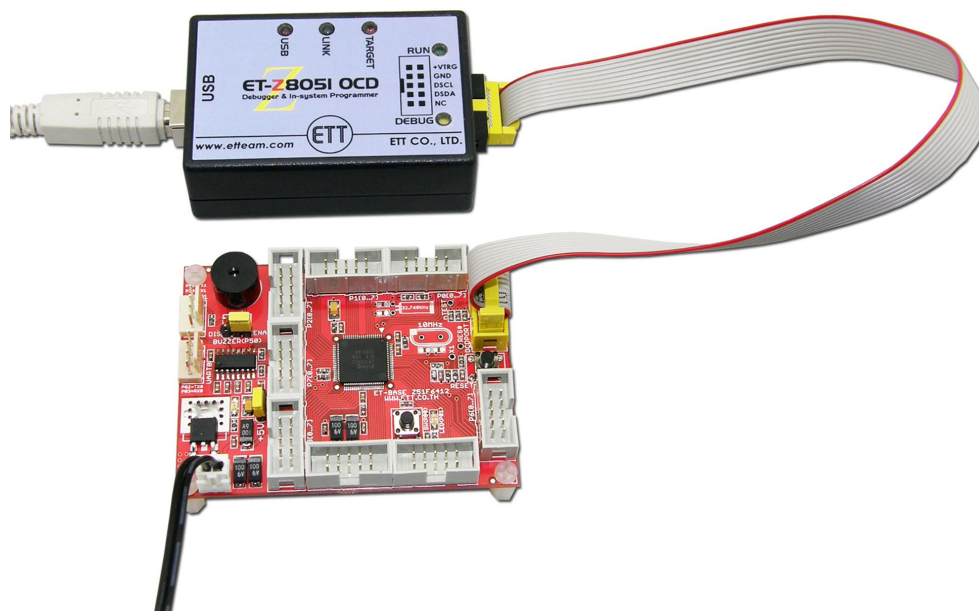
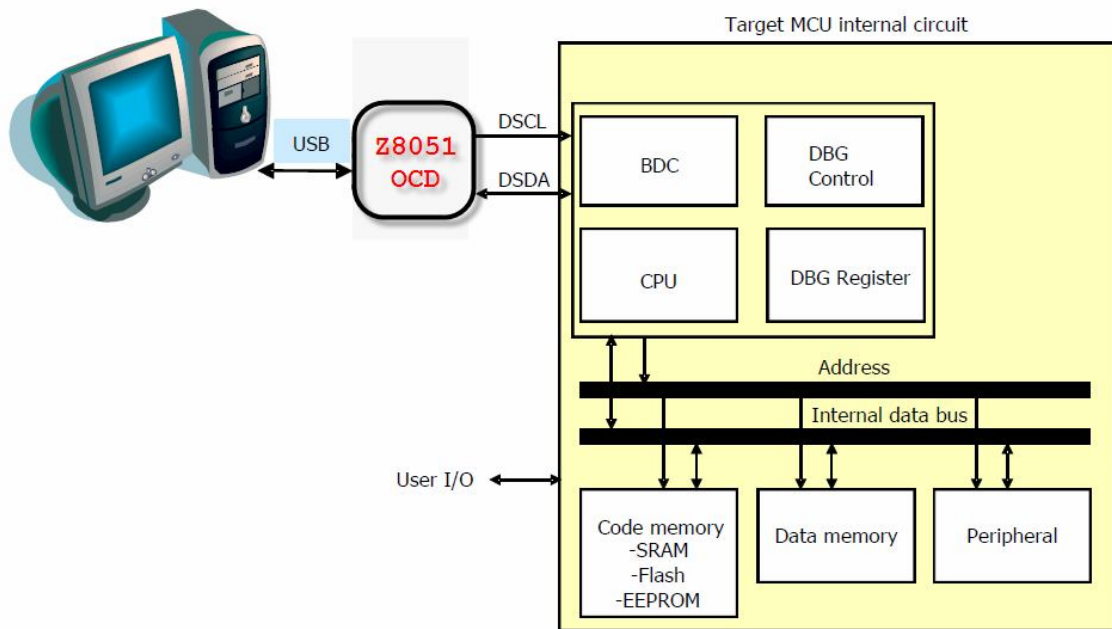


ET-Z8051 OCD

ET-Z8051 OCD เป็นเครื่องมือที่ได้รับการออกแบบมาสำหรับใช้พัฒนาโปรแกรม MCU ตระกูล Z8051 ของ Zilog Inc. โดยเป็นเครื่องมือทำหน้าที่เป็นตัวกลางในการสื่อสารกันระหว่างคอมพิวเตอร์ PC และ Target Board ที่ใช้ MCU ตระกูล Z8051 เพื่อให้ผู้ใช้งานสามารถ สั่งงาน ตรวจสอบ ทดสอบ ดูค่าของ รีจิสเตอร์ ค่าของข้อมูลในหน่วยความจำ ค่าของตัวแปร ของ MCU ในขณะที่ทำงานจริงได้ โดยสามารถทำหน้าที่ได้ทั้งการ Debugger และ In-system Programming(ISP)



โดย ET-Z8051 OCD จะใช้การเชื่อมต่อกับคอมพิวเตอร์ PC ผ่านทางพอร์ตสื่อสารแบบ USB และเชื่อมต่อกับ Target Board ของ MCU ผ่านทางหัวต่อ IDE 10PIN ด้วยสายแพร์ขนาด 10PIN โดยทำงานร่วมกับโปรแกรมซึ่งทำงานบนคอมพิวเตอร์ PC รองรับการทำงานบนระบบปฏิบัติการ Windows XP, Windows Vista 32Bit / 64Bit และ Windows-7 32Bit / 64Bit



