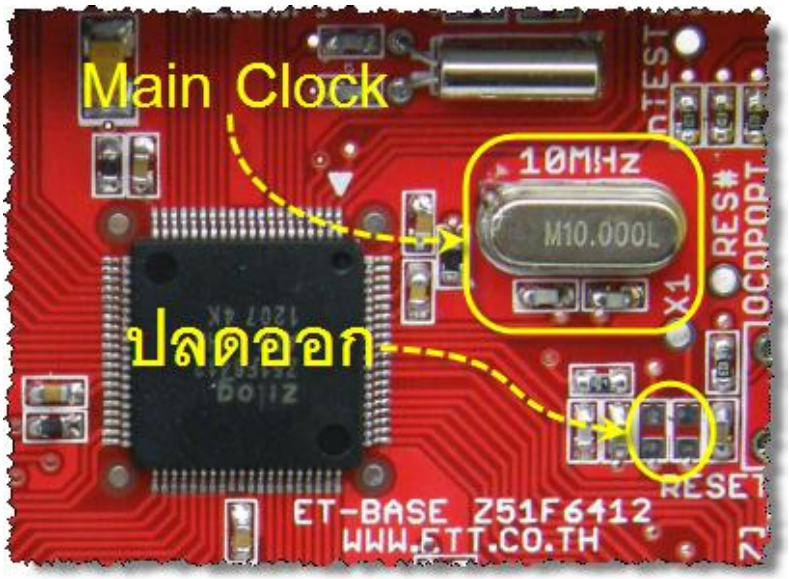
แสดง ตัวอย่าง Code สำหรับ Initial Buzzer

การใช้งานวงจรกำเนิดความถี่ Main Clock Oscillator

Main Clock Oscillator เป็นสัญญาณนาฬิกาจากภายนอก โดยใช้ Crystal Oscillator ค่าระหว่าง 1MHz ถึง 10MHz เป็นตัวกำเนิดความถี่ ซึ่งสัญญาณนาฬิกาที่ได้จะมีความแม่นยำสูงมาก เหมาะสำหรับงานที่ต้องการค่าความแม่นยำสูงๆ โดยเมื่อต้องการเลือกกำหนดให้ MCU ทำงานจากสัญญาณนาฬิกา Main Clock Oscillator นี้ ผู้ใช้ต้องทำการติดตั้ง Crystal ค่าระหว่าง 1 – 10 MHz และ ตัวเก็บประจุค่า 22pF เพิ่มเติมให้กับบอร์ดด้วย ซึ่งในกรณีนี้จะต้องสูญเสียขาสัญญาณ I/O ในการใช้งานลดลงไป จำนวน 2 เส้น คือ P6.2 และ P6.3 เนื่องจากจำเป็นต้องใช้ ขาสัญญาณทั้ง 2 เส้นเชื่อมต่อกับโมดูล Crystal Oscillator และควรปลดตัวต้านทานค่า 0-OHM เพื่อปลดการเชื่อมต่อสัญญาณ P6.2 และ P6.3 ไปยัง Connector IDE 10 Pin ของ Port P6[0..7] ออกด้วย เพื่อป้องกันไม่ให้สัญญาณนาฬิกานี้ถูกโหลดจากสัญญาณภายนอกอื่นๆ

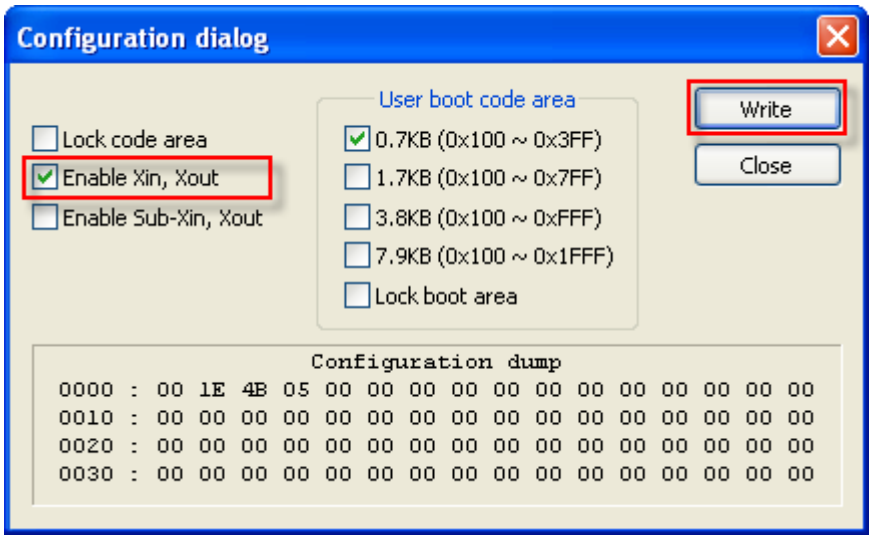


รูปแสดง การติดตั้ง Crystal 1-10 MHz และ ตัวเก็บประจุ สำหรับวงจร Main Clock Oscillator



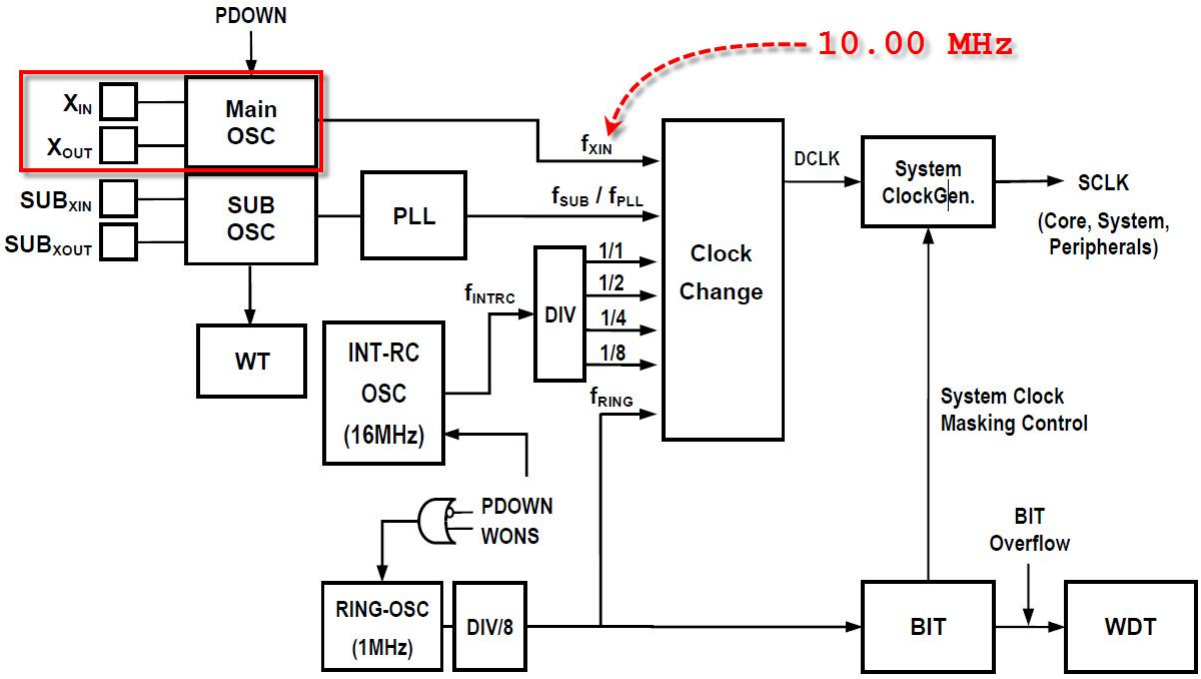
รูปแสดง การติดตั้ง Main Clock และการปลด P6.2/P6.3 ออกจากขั้วต่อ Port P6[7..0]

ผู้ใช้สามารถกำหนดให้วงจรกำเนิดสัญญาณนาฬิกา Main Clock Oscillator นี้ทำงานตอน MCU เริ่มต้นทำงานได้จากการกำหนดค่าใน Configuration Bit จากขั้นตอนของการโปรแกรม MCU ด้วย และยังสามารถสั่งเปิดปิดการทำงานของวงจรกำเนิดความถี่จากคำสั่งในโปรแกรมได้อีกช่องทางหนึ่งด้วย



รูปแสดง การกำหนด Configuration Bit **เพื่อเปิดการทำงานของวงจร** Main Clock Oscillator

ผู้ใช้สามารถกำหนดให้ MCU ทำงานจากแหล่งกำเนิดสัญญาณนาฬิกา Main Clock Oscillator นี้ด้วยคำสั่งในโปรแกรม และสามารถสั่งสลับสัญญาณนาฬิกาไปยังแหล่งอื่นๆได้ตามต้องการด้วย



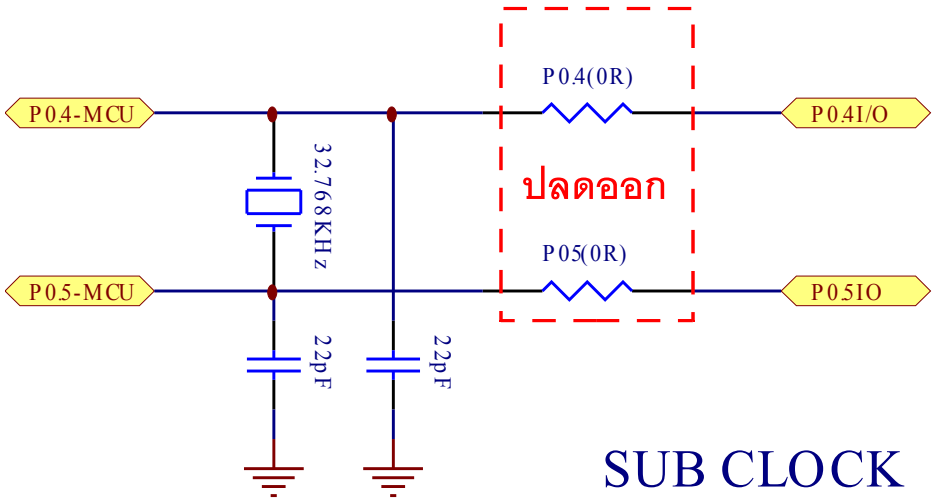
รูปแสดง การเลือกใช้สัญญาณนาฬิกาจากวงจร Main Clock Oscillator

```
#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
  // xx0xxxxx : 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
  // x00xxxxx : Clock Divide 1
  // xxx1xxxx : 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 */
  .
  .
  .
}
```

