

การควบคุมแบนด์วิดท์สำหรับหลายผู้ใช้บริการ
MULTI USER BANDWIDTH CONTROLLER

โดย

นางสาวดิษยา	วงศ์บุญยิ่ง
นางสาวเต็มสิริ	แสงนวกิจ
นางสาวทองดี	สินปุณย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การควบคุมแบนด์วิดท์สำหรับหลายผู้ใช้บริการ
MULTI USER BANDWIDTH CONTROLLER

โดย

นางสาวดิษยา วงศ์บุญยิ่ง
นางสาวเต็มสิริ แสงนวกิจ
นางสาวทองดี สิ้นบุญย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ปริญญาโทปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมแบนด์วิดท์สำหรับหลายผู้ใช้บริการ

MULTI USER BANDWIDTH CONTROLLER

ผู้จัดทำ

1. นางสาวดิชยา วงศ์บุญยิ่ง 53010544
2. นางสาวเต็มสิริ แสงนวกิจ 53010555
3. นางสาวทองดี สินบุญ 53010580

..... อาจารย์ที่ปรึกษา
(รศ.ดร.สุวิมล สิทธิชีวภาค)

..... อาจารย์ที่ปรึกษา
(รศ.เกรียงไกร วงศ์โรจนภรณ์)

กิตติกรรมประกาศ

ปริญญานิพนธ์ครั้งนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำและคำปรึกษาจากอาจารย์ที่ปรึกษา รศ.ดร.สุวิพล สิริชีวภาค และรศ.เกรียงไกร วงโรจนภรณ์ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงและขอบคุณเพื่อนๆที่คอยให้กำลังใจ และคำปรึกษาเกี่ยวกับโครงการ และขอขอบคุณเจ้าหน้าที่ทุกท่านที่สำนักบริการคอมพิวเตอร์สำหรับคำแนะนำต่างๆจนสามารถทำปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณ บิดา มารดา และครอบครัวของผู้จัดทำที่ให้การสนับสนุนช่วยเหลือผู้จัดทำ และให้กำลังใจตลอดมา ผู้จัดทำหวังว่าปริญญานิพนธ์เรื่องนี้จะมียุทธศาสตร์และสามารถนำไปประยุกต์ใช้สืบต่อไปได้

ดิษยา วงศ์บุญยิ่ง
เต็มสิริ แสงนวกิจ
ทองดี สิ้นบุญย์

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1	1
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
บทที่ 2	3
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51(MCS-51)	3
2.2 โปรแกรม FLIP	4
2.3 โปรแกรม KEIL U VISION3	6
2.4 การสื่อสารข้อมูลแบบอนุกรม	7
2.4.1 การสื่อสารข้อมูลแบบอนุกรม	7
2.4.2 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	8
2.4.3 มาตรฐานอาร์เอส 232	12
2.4.4 ทีทีแอล	12
2.4.5 การแปลงระดับแรงดัน	13
2.5 PPPOE	14
2.6 ภาษาจาวา	15
2.6.1 ข้อดีของภาษาจาวา	15
2.6.2 ข้อเสียของภาษาจาวา	16
2.7 ระบบปฏิบัติการ LINUX	16
2.8 SHELL SCRIPT	17

สารบัญ(ต่อ)

	หน้า
2.8.1 ตัวแปร (VARIABLE)	18
2.8.2 TEST และ EXPRESSION	23
2.8.3 การใช้เงื่อนไข (IF)	24
2.8.4 การใช้เงื่อนไข (CASE หรือ SWITCH)	24
2.8.5 การใช้ LOOP WHILE	25
2.8.6 การใช้ LOOP FOR	25
2.9 โมดูลแอลซีดี (LIQUID CRYSTAL DISPLAY)	25
2.9.1 คำสั่งควบคุมโมดูลที่สำคัญ	26
2.10 คีย์แพด ET-TOUCH 4x4	28
บทที่ 3	
การออกแบบและการจัดทำปฏิญญานิพนธ์	29
3.1 การออกแบบ	29
3.1.1 การออกแบบการทำงานรวมของระบบ	29
3.1.1.1 ส่วนของ PPPOE SERVER	30
3.1.1.2 ส่วนของไมโครคอนโทรลเลอร์	31
3.1.2 การออกแบบการทำงานของชุดคำสั่งของระบบ	31
3.1.3 การออกแบบหน้าจอแอลซีดี (LCD)	32
3.2 อุปกรณ์ที่ใช้ในการทดลอง	33
3.3 การจัดเก็บผลการทดลอง	33
3.3.1 การทำงานของชุดคำสั่งสำหรับเช็คค่าข้อมูลในฝั่งแอดมิน	33
3.3.2 การทำงานในฝั่งผู้ใช้งาน (USER)	34
บทที่ 4	
ผลการทดลอง	35
4.1 ผลการทดลองการทำงานของชุดคำสั่งสำหรับเช็คค่าข้อมูลในฝั่งแอดมิน	35
4.1.1 การตั้งค่า IP ADDRESS ให้กับ UPLINK	36

สารบัญ(ต่อ)

	หน้า
4.1.2 การตั้งค่า IP ADDRESS ให้กับ DOWNLINK	37
4.1.3 การตั้งค่า SPEED ของ UPLOAD และ DOWNLOAD	38
4.2 ผลการทดลองการทำงานในฝั่งผู้ใช้งาน (USER)	41
บทที่ 5	
สรุปผลและข้อเสนอแนะ	49
5.1 สรุปผล	49
5.2 ข้อเสนอแนะ	49

บรรณานุกรม

ภาคผนวก

สารบัญรูป

รูปที่	หน้า
2.1 การจัดวางขาของAT89C51RE2	3
2.2 หน้าต่าง ATMEL FLIP	4
2.3 หน้าต่าง DEVICE SELECTION สำหรับเลือกเบอร์ไมโครคอนโทรลเลอร์ที่ใช้งาน	5
2.4 หน้าต่าง RS232 SETUP สำหรับเลือก COM PORT และ BAUD RATE ในการดาวน์โหลด	5
2.5 หน้าต่าง ATMEL FLIP หลังจาก CONNECT PORT สำเร็จ	5
2.6 หน้าต่าง TIMEOUT ERROR หลังจาก CONNECT PORT เมื่อCONNECT PORT ไม่สำเร็จ	6
2.7 HEX FILE ที่ถูกโหลดเข้า	6
2.8 สัญญาณรับและส่งข้อมูลในโหมด 0	8
2.9 สัญญาณการส่งข้อมูลในโหมดหนึ่ง	9
2.10 สัญญาณการส่งข้อมูลในโหมดสอง	9
2.11 การเรียงขาของไอซีแม็กซ์ 232	13
2.12 การเชื่อมต่ออีเทอร์เน็ตผ่าน PPPOE	14
2.13 โปรแกรมจาวา	15
2.14 การใช้คำสั่ง LS-L และ CHMOD	17
2.15 การรัน FILE HELLO.BASH	18
2.16 การสร้างตัวแปร	18
2.17 การเรียกใช้ตัวแปร	18
2.18 การตั้งค่าตัวแปร	19
2.19 FILE TESTVARIABLE BASH	19
2.20 ผลลัพธ์ที่ได้	20
2.21 การสร้างตัวแปรในFILE TESTENVIRONMENT BASH	20
2.22 การเปลี่ยนตัวแปร ENV ให้เป็นตัวแปรสภาพแวดล้อมใน PROMPT	21
2.23 FILE TESTARRAY.BASH	22
2.24 FILE TESTARRAY2.BASH	23
2.25 คีย์แพดขนาด 4x4	28
3.1 บล็อกไดอะแกรมรวมของระบบ	29
3.2 บล็อกไดอะแกรมของแอปพลิเคชัน	30
3.3 โฟลว์ชาร์ตการทำงานของชุดคำสั่งของระบบ	31

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.4 การต่ออุปกรณ์ทั้งหมดเข้าด้วยกัน	34
4.1 หน้าต่างสำหรับใส่รหัส	35
4.2 หน้า MAIN MENU	36
4.3 หน้า POLICY BASED BANDWIDTH SHAPING	36
4.4 SET INTERFACE IP ของ UPLINK	37
4.5 SET INTERFACE IP ของ DOWNLINK	37
4.6 ปรากฏหน้าต่างสำหรับใส่ IP ผิด	38
4.7 การตั้งค่าอัปโหลดและดาวน์โหลดของท่อ 0	38
4.8 การเลือกท่อ 0	39
4.9 การตั้งค่าอัปโหลดและดาวน์โหลดของท่อ 1	39
4.10 การเลือกท่อ 1	39
4.11 การกดปุ่มหมายเลข 4 ปรากฏหน้าต่าง CONCURRENT LIST ของ PPP0	40
4.12 หน้า CONCURRENT LIST ของ PPP1	40
4.13 ปรากฏหน้าต่าง CLEAR ALL SHAPER หลังกดปุ่มหมายเลข 3	41
4.14 การเชื่อมต่อ WI-FI T304-PPPOE-PROJECT	42
4.15 หน้า NETWORK AND SHARING CENTER	42
4.16 หน้า SET UP A NEW CONNECTION OR NETWORK	42
4.17 หน้า CONNECT TO THE INTERNET	43
4.18 การเลือก BROADBAND (PPPOE)	43
4.19 หน้าต่างสำหรับใส่ USER NAME และ PASSWORD	43
4.20 ข้อมูลการดาวน์โหลดด้านผู้ใช้โดยยังไม่ได้ทำการบิบบนดวิดิธ	44
4.21 อัตราเร็วในการดาวน์โหลดข้อมูลทางด้านแอดมินโดยที่ยังไม่ได้ทำการบิบบนดวิดิธ	44
4.22 ข้อมูลการดาวน์โหลดข้อมูลทางด้านผู้ใช้หลังจากทำการบิบบนดวิดิธแล้ว	44
4.23 อัตราเร็วในการดาวน์โหลดข้อมูลทางด้านแอดมินหลังจากทำการบิบบนดวิดิธ	45
4.24 ข้อมูลการดาวน์โหลดด้านผู้ใช้คนที่ 2 โดยยังไม่ได้ทำการบิบบนดวิดิธ	45
4.25 อัตราเร็วในการการดาวน์โหลดทางด้านแอดมินโดยยังไม่ได้ทำการบิบบนดวิดิธ	45

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.26 ข้อมูลการดาวน์โหลดด้านผู้คนที่ 2 หลังจากทำการบีบแบนด์วิดธ์แล้ว	46
4.27 อัตราเร็วในการการดาวน์โหลดทางด้านแอดมินหลังจากทำการบีบแบนด์วิดธ์แล้ว	46
4.28 กราฟของทั้งระบบสำหรับแสดงข้อมูลการดาวน์โหลดทางด้านแอดมินโดยยังไม่ได้ทำการบีบแบนด์วิดธ์	46
4.29 กราฟของทั้งระบบสำหรับแสดงข้อมูลการดาวน์โหลดทางด้านแอดมินหลังจากทำการบีบแบนด์วิดธ์แล้ว	47
4.30 การตั้งค่ารหัสใหม่	47
4.31 ข้อมูลของระบบ	48
4.32 หน้า CHASSIS CONTROL	48

สารบัญตาราง

ตารางที่	หน้า
2.1 บิตต่างๆของรีจิสเตอร์ SCON(SERIAL PORT CONTROL REGISTER)	10
2.2 การกำหนดบิต SMO และ SM1 เพื่อกำหนดโหมดการทำงาน	11
2.3 รีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล	12
2.4 รายละเอียดของขาต่างๆที่เกี่ยวข้องกับการสื่อสารข้อมูล	13
2.5 ตัวแปรพิเศษที่นิยมใช้กันบ่อยๆ	21
2.6 คำสั่งเกี่ยวกับจำนวนเต็ม	23
2.7 คำสั่งเกี่ยวกับ STRING	23
2.8 คำสั่งเกี่ยวกับ FILE	23
2.9 คำสั่งเกี่ยวกับ LOGICAL อื่นๆ	24
2.10 รายละเอียดตำแหน่งของขาและหน้าที่การใช้งานโมดูลแอลซีดี	25
2.11 คำสั่งควบคุมการเลื่อนตัวชี้และตัวอักษร	27

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการสื่อสารผ่านอินเทอร์เน็ตในปัจจุบันมีแนวโน้มที่จะมีการใช้งานมากขึ้น เมื่อมีผู้ใช้งานจำนวนมากใช้งานผ่านอินเทอร์เน็ตพร้อมกัน ทำให้การใช้งานผ่านอินเทอร์เน็ตเกิดความล่าช้า อันเนื่องมาจากปริมาณช่องทางการสื่อสารมีไม่เพียงพอต่อความต้องการของผู้ใช้

จากเหตุผลดังกล่าว จึงนำมาสู่การศึกษาค้นคว้าโครงการนี้ขึ้น โดยโครงการนี้เป็นการพัฒนาแอปพลิเคชันสำหรับควบคุมการแบ่งช่องทางในการสื่อสารและควบคุมปริมาณการใช้งานอินเทอร์เน็ตภายในเครือข่ายให้เพียงพอ โดยใช้หลักการของโปรโตคอลพีพีพี (Point-to-Point Protocol) วิ่งผ่านไปในระบบอินเทอร์เน็ต หรือที่เรียกว่า PPPoE ซึ่งแอปพลิเคชันดังกล่าวสามารถแบ่งช่องทางในการสื่อสารและควบคุมปริมาณการวิ่งเข้าออกของแพ็กเก็ตข้อมูล ตามอัตราส่วนที่กำหนดอย่างเหมาะสม โดยแอปพลิเคชันดังกล่าวถูกพัฒนาด้วยภาษาจาวาร่วมกับโปรเซสสวิตเซอร์และเซลล์สคริปต์ ที่ทำงานร่วมกับระบบปฏิบัติการลินุกซ์ แอปพลิเคชันดังกล่าวสามารถรับคำสั่งจากผู้ใช้งานผ่านอินเทอร์เน็ตเพสจากนั้นแปลงค่าอินเทอร์เน็ตให้เป็นค่าพารามิเตอร์ไปยังรูปแบบที่ manipulate traffic control setting เข้าใจความหมายของการควบคุมและเลือกวิธีการแบ่งช่องทางในการสื่อสารหรือควบคุมปริมาณการใช้งานอินเทอร์เน็ตให้แก่ผู้ใช้งานได้ตามอัตราส่วนที่ต้องการได้

แต่ในแอปพลิเคชันดังกล่าวยังมีข้อจำกัดด้านการเข้าไปใช้งาน เนื่องจากการเข้าใช้งานจำเป็นต้องเชื่อมต่อผ่านระบบอินเทอร์เน็ต หากระบบอินเทอร์เน็ตเกิดความขัดข้อง แอปพลิเคชันจะไม่สามารถใช้งานได้ ด้วยเหตุนี้จึงนำไมโครคอนโทรลเลอร์มาใช้งานในการเป็นตัวควบคุมแอปพลิเคชันโดยใช้เป็นตัวควบคุมหน้าจอแสดงผลเจ็ดส่วน ทำให้ผู้ใช้งานสามารถใช้งานแอปพลิเคชันได้โดยตรงโดยไม่ต้องมีหน้าจอคอมพิวเตอร์

1.2 วัตถุประสงค์

โครงการนี้จัดทำขึ้นเพื่อพัฒนาแอปพลิเคชันสำหรับควบคุมการแบ่งช่องทางในการสื่อสารและควบคุมปริมาณการใช้งานอินเทอร์เน็ตภายในเครือข่ายให้เพียงพอ โดยควบคุมผ่านหน้าพาแนลที่สามารถแสดงผลข้อมูลและป้อนคำสั่งได้โดยตรง

1.3 ขอบเขตของปริญญาโท

ใช้หลักการของโปรโตคอลพีพีพีวีงผ่านไปในระบบอีเทอร์เน็ต หรือเรียกว่า PPPoE โดยเป็นการทำงานบนระบบปฏิบัติการลินุกซ์ แอปพลิเคชันนี้จะถูกพัฒนาด้วยภาษาจาวาร่วมกับโปรเซสเซอร์ไลบรารีและเซลล์สคริปต์ โดยใช้โมดูลแอลซีดีเป็นหน้าพาแนลสำหรับแสดงผลข้อมูลและป้อนคำสั่งผ่านคีย์แพดได้โดยตรง โดยส่งข้อมูลผ่านอาร์เอส 232 ไม่จำกัดจำนวนผู้ใช้งานเริ่มต้นที่ 1 ผู้ใช้งาน มีความเร็วของฮาร์ดดิสก์ไม่เกิน 1 Mbps และความเร็วของดาวเทียมไม่เกิน 10 Mbps

บทที่ 2

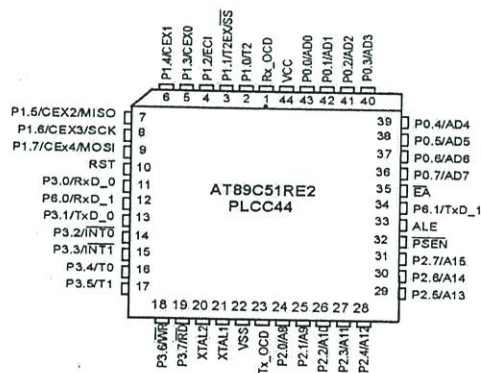
ทฤษฎีและหลักการ

การควบคุมแบบตัวจริงสำหรับหลายผู้ใช้บริการ เป็นการจัดทำขึ้นเพื่อควบคุมการแบ่งช่องทางในการสื่อสารและควบคุมการใช้งานอินเทอร์เน็ตภายในเครือข่ายให้เพียงพอ โดยควบคุมผ่านหน้าพาแนลที่สามารถแสดงผลข้อมูลและป้อนคำสั่งผ่านคีย์แพดได้โดยตรง

2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 (MCS-51 Microcontroller)

สถาปัตยกรรมของ MCS-51 ประกอบด้วยหลากหลายขนาดแต่ในปริมาณที่น้อยนี้จะใช้ไมโครคอนโทรลเลอร์รุ่น AT89C51RE2 ซึ่งเป็นสถาปัตยกรรมแบบ 8 บิต (8 bits Microcontroller) ที่มีการจัดเรียงขาต่างๆดังรูปที่ 2.1 และมีข้อมูลจำเพาะดังนี้

- ก. หน่วยความจำโปรแกรมแบบแฟลช (Flash Memory) ขนาด 128 กิโลไบต์
- ข. หน่วยความจำข้อมูลแบบแรม (RAM) ขนาด 8 กิโลไบต์
- ค. มีพอร์ต I/O ขนาด 8 บิต จำนวน 4 พอร์ต
- ง. ความถี่ใช้งานสูงสุด 18.432 เมกะเฮิร์ต
- จ. มีวงจรสื่อสารอนุกรม UART จำนวน 2 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
- ฉ. มีวงจร Timer/Counter ขนาด 16 บิต สำหรับ Timer_0 ,Timer_1 และ Timer_2
- ช. ทำงานตั้งแต่แรงดัน 2.7-5.5 โวลต์



รูปที่ 2.1 การจัดวางขาของ AT89C51RE2

2.2 โปรแกรม Flip

ในการที่จะให้ไมโครคอนโทรลเลอร์ทำงานนั้นผู้ใช้งานจำเป็นจะต้องเขียนโปรแกรมคำสั่งเข้าไปสั่งการให้ไมโครคอนโทรลเลอร์ทำงานต่างๆ เช่น แสดงผลผ่านโมดูลแอลซีดี เป็นต้น ในปริญญาโทฉบับนี้จะใช้ตัวโปรแกรมที่มีชื่อว่า Flip (Flexible In-system Programmer) ในการพัฒนา ทำงานภายใต้ระบบปฏิบัติการของ Window98/Me/NT/2000/XP และ Linux โดยสนับสนุนการเชื่อมต่อผ่านทางพอร์ตอนุกรมอาร์เอส 232 โดยโปรแกรม Flip จะใช้สำหรับดาวน์โหลดข้อมูลให้กับหน่วยความจำไมโครคอนโทรลเลอร์ที่ทำงานในโหมดมอนิเตอร์ (Monitor Mode) เพื่อให้ผู้ใช้สั่งจัดการกับหน่วยความจำภายในตัวซีพียู (CPU) ไม่ว่าจะเป็นการล้างข้อมูล (Erase) สั่งตรวจสอบข้อมูลในหน่วยความจำ (Blank Check) สั่งโปรแกรมข้อมูลให้กับหน่วยความจำโปรแกรมของ CPU (Program) สั่งเปรียบเทียบข้อมูลจาก Buffer กับหน่วยความจำในตัว CPU (Verify) หรือสั่งอ่านข้อมูลจากหน่วยความจำของ CPU (Read) เป็นต้น

การอธิบายวิธีการดาวน์โหลด Hex File ด้วยโปรแกรม FLIP V3.1.0 Build1

2.2.1 ทำการติดตั้งโปรแกรมจาวา (Java)

2.2.2 ทำการติดตั้งโปรแกรม Flip

2.2.3 ต่อสาย RS232 จากเครื่องพีซี เข้ากับขั้วต่อ RS232#1 ของบอร์ดไมโครคอนโทรลเลอร์

2.2.4 ทำการกดสวิตช์ PSEN และ RESET เพื่อเข้าสู่ Monitor Mode

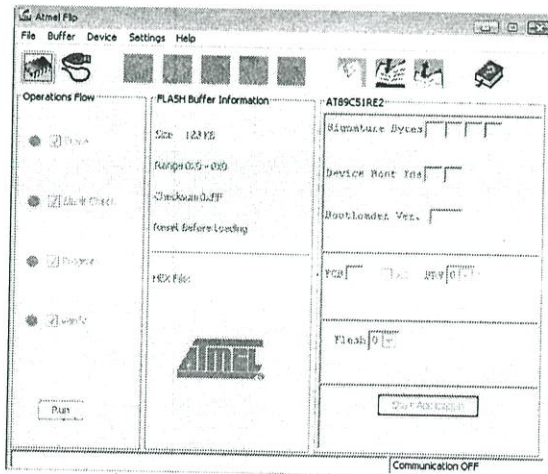
ก. กด SW. PSEN ค้างไว้

ข. ตามด้วยการกด SW. RESET ค้างไว้

ค. ปล่อย SW. RESET ในขณะที่ SW. PSEN ยังถูกกดค้างอยู่

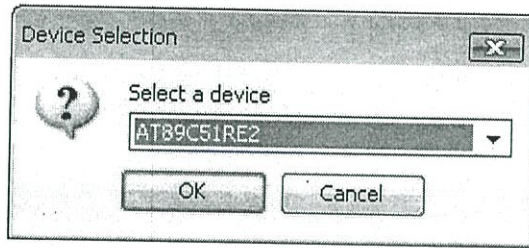
ง. ปล่อย SW. PSEN เป็นลำดับสุดท้าย

2.2.5 ทำการเปิดโปรแกรม Flip จะได้หน้าต่าง Atmel Flip ดังรูปที่ 2.2



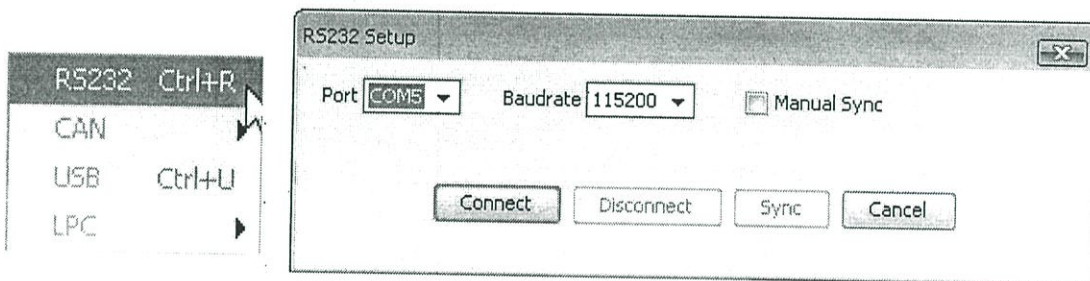
รูปที่ 2.2 หน้าต่าง Atmel Flip

2.2.6 ทำการคลิก Select a Target Device จะได้หน้าต่าง Device Selection ดังรูปที่ 2.3 จากนั้นทำการเลือกเบอร์ไมโครคอนโทรลเลอร์ที่ใช้งาน เมื่อเลือกเสร็จให้กด OK

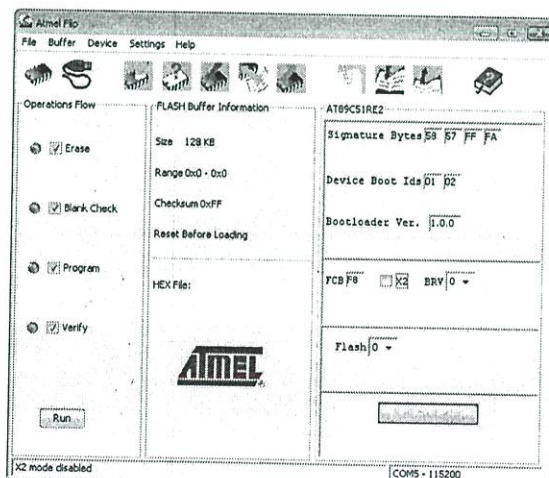


รูปที่ 2.3 หน้าต่าง Device Selection สำหรับเลือกเบอร์ไมโครคอนโทรลเลอร์ที่ใช้งาน

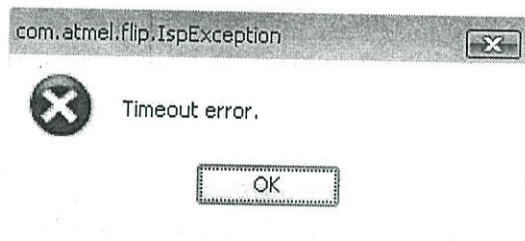
2.2.7 ทำการคลิกที่ไอคอน Select a Communication Medium เพื่อเลือก Port ในการสื่อสาร เมื่อคลิกแล้วจะปรากฏหน้าต่างรูปเล็กออกมา ให้ทำการเลือก RS232 จะได้หน้าต่าง RS232 Setup ดังรูปที่ 2.4 จากนั้นให้ทำการเลือก Com Port ของเครื่องพีซีและเลือก Baud Rate (default = 115200) ที่จะใช้ในการดาวน์โหลดโปรแกรม จากนั้นทำการกด Connect หากทำการ Connect สำเร็จจะขึ้นหน้าต่างดังรูปที่ 2.5 แต่หาก Connect ไม่สำเร็จจะขึ้นหน้าต่างดังรูปที่ 2.6



รูปที่ 2.4 หน้าต่าง RS232 Setup สำหรับเลือก Com Port และ Baud Rate ในการดาวน์โหลด

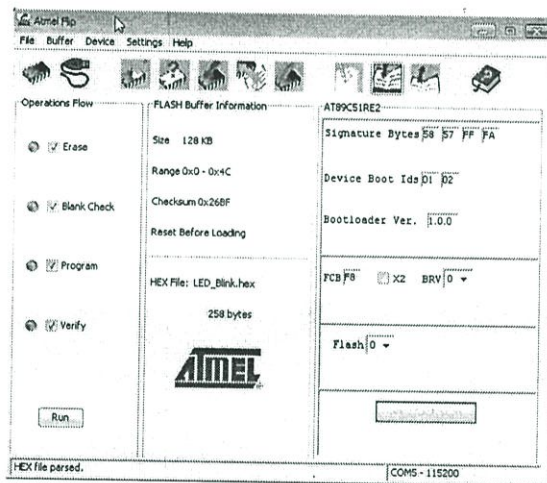


รูปที่ 2.5 หน้าต่าง Atmel Flip หลังจาก Connect Port สำเร็จ



รูปที่ 2.6 หน้าต่าง Timeout error หลังจาก Connect Port เมื่อ Connect Port ไม่สำเร็จ

2.2.8 หลังจาก Connect Port ได้เรียบร้อยแล้วให้ทำการคลิกที่ไอคอน Load HEX file แล้วทำการเลือก file.Hex จากภายนอกเข้ามายังโปรแกรม Flip ชื่อของ file ที่เลือกเข้ามาจะปรากฏที่หน้าต่าง Atmel Flip ในช่อง Hex File : ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 Hex File ที่ถูกโหลดเข้า

2.2.9 เมื่อโหลดไฟล์เข้ามาแล้ว ให้คลิกที่ปุ่ม Run เพื่อดาวน์โหลดโปรแกรมลงในไมโครคอนโทรลเลอร์

2.2.10 หลังจากทำการดาวน์โหลดเรียบร้อยแล้ว ให้กดที่สวิตช์ RESET ที่บอร์ดไมโครคอนโทรลเลอร์ เพื่อรันโปรแกรมดูผลการทำงานของโปรแกรมที่เขียนขึ้น

2.3 โปรแกรม Keil u Vision3

ในการที่จะให้ไมโครคอนโทรลเลอร์ทำงานนั้นผู้ใช้งานจำเป็นต้องเขียนโปรแกรมคำสั่งเข้าไปสั่งการให้ไมโครคอนโทรลเลอร์ทำงานต่างๆ ในโครงงานนี้จะใช้โปรแกรม Keil u Vision3 ซึ่งเป็นโปรแกรมอีกตัวหนึ่งที่ใช้ในการพัฒนานอกจากโปรแกรม Flip โดยใช้ภาษาซีในการพัฒนา ซึ่งตัวโปรแกรมนี้อาจทำหน้าที่เป็นคอมไพเลอร์ (Compiler) ในตัวและจะสร้างเฮกไฟล์ (Hex File) เพื่อนำไปดาวน์โหลดลงในตัวไมโครคอนโทรลเลอร์

2.4 การสื่อสารข้อมูล (Data Communication)

การสื่อสารข้อมูลคือการส่งข้อมูลที่ทำให้การเข้ารหัสแล้วระหว่างอุปกรณ์ 2 ชนิด โดยมีองค์ประกอบหลักสำคัญอยู่ 3 ส่วน คือ อุปกรณ์ฝั่งส่ง อุปกรณ์ฝั่งรับ และตัวกลาง ซึ่งตัวกลางจะสามารถแบ่งได้เป็น 2 ประเภท ได้แก่ แบบมีสาย และแบบไร้สาย โดยจะสามารถแบ่งแยกรูปแบบในการสื่อสารข้อมูลได้เป็น 3 แบบคือ การสื่อสารทางเดียว การสื่อสารกึ่งสองทาง และการสื่อสารสองทาง

ก. การสื่อสารทางเดียว (Simplex) ใช้ในการสื่อสารของข้อมูลที่อุปกรณ์แต่ละชนิดจะทำหน้าที่เป็นตัวรับ หรือส่งเพียงอย่างเดียวเท่านั้นทำให้การสื่อสารแบบนี้สื่อสารได้เพียงทางเดียว

ข. การสื่อสารแบบกึ่งสองทาง (Half Duplex) ใช้ในการสื่อสารข้อมูลระหว่างอุปกรณ์ที่มีการแลกเปลี่ยนข้อมูลกัน อุปกรณ์ในแต่ละฝั่งจะเป็นได้ทั้งตัวส่งและตัวรับ แต่จะไม่สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน (รับ-ส่งสลับกัน) จึงทำให้ทิศทางของการสื่อสารมีทางเดียวในเวลาใดๆ

ค. การสื่อสารสองทาง (Full Duplex) ใช้ในการสื่อสารแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ 2 ชนิดที่ติดต่อกัน โดยจะมีข้อมูลไหลได้ทั้งด้านรับและส่งในเวลาเดียวกัน จึงทำให้ทิศทางของการสื่อสารมี 2 ทาง

2.4.1 การสื่อสารข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมเป็นการสื่อสารที่มีการรับส่งข้อมูลที่ละบิตเรียงต่อกันไปจนสิ้นสุดซึ่งการสื่อสารแบบนี้จะแตกต่างจากการสื่อสารแบบขนานเนื่องจากการสื่อสารแบบขนานนั้นถ้าจำนวนข้อมูลมากขึ้นก็ต้องเพิ่มจำนวนสายสัญญาณจึงไม่มีความเหมาะสมในการใช้งานที่มีระยะทางในการรับส่งข้อมูลไกลๆ ซึ่งเป็นการสิ้นเปลืองค่าใช้จ่ายอย่างมาก แต่การสื่อสารแบบอนุกรมนั้นใช้จำนวนสายสัญญาณเพิ่ม 2 ถึง 3 เส้นเท่านั้น จึงทำให้ใช้เวลาในการรับส่งข้อมูลมากกว่าการสื่อสารแบบขนานแต่ค่าใช้จ่ายจะลดลง โดยการสื่อสารแบบอนุกรมนั้นจะสามารถแบ่งแยกลักษณะของการสื่อสารได้เป็น 2 ประเภท คือ การสื่อสารแบบซิงโครนัส และการสื่อสารแบบอซิงโครนัส

ก. การสื่อสารแบบซิงโครนัส (Synchronous) การสื่อสารแบบซิงโครนัสนั้นจำเป็นต้องอาศัยสัญญาณนาฬิกาช่วยกำหนดจังหวะในการสื่อสาร

ข. การสื่อสารแบบอซิงโครนัส (Asynchronous) การสื่อสารแบบอซิงโครนัสนั้นไม่จำเป็นต้องอาศัยสัญญาณนาฬิกาใช้ในการกำหนดจังหวะการสื่อสารแต่จะใช้รูปแบบในการส่งข้อมูล (Bit Pattern) เป็นตัวกำหนดว่าข้อมูลส่วนไหนเป็นข้อมูลอะไรซึ่งรูปแบบโดยทั่วไปจะเป็นดังนี้คือ เริ่มต้นด้วยบิตเริ่มต้น (Start Bit) ตามด้วยข้อมูล (Data) ตามด้วยบิตพาริตี (Parity Bit) สิ้นสุดด้วยบิตสิ้นสุด (Stop Bit) โดยการส่ง

ข้อมูลแบบอซิงโครนัสนั้นจะมีอุปกรณ์ที่เรียกว่า ยูอาร์ที (UART: Universal Asynchronous Receiver/Transmitter) เป็นตัวควบคุมในการรับส่ง

ในการสื่อสารแบบอนุกรมข้อมูลที่ใช้ในการสื่อสารจะอยู่ในลักษณะกลุ่มบิตข้อมูล (Bit Stream) จำเป็นจะต้องสนใจในเรื่องอัตราเร็วในการรับส่งข้อมูลโดยจะใช้การกำหนดอัตราบอर्ड (Baud Rate) ในการกำหนดซึ่งจะมีค่ามาตรฐานได้แก่ 110, 150, 300, 1200, 2400, 4800, 9600 และ 19200 เป็นต้น

