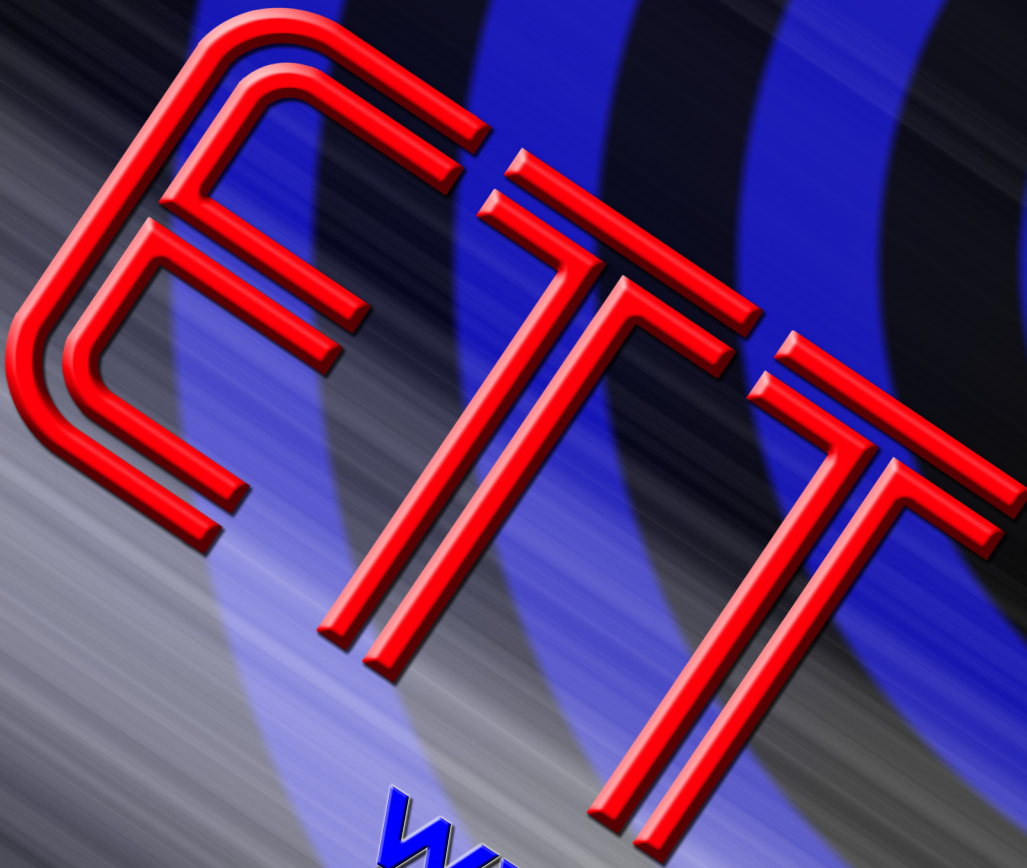


คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์

CP-JR51USB V1.0



[www.ett.co.th](http://www.ett.co.th)

1112/96-98 ถนนสุขุมวิท แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10110

1112/96-98 Sukhumvit Rd., Phrakanong Klongtoey BANKOK 10110

TEL. 02-712 1120 FAX : 02-391 7216

## คำนำ

### *Preface*

ปัจจุบันผู้ผลิตคอมพิวเตอร์ส่วนใหญ่เริ่มลดจำนวน RS-232 COM Port น้อยลงเรื่อยๆ หรือ บางเครื่องไม่มี COM Port เลย แต่กลับให้ความสำคัญกับการเพิ่มจำนวนของพอร์ต USB มากขึ้น โดยเฉพาะกับคอมพิวเตอร์แบบ Notebook , ดังนั้น บอร์ด CP-JR51USB v1.0 จึงใช้ CPU ที่รวมความสามารถในการเชื่อมต่อระบบบัสแบบ USB ไว้ภายในด้วย จึงทำให้บอร์ด CP-JR51USB v1.0 ใช้การติดต่อสื่อสารในระบบบัส USB ได้ ซึ่งรองรับทั้ง USB เวอร์ชัน 1.1 และ 2.0 Full Speed

บอร์ด CP-JR51USB v1.0 ภายในใช้ชิปเบอร์ AT89C5131 ของบริษัท Atmel ที่มีคุณสมบัติเด่นในเรื่องของการสื่อสารแบบ USB และ สามารถทำงานที่ความถี่สูงสุด 48 MHz , มีพื้นที่หน่วยความจำโปรแกรมแบบ Flash memory จำนวน 32 kBytes, มีรีจิสเตอร์ DPTR 2 ตัว , 16-bit Timer/Counter จำนวน 3 ตัว และอื่นๆ

สุดท้ายนี้ผู้เขียน และ ทีมงาน ETT หวังเป็นอย่างยิ่งว่าหนังสือคู่มือ และ บอร์ดทดลอง CP-JR51USB v1.0 นี้จะช่วยให้ผู้อ่าน และ ผู้สนใจในชิป AT89C5131 ได้รับประโยชน์อย่างเต็มประสิทธิภาพ และ หากผู้อ่านมีข้อคิดเห็น หรือ มีข้อท้วงติงเกี่ยวกับเนื้อหาภายในหนังสือคู่มือฉบับนี้ สามารถให้ข้อคิดเห็นผ่านทาง e-mail : [sales@etteam.com](mailto:sales@etteam.com) ซึ่งทางผู้เขียน และ ทีมงาน ETT จะขอน้อมรับไว้เพื่อการปรับปรุงในการพิมพ์ครั้งต่อไป

จักรพันธ์ จิตรทรัพย์

16 กันยายน 2547

## สารบัญ

## Contents

## บทที่ 1

ลักษณะทั่วไปของบอร์ด CP-JR51USB v1.0	1
✦ รายละเอียด	1

## บทที่ 2

ทรัพยากร บนบอร์ด CP-JR51USB v1.0	3
✦ การจัดสรร I/O	3
✦ พอร์ต USB	3
✦ พอร์ต 34 Pin	5
✦ พอร์ต I2C BUS	6
✦ พอร์ต RS232	6
✦ พอร์ต RS422/485	7
✦ การสื่อสารอนุกรมแบบ RS422	8
✦ การสื่อสารอนุกรมแบบ RS485	9
✦ การกำหนด Jumper สำหรับการสื่อสารแบบ RS422/485	10

## บทที่ 3

การติดตั้งโปรแกรม Flip 1.8.2	14
✦ ระบบที่ต้องการ	14
✦ การติดตั้งโปรแกรม Flip1.8.2	15