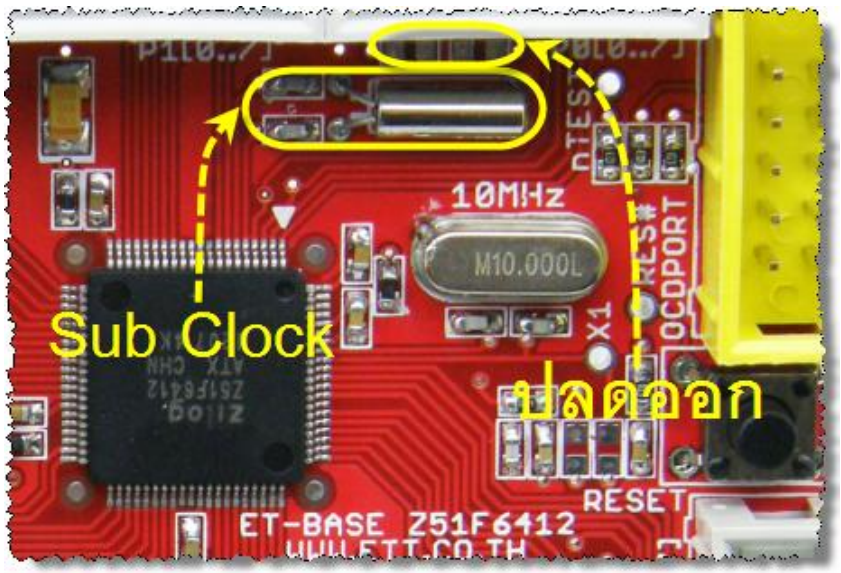
ตัวอย่างโปรแกรม สำหรับกำหนด MCU **ทำงานจาก** Main Clock Oscillator

การใช้งานวงจรกำเนิดความถี่ Sub Clock Oscillator

Sub Clock Oscillator เป็นระบบ Clock จากภายนอก MCU โดยใช้ Crystal ความถี่ต่ำค่า 32.768KHz เพื่อลดสัญญาณรบกวนที่อาจเกิดจากวงจรกำเนิดความถี่ โดยความถี่นี้สามารถนำไปเข้าวงจรคูณความถี่ภายใน (Phase-Lock-Loop) เพื่อคูณความถี่ให้สูงขึ้นได้ถึง 17.45MHz โดยเมื่อต้องการเลือกกำหนดให้ MCU ทำงานจากสัญญาณนาฬิกา Sub Clock Oscillator นี้ ผู้ใช้ต้องทำการติดตั้ง Crystal ค่า 32.768 KHz และ ตัวเก็บประจุค่า 22pF เพิ่มเติมให้กับบอร์ดด้วย ซึ่งในกรณีนี้จะต้องสูญเสียขาสัญญาณ I/O ในการใช้งานลดลงไป จำนวน 2 เส้น คือ P0.4 และ P0.5 เนื่องจากจำเป็นต้องใช้ ขาสัญญาณทั้ง 2 เส้นเชื่อมต่อกับโมดูล Crystal Oscillator ค่า 32.768 KHz และควรปลดตัวต้านทานค่า 0-OHM เพื่อปลดการเชื่อมต่อสัญญาณ P0.4 และ P0.5 ไปยัง Connector IDE 10 Pin ของ Port P0[0..7] ออกด้วย เพื่อป้องกันไม่ให้สัญญาณนาฬิกานี้ถูกโหลดจากสัญญาณภายนอกอื่นๆ

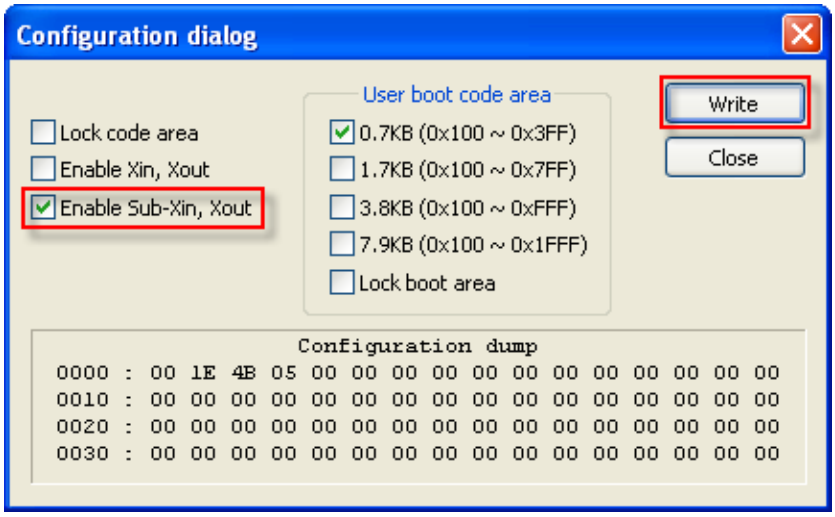


รูปแสดง การติดตั้ง Crystal 32.768 KHz และ ตัวเก็บประจุ สำหรับวงจร Sub Clock Oscillator



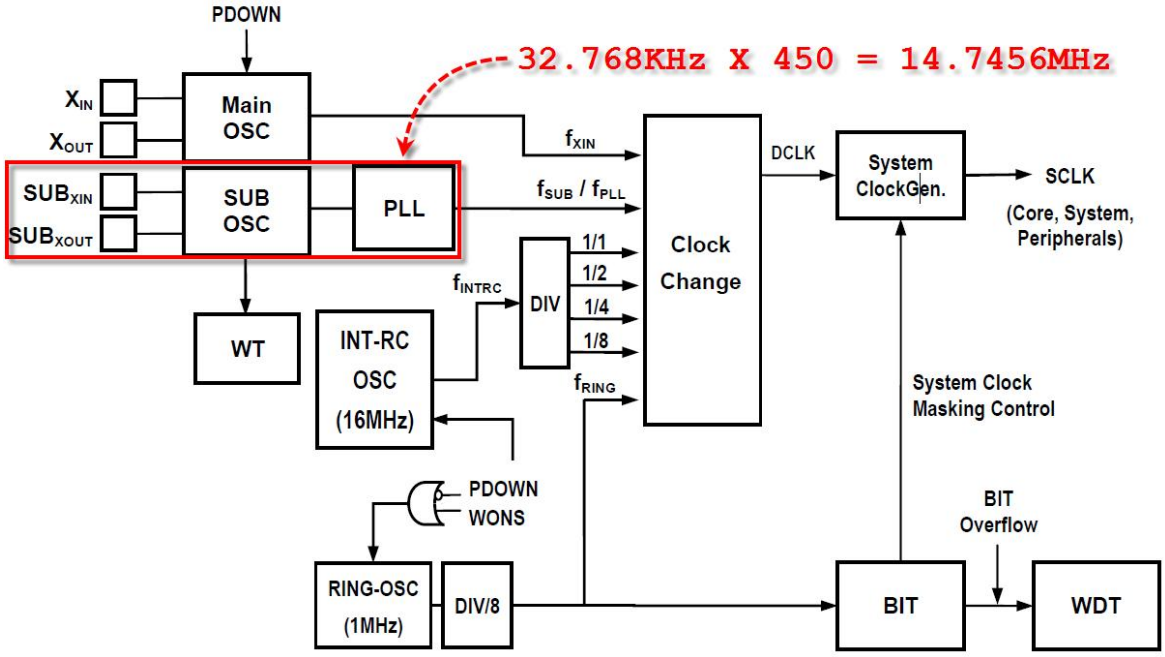
รูปแสดง การติดตั้ง Sub Clock **และการปลด** P0.4/P0.5 **ออกจากขั้วต่อ** Port P0[7..0]

ผู้ใช้สามารถกำหนดให้วงจรกำเนิดสัญญาณนาฬิกา Sub Clock Oscillator นี้ทำงานตอน MCU เริ่มต้นทำงานได้จากการกำหนดค่าใน Configuration Bit จากขั้นตอนของการโปรแกรม MCU ด้วย และยังสามารถสั่งเปิดปิดการทำงานของวงจรกำเนิดความถี่จากคำสั่งในโปรแกรมได้อีกช่องทางหนึ่งด้วย



รูปแสดง การกำหนด Configuration Bit **เพื่อเปิดการทำงานของวงจร** Sub Clock Oscillator

ผู้ใช้สามารถกำหนดให้ MCU ทำงานจากแหล่งกำเนิดสัญญาณนาฬิกา Sub Clock Oscillator นี้ด้วยคำสั่งในโปรแกรม และยังสามารถเปิดใช้งานวงจรคูณความถี่ภายใน เพื่อคูณความถี่ ให้ MCU ทำงานที่ความถี่ 11.0755 MHz(คูณ 338) หรือที่ความเร็วสูงสุด 14.7456 MHz(คูณ 450) ได้ตามต้องการด้วย