คุณสมบัติ

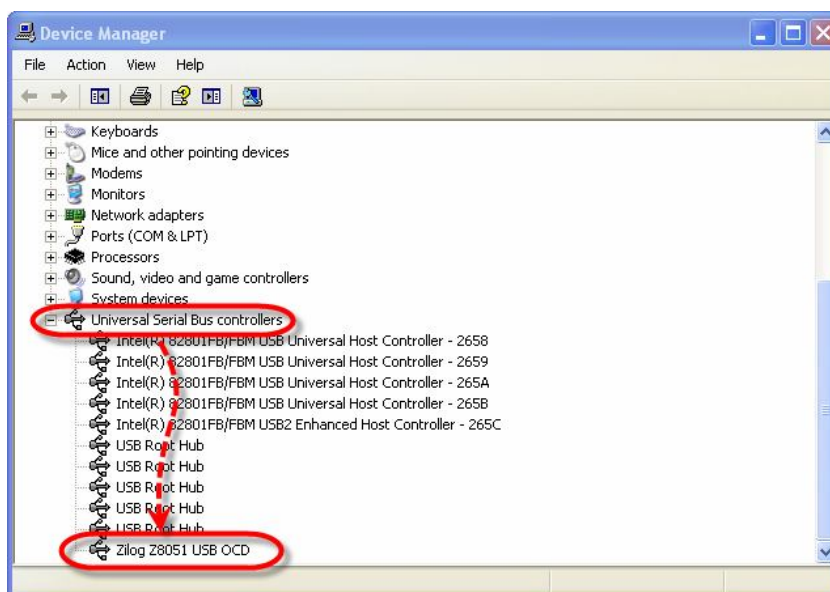
- สนับสนุนการใช้งานร่วมกับ MCU ตระกูล Z8051 ของ Zilog Inc.
- รองรับไฟล์ HEX และ map/symbol File
- สามารถทำการ Debug โดยแสดงค่าเป็นตัวแปรในโปรแกรมได้
- สามารถเข้าถึง Code ใน MCU ได้
- สนับสนุนการใช้งานแบบ In-system Programming ได้
- สามารถแสดงค่า Code และ Disassembler ได้
- สามารถกำหนดค่า PC(Program Counter) และ Break Point ได้
- สามารถแสดงผล และแก้ไขค่าของ ตัวแปร หน่วยความจำ และรีจิสเตอร์ต่างๆได้
- สามารถแสดง Code และค่าในหน่วยความจำ XDATA ได้
- สามารถทำการ Auto Detect เบอร์ MCU และปรับค่าพารามิเตอร์ต่างๆให้เองโดยอัตโนมัติ
- สนับสนุนการใช้งานกับ MCU ในระดับแรงดันตลอดย่านแรงดันมาตรฐานได้ (2V-5.5V)
- สนับสนุนการใช้งาน MCU ทั้ง Internal Clock และ External Crystal Clock
- สามารถสั่งงาน Run, Step, Auto Step, Break,...

การเชื่อมต่อ MCU กับ ET-Z8051 OCD ใน Debug Mode

ในโหมดนี้จะเหมาะกับการใช้งานในรูปแบบที่อยู่ในขั้นตอนของการพัฒนาโปรแกรมเพราะสามารถสั่งโปรแกรม MCU พร้อมทั้งควบคุม ทดสอบ หยุดการทำงาน รวมไปถึงการตรวจสอบและปรับแต่งค่ารีจิสเตอร์ต่างๆ ในขณะที่ 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 พร้อมทั้งทำการติดตั้ง Driver ของอุปกรณ์ให้เรียบร้อย ซึ่งขั้นตอนนี้จะกระทำเพียงครั้งแรกครั้งเดียวเท่านั้น โดยอุปกรณ์ ET-Z8051 OCD ของ อีทีที จะใช้ Driver ชุดเดียวกับ Zilog Z8051 OCD ของ Zilog Inc. ซึ่งตามปกติ Driver จะถูกติดตั้งเตรียมไว้พร้อมกันกับการติดตั้งโปรแกรม ชื่อ "Z8051_1.1.exe" ในขั้นตอนที่ 1 ซึ่งถ้าติดตั้งโปรแกรมตามค่ามาตรฐาน Driver จะอยู่ที่ "device drivers\OCD USB\" ภายใต้ Directory ที่ทำการติดตั้งโปรแกรมไว้ โดยปัจจุบันจะมี Driver สำหรับรองรับกับระบบปฏิบัติการ Windows สำหรับเครื่องคอมพิวเตอร์ทั้งแบบ 32bit และ 64bit คือ
 - a. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\32" สำหรับ 32bit
 - b. "C:\Program Files\Zilog\Z8051_1.1\device drivers\OCD USB\64" สำหรับ 64bit

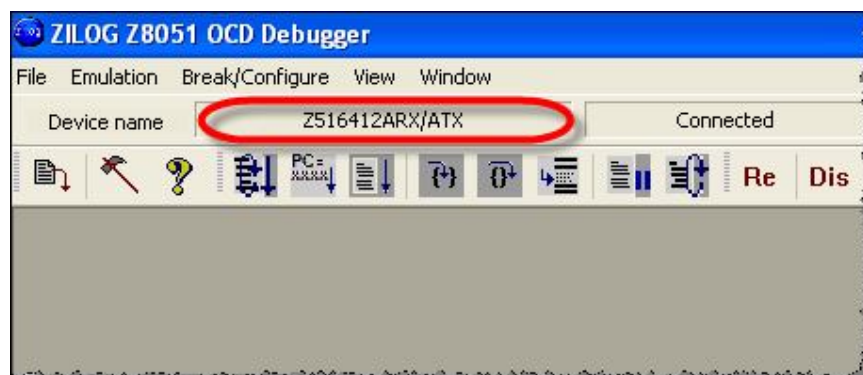
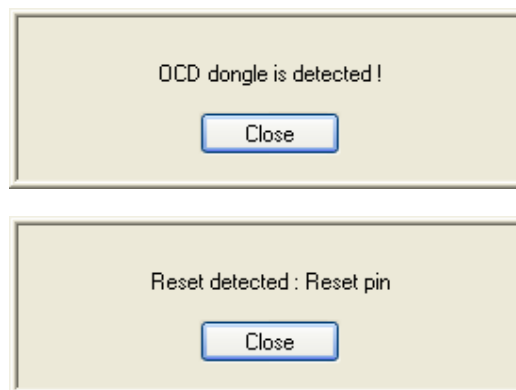
ซึ่งถ้าทุกอย่างถูกต้องเรียบร้อยแล้ว LED USB สีแดง ที่ตัวเครื่อง ET-Z8051 OCD จะติดสว่าง และเมื่อเข้าไปตรวจสอบที่ Device Manager จะต้องพบอุปกรณ์ USB ใน Tab ของ Universal Serial Bus controller ควรพบอุปกรณ์ชื่อ "Zilog Z8051 USB OCD" ดังตัวอย่าง