## บทที่ 4

การใช้โปรแกรม Flip 1.8.2 เบื้องต้น

19

ภาคผนวก ก. SFRs MAP & Reset Values

ภาคผนวก ข. PINOUT & BLOCK DIAGRAM

ภาคผนวก ค. CP-JR51USB v1.0 SCHEMATICs

ภาคผนวก ง. โปรแกรม FLIP 1.8.2



## 1

### ลักษณะทั่วไปของบอร์ด

#### CP-JR51USB v1.0

บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-JR51USB v1.0 ภายในบอร์ดใช้ชิปเบอร์ AT89C5131 ของบริษัท Atmel ซึ่งชิปเบอร์นี้มีคุณสมบัติเด่นในเรื่องของการเชื่อมต่อระบบบัส USB และสามารถเลือกโหมดการทำงานเป็นแบบ 6clocks /1 Machine cycle หรือ 12 clocks /1 Machine cycle ได้ ซึ่ง จะไม่มีปัญหาเกี่ยวกับการสื่อสารอนุกรมเนื่องจากภายในชิป AT89C5131 นั้นได้มีส่วนของ Internal Baud Rate Generator อยู่ภายใน ซึ่งจะทำให้ผู้ใช้สามารถสร้างค่า Baud rate ใดๆ ได้ตามต้องการ

คุณสมบัติเด่นในส่วนของการเชื่อมต่อ USB, AT89C5131 ภายในได้บรรจุโมดูล Full-speed USB ที่ใช้งานได้กับเวอร์ชัน 1.1 และ 2.0 ซึ่งผู้อ่านสามารถส่ง หรือ รับข้อมูลผ่านทางบัส USB ด้วยอัตรา 12MHz ได้ นอกจากนั้นชิป AT89C5131 ยังมีหน่วยความจำ Flash ถึง 32 Kbytes , Internal RAM 256 bytes , ระดับความสำคัญของสัญญาณ Interrupt 4 – Level , 16-bit Timer/Counters มี 3 ตัว , Full duplex enhanced UART (EUART) , ERAM = 1024 bytes , Dual data pointer , แหล่งกำเนิดกระแสที่สามารถเลือกปริมาณกระแสเพื่อใช้ขับ LED 4 แหล่ง ฯลฯ

นอกจากนั้น AT89C5131 ยังมีโหมดเกี่ยวกับการประหยัดพลังงานอีก 2 โหมด คือ 1.) Idle mode และ 2.) Power-down mode ซึ่งจะทำให้การใช้พลังงานของ CPU ในขณะนั้นน้อยลง







## 2

### ทรัพยากรบนบอร์ด

CP-JR51USB v1.0

การจัดสรรทรัพยากรบนบอร์ด CP-JR51USB v1.0 ภายในใช้ชิป AT89C5131 ซึ่งมี I/O ทั้งหมด 34 Pins และ สามารถใช้ x'tal สูงสุดที่ 48 MHz ( X1 mode ) และ สูงสุดที่ 24MHz ( X2 mode ) แต่สำหรับ บอร์ด CP-JR51USB v1.0 ภายในจะใช้ x'tal = 24MHz

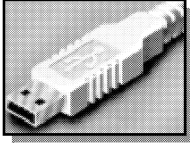

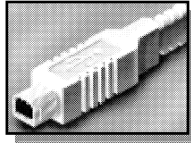
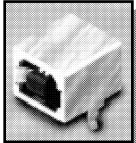
### การจัดสรร I/O

จากที่ได้กล่าวไปแล้วว่า I/O ของชิป AT89C5131 นั้นมีทั้งหมด 34 Pins 5 พอร์ต ซึ่งได้แก่ Port 0, Port 1, Port 2, Port 3, Port 4 โดยพอร์ต 4 จะมีแค่ 2 บิตซึ่งใช้ในการสื่อสารอนุกรม TWI (I<sup>2</sup>C) โดย 2 ขานี้ จะต่อออกไปที่พอร์ต I2C BUS และ ขณะเดียวกันก็ต่อให้กับ IC DS1307 กับ EEPROM : 24LS256 ด้วย

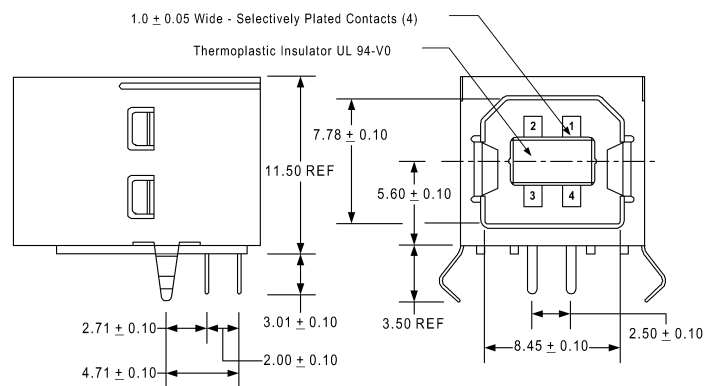
ในส่วนของพอร์ต P0 – P3 จะต่อออกไปที่พอร์ต 34 Pin ทั้งหมด ยกเว้น 2 ขา คือ P3.0 และ P3.1 ซึ่งถูกใช้งานในส่วนของการสื่อสารอนุกรม ( UART ) เท่านั้น

### พอร์ต USB

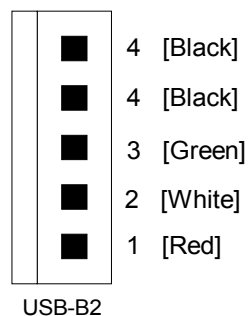
พอร์ต USB จะมีทั้งหมด 4 ขา ดังแสดงในรูปด้านล่าง ซึ่งจะต่อกับ CPU ที่ขา D+, D- และ Vref โดยทาง ETT ได้จัดทำพอร์ต USB ไว้ 2 แบบ คือ 1. ปลั๊ก USB ตัวเมียอนุกรม B, และ 2. คอนเน็คเตอร์ 5 Pin ซึ่ง 2 พอร์ตนี้จะต่อขนานกัน เพื่อใช้สำหรับ Download โปรแกรมของผู้ใช้ผ่านทางพอร์ตนี้

Series "A" Connectors	Series "B" Connectors
<p>♦ Series "A" plugs are always oriented <b>upstream</b> towards the <i>Host System</i></p>  <p><b>"A" Plugs</b> (From the USB Device)</p> <p><b>"A" Receptacles</b> (Downstream Output from the USB Host or Hub)</p> 	<p>♦ Series "B" plugs are always oriented <b>downstream</b> towards the USB Device</p>  <p><b>"B" Plugs</b> (From the Host System)</p> <p><b>"B" Receptacles</b> (Upstream Input to the USB Device or Hub)</p> 

แสดงรูปร่างพอร์ต USB



แสดงตำแหน่งขาพอร์ต USB "B"



แสดงตำแหน่งขาคอนเน็คเตอร์ USB, 5 Pin



ขา	สีสายไฟ	หน้าที่
1	แดง	Vbus
2	ขาว	D-
3	เขียว	D+
4	ดำ	Ground

แสดงสีของขา พอร์ต USB

## พอร์ต 34 PIN

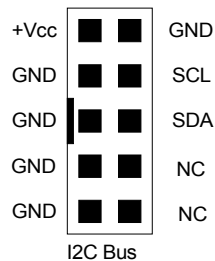
พอร์ตสื่อสาร 34 Pin เป็นพอร์ตที่รวมพอร์ต P0, P1, P2, P3(บางขา) ของ CPU เข้าไว้ด้วยกัน โดย การต่อใช้งานผู้อ่านควรใช้สายแพร 34 Pin ในการต่อไปใช้งาน ในกรณีที่ผู้อ่านซื้อบอร์ด I/O ของทางบริษัท ETT ผู้อ่านสามารถต่อสายแพร 34 ได้โดยตรง หรือ ถ้าผู้อ่านออกแบบ Hardware เอง, ผู้อ่านควรจัดให้ขา ให้อยู่ในรูปแบบของพอร์ต 34 Pin เพื่อจะได้สะดวกในการใช้งาน