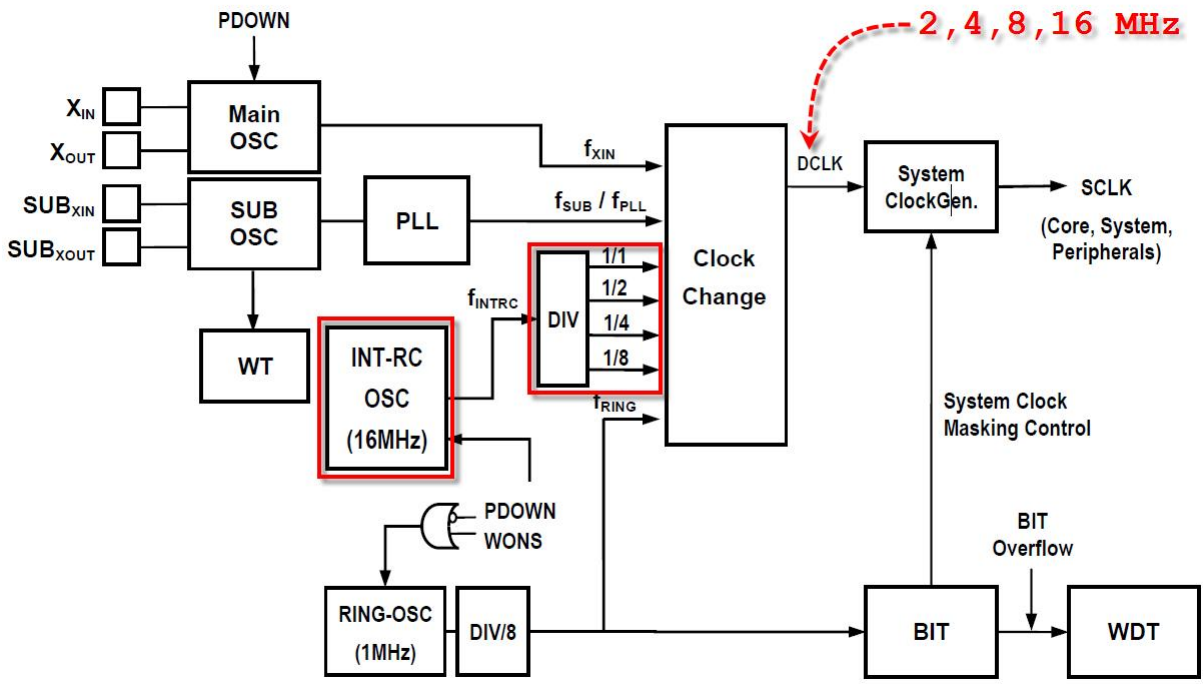
รูปแสดง การเลือกใช้สัญญาณนาฬิกาจากวงจร Sub Clock Oscillator + PLL

```
#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 */
  .
  .
  .
}
```

ตัวอย่างโปรแกรม สำหรับกำหนด MCU **ทำงานจาก** Sub Clock Oscillator + PLL

การใช้งานวงจรกำเนิดความถี่จาก Internal RC Oscillator

การเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาจาก Internal RC Oscillator ก็เป็นอีกทางเลือกหนึ่ง โดยเมื่อต้องการเลือกกำหนดให้ MCU ทำงานจากสัญญาณนาฬิกา Internal RC Oscillator นี้จะมีข้อดีหลายๆ อย่าง ทั้งเรื่องของการประหยัดพลังงาน ประหยัดอุปกรณ์ และได้ขาสัญญาณใช้งาน เพิ่มมากขึ้นถึง 4 ขา เพราะไม่ต้องสูญเสียขาสัญญาณ I/O ในการเชื่อมต่อกับโมดูล Crystal Oscillator ภายนอก โดยสัญญาณนาฬิกาภายในนี้มีความเที่ยงตรงพอสมควร โดยมีค่าความคาดเคลื่อนเพียง +/-2% เท่านั้น ซึ่งนับว่าแม่นยำเที่ยงตรงเพียงพอกับการใช้งานทั่วๆไปได้เป็นอย่างดี โดยผู้ใช้สามารถเลือกกำหนดค่าตัวหารสำหรับหารความถี่ที่ได้จากวงจรกำเนิดความถี่ของ Internal RC Oscillator เพื่อให้ได้ค่าความถี่เป็น 2MHz, 4MHz, 8MHz หรือ 16MHz ได้ตามต้องการจากคำสั่งในโปรแกรม



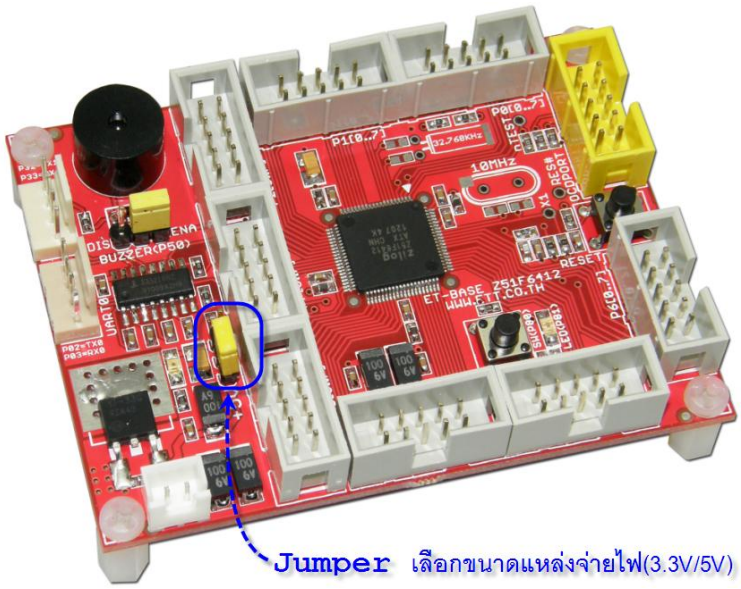
รูปแสดง การเลือกใช้สัญญาณนาฬิกาจากวงจร Internal RC Oscillator

```
#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
  // x0xxxxxx : PLL Output Bypass
  // xx0xxxxx : 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
  // x01xxxxx : INTRC Clock Divide = INTRC(16MHz)/2 = 8MHz
  // xxx0xxxx : 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 */
  .
  .
  .
}
```

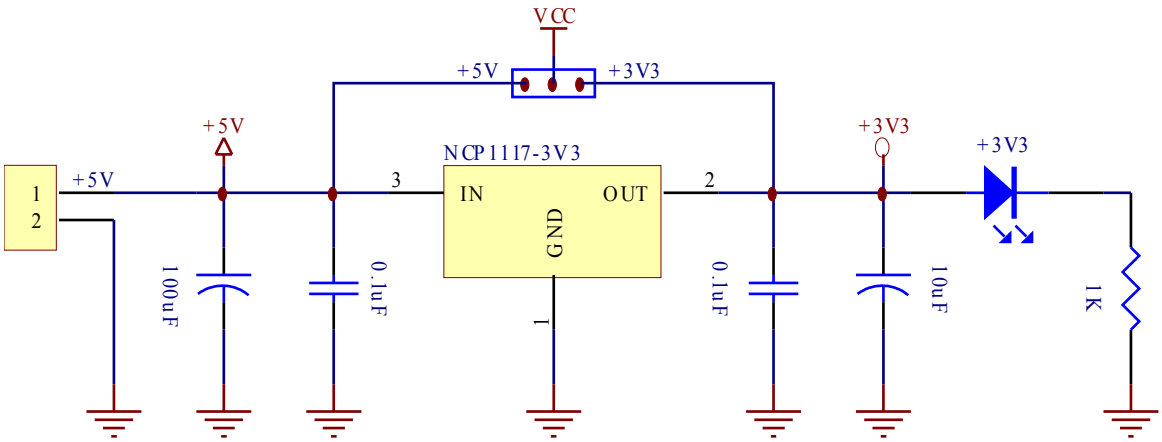
ตัวอย่างโปรแกรม สำหรับกำหนด MCU **ทำงานจาก** Internal RC **ค่า** 8.00 MHz

วงจรภาคจ่ายไฟ

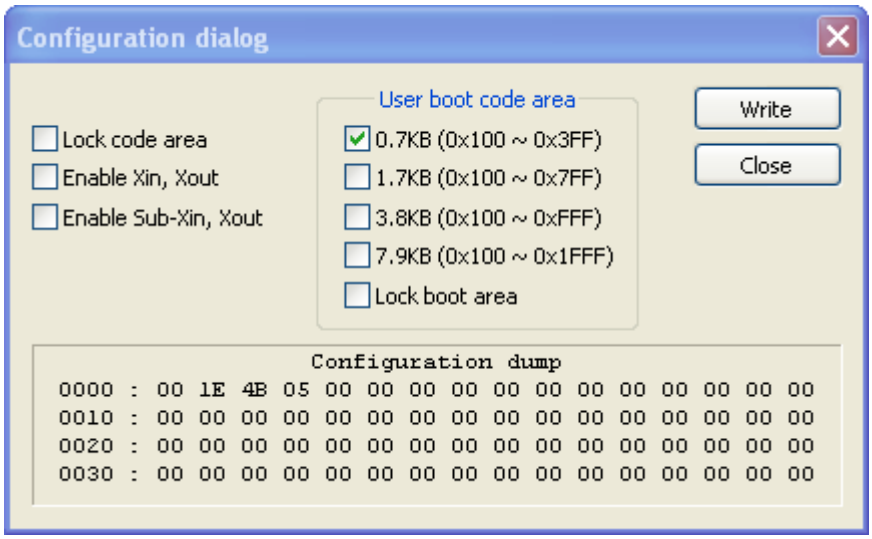
สำหรับวงจรภาคจ่ายไฟของบอร์ด จะใช้ได้กับแรงดันภายนอกขนาด +5VDC โดยภายในบอร์ดจะมีวงจร Regulate ขนาด 3.3V/1A พร้อม Jumper สำหรับเลือกใช้แรงดัน +5VDC หรือ +3V3 ให้เป็นแรงดันไฟเลี้ยงของ MCU (+VCC) และวงจร I/O ภายในบอร์ด