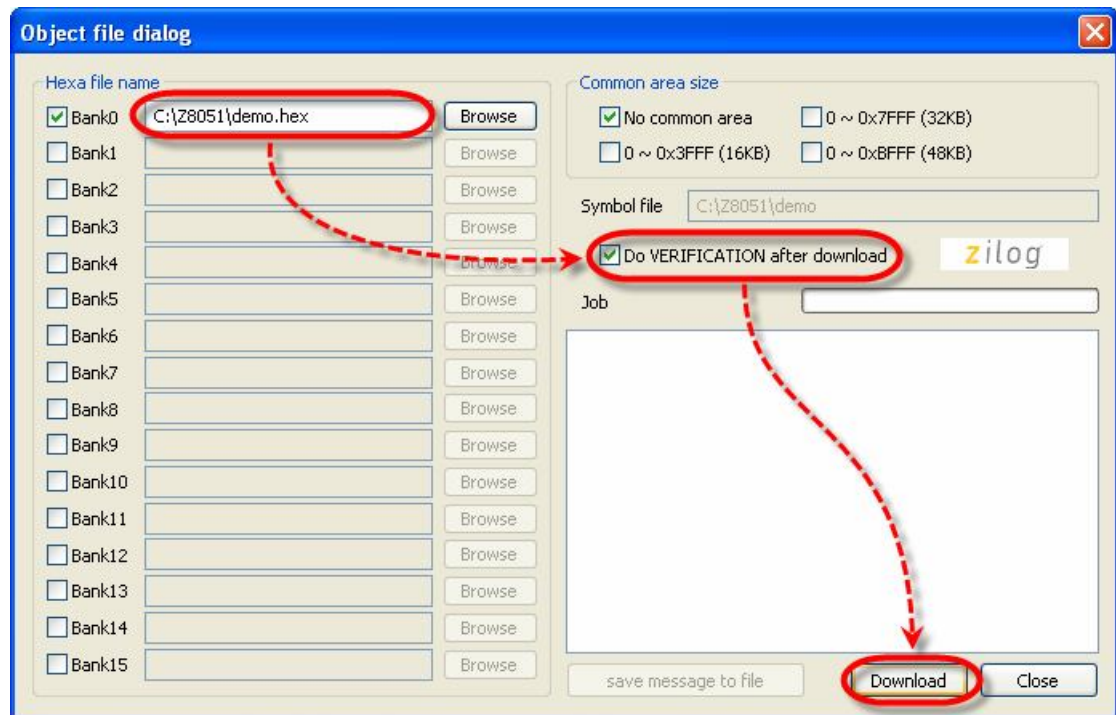
3. สั่ง Run โปรแกรม Zilog Z8051 OCD V1.147 ซึ่งเมื่อโปรแกรมเริ่มทำงานในครั้งแรก จะปรากฏ Dialog ข้อความแจ้งการตรวจพบ อุปกรณ์ Z8051 OCD ดังตัวอย่าง พร้อมกับที่อุปกรณ์ OCD ของ ET-Z8051 OCD จะปรากฏ LED LINK สีเขียว ติดสว่างให้เห็น เพื่อแสดงว่าอุปกรณ์ OCD พร้อมทำงาน



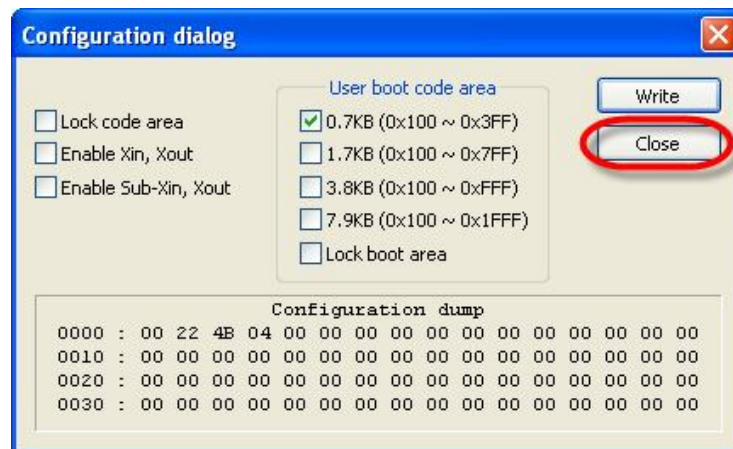
4. ถ้ามีการจ่ายไฟเลี้ยงวงจรให้บอร์ด ET-BASE Z51F6412 ไว้แล้วให้ทำการปลดสายไฟเลี้ยงของบอร์ด ET-BASE Z51F6412 ออกจากบอร์ดให้เรียบร้อยก่อน แล้วจึงทำการต่อสายแพรขนาด 10Pin ระหว่างเครื่อง ET-Z8051 OCD เข้ากับขั้วต่อ IDE10Pin สีเหลือง (PORT-OCD) ของบอร์ด ET-BASE Z51F6412 ให้เรียบร้อย จากนั้นจึงทำการจ่ายไฟเข้าบอร์ด 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



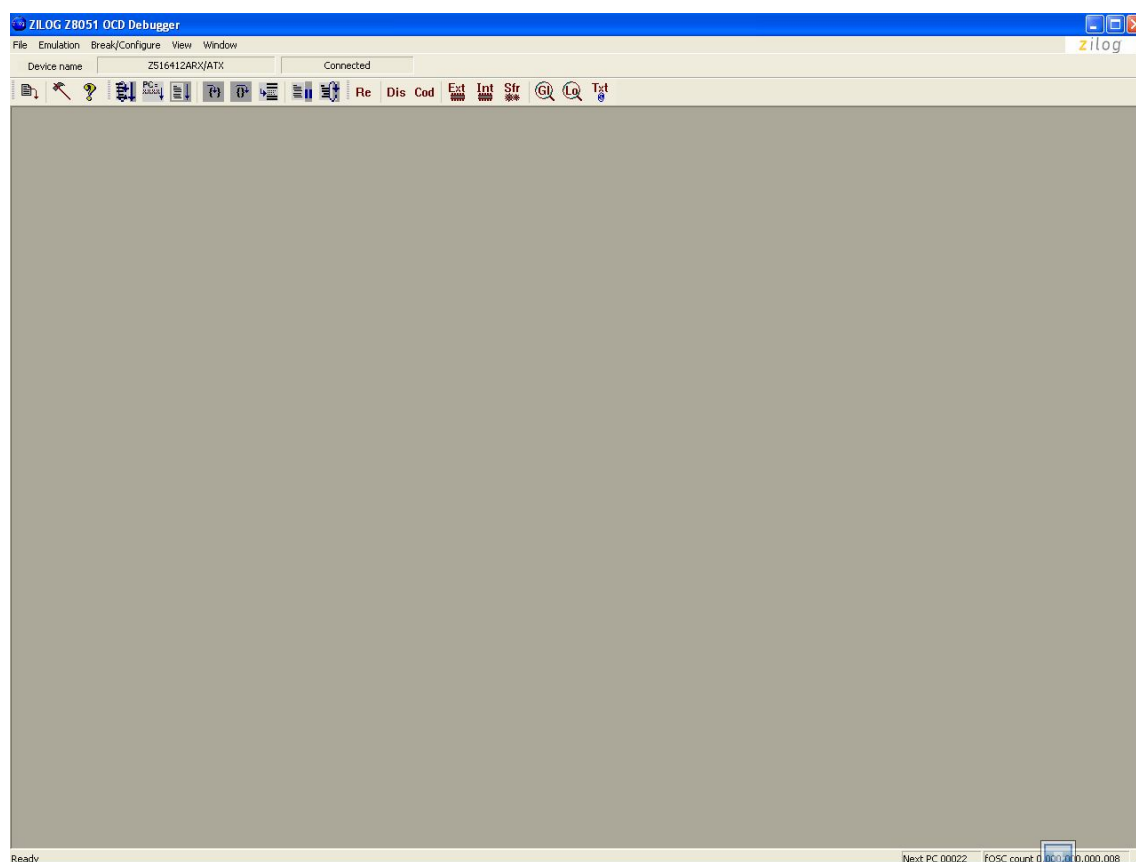
5. มาถึงขั้นตอนนี้ก็แสดงว่าการเชื่อมต่อระหว่าง MCU ในบอร์ด ET-BASE Z51F6412 กับเครื่อง OCD ของ ET-Z8051 OCD สามารถเชื่อมต่อกันได้อย่างถูกต้องเป็นที่เรียบร้อยแล้ว จากนั้นไปผู้ใช้ก็สามารถสั่งงานโปรแกรม Zilog Z8051 OCD ให้ทำงานต่างๆตามที่ต้องการได้แล้ว โดยขั้นตอนต่อไปที่ต้องทำคือทำการส่งโปรแกรม Hex File 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 จากนั้นก็ก็สามารถสั่งงาน MCU ให้ทำงานต่างๆตามต้องการได้ทันที ซึ่งอาจเป็นการสั่ง Run(Go) เพื่อดูผลการทำงานจริง หรือกำหนดตำแหน่งสำหรับหยุดการทำงาน (Break) หรือ สั่ง Run ทีละคำสั่ง(Step) เพื่อตรวจสอบผลการทำงานของโปรแกรมว่าถูกต้องตามที่ออกแบบไว้หรือไม่ ซึ่งในครั้งแรกโปรแกรมจะแสดงหน้าต่างโปรแกรมต่างๆ เนื่องจากเรายังไม่ได้กำหนดการแสดงผลของโปรแกรม ว่าต้องการให้โปรแกรมแสดงค่าของอะไรบ้าง ดังตัวอย่าง