P0.0	■	■	P0.1
P0.2	■	■	P0.3
P0.4	■	■	P0.5
P0.6	■	■	P0.7
P1.0	■	■	P1.1
P1.2	■	■	P1.3
P1.4	■	■	P1.5
P1.6	■	■	P1.7
P2.0	■	■	P2.1
P2.2	■	■	P2.3
P2.4	■	■	P2.5
P2.6	■	■	P2.7
+5v	■	■	NC
GND	■	■	NC
P3.2	■	■	P3.3
P3.4	■	■	P3.5
P3.6	■	■	P3.7

แสดงตำแหน่งขาคอนเน็คเตอร์ I/O พอร์ต 34 Pin

## พอร์ต I2C BUS

การสื่อสารในรูปแบบ Two Wire Interface (TWI), ที่ตำแหน่งขา P4.0 และ P4.1 ได้ต่อออกมายังคอนเน็คเตอร์ขนาด 10 Pin ซึ่งข้อต่อนี้จะใช้เพิ่มเติมจำนวนอุปกรณ์ที่ติดต่อสื่อสารแบบ I<sup>2</sup>C ให้กับบอร์ดทดลอง



แสดงตำแหน่งขาคอนเน็คเตอร์ I2C BUS, 10 Pin

## พอร์ต RS232

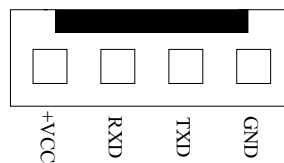
การสื่อสารอนุกรม RS232 ภายในได้บรรจุไอซี Line Driver ไว้เรียบร้อยแล้ว ซึ่งผู้อ่านสามารถต่อใช้งานได้ทันที โดย ไอซี Line Driver มีหน้าที่ในการเปลี่ยนระดับแรงดัน TTL 5V. ให้เป็นแรงดันในรูปแบบ RS232 ที่ 12 V. ผลจะทำให้ผู้อ่านสามารถต่อสายสัญญาณได้ถึง 15 เมตรเลยทีเดียวในรูปแบบ ตัวต่อตัว (Point-to-Point) เท่านั้น

สำหรับสายสัญญาณที่จะนำมาใช้สำหรับการสื่อสารแบบ RS232 นั้น จะใช้สัญญาณเพียง 2-3 เส้น เท่านั้น ทั้งนี้ขึ้นอยู่กับความต้องการในการสื่อสารว่าต้องการสื่อสารแบบทิศทางเดียวหรือสองทิศทาง

- **การสื่อสาร RS232 แบบสองทิศทาง** ซึ่งจะมีทั้งการรับข้อมูลและส่งข้อมูลไปมา ระหว่างด้านรับและด้านส่ง โดยในกรณีนี้จะต้องใช้สายสัญญาณจำนวน 3 เส้น สัญญาณรับข้อมูล (RXD) สัญญาณส่งข้อมูล(TXD) และสัญญาณอ้างอิง (GND) โดยในการเชื่อมต่อสายนั้นจะต้องทำการสลับสัญญาณกับอุปกรณ์ปลายทางด้วย คือ สัญญาณส่ง (TXD) จากบอร์ด CP-JR51USB v1.0 จะต้องต่อเข้ากับสัญญาณรับ (RXD) ของอุปกรณ์อื่น และ สัญญาณส่ง (TXD) จากอุปกรณ์อื่น ก็ต้องต่อกับสัญญาณรับ (RXD) ของบอร์ด ส่วนสัญญาณอ้างอิง (GND) จะต้องต่อตรงถึงกัน จึงจะสามารถทำการ รับ-ส่ง ข้อมูลกันได้
- **การสื่อสาร RS232 แบบทิศทางเดียว** ซึ่งอาจเป็นการรอรับข้อมูลจากด้านส่งเพียงอย่างเดียว หรือ อาจเป็นการส่งข้อมูลออกไปยังปลายทางเพียงอย่างเดียว โดยไม่มีการโต้ตอบข้อมูลซึ่งกันและกัน ซึ่งวิธีนี้จะใช้สายสัญญาณเพียง 2 เส้น เท่านั้น โดยถ้าเป็นทางด้านส่ง ก็จะต่อ

เพียงสัญญาณส่ง (TXD) และสัญญาณอ้างอิง (GND) แต่ถ้าเป็นทางด้านรับ ก็จะต้องเพียงสัญญาณรับ (RXD) และ สัญญาณอ้างอิง (GND) เท่านั้น

โดยหัวต่อของสัญญาณ RS232 ของบอร์ด CP-JR51USB v1.0 นั้น จะเป็นจุดเชื่อมต่อของสัญญาณ รับ-ส่ง ข้อมูล ที่เปลี่ยนระดับสัญญาณเป็นแบบ RS232 แล้ว ซึ่งจะมีลักษณะเป็นแบบหัว CPA ขนาด 4 PIN สำหรับใช้เป็นจุดเชื่อมต่อสัญญาณ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอก โดยมีลักษณะการจัดเรียงสัญญาณดังนี้



แสดงหัวต่อสัญญาณ RS232 ของบอร์ด CP-JR51USB v1.0

ซึ่งจะเห็นได้ว่าหัวต่อสัญญาณ RS232 ของบอร์ดนั้น จะมีทั้งหมด 4 เส้น แต่ในการ รับ-ส่ง ข้อมูลแบบปรกติ นั้น จะใช้สัญญาณเพียงแค่ 3 เส้น คือ RXD, TXD และ GND เท่านั้น ส่วน +VCC ซึ่งเป็นไฟเลี้ยงวงจร +5V นั้น จะไม่จำเป็นต้องนำมาใช้ในการสื่อสารกันแต่อย่างใด โดย +VCC หรือ +5V นี้ จะออกแบบเพื่อไว้ในกรณีที่อุปกรณ์ปลายทางเป็นวงจรขนาดเล็กและไม่สะดวกที่จะหาแหล่งจ่ายไฟให้กับอุปกรณ์ปลายทางด้วย ก็อาจต่อไฟเลี้ยงวงจร +VCC นี้ออกไปให้กับอุปกรณ์ปลายทางด้วยก็ได้เช่นกัน