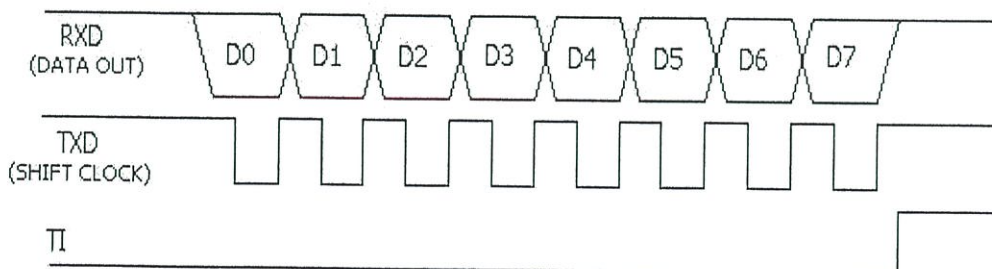
2.4.2 การติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะมีโครงสร้างเป็นแบบฟูลดูเพล็กซ์ ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้ โดยจะมีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต เพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้ว จึงจะทำการเซตบิตโดยกำหนดค่าของแฟล็ก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้วและพร้อมที่จะส่งข้อมูลบิตต่อไปได้

การรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้น โดยการกำหนดค่าของบิต REN ที่อยู่ในรีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ " 1 " หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอก ก็จะมีการเลื่อนข้อมูลไปโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ในรีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ "1" ซึ่งส่งผลให้เกิดการอินเทอร์รัพท์โปรแกรมขึ้น

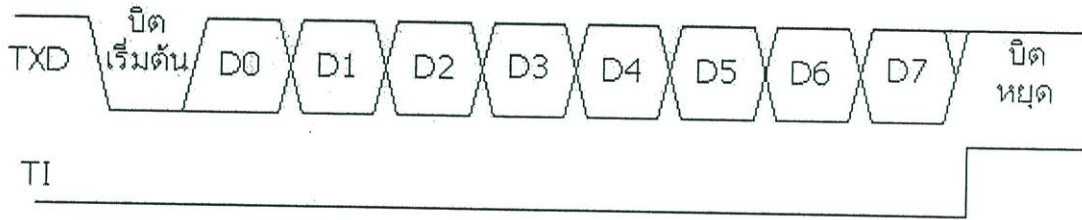
2.4.2.1 โหมดการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

1) โหมด 0 เป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียว และจะไม่มีบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อนแสดงดังรูปที่ 2.8 ข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพรีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุต ให้มีจำนวนมาก แต่ในโหมด 0 ไม่ค่อยนิยมนำมาใช้งาน เพราะไอซีไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่มีจำนวนพอร์ตที่มากพอ และมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



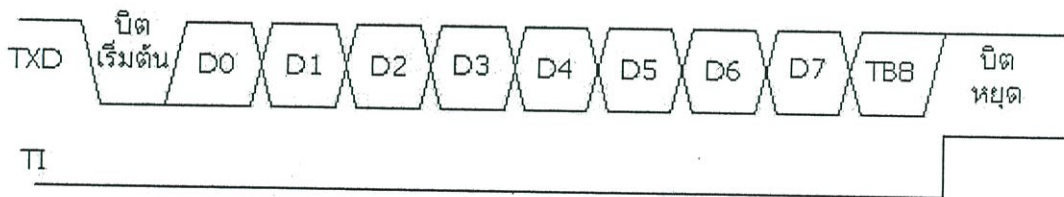
รูปที่ 2.8 สัญญาณรับและส่งข้อมูลในโหมด 0

2) โหมด 1 เป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โพล์ของไทมเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการแสดงดังรูปที่ 2.9



รูปที่ 2.9 สัญญาณการส่งข้อมูลในโหมดหนึ่ง

3) โหมด 2 เป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และ ส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โพล์ของไทมเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการแสดงดังรูปที่ 2.10



รูปที่ 2.10 สัญญาณการส่งข้อมูลในโหมดสอง

4) โหมด 3 เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับและส่งข้อมูลได้ตามต้องการ

UART (Universal Asynchronous Receiver Transmitters) เป็นการส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่ง และด้านรับ เป็นการส่งข้อมูลโดยทำการเพิ่มเติมข้อมูลบางอย่างเข้าไป (Start bit ,Stop bit ,Parity bit) เพื่อให้การรับ และการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

การสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับและส่งข้อมูลทางอนุกรมมีอยู่ด้วยกัน 2 ระบบ คือ

ก. Single Processor System คือระบบการสื่อสารโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมต่อกัน

ข. Multiprocessor System คือระบบการสื่อสารแบบมัลติโพรเซสเซอร์ โดยใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นตัวแม่ (Master) และสามารถที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นตัวลูก (Slave) ได้อีกเป็นจำนวนหลายๆ ตัว

2.4.2.2 รีจิสเตอร์ที่ใช้ในการควบคุม และรับส่งข้อมูลของพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้ในการติดต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะประกอบด้วย รีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และ รีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับส่ง โดยรีจิสเตอร์ แต่ละตัวจะมีหน้าที่ และการทำงานในแต่ละบิตดังต่อไปนี้

1) รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ (Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 (ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล(TB8) รายละเอียดของแต่ละบิตดังตารางที่ 2.1

ตารางที่ 2.1 บิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ก. SM0, SM1: (Serial port mode bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด ดังตารางที่ 2.2

ตารางที่ 2.2 การกำหนดบิต SM0 และ SM1 เพื่อกำหนดโหมดการทำงาน

SM0	SM1	โหมด การทำงาน อัตรารับส่ง
0	0	0 Shift register $f_{osc} / 12$
0	1	1 8 bit UART Variable
1	0	2 9 bit UART $f_{osc}/32$ หรือ $f_{osc}/6$
1	1	3 9 bit UART Variable

ข. SM2 : เป็นบิตที่ทำหน้าที่ควบคุมการทำงานและเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ แบบ Single Processor System หรือ Multi processors System โดยกำหนดให้ SM2 = 1 เป็นการเลือกแบบ Multi processors System คือระบบการสื่อสารแบบใช้ซีพียูหลายๆตัว ร่วมกันทำงาน จะใช้งานในโหมด 2 หรือโหมด 3

SM2 = 0 เป็นการเลือกแบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานในโหมด 0 ต้องกำหนดให้ SM2 = 0)

ในกรณีที่เลือกให้ SM2 = 1 แบบ Multi processors System

ถ้าข้อมูลที่รับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น "1" ทำให้แฟล็กอินเตอร์รัพต์ทางด้านรับ จะถูกเซตให้เป็น 1 (RI = 1) แต่ถ้าข้อมูลในบิตที่ 9 รับเข้ามามีค่าเป็น "0" จะทำให้แฟล็กอินเตอร์รัพต์ทางด้านรับเป็น 0 (RI = 0)

การทำงาน ในโหมด 1 ถ้าให้ SM2 = 1 แฟล็กอินเตอร์รัพต์ทางด้านรับ (แฟล็ก RI) จะไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop bit)

ค. REN : (Enable Serial Reception) เป็นบิตที่ควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์

1 = ให้มีการรับข้อมูล

0 = ไม่ให้มีการรับข้อมูล

ง. TB8 : (Transmit bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

RB8 : (Receive bit D8) เป็นบิตของข้อมูลบิตที่ 9 ในการรับข้อมูล โดยใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้จะเป็นค่าของ Stop Bit ที่รับเข้ามา สำหรับโหมด 0 จะไม่ใช้งานบิตนี้

จ. TI : (Transmit Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพต์ด้านส่งข้อมูล และจะถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 (Shift register) หรือเมื่อเริ่มต้นส่ง Stop bit ในโหมด 1, 2 หรือ 3 และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัพต์ของการส่งข้อมูลเรียบร้อยแล้ว

ฉ. RI : (Receive Interrupt Flag) เป็นบิตที่ใช้งานในการอินเตอร์รัพต์ทางด้านรับข้อมูล จะถูกเซตทางฮาร์ดแวร์ เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ในโหมด 0 (Shift register) และจะต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง เมื่อโปรแกรมตอบสนองการอินเตอร์รัพต์ของการรับข้อมูลเรียบร้อยแล้ว หรืออาจกล่าวได้ว่า ถ้าบิต RI ถูกเซต เมื่อใด หมายถึงข้อมูลได้เข้ามาเก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้ว สามารถที่จะอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

2) รีจิสเตอร์ SBUF (serial data buffer register) เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ที่ตำแหน่งที่ 99H และเข้าถึงข้อมูลแบบไบต์ได้อย่างเดียว จะทำหน้าที่รับและส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

ในการอ่านค่าข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะต้องอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งเป็นบัฟเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้ามาได้จากภายนอก และในทำนองเดียวกันขณะที่ต้องการส่งข้อมูล จะนำเอาค่าข้อมูลที่ส่งออกไปไว้ที่ในรีจิสเตอร์ SBUF ก่อน และหลังจากนั้นจึงจะส่งออกไป โดยจะใช้คำสั่งการโอนย้ายข้อมูลแบบไบต์เช่น MOV SBUF,#20H หรือ MOV SBUF,@R1 การรับข้อมูลในโหมด 0 จะเริ่มต้นรับ เมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start bit เข้ามาที่ขา RXD

3) รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ที่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียวกัน โดยจะประกอบด้วยบิตดังตารางที่ 2.3

ตารางที่ 2.3 รีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

2.4.3 มาตรฐานอาร์เอส 232

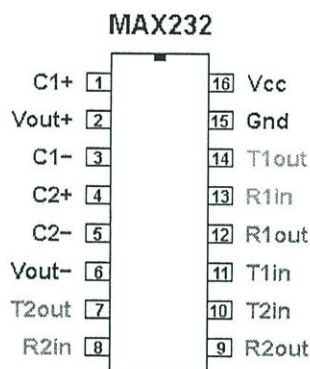
ในปัจจุบันมีอุปกรณ์ต่างๆ มากมายจากผู้ผลิตหลายๆ ค่ายจึงจำเป็นต้องมีตัวกำหนดมาตรฐาน เพื่อให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ โดยมาตรฐานที่นิยมใช้กันอย่างแพร่หลายนั่นก็คือ มาตรฐานอาร์เอส 232 (RS 232 : Recommended Standard 232) ซึ่งจะมีระดับแรงดันให้อยู่ในช่วง -15 ถึง 15 โวลต์ โดยลอจิก "0" จะมีแรงดันอยู่ในช่วง 3 ถึง 15 โวลต์ และลอจิก "1" จะมีแรงดันในช่วง -15 ถึง -3 โวลต์ ซึ่งอาร์เอส 232 นั้นมีสายสัญญาณ 3 เส้นก็คือ สายส่งข้อมูล(Tx) สายรับข้อมูล (Rx) และกราวด์(Gnd)

2.4.4 ทีทีแอล

ทีทีแอล (TTL: Transistor-Transistor Logic) เป็นการกำหนดระดับแรงดันที่เกิดขึ้นในยุคแรกๆ ใช้ในการสื่อสารระหว่างทรานซิสเตอร์โดยมีช่วงของระดับแรงดันอยู่ที่ 0 ถึง 5 โวลต์ และในปัจจุบันมีการพัฒนาระดับแรงดันทีทีแอลแบบทีทีแอลแรงดันต่ำ (LVTTTL:Low-Voltage TTL) ซึ่งมีระดับแรงดันอยู่ที่ 0 ถึง 3.3 โวลต์

2.4.5 การแปลงระดับแรงดัน

การที่จะทำการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์นั้นจำเป็นต้องทำการแปลงระดับแรงดัน เนื่องจากข้อมูลที่ออกมาจากพอร์ตอนุกรมของคอมพิวเตอร์มีระดับแรงดันตามมาตรฐาน อาร์เอส 232 แต่ข้อมูลที่สามารถสื่อสารกับไมโครคอนโทรลเลอร์ได้นั้นต้องมีระดับแรงดันแบบทีทีแอล จะเห็นว่ามีระดับแรงดันที่ต่างกันจึงจำเป็นต้องทำการแปลงระดับแรงดันโดยที่การแปลงระดับแรงดันนั้นจะอาศัย ไอซีเบอร์แม็กซ์ 232 (IC MAX232) ในการแปลงระดับแรงดันซึ่งแม็กซ์ 232 นั้นมีการจัดเรียงขาแสดงดังรูปที่ 2.11



รูปที่ 2.11 การเรียงขาของไอซีแม็กซ์ 232

จากรูปที่ 2.11 จะเห็นได้ว่าขาของไอซีแม็กซ์ 232 ที่เกี่ยวข้องกับการสื่อสารข้อมูลจะได้แก่ ขา 7 8 9 10 11 12 13 และ 14 โดยที่รายละเอียดต่างๆของขาดังกล่าวแสดงได้ตามตารางที่ 2.4

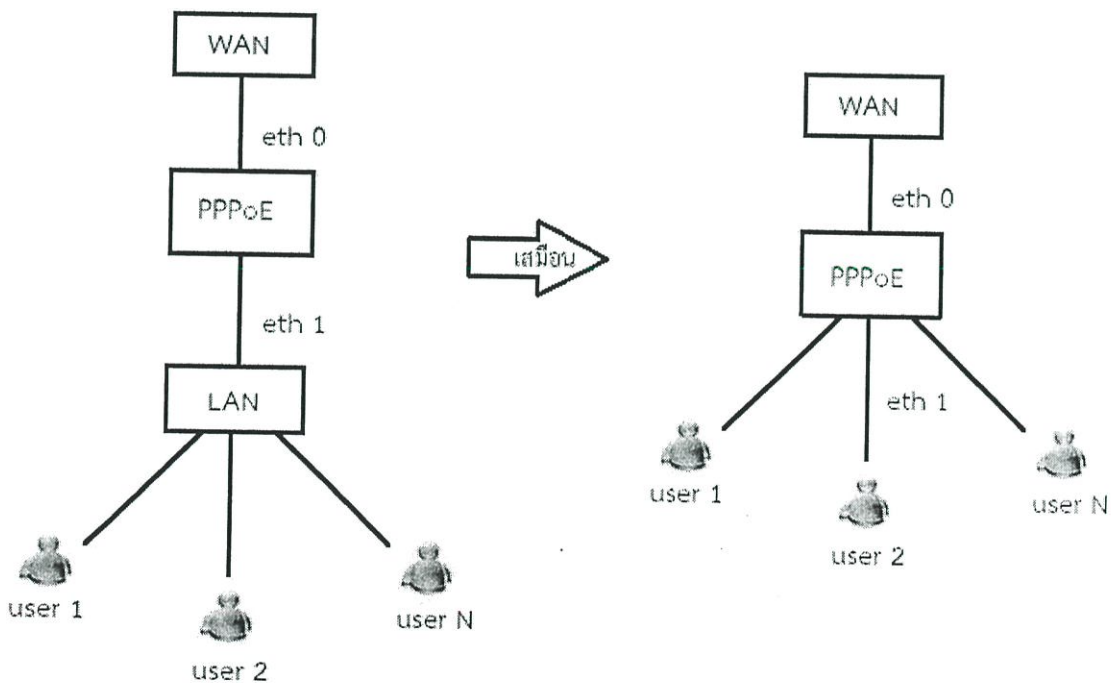
ตารางที่ 2.4 รายละเอียดของขาต่างๆที่เกี่ยวข้องกับการสื่อสารข้อมูล

ชื่อขา	หมายเลข	รายละเอียดขา	ระดับแรงดัน(โวลต์)
T2OUT	7	ส่งออกสัญญาณอาร์เอส232	-15 ถึง 15
R2IN	8	รับเข้าสัญญาณอาร์เอส232	-15 ถึง 15
R2OUT	9	ส่งสัญญาณทีทีแอล	0 ถึง 3.3 หรือ 0 ถึง 5
T2IN	10	รับเข้าสัญญาณทีทีแอล	0 ถึง 3.3 หรือ 0 ถึง 5
T1IN	11	รับเข้าสัญญาณทีทีแอล	0 ถึง 3.3 หรือ 0 ถึง 5
R1OUT	12	ส่งออกสัญญาณทีทีแอล	0 ถึง 3.3 หรือ 0 ถึง 5
R1IN	13	รับเข้าสัญญาณอาร์เอส232	-15 ถึง 15
T1OUT	14	ส่งออกสัญญาณอาร์เอส232	-15 ถึง 15

จากตารางที่ 2.4 จะเห็นว่าไอซีแม็กซ์ 232 สามารถแปลงระดับแรงดันได้ 2 ชั้นแนล และการที่จะแปลงสัญญาณนั้นจำเป็นจะต้องเลือกใช้ขาของไอซีแม็กซ์ 232 ให้ตรงกัน คือ ถ้าเลือกใช้ขา 8 ในการรับเข้า สัญญาณอาร์เอส 232 สัญญาณที่ที่แอลที่จะออกที่ขา 9 เป็นต้น

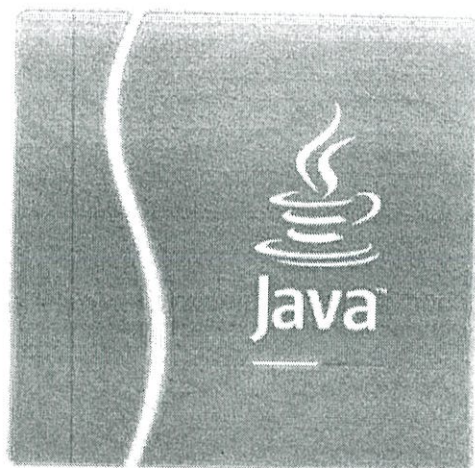
2.5 PPPoE

PPPoE คือ Point-to-Point Protocol over Ethernet เป็นการพัฒนาต่อมาจาก PPP มาเป็น PPPoE ซึ่งให้ความเร็วอินเทอร์เน็ตที่มากขึ้นและ security ที่มากขึ้นกว่าเดิม PPPoE เป็นการทำ encapsulation ห่อหุ้มเฟรมโดยใช้โปรโตคอลพีพีพี (point to point protocol) วิ่งผ่านไปในระบบอีเทอร์เน็ต การที่ต้องใช้ PPPoE เนื่องจากว่า core network ของ network service provider นั้นใช้เป็น Ethernet technology ดังนั้นจึงต้องเซตให้โปรโตคอลพีพีพี วิ่งผ่านไปใน Ethernet network ได้ ตัวอย่างเช่น ผู้ใช้เชื่อมต่ออีเทอร์เน็ตผ่านระบบแลน (LAN) และใช้งานแบบ PPPoE แล้วต้องการทราบไอพีแอดเดรส (IP Address) โดยใช้คำสั่ง ifconfig ในคอมมานด์ไลน์ (Command Line) จะได้ค่าไอพีแอดเดรสของ eth0, eth1 และไอพีแอดเดรสของผู้ใช้ที่เชื่อมต่ออยู่ ซึ่งโดยปกติแล้วหากไม่ใช้งานแบบ PPPoE จะมีแต่ค่าไอพีแอดเดรสของ eth0 และ eth1 เท่านั้นและไอพีแอดเดรสของผู้ใช้จะต้องแสดงในส่วนของ การต่อแบบแลน จะเห็นว่า PPPoE จะมองว่าผู้ใช้ไม่ได้ต่อผ่านแลนแต่ต่อโดยตรงกับเซิร์ฟเวอร์เลยซึ่งจะมีข้อดีในเรื่องของความเร็วที่เพิ่มขึ้นแสดงดังรูปที่ 2.12



รูปที่ 2.12 การเชื่อมต่ออีเทอร์เน็ตผ่าน PPPoE

2.6 ภาษาจาวา



รูปที่ 2.13 โปรแกรมจาวา

ภาษา Java แสดงดังรูปที่ 2.13 เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) ซึ่งเป็นเทคนิคที่ใช้วัตถุ (Object) เป็นหลักในการพิจารณาสิ่งต่างๆ ที่สนใจ ดังนั้นโปรแกรมที่เขียนด้วยภาษาจาวาจึงประกอบไปด้วยกลุ่มของ Object โดยแต่ละ Object จะถูกจัดกลุ่มอยู่ในรูปของคลาส (Class) โดยที่แต่ละคลาสสามารถมีการถ่ายทอดลักษณะ (Inheritance) กันลงมาอย่างเป็นลำดับ (Hierarchy) ภาษา Java ได้รับความนิยมนอย่างสูง เนื่องจากโปรแกรมคอมพิวเตอร์ที่เขียนด้วยภาษา Java สามารถทำงานบนเครื่องคอมพิวเตอร์ที่มีสภาพแวดล้อมต่างกันได้ โดยไม่ต้องคอมไพล์ (Compile) ใหม่ เป็นภาษาที่มีไวยากรณ์ที่สามารถเรียนรู้ได้ง่าย มีกลไกของภาษาที่ไม่ซับซ้อน เน้นความถูกต้องของชนิดข้อมูล (Data Type) ที่ใช้ในโปรแกรม นอกจากนี้ ด้วยเทคนิคการเขียนโปรแกรมเชิงวัตถุจึงสามารถนำบางส่วนของโปรแกรมที่ถูกเขียนไปแล้วไปใช้กับโปรแกรมที่จะถูกเขียนหรือพัฒนาขึ้นมาใหม่ได้

2.6.1 ข้อดีของ ภาษา Java

2.6.1.1 ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้อำนาจหรือชื่อต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น

2.6.1.2 โปรแกรมที่เขียนขึ้นโดยใช้ภาษาจาวาจะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารรถูกคอมไพล์ และรันบนเครื่องพีซีธรรมดาได้

2.6.1.3 ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย

2.6.1.4 ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวนโค้ด (code) น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

2.6.1.5 ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificates

2.6.1.6 มี IDE, application server, และ library ต่าง ๆ มากมายสำหรับจาวาที่สามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่าง ๆ

2.6.2 ข้อเสียของ ภาษา Java

2.6.2.1 ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่ง ทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา

2.6.2.2 tool ที่มีในการใช้พัฒนาโปรแกรมจาวามักทำไม่ได้ ทำให้ต้องเสียเวลาทำงานในส่วนที่ tool ทำไม่ได้ tool ของไมโครซอฟท์จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า

2.7 ระบบปฏิบัติการ Linux

ลินุกซ์ (Linux) คือ โปรแกรมเคอร์เนล (kernel) ซึ่งทำหน้าที่เป็นศูนย์กลางของระบบปฏิบัติการ (Operating System) ซึ่งก็จะเหมือนกับระบบปฏิบัติการอื่นๆ เช่น Windows , Unix โดยที่ Linux ได้รับการพัฒนาจากระบบปฏิบัติการ Unix ลินุกซ์มีระบบปฏิบัติการแบบ 32 บิต มีระบบ X วินโดวส์ซึ่งเป็นระบบการติดต่อผู้ใช้แบบกราฟฟิก และเป็นระบบปฏิบัติการที่อยู่ภายใต้เงื่อนไขของ GPL (GNU General Public Licence) หมายความว่า สามารถเปลี่ยนแปลงแก้ไข พัฒนาได้ จุดเด่นของ Linux ก็คือเป็นระบบปฏิบัติการที่

ใช้งานได้ฟรี ทำงานได้บนเครื่องพีซีทั่วไป สามารถทำงานได้รวดเร็ว เนื่องจากมีระบบการจัดการหน่วยความจำเสมือน (Virtual Memory) มีการทำงานแบบมัลติแทสกกิง (Multitasking) และระบบป้องกันการรบกวนการทำงานระหว่างโพรเซส (Process) ต่างๆ สามารถใช้งานร่วมกับดอส (DOS) และไมโครซอฟท์วินโดวส์ (Microsoft Windows) โดยการแบ่งพาร์ติชัน (Partition) และเป็นระบบปฏิบัติการแบบเปิด เนื่องจากทุกฟังก์ชันมีซอร์สโค้ด (Source Code) แนบมาด้วย

2.8 Shell Script

Shell คือ โปรแกรมหนึ่งบนระบบยูนิกซ์ที่ทำหน้าที่เป็นอินเตอร์เฟซระหว่าง user กับยูนิกซ์ (Kernel) user สามารถสั่งงานยูนิกซ์ได้โดยผ่านทาง Shell เท่านั้น โปรแกรม Shell ยังมีคุณสมบัติของ Shell Programming Language ทำให้ user สามารถนำคำสั่งต่างๆของ Shell มาเขียนเป็นโปรแกรมเก็บเป็นไฟล์ไว้ได้ เรียกว่า Shell Script

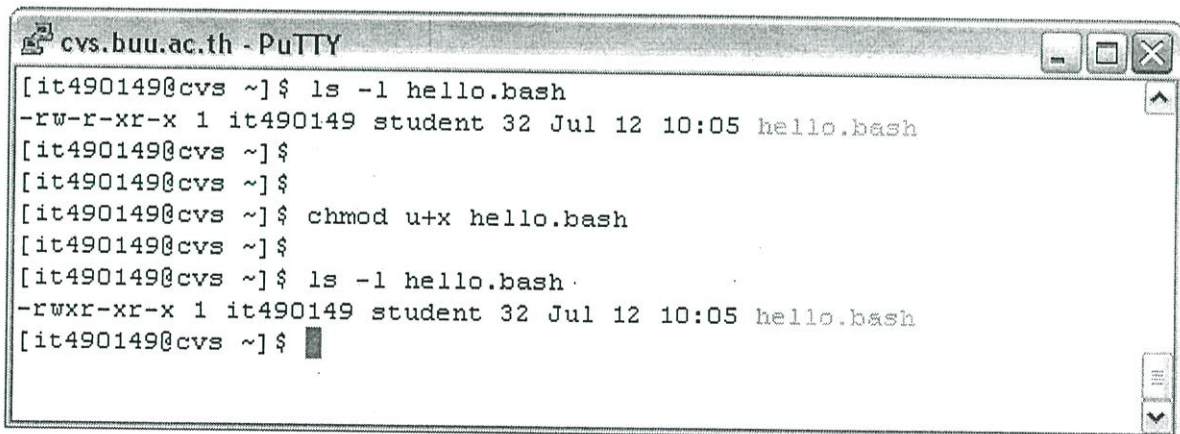
Shell script เป็นไฟล์เท็กซ์ธรรมดาสามารถสร้างด้วย editor ชนิดไหนก็ได้ เช่น vi, nano, gedit, emacs เป็นต้น

การที่จะรันสคริปต์ได้ต้องทำให้ไฟล์ที่สร้างขึ้นสามารถกระทำการได้ ถ้าดูไฟล์ด้วยคำสั่ง “ls -l hello.bash” (สมมติให้ไฟล์ชื่อว่า hello.bash) จะเห็นว่าสิทธิ์การใช้ไฟล์จะเป็น -rw-r-xr-x ซึ่งหมายความว่าไฟล์นั้นยังไม่สามารถกระทำการได้ การที่จะทำให้ไฟล์นั้นสามารถกระทำการได้ ต้องใช้คำสั่ง chmod กำหนดสิทธิ์ให้ไฟล์นั้นเสียก่อนแสดงดังรูปที่ 2.14

การเข้าใช้งาน vi (ใน prompt) : \$ vi filename

การเรียกดูสิทธิ์ในการเข้าถึงไฟล์ : \$ ls -l filename

การกำหนดสิทธิ์ให้กับไฟล์ : \$ chmod [u / g / o] [+ / -] [r / w / x] filename



```

cvs.buu.ac.th - PuTTY
[it490149@cvs ~]$ ls -l hello.bash
-rw-r-xr-x 1 it490149 student 32 Jul 12 10:05 hello.bash
[it490149@cvs ~]$
[it490149@cvs ~]$
[it490149@cvs ~]$ chmod u+x hello.bash
[it490149@cvs ~]$
[it490149@cvs ~]$ ls -l hello.bash
-rwxr-xr-x 1 it490149 student 32 Jul 12 10:05 hello.bash
[it490149@cvs ~]$
  
```

รูปที่ 2.14 การใช้คำสั่ง ls -l และ chmod

ถ้าดูไฟล์ด้วยคำสั่ง “ls -l hello.bash” จะเห็นว่าสิทธิ์การใช้ไฟล์เปลี่ยนเป็น -rwxr-xr-x ซึ่งหมายความว่า ไฟล์นั้นสามารถกระทำการได้แล้ว (x = การexecutable นั้นเอง)

การรัน shell script (ใน prompt) : \$./ filename แสดงดังรูปที่ 2.15

```

cvs.buu.ac.th - PuTTY
[~] $ ./hello.bash
Hello World
[~] $

```

รูปที่ 2.15 การรัน File hello.bash

2.8.1 ตัวแปร (variable)

เชลล์สคริปต์เหมือนกับโปรแกรมทั่ว ๆ ไปที่มีตัวแปรไว้เก็บค่าต่างๆ สำหรับใช้งานตัวแปรที่ใช้ในเชลล์นั้นไม่จำเป็นต้องประกาศชนิดตัวแปรเหมือนกับภาษาซี สามารถตั้งค่าแล้วนำไปใช้ได้ทันที

รูปแบบการตั้งค่าตัวแปร : variable = value แสดงดังรูปที่ 2.16

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
variable1=shell
variable2=script

```

รูปที่ 2.16 การสร้างตัวแปร

ข้อควรระวังคือในการใช้ตัวแปร คือในช่วงก่อนหน้าและหลังเครื่องหมาย "=" ห้ามมีช่องว่าง เพราะเชลล์ถือว่าช่องว่างคือตัวแบ่ง argument

การเรียกใช้ตัวแปร (ใน vi) : \$ variable (เวลาที่จะต้องการแสดงค่าให้ใช้เครื่องหมาย "\$" นำหน้าตัวแปร) แสดงดังรูปที่ 2.17

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
variable1=shell
variable2=script

echo $variable1
echo $variable2

```

รูปที่ 2.17 การเรียกใช้ตัวแปร

เครื่องหมาย Quote (') และ Double quote (") ช่องว่างในเชลล์นั้นถือว่าเป็นตัวแบ่งอาร์กิวเมนต์ ถ้าต้องการตั้งค่าตัวแปรที่มีช่องว่าง ให้ใช้เครื่องหมาย Quote (') หรือ Double quote (") คล่อมคำที่ต้องการตั้งค่าตัวแปรที่มีช่องว่าง (ใน vi) แสดงดังรูปที่ 2.18

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
variable4='shell script'
variable4="shell script"
4,24 Top

```

รูปที่ 2.18 การตั้งค่าตัวแปร

การตั้งค่าตัวแปรพร้อมการเรียกใช้แสดงตัวอย่างได้ดังรูป 2.19 และ 2.20

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
variable1=shell
variable2=script
variable3='shell script'
variable4="shell script"
variable5='$variable1 $variable2'
variable6="$variable1 $variable2"
variable7='$variable2 $variable1'
variable8="$variable2 $variable1"

echo $variable1
echo $variable2
echo $variable1 $variable2
echo $variable3
echo $variable4
echo $variable5
echo $variable6
echo $variable7
echo $variable8
~
20,15 All

```

รูปที่ 2.19 File testvariable.bash

```

cvs.buu.ac.th - PuTTY
[it490149@cvs ~]$ ./testvariable.bash
shell
script
shell script
shell script
shell script
$variable1 $variable2
shell script
$variable2 $variable1
script shell
[it490149@cvs ~]$

```

รูปที่ 2.20 ผลลัพธ์ที่ได้

ความแตกต่างระหว่าง Quote (') และ Double quote (") คือสิ่งที่พิมพ์ในเครื่องหมาย Quote (') จะมีค่าตามสิ่งที่พิมพ์ ส่วนในเครื่องหมาย Double quote (") จะเป็นการอ้างอิงนำผลลัพธ์ที่ตัวแปรเก็บไว้หรือค่าที่ตัวแปรเก็บไว้มาใช้

2.8.1.1 ตัวแปรสภาพแวดล้อม (environment variable)

ตัวแปรที่ใช้ในเชลล์จะมีสองชนิดด้วยกัน คือ 1.ตัวแปรธรรมดา และ 2.ตัวแปรสภาพแวดล้อม (environment variable) ตัวแปรสภาพแวดล้อมนั้นคล้ายกับตัวแปรธรรมดาแต่แตกต่างตรงที่ เมื่อโปรแกรมรันในเชลล์นั้น จะสืบทอดตัวแปรสภาพแวดล้อมและค่าที่อยู่ในตัวแปรสภาพแวดล้อมไปด้วย การเปลี่ยนตัวแปรธรรมดาให้เป็นตัวแปรสภาพแวดล้อม (ใน prompt) : \$ export variable แสดงดังรูปที่ 2.21 และ 2.22

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
# file: environment.bash

echo $ENV
echo $VAR
~
~
5,9 111

```

รูปที่ 2.21 การสร้างตัวแปรใน File testenvironment.bash

```

cvs.buu.ac.th - PuTTY
[ it490149@cvs ~ ] $ ENV="environment"
[ it490149@cvs ~ ] $ VAR="normal"
[ it490149@cvs ~ ] $
[ it490149@cvs ~ ] $ export ENV
[ it490149@cvs ~ ] $
[ it490149@cvs ~ ] $ ./testenvironment.bash
environment