โดยเราสามารถเลือกกำหนดให้หน้าจอของโปรแกรมแสดงค่าต่างๆของ Code โปรแกรม หรือ ตัวแปร และรีจิสเตอร์ต่างๆของ MCU ได้ตามต้องการ ซึ่งผู้ใช้สามารถปรับแต่งขนาดหน้าต่างการแสดงผลของส่วนต่างๆได้เองตามความเหมาะสม โดยเลือกที่เมนูคำสั่ง View แล้วเลือกสิ่งที่ต้องการแสดงผลหรือเลือกที่เมนูสัญลักษณ์ในส่วนของ Window open bar ดังตัวอย่าง



การแสดง Code คำสั่ง



การสั่งแสดง Code คำสั่ง ของโปรแกรม ให้เลือกที่เมนู View → Text file แล้วเลือกไฟล์ Source Code ที่เราต้องการดูค่า ซึ่งในที่นี้ให้เลือก main.c โดยโปรแกรมจะแสดงบรรทัด Code คำสั่งและตำแหน่งแอดเดรสของคำสั่งที่ได้จากการแปลคำสั่งให้เห็น ดังตัวอย่าง

```

main.c
Goto line Find Up Down
7 #define LED_OFF() LED_PORT_DATA &= ~(LED_PIN)
8 #define LED_ON() LED_PORT_DATA |= (LED_PIN)
9 #define LED_TOGGLE() LED_PORT_DATA ^= (LED_PIN)
10
11 void delay(unsigned int i)
12 {
0_0022 while(i > 0) {i--;}
14 return;
0_0032 }
16
17 void main(void)
18 {
0_0003 PLLCR = 0x00;
0_0006 SCCR = 0x24;
0_0009 LED_PORT_INIT();
22
23 while(1)
24 {
0_000C LED_OFF();
0_000F delay(20000);
0_0016 LED_ON();
0_0019 delay(20000);
0_0020 }
30 }
31

```

การแสดงค่า รีจิสเตอร์ และ SFR Register



การสั่งแสดงค่าของ Special Function Register ให้เลือกที่เมนู View → SFR dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า SFR Register พร้อมกับค่าในรีจิสเตอร์ต่างๆให้เห็นดังตัวอย่าง

Pattern	Refresh	Save	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0080			08	07	00	00	00	00	D1	00	00	00	24	85	8C	00	00	FF
0090			01	00	00	00	00	00	00	00	08	00	8F	01	E0	00	00	00
00A0			80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B0			01	00	00	00	00	00	00	00	00	00	00	00	00	00	FF	FF
00C0			0C	00	00	00	00	00	00	FF	FF	00	00	00	00	00	FF	FF
00D0			00	00	00	00	00	00	00	00	00	00	00	00	3F	3F	01	FF
00E0			00	00	00	00	00	00	80	FF	00	00	00	00	80	00	00	01
00F0			00	00	00	00	00	00	00	03	02	00	00	00	00	80	FF	00

การแสดงค่าของ Internal RAM



การสั่งแสดงค่าของ Internal RAM ให้เลือกที่เมนู View → IRAM dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า Internal RAM พร้อมกับค่าในหน่วยความจำให้เห็นดังตัวอย่าง

IRAM : 000

Pattern

Save

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	00	00	00	00	00	00	4E	1F	16	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0080	A1	DB	92	5B	ED	35	B9	09	52	87	E8	C7	AC	F2	57	E1
0090	C8	13	75	71	DA	49	5D	47	3B	94	65	78	62	36	49	6C
00A0	50	1A	6E	1C	00	F4	4C	B1	9C	0A	64	72	FD	BD	31	98
00B0	29	51	B6	B3	0F	12	54	41	8A	8A	94	88	09	52	62	CB
00C0	F8	FC	3F	94	A5	EF	CF	D8	72	C8	91	82	8A	32	CC	B6
00D0	92	AD	94	86	2C	F2	EC	54	D7	9F	79	33	00	1D	D2	3E
00E0	44	FC	A4	31	D2	40	B7	E2	D6	CA	82	8C	97	2D	EC	D4
00F0	E3	22	E8	43	99	7F	BA	D8	55	7D	09	C9	D5	85	3D	D1

การแสดงค่าของ External RAM(XRAM)



การสั่งแสดงค่าของ Internal RAM ให้เลือกที่เมนู View → XDATA dump ซึ่งโปรแกรมจะแสดงหน้าต่างแสดงค่า XRAM พร้อมกับค่าในหน่วยความจำให้เห็นดังตัวอย่าง