สำหรับการเลือกระดับแรงดัน +VCC ให้กับ MCU ภายในบอร์ดนั้น สามารถกระทำได้โดยการเลือกกำหนดจาก Jumper 3V3/+5V ซึ่งการจะเลือกใช้ +VCC เป็นเท่าใดนั้น ให้พิจารณาจากจุดประสงค์การใช้งานและอุปกรณ์การเชื่อมต่อที่ต้องการ ซึ่ง MCU เบอร์ Z51F6412 สามารถทำงานได้ดีในย่านแรงดัน 2.0V ถึง 5.5V ซึ่งบนบอร์ดจะสามารถเลือกระบบแรงดัน ได้ 2 ย่าน คือ +5V และ +3.3V ซึ่งผู้ใช้ต้องระลึกไว้เสมอว่า ถ้าเลือกใช้แรงดันของแหล่งจ่ายให้บอร์ดในย่านใด ระดับสัญญาณโลจิกในการเชื่อมต่อกับอุปกรณ์ภายนอกต้องอยู่ในระดับที่มีค่าไม่เกินแหล่งจ่ายด้วย เช่น ถ้าเลือก แต่ที่จะใช้แหล่งจ่ายเป็น 3.3V สัญญาณที่จะเชื่อมต่อก็ต้องเป็น 3.3V ด้วย ถ้านำสัญญาณ 5V มาเชื่อมต่ออาจทำให้ MCU เกิดความเสียหายได้ ดังนั้นผู้ใช้ต้องระมัดระวังในการเลือกขนาดของแรงดันให้กับ MCU ด้วย



Fuse Configuration ของ Z51F6412



Z51F6412 จะมีรีจิสเตอร์พิเศษชื่อ FUSE_CONF ใช้สำหรับกำหนดการทำงานของ MCU โดยจะกำหนดค่าของ Fuse นี้ผ่านขั้นตอนของการโปรแกรม Code ด้วยเครื่องมือ OCD ทั้งแบบ Debug และ ISP และ Parallel Programming ซึ่งรีจิสเตอร์ตัวนี้จะมีขนาด 8 บิต โดยมีค่าเริ่มต้น ซึ่งเป็นค่ามาตรฐานจากโรงงานที่กำหนดไว้ในตัว MCU คือ 0x00 โดยความหมายของรีจิสเตอร์ FUSE_CONF มีดังนี้

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

- BZSIE1:BSIZE0 ใช้ร่วมกันสำหรับกำหนดขนาดของ Boot Code Size ซึ่งกำหนดได้ 4 ขนาด
  - [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 ใช้สำหรับกำหนดการทำงานของ Sub Clock Oscillator ซึ่งเป็นวงจรกำเนิดสัญญาณนาฬิกาจากภายนอกค่า 32.768KHz ซึ่งจะต้องเชื่อมต่อเข้ากับ P0.4 และ P0.5 โดยค่ามาตรฐานจากโรงงานจะกำหนดไว้ที่ Disable
  - 0 : Sub Clock Oscillator Disable
  - 1 : Sub Clock Oscillator Enable

- **XINENA** ใช้สำหรับกำหนดการทำงานของ Main Clock Oscillator ซึ่งเป็นวงจรกำเนิดสัญญาณนาฬิกาจากภายนอกค่าระหว่าง 1 MHz – 10 MHz ซึ่งจะต้องเชื่อมต่อเข้ากับ P6.2 และ P6.3 โดยค่ามาตรฐานจากโรงงานจะกำหนดไว้ที่ Disable
    - **0** : Main Clock Oscillator Disable
    - **1** : Main Clock Oscillator Enable
- **OCDSEL** ใช้สำหรับกำหนด การทำงานของวงจรกำจัดสัญญาณรบกวนที่ขาสัญญาณ OCD (Select Noise Cancelling of OCD Pin)
    - **0** : กำหนดให้สัญญาณ OCD ผ่านการกรองด้วยวงจรกรองสัญญาณรบกวน 10nS
    - **1** : กำหนดให้สัญญาณ OCD Sync กับสัญญาณนาฬิกาภายใน INTRC Clock
- **LOCKE** ใช้กำหนดการป้องกันการลบหน่วยความจำในส่วนพื้นที่ของ Boot Code ตามขนาดที่กำหนดไว้โดย BSIZE[1:0]
    - **0** : Boot Lock Disable
    - **1** : Boot Lock Enable ซึ่งจะเป็นการป้องกันการลบข้อมูลในหน่วยความจำในส่วนที่เป็นพื้นที่ Boot Code
- **LOCKF** ใช้กำหนดรูปแบบการป้องกันหน่วยความจำ Code โดยเมื่อสั่ง Enable ค่า Fuse ของบิตนี้เป็น 1 แล้วจะไม่สามารถสั่งอ่านข้อมูลจากหน่วยความจำ Code ของ MCU จากเครื่องโปรแกรมภายนอกได้ และ Fuse นี้จะยกเลิกได้จากการสั่งลบข้อมูลในหน่วยความจำทั้งหมดแล้วเท่านั้น
    - 0 : Lock Disable
    - 1 : Lock Enable(ป้องกันการอ่าน)

หมายเหตุ ในโหมดของการ Debugger ผู้ใช้สามารถสั่งแก้ไขค่าของรีจิสเตอร์นี้เป็นการชั่วคราวโดยไม่ต้องสั่งโปรแกรมค่า Config จริงๆ เพื่อทำการทดสอบผลการทำงานของ MCU แบบชั่วคราวก่อนได้ ยกเว้นค่าของบิต “LOCKF” ซึ่งไม่สามารถสั่งเปลี่ยนแปลงแก้ไขแบบชั่วคราวได้จากเครื่องมือ Debugger โดยเมื่อ MCU ออกจากการทำงานใน Debugger แล้ว MCU ก็จะมีคุณสมบัติตามค่า Config ปรกติ

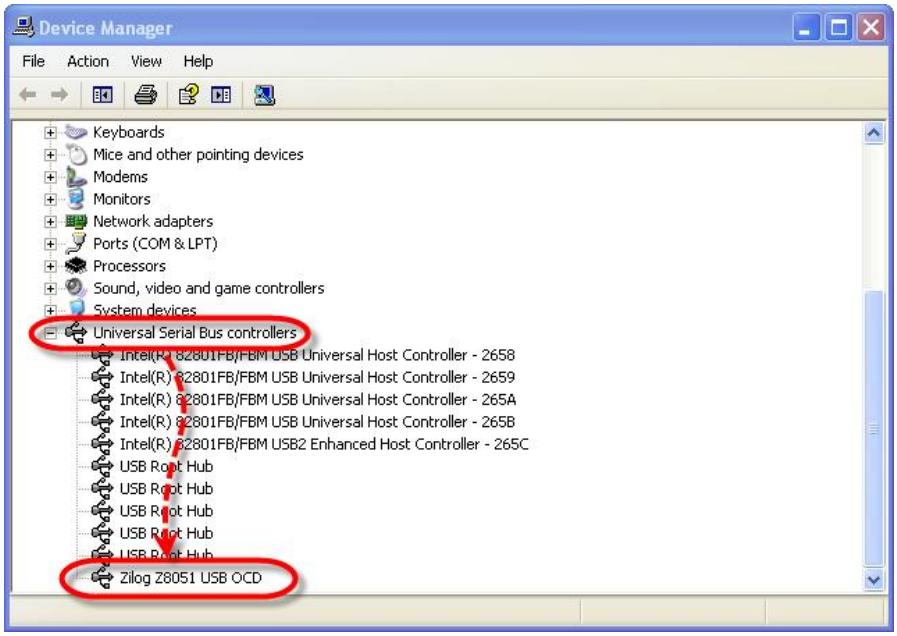
การเชื่อมต่อ MCU กับ ET-Z8051 OCD ใน Debug Mode

รูปแสดงการเชื่อมต่อ ET-BASE Z51F6412 กับ ET-Z8051 OCD

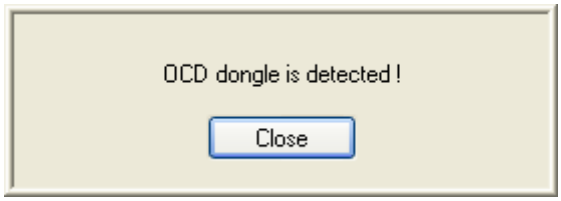
ในโหมดนี้จะเหมาะกับการใช้งานในรูปแบบที่อยู่ในขั้นตอนของการพัฒนาโปรแกรมเพราะสามารถสั่งโปรแกรม MCU พร้อมทั้งควบคุม ทดสอบ หยุดการทำงาน รวมไปถึงการตรวจสอบและปรับแต่งค่ารีจิสเตอร์ ต่างๆในขณะที่ MCU ทำงานอยู่จริงได้ด้วย ทำให้ไม่ต้องคอย ลบ และโปรแกรม MCU ซ้ำบ่อยๆ โดยในโหมดนี้จะใช้งานร่วมกับโปรแกรม Zilog Z8051 OCD ซึ่งปัจจุบัน (สิงหาคม 2555) โปรแกรมจะได้รับการปรับปรุงเป็นรุ่น Zilog Z8051 OCD Version1.147 โดยมีขั้นตอนการใช้งานพอสังเขปดังนี้