[ it490149@cvs ~ ] $ . testenvironment.bash
environment
normal
[ it490149@cvs ~ ] $ █

```

รูปที่ 2.22 การเปลี่ยนตัวแปร ENV ให้เป็นตัวแปรสภาพแวดล้อมใน prompt

จะเห็นว่าผลลัพธ์นั้นต่างกัน เพราะการรัน shell script แบบแรกจะเป็นการเรียก shell (สร้างโปรเซสใหม่) และ shell ตัวใหม่จะเป็นตัวกระทำการ shell ตัวใหม่ก็คือ #!/bin/bash ที่เขียนใน file testenvironment.bash บรรทัดแรก shell (โปรแกรม) ที่เกิดใหม่สืบทอดตัวแปรสภาพแวดล้อม ENV มาด้วยทำให้แสดงผลได้ แต่ไม่สามารถแสดงค่าของตัวแปร VAR ได้เพราะตัวแปร VAR ไม่ใช่ตัวแปรสภาพแวดล้อมจึงไม่สามารถสืบทอดค่าที่อยู่ในตัวแปรไปใช้ในโปรแกรมอื่นได้

ส่วนการรันแบบที่สองใช้การรันแบบ "." (dot) เป็นการใช้ shell ที่ทำงานอยู่ในปัจจุบันอ่านไฟล์ testenvironment.bash จึงสามารถแสดงค่าของตัวแปรทั้งสองตัวได้

เราสามารถรู้ได้ว่ามีตัวแปรสภาพแวดล้อมอะไรบ้างที่อยู่ในระบบ โดยการใช้คำสั่ง export ซึ่งเป็นคำสั่งเดียวกันกับที่ใช้ประกาศตัวแปรสภาพแวดล้อม

เมื่อลองใช้คำสั่งดูจะเห็นได้ว่ามีตัวแปรสภาพแวดล้อมมากมายที่ตั้งค่าไว้อัตโนมัติในระบบถ้าต้องการเปลี่ยนตัวแปรสภาพแวดล้อมเป็นตัวแปรธรรมดาสามารถทำได้โดยใช้ option -n ต่อท้ายคำสั่ง export คือ \$ export -n variable

2.8.1.2 ตัวแปรพิเศษ(environment variable)

ตัวแปรพิเศษเหล่านี้โดยปกติเชลล์จะเตรียมตัวแปรเหล่านี้ไว้ให้อยู่แล้วไม่ต้องสร้างเองและมักจะเป็นตัวแปรที่ไม่สามารถเปลี่ยนค่าได้ (read-only) ดังตารางที่ 2.5

ตารางที่ 2.5 ตัวแปรพิเศษที่นิยมใช้

ตัวแปร	คำอธิบาย
\$0 \$1 \$2	ใช้สำหรับอ้างอิงชื่อ shell script และ Argument ของ shell script เรียกว่า position parameter
\$#	ใช้บอกจำนวน Argument ที่อยู่ใน shell script นั้นๆ
\$*	แทน Argument ของ script เรียงกันทั้งหมด
\$@	คล้ายกับ \$* แต่จะใช้ช่องว่างคั่นระหว่าง position parameter
\$?	ใช้แสดงสถานการณ์จบการทำงานครั้งสุดท้าย ถ้าสั่งได้ถูกต้องไม่มี error จะ

	แสดงค่า 0 ออกมา แต่ถ้าผิดพลาดจะแสดงค่าที่ไม่ใช่ 0 ออกมา
\$!	ใช้แสดง process ID ของ shell ที่ทำงานอยู่
\$hostname	ใช้แสดงชื่อเครื่องที่กำลังใช้งานอยู่
\$pwd	ใช้แสดง directory ที่ทำงานอยู่
\$oldpwd	ใช้แสดง directory ก่อนหน้า directory ปัจจุบัน
\$randow	ใช้ในการสุ่มตัวเลข ตั้งแต่ 0 ถึง 32767
\$home	ใช้แสดงชื่อ home directory

หมายเหตุ เครื่องหมาย “ # ” ใช้แสดงหมายเหตุ เขียนไว้ตรงไหนในเชลล์สคริปต์ก็ได้ (ยกเว้นบรรทัดแรก) ใช้สำหรับการเขียนบันทึกใน script โดย shell จะถือว่าข้อความที่อยู่หลังเครื่องหมาย “ # ” เป็นหมายเหตุไม่มีผลต่อตัวโปรแกรม

2.8.1.3 ตัวแปรแถวลำดับ (array variable)

ตัวแปร array นั้นจะมี index เป็นตัวเลขเริ่มตั้งแต่ 0 เหมือนกับตัวแปรในภาษา C หรือภาษาอื่นๆ แต่เวลาที่ต้องการเรียกใช้ตัวแปรต้องใช้เครื่องหมาย “ { } ” ช่วยจับกลุ่มด้วย เพราะไม่เช่นนั้นค่าที่ออกมาจะออกมาเฉพาะค่าแรกเท่านั้น แสดงดังรูป 2.23

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
#file: testarray.bash

text[0]="ant"
text[1]="bee"
text[2]="cat"
text[3]="dog"
text[4]="fish"

echo $text[0] $text[1] $text[2] $text[3] $text[4]
echo ${text[0]} ${text[1]} ${text[2]} ${text[3]} ${text[4]}
~
12,0-1 All

```

รูปที่ 2.23 File testarray.bash

เราสามารถประกาศตัวแปรพร้อมกับการตั้งค่าไปในตัวได้ ดังรูป 2.24

```

cvs.buu.ac.th - PuTTY
#!/bin/bash
# file testarray2.bash

text=(ant bee cat dog fish)

echo $text[0] $text[1] $text[2] $text[3] $text[4]
echo ${text[0]} ${text[1]} ${text[2]} ${text[3]} ${text[4]}

```

รูปที่ 2.24 File testarray2.bash

2.8.2 test และ expression

ใน bash และ ksh จะมีคำสั่ง test และการเขียนเงื่อนไขในการตรวจสอบ expression ต่างๆ ในการเขียนประโยคเงื่อนไข Expression มีคำสั่งที่เกี่ยวข้องดังตารางที่ 2.6 – 2.9 การใช้คำสั่ง test (ใน vi) : test expression หรือ [expression]

ตารางที่ 2.6 คำสั่งเกี่ยวกับจำนวนเต็ม

คำสั่ง	คำอธิบาย
int1 -eq int2	เป็นจริงเมื่อ int1 เท่ากับ int2
int1 -ne int2	เป็นจริงเมื่อ int1 ไม่เท่ากับ int2
int1 -gt int2	เป็นจริงเมื่อ int1 มากกว่า int2
int1 -ge int2	เป็นจริงเมื่อ int1 มากกว่า หรือ เท่ากับ int2
int1 -le int2	เป็นจริงเมื่อ int1 น้อยกว่า int2
int1 -lt int2	เป็นจริงเมื่อ int1 น้อยกว่า หรือ เท่ากับ int2

ตารางที่ 2.7 คำสั่งเกี่ยวกับ string

คำสั่ง	คำอธิบาย
str1 = str2	เป็นจริงเมื่อ str1 เหมือนกับ str2
str1 != str2	เป็นจริงเมื่อ str1 ไม่เหมือนกับ str2
Str	เป็นจริงเมื่อ str ไม่เป็นค่าว่าง
-n str	เป็นจริงเมื่อ str มีความยาวมากกว่า 0
z str	เป็นจริงเมื่อ str มีความยาวเป็น 0

ตารางที่ 2.8 คำสั่งเกี่ยวกับ file

คำสั่ง	คำอธิบาย
-d filename	เป็นจริงเมื่อ filename เป็น directory
-f filename	เป็นจริงเมื่อ filename เป็น file

-r filename	เป็นจริงเมื่อ filename อ่านได้โดยโปรแกรม
-w filename	เป็นจริงเมื่อ filename เขียนได้โดยโปรแกรม
-x filename	เป็นจริงเมื่อ filename run ได้โดยโปรแกรม
-s filename	เป็นจริงเมื่อ filename มีขนาดไม่เป็น 0

ตารางที่ 2.9 คำสั่งเกี่ยวกับ Logical อื่นๆ

คำสั่ง	คำอธิบาย
!expr	เป็นจริงเมื่อ exp เป็นเท็จ
exp1 -a exp2	เป็นจริงเมื่อ exp1 และ exp2 เป็นจริง
exp1 -o exp2	เป็นจริงเมื่อ exp1 หรือ exp2 เป็นจริง

2.8.3 การใช้เงื่อนไข if

คำสั่ง if เป็นคำสั่งที่ใช้ในการเปรียบเทียบ รูปแบบคำสั่ง if (ใน vi) มีดังนี้

```
if [ expression ]
then
    commands
elif [ expression ]
then
    commands
else
    commands
if
```

2.8.4 การใช้เงื่อนไข (case หรือ switch)

คำสั่ง case เป็นคำสั่งที่ใช้ในการเปรียบเทียบเช่นเดียวกับคำสั่ง if แต่มีรูปการทำงานที่ต่างกัน รูปแบบคำสั่ง case (ใน vi) มีดังนี้

```
case string in
    str1)
        commands;
    str2)
        commands;
    str3)
        commands;
    *)
        commands;
esac
```

2.8.5 การใช้ loop while

คำสั่ง while เป็นคำสั่งที่ใช้ในการวนรอบการทำงานของโปรแกรม โปรแกรมจะวนรอบจนกว่าเงื่อนไขที่กำหนดไว้จะเป็นเท็จ รูปแบบคำสั่ง while (ใน vi) มีดังนี้

```
while [ expression ]
do
    commands
    .....
Done
```

2.8.6 การใช้ loop for

คำสั่ง for เป็นคำสั่งที่มีการทำงานคล้ายคลึงกับ คำสั่ง while แต่จะต่างกันตรงที่การเขียนเงื่อนไขที่ใช้ในการวน loop รูปแบบคำสั่ง for (ใน vi) มีดังนี้

```
for var in list
do
    commands
    .....
Done
```

2.9 โมดูลแอลซีดี (Liquid Crystal Display)

โมดูลแอลซีดี (LCD) เป็นอุปกรณ์แสดงผลแบบหนึ่งได้รับความนิยมในการใช้งานสูงมากโดยเฉพาะอย่างยิ่งเมื่อนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์ ข้อมูลเบื้องต้นของโมดูลแอลซีดีสามารถดูได้ดังตารางที่ 2.10

ตารางที่ 2.10 รายละเอียดตำแหน่งขั้วขาและหน้าที่การใช้งานโมดูลแอลซีดี

ตำแหน่งขา	สัญลักษณ์	การต่อใช้งาน	ระดับ	การทำงาน
1	VSS	Ground	-	Ground
2	VDD	Power Supply	-	ต่อกับแรงดันไฟเลี้ยง+5V
3	VO	LCD Control	-	ต่อกับแรงดันเพื่อปรับความเข้มของการแสดงผล
4	RS	Register Select	H/L	RS = 0 หมายถึงต้องการติดต่อกับรีจิสเตอร์คำสั่ง RS = 1 หมายถึงต้องการติดต่อกับรีจิสเตอร์ข้อมูล

ตำแหน่งขา	สัญลักษณ์	การต่อใช้งาน	ระดับ	การทำงาน
5	R/W	Read /write	H/L	R/W = 0 หมายถึงต้องการเขียนข้อมูลไปยัง LCD โมดูล R/W = 1 หมายถึงต้องการอ่านข้อมูลจาก LCD โมดูล
6	E	Enable	H,H->L	Enable Signal
7-14	DB0-DB7	Data Bus	H/L	Data Bus Line
15	A	Back Light A	-	Back Light +5V (สำหรับรุ่นที่มี Back Light)
16	K	Back Light K	-	Back Light 0V (สำหรับรุ่นที่มี Back Light)

2.9.1 คำสั่งควบคุมโมดูลที่สำคัญ

2.9.1.1 คำสั่งเคลียร์จอแสดงผล

มีข้อมูลคำสั่งเป็น 0x01 เป็นคำสั่งเขียนข้อมูลช่องว่างเข้าไปในหน่วยความจำข้อมูลสำหรับการแสดงผลหรือ DDRAM ภายในโมดูล เมื่อเคลียร์จอแสดงผลแล้วจะกำหนดให้ตัวชี้กลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล

2.9.1.2 คำสั่งกลับสู่หน้าหลัก

มีค่าข้อมูลเท่ากับ 0x02 หรือ 0x03 ก็ได้เป็นการกำหนดให้ตัวชี้ไปแสดงยังตำแหน่งซ้ายสุดของจอแสดงผลโดยข้อมูลที่แสดงบนจอแสดงผลจะไม่มีเปลี่ยนแปลง

2.9.1.3 คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode set)

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

บิต I/D ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของหน่วยความจำแสดงผล (DDRAM) เพิ่มหรือลดลง

ก. "0" แอดเดรสลดลง 1 แอดเดรส

ข. "1" แอดเดรสเพิ่มขึ้น 1 แอดเดรส

ค. บิต S ใช้กำหนดลักษณะในการแสดงผล

ง. "0" เมื่อเกิดตัวอักษรใหม่ตัวชี้เลื่อนไปทางขวามือ

จ. "1" เมื่อเกิดตัวอักษรใหม่ตัวชี้อยู่ที่เดิมแต่ตัวอักษรเลื่อนไปทางซ้าย

ที่นิยมใช้มากที่สุดคือ ข้อมูลคำสั่ง 0x06 หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ตัวชี้จะเลื่อนไปทางขวามือ และแอดเดรสของหน่วยความจำสำหรับแสดงผลจะเพิ่มขึ้น

2.9.1.4 คำสั่งควบคุมการแสดงผล

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	1	B	C	D

ก. บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าเป็น "0" เป็นการปิดจอแสดงผล ถ้าเป็น "1" เป็นการเปิดจอแสดงผล

ข. บิต C ใช้ควบคุมการแสดงผลตัวชี้บนตัวแสดงผล ถ้าเป็น "0" เป็นการปิดตัวชี้หรือไม่แสดงผลตัวชี้ ถ้าเป็น "1" เป็นการแสดงผลตัวชี้

ค. บิต B ใช้ควบคุมการกระพริบของตัวชี้ ถ้าต้องการให้ตัวชี้กระพริบ ต้องกำหนดให้เป็น "1" คำสั่งที่ใช้บ่อยคือ 0x0C เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงผลตัวชี้ และ 0x0F เป็นการสั่งให้เปิดจอแสดงผลแสดงผลตัวชี้

2.9.1.5 คำสั่งควบคุมการเลื่อนตัวชี้ และตัวอักษร

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	S/C	R/L	*	*

การเลื่อนตัวชี้และตัวอักษรขึ้นอยู่กับกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังตารางที่

2.11

ตารางที่ 2.11 คำสั่งควบคุมการเลื่อนตัวชี้และตัวอักษร

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนตัวชี้ไปทางซ้าย	0x10 - 0x13
0	1	เลื่อนตัวชี้ไปทางขวา	0x14 - 0x17
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	0x18 - 0x1B
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	0x1C - 0x1F

2.9.1.6 คำสั่งกำหนดฟังก์ชันการทำงาน เป็นคำสั่งที่มีความสำคัญมากที่สุดอีกคำสั่งหนึ่ง มีรูปแบบของข้อมูลคำสั่งดังนี้

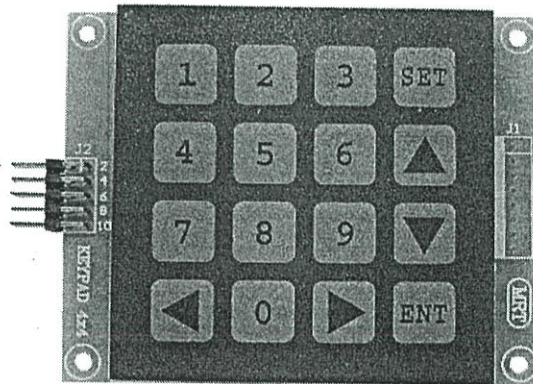
บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	1	DL	N	F	*	*

ก. บิต DL ใช้ในการกำหนดจำนวนบิตในการติดต่อกับโมดูล LCD บิต "0" กำหนดให้ทำงานในโหมด 4 บิต "1" กำหนดให้ทำงานในโหมด 8 บิต

ข. บิต N ใช้กำหนดจำนวนบรรทัดที่ต้องการใช้แสดงผล "0" หมายถึง แสดงผล 1 บรรทัด และ "1" หมายถึงแสดงผล 2 บรรทัด

2.10 คีย์แพด ET-TOUCH PAD 4x4

คีย์แพดดังรูปที่ 2.25 เป็นอุปกรณ์สำหรับรับอินพุตจากผู้ใช้งาน มีลักษณะเป็นปุ่มกดหลายปุ่ม ถูกจัดเรียงกันในลักษณะเป็นอาร์เรย์ แบ่งเป็นแถวแนวนอน (Rows) และแถวแนวตั้ง (Columns) เช่น 3x4 (= 12 ปุ่ม) หรือ 4x4 (= 16 ปุ่ม) เป็นต้น แต่ละปุ่มก็จะมีสัญลักษณ์เขียนกำกับไว้ เช่น ตัวเลข 0-9, #, * เป็นต้น โดยปกติถ้าต่อปุ่มกดแยกจำนวน 16 ตัว จะต้องใช้ขาสัญญาณทั้งหมด 16 ขา แต่ถ้าใช้การจัดเรียงแบบ 4x4 จะใช้ขาสัญญาณเพียง 8 ขา แต่ต้องมีเทคนิคในการตรวจดูว่า ปุ่มกดใดถูกกดบ้างในขณะนั้น วิธีการนี้เรียกว่า การสแกนปุ่มกด (key scan)



รูปที่ 2.25 คีย์แพดขนาด 4x4

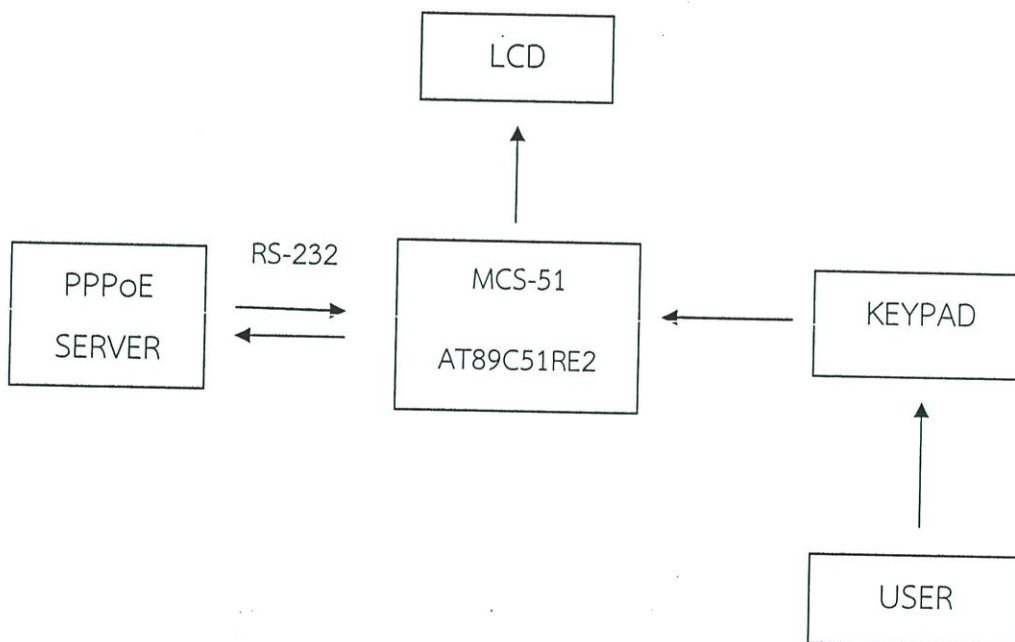
บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานรวมของระบบ

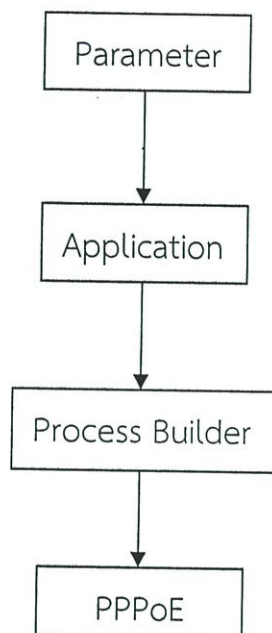
การควบคุมแบนด์วิดธ์สำหรับหลายผู้ใช้บริการ ใช้หลักการของโปรโตคอลพีพีพี (Point-to-Point) วิ่งผ่านไปในระบบอินเทอร์เน็ต หรือเรียกว่า PPPoE เป็นการทำงานบนระบบปฏิบัติการลินุกซ์ (Linux) พัฒนาด้วยภาษาจาวาร่วมกับโปรเซสซิงเตอร์ไลบรารี และเชลล์สคริปต์ (Shell Script) โดยใช้โมดูลแอลซีดีเป็นหน้าพาเนลสำหรับแสดงผลข้อมูลและป้อนคำสั่งผ่านคีย์แพดได้โดยตรง โดยใช้อาร์เอส 232 เป็นพอร์ตรับส่งข้อมูลระหว่างส่วนของไมโครคอนโทรลเลอร์และส่วนของ PPPoE Server โดยภาพรวมของระบบจะแสดงดังรูปที่ 3.1 จะแสดงให้เห็นว่าการควบคุมแบนด์วิดธ์สำหรับหลายผู้ใช้นั้นสามารถแบ่งเป็น 2 ส่วนหลักๆ คือ ส่วนของ PPPoE Server และส่วนของไมโครคอนโทรลเลอร์



รูปที่ 3.1 บล็อกไดอะแกรมรวมของระบบ

3.1.1.1 ส่วนของ PPPoE Server

ในส่วนของ PPPoE Server ได้ทำการพัฒนาแอปพลิเคชันโดยใช้ภาษาจาวา เพื่อรับค่าพารามิเตอร์จากไมโครคอนโทรลเลอร์ในการแบ่งแบนด์วิดธ์ โดยนำค่าพารามิเตอร์ที่ได้รับไปรวมเป็นคำสั่งในรูปแบบของเซอร์วิส PPPoE โดยบล็อกไดอะแกรมของแอปพลิเคชันแสดงตามรูปที่ 3.2



รูปที่ 3.2 บล็อกไดอะแกรมของแอปพลิเคชัน

จากรูปที่ 3.2 เมื่อแอปพลิเคชันได้รับพารามิเตอร์แล้ว แอปพลิเคชันจะทำการเก็บค่าพารามิเตอร์และทำการตัดสตริง (string) เพื่อนำค่าพารามิเตอร์ไปรวมกับซินแทกซ์ (Syntax) ทำเป็นชุดคำสั่ง (Command) ซึ่งพารามิเตอร์ดังกล่าว ได้แก่ ไอพีแอดเดรส (IP Address) ,ท่อ (pipe) และ speed ของ upload และสปีดของอัปโหลดและดาวน์โหลด (Speed of upload and download) โดยใช้ภาษาจาวาตามรูปแบบของเซอร์วิส PPPoE จากนั้นเมื่อได้ชุดคำสั่งแล้วแอปพลิเคชันจะทำการส่งชุดคำสั่งไปยังระบบปฏิบัติการลินุกซ์เพื่อใช้งาน PPPoE ในการควบคุมแบนด์วิดธ์ ซึ่งในการติดต่อกันระหว่างแอปพลิเคชันกับระบบปฏิบัติการลินุกซ์จะอาศัยกระบวนการของโปรเซสบิวเดอร์ (Process Builder) เป็นตัวกลางในการสื่อสาร

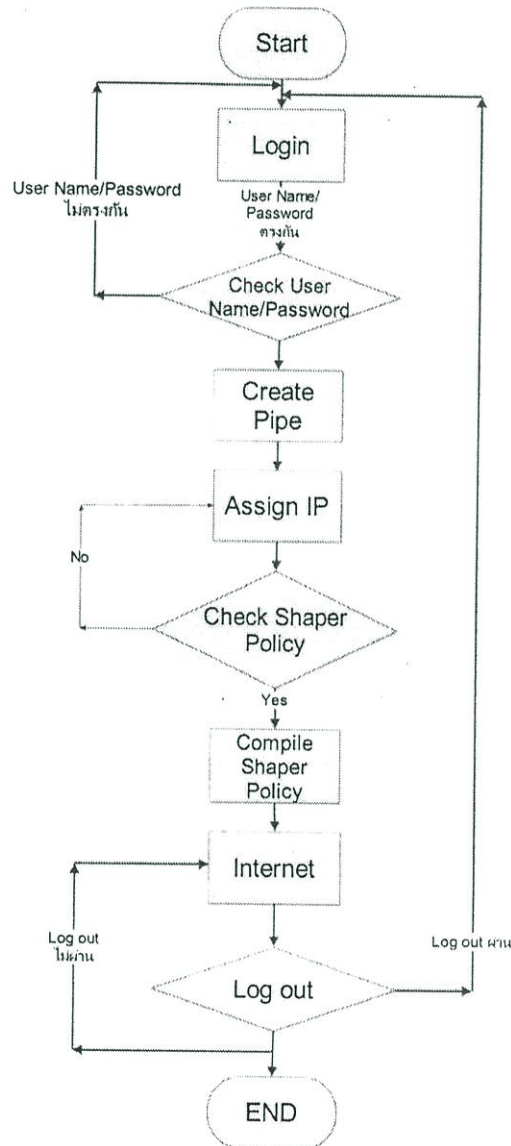
หลังจากที่ได้รับชุดคำสั่งจากกระบวนการโปรเซสบิวเดอร์ โปรเซสบิวเดอร์จะทำการส่งชุดคำสั่งไปยังชั้นของเชลล์ (shell) บนระบบปฏิบัติการลินุกซ์ โดยผ่านเชลล์สคริปต์ (shell script) ซึ่งก็คือการรวมซินแทกซ์ของชุดคำสั่งที่ได้เข้าด้วยกันเป็นชุดคำสั่งในรูปแบบของเซอร์วิส PPPoE ซึ่งถ้าไม่มีชั้นเชลล์สคริปต์จะส่งได้เพียงทีละคำสั่งหรือคอมมานด์ไลน์ (command line) หลังจากนั้นเชลล์สคริปต์จะทำการติดต่อกับเซอร์วิส PPPoE เพื่อส่งชุดคำสั่งเข้าไปทำงานเพื่อทำการควบคุมแบนด์วิดธ์ตามที่ต้องการ

3.1.1.2 ส่วนของไมโครคอนโทรลเลอร์

ในส่วนของไมโครคอนโทรลเลอร์จะใช้ MSC-51 AT89C51RE2 ภายในจะบรรจุชุดคำสั่งภาษาซี ซึ่งประกอบด้วยส่วนของคำสั่งสแกนคีย์สำหรับให้คีย์แปดทำงานได้ตามที่กำหนด ในส่วนนี้จะมีการกำหนดชุดคำสั่งสำหรับการส่งข้อมูลจากผู้ใช้งานผ่านคีย์แปดส่งผ่านพอร์ตสื่อสารอนุกรมอาร์เอส 232 ไปยัง PPPoE Server ชุดคำสั่งสำหรับการรับค่าที่ส่งมาจาก PPPoE Server ถูกควบคุมให้แสดงผลบนหน้าจอโมดูลแอลซีดี

3.1.2 การออกแบบ การทำงานของชุดคำสั่งของระบบ

การออกแบบการทำงานของชุดคำสั่งของระบบ แสดงโพล์ชาร์ตได้ดังรูปที่ 3.3



รูปที่ 3.3 โพล์ชาร์ตการทำงานของชุดคำสั่งของระบบ

จากรูปที่ 3.3 ได้ออกแบบการทำงานของชุดคำสั่งของระบบ เริ่มต้นผู้ใช้งานทำการล็อกอิน (Log in) เข้าสู่ระบบ โดยทำการเช็คยูสเซอร์ (User) และพาสเวิร์ด (Password) ถ้าล็อกอินถูกต้องเซิร์ฟเวอร์จะทำการสร้างท่อขึ้นมาแต่ถ้าล็อกอินไม่ถูกต้องจะต้องทำการล็อกอินใหม่อีกครั้ง หลังจากทำการสร้างท่อเสร็จระบบจะทำการ Assign IP นั่นก็คือทำการแจกไอพีให้กับเครื่องลูกข่าย จากนั้นทำการ Check Shaper Policy ในแต่ละท่อว่ามีหรือไม่ ถ้ามีให้ทำการ Compile Shaper Policy นั่นก็คือการบีบแบนด์วิดธ์นั่นเอง แต่ถ้าไม่มีให้ออกอินเทอร์เน็ท และถ้ามีการล็อกเอาท์ (Log out) การทำงานสิ้นสุดทันทีแต่ถ้าผู้ใช้งานยังใช้งานอย่างต่อเนื่องระบบจะให้เล่นอินเทอร์เน็ทต่อไป

3.1.3 การออกแบบหน้าจอแอลซีดี (LCD)

ในส่วนของหน้าจอแอลซีดีได้ทำการออกแบบฟังก์ชันการใช้งาน แบ่งเป็น 4 ฟังก์ชันหลัก ซึ่งมีรายละเอียดดังนี้

3.1.3.1 ฟังก์ชัน Show system info เป็นฟังก์ชันที่ออกแบบไว้สำหรับบอกถึงข้อมูลของระบบ ประกอบด้วย

- 1) CPU Utilize จะบอกถึงการใช้ประสิทธิภาพของซีพียู (CPU) โดยจะบอกเป็นเปอร์เซ็นต์
- 2) HD จะบอกถึงจำนวนฮาร์ดดิสก์ที่ใช้งานต่อจำนวนฮาร์ดดิสก์ที่เหลือให้ใช้งาน
- 3) Ram Used และ Ram free จะบอกถึงจำนวนแรมที่ใช้งานและจำนวนแรมที่เหลือให้ใช้งาน

3.1.3.2 ฟังก์ชัน System setting เป็นฟังก์ชันที่ออกแบบไว้สำหรับตั้งค่าพารามิเตอร์ให้กับระบบ มีฟังก์ชันย่อยคือ Policy Based Bandwidth Shaping ประกอบด้วย

- 1) Set interface IP ใช้สำหรับตั้งค่าการ์ดแลนของเครื่องเซิร์ฟเวอร์ (Eth0 Eth1 Gateway) โดยกำหนดไอพีแอดเดรส (IP Address) ให้กับการ์ดแลนเพื่อให้เป็นทางเข้าและออกของระบบเครือข่าย
- 2) Speed Allocation ใช้สำหรับตั้งค่าสปีดของอัปโหลดและดาวน์โหลด
- 3) Show Summary ใช้สำหรับแสดงอินเตอร์เฟซที่เรากำหนดไว้ (Eth0 Eth1 Gateway) และใช้ในการแก้ไขอินเตอร์เฟซต่างๆ
- 4) No Implement
- 5) Main Menu ใช้สำหรับกลับสู่หน้าหลัก

3.1.3.3 ฟังก์ชัน Service Control ควบคุมคำสั่งพื้นฐานสำหรับควบคุมเครื่องเซิร์ฟเวอร์ ประกอบด้วย

- 1) Reboot system ใช้สำหรับรีสตาร์ทเครื่องเซิร์ฟเวอร์
- 2) Factory Default ใช้สำหรับเคลียร์พารามิเตอร์ทั้งหมดทิ้ง

- 3) Shutdown System ใช้สำหรับปิดเครื่องเซิร์ฟเวอร์
- 4) Back to Main Menu ใช้สำหรับกลับสู่หน้าหลัก
- 5) Log Out ใช้สำหรับล็อกเอาท์ออกจากระบบ

3.1.3.4 ฟังก์ชัน Concurrent List สำหรับแสดงข้อมูลกรณีที่มีการแอสซายน์ไอพีแอดเดรสสปีดของอัปโหลดและดาวน์โหลด

3.2 อุปกรณ์ที่ใช้ในการทดลอง

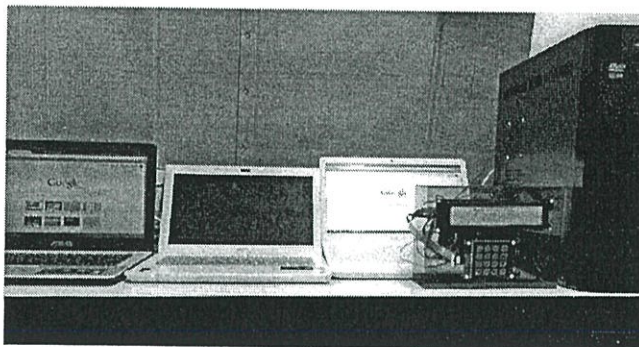
อุปกรณ์ที่ใช้ในการทดลอง ประกอบด้วย

- 1.บอร์ดไมโครคอนโทรลเลอร์
- 2.เครื่องเซิร์ฟเวอร์
- 3.Power supply
- 4.อาร์เอส 232
- 5.จอคอมพิวเตอร์
- 6.สวิชต์ I2

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทำงานของชุดคำสั่งสำหรับเซตค่าข้อมูลในฝั่งแอดมิน

ในการทดลองการทำงานของระบบจะต้องทำการติดตั้งอุปกรณ์ทั้งหมดเข้าด้วยกันดังรูปที่ 3.5 และเริ่มต้นการทดลองโดยการล็อกอินเข้าสู่ระบบโดยใส่พาสเวิร์ด ทำการเซตไอพีแอดเดรสให้กับอัปลิงค์ (Uplink) และดาวน์โหลดลิงค์ (Downlink) และเซตค่าสปีดของอัปโหลดและดาวน์โหลด ทำการบันทึกผลและจัดเก็บผลการทดลอง



รูปที่ 3.4 การต่ออุปกรณ์ทั้งหมดเข้าด้วยกัน

3.3.2 การทำงานในฝั่งผู้ใช้งาน (User)

ในส่วนการทำงานในฝั่งผู้ใช้งาน เริ่มต้นให้ทำการเชื่อมต่ออินเทอร์เน็ต โดยเลือกสัญญาณ Wi-Fi ที่ชื่อ T304-PPPoE-Project เมื่อเชื่อมต่อเสร็จแล้วทำการใส่ User name, Password และ connect name จากนั้นให้ผู้ใช้งานทำการดาวน์โหลดข้อมูล การจัดเก็บผลการทดลองโดยแสดงกราฟเปรียบเทียบการดาวน์โหลดข้อมูลระหว่างทำการบีบแบนด์วิดท์แล้วและยังไม่ทำการบีบแบนด์วิดท์ โดยในการบีบแบนด์วิดท์จะควบคุมผ่านไมโครคอนโทรลเลอร์ เมื่อทำการกดคีย์แพดส่งคำสั่งให้ระบบที่ตั้งค่าพารามิเตอร์ไว้แอคทีฟ (Active) การอัปโหลดและดาวน์โหลดก็จะไม่เกินค่าที่เราได้กำหนดไว้