XDATA 0_0000 (-) Page CS 80A6

Bank #

00

Goto

Pattern

Load

Save

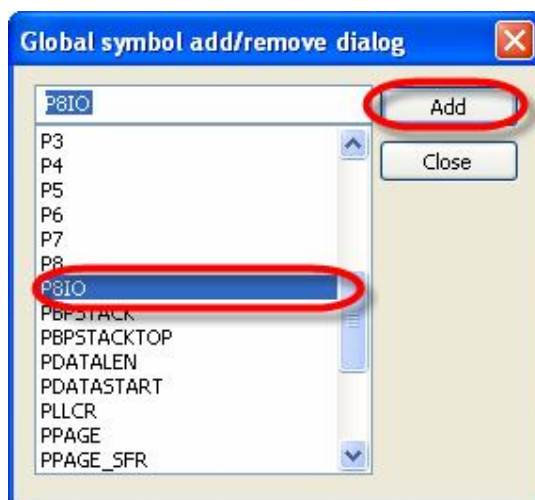
Fill

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0_0000	53	A8	61	D1	48	3E	7F	F6	72	B5	95	5E	0D	9E	17	03
0_0010	4A	15	BA	85	B0	67	9B	62	58	FD	2A	F6	EA	E5	2D	E2
0_0020	90	68	5C	A9	82	7F	48	4D	C0	B5	1D	75	B9	E6	95	DF
0_0030	94	27	2F	EA	81	72	BA	3E	FD	FF	A1	43	A4	7C	51	0E
0_0040	66	AB	0A	36	83	7C	19	1C	57	A4	F4	0F	BE	C2	56	28
0_0050	58	42	D4	A0	B2	10	D0	24	5A	84	BD	65	30	70	62	55
0_0060	21	41	7D	2D	F6	54	35	2C	2D	CE	29	44	EC	83	1A	89
0_0070	FA	8D	72	AA	6B	17	A6	D4	0D	77	66	09	31	D6	AE	36
0_0080	FA	6E	7A	95	6F	A2	74	15	7B	CE	A1	DE	F2	F3	15	C6
0_0090	6C	DC	8B	CC	93	DC	29	92	14	7F	9C	EF	BC	06	28	7E
0_00A0	B8	5B	4B	B4	C1	2E	F1	B7	92	FD	92	4F	94	AF	29	FE
0_00B0	8C	EA	C6	8D	A4	89	A6	CA	85	6D	D1	82	94	70	A4	0B
0_00C0	91	30	CF	BC	4F	E0	F0	99	3C	C9	18	A5	EA	1B	16	FE
0_00D0	8A	7F	64	19	7A	C5	C5	4A	F7	76	8C	39	D0	E8	ED	92
0_00E0	E0	17	03	55	0E	59	24	46	32	05	F8	15	45	DC	57	DE
0_00F0	04	D0	B3	CA	2E	35	11	63	13	CE	B8	4A	91	DC	A8	C6

การแสดงค่าตัวแปร Global



การแสดงค่าตัวแปร Global เลือกเมนู View → Watch Global จากนั้นเลือกหัวข้อ Add symbol แล้วคลิกเมาส์เลือกชื่อตัวแปร หรือ รีจิสเตอร์ที่เราต้องการตรวจสอบค่าแล้วเลือก Add หรือถ้าต้องการปิดการแสดงผลของรีจิสเตอร์ตัวใดก็สามารถยกเลิกโดยเลือกที่ Remove symbol ตามต้องการ ดังตัวอย่าง

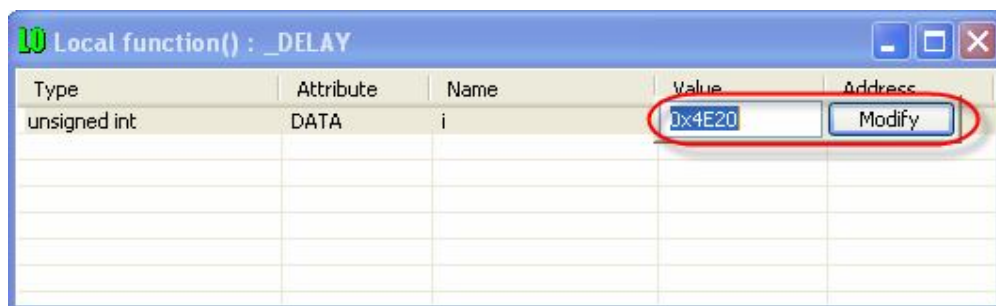
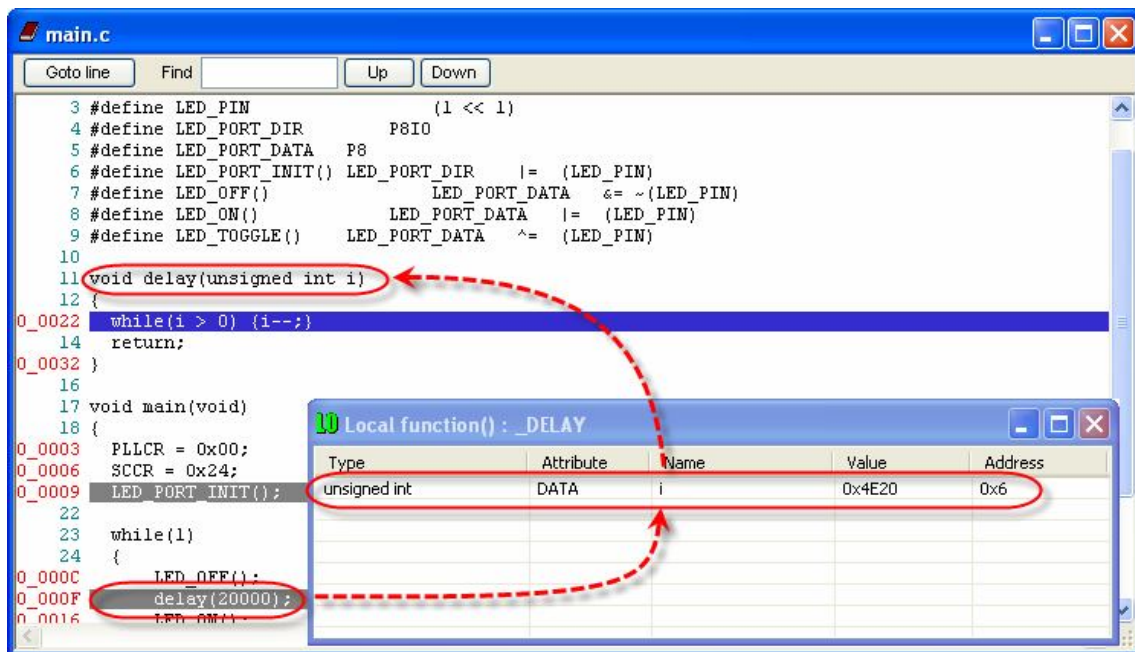


GL Global variables				
Add symbol		Remove symbol		
Type	Attribute	Name	Value	Address
unsigned char	DATA	SCCR	0x24	0x8A
unsigned char	DATA	PLLCR	0x0	0xD9
unsigned char	DATA	P8IO	0x2	0xD1
unsigned char	DATA	P8	0x0	0xD8