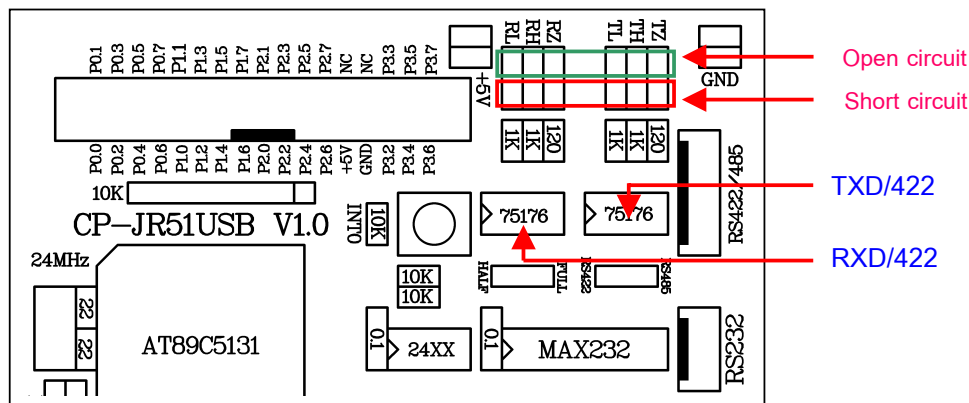
**Note :** การติดตั้งไอซี Line Driver สำหรับเลือกชนิดสัญญาณทางไฟฟ้าของการสื่อสารอนุกรมนั้นผู้อ่านสามารถต่อได้เพียงอย่างเดียวอย่างหนึ่งเท่านั้น เช่น เมื่อเลือกติดตั้งไอซี Line Driver เป็นแบบ RS232 โดยการติดตั้งไอซี MAX232 ในบอร์ดแล้ว จะต้องไม่ติดตั้งไอซี Line Driver แบบอื่น เช่น RS422 หรือ RS485 เข้าไปด้วย เพราะจะทำให้ไม่สามารถรับส่งข้อมูลกันได้อย่างถูกต้อง ผู้ใช้ต้องเลือกติดตั้งไอซี Line Driver อย่างใดอย่างหนึ่งเท่านั้น

## พอร์ต RS422/485

การสื่อสารด้วย RS422 หรือ RS485 นั้นผู้อ่านจะต้องใส่ไอซี Line Driver เพิ่มเข้าไปบนบอร์ดไมโครคอนโทรลเลอร์ และ จะต้องถอดไอซี Line Driver ของ RS232 ออกด้วย ซึ่งการใช้งาน RS422 และ RS485 นั้นอธิบายได้ดังนี้ คือ

### ● การสื่อสารอนุกรมแบบ RS422

ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 จำนวน 1-2 ตัว เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณการไฟฟ้าในการ รับ-ส่ง แบบ TTL จาก CPU ให้เป็นระดับสัญญาณแบบ Balance Line เพื่อ รับ-ส่งสัญญาณกับอุปกรณ์ที่มีระดับสัญญาณทางไฟฟ้าเป็นแบบ Balance Line เหมือนกัน โดยถ้าต้องการใช้การสื่อสารแบบ 2 ทิศทาง ก็จะต้องติดตั้งไอซี Line Driver จำนวน 2 ตัว โดยแบ่งเป็นตัวแปลงสัญญาณทางด้านรับ 1 ตัว และตัวแปลงสัญญาณด้านส่งอีก 1 ตัว แต่ถ้าต้องการสื่อสารแบบทิศทางเดียวก็อาจทำการติดตั้งไอซี Line Driver เพียงตัวเดียว โดยถ้าต้องการให้เป็นฝ่ายรับข้อมูลเพียงอย่างเดียวก็ให้ติดตั้งไอซี Line Driver เฉพาะในตำแหน่งของ “RXD/422” เพียงตัวเดียว แต่ถ้าต้องการให้เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียวก็ให้ทำการติดตั้งไอซี Line Driver เฉพาะในตำแหน่ง “TXD/485” เพียงตัวเดียวเท่านั้น ดังแสดงในรูปด้านล่าง

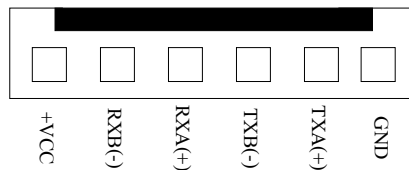


แสดงการต่อไอซี Line Driver ของ RS422

ซึ่งการสื่อสารแบบ RS422 นี้ สามารถนำไปทดแทนการสื่อสารแบบ RS232 ได้ทันที โดยไม่ต้องดัดแปลงหรือแก้ไขโปรแกรมเลย ซึ่งการสื่อสารโดยใช้ระดับสัญญาณในการ รับ-ส่ง แบบ RS422 นี้จะมีข้อดี คือ สามารถทำการสื่อสารกันได้ในระยะทางที่ไกลขึ้นกว่าแบบ RS232 มาก กล่าวคือ สามารถจะทำการ รับ-ส่ง ข้อมูลกันได้ในระยะทางประมาณ 4000 ฟุต หรือ 1200 เมตร หรือ 1.2 กิโลเมตรเลยทีเดียว เพียงแต่ต้องใช้สายสัญญาณที่ออกแบบมาสำหรับรองรับการใช้งานในด้านการสื่อสารแบบนี้โดยเฉพาะ ซึ่งได้แก่ สายสัญญาณแบบ UTP (Un-Shielded Twist Pair) หรือ STP (Shielded Twist Pair) โดยการสื่อสารด้วยระดับสัญญาณทางไฟฟ้าแบบ RS422 นี้ ถ้าเป็นการสื่อสารแบบ 2 ทิศทาง คือ ทั้งรับข้อมูลและส่งข้อมูล จะสามารถทำการรับส่งข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เหมือนกับ RS232 ทุกประการ แต่ในกรณีที่เป็นการสื่อสารแบบทิศทางเดียวนั้น สามารถจะทำการต่อขนานสัญญาณทางด้าน

รับ จำนวนหลายๆจุด เข้ากับสัญญาณส่งเพียงจุดเดียวได้ โดยถ้าเลือกใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานจำนวนอุปกรณ์สำหรับด้านรับข้อมูลได้ประมาณ 32จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 นั้น จะสามารถต่อขนานจำนวนอุปกรณ์ทางด้านรับข้อมูลได้มากถึง 256 จุด เลยทีเดียว แต่ถ้าเป็นอุปกรณ์ทางด้านส่งนั้น จะไม่สามารถนำมาต่อขนานสัญญาณส่งข้อมูลเข้าด้วยกันมากกว่า 1 จุด เหมือนทางด้านฝ่ายรับได้

สำหรับลักษณะของหัวต่อของสัญญาณ RS422 นั้น จะเป็นแบบ CPA ขนาด 6 PIN ดังรูป โดยในการสื่อสารกันนั้น จะใช้สายสัญญาณในการ รับ-ส่ง ข้อมูลกัน จำนวน 4 เส้น สัญญาณ คือ สัญญาณในการรับข้อมูล จำนวน 2 เส้น คือ RXA (RX+) และ RXB (RX-) และสัญญาณในการส่งข้อมูลอีก 2 เส้น คือ TXA (TX+) และ TXB (TX-) ซึ่งในการต่อสัญญาณนั้น จะต้องทำการต่อสัญญาณในลักษณะของการสลับกัน คือ สัญญาณส่งจะต้องต่อเข้ากับสัญญาณรับ นั่นก็คือ สัญญาณ RXA (RX+) จะต้องต่อกับ TXA (TX+) ส่วน RXB (RX-) ก็จะต้องต่อกับ TXB (TX-) ด้วยเช่นกัน โดยลักษณะของหัวต่อสัญญาณ RS422 เป็นดังรูป