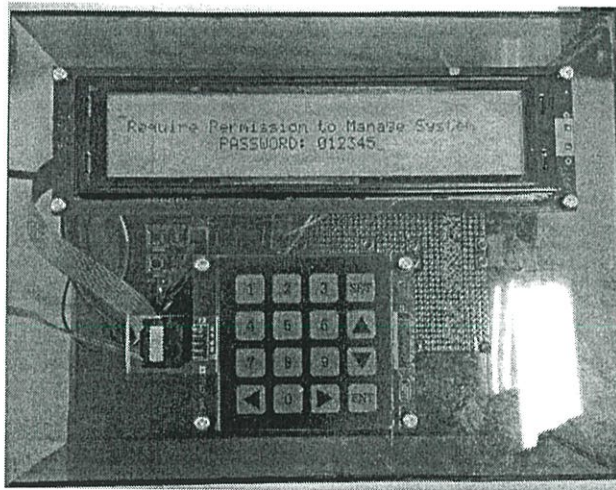
บทที่ 4

ผลการทดลอง

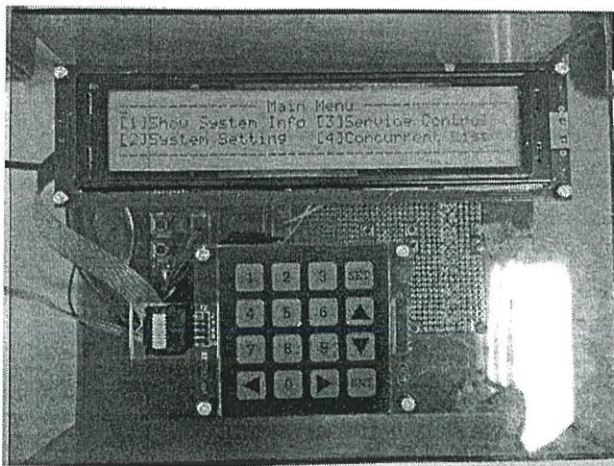
ในบทนี้จะกล่าวถึงผลการทดลองในส่วนของการทำงานของชุดคำสั่งสำหรับการตั้งค่าข้อมูลในฝั่งแอดมินซึ่งมีการตั้งค่า IP Address และ Gateway ของฝั่ง Uplink กับ Downlink, Speed Upload และ Speed Download ของแต่ละท่อ มีการดูข้อมูลการตั้งค่าต่างๆ และผลการทดลองการทำงานในฝั่งผู้ใช้งาน (User) ที่ทำการล็อกอินเข้ามาใช้งาน มีการเปรียบเทียบระหว่างการดาวน์โหลดข้อมูลที่มีการบีบแบนด์วิดท์และไม่ได้ทำการบีบแบนด์วิดท์ โดยมีรายละเอียดดังนี้

4.1 ผลการทดลองการทำงานของชุดคำสั่งสำหรับตั้งค่าข้อมูลในฝั่งแอดมิน

ในการทดลองชุดคำสั่ง หลังจากต่ออุปกรณ์ทั้งหมดเข้าด้วยกันแล้ว เริ่มต้นให้ทำการเข้ารหัส จะปรากฏหน้า Main Menu ดังรูปที่ 4.1 และ 4.2 จากนั้นทำการเซตค่าต่างๆ โดยทำการเซตดังนี้



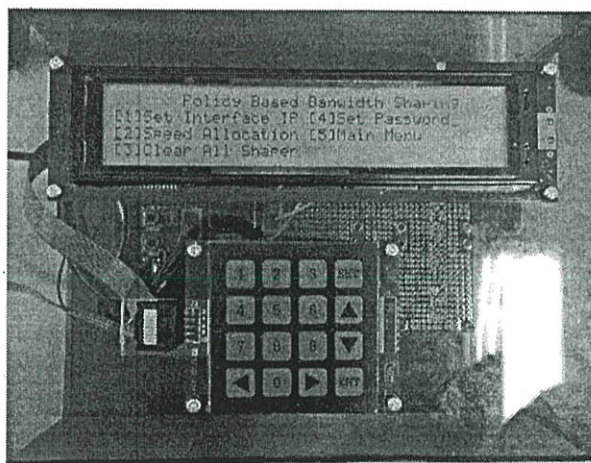
รูปที่ 4.1 หน้าต่างสำหรับใส่รหัส



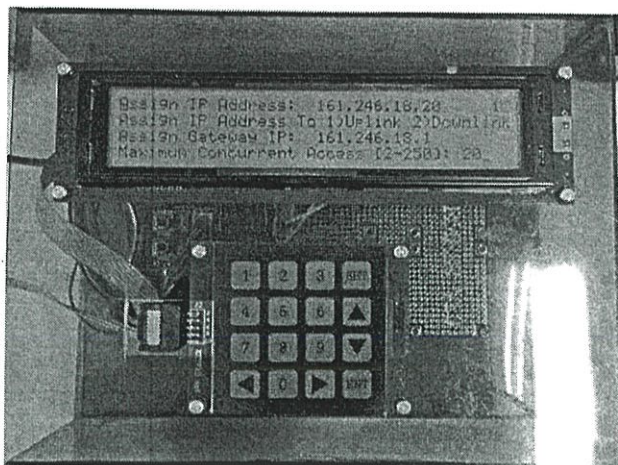
รูปที่ 4.2 หน้า Main Menu

4.1.1 การตั้งค่า IP Address ให้กับ uplink

ทำการตั้งค่า IP Address โดยทำการกดปุ่มหมายเลข 1 Set Interface IP ของหน้า Policy Base Bandwidth Shaping จากนั้นตั้งค่า IP Address เป็น 161.246.18.20 และ Gateway IP เป็น 161.246.18.1 และทำการกดปุ่มหมายเลข 1 uplink เป็นการกำหนดผู้ใช้ 20 เครื่อง ดังรูปที่ 4.3 และ 4.4



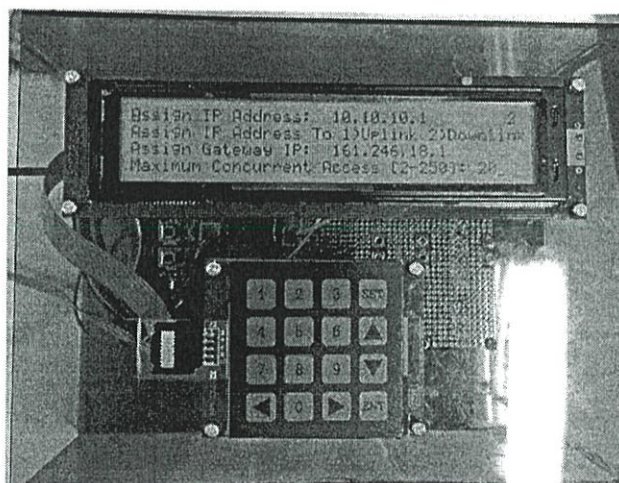
รูปที่ 4.3 หน้า Policy Based Bandwidth Shaping



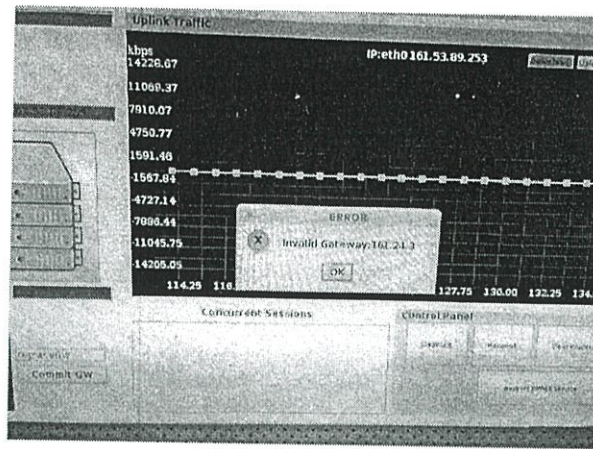
รูปที่ 4.4 Set Interface IP ของ Uplink

4.1.2 การตั้งค่า IP Address ให้กับ downlink

ทำการตั้งค่า IP Address โดยทำการกดปุ่มหมายเลข 1 Set Interface IP ของหน้า Policy Base Bandwidth Shaping จากนั้นตั้งค่า IP Address เป็น 10.10.10.1 และ Gateway IP เป็น 161.246.18.1 และทำการกดปุ่มหมายเลข 2 downlink เป็นการกำหนดผู้ใช้ 20 เครื่อง ดังรูปที่ 4.5 และหากใส่ IP ผิดจะปรากฏดังรูปที่ 4.6



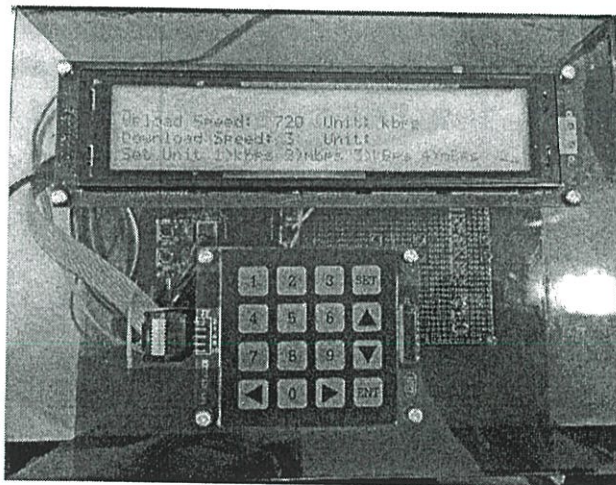
รูปที่ 4.5 Set Interface IP ของ Downlink



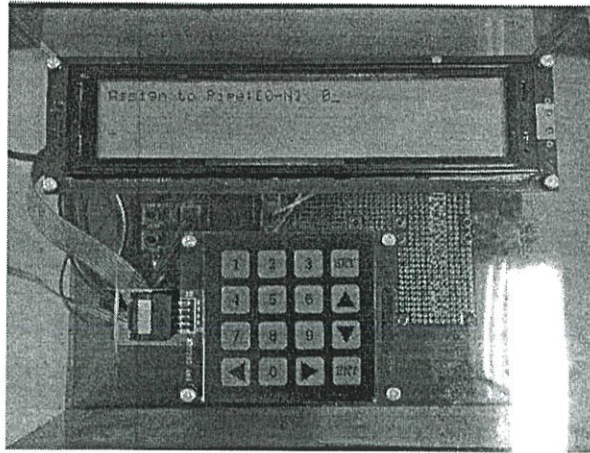
รูปที่ 4.6 หน้าต่างแสดงการใส่ค่า IP ผิด

4.1.3 การตั้งค่าอัตราเร็วข้อมูลในการ Upload และ Download

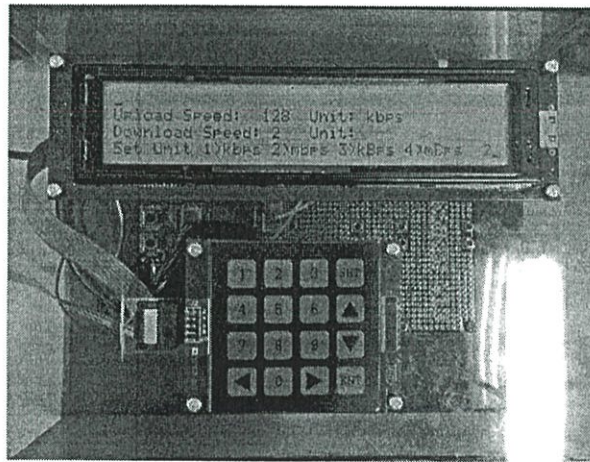
ทำการตั้งค่า speed โดยทำการกดปุ่มหมายเลข 2 Speed allocation ของหน้า Policy Base Bandwidth Shaping ให้กับ Upload เป็น 720 kbps และ Download เป็น 3 Mbps ของท่อน 0 ดังรูปที่ 4.7 และ 4.8 และทำการตั้งค่า Speed ของ Upload และ Download ของท่อน 1 โดยจะให้ Upload เป็น 128 kbps และ Download เป็น 2 Mbps ดังรูปที่ 4.9 และ 4.10



รูปที่ 4.7 การตั้งค่าอัพโหลดและดาวน์โหลดของท่อน 0



รูปที่ 4.8 การเลือกท่อ 0

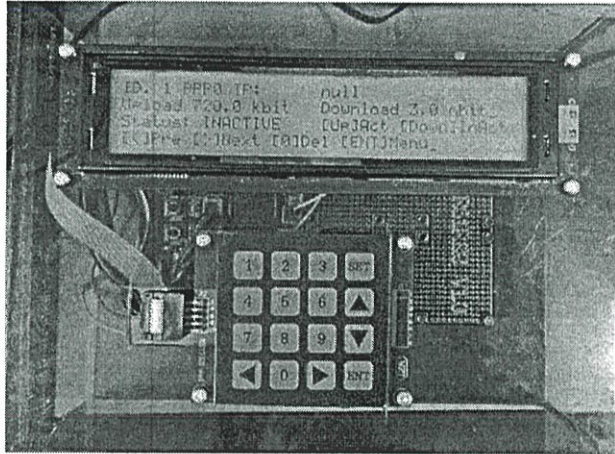


รูปที่ 4.9 การตั้งค่าอัพโหลดและดาวน์โหลดของท่อ 1

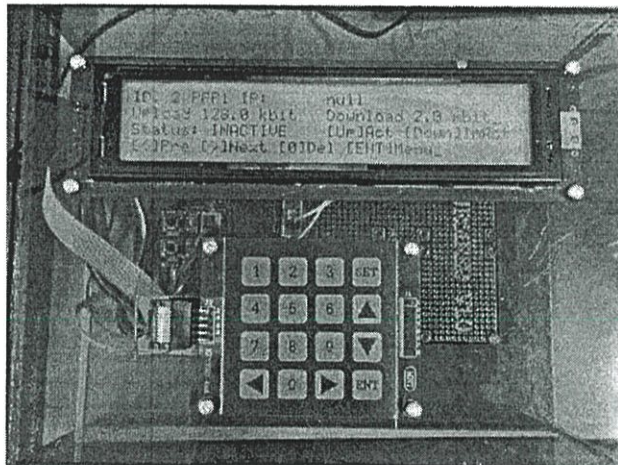


รูปที่ 4.10 การเลือกท่อ 1

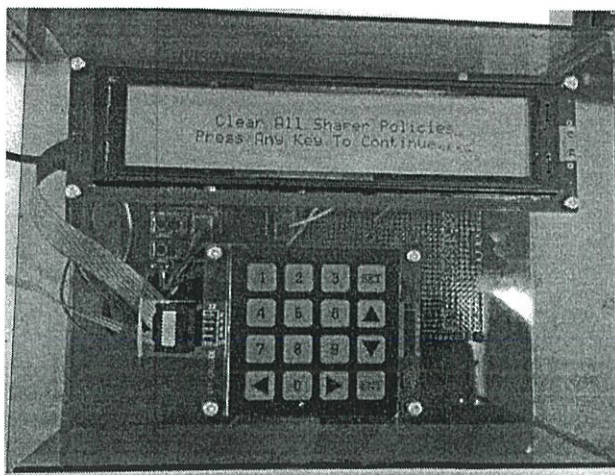
เราสามารถตั้งค่า Speed ของ Upload และ Download ก็ทำได้ตามต้องการ เมื่อตั้งค่าต่างๆครบแล้ว ทำการกดปุ่มหมายเลข 5 Main Menu จากนั้นกดปุ่มหมายเลข 4 Concurrent List เพื่อดูค่า Upload และ Download ที่ได้ตั้งค่าไว้ และหากต้องการลบค่า Speed ทั้งหมดที่ได้ป้อนไว้ ให้ทำการกดปุ่มหมายเลข 3 Clear All Shaper ของหน้า Policy Based Bandwidth Shaping ดังรูปที่ 4.11 ,4.12 และ 4.13



รูปที่ 4.11 การกดปุ่มหมายเลข 4 ปรากฏหน้า Concurrent List ของ PPP0



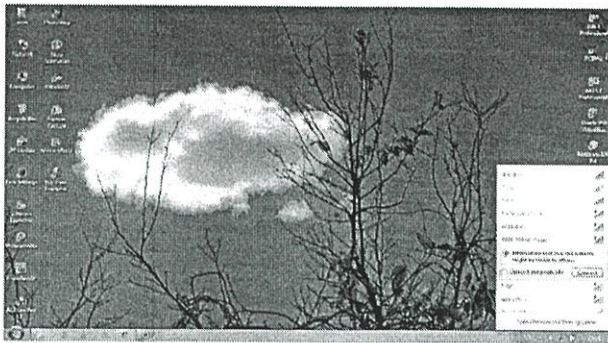
รูปที่ 4.12 หน้า Concurrent List ของ PPP1



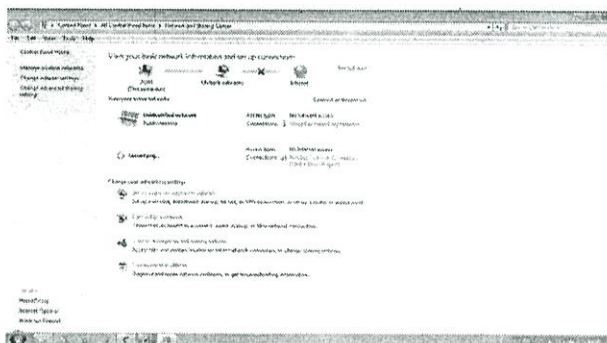
รูปที่ 4.13 ปรากฏหน้าจอ Clear All Shaper หลังกดปุ่มหมายเลข 3

4.2 ผลการทดลองการทำงานในฝั่งผู้ใช้งาน

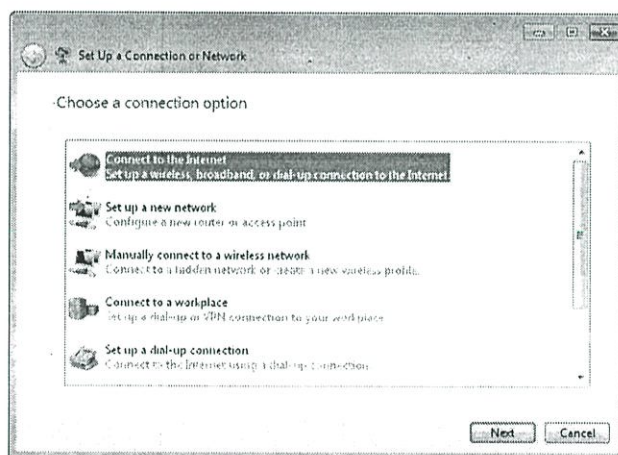
ให้ผู้ใช้ทำการถือคอร์ดเข้ามา โดยในที่นี้จะใช้ Access Point ในการปล่อยสัญญาณ Wifi ให้กับผู้ใช้ โดยคอมพิวเตอร์ของผู้ใช้จะจับสัญญาณชื่อ T304-PPPoE-Project จากนั้นทำการเชื่อมต่อ ดังรูปที่ 4.14 จากนั้นทำการตั้งค่าที่ Network and Sharing Center โดยทำการกด Set up a new Connection or Network แล้วทำการเลือก Connect to the Internet ดังรูปที่ 4.15 และ 4.16 จากนั้นทำการเลือก Set up a new connection anyway และทำการเลือก Broadband (PPPoE) ดังรูปที่ 4.17 และ 4.18 จากนั้นทำการใส่ User name และ Password ซึ่งแอดมินจะให้ใช้ User name เป็น user1,user2,user3,...,userN และ Password เป็น password1,password2,password3,...,passwordN และใส่ Connection name เป็น MyNET เมื่อใส่ข้อมูลครบแล้วก็เชื่อมต่อได้ทันทีดังรูปที่ 4.19 โดยผู้ใช้คนแรกทำการดาวน์โหลดข้อมูล โดยจะยังไม่มีการบีบแบนด์วิดธ์ อ่านค่าได้ 2.874 MB/วินาที หรือ 22.992 Mbps/วินาที หากต้องการบีบแบนด์วิดธ์ ให้กดปุ่มชี้ขึ้น [Up] Act และไม่ต้องการบีบให้กดปุ่มชี้ลง [Down] InAct ดังรูปที่ 4.20 และ 4.21 หลังจากนั้นทำการบีบแบนด์วิดธ์ แสดงดังรูปที่ 4.22 และ 4.23 จากนั้นให้ผู้ใช้คนที่สองทำการดาวน์โหลดข้อมูล โดยจะยังไม่มีการบีบแบนด์วิดธ์ อ่านค่าได้ 2020 KB/วินาที หรือ 16.16 Mbps แสดงดังรูปที่ 4.24 และ 4.25 และเมื่อทำการบีบแบนด์วิดธ์แล้ว อ่านค่าได้ 206 KB/วินาที หรือ 1.648 Mbps/วินาที แสดงดังรูปที่ 4.26 และ 4.27



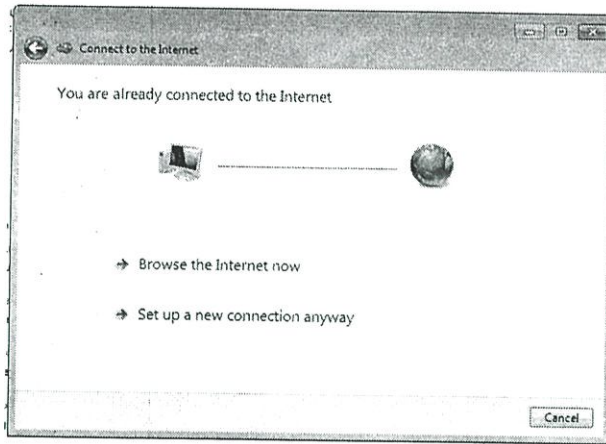
รูปที่ 4.14 การเชื่อมต่อ Wi-Fi T304-PPPoE-Project



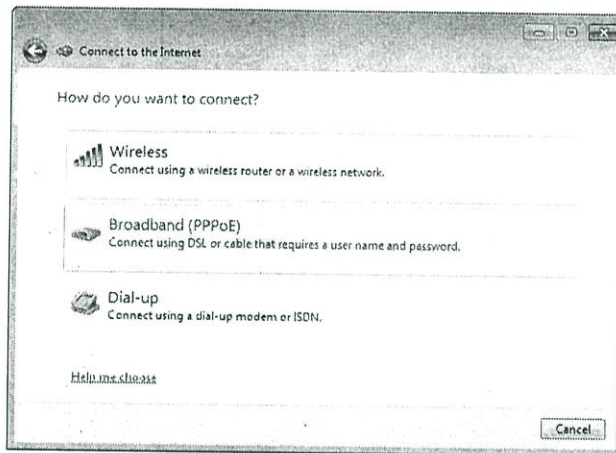
รูปที่ 4.15 หน้า Network and Sharing Center



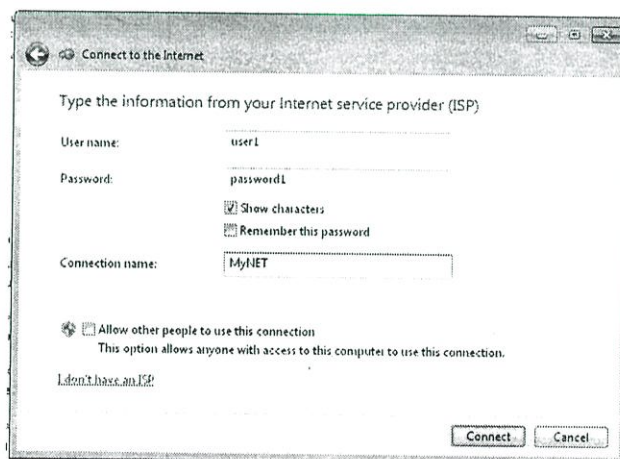
รูปที่ 4.16 หน้า Set up a new Connection or Network



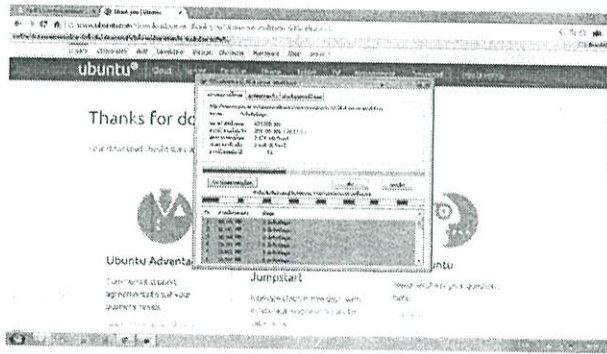
รูปที่ 4.17 หน้า Connect to the Internet



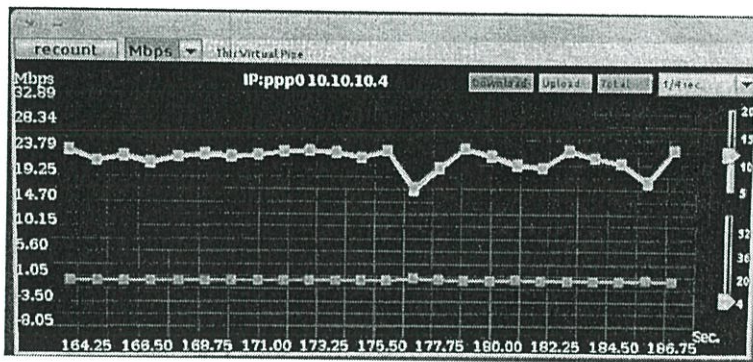
รูปที่ 4.18 การเลือก Broadband (PPPoE)



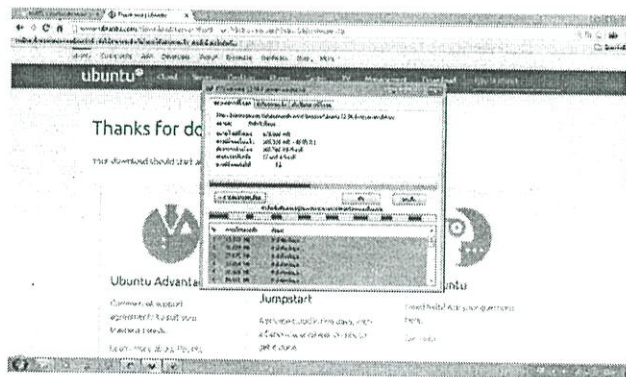
รูปที่ 4.19 หน้าต่างสำหรับใส่ User name และ Password



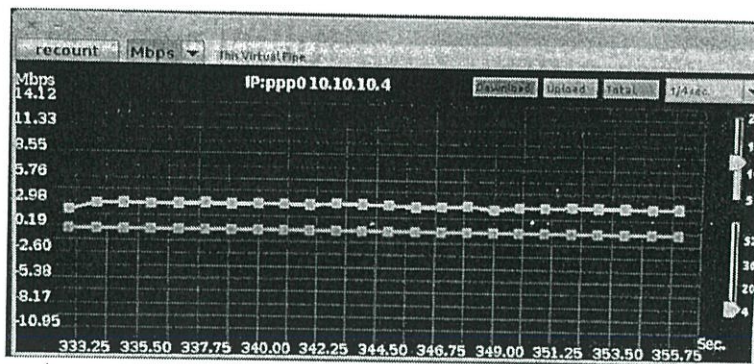
รูปที่ 4.20 ข้อมูลการดาวน์โหลดด้านผู้ใช้ โดยยังไม่ได้ทำการบีบแบนด์วิดธ์



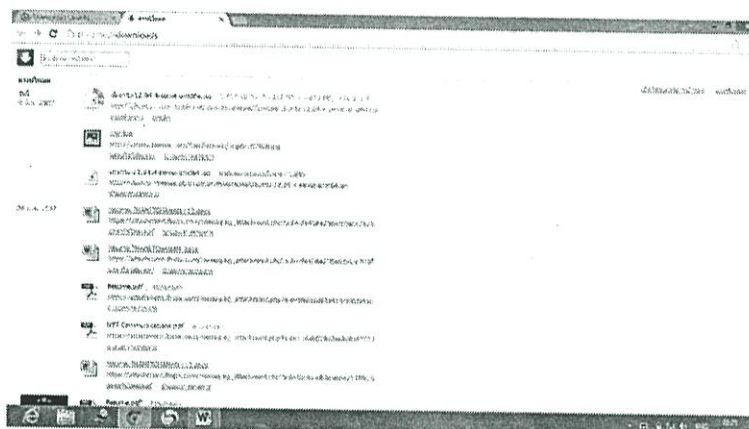
รูปที่ 4.21 อัตราเร็วในการดาวน์โหลดข้อมูลทางด้านแอดมิน โดยที่ยังไม่ได้ทำการบีบแบนด์วิดธ์



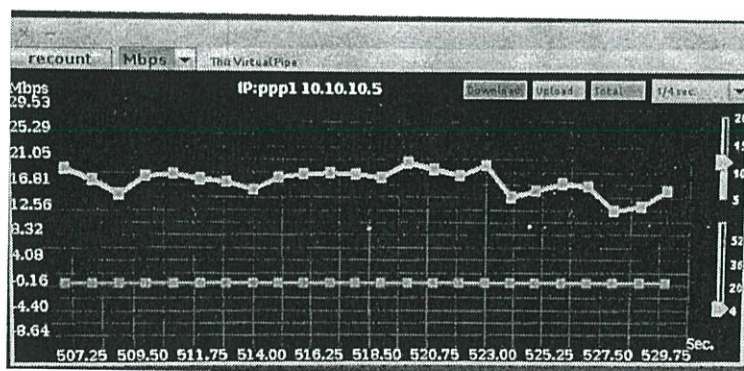
รูปที่ 4.22 ข้อมูลการดาวน์โหลดข้อมูลทางด้านผู้ใช้ หลังจากทำการบีบแบนด์วิดธ์แล้ว



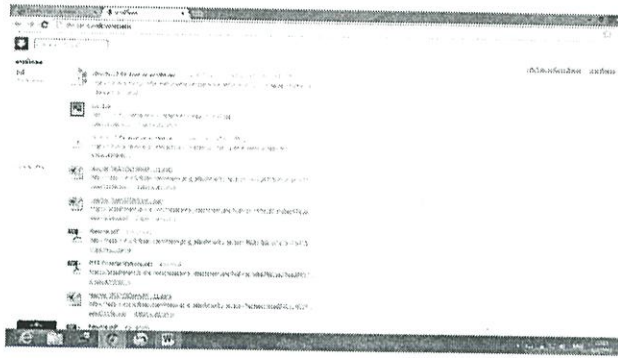
รูปที่ 4.23 อัตราเร็วในการดาวน์โหลดข้อมูลทางด้านแอดมิน หลังจากทำการบีบแบนด์วิดธ์



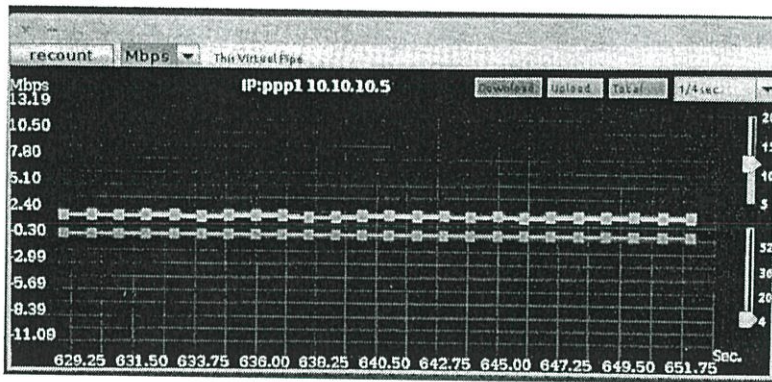
รูปที่ 4.24 ข้อมูลการดาวน์โหลดด้านผู้ใช้นั้นที่ 2 โดยยังไม่ได้ทำการบีบแบนด์วิดธ์



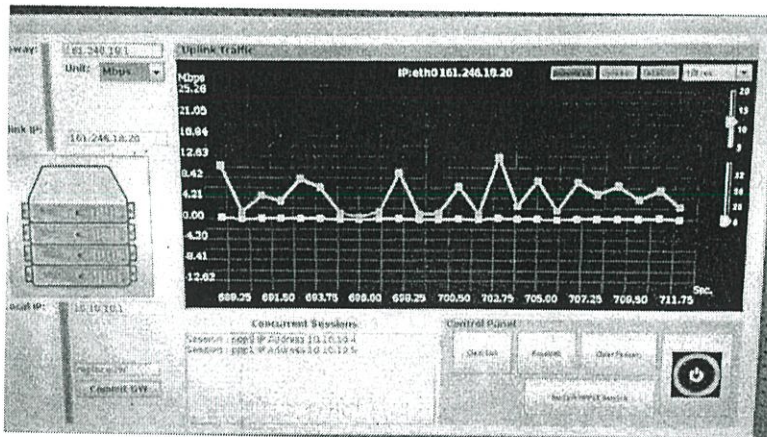
รูปที่ 4.25 อัตราเร็วในการดาวน์โหลดทางด้านแอดมิน โดยยังไม่ได้ทำการบีบแบนด์วิดธ์



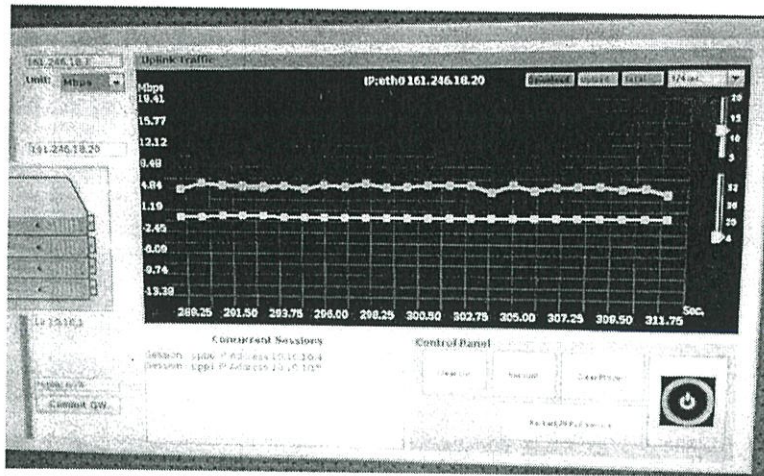
รูปที่ 4.26 ข้อมูลการดาวน์โหลดด้านผู้ใช้งานที่ 2 หลังจากทำการบีบแบนด์วิดธ์แล้ว



รูปที่ 4.27 อัตราเร็วในการดาวน์โหลดทางด้านแอดมิน หลังจากทำการบีบแบนด์วิดธ์แล้ว

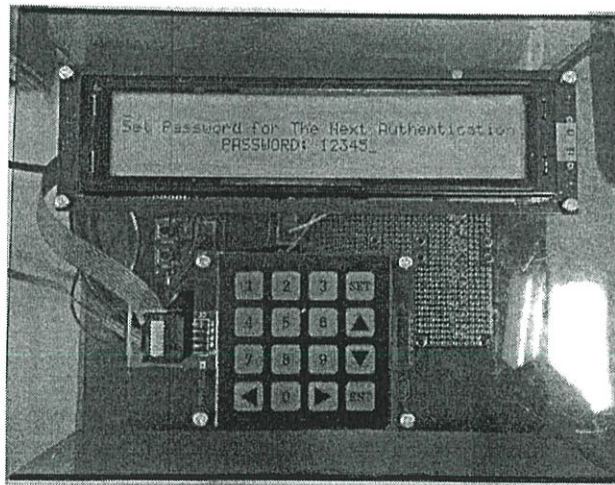


รูปที่ 4.28 กราฟของทั้งระบบสำหรับแสดงข้อมูลการดาวน์โหลดทางด้านแอดมินโดยยังไม่ได้ทำการบีบแบนด์วิดธ์