1. ทำการติดตั้งโปรแกรม Zilog Z8051 OCD ให้เรียบร้อย โดย Run File ชื่อ “Z8051_1.1.exe”
2. ทำการเชื่อมต่อสาย USB ของ ET-Z8051 OCD เข้ากับคอมพิวเตอร์ PC พร้อมกับทำการติดตั้ง Driver ของอุปกรณ์ให้เรียบร้อย ซึ่งขั้นตอนนี้จะกระทำเพียงครั้งแรกครั้งเดียวเท่านั้น โดยอุปกรณ์ ET-Z8051 OCD ของ อีทีที จะใช้ Driver ชุดเดียวกับ Zilog Z8051 OCD ของ Zilog Inc. ซึ่งตามปรกติ Driver จะถูกติดตั้งเตรียมไว้พร้อมกันกับการติดตั้งโปรแกรม ชื่อ “Z8051_1.1.exe” ในขั้นตอนที่ 1 ซึ่งถ้าติดตั้งโปรแกรมตามค่ามาตรฐาน Driver จะอยู่ที่ “\device drivers\OCD USB\” ภายใต้ Directory ที่ทำการติดตั้งโปรแกรมไว้ โดยปัจจุบันจะมี Driver สำหรับรองรับกับระบบปฏิบัติการ Windows สำหรับเครื่องคอมพิวเตอร์ทั้งแบบ 32บิต และ 64บิต คือ
    a. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x32” สำหรับ 32บิต
    b. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x64” สำหรับ 64บิต

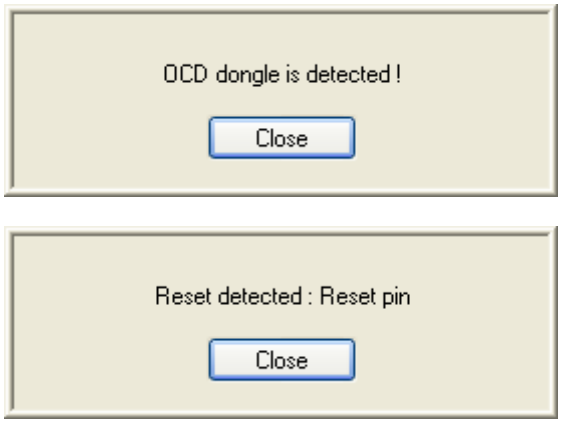
ซึ่งถ้าทุกอย่างถูกต้องเรียบร้อยแล้ว LED USB สีแดง ที่ตัวเครื่อง ET-Z8051 OCD จะติดสว่าง และเมื่อเข้าไปตรวจสอบที่ Device Manager จะต้องพบอุปกรณ์ USB ใน Tab ของ Universal Serial Bus controller ควรพบอุปกรณ์ชื่อ “Zilog Z8051 USB OCD” ดังตัวอย่าง



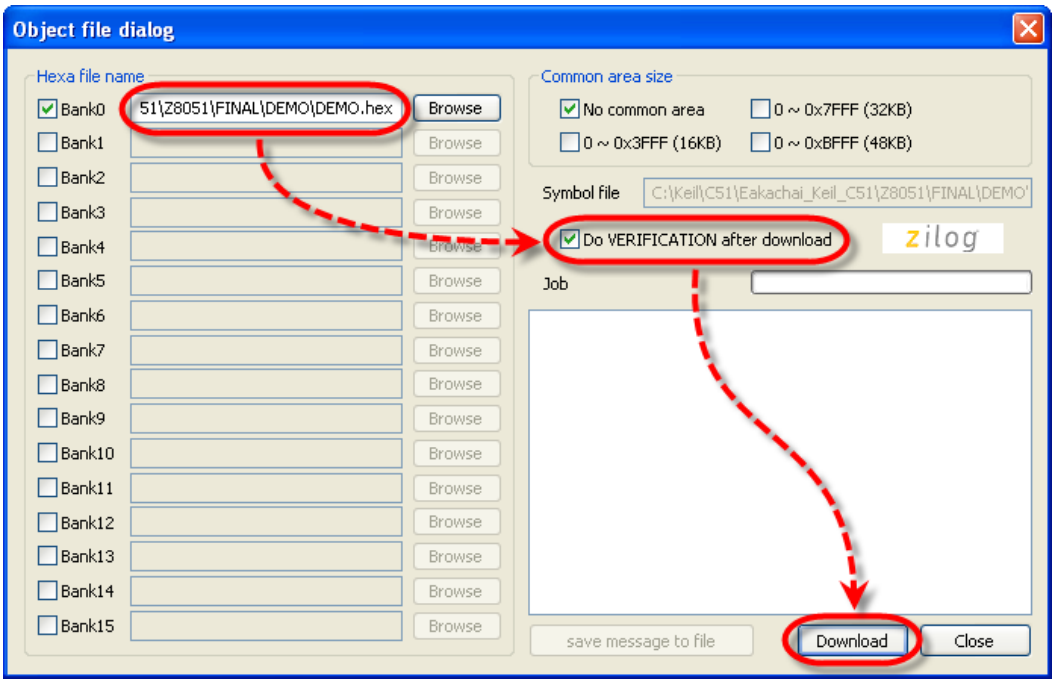
3. สั่ง Run โปรแกรม Zilog Z8051 OCD V1.147 ซึ่งเมื่อโปรแกรมเริ่มทำงานในครั้งแรก จะปรากฏ Dialog ข้อความแจ้งการตรวจพบ อุปกรณ์ Z8051 OCD ดังตัวอย่าง พร้อมกับที่อุปกรณ์ OCD ของ ET-Z8051 OCD จะปรากฏ LED LINK สีเขียว ติดสว่างให้เห็น เพื่อแสดงว่าอุปกรณ์ OCD พร้อมทำงาน



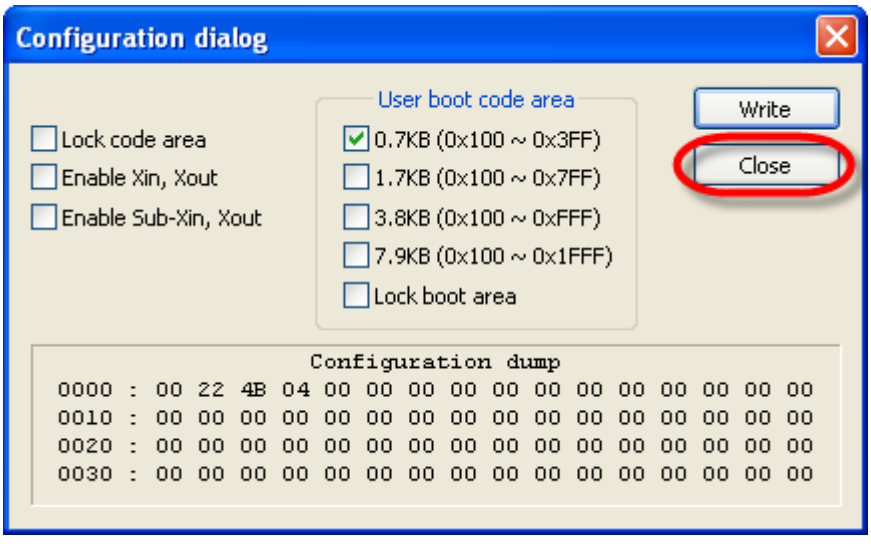
4. ถ้ามีการจ่ายไฟเลี้ยงวงจรให้บอร์ด ET-BASE Z51F6412 ไว้แล้วให้ทำการปลดสายไฟเลี้ยงของบอร์ด ET-BASE Z51F6412 ออกจากบอร์ดให้เรียบร้อยก่อน แล้วจึงทำการต่อสายแพร์ขนาด 10Pin ระหว่างเครื่อง ET-Z8051 OCD เข้ากับขั้วต่อ IDE10Pin สีเหลือง (PORT-OCD) ของบอร์ด ET-BASE Z51F6412 ให้เรียบร้อย จากนั้นจึงทำการจ่ายไฟเข้าบอร์ด ET-BASE Z51F6412 ซึ่งถ้าทุกอย่างถูกต้องหลอด LED สถานะของเครื่อง ET-Z8051 OCD จะติดสว่างหมดทั้ง 3 ดวง คือ USB LINK และ TARGET และที่หน้าจอโปรแกรมจะปรากฏ Dialog ข้อความ 2 ข้อความ ตามลำดับ ให้เห็น และปรากฏเบอร์ “Z516412ARX/ATX” ในช่อง Device name ดังตัวอย่าง