แสดงหัวต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51USB V1.0 เมื่อเลือกเป็น RS422

## ● การสื่อสารอนุกรมแบบ RS485

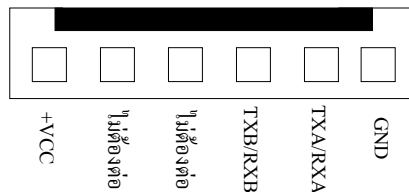
ในการสื่อสารแบบ RS485 นี้จะมีคุณสมบัติของสัญญาณทางไฟฟ้าเหมือนกับ RS422 ทุกประการ เพียงแต่ว่าในการสื่อสารแบบ RS485 นี้จะใช้สายสัญญาณในการรับส่งข้อมูลเพียง 2 เส้น เท่านั้น และจะมีความพิเศษกว่าแบบ RS422 ตรงที่ ทิศทางของสัญญาณจะสามารถปรับเปลี่ยนได้จากโปรแกรม กล่าวคือ สัญญาณทั้ง 2 เส้น นี้สามารถจะสลับหน้าที่เป็นด้านส่ง และ เป็นด้านรับได้ ตามต้องการ โดยการควบคุมจาก CPU , บอร์ด CP-JR51USB V1.0 นั้น จะกำหนดให้สัญญาณขา P3.7 ทำหน้าที่สำหรับควบคุมทิศทางของข้อมูลว่าจะให้เป็นรับหรือส่ง โดยถ้าควบคุมให้ ขา P3.7 มีสถานะเป็น “1” จะเป็นการกำหนดทิศทางให้เป็นฝ่ายส่งข้อมูล แต่ถ้าสถานะของ ขา P3.7 เป็น “0” จะเป็นการกำหนดทิศทางให้เป็นฝ่ายรับข้อมูล ซึ่งจากคุณสมบัติข้อนี้จะทำให้การสื่อสารแบบ RS485 สามารถทำการต่อขนานอุปกรณ์ร่วมกันในสายส่งเดียวกันได้จำนวนหลายๆจุด โดยถ้าใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานอุปกรณ์กันได้ จำนวน 32 จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 แล้วจะสามารถต่อขนานอุปกรณ์ในสายคู่เดียวกันได้มากถึง 256 จุด เลยทีเดียว แต่มีข้อแม้ว่า เมื่อมีการต่ออุปกรณ์ขนานกันในสายสัญญาณคู่เดียวกันมากกว่า

2 จุดแล้ว จะต้องเขียนโปรแกรมควบคุมให้มีการส่งข้อมูลออกมาในสายครั้งละ 1 จุดเท่านั้น เพราะถ้ามีการกำหนดทิศทางของข้อมูลให้เป็นส่งในเวลาเดียวกันมากกว่า 1 จุดแล้วจะทำให้เกิดการชนกันของข้อมูลและไม่สามารถสื่อสารกันได้อย่างถูกต้อง

โดยเมื่อต้องการใช้วิธีการสื่อสารแบบ RS485 นี้ จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 ในตำแหน่งของ “TXD/485” เพียงตัวเดียว พร้อมกับเลือกกำหนดเป็นแบบ RS485 ดังนี้

- ทำการเลือก Jumper สำหรับเลือก “422/485” ไว้ทางด้าน “RS485”
- ทำการเลือก Jumper “F/H” ไว้ทางด้าน “HALF” (Half Duplex)
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ “TL” ไว้
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ “TH” ไว้
- สายสัญญาณที่ใช้จะต่อจาก TXB(TX-) และ TXA(TX+) เพียง 2 เส้น ออกไปใช้งาน

ซึ่งในการสื่อสารข้อมูลแบบ RS485 นี้ จะต้องเขียนโปรแกรมขึ้นมารองรับการสื่อสารโดยเฉพาะ เนื่องจากทิศทางของข้อมูลสามารถจะกำหนดจากโปรแกรมได้โดยตรง ซึ่งการสื่อสารวิธีนี้จะมีข้อดีคือ ใช้สายสัญญาณในการรับส่งน้อยเส้น แต่จะเสียเวลาในการสื่อสารมากกว่าวิธีอื่นๆ เนื่องจากว่า การสื่อสารแบบนี้จะไม่สามารถทำการรับและส่งข้อมูลในเวลาเดียวกันได้ แต่จะต้องใช้วิธีการ ผลัดกันรับ ผลัดกันส่งแทน ซึ่งในความเป็นจริงแล้วในปัจจุบันนี้ ราคาของสายสัญญาณแบบ 2 เส้น และ 4 เส้น แทบจะไม่มี ความแตกต่างกันเลย ดังนั้นเพื่อลดความยุ่งยากในการเขียนโปรแกรมสำหรับควบคุมการรับส่งข้อมูลของ CPU ขอแนะนำให้เลือกใช้วิธีการสื่อสารแบบ RS422 จะง่ายและสะดวกรวดเร็วกว่ากันมาก



แสดงหัวต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51USB V1.0 เมื่อเลือกเป็น RS485

### การกำหนด Jumper สำหรับการสื่อสารแบบ RS422/485

เนื่องจากวงจร Line Driver ของพอร์ตสื่อสารอนุกรมของบอร์ดนั้น ออกแบบให้ผู้ใช้สามารถเลือกกำหนดได้หลายแบบ ดังนั้น จึงต้องมีการใช้ Jumper สำหรับเป็นตัวเลือกรูปแบบการสื่อสารร่วมด้วย โดยจะมี Jumper ที่เกี่ยวข้องกับการใช้งานการสื่อสารแบบ RS422 และ RS485 ดังต่อไปนี้ คือ

- Jumper 422/485 เป็น Jumper สำหรับเลือกกำหนดการทำงานของไอซี Line Driver ในตำแหน่ง TXD/485 ให้ทำงานเป็นแบบ RS422 หรือ RS485 โดยถ้าต้องการให้เป็นแบบ RS422 จะต้องกำหนด Jumper ไว้ทางด้าน “422” ซึ่งจะทำให้ไอซี Line Driver ตำแหน่ง “TXD/485” ทำหน้าที่เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียว แต่ถ้าต้องการใช้งานแบบ RS485 จะต้องกำหนด Jumper ไว้ทางด้าน “485” เพื่อกำหนดให้ไอซี Line Driver ในตำแหน่ง “TXD/485” ทำหน้าที่เป็นทั้งฝ่ายรับและฝ่ายส่ง ตามการควบคุมของสัญญาณขา P3.7
- Jumper F/H (Full/Half) เป็น Jumper ใช้สำหรับเลือกกำหนดรูปแบบการสื่อสารให้เป็นแบบ Full Duplex (F) หรือ Half Duplex (H) โดยถ้าต้องการใช้งานแบบ RS422 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้าน F(Full Duplex) แต่ถ้าต้องการใช้งานแบบ RS485 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้าน H(Half Duplex) แทน
- Jumper RL เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXB (RX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXB (RX-) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางใกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper RH เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXA (RX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXA (RX+) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางใกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ

- Jumper RZ เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน RZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านรับ ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากๆจะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทานจากภายนอกไปชดเชยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง RZ นี้ไว้ก็จะเป็นการต่อตัวต้านทานคร่อมระหว่าง RXA (RX+) และ RXB (RX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้
- Jumper TL เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXB (TX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXB (TX-) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางใกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อรวมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper TH เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXA (TX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXA (TX+) มีสถานะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางใกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อรวมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper TZ เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน TZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านส่ง ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากๆ จะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทาน



จากภายนอกไปขัดเซยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง TZ นี้ไว้ก็จะเป็นการต่อตัวต้านทานคร่อมระหว่าง TXA (TX+) และ TXB (TX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้

**Note :** จะเห็นได้ว่าวงจร Line Driver ทั้งแบบ RS422 และ RS485 นั้นจะมีความใกล้เคียง กันมาก แต่มีข้อแตกต่างอย่างหนึ่งที่เห็นได้ชัดเจนที่สุด คือ ถ้าเป็นแบบ RS422 จะไม่สามารถสลับเปลี่ยน ทิศทางการรับส่งข้อมูลด้วยโปรแกรมได้ ซึ่งทิศทางการรับส่งจะกำหนดตายตัวจากวงจร แต่ถ้าเป็นแบบ RS485 นั้น จะสามารถสลับควบคุมทิศทางการรับส่งจากโปรแกรมได้ว่าจะให้ทำหน้าที่เป็นฝ่ายรับ หรือฝ่ายส่ง อย่างใดอย่างหนึ่งได้ตามต้องการได้



## 3

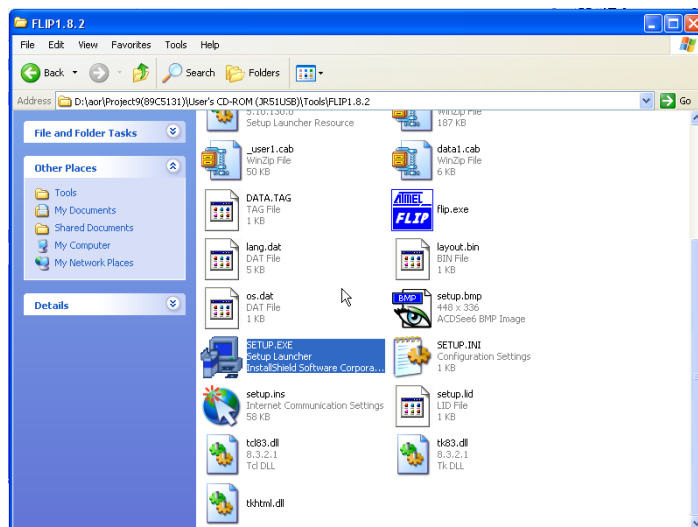
### การติดตั้งโปรแกรม FLIP 1.8.2

ในบทนี้ผู้เขียนจะอธิบายการติดตั้งโปรแกรม Flip โดยในที่นี้ผู้เขียนอ้างอิงกับเวอร์ชัน 1.8.2 ซึ่งผู้อ่านสามารถ Download เวอร์ชันที่ใหม่กว่านี้ได้ ฟรี จากทางเว็บไซต์ [www.atmel.com](http://www.atmel.com) โดยไฟล์มีขนาด 3.26 Mbytes แต่อย่างไรก็ตาม ทางบริษัท ETT ได้รวมโปรแกรม Flip เวอร์ชัน 2.2.4 ซึ่งเป็นเวอร์ชันใหม่ล่าสุดไว้ใน CD-Rom แล้วซึ่งผู้อ่านสามารถเลือกใช้เวอร์ชันไหนก็ได้

#### ระบบที่ต้องการ

- Windows 98
- Windows ME
- Windows 2000
- Windows NT
- Windows XP
- Linux

ขั้นที่ 1 . เปิดแผ่น CD-ROM ของทางบริษัท ETT แล้วเข้าไปในโฟลเดอร์ Tools => Flip1.8.2 จากนั้นดับเบิลคลิกที่ SETUP.EXE



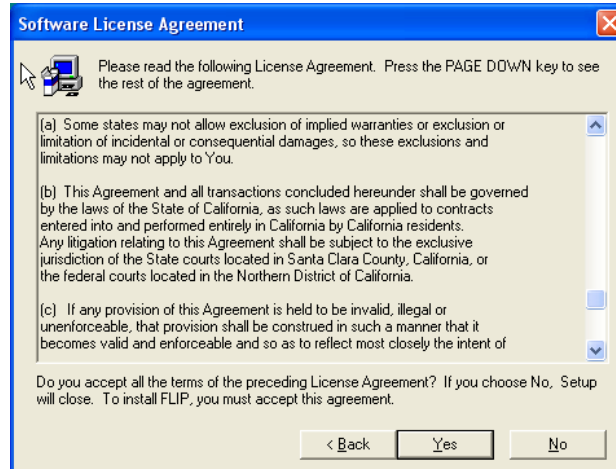
แสดงไฟล์ในโฟลเดอร์ Flip1.8.2

ขั้นที่ 2 . หลังจากนั้นผู้อ่านจะเห็นภาพดังแสดงในรูปด้านล่าง



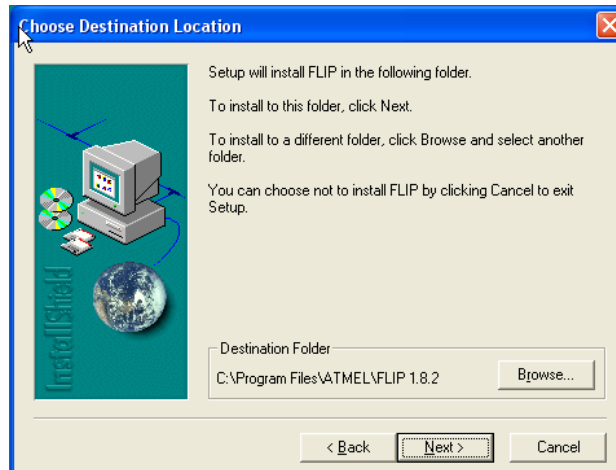
แสดงการเริ่มต้นติดตั้งโปรแกรม

ขั้นที่ 3 . หลังจากนั้นก็กดปุ่ม Next ซึ่งจะมีหน้าต่าง Software License Agreement ขึ้นมา จากนั้นให้กดปุ่ม Yes



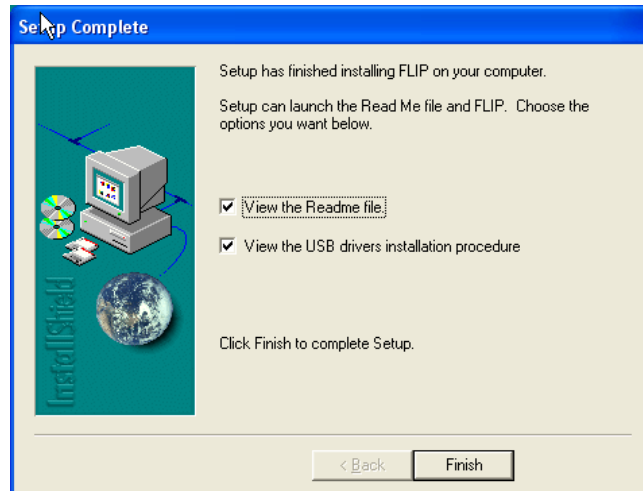
แสดงหน้าต่าง Software License Agreement

ขั้นที่ 4 . จากนั้นผู้อ่านจะเห็นหน้าต่าง Choose Destination Location ซึ่งเป็นการเลือกตำแหน่งปลายทางของการติดตั้งโปรแกรม Flip ซึ่งในที่นี้ผู้อ่านไม่เปลี่ยนแปลงค่าใดๆ จากนั้นกดปุ่ม Next