รูปที่ 4.29 กราฟของทั้งระบบสำหรับแสดงข้อมูลการดาวน์โหลดทางด้านแอดมิต หลังจากทำการบีบแบนด์วิดท์แล้ว

หากต้องการเปลี่ยน Password ของแอดมิตให้ทำการกดปุ่มหมายเลข 4 Set Password ของหน้า Policy Based Bandwidth Shaping แล้วทำการตั้งค่าน์ใหม่ ดังรูปที่ 4.30

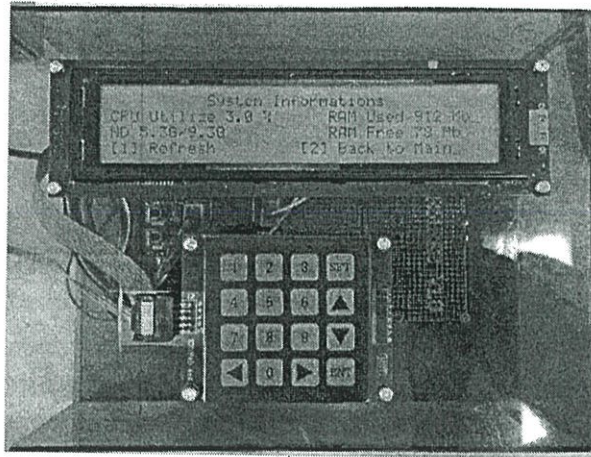


รูปที่ 4.30 การตั้งค่าน์ใหม่

หากต้องการดูข้อมูลของระบบให้ทำการกดปุ่มหมายเลข 1 Show System Info ของหน้า Main Menu จะปรากฏดังรูปที่ 4.31 ซึ่งมีรายละเอียดดังนี้

- 1) CPU Utilize จะบอกถึงการใช้ประสิทธิภาพของซีพียู (CPU) โดยจะบอกเป็นเปอร์เซ็นต์
- 2) HD จะบอกถึงจำนวนฮาร์ดดิสก์ที่ใช้งานต่อจำนวนฮาร์ดดิสก์ที่เหลือให้ใช้งาน

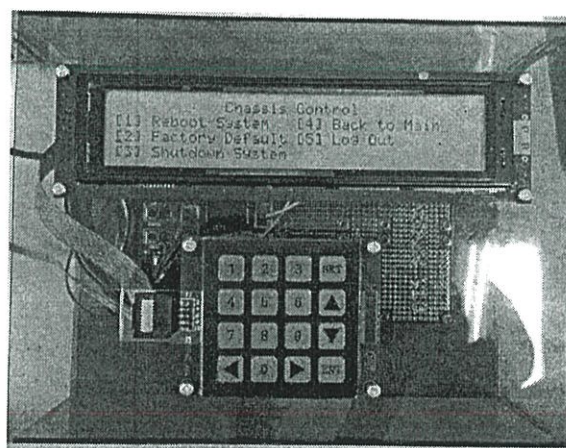
3) Ram Used และ Ram free จะบอกถึงจำนวนแรมที่ใช้งานและจำนวนแรมที่เหลือให้ใช้งาน



รูปที่ 4.31 ข้อมูลของระบบ

และทำการปิดเครื่องให้ทำการกดปุ่มหมายเลข 3 Service Control ของหน้า Main Menu จะปรากฏดังรูปที่ 4.32 และทำการกดปุ่มหมายเลข 3 เพื่อทำการปิดเครื่อง ซึ่งหน้าจอแอลซีดีมีรายละเอียดดังนี้

- 1) Reboot system ใช้สำหรับรีสตาร์ทเครื่องเซิร์ฟเวอร์
- 2) Factory default ใช้สำหรับเคลียร์พารามิเตอร์ทั้งหมดทิ้ง
- 3) Shutdown system ใช้สำหรับปิดเครื่องเซิร์ฟเวอร์
- 4) Back to Main Menu ใช้สำหรับกลับสู่หน้าหลัก
- 5) Log Out ใช้สำหรับล็อกเข้าที่ออกจากระบบ



รูปที่ 4.32 หน้า Chassis Control

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์เรื่องการควบคุมแบนด์วิดท์สำหรับหลายผู้ใช้บริการ ใช้หลักการของโปรโตคอล พีพีพีวีงผ่านไปในระบบอีเทอร์เน็ต หรือเรียกว่า PPPoE โดยเป็นการทำงานบนระบบปฏิบัติการลินุกซ์ แอปพลิเคชันนี้จะถูกพัฒนาด้วยภาษาจาวา วาร่วมกับโปรเซสบีวเดอร์ไลบรารีและเซลล์สคริปต์ โดยใช้โมดูลแอลซีดีเป็นหน้าพาแนลสำหรับแสดงผลข้อมูลและป้อนคำสั่งผ่านคีย์แพดได้โดยตรง โดยส่งข้อมูลผ่านอาร์เอส 232 แบ่งการทำงานเป็น 2 ส่วน คือ ส่วนของ PPPoE Server และส่วนของไมโครคอนโทรลเลอร์ ซึ่งจากการทดลองพบว่าสามารถแบ่งแบนด์วิดท์ได้ตามที่ต้องการ โดยป้อนคำสั่งผ่านคีย์แพดได้โดยตรงและแสดงผลบนหน้าจอโมดูลแอลซีดีได้

5.2 ข้อเสนอแนะ

แอปพลิเคชันสำหรับควบคุมการแบ่งช่องทางในการสื่อสารและควบคุมปริมาณการใช้งาน ภายในเครือข่ายมีข้อจำกัดในด้านการให้บริการสำหรับเครื่องลูกข่ายที่ จำกัดจำนวน ดังนั้นแอปพลิเคชันนี้สามารถนำไปพัฒนาต่อยอดในการให้บริการสำหรับเครื่องลูกข่ายที่มีจำนวนมากขึ้นได้ และสามารถให้บริการได้อย่างมีประสิทธิภาพได้

บรรณานุกรม

- [1] เอกชัย ศรีปฐมภรณ์. Bandwidth Management บริหารโครงข่ายอย่างมีประสิทธิภาพด้วยตนเอง. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด, 2551
- [2] วีระศักดิ์ ชิงถาวร .JAVA PROGRAMMING Volume 1 . กรุงเทพมหานคร : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
- [3] วีระศักดิ์ ชิงถาวร .JAVA PROGRAMMING Volume 2 . กรุงเทพมหานคร : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
- [4] วีระศักดิ์ ชิงถาวร .JAVA PROGRAMMING Volume 3 . กรุงเทพมหานคร : บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
- [5] พนา ดุสิตากร. “ไมโครคอนโทรลเลอร์ ตระกูล MCS-51.” http://eerg.eng.rmutp.ac.th/E_Learning/Microprocessor/chapter/1.html.
- [6] ยงยุทธ ชมไชย. “การสื่อสารข้อมูล.” <https://sites.google.com/site/kruyutsw.html>.
- [7] Soowoi Detsuka. “Shell Script.” <http://soowoi.wordpress.com/2011/08/14.html>.

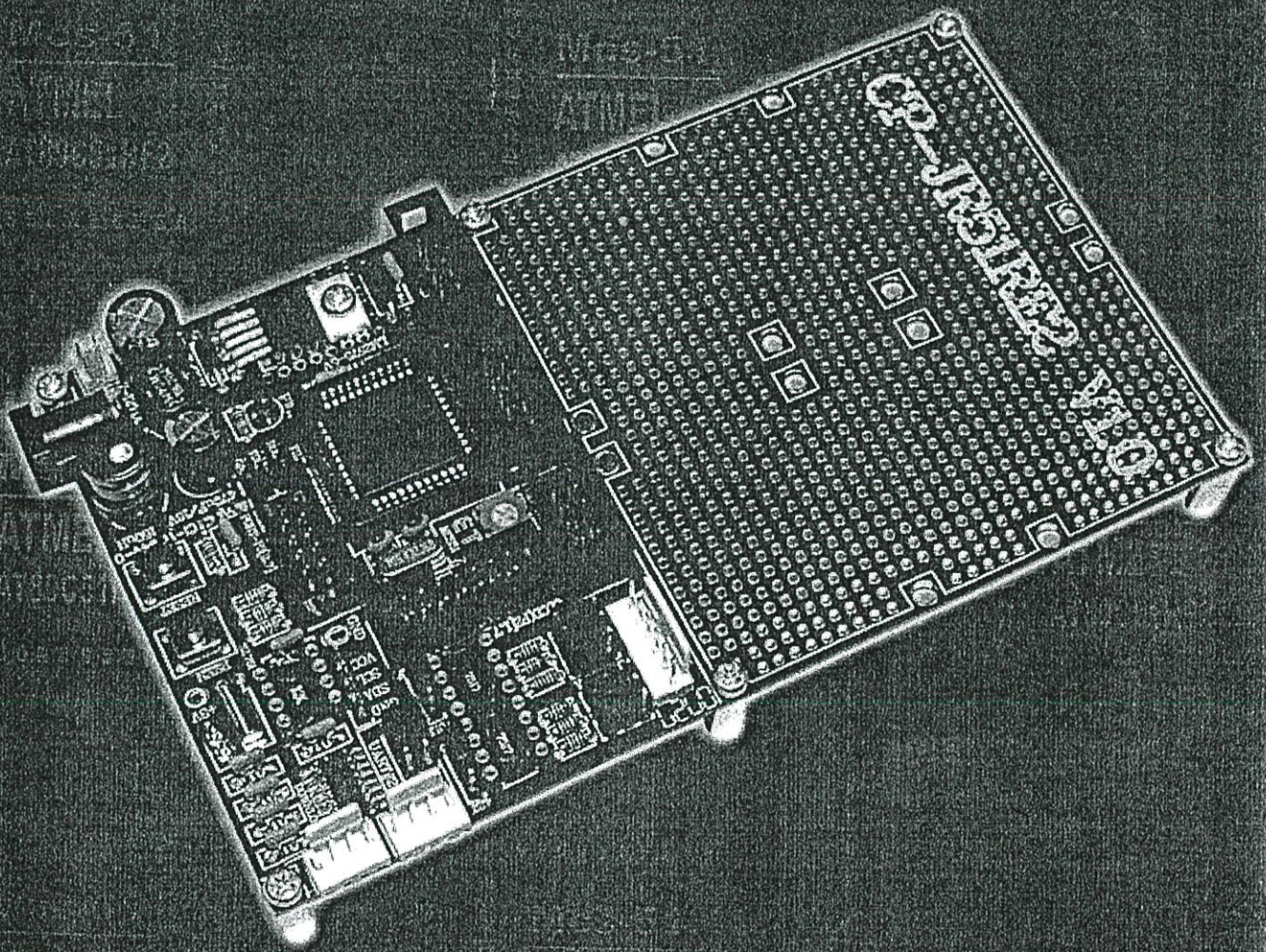
ภาคผนวก

Mcs51

คู่มือการใช้งาน

User's manual

CP-JR51RE2 V1.0



ETT
www.ett.co.th

บริษัท อีทีที จำกัด ETT CO., LTD.

1112/96-98 ถนนสุขุมวิท แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10110 <http://www.etteam.com>

1112/96-98 Sukhumvit Rd., Phraakanong Klongtoey Bangkok 10110 <http://www.ett.co.th>

Tel : 02-7121120 Fax : 02-3917216

email : sale@etteam.com

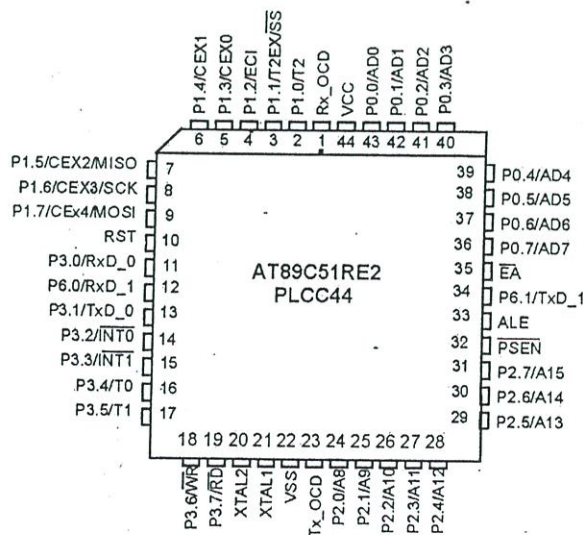
CP - JR51RE2 V1.0

บอร์ดไมโครคอนโทรลเลอร์รุ่นนี้จะใช้ MCU ขนาด 8 บิต ของ Atmel เบอร์ # AT89C51RE2 ซึ่ง MCU ตัวนี้จะบรรจุในตัวถังแบบ PLCC ขนาด 44 ขา จุดเด่นของ MCU เบอร์นี้คือ มี UART ให้ใช้งาน 2 แชนแนล , มี Timer/Counter ขนาด 16 บิต , มีพื้นที่สำหรับ Flash โปรแกรมถึง 128 Kbyte และมีขนาด RAM มากถึง 8 Kbyte ให้ใช้งาน การจัดสรร Port I/O ของบอร์ดที่ได้ต่อออกมาไว้ให้ผู้ใช้งานได้ใช้งานมีดังนี้ มี Port I/O = 4 Port , Port RS232 = 2 Port , Port RS422/485 = 1 Port , Port LCD แบบ 4 bit 1 Port และวงจรสำหรับในส่วนของ RTC ที่ใช้กับ # DS1307

ในส่วนของการ Download โปรแกรมลงบอร์ดนั้นจะ Download ผ่านทาง Port RS232 โดยใช้โปรแกรม Flip V3.1.0 เป็นตัว Download และใช้คอมไพเลอร์ Keil μ Vision3 เป็นตัวพัฒนาโปรแกรมด้วยภาษา C

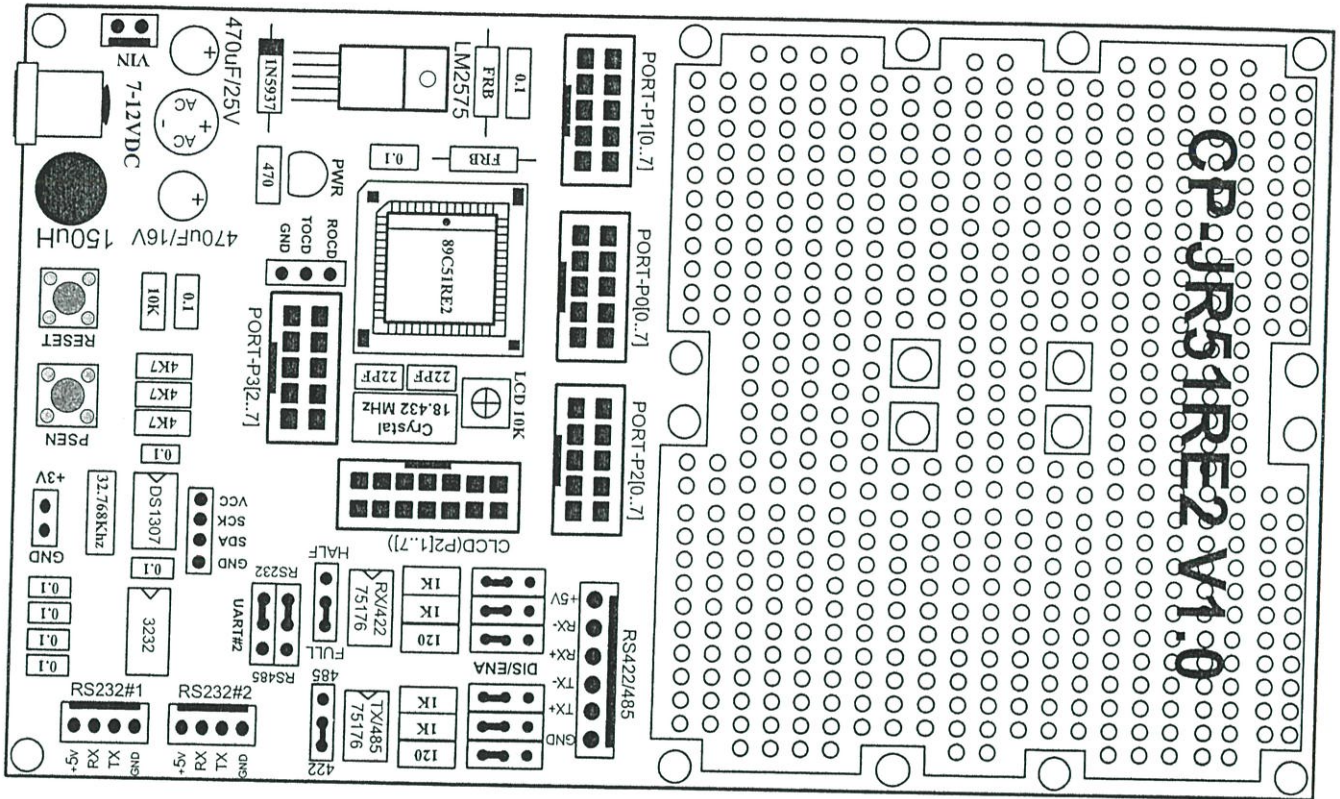
1. คุณสมบัติของบอร์ด CP-JR51RE2 และ MCU

- MCU เป็นตัวถังแบบ PLCC 44 Pin
- MCU ทำงานที่แรงดัน 2.7 - 5.5 V
- ความถี่ Crystal ที่ใช้งานบนบอร์ด 18.432 MHz
- หน่วยความจำ : Flash 128 KB , RAM 8KB
- การสื่อสารอนุกรมประกอบด้วย SPI 1 แชนแนล และ Uart 2 แชนแนล
- 16 บิต Timer/Counter สำหรับ Timer_0 , Timer_1 และ Timer_2
- Watch-Dog Timer 14 bit Counter
- PORT I/O 34 PIN(P0-P3 , P6.0 , P6.1)
- 11 Interrupt Source ซึ่งกำหนดระดับความสำคัญของ Interrupt ได้ 4 ระดับ
- Download โปรแกรมด้วย Flip V3.1.0 ผ่านทาง RS232
- MCU ทำงานที่อุณหภูมิ - 40 ถึง +85 องศาเซลเซียส



รูปที่ 1.1 แสดงโครงสร้าง MCU AT89C51RE2

2. โครงสร้าง และการจัดสรร I/O ของบอร์ด CP-JR51RE2



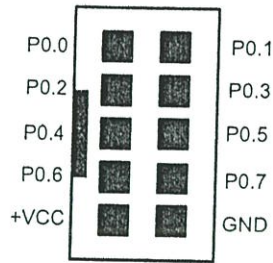
รูป 2.1 แสดงลักษณะโครงสร้างของบอร์ด CP-JR51RE2 V1.0

2.1) แหล่งจ่ายไฟ : สำหรับแหล่งจ่ายไฟของบอร์ดนี้สามารถต่อใช้งานได้ทั้งไฟกระแสตรงและกระแสสลับ โดยป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดันประมาณ 9-12 V ให้กับบอร์ด ซึ่งสามารถเลือกต่อกับหัว Connector แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านหัว Connector สำหรับ Adapter จ่ายไฟก็ได้เช่นกัน โดยจะแสดงผลการทำงานของแหล่งจ่ายไฟให้ทราบด้วย LED “PWR”

2.2) สัญญาณนาฬิกา CLOCK : ความถี่ของสัญญาณนาฬิกาที่ป้อนให้กับ MCU #AT89C51RE2 นั้นตามปกติแล้วสามารถป้อนค่าความถี่ของ Crystal ได้ถึง 40 MHz ใน Standard Mode (12 Clock / 1 Machine Cycle) แต่ในกรณีที่ให้ MCU ทำงานใน X2 Mode จะสามารถใช้ค่าความถี่สูงสุดได้ที่ 20 MHz โดยสำหรับบอร์ด CP-JR51RE2 นั้นจะกำหนดให้ใช้ค่าความถี่ของ Crystal ที่ป้อนให้กับ MCU ด้วยค่าความถี่ 18.432 MHz เพื่อให้การสื่อสารพอร์ตอนุกรมสามารถหาร Baud Rate ได้ลงตัว

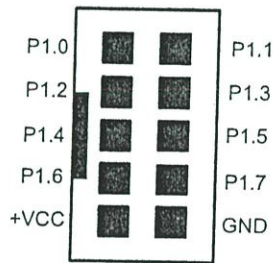
2.3) หัวต่อ I/O Port : สำหรับบอร์ดนี้จะต่อขาสัญญาณ I/O Port ของ MCU เข้ากับหัวต่อ Connector 10 Pin เพื่อให้ผู้ใช้งานไปต่อใช้งานได้สะดวกขึ้น โดยมีการจัดขาสัญญาณดังนี้

Port-P0[0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่หัวต่อ Connector ขนาด 10 Pin โดยหัวต่อนี้จะเชื่อมต่อกับสัญญาณมาจาก P0 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



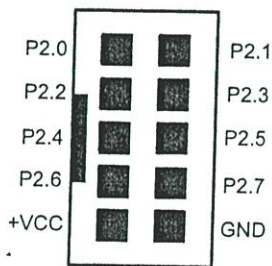
รูปที่ 2.2 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P0

Port-P1[0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 10 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก P1 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



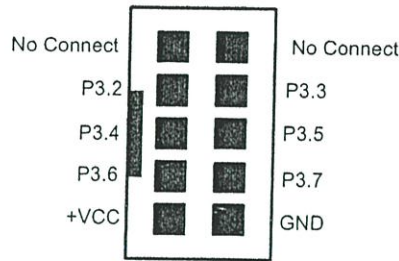
รูปที่ 2.3 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P1

Port-P2[0..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 10 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก P2 ของ MCU ทั้ง 8 เส้น ลักษณะการจัดขาสัญญาณแสดงดังรูป



รูปที่ 2.4 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P2

Port-P3[2..7] : ขาสัญญาณเหล่านี้ สามารถใช้งานเป็น Input หรือ Output ได้ โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 10 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก P3 ของ MCU อยู่ 6 เส้น ส่วนอีก 2 เส้นที่เหลือคือ P3.0 และ P3.1 นั้นจะถูกต่อไปยัง ขั้วต่อ RS232 #1 ลักษณะการจัดขาสัญญาณแสดงดังรูป



รูปที่ 2.5 แสดงการจัดเรียงขาขั้วต่อ 10 Pin ของ Port P3

หมายเหตุ เมื่อจะใช้งานขาสัญญาณใดๆหรือ Port ใดเป็น Input ผู้ใช้จะต้องทำการส่งค่า 0xFF (ทำขาสัญญาณที่จะใช้เป็น Input ให้เป็น 1) ออกไปให้ Pin หรือ Port นั้นก่อนแล้วถึงทำการอ่านข้อมูลเข้ามาได้ แต่ถ้าใช้เป็น Output สามารถส่งข้อมูลออกไปได้เลย

2.4) ขั้วต่อ RS232/RS422/RS485 : สำหรับบอร์ดนี้ได้จัดสรรขั้วต่อ สำหรับสื่อสารแบบอนุกรมไว้ 3 รูปแบบ โดยรายละเอียดและหน้าที่การใช้งานของแต่ละแบบนี้จะเป็นดังนี้

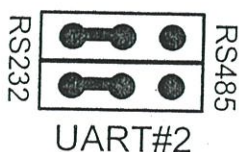
RS232#1 : ขั้วต่อนี้จะถูกต้องไว้ที่ขั้ว Connector 4 Pin โดยจะเชื่อมต่อสัญญาณมาจาก P3.0(RxD_0) และ P3.1(TxD_0) ซึ่งขั้วต่อนี้นอกจากจะใช้ในการ สื่อสารทาง RS232 แบบปกติแล้วยัง ใช้สำหรับ Download โปรแกรม ลงใน MCU ด้วย โดยไม่ต้องทำการ Set Jumper ใดๆเพียงแต่เวลาจะ Download โปรแกรม จะต้องทำการกดสวิทช์ PSEN และ RESET ดังต่อไปนี้เพื่อเข้าสู่ Monitor Mode

- กด SW. PSEN ค้างไว้
- ตามด้วยการกด SW. RESET ค้างไว้
- ปลด SW. RESET ในขณะที่ SW. PSEN ยังถูกกดค้างอยู่
- ปลด SW. PSEN เป็นลำดับสุดท้าย

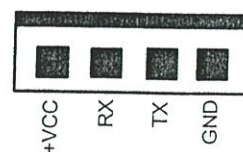
หลังจาก Download เรียบร้อยให้ทำการกด SW. RESET เพื่อให้ MCU เริ่ม Run โปรแกรมที่เขียน ซึ่ง Port RS232#1 ก็จะเข้าสู่การทำงานแบบปกติ คือ การสื่อสารของ Port ก็จะถูกควบคุมตามโปรแกรมที่ผู้ใช้เขียน

RS232#2 : สำหรับขั้วต่อนี้ จะใช้สำหรับสื่อสารข้อมูลทาง RS232 เช่นกัน ซึ่งจะแยกอิสระกับขั้วต่อ RS232#1 โดยจะถูกต้องไว้ที่ขั้ว Connector 4 Pin ซึ่งขั้วต่อนี้จะเชื่อมต่อสัญญาณผ่าน Jumper UART#2 ไปยังขาสัญญาณ P6.0 (RxD_1) และ P6.1(TxD_1)

เนื่องจาก ขาสัญญาณ P6.0 และ P6.1 นี้จะถูกนำไปใช้งานสำหรับขั้วต่อ RS422/485 ด้วย ดังนั้นเวลาจะใช้งาน ขั้วต่อ RS232#2 จะต้อง Set Jumper UART#2 มาทางด้าน RS232 ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการ Set Jumper UART#2 เมื่อใช้งานขั้วต่อ RS232#2

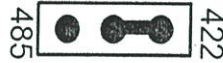


รูปที่ 2.7 แสดงการจัดเรียงขาขั้วต่อ RS232#1 และ RS232#2

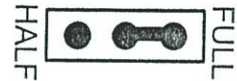
RS422/485 : ขั้วต่อนี้จะใช้สื่อสารข้อมูลแบบ RS422 หรือ RS485 อย่างใดอย่างหนึ่ง ซึ่งการใช้งานขั้วต่อนี้จะต้องทำการต่อ IC Line Driver 75176 จำนวน 2 ตัว สำหรับ RS 422 หรือ 1 ตัวสำหรับ RS485 ลงใน Socket เสียก่อน จากนั้นก็ทำการ Set Jumper UART#2 มาทางด้าน RS485 ดังรูปที่ 2.8 ต่อมาก็ทำการ Set Jumper 422/485 ไปทางด้านที่จะใช้งาน ซึ่งเมื่อเลือกมาทางด้าน 422 จะต้อง Set Jumper FULL/HALF ไปทางด้าน FULL เพื่อให้ IC Line Driver 2 ตัวในการรับข้อมูล(ICตัวใน) และส่งข้อมูล(ICตัวนอก) แต่ถ้าเลือกมาทางด้าน 485 ให้ Set Jumper FULL/HALF มาทางด้าน HALF เพื่อให้ IC Line Driver ตัวนอกตัวเดียวในการรับและส่งข้อมูล(แบบ 2 Line) โดยใช้ขา P3.7 เป็นตัวควบคุมการรับ-ส่งข้อมูล (P3.7 = 1 : Tx , P3.7 = 0 : Rx)



1) Set Jumper. UART#2

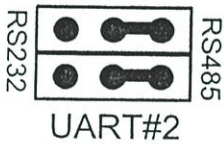


2) Set Jumper. 422/485



3) Set Jumper FULL/HALF

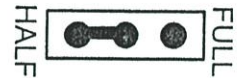
รูปที่ 2.8 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS422



1) Set Jumper. UART#2

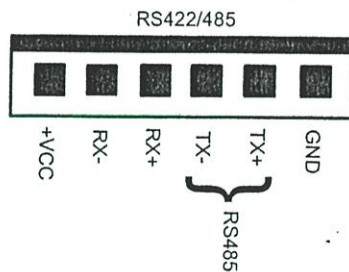


2) Set Jumper. 422/485



3) Set Jumper FULL/HALF

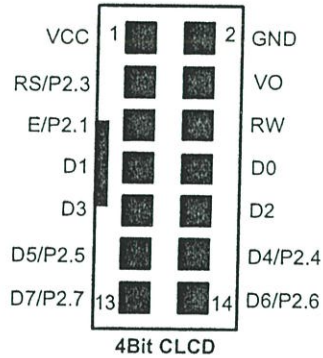
รูปที่ 2.9 แสดงการ Set Jumper เมื่อใช้งานขั้วต่อ RS485



รูปที่ 2.10 แสดงการจัดเรียงขาของขั้วต่อ RS422/485

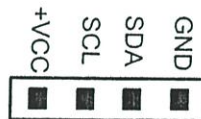
หมายเหตุ การใช้งาน ขั้วต่อ RS232#2 หรือ RS422/485 จะต้องเลือกใช้งานขั้วต่อใดขั้วต่อหนึ่งโดยการ Set Jumper UART#2 ไม่สามารถจะใช้งานขั้วต่อทั้งสองนี้พร้อมกันได้ เนื่องจาก ได้ใช้ขาสัญญาณ Rx_D_1(P6.0) และ Tx_D_1 (P6.1) ของ MCU ร่วมกันอยู่ และในกรณีเลือกใช้งาน RS422/485 จะต้องใส่ IC Line Driver #75176 ใน Socket ทั้ง 2 ตัวด้วย สำหรับ RS422 หรือ 1 ตัวทางริมด้านนอกสำหรับ RS485

2.5) **ขั้วต่อ CLCD** : ขั้วต่อนี้จะใช้สำหรับต่อ DOT Matrix LCD โดยถูกจัดไว้ที่ขั้วต่อ Connector ขนาด 14 Pin โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก Port P2 ของ MCU การจัดวงจรของ Port CLCD นี้จะต่อในลักษณะแบบ 4 บิต มี VR ต่อไว้สำหรับปรับความเข้มของ LCD ให้ด้วย เมื่อผู้ใช้จะใช้งานจะต้องต่อ PIN ของ Modul LCD ให้ตรงกับขาที่กำหนดไว้บน Port ด้วย โดยมีการจัดเรียงขาดังนี้



รูปที่ 2.11 แสดงการจัดเรียงขาของขั้วต่อ CLCD แบบ 4 บิต

2.6) **Socket DS1307** : สำหรับ Socket 1307 ที่จัดไว้บนบอร์ดนี้ จะเป็นการจัดวงจรไว้สำหรับรองรับการต่อใช้งาน RTC #DS1307 ซึ่งจะเป็นการสื่อสารแบบ I2C เมื่อผู้ใช้จะใช้งานจะต้องนำ IC RTC #DS1307 นี้มาเสียบที่ Socket จากนั้นก็ทำการต่อสาย SDA และ SCL จาก Connector 4 Pin ที่อยู่ข้างๆซึ่งได้เชื่อมต่อมาจาก ขา SDA และ SCL ของ DS1307 ไว้แล้วไปเข้าขา I/O ของ MCU ที่จะใช้ควบคุม ซึ่ง MCU เบอร์นี้ไม่มีขา I2C ให้ใช้โดยตรงดังนั้นจะต้องสร้าง ขา SDA และ SCL จาก I/O เาเอง



รูปที่ 2.12 แสดงการจัดเรียงขาของขั้วต่อ #DS1307 4 PIN

3. การใช้งานโปรแกรม Keil μ Vision3 กับ AT89C51RE2

สำหรับโปรแกรม Keil μ Vision3 นี้จะเป็นโปรแกรมที่ใช้สำหรับพัฒนาการเขียนโปรแกรมของผู้ใช้ด้วยภาษาซี ซึ่งโปรแกรมตัวนี้จะทำหน้าที่เป็น Compiler ในตัว และจะสร้าง Hex ไฟล์ให้กับผู้ใช้เพื่อนำไปใช้ Download ลงในตัว MCU โดยตัวที่ให้ไปกับบอร์ดนั้นจะเป็นตัว DEMO สามารถ Flash Code ได้ไม่เกิน 2 K

ก่อนที่จะเริ่มต้นใช้งาน Keil μ Vision3 เราจะมาทำความเข้าใจเกี่ยวกับ MCU เบอร์นี้เสียก่อน คือ สำหรับ AT89C51RE2 เบอร์นี้จะมี RAM 8 K และพื้นที่ Flash Memory ภายในสำหรับเก็บ Code 128 K แต่ลักษณะของพื้นที่ Flash นั้นไม่ได้เป็นพื้นที่ต่อเนื่อง แต่จะถูกแบ่งออกเป็น 4 Bank Bank ละ 32 K ได้แก่ Common Bank (0000h-7FFFh), Bank0 (8000h-FFFFh) , Bank1 (8000h-FFFFh) , Bank2(8000h-FFFFh) เมื่อผู้ใช้พัฒนาโปรแกรมด้วย Software Keil μ Vision3 ผู้ใช้จะต้องทำการกำหนดค่าต่างๆของ Project ให้ถูกต้องเพื่อให้สามารถใช้งานพื้นที่ Flash Memory ได้ครอบคลุม 128 K มิฉะนั้น พื้นที่ Flash จะถูกใช้งานได้เพียง 64 K เท่านั้น ถ้าโปรแกรมมีขนาดเกินกว่านี้ก็จะไม่สามารถ Flash ลงไปได้ถ้าไม่ทำการกำหนดคุณสมบัติของ Project File ให้ถูกต้อง