การแสดงค่าตัวแปร Local

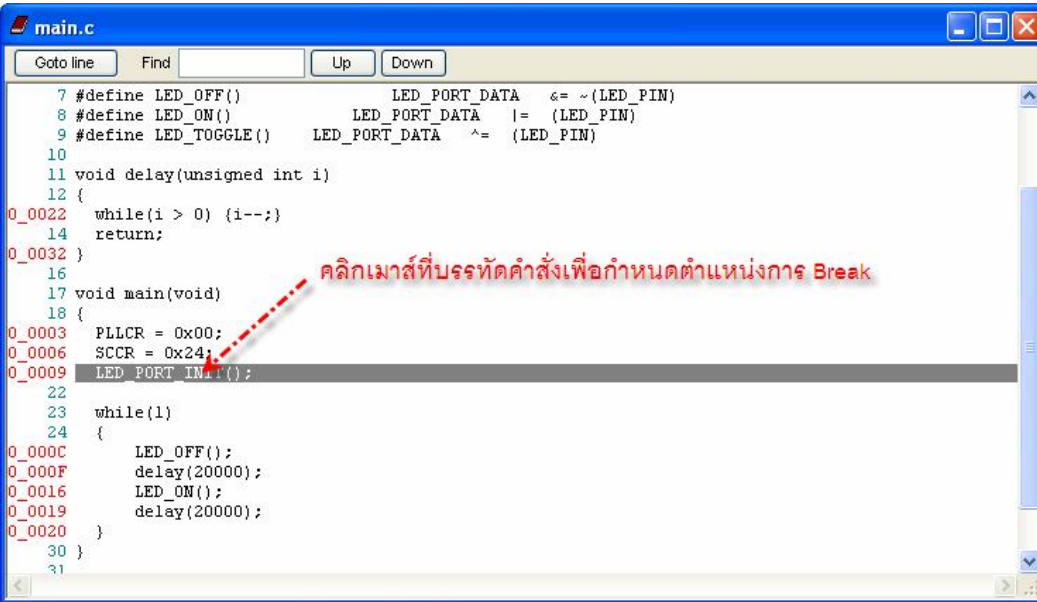


การแสดงค่าตัวแปร Local เลือกเมนู View → Watch Local ซึ่งโปรแกรมจะแสดงหน้าต่างสำหรับแสดงค่าตัวแปร Local ที่ใช้ในโปรแกรม โดยแสดงชนิดตัวแปร ชื่อตัวแปร ตำแหน่งหน่วยความจำที่ใช้สร้างตัวแปร และค่าในตัวแปรให้เราทราบ ซึ่งตัวแปรเหล่านี้จะแสดงค่าให้เราทราบเมื่อโปรแกรมทำงานถึงบรรทัดคำสั่งที่มีการผ่านค่าให้ตัวแปรแล้วเท่านั้น เช่น เมื่อมีการเรียกใช้ฟังก์ชันหน่วงเวลา Delay จะมีการผ่านค่าตัวแปรไปให้กับตัวแปร i ซึ่งประกาศเป็น unsigned int ไว้ ก็จะปรากฏค่ารายละเอียดของตัวแปรนี้ให้เราเห็น ซึ่งเราสามารถสั่งหยุด และเข้าไปแก้ไขค่าให้กับตัวแปร เพื่อทดสอบการทำงานให้กับตัวแปรได้ เช่น เราอาจทดสอบปรับค่า Value ให้กับ ตัวแปร i เพื่อทดสอบค่าการหน่วงเวลาที่เหมาะสมดูได้ โดยคลิกเมาส์ที่ช่อง Value และแก้ไขค่าตามที่ต้องการแล้วเลือก Modify ดังตัวอย่าง



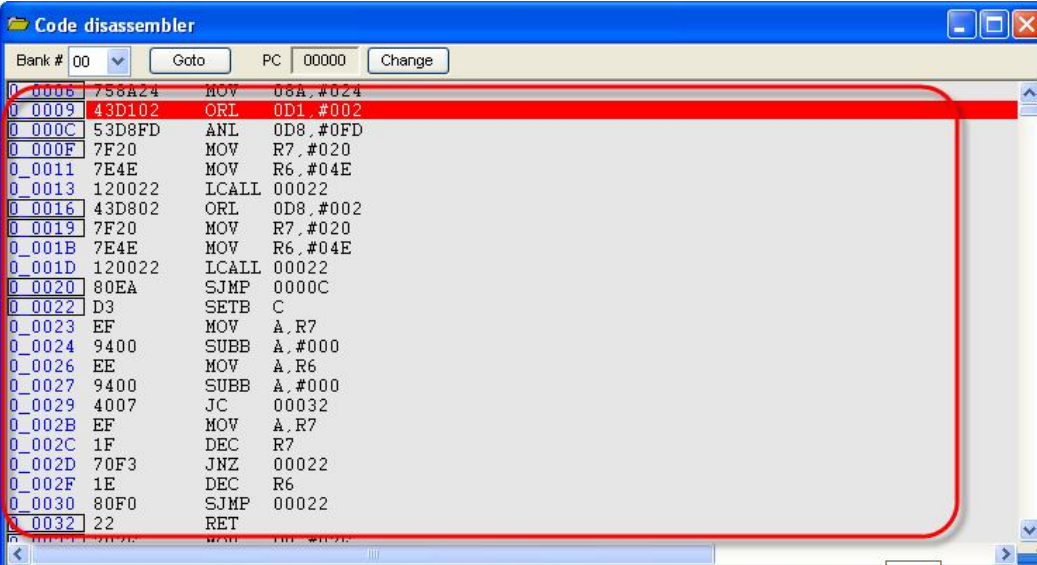
การกำหนดตำแหน่ง Break

การกำหนดตำแหน่ง Break เพื่อหยุดการทำงานของโปรแกรมในบรรทัดคำสั่งที่เราต้องการเพื่อดูว่าเมื่อ MCU ทำงานถึงบรรทัดคำสั่งที่เรากำหนดแล้ว ค่าของรีจิสเตอร์ และ ตัวแปรต่างๆ มีค่าเป็นอย่างไร ถูกต้องหรือไม่ โดยเราสามารถกำหนดตำแหน่งการ Break ได้จากหน้าต่างของ Text File ที่แสดงค่าของ Source Code โดยถ้าต้องการกำหนดการ Break ที่บรรทัดคำสั่งใดก็ให้คลิกเมาส์ที่บรรทัดคำสั่งนั้น ถ้าต้องการยกเลิกก็คลิกเมาส์ซ้ำที่ตำแหน่งเดิม ซึ่งถ้าเราสั่งแสดงค่าของ Code dissembler ไว้ด้วย ที่หน้าต่างของ Code disable ก็จะมีแถบสี แสดงบรรทัดคำสั่งภาษา Assemble ที่ตำแหน่งแอดเดรสเดียวกันกับบรรทัดคำสั่งที่เราสั่ง Break ด้วย ดังตัวอย่าง



```

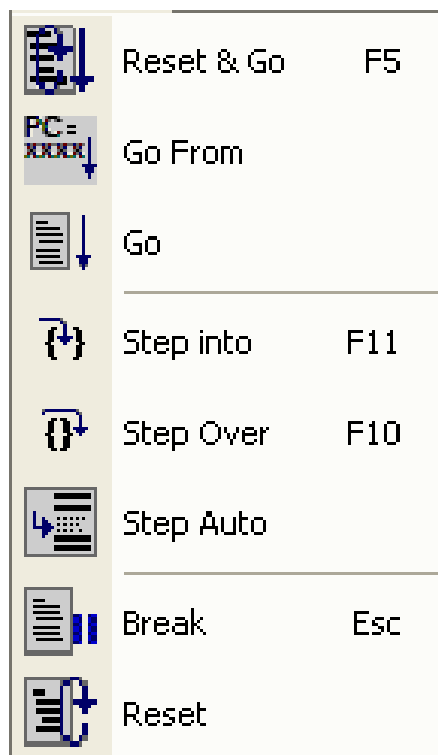
main.c
Goto line Find Up Down
7 #define LED_OFF() LED_PORT_DATA &= ~(LED_PIN)
8 #define LED_ON() LED_PORT_DATA |= (LED_PIN)
9 #define LED_TOGGLE() LED_PORT_DATA ^= (LED_PIN)
10
11 void delay(unsigned int i)
12 {
0_0022 while(i > 0) {i--;}
14 return;
0_0032 }
16
17 void main(void)
18 {
0_0003 PLLCR = 0x00;
0_0006 SCCR = 0x24;
0_0009 LED_PORT_INIT();
22
23 while(1)
24 {
0_000C LED_OFF();
0_000F delay(20000);
0_0016 LED_ON();
0_0019 delay(20000);
0_0020 }
30 }
31
  