แสดงหน้าต่าง Choose Destination Location

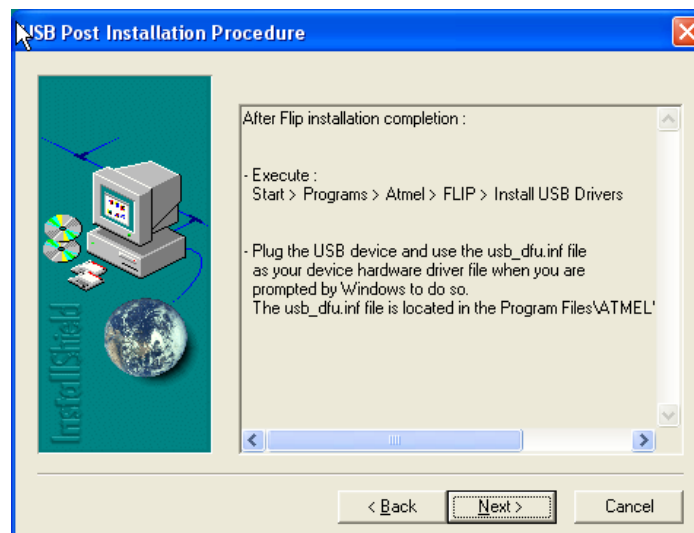
ขั้นที่ 5 . จากนั้นโปรแกรมจะทำการติดตั้ง จากนั้นกดปุ่ม Finish เป็นอันเสร็จขั้นตอนการติดตั้ง



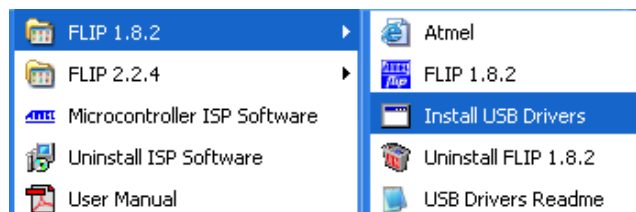
แสดงหน้าต่าง Setup Complete

ขั้นที่ 6 . จากนั้นโปรแกรมจะแสดงหน้าต่าง ReadMe File ซึ่งจะบอกระบบที่รองรับ และ คุณสมบัติเด่นหลักๆ ของตัวโปรแกรม, ให้ผู้อ่านกดปุ่ม Next

ขั้นที่ 7 . โปรแกรมจะแสดงหน้าต่าง USB Post Installation Procedure ซึ่งบอกวิธีการติดตั้ง USB Driver จากนั้นกดปุ่ม Next



ขั้นที่ 8 . ทำการติดตั้ง USB Driver โดย ไปที่ “ Start => Programs => Atmel => Flip1.8.2 =>Install USB Driver “ เท่านั้นเป็นอันเสร็จพิธี





## 4

## การใช้งานโปรแกรม FLIP 1.8.2 เบื้องต้น

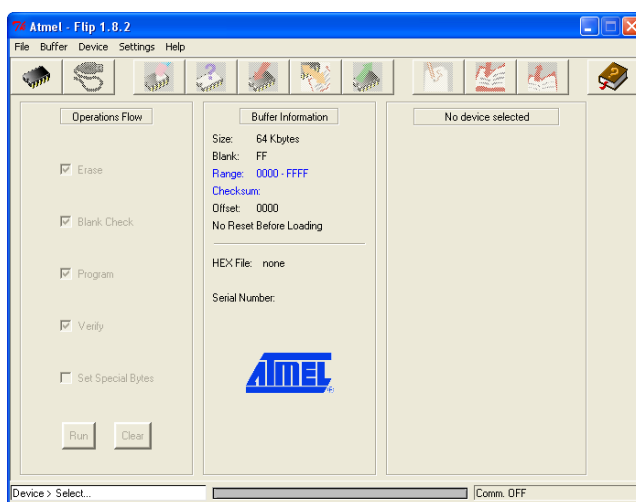
หลังจากที่ผู้อ่านทำการติดตั้งโปรแกรม Flip1.8.2 เป็นที่เรียบร้อยแล้ว ในบทนี้ผู้เขียนจะนำเสนอวิธีการใช้งานโปรแกรม Flip ในการดาวน์โหลดโปรแกรมที่ผู้อ่านได้สร้างขึ้นผ่านทางพอร์ต USB ซึ่งมีขั้นตอนดังนี้ คือ

ขั้นที่ 1 : ต่อสายดาวน์โหลด USB เข้าที่คอมพิวเตอร์ และ ปลายอีกข้างหนึ่งต่อเข้ากับพอร์ต USB บนบอร์ด CP-JR51USB v1.0

ขั้นที่ 2 : ต่อแหล่งจ่ายไฟ 9 โวลท์ให้กับบอร์ด CP-JR51USB v1.0

ขั้นที่ 3 : เลื่อนสวิตช์ USB ON/OFF มาที่ตำแหน่ง USB ON ซึ่งจะส่งผลให้ LED สีเขียวที่ตำแหน่งข้างๆ สวิตช์นั้นติดขึ้นมา

ขั้นที่ 4 : เปิดโปรแกรม Flip 1.8.2 โดยดับเบิลคลิกจาก Icon บน Desktop



แสดงหน้าต่างโปรแกรม Flip 1.8.2


ขั้นที่ 5 : เลือกเบอร์อุปกรณ์ โดยไปที่เมนู Device => Select... จากนั้นกดปุ่ม OK

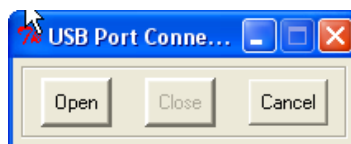


แสดงหน้าต่าง Device Selection

ขั้นที่ 6 : เข้าสู่โหมดดาวน์โหลด ( Monitor mode ) โดยกดสวิทช์บนบอร์ด ดังนี้

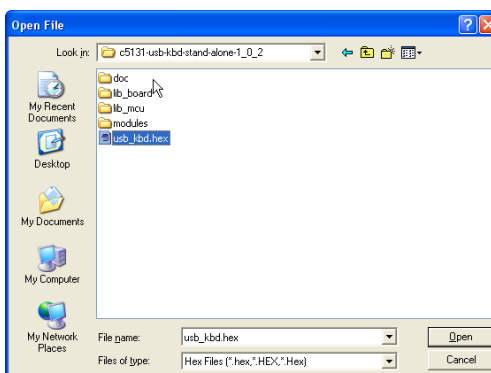
- ◆ กดสวิทช์ PSEN ค้างไว้
- ◆ กดสวิทช์ RESET แล้วปล่อย
- ◆ ปล่อยสวิทช์ PSEN
- ◆ เลื่อนสวิทช์ USB ON/OFF ไปที่ตำแหน่ง USB OFF แล้ว เลื่อนกลับมาที่ตำแหน่ง USB ON