ดังนั้นใน Software Keil μ Vision3 นี้จะมีวิธีการกำหนดคุณสมบัติให้กับตัว Project File เพื่อให้สามารถใช้งานพื้นที่ Flash Memory ได้ถึง 128 K ตามคุณสมบัติของ MCU และสามารถที่จะเลือก Flash ลง Bank ใดก็ได้ ใน 4 Bank และในแต่ละ Bank นั้นสามารถเรียกใช้โปรแกรมย่อยที่เขียนอยู่ต่าง Bank กันได้โดยผู้ใช้ไม่ต้องเสียเวลาในการเขียนโปรแกรมสลับ Bank ไปมา ซึ่งอาจจะทำให้ผู้ใช้เขียนผิดพลาดได้เนื่องจากจะเห็นว่าโครงสร้างของ MCU ตั้งแต่ Bank0-2 นั้น จะอ้างตำแหน่งแอดเดรสเดียวกัน ดังนั้นใน Software Keil μ Vision3 จะช่วยจัดการในส่วนต่างๆเหล่านี้ให้ต่อไปเราจะมาพูดถึงการใช้งานในส่วนของโปรแกรม Keil μ Vision3 กันบ้าง ซึ่งเราจะขอแยกอธิบายการใช้งานออกเป็น 2 แบบ ดังนี้

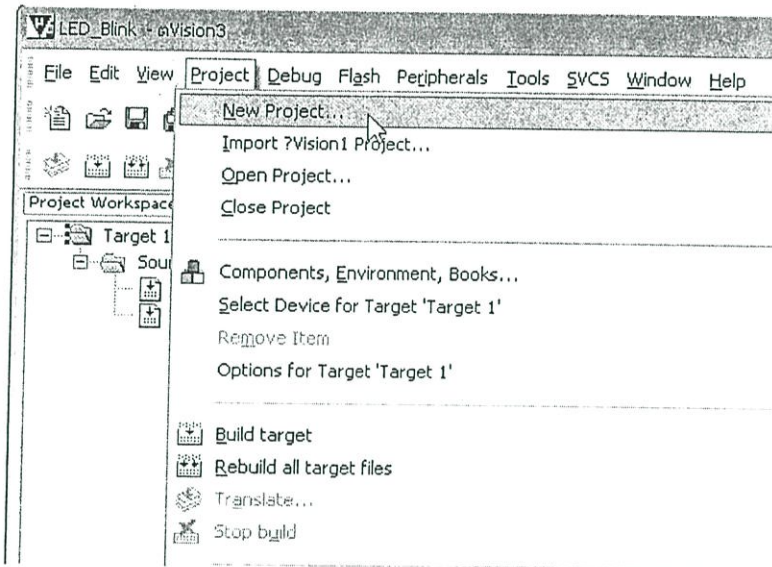
3.1 การใช้งาน Keil μ Vision3 กับ AT89C51RE2 แบบไม่ใช้การ SWITCH Bank

ซึ่งการใช้งานในส่วนนี้จะเป็นการใช้งานแบบปกติ การกำหนดคุณสมบัติให้กับ Project File ที่เขียนขึ้นก็จะไม่ยุ่งยาก แต่จะมีข้อจำกัดอยู่ว่า พื้นที่ที่ใช้สำหรับ Flash Code ของ MCU จะใช้ได้เพียง 64 K ไม่เต็ม 128 K โดยพื้นที่ Flash 64 K ที่ใช้ได้นี้ ก็จะเป็นพื้นที่ในส่วนของ Common Bank 32 K ส่วนอีก 32 K ตัว Keil จะเป็นตัวสุ่มเองว่าจะใช้ Bank ไหน แต่ขอแนะนำให้ผู้ใช้งานกำหนดพื้นที่ Bank ที่เหลือนี้เองเพื่อเวลาเขียนโปรแกรมจะได้ไม่มีปัญหาในภายหลัง โดยสามารถกำหนดได้คือ ในการเขียนโปรแกรมทุกครั้งเมื่อเลือกใช้งานแบบ ไม่ใช้การ Switch Bank ก็ให้ทำการกำหนดค่าใน Register BMSEL = 0x00 ในตอนต้นของโปรแกรมเสีย (ดูได้จากตัวอย่างใน CD) เพื่อกำหนดให้ใช้พื้นที่ Flash 64 K นี้ ใน Common Bank และ Bank0 เสมอ ต่อไปเราจะมาดูการใช้งาน Keil แบบไม่ใช้การ Switch Bank ตามขั้นตอนดังต่อไปนี้

- 1) ติดตั้งโปรแกรม Keil μ Vision3 ลงบนเครื่อง PC เมื่อติดตั้งเสร็จโดยปกติโปรแกรมจะถูกเก็บไว้ที่ C:\Keil
- 2) ให้ทำการ Copy File: at89c51re2.h ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\INC\Atmel เพื่อเอาไว้เรียกใช้ในส่วนของ Include File เมื่อเขียน โปรแกรมจะได้ไม่เสียเวลาประกาศตัวแปรต่างๆอีก
- 3) Copy File: STARTUP.A51 และ L51_BANK.A51 ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\LIB

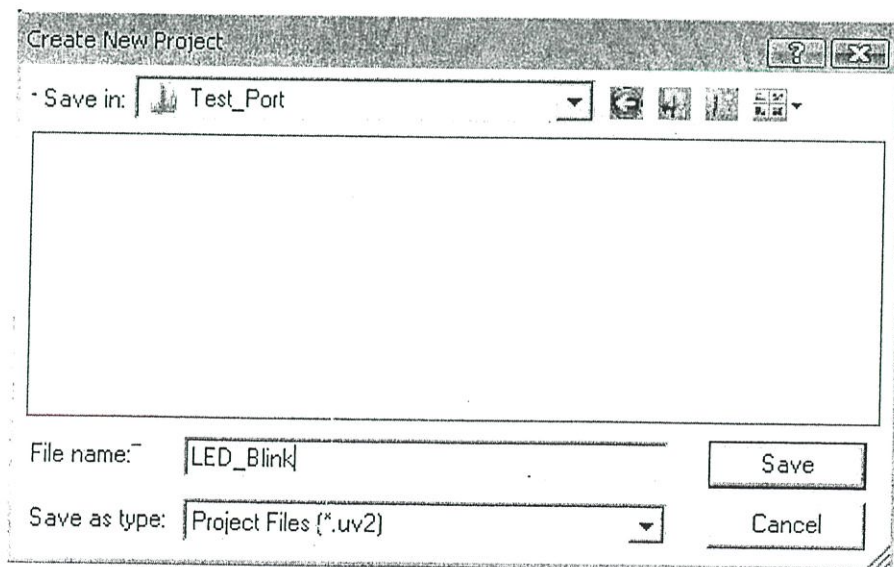
ซึ่งได้ทำการแก้ไขให้รองรับ MCU เบอร์นี้ไว้แล้ว แต่อาจจะไม่รองรับเบอร์อื่น ดังนั้นก็ควร Copy 2 ไฟล์นี้ ของเดิมที่มาพร้อมกับการติดตั้งเก็บไว้ด้วย เวลาจะเปลี่ยนไปใช้ MCU เบอร์อื่นจะได้นำมาวางทับได้เลย ไม่ต้องเสียเวลาติดตั้งโปรแกรมใหม่

- 4) เปิดโปรแกรม Keil μ Vision3 () ขึ้นมา เลือกที่เมนู Project และเลือก New Project... ดังรูปที่ 3.1.1



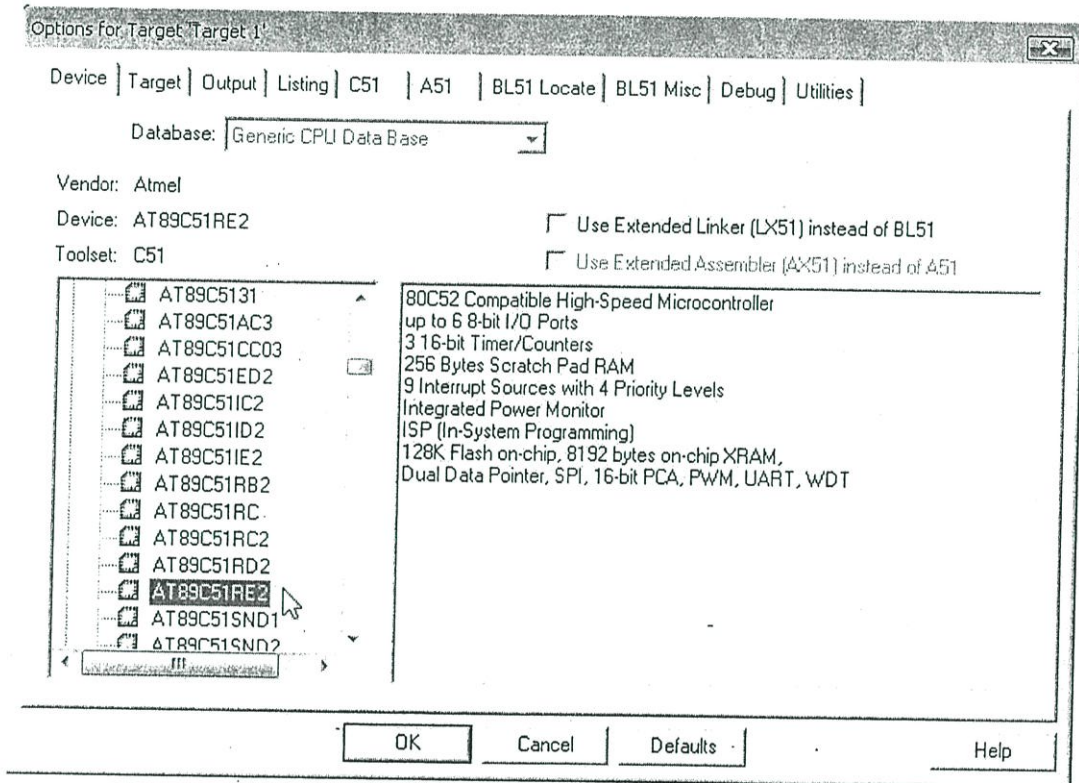
รูปที่ 3.1.1 แสดงการเลือก New Project

- 5) จะปรากฏ หน้าต่าง Create New Project ขึ้นมาดังรูปที่ 3.1.2 ให้เลือก Folder ที่จะเก็บ Project File และตั้งชื่อ Project File ในตัวอย่างจะตั้งชื่อ Project file คือ LED_Blink และเลือกเก็บที่ Folder Test Port จากนั้น กด Save

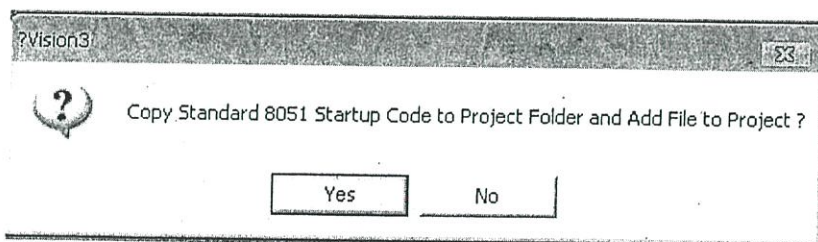


รูปที่ 3.1.2 แสดงหน้าต่าง Create New Project


- 6) หลังจากกด Save แล้ว จะปรากฏหน้าต่าง Select Device for Target ‘Target1’ ขึ้นมา ดังรูปที่ 3.1.3 (a) เพื่อให้ผู้ใช้เลือกเบอร์อุปกรณ์ โดยในช่อง Data base ให้ผู้ใช้เลือกที่ Atmel และเลือกที่เบอร์ AT89C51 RE2 ให้สังเกต เมื่อเลือกแล้วเบอร์จะแสดงในช่อง Device: ด้านบน จากนั้นคลิก OK จะมีหน้าต่าง Pop-Up ขึ้นมาดังรูปที่ 3.1.3 (b) ถ้ากด YES จะเป็นการนำ File Start Up (STARTUP.A51) มาตรฐานของ C51 เข้ามายัง Project หรือกด No ก็ได้ถ้าไม่ต้องการนำ File Start Up เข้ามายัง Project ในตัวอย่างนี้จะขอเลือก No

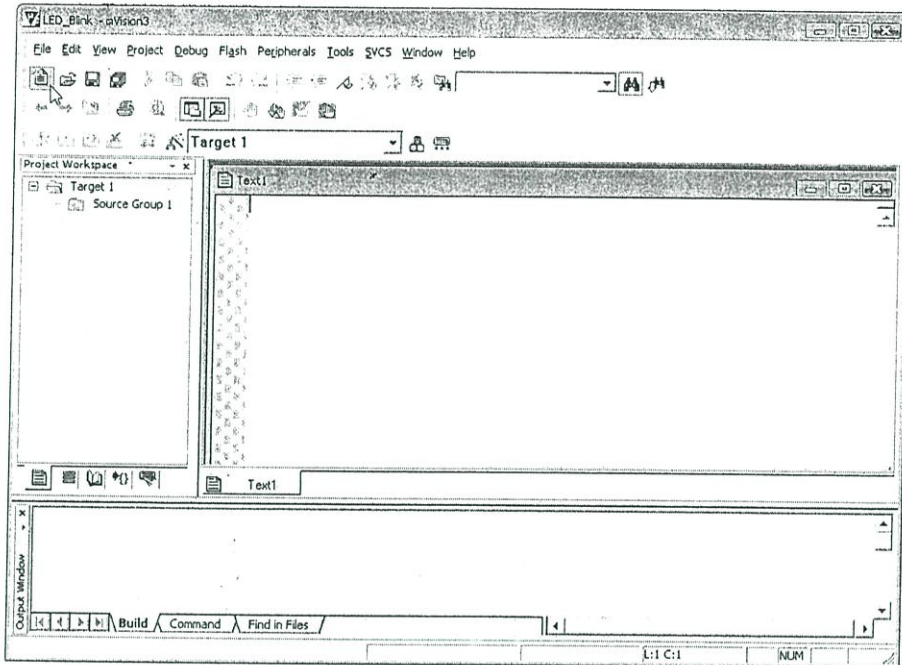


รูปที่ 3.1.3 (a) แสดงหน้าต่าง Select Device for Target ‘Target’



รูปที่ 3.1.3 (b)

- 7) จากนั้นให้คลิกที่ไอคอน Create a New File () จะได้นหน้าต่าง Text1 ออกมาดังรูปที่ 3.1.4 ซึ่งจะใช้สำหรับเขียนโปรแกรม



รูปที่ 3.1.4 แสดงหน้าต่าง Text1 สำหรับใช้เขียนโปรแกรม

- 8) ทำการเขียนโปรแกรมตามตัวอย่างด้านล่างลงในหน้าต่าง Text1 เมื่อเขียนโปรแกรมเรียบร้อยแล้วให้ไปที่เมนู File เลือก save as... (รูป ก.) จะได้นหน้าต่างดังรูป ข. ให้เลือก Save ไว้ใน Folder เดียวกับ Project File ในตอนแรก จากนั้นทำการตั้งชื่อ File เป็นนามสกุลจุด C ในตัวอย่างจะตั้งชื่อเป็น led_blink.c และกด Save

ตัวอย่างโปรแกรม led_blink.c

```

/*****
 * Example. LED Blink Port *
 * MCU : AT89C51RE2 *
 * Compiler : Keil C51 (V8.05 a) *
 * Use PORT : P0-P3 = Output Connect LED *
 *****/

#include <at89c51re2.h>
#include <stdio.h>

//----- Delay -----
void delay(int count)
{ int i, j;
  for(i=0 ; i<=count ; i++)
  for(j=0 ; j<=count ; j++) ;
}

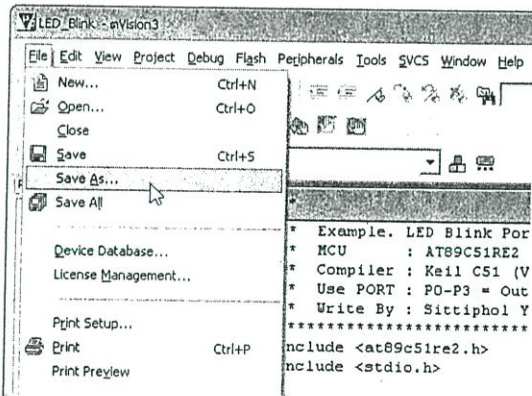
```

```

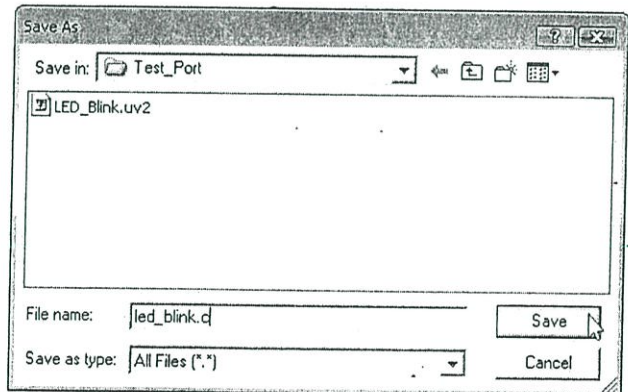
main()
{
    CKCON0 = 0xFE ; // Set MCU 12 Clock Mode

    BMSEL  = 0x00 ; // Select Bank 0+Command Bank สำหรับพื้นที่ Flash
//----- Test Out put port -----
while(1){
    P0 = 0x00 ; // Sent data 0 Out Port P0
    P1 = 0x00 ;
    P2 = 0x00 ;
    P3 = 0x00 ;
    delay(200) ;
    P0 = 0xFF ; // Sent data 1 Out Port P0
    P1 = 0xFF ;
    P2 = 0xFF ;
    P3 = 0xFF ;
    delay(200) ;
}
}

```



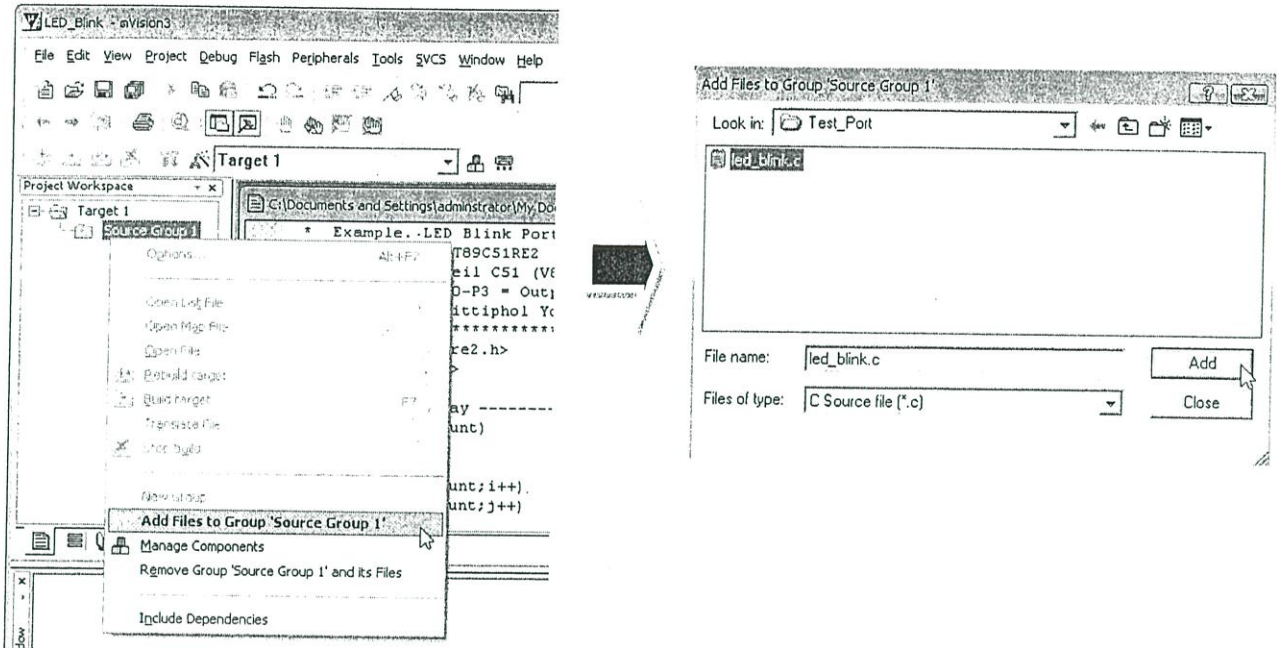
ก) เลือก Save As...



ข) ทำการตั้งชื่อ File.c และ Save File

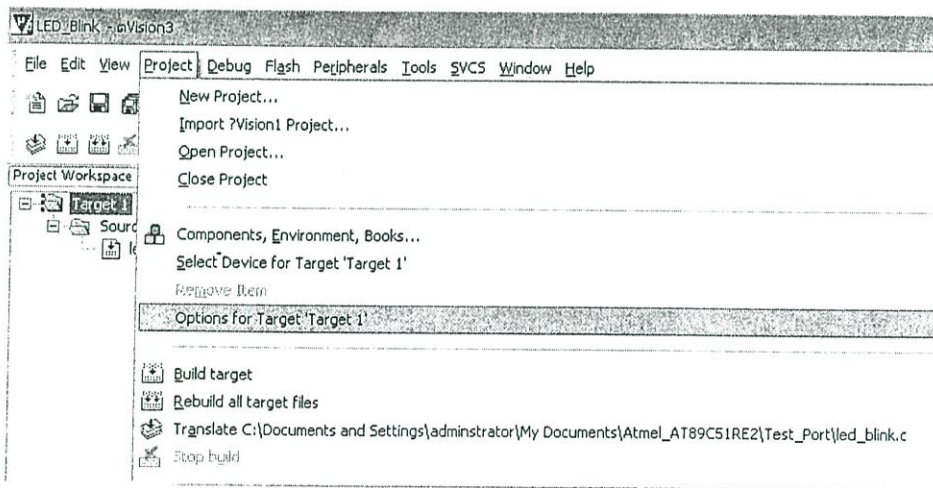
รูปที่ 3.1.5 แสดงหน้าต่างการ Save File . c

- 9) เมื่อ Save File แล้วให้ทำการ Add File led_blink.c เข้ามายัง Project โดยให้ Double Click หรือ คลิกขวาที่ Folder Source Group 1 ที่อยู่ในหน้าต่างด้านซ้ายมือ จากนั้นเลือก Add Files to Group 'Sourec Group 1' จะปรากฏหน้าต่างให้ Add file ขึ้นมา ให้ผู้ใช้เลือก File led_blink.c แล้วกด Add จากนั้นกด Close เพื่อปิดหน้าต่าง Add File ไฟล์ที่ Add ก็จะมาอยู่ใน Folder Source Group 1



รูปที่ 3.1.6 แสดงหน้าต่างการ Add ไฟล์

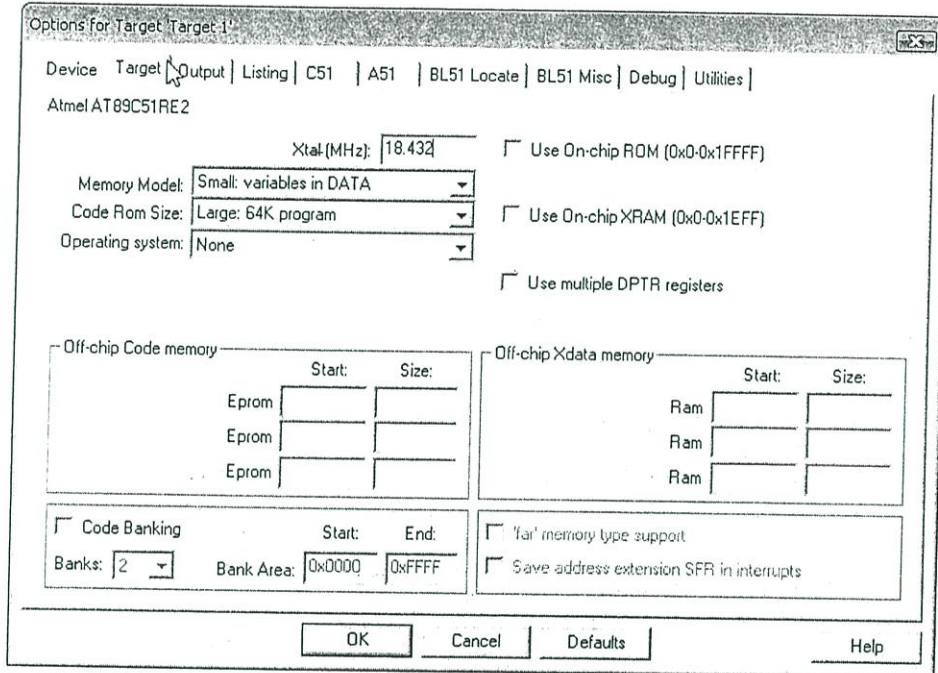
- 10) เมื่อแอดไฟล์ที่เขียนเข้ามาเรียบร้อยแล้วให้ไปคลิกที่ Folder Target 1 ในช่องด้านซ้ายมือให้เป็นแถบสีน้ำเงิน จากนั้นไปที่เมนู Project แล้วเลือก Options for Target 'Target 1' ดังแสดงในรูปที่ 3.1.7



รูปที่ 3.1.7 แสดงการเลือก Option เพื่อ Setup Project

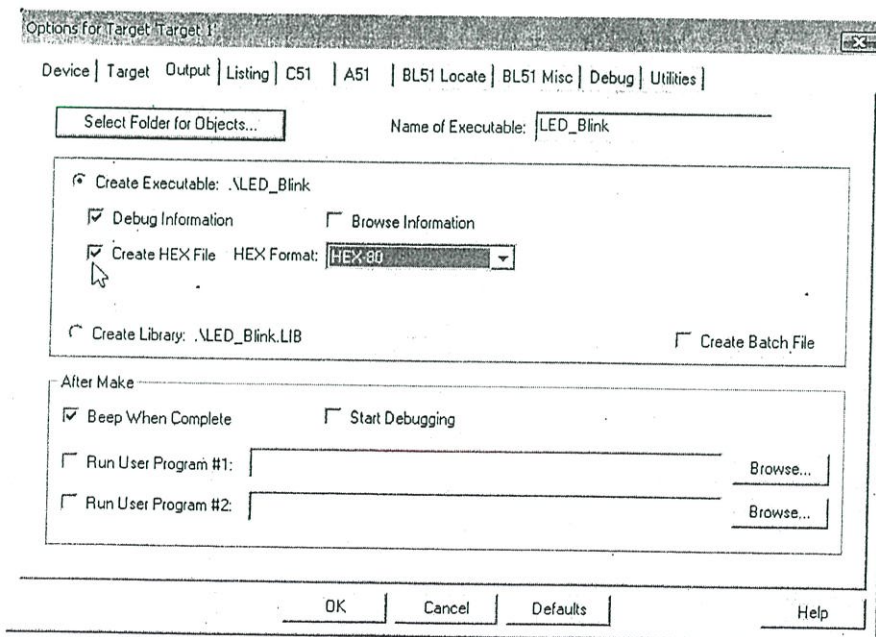
- 11) จะได้นหน้าต่าง Options for Target 'Target1' ออกมา ให้คลิกที่ TAB Target แล้วทำการกำหนดค่าตามรูปที่ 3.1.8 ให้สังเกตในช่อง Memory Model : ซึ่งในช่องนี้จะเป็นการเลือกขนาดของ Ram ที่ใช้ในการเขียนโปรแกรม ถ้าผู้ใช้เลือก Small: variables in DATA ตัวแปรที่ ประกาศใน โปรแกรม จะถูกเก็บไว้ยังพื้นที่ RAM ภายใน ซึ่งขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 128 byte , ถ้าเลือก Compact : variable in PDATA

ขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 256 Byte , และถ้าเลือก Large : variable in XDATA ตัวแปรก็จะถูกเก็บไว้ในพื้นที่ XRAM ดังนั้น ขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 8 K byte ซึ่งแล้วแต่ผู้ใช้งานจะเลือกใช้



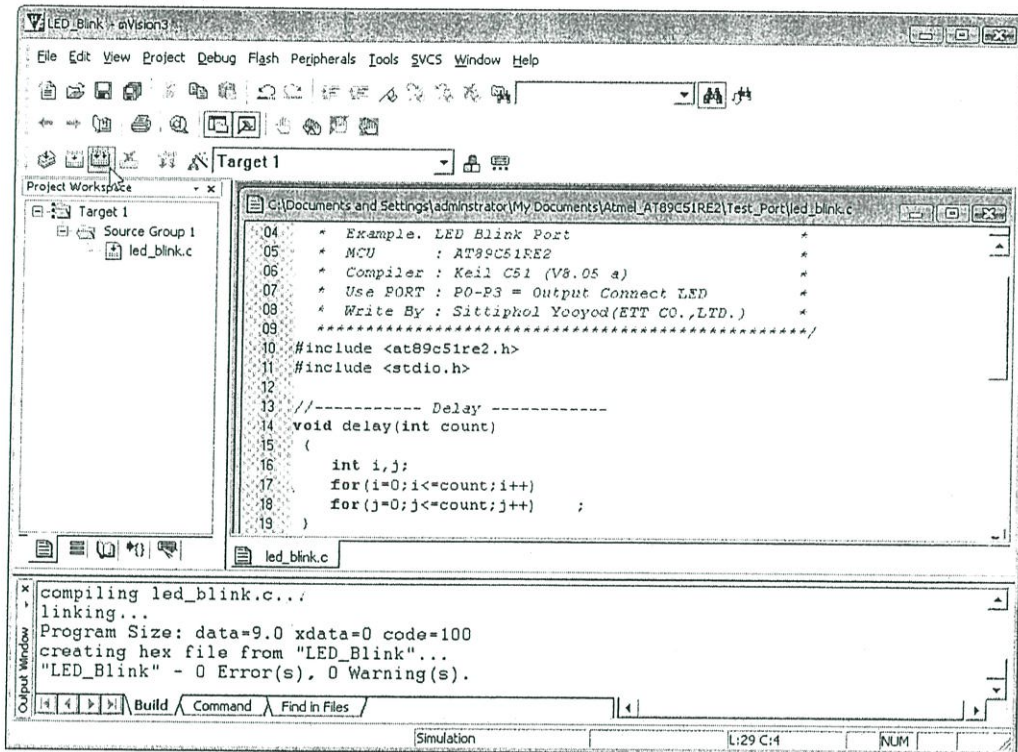
รูปที่ 3.1. 8 แสดงหน้าต่าง Option for Target ‘Target 1’ ที่ TAB Target

- 12) คลิกที่ TAB Output แล้วทำการ Tick เครื่องหมายถูกหน้าช่อง Create HEX File ส่วนช่อง HEX Format : ให้เลือก HEX-80 ส่วนช่องอื่นๆก็ให้กำหนดเหมือนในรูป เสร็จแล้วกด OK



รูปที่ 3.1.9 แสดงหน้าต่าง Option for Target ‘Target 1’ ที่ TAB Output

- 13) เมื่อ Set ค่าให้กับ Project เรียบร้อยแล้ว ให้ทำการ Compile โปรแกรมเพื่อตรวจสอบว่ามี Error หรือไม่ โดยคลิกที่ไอคอน Rebuild all target files (🔧) ถ้าโปรแกรมที่เขียนไม่มีปัญหา ก็จะแสดง Error เป็น 0 อยู่ในหน้าต่างด้านล่าง ดังแสดงในรูปที่ 3.1.10




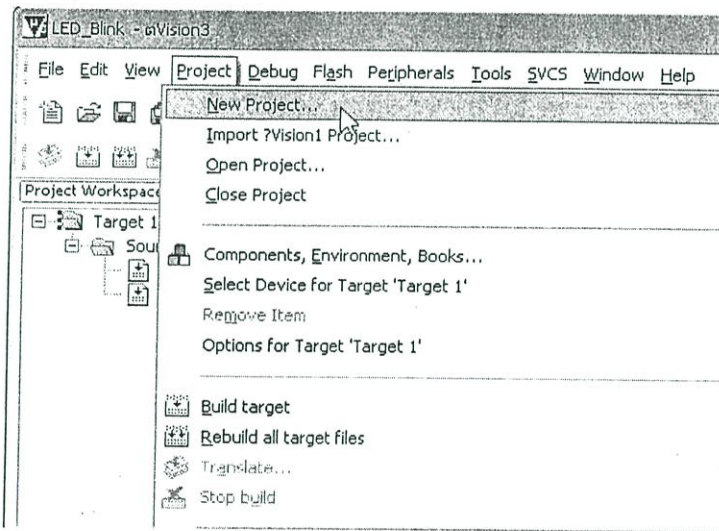
รูปที่ 3.1.10 แสดงหน้าต่างหลังจาก Compile ผ่านแล้ว

- 14) เมื่อ Compile เรียบร้อย ก็ให้ทำการ Download โปรแกรมลงใน MCU ได้ โดยใช้โปรแกรม Flip คู่มือการ Download ได้ในหัวข้อที่ 4 การ Download โปรแกรม ด้วย FLIP

3.2 การใช้งาน Keil uVision3 กับ AT89C51RE2 แบบใช้การ SWITCH Bank

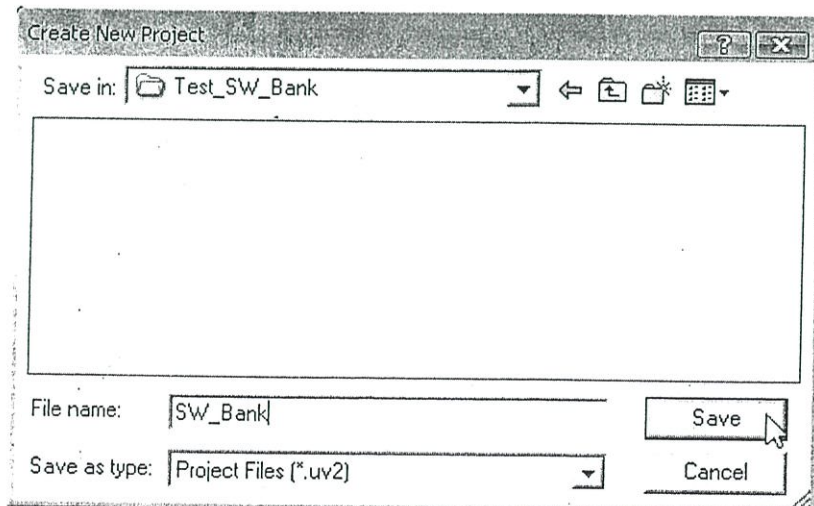
การใช้งานในส่วนนี้จะเป็นการใช้งานพื้นที่ Flash ของ MCU เบอร์นี้ได้เต็มพื้นที่ Flash 128 K แต่การกำหนดคุณสมบัติให้กับ Project File ที่เขียนขึ้นก็จะยุ่งยากขึ้นเล็กน้อย โดยวิธีนี้จะสามารถ Flash Code เข้าไปเก็บไว้ยังพื้นที่ Flash ได้ทั้ง 4 Bank ซึ่งแต่ละ Bank จะเก็บ Code ได้ไม่เกิน 32 K โดยในหนึ่ง Project ของโปรแกรมที่เขียน ผู้ใช้สามารถเขียนไฟล์ขึ้นมาได้มากกว่า 1 ไฟล์ และสามารถเลือกได้ว่าต้องการให้ไฟล์แต่ละไฟล์นั้นถูก Flash ไปเก็บไว้ที่ Bank ใดใน 4 Bank ก็ได้ ซึ่งโปรแกรมน้อยที่อยู่ต่าง Bank กันสามารถเรียกใช้งานสลับกันไปมาได้ ถึงแม้ว่าโครงสร้างแล้วตำแหน่งแอดเดรสของพื้นที่ Flash จะเป็นแอดเดรสที่ซ้ำกันอยู่ก็ตาม ตัวโปรแกรม Keil จะทำการจัดการในส่วนการ Switch Bank นี้ให้ ซึ่งผู้ใช้งานจะต้องทำการกำหนดคุณสมบัติของ Project ให้เป็นไปตามขั้นตอนที่จะกล่าวถึงดังต่อไปนี้