```



Bank #	PC	Address	Instruction
00	00000	0_0006	756A24 MOV 08A,#024
00	00009	43D102	ORL 0D1,#002
00	0000C	53D8FD	ANL 0D8,#0FD
00	0000F	7F20	MOV R7,#020
00	00011	7E4E	MOV R6,#04E
00	00013	120022	LCALL 00022
00	00016	43D802	ORL 0D8,#002
00	00019	7F20	MOV R7,#020
00	0001B	7E4E	MOV R6,#04E
00	0001D	120022	LCALL 00022
00	00020	80EA	SJMP 0000C
00	00022	D3	SETB C
00	00023	EF	MOV A,R7
00	00024	9400	SUBB A,#000
00	00026	EE	MOV A,R6
00	00027	9400	SUBB A,#000
00	00029	4007	JC 00032
00	0002B	EF	MOV A,R7
00	0002C	1F	DEC R7
00	0002D	70F3	JNZ 00022
00	0002F	1E	DEC R6
00	00030	80F0	SJMP 00022
00	00032	22	RET

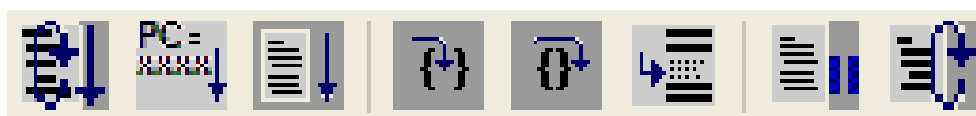
การสั่งงานโปรแกรม

การสั่งงานโปรแกรม Z8051 OCD สามารถเลือกที่เมนู **Emulation** แล้วเลือกหัวข้อคำสั่งที่ต้องการ หรือสามารถเลือกที่เมนูสัญลักษณ์ของ **Emulation Toolbar** โดยตรงก็ได้ โดยจะต้องทำการสั่งแสดงผลของ **Emulation Toolbar** ในเมนู **View** ให้อยู่ดังตัวอย่าง

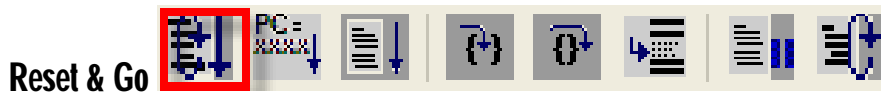


การแสดง **Emulation Toolbar** ในเมนู **View**

เมนูคำสั่งใน **Emulation**



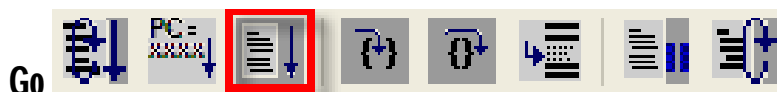
Emulation Toolbar



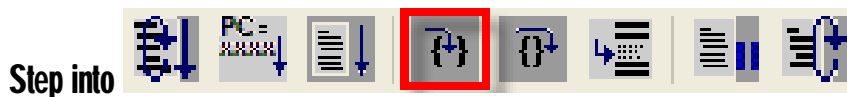
คำสั่งนี้ใช้สำหรับสั่งให้ **Reset** และ **Run** ด้วยความเร็วจริง โดยโปรแกรมจะสั่ง **Reset** การทำงานของ **MCU** และสั่งให้ **MCU** เริ่มต้นทำงานจากตำแหน่งแอดเดรส **0x0000**



คำสั่งนี้ใช้สั่ง **Run** ด้วยความเร็วจริง แต่สามารถกำหนดตำแหน่งแอดเดรสเริ่มต้นที่ต้องการให้โปรแกรมทำงานได้



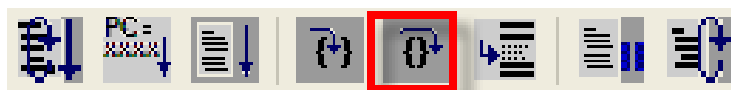
คำสั่งนี้ใช้สั่ง **Run** ด้วยความเร็วจริง โดยโปรแกรมจะเริ่มต้นทำงานต่อจากตำแหน่งแอดเดรสที่ **PC** ชี้อยู่ในขณะนั้น



ใช้สำหรับสั่ง **Run** โปรแกรมทีละคำสั่งเพื่อตรวจสอบผลการทำงานของโปรแกรมโดยละเอียด โดยเมื่อ **MCU** ทำงานตามคำสั่งเสร็จเรียบร้อยแล้วก็จะหยุดการทำงานเพื่อให้เราตรวจสอบผลการทำงาน โดยจะแสดงออกทั้งทาง **Hardware** และ **Software** โดยทาง **Hardware** ก็จะเกิดจากเปลี่ยนแปลงตามความเป็นจริงที่เกิดขึ้น ส่วนทาง **Software** ก็จะเกิดการเปลี่ยนแปลงทางรีจิสเตอร์และ หน่วยความจำ

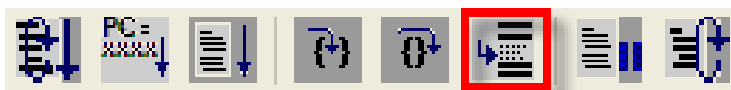
โดยคำสั่งนี้เหมาะสำหรับใช้ตรวจสอบความถูกต้องสมบูรณ์ของการทำงานของโปรแกรมน้อยหรือฟังก์ชันย่อยของโปรแกรมที่เขียนขึ้นโดยแยกทดสอบเป็นส่วนๆ

Step Over



ใช้สำหรับสั่ง Run โปรแกรมทีละคำสั่ง ใช้สำหรับตรวจสอบการทำงานของ Main Program เพื่อทดสอบวงจรการทำงานของโปรแกรมว่าถูกต้องดีหรือไม่ เหมาะสำหรับ ใช้ตรวจสอบผลการการทำงานของโปรแกรม Main หลังจากทำการทดสอบการทำงานของแต่ละฟังก์ชันย่อยๆเสร็จสมบูรณ์หมดแล้ว

Step Auto



คำสั่งนี้ใช้สั่ง Run โปรแกรมแบบทีละ Step แบบต่อเนื่องกันไป โดยจะมีการแสดงค่าของรีจิสเตอร์ ตัวแปร และหน่วยความจำต่างๆ ที่เปลี่ยนแปลงไปจากการทำงานของคำสั่งให้เห็นตลอดเวลาด้วย ซึ่งการทำงานของโปรแกรมจะทำงานเป็น Step ดังนั้นความเร็วในการทำงานจะเกิดการสะดุด เป็นจังหวะการทำงานของโปรแกรมจึงช้ากว่าการ Run ตามปกติ