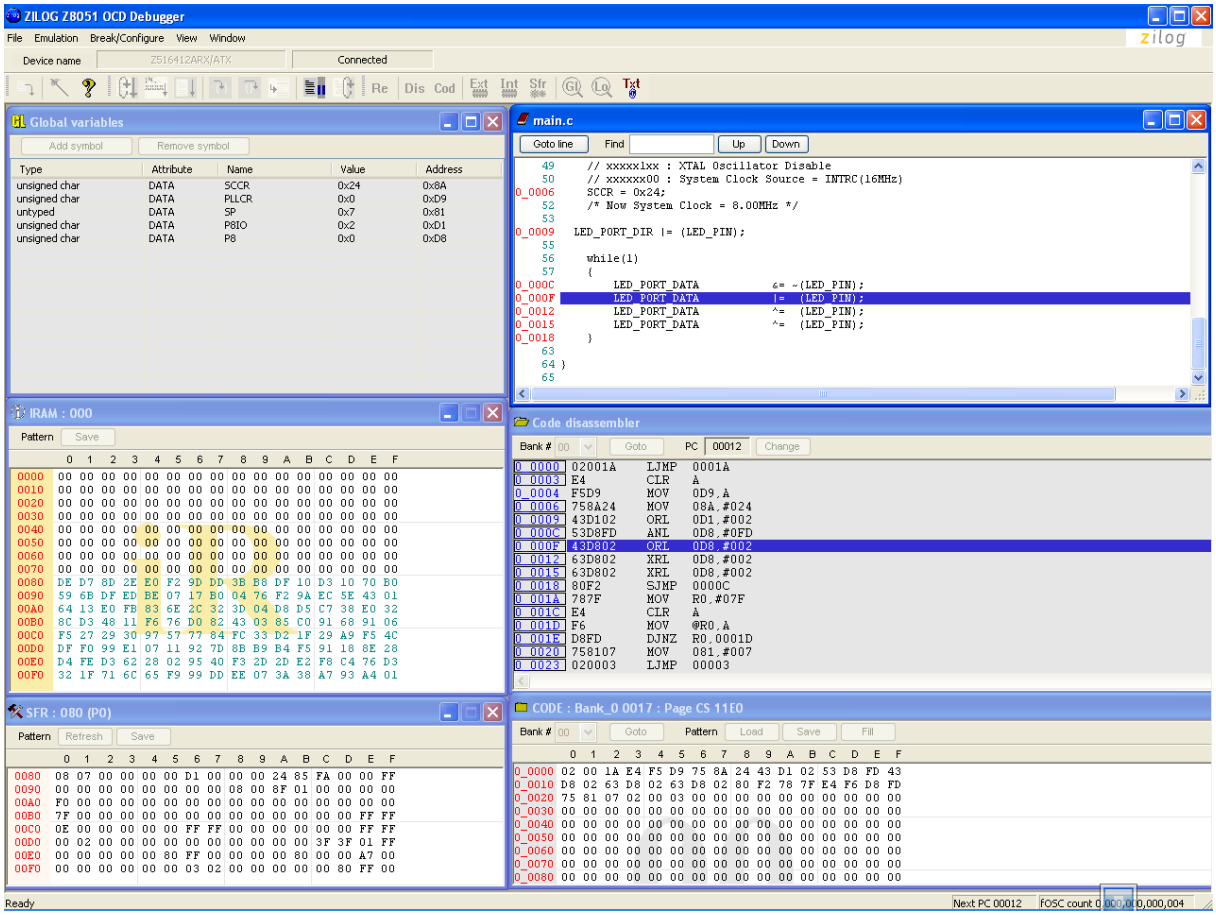
5. มาถึงขั้นตอนนี้ก็แสดงว่าการเชื่อมต่อระหว่าง MCU ในบอร์ด ET-BASE Z51F6412 กับเครื่อง OCD ของ ET-Z8051 OCD สามารถเชื่อมต่อกันได้อย่างถูกต้องเป็นที่เรียบร้อยแล้ว จากนี้ไปผู้ใช้ก็สามารถสั่งงานโปรแกรม Zilog Z8051 OCD ให้ทำงานต่างๆตามที่ต้องการได้แล้ว โดยขั้นตอนต่อไปที่ต้องทำคือทำการสั่งโปรแกรม Hex File ที่ต้องการจะ Download ให้กับ MCU ในบอร์ด โดยให้เลือกที่เมนู File ⟶ Load HEX แล้ว กำหนดชื่อไฟล์ที่ต้องการ แล้วสั่ง Download ดังตัวอย่าง



ส่วนของ Configuration ถ้าไม่มีการเปลี่ยนแปลงแก้ไขค่าตัวเลือกใดๆ ให้เลือก Close ได้เลย แต่ถ้าต้องการเปลี่ยนแปลงแก้ไขค่าใดก็ให้ทำการเลือกกำหนดค่าตามต้องการแล้วเลือก Write ในที่นี้ให้เลือก Close เพื่อกำหนดค่า Configuration ตามค่ามาตรฐานเดิมจากโรงงานดังตัวอย่าง



6. หลังจากสั่ง Download Hex File ให้กับ MCU เรียบร้อยแล้ว จากนี้ไปก็สามารถสั่งงาน MCU ให้ทำงานต่างๆตามต้องการได้ทันที ซึ่งอาจเป็นการสั่ง Run(Go) เพื่อดูผลการทำงานจริง หรือกำหนดตำแหน่งสำหรับหยุดการทำงาน (Break) หรือ สั่ง Run ทีละคำสั่ง(Step) เพื่อตรวจสอบผลการทำงานของโปรแกรมว่าถูกต้องตามที่ออกแบบไว้หรือไม่ ซึ่งรายละเอียดส่วนนี้ขอให้ศึกษาเพิ่มเติมจากส่วนของคู่มือการใช้งาน ET-Z8051 OCD และโปรแกรม Zilog Z8051 OCD



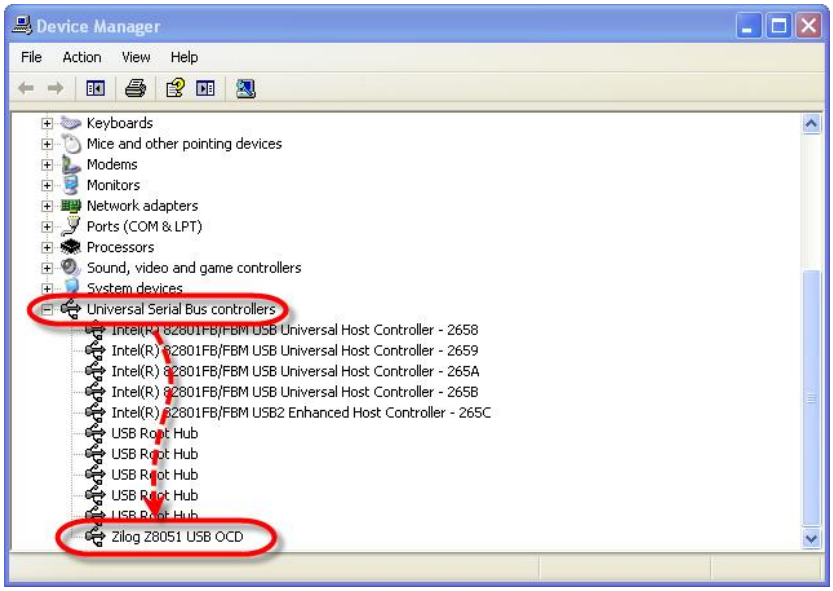
รูปแสดงตัวอย่าง การ Debug การทำงานของ MCU ผ่านโปรแกรม Zilog Z8051 OCD v1.147

การเชื่อมต่อ MCU กับ ET-Z8051 OCD ใน ISP Programming Mode

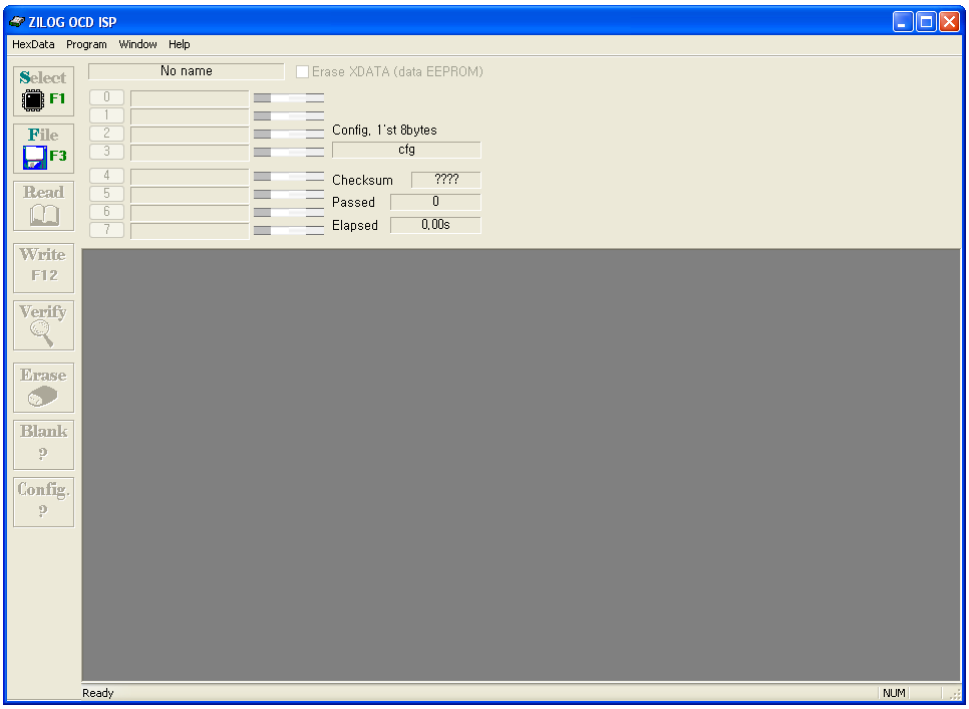
ในโหมดนี้ จะเหมาะกับการใช้งานในรูปแบบที่ผ่านพ้นขั้นตอนของการพัฒนาโปรแกรมเรียบร้อยไปแล้ว โดยในโหมดนี้จะทำหน้าที่ Program Hex Code ให้ MCU เพียงอย่างเดียว เหมาะสำหรับการโปรแกรม Hex Code สำหรับผลิตเป็นสินค้าจำนวนมากๆ เพราะสามารถทำได้อย่างรวดเร็ว โดยในโหมดนี้จะใช้งานร่วมกับโปรแกรม “Zilog Z8051 ISP” ซึ่งปัจจุบัน (สิงหาคม 2555) โปรแกรมจะได้รับการปรับปรุงเป็นรุ่น “Zilog Z8051 ISP Version1.147” โดยมีขั้นตอนการใช้งานพอสังเขปดังนี้

1. ทำการติดตั้งโปรแกรม Zilog Z8051 ISP ให้เรียบร้อย โดย Run File ชื่อ “Z8051_1.1.exe”
2. ทำการเชื่อมต่อสาย USB ของ ET-Z8051 OCD เข้ากับคอมพิวเตอร์ PC พร้อมกับทำการติดตั้ง Driver ของอุปกรณ์ให้เรียบร้อย ซึ่งขั้นตอนนี้จะกระทำเพียงครั้งแรกครั้งเดียวเท่านั้น โดยอุปกรณ์ “ET-Z8051 OCD” ของ อีทีที จะใช้ Driver ชุดเดียวกับ “Zilog Z8051 OCD” ของ Zilog Inc. ซึ่งตามปรกติ Driver จะถูกติดตั้งเตรียมไว้พร้อมกันกับการติดตั้งโปรแกรม “Z8051_1.1.exe” ในขั้นตอนที่1 ซึ่งถ้าติดตั้งโปรแกรมตามค่ามาตรฐาน Driver จะอยู่ที่ “..\device drivers\OCD USB\” ภายใต้ Directory ที่ทำการติดตั้งโปรแกรมไว้ โดยปัจจุบันจะมี Driver สำหรับรองรับกับระบบปฏิบัติการ Windows สำหรับเครื่องคอมพิวเตอร์ทั้งแบบ 32บิต และ 64บิต คือ
   a. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x32” สำหรับ 32บิต
   b. “C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\x64” สำหรับ 64บิต