- 1) ติดตั้งโปรแกรม Keil μ Vision3 ลงบนเครื่อง PC เมื่อติดตั้งเสร็จโดยปกติโปรแกรมจะถูกเก็บไว้ที่ C:\Keil
- 2) ให้ทำการ Copy File: at89c51re2.h ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\INC\Atmel เพื่อเอาไว้เรียกใช้ในส่วนของการ Include File เมื่อเขียนโปรแกรมจะได้ไม่เสียเวลาประกาศตัวแปรต่างๆอีก
- 3) Copy File: STARTUP.A51 และ L51_BANK.A51 ที่ให้มากับแผ่น CD ไปวางไว้ที่ Folder C:\Keil\C51\LIB ซึ่งได้ทำการแก้ไขให้รองรับ MCU เบอร์นี้ไว้แล้ว แต่อาจจะไม่รองรับเบอร์อื่น ดังนั้นก็ควร Copy 2 ไฟล์นี้ ของเดิมที่มาพร้อมกับการติดตั้งเก็บไว้ด้วย เวลาจะเปลี่ยนไปใช้ MCU เบอร์อื่นจะได้นำมาวางทับได้เลย ไม่ต้องเสียเวลาติดตั้งโปรแกรมใหม่
- 4) เปิดโปรแกรม Keil μ Vision3 () ขึ้นมา เลือกที่เมนู Project และเลือก New Project... ดังรูปที่ 3.2.1



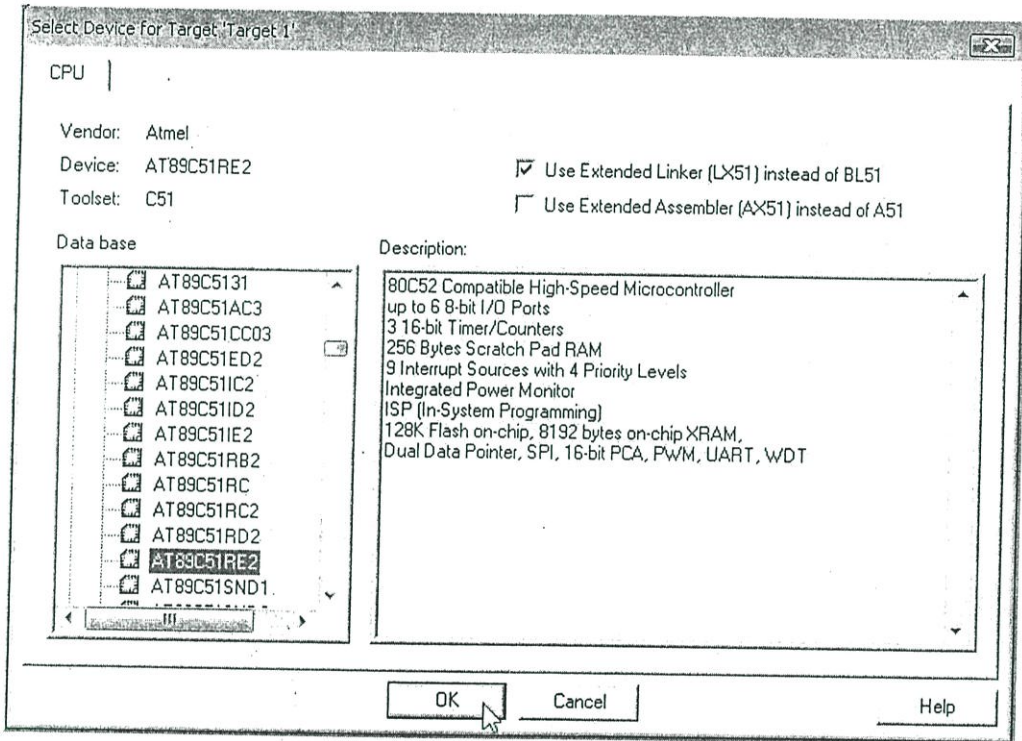
รูปที่ 3.2.1 แสดงการเลือก New Project

- 5) จะปรากฏ หน้าต่าง Create New Project ขึ้นมาดังรูปที่ 3.2.2 ให้เลือก Folder ที่จะเก็บ Project File และตั้งชื่อ Project File ในตัวอย่างจะตั้งชื่อ Project file คือ SW_Bank และเลือกเก็บที่ Folder Test_SW_Bank จากนั้น กด Save

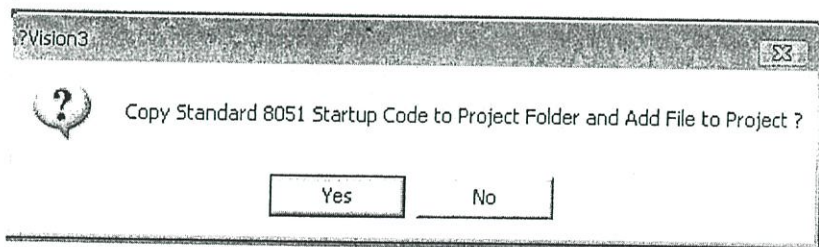


รูปที่ 3.2.2 แสดงหน้าต่าง Create New Project

- 6) หลังจากกด Save แล้ว จะปรากฏหน้าต่าง Select Device for Target ‘Target1’ ขึ้นมา ดังรูปที่ 3.2.3 (a) เพื่อให้ผู้ใช้เลือกเบอร์ MCU โดยในช่อง Data base ให้ผู้ใช้เลือกที่ Atmel และเลือกที่เบอร์ AT89C51 RE2 ให้สังเกต เมื่อเลือกแล้วเบอร์จะแสดงในช่อง Device: ด้านบน จากนั้นให้ทำการ Tick ที่หน้าช่อง Use Extended Linker(LX51) instead of BL51 แล้วกด OK จะมีหน้าต่าง Pop-Up ขึ้นมาดังรูปที่ 3.2.3 (b) ถ้ากด YES จะเป็นการนำ File Start Up (STARTUP.A51) มาตรฐานของ C51 เข้ามายัง Project หรือกด No ก็ได้ถ้าไม่ต้องการนำ File Start Up เข้ามายัง Project ในตัวอย่างนี้จะขอเลือก No

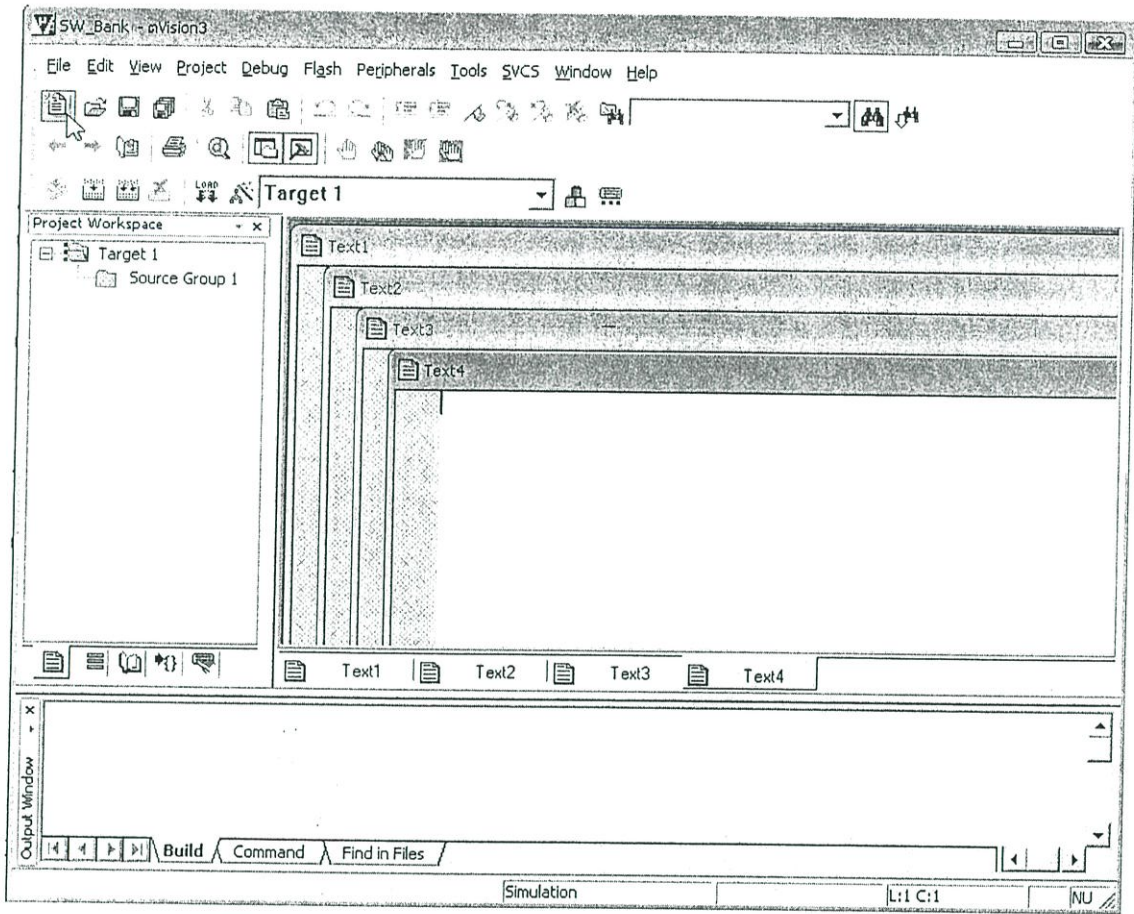


รูปที่ 3.2.3 (a) แสดงหน้าต่าง Select Device for Target ‘Target’



รูปที่ 3.2.3 (b)

- 7) จากนั้นให้คลิกที่ไอคอน Create a New File () 4 ครั้ง เนื่องจากในตัวอย่างนี้เราจะขอสร้างไฟล์ขึ้นมาใน Project 4 ไฟล์ (ในความเป็นจริงผู้ใช้จะสร้างขึ้นมาที่ ไฟล์ก็ได้ใน 1 Project) ดังนั้นจะทำให้ได้ หน้าต่าง Text1-Text4 ออกมา 4 หน้าต่าง ดังรูปที่ 3.2.4 ซึ่งจะใช้สำหรับ เขียน โปรแกรม



รูปที่ 3.2.4 แสดงหน้าต่าง Text1 สำหรับใช้เขียนโปรแกรม

- 8) ทำการเขียนโปรแกรมตามตัวอย่างด้านล่าง โดย File1เขียนลงในหน้าต่าง Text1 ,File2 เขียนลงในหน้าต่าง Text2 ไปเรื่อยๆจนครบ 4 ไฟล์ เมื่อเขียน โปรแกรมเรียบร้อยแล้วก็ให้ทำการ Save ไฟล์ทั้ง 4 โดยให้คลิกหน้าต่างของไฟล์ที่จะ Save ให้ Active ขึ้นมาอยู่ด้านหน้าสุด จากนั้นไปที่เมนู File เลือก save as...(รูป ก.) จะได้หน้าต่างดังรูป ข. ให้เลือก Save ไว้ใน Folder เดียวกับ Project File ในตอนแรก แล้วทำการตั้งชื่อไฟล์ เป็นจุด C ให้ทำการ Save จนครบ 4 ไฟล์ โดยในตัวอย่างจะขอตั้งชื่อไฟล์ดังนี้ File1 = Common_bank.c ,File2 = bank1.c , File3 = bank2.c และ File4 = bank3.c

ตัวอย่างโปรแกรม

File1 = common_bank.c

```

/*****
* Uart0_Test_Bank : RS232 *
* MCU Control : AT89C51RE2 *
* Use Port : ***** UART0 ***** *
* P3.0 = Rx *
* P3.1 = Tx *
* Compiler : Keil C51 (V8.05 a) *
* Function : Display Message to Hyper Terminal (BR9600) *
* *
*****/

#include <at89c51re2.h>
#include <stdio.h>
extern void Mess_Bank_1();
extern void Mess_Bank_2();
extern void Mess_Bank_3();
void Mess_Common_Bank()
{
    printf("\n\r Message form Common Bank:ETT \n\r");
}

//----- Main Program -----
main(void)
{
    char ch ;
    CKCON0 = 0xFE ; //use 12 clock
    //----- Initial Uart0 -----
    SCON_0 = 0x50 ; //Serial Model:8bit Uart
    TH1 = 0xFB ; //Set Baud rate = 9600 at Crystal=18.432 Mhz
    TL1 = 0xFB ;
    TMOD = 0x20 ; //Use Sclk Timer1 Mode2
    TI_0 = 1 ; //SCON Reg.: SET Flag TI
    TR1 = 1 ; //Set TCON Reg. On Timer1
    printf("Press Key : '0' or '1' or '2' or '3'\n\r");
    while(1)
    {
        ch = getchar() ; // Check Key
    }
}

```

```
switch(ch)
{
    case '0' : Mess_Common_Bank();
        break ;
    case '1' : Mess_Bank_1();
        break ;
    case '2' : Mess_Bank_2();
        break ;
    case '3' : Mess_Bank_3();
        break ;
    default : printf("\n\r Press Key : '0' or '1' or '2' or '3'\n\r");
}
}
```

File2 = bank1.c

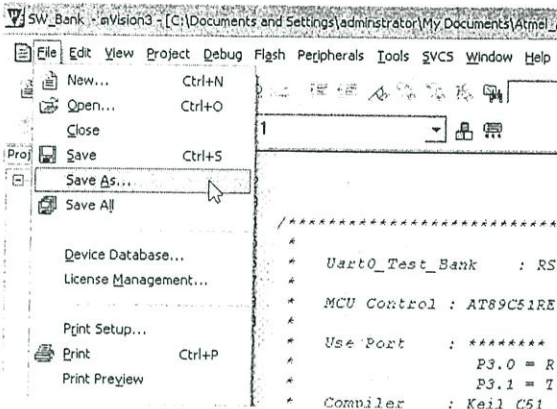
```
#include <at89c51re2.h>
#include <stdio.h>
void Mess_Bank_1()
{
    printf("\n\r Message form Bank1: Welcom \n\r");
}
```

File3 = bank2.c

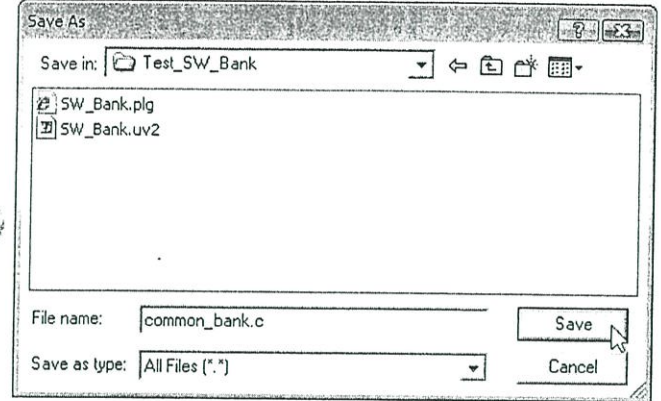
```
#include <at89c51re2.h>
#include <stdio.h>
void Mess_Bank_2()
{
    printf("\n\r Message form Bank2: Hello! \n\r");
}
```

File4 = bank3.c

```
#include <at89c51re2.h>
#include <stdio.h>
void Mess_Bank_3()
{
    printf("\n\r Message form Bank3: Good Morning. \n\r");
}
```



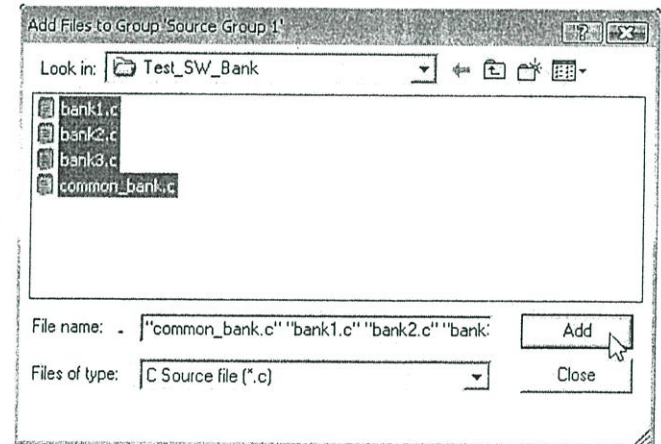
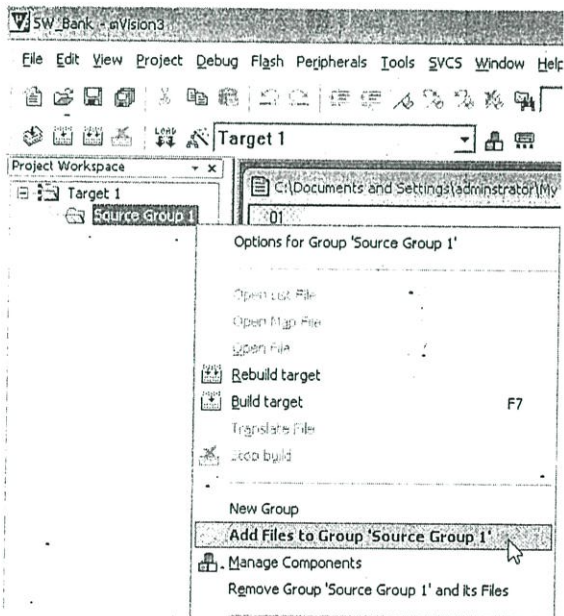
ก) เลือก Save As...



ข) ทำการตั้งชื่อ File.c และ Save File

รูปที่ 3.2.5 แสดงหน้าต่างการ Save File .c

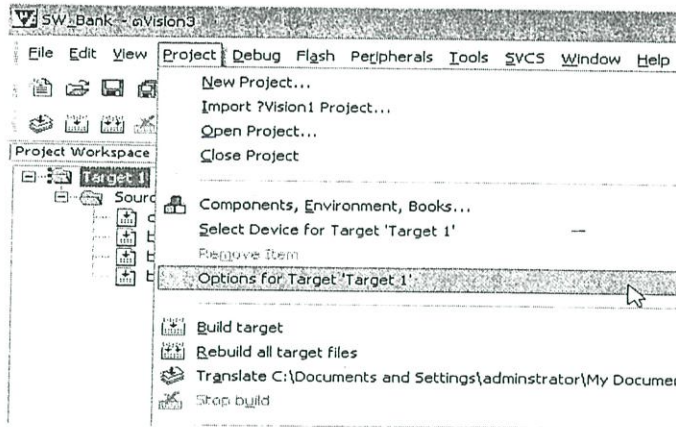
- 9) เมื่อ Save File ครบ 4 ไฟล์แล้วให้ทำการ Add File ทั้ง 4 เข้ามายัง Project โดยให้ Double Click หรือ คลิกขวาที่ Folder Source Group 1 ที่อยู่ในหน้าต่างด้านซ้ายมือ จากนั้นเลือก Add Files to Group 'Source Group 1' จะปรากฏหน้าต่างให้ Add file ขึ้นมา ให้ผู้ใช้เลือกไฟล์ทั้ง 4 แล้วกด Add จากนั้นกด Close เพื่อปิดหน้าต่าง Add File ไฟล์ที่ Add ก็จะสามารถเข้ามาอยู่ใน Folder Source Group 1



รูปที่ 3.2.6 แสดงหน้าต่างการ Add ไฟล์

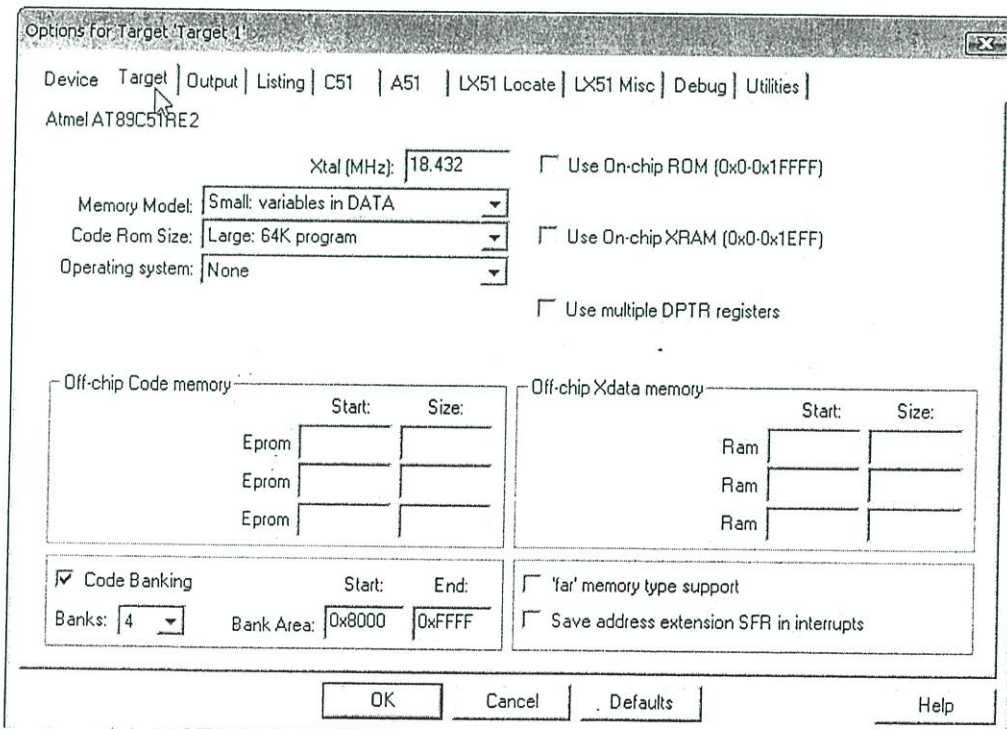
- 10) ให้ทำการ Add File L51_BANK.A51 เข้ามายัง Project ด้วย โดยให้ทำการ Add ไฟล์เหมือนในขั้นตอนที่ 9 จากนั้นเข้าไปยังตำแหน่งที่เก็บไฟล์นี้ นั่นก็คือ C:\Keil\C51\LIB (สังเกตว่าจะเป็นไฟล์ที่ให้ Copy จากแผ่นไปวางไว้ในตอนแรกซึ่งได้ทำการแก้ไขให้ใช้กับ MCU เบอร์นี้ไว้แล้ว)

- 11) เมื่อ Add ไฟล์ทั้งหมดเข้ามาเรียบร้อยแล้วให้ไปคลิกที่ Folder Target 1 ในช่องด้านซ้ายมือให้เป็นแถบสีน้ำเงิน จากนั้นไปที่เมนู Project แล้วเลือก Options for Target 'Target 1' ดังแสดงในรูปที่ 3.2.7



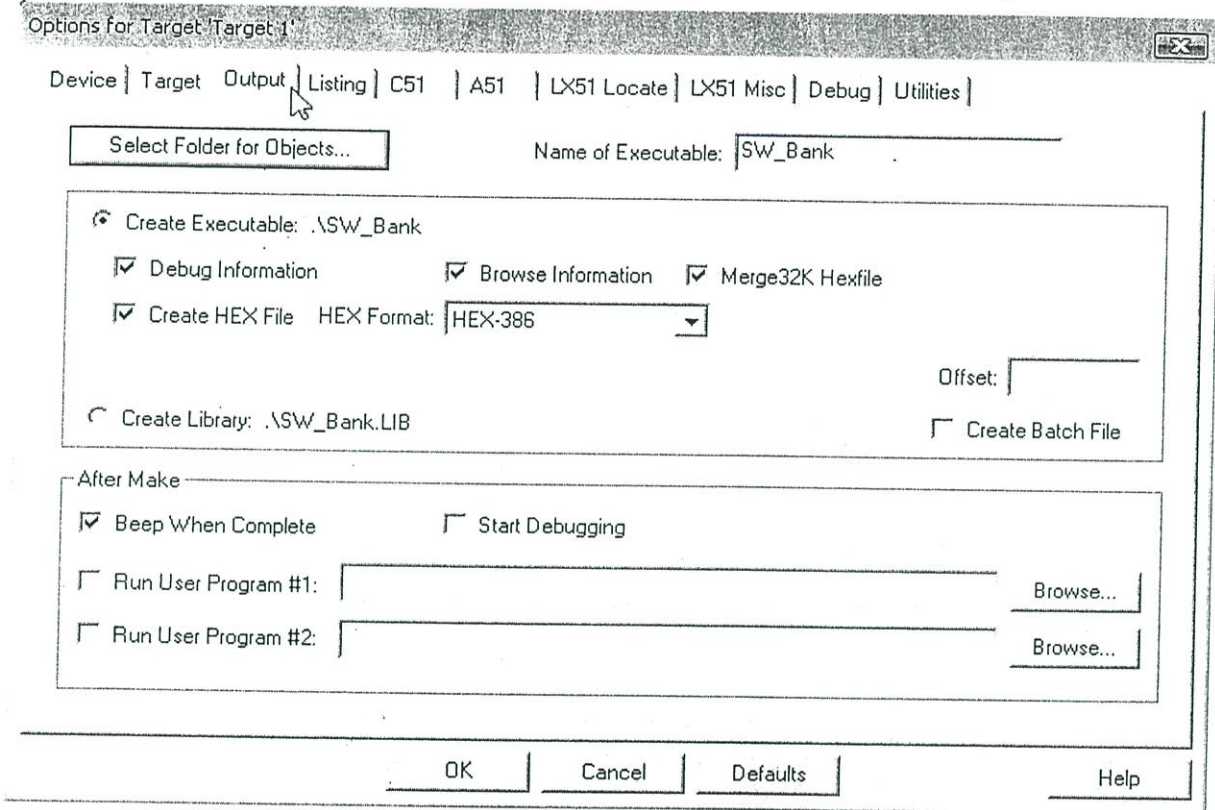
รูปที่ 3.2.7 แสดงการเลือก Option เพื่อ Setup Project

- 12) จะได้นหน้าต่าง Options for Target 'Target1' ออกมา ให้คลิกที่ TAB Target แล้วทำการกำหนดค่าตามรูปที่ 3.2.8 ให้สังเกตในช่อง Memory Model : ซึ่งในช่องนี้จะเป็นการเลือกขนาดของ Ram ที่ใช้ในการเขียน โปรแกรม ถ้าผู้ใช้เลือก Small: variables in DATA ตัวแปรที่ ประกาศในโปรแกรม จะถูกเก็บไว้ยังพื้นที่ RAM ภายใน ซึ่งขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 128 byte , ถ้าเลือก Compact : variable in PDATA ขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 256 Byte , และถ้าเลือก Large : variable in XDATA ตัวแปรก็จะถูกเก็บไว้ในพื้นที่ XRAM ดังนั้น ขนาดของตัวแปรที่ประกาศในโปรแกรมจะต้องมีขนาดรวมกันไม่เกิน 8 K byte ซึ่งแล้วแต่ผู้ใช้จะเลือกใช้ จากนั้นให้ Tick ที่ช่อง Code Banking แล้วกำหนดค่าตามรูป



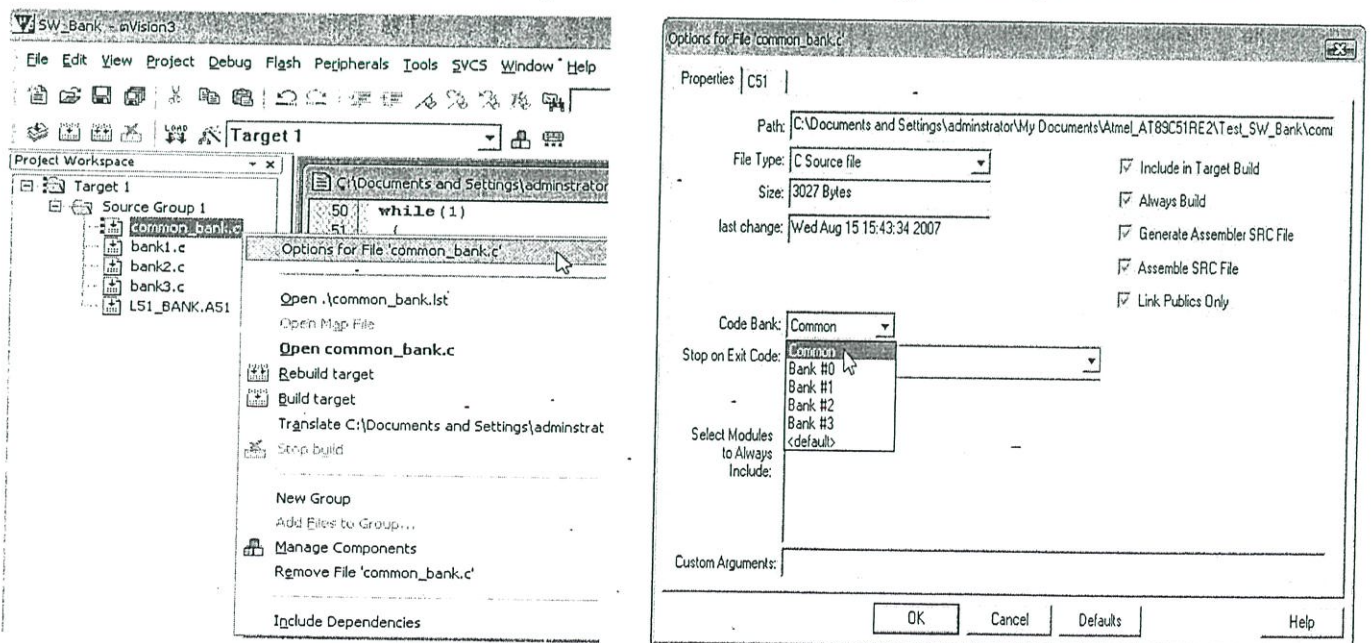
รูปที่ 3.2. 8 แสดงหน้าต่าง Option for Target 'Target 1' ที่ TAB Target

- 13) คลิกที่ TAB Output แล้วทำการ Tick เครื่องหมายถูกหน้าช่อง Create HEX File ในช่อง HEX Format : ให้เลือก HEX-386 แล้วทำการ Tick ช่อง Merge32K Hexfile ส่วนช่องอื่นๆก็ให้กำหนดเหมือนในรูป เสร็จแล้วกด OK



รูปที่ 3.2.9 แสดงหน้าต่าง Option for Target 'Target 1' ที่ TAB Output

- 14) ต่อไปจะเป็นการเลือก Bank ในการ Flash Code ให้กับไฟล์แต่ละไฟล์ดังนี้ ให้คลิกขวาที่ไฟล์ที่ต้องการจะกำหนด Bank สำหรับ Flash Code แล้วเลือกที่ Option for File ดังรูปที่ 3.2.10 จะปรากฏหน้าต่าง Option for File ออกมา




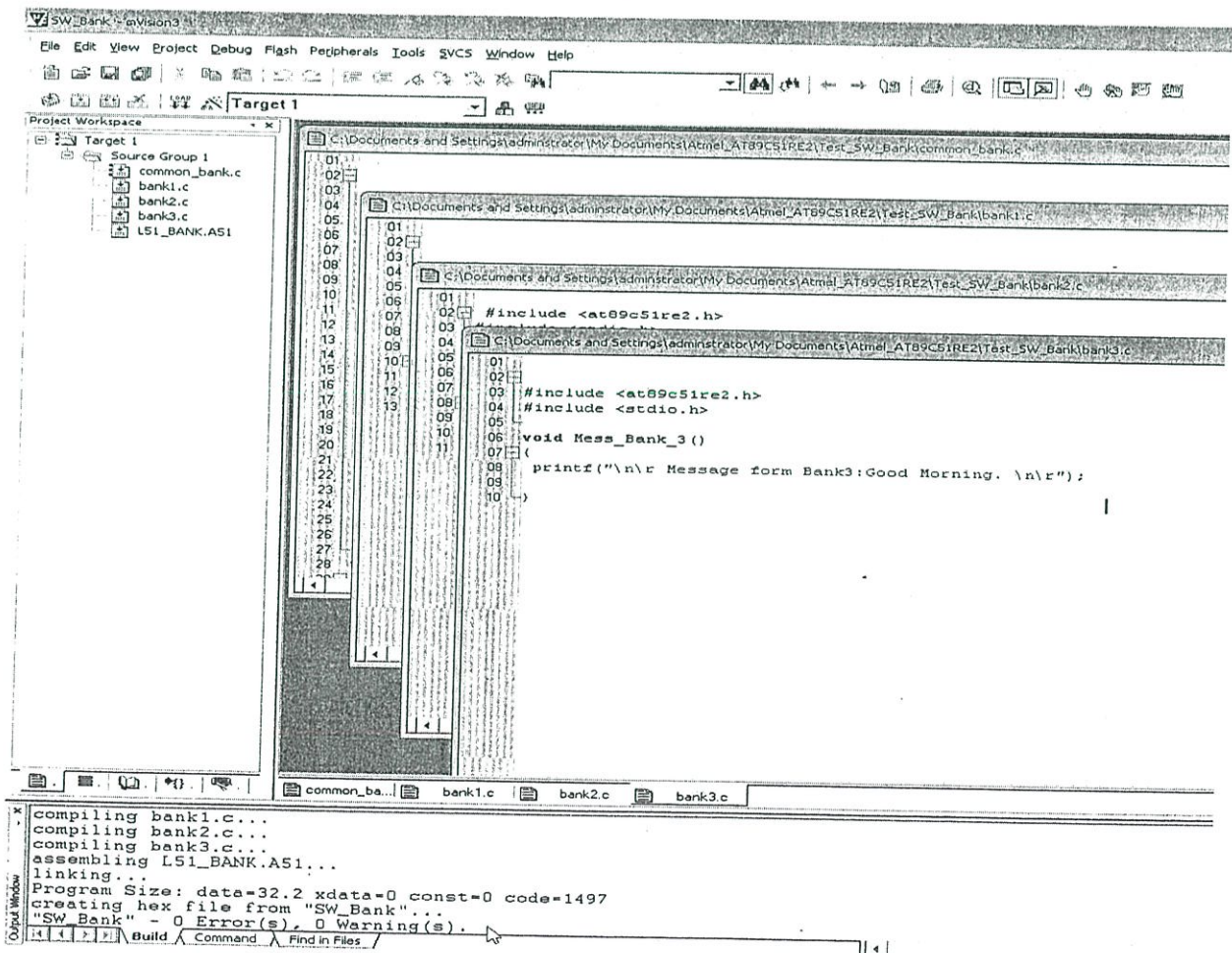
รูปที่ 3.2.10 แสดงการเลือกกำหนด Bank สำหรับ Flash Code