Break

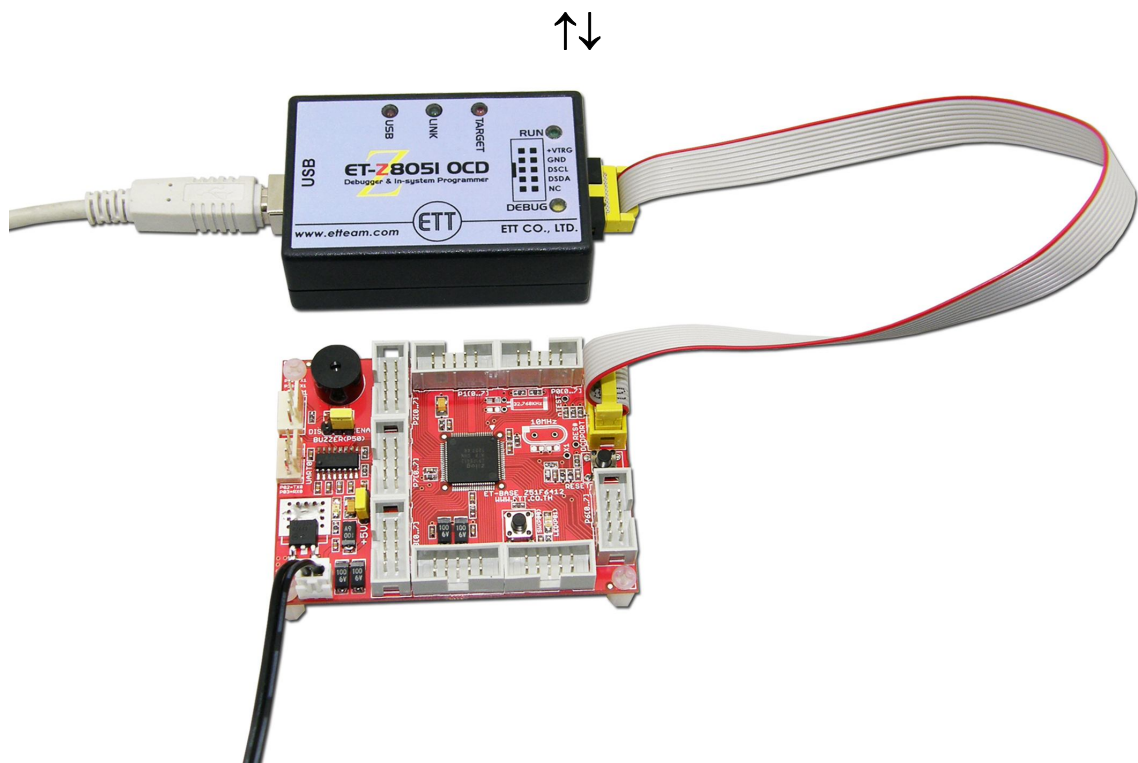
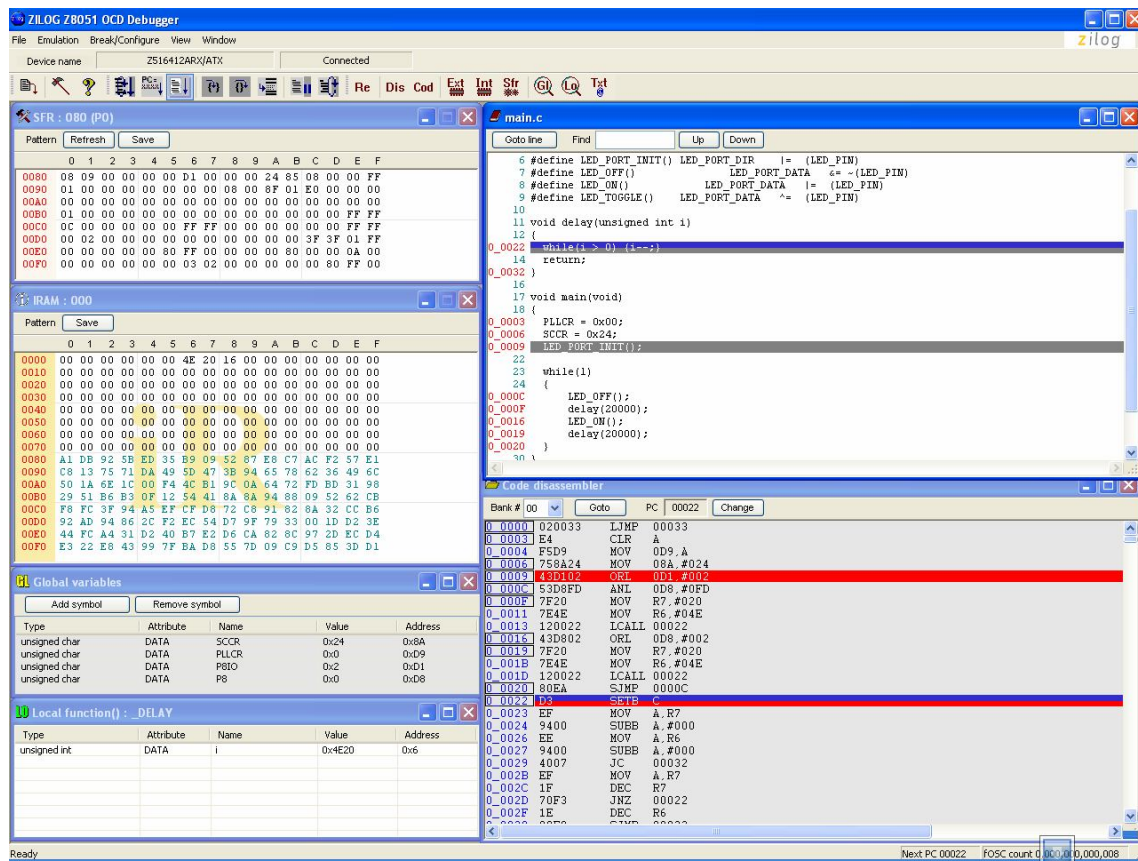


คำสั่งนี้ใช้สำหรับสั่งหยุดการทำงานของโปรแกรมที่กำลัง Run อยู่ในขณะนั้น ใช้สำหรับสั่งหยุดการทำงานของโปรแกรมเพื่อตรวจสอบค่า เปลี่ยนแปลง แก๊สค่า ของรีจิสเตอร์ ตัวแปร และหน่วยความจำต่างๆ

Reset



คำสั่งนี้ใช้สำหรับสั่งรีเซ็ตการทำงานของ MCU และกำหนดค่า PC ให้กลับไปเริ่มต้นเตรียมพร้อมที่ตำแหน่งแอดเดรส 0x0000



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