   ซึ่งถ้าทุกอย่างถูกต้องเรียบร้อยแล้ว LED USB สีแดง ที่ตัวเครื่อง ET-Z8051 OCD จะติดสว่าง และเมื่อเข้าไปตรวจสอบที่ Device Manager จะต้องพบอุปกรณ์ USB ใน Tab ของ Universal Serial Bus controller ควรพบอุปกรณ์ชื่อ “Zilog Z8051 USB OCD” ดังตัวอย่าง



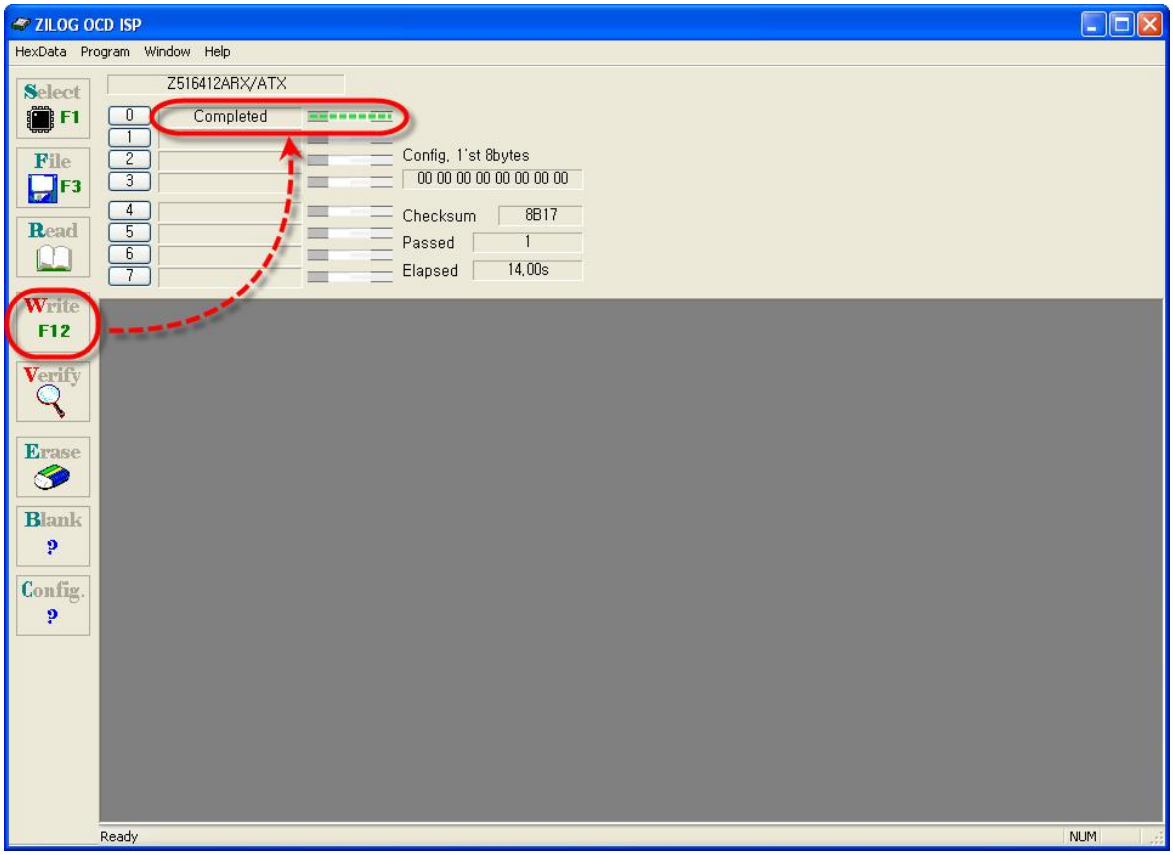
3. สั่ง Run โปรแกรม “Zilog Z8051 ISP V1.147” ให้เรียบร้อย ซึ่งเมื่อโปรแกรมเริ่มทำงานในครั้งแรกจะยังไม่มีค่าตัวเลือกใดๆแสดงบนหน้าจอโปรแกรมดังรูป



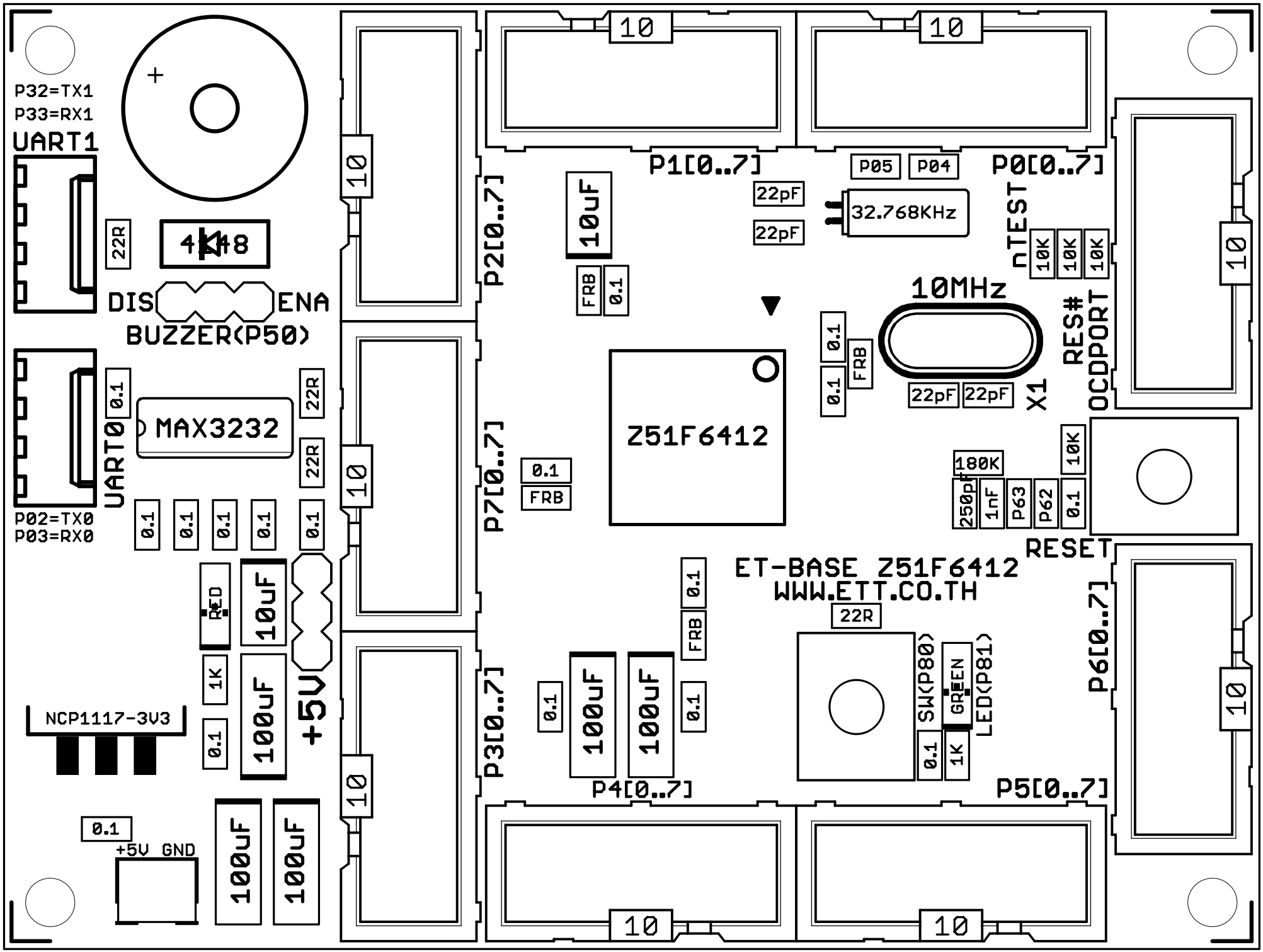
4. ถ้ามีการจ่ายไฟเลี้ยงวงจรให้บอร์ด ET-BASE Z51F6412 ไว้แล้วให้ทำการปลดสายไฟเลี้ยงของบอร์ด ET-BASE Z51F6412 ออกจากบอร์ดให้เรียบร้อยก่อน แล้วจึงทำการต่อสายแพร์ขนาด 10Pin ระหว่างเครื่อง ET-Z8051 OCD เข้ากับขั้วต่อ IDE10Pin สีเหลือง (PORT-OCD) ของบอร์ด ET-BASE Z51F6412 ให้เรียบร้อย จากนั้นจึงทำการจ่ายไฟเข้าบอร์ด ET-BASE Z51F6412 ให้เรียบร้อยแล้วให้ทำการเลือกกำหนดเบอร์ MCU โดยให้เลือกที่ “Select Device” แล้วเลือกกำหนดเบอร์ MCU เป็น “Z51F6412ARX/ATX” ดังรูป