ให้ดูที่ช่อง Code Bank : ให้ผู้ใช้ทำการเลือก Bank ที่ต้องการจะ Flash Code ลงไป เมื่อเลือกแล้วให้กด OK จากนั้นก็ทำเช่นเดียวกันนี้กับไฟล์อื่นที่เหลือเมื่อต้องการจะเปลี่ยน Bank สำหรับ Flash Code โดยปกติแล้ว ทุกไฟล์ที่ถูก Add เข้ามาจะถูกกำหนดไว้ที่ค่า default ซึ่งก็จะเป็น Bank เดียวกับ Common ซึ่งผู้ใช้สามารถเปลี่ยนแปลงเป็น Bank อื่นได้

ในการเลือก Bank นี้ มีข้อจำกัดอยู่ว่า ห้ามเลือก Bank#0 และ ในไฟล์ที่แอดเข้ามาทั้งหมด ห้ามกำหนดให้เป็น Common หรือ default เหมือนกันทุกไฟล์จะต้องมีอย่างน้อย 1 ไฟล์ที่ถูกกำหนดเป็น Bank#1 หรือ Bank#2 หรือ Bank#3 มิเช่นนั้นจะทำให้มีปัญหาเวลาใช้ FLIP Down Load Code

ข้อควรจำ การกำหนดค่าในช่อง Code Bank : ห้ามกำหนดให้ File ทุก File ใน Project เป็น Common หรือ default เหมือนกันหมดต้องมีไฟล์ใดไฟล์หนึ่งที่แตกต่าง และห้ามกำหนดทุกไฟล์ให้อยู่ใน Bank#0 (ค่าที่สามารถเลือกได้คือ Common, default, bank#1, bank#2, bank#3) ไฟล์ที่กำหนดลงไปในแต่ละ Bank จะต้องรวมกันไม่เกิน 32K

- 15) เมื่อ Set ค่าให้กับ Project และกำหนด Bank ให้แต่ละไฟล์เรียบร้อยแล้วให้ทำการ Compile โปรแกรมเพื่อตรวจสอบว่ามี Error หรือไม่ โดยคลิกที่ไอคอน Rebuild all target files () ถ้าโปรแกรมที่เขียนไม่มีปัญหา ก็จะแสดง Error เป็น 0 อยู่ในหน้าต่างด้านล่าง ดังแสดงในรูปที่ 3.2.11



รูปที่ 3.2.11 แสดงหน้าต่างหลังจาก Compile ผ่านแล้ว

- 14) เมื่อ Compile เรียบร้อย ก็ให้ทำการ Download โปรแกรมลงใน MCU ได้ โดยใช้โปรแกรม Flip คู่มือการ Download ได้ในหัวข้อที่ 4 การ Download โปรแกรม ด้วย FLIP



สำหรับในตัวอย่างนี้เมื่อผู้ใช้ Download โปรแกรมเรียบร้อยแล้ว สามารถทำการทดสอบโปรแกรมโดยต่อสาย RS232 จาก COM PORT ของ PC มายัง Port RS232#1 ของบอร์ด จากนั้นเปิดโปรแกรม HyperTerminal ขึ้นมา โดย Set Baud Rate ที่ 9600 จากนั้นกด Reset ที่บอร์ด จะมีข้อความแสดงขึ้นที่หน้าจอว่า “ Press Key : '0' or '1' or '2' or '3' ” จากนั้นลองกดคีย์บอร์ดที่ key 0 หรือ 1 หรือ 2 หรือ 3 จะต้องปรากฏข้อความแสดงออกมาที่หน้าจอ ซึ่งในตัวอย่างนี้จะแสดงให้เห็นถึงการให้ Switch Bank ระหว่างโปรแกรมที่ถูกเก็บอยู่ในแต่ละ Bank ว่าสามารถทำงานได้อย่างถูกต้องหรือไม่เมื่อทำการเรียกใช้งาน Function อื่นที่อยู่กันคนละ Bank

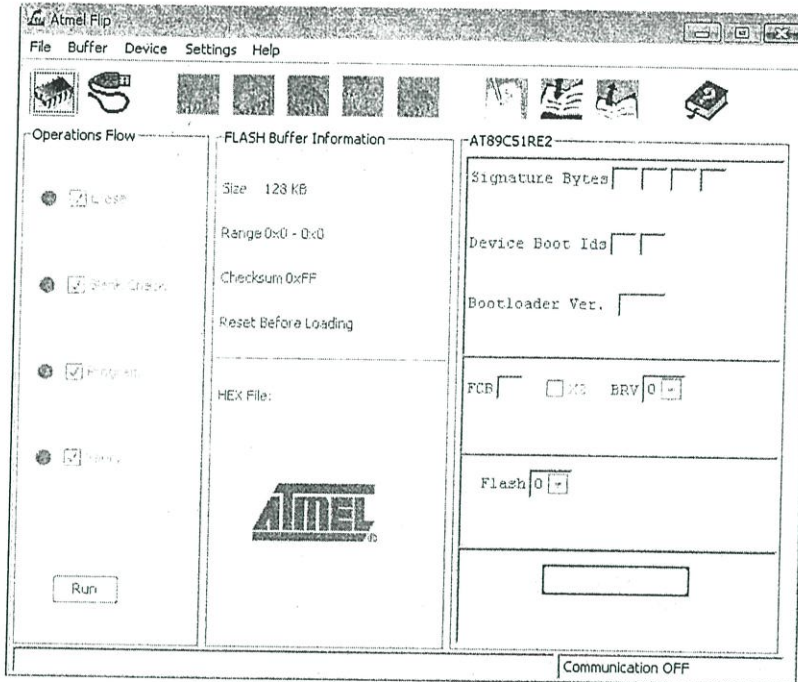
4. การ Download โปรแกรม ด้วย FLIP

สำหรับบอร์ด CP-JR51RE2 V1.0 นี้ เราจะโหลดโปรแกรมที่เขียนขึ้นลงไปในตัว MCU โดยใช้โปรแกรม FLIP V3.1.0 Build 1 เป็นตัว Download โดย File ที่ Flip จะใช้ Download นั้น จะต้องเป็น File.hex ซึ่งไฟล์นี้จะได้มาจากในหัวข้อที่ 3 คือเมื่อเราเขียนโปรแกรมเรียบร้อยแล้ว และ Compile ผ่านแล้ว ตัว Keil uVision3 ก็จะสร้าง File.hex ให้กับผู้ใช้ จากนั้นถึงจะใช้ โปรแกรม Flip เป็นตัว Download File.hex นั้นลงบน MCU อีกทีหนึ่ง

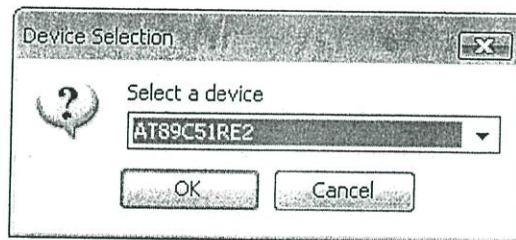
ในการ Download ก่อนอื่นผู้ใช้จะต้องทำการติดตั้งโปรแกรม Flip ที่ให้มากับแผ่น CD ลงในเครื่องก่อนซึ่งโปรแกรม Flip นี้จะต้องทำงานร่วมกับ Java ดังนั้นผู้ใช้จะต้องทำการติดตั้ง Java ลงไปในเครื่องด้วย ซึ่งจะสรุปขั้นตอนการ Download ได้ดังนี้ -

ขั้นตอนการ Download โปรแกรม (Hex File)


- 4.1) ทำการติดตั้งโปรแกรม Java ที่ให้มากับแผ่น CD ลงในเครื่องก่อนถ้าเครื่องของผู้ใช้ยังไม่มีโปรแกรม Java อยู่
- 4.2) ทำการติดตั้งโปรแกรม Flip ที่ให้มากับแผ่น CD ลงในเครื่องให้เรียบร้อยแล้ว
- 4.3) ต่อสาย RS232 จาก PC เข้ากับขั้วต่อ RS232#1 ของบอร์ด MCU
- 4.4) ทำการกดสวิทช์ PSEN และ RESET ดังต่อไปนี้เพื่อเข้าสู่ Monitor Mode
 - กด SW. PSEN ค้างไว้
 - ตามด้วยการกด SW. RESET ค้างไว้
 - ปลด SW. RESET ในขณะที่ SW. PSEN ยังถูกกดค้างอยู่
 - ปลด SW. PSEN เป็นลำดับสุดท้าย
- 4.5) ให้เปิดโปรแกรม Flip ขึ้นมา [] จะได้หน้าต่าง Atmel Flip ดังรูปที่ 4.1 ออกมา
- 4.6) ให้คลิกที่ไอคอน Select a Target Device [] จะได้หน้าต่าง Device Selection ดังรูปที่ 4.2 ในหน้าต่างนี้ให้ผู้ใช้ทำการเลือกเบอร์ MCU ที่ใช้งานในที่นี้คือ #AT89C51RE2 เมื่อเลือกเสร็จให้กด OK.

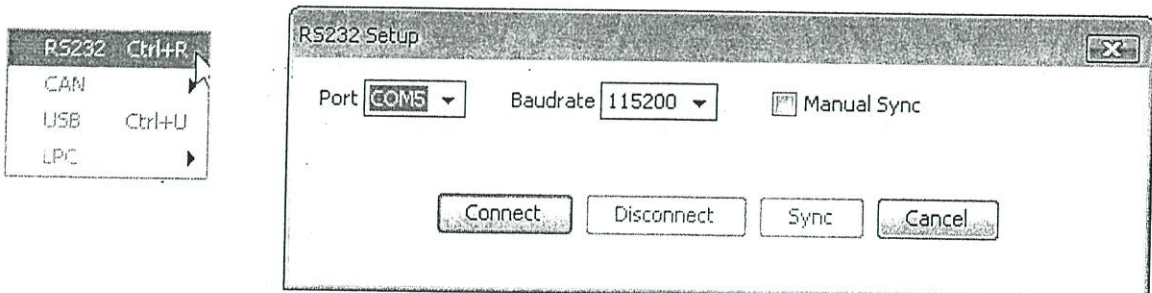


รูปที่ 4.1 แสดงหน้าต่าง Atmel Flip



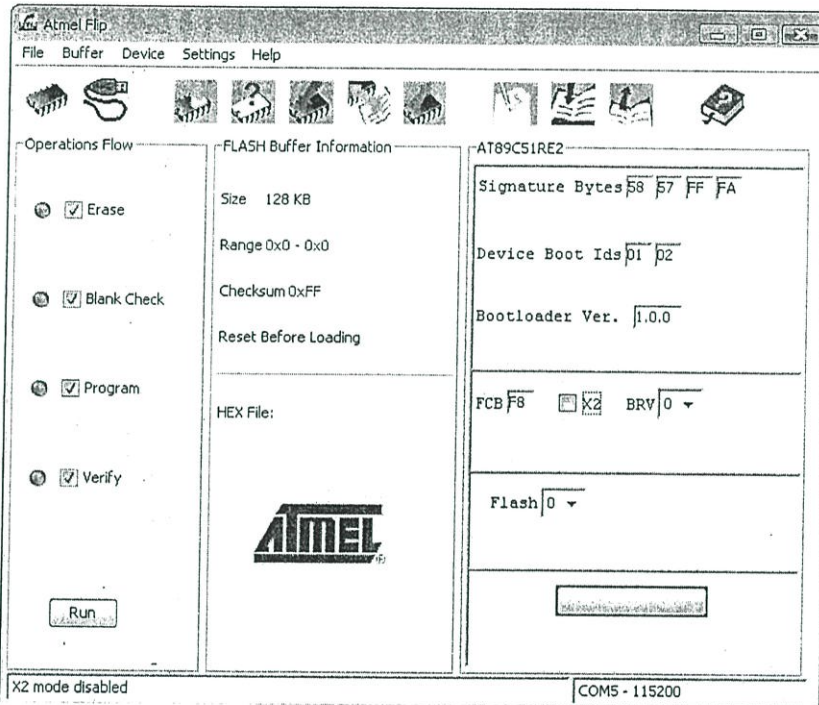
รูปที่ 4.2 แสดงหน้าต่าง Device Selection สำหรับเลือกเบอร์ MCU ที่ใช้งาน

- 4.7) ให้คลิกที่ไอคอน Select a Communication Medium [] เพื่อเลือก Port ในการสื่อสาร เมื่อคลิกแล้วจะได้หน้าต่างรูปเล็กออกมานั้นให้เลือกที่ RS232 ก็จะได้หน้าต่าง RS232 Setup ดังรูปที่ 4.3 จากนั้นให้เลือก ComPort ของ PC และเลือก Baud Rate (default = 115200) ที่จะใช้ในการ Download โปรแกรม เมื่อเลือกเรียบร้อยแล้วให้คลิก Connect

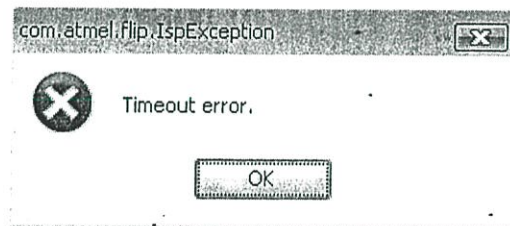


รูปที่ 4.3 แสดงหน้าต่าง RS232 Setup สำหรับเลือก Com Port และ Baud Rate ในการ Download

ก่อนที่จะคลิก Connect นั้นต้องแน่ใจว่า MCU ทำงานอยู่ใน Monitor Mode แล้ว ถ้ายังไม่แน่ใจก็ให้ทำการกด SW. PSEN และ RESET ตามขั้นตอนในข้อ 4.4 อีกครั้งหนึ่ง ถ้าการ Connect ผ่าน ไอคอนต่างๆ ในหน้าต่าง Atmel Flip ก็จะถูก Enable ให้ใช้งานได้ ดังแสดงในรูปที่ 4.4 แต่ถ้า Connect ไม่ผ่านก็จะมีหน้าต่างขึ้นมาเตือนดังในรูปที่ 4.5 ซึ่งผู้ใช้งานจะต้องทำการตรวจสอบ การต่อสาย RS232 และ การเข้าสู่ Monitor Mode ว่าถูกต้องหรือไม่ แล้วค่อยลองทำการ Connect ใหม่อีกครั้ง

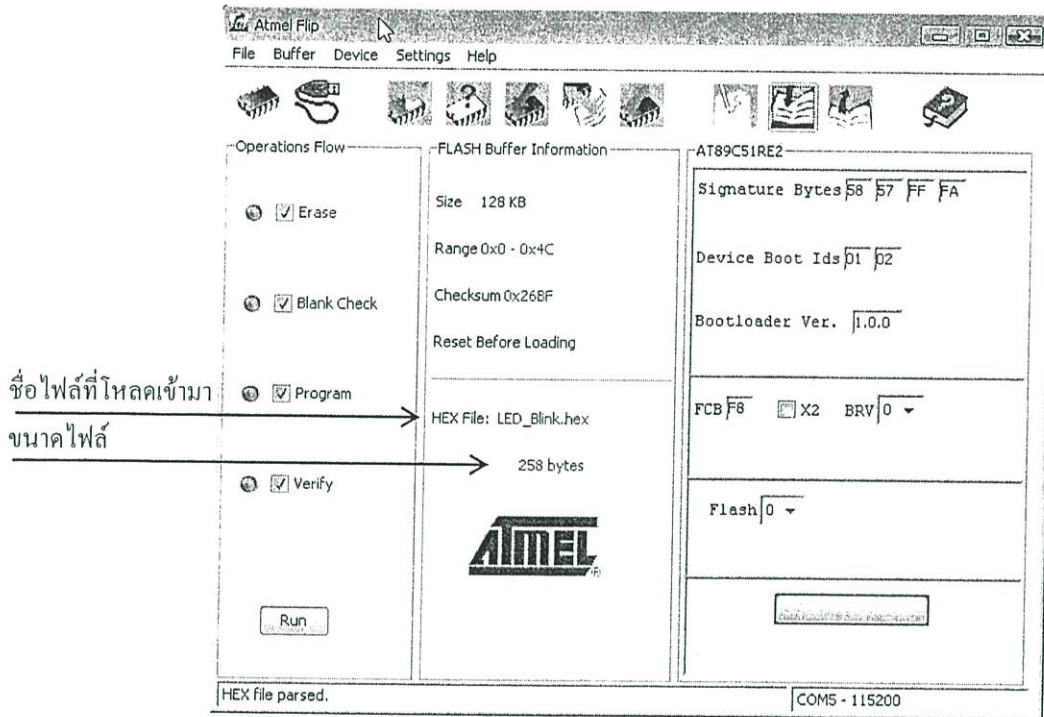


รูปที่ 4.4 แสดงหน้าต่าง Atmel Flip หลังจาก Connect Port สำเร็จ



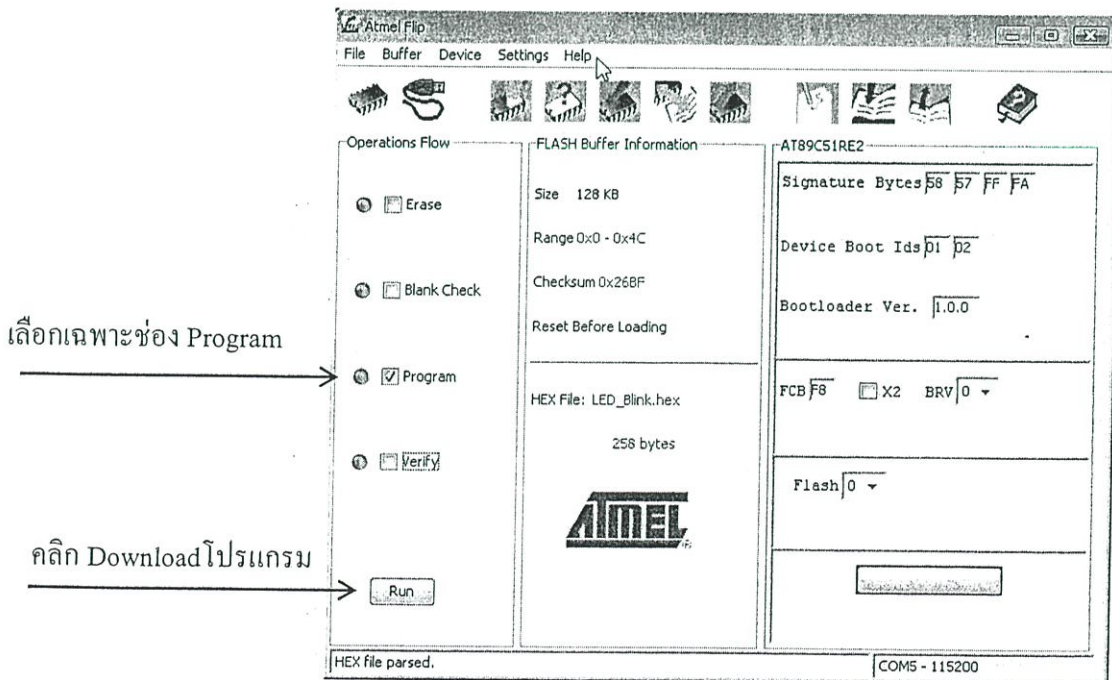
รูปที่ 4.5 แสดงหน้าต่าง Timeout error หลังจาก Connect Port เมื่อ Connect Port ไม่สำเร็จ

4.8) หลังจาก Connect Port ได้เรียบร้อยแล้วให้คลิกที่ไอคอน Load HEX file แล้วทำการเลือก file.Hex จากภายนอก เข้ามายัง โปรแกรม Flip ชื่อของ file ที่เลือกเข้ามาจะปรากฏที่หน้าต่าง Atmel Flip ในช่อง Hex File : ดังแสดงใน รูปที่ 4.6



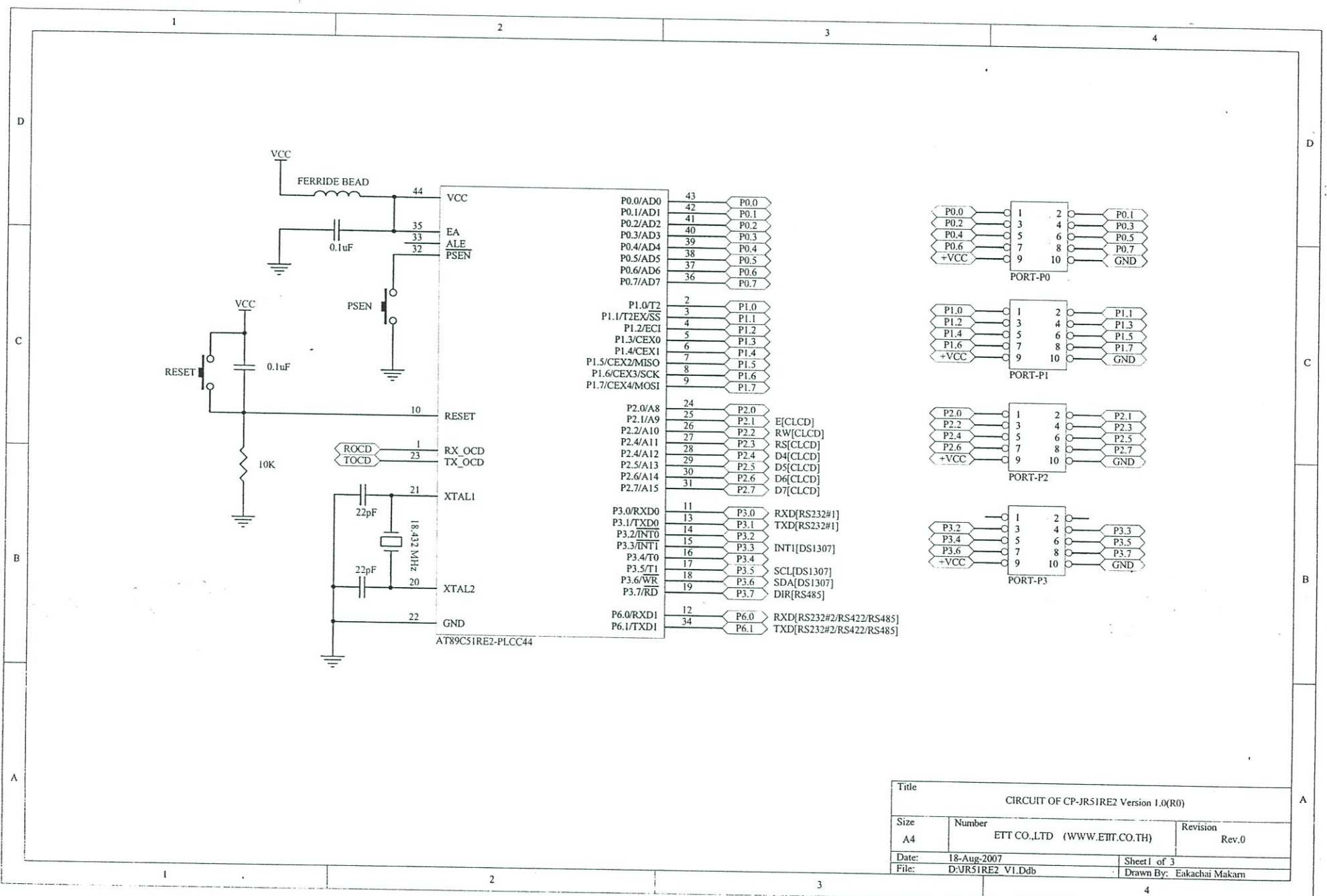
รูปที่ 4.6 แสดง Hex File ที่ถูกโหลดเข้า

- 4.9) เมื่อโหลดไฟล์เข้ามาแล้ว ให้คลิกที่ปุ่ม Run เพื่อ Download โปรแกรมลงไปใน MCU ซึ่งถ้าผู้ใช้ต้องการให้โหลดเร็วขึ้น จากหน้าต่าง Atmel Flip ก่อนที่จะคลิกปุ่ม Run ในช่อง Operations Flow ให้ Tick เฉพาะช่อง Program เพียงช่องเดียวพอ ส่วนช่องอื่นให้ Tick ออกทั้งหมด ดังแสดงในรูปที่ 4.7 แล้วถึงคลิกปุ่ม Run

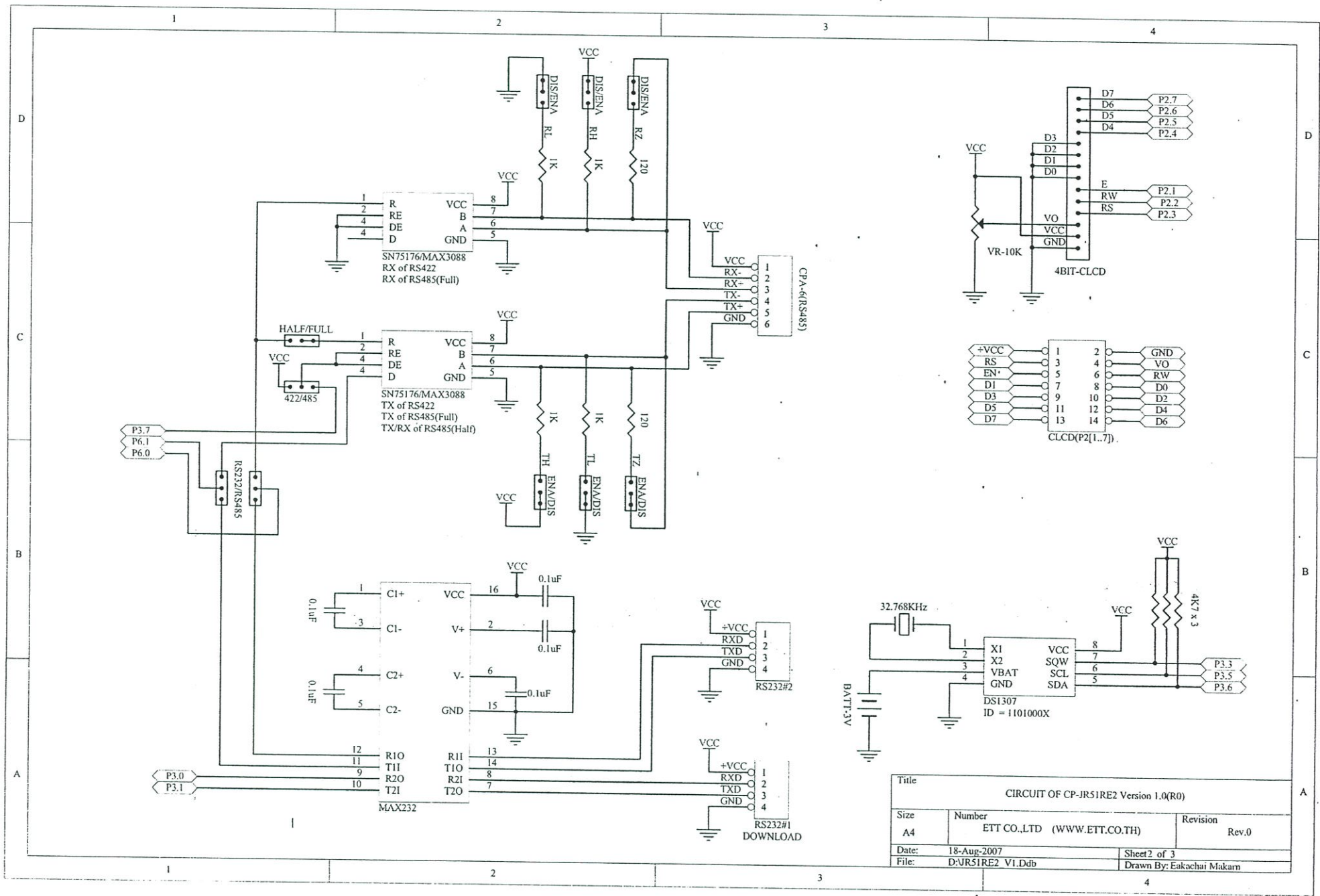


รูปที่ 4.7 แสดงการ Tick ช่อง Program และปุ่ม Run เพื่อ download

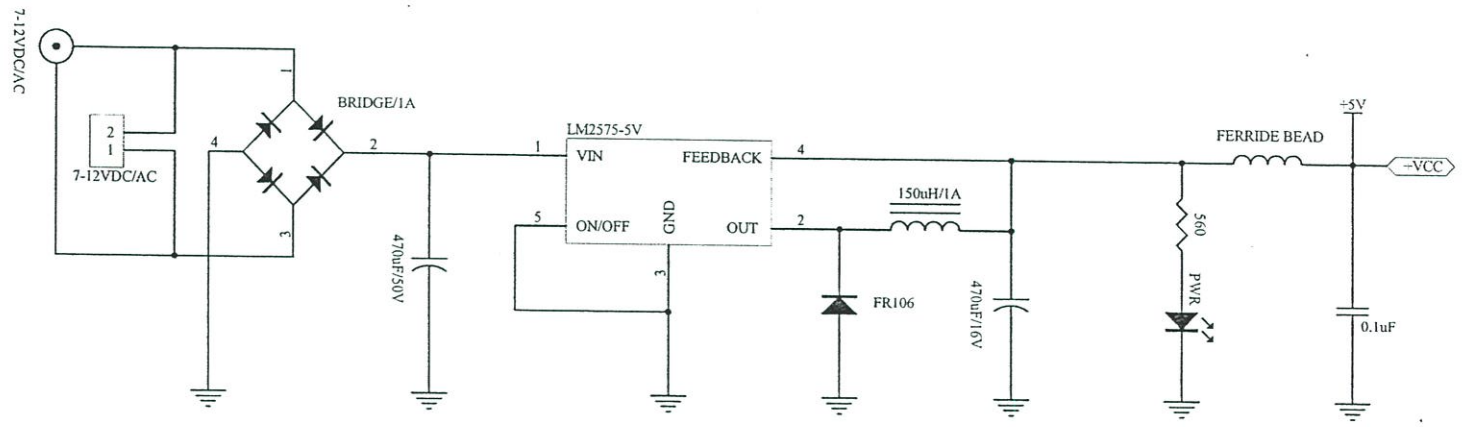
-
- 4.10) หลังจาก Download เรียบร้อยแล้ว ให้กดที่สวิตซ์ RESET ของบอร์ด CP-JR51RE2 V1.0 เพื่อ RUN โปรแกรมดูผลการทำงานของโปรแกรมที่เขียนขึ้น เมื่อจะ Download โปรแกรมใหม่ก็ให้ไปเริ่มทำในขั้นตอนที่ 4.4 ใหม่



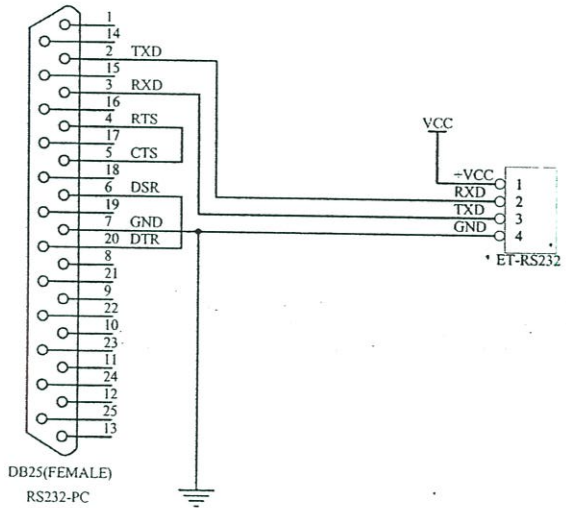
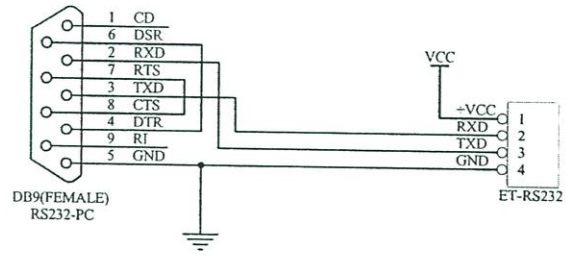
Title		
CIRCUIT OF CP-JRS1RE2 Version 1.0(R0)		
Size	Number	Revision
A4	ETT CO.,LTD (WWW.ETT.CO.TH)	Rev.0
Date:	18-Aug-2007	Sheet 1 of 3
File:	D:\JRS1RE2_V1.Ddb	Drawn By: Eiakchai Makam



Title			CIRCUIT OF CP-JR51RE2 Version 1.0(R0)		
Size	Number	Revision			
A4	ETT CO.,LTD (WWW.ETT.CO.TH)	Rev.0			
Date:	18-Aug-2007	Sheet 2 of 3			
File:	D:\UR51RE2 V1.Ddb	Drawn By: Eakachai Makam			



Title		
CIRCUIT OF CP-JR51RE2 Version 1.0(R0)		
Size	Number	Revision
A4	ETT CO.,LTD (WWW.ETT.CO.TH)	Rev.0
Date:	18-Aug-2007	Sheet 3 of 3
File:	D:\JR51RE2 V1.Ddb	Drawn By: Eakachai Makarn



Title			
ETT CO.,LTD (WWW.ETT.CO.TH)			
Size	Number	Revision	
A4	CABLE RS232	1.0	
Date:	19-Jan-2006	Sheet 1 of 1	
File:	D:\My Circuit\RS232 Cable\RS232CABLE.Dwg		