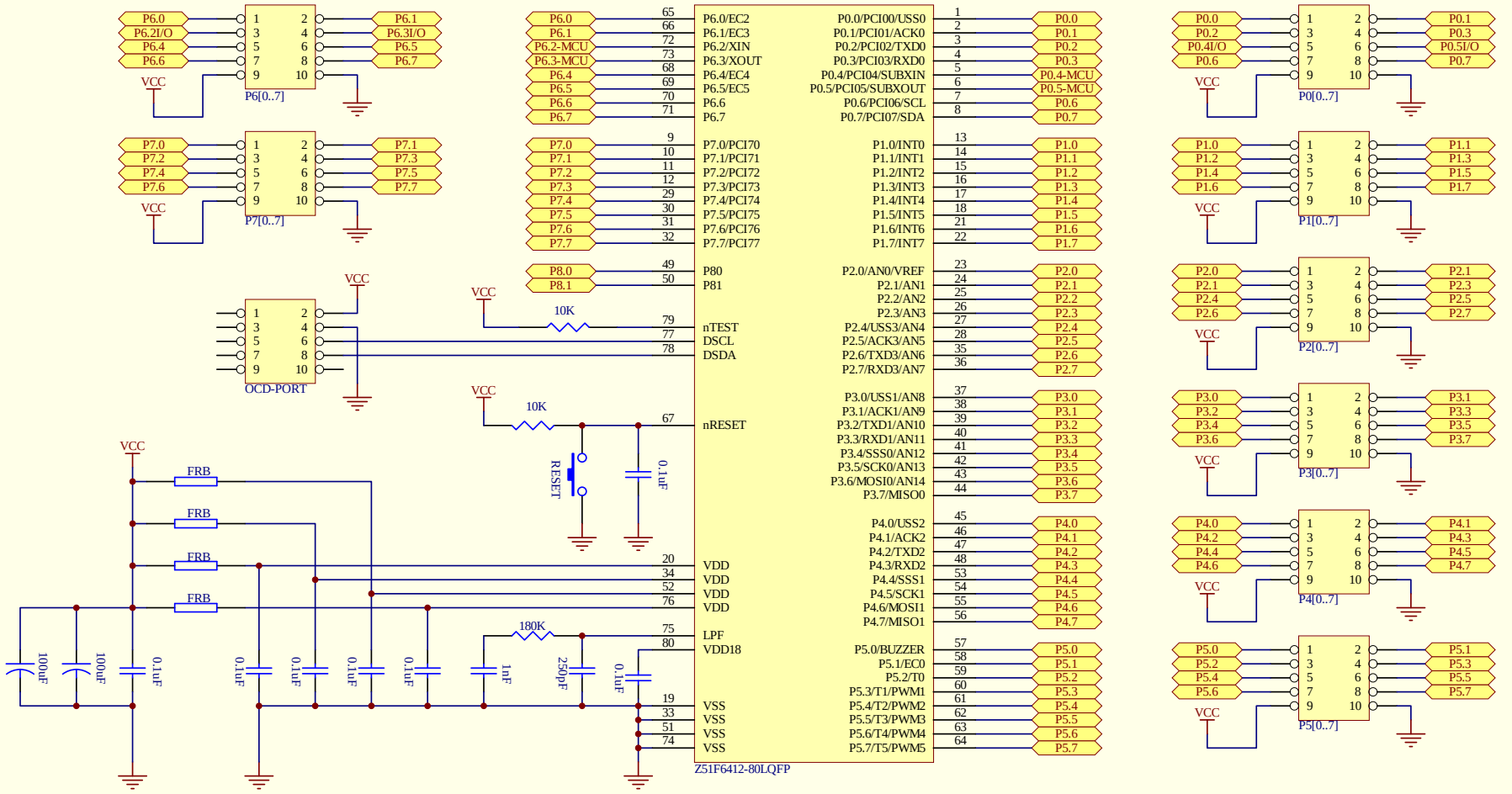
5. เมื่อทำการเลือกกำหนดเบอร์ได้เรียบร้อยแล้ว ให้ทำการสั่งเปิด Hex ไฟล์ที่ต้องการนำมาโปรแกรมให้กับ MCU โดยให้เลือกที่ “Load Code Hex File” จากนั้นจะปรากฏ Dialog Box ให้กำหนดรูปแบบการ Fill Buffer โดยให้ Fill เป็น FF ดังตัวอย่าง แล้วเลือกกำหนดชื่อและที่อยู่ของ Hex File ที่ต้องการให้เรียบร้อย ดังตัวอย่าง

6. หลังจากสั่งเปิด Hex File มาเตรียมไว้ใน Buffer ของโปรแกรมเรียบร้อยแล้ว ให้เลือก “Write” เพื่อเป็นการสั่งโปรแกรม Code ใน Buffer ที่โหลดไว้ให้กับหน่วยความจำใน MCU ให้รอจนโปรแกรมทำงานเสร็จเรียบร้อย จากนั้นให้ทำการปลดสาย OCD PORT ออกจากบอร์ดพร้อมกับปลดแหล่งจ่าย Power ออกจากบอร์ดแล้วรอชักครู่ 2-3 วินาที แล้วจ่ายไฟเลี้ยงให้กับบอร์ดใหม่ MCU ในบอร์ดก็จะเริ่มทำงานตามโปรแกรมที่ได้ทำการสั่งโปรแกรมไว้แล้วในทันที

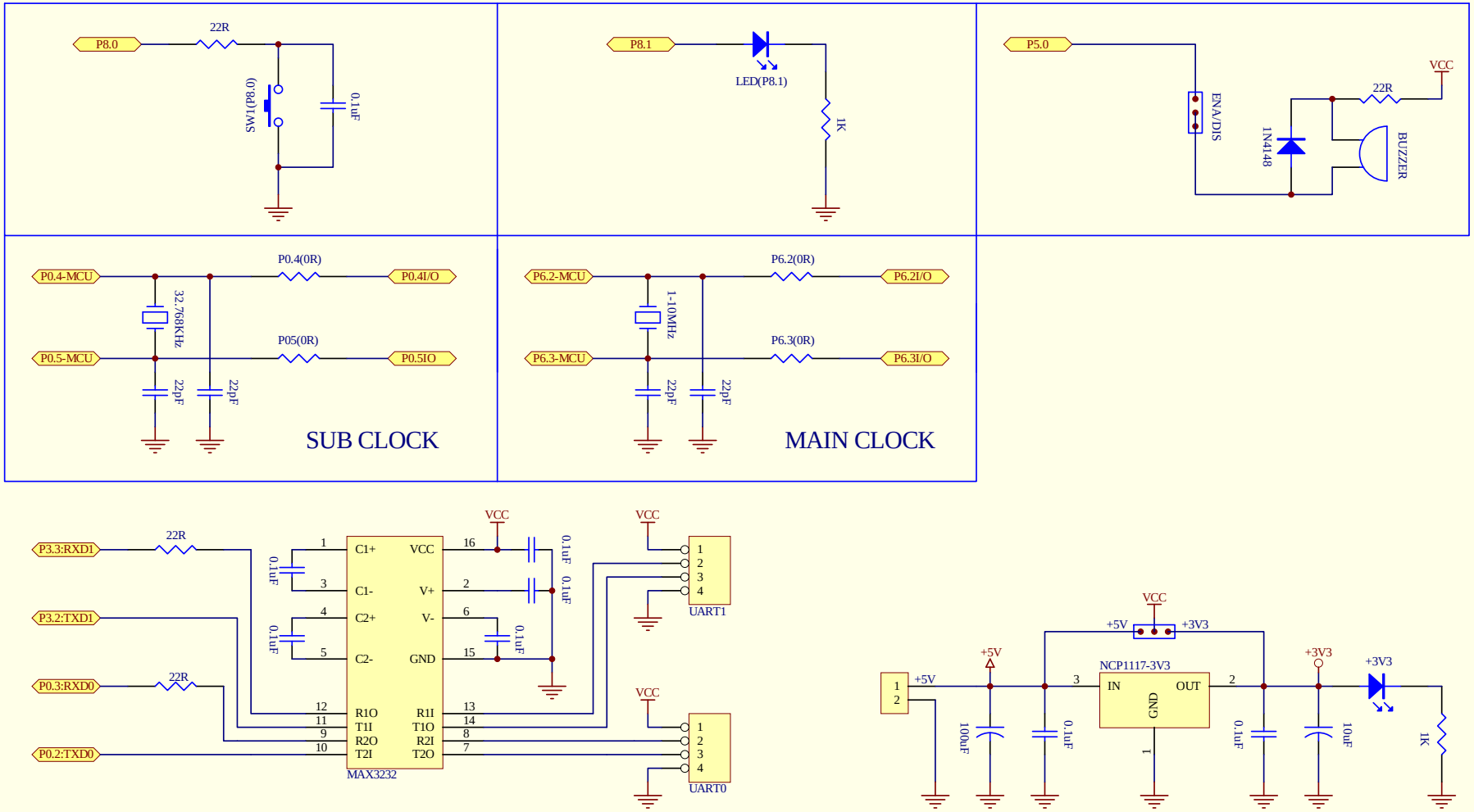


รูปแสดงตัวอย่างการ Program MCU ด้วย OCD ผ่านโปรแกรม Zilog Z8051 ISP v1.147


P32=TX1
P33=RX1
UART1
22R
4K48
DIS
ENA
BUZZER(P50)
UART0
0.1
MAX3232
22R
P02=TX0
P03=RX0
RED
1K
10uF
100uF
+5V
NCP1117-3V3
+5V GND
P1[0..7]
P2[0..7]
P7[0..7]
P3[0..7]
P4[0..7]
P5[0..7]
P6[0..7]
P0[0..7]
P05
P04
22pF
32.768KHz
nTEST
10K
10MHz
X1
RES#
OCDPORT
FRB
Z51F6412
180K
250pF
1nF
P63
P62
RESET
ET-BASE Z51F6412
WWW.ETT.CO.TH
SW(P80)
GREEN
LED(P81)
10



Title	ET-BASE Z51F6412	
Size: A4	Number: WWW.ETT.CO.TH	Revision: 1.0
Date: 28-Aug-2012		Sheet1 of2
File: G:\Portable Protel99se\Z8051\Z8051.Ddb		Drawn By:



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