ขั้นที่ 7 : ไปที่เมนู Settings => Communication => USB หรือ  , จากนั้นกดปุ่ม Open



แสดงหน้าต่าง USB Port Connection

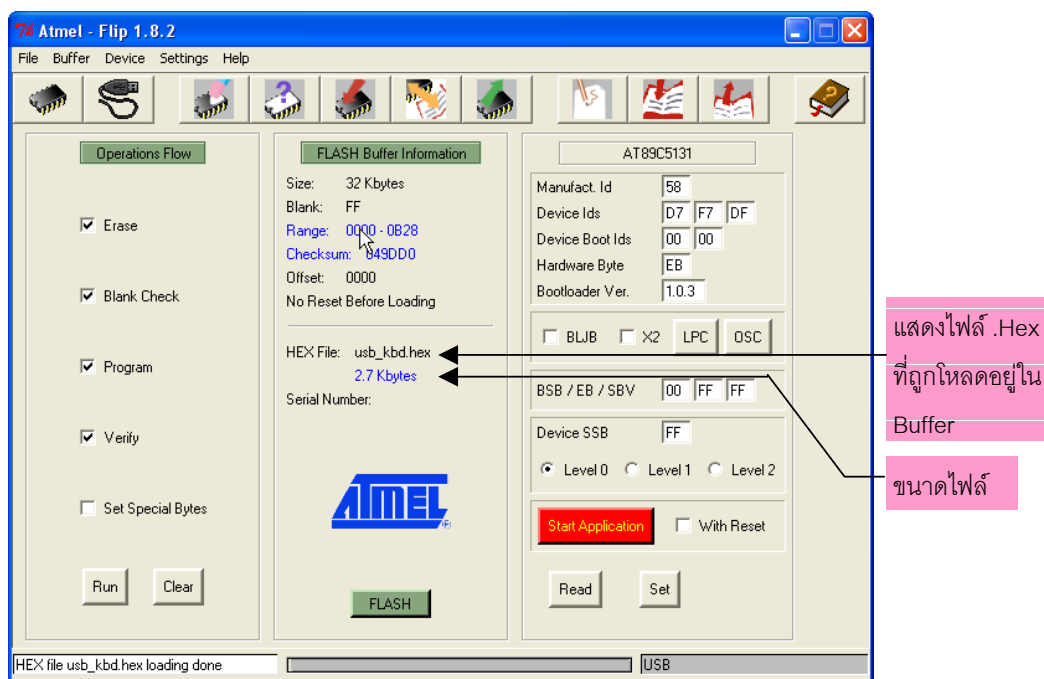
ขั้นที่ 8 : จากนั้นในเมนู File เลือก Load Hex .. และ เลือกไฟล์โปรแกรมที่มีนามสกุล “ . Hex ” ชื่อ usb\_kbd.hex ซึ่งอยู่ในแผ่น CD-ROM ที่แถมไปกับบอร์ดทดลอง โดยจะอยู่ที่ “ \Source Examples\HID Keyboard demonstration software\c5131-usb-kbd-stand-alone-1\_0\_2 “



แสดงการเปิดไฟล์ . Hex



ขั้นที่ 9 : จากนั้นหน้าจอโปรแกรม Flip จะมีลักษณะดังนี้ คือ



แสดงหน้าต่างโปรแกรม Flip

ขั้นที่ 10 : แนใจว่าผู้อ่านได้ทำการ Check Boxes ในส่วนของ Operations Flow Section ดังนี้ คือ

- ☒ Erase
- ☒ Blank Check
- ☒ Program
- ☒ Verify

ขั้นที่ 11 : แนใจว่าบล็อกร BLJB นั้นไม่ถูก Check Boxes ( รายละเอียดของแต่ละส่วนผู้อ่านสามารถดูได้จาก ภาคผนวก ง. )

ขั้นที่ 12 : จากนั้นกดปุ่ม RUN

ขั้นที่ 13 : จากกดปุ่ม Start Application โปรแกรมจะเริ่มทำงาน เท่านั้นเป็นอันเสร็จพิธี

**Note :** ในกรณีที่ผู้อ่านทำการแก้ไขโปรแกรม และ ต้องการโหลดโปรแกรมลงบอร์ดใหม่ (ครั้งถัดไป), ผู้อ่านจะต้องทำขั้นตอนที่ 6 ถึง ขั้นตอนที่13 ใหม่ทุกครั้ง โดยเฉพาะขั้นตอนที่ 6 กับ 7 ไม่ควรสลับกัน ซึ่งในความเป็นจริงขั้นตอนที่ 7 นั้นจะมีหน้าต่าง USB Port Connection ขึ้นอยู่แล้วก็ตาม

ขั้นที่ 14 : จากนั้นให้ผู้อ่านเปิดโปรแกรม Notepad ขึ้นมา

ขั้นที่ 15 : จากนั้นกดสวิทช์ INT0 บนบอร์ด CP-JR51USB v1.0 ผู้อ่านควรเห็นข้อความขึ้นในหน้าต่างโปรแกรม Notepad นั้น

จากที่ได้กล่าวมาทั้งหมดนี้เป็นวิธีการใช้โปรแกรม Flip1.8.2 ในขั้นต้นเท่านั้น ซึ่งผู้อ่านสามารถดูรายละเอียดการใช้งานเพิ่มเติมได้จากภาคผนวก ง, ในส่วนโปรแกรมตัวอย่างที่ผู้เขียนได้นำเสนอไปนั้น เป็นการใช้งานการสื่อสารผ่านทางระบบบัส USB โดยใช้ Class HID ซึ่งโปรแกรมนี้เป็นโปรแกรมตัวอย่างจากบริษัท ATMEL ที่เขียนด้วยภาษา C และ ในการใช้ไฟล์นี้ผู้อ่านจะต้องใช้ไฟล์ในแผ่น CD-ROM ที่แถมไปกับบอร์ด เท่านั้นเนื่องจากถ้าผู้อ่านดาวน์โหลดโปรแกรมนี้จากเว็บไซต์ ของ Atmel จะไม่สามารถรันโปรแกรมได้ เนื่องจากโปรแกรมในเว็บไซต์ Atmel จะรองรับ Crystal ที่ 16 MHz แต่ในบอร์ดของบริษัท ETT จะใช้ที่ 24 MHz ซึ่งผู้อ่านจะต้องเข้าไปแก้ไขในไฟล์ config.h ในส่วนของการกำหนดค่า Crystal และ จะต้อง Compile โปรแกรมใหม่ด้วย ซึ่งโปรแกรม Keil ในเวอร์ชันทดลองใช้ที่แจกฟรีนั้นไม่สามารถ Compile ได้เนื่องจากขนาดไฟล์มีขนาดเกิน 2kBytes

## ภาคผนวก ก.

### SFRs MAP & RESET VALUEs

# คู่มือการใช้งานบอร์ด CP-JR51USB v1.0

ETT

The table below shows all SFRs with their address and their reset value.

**Table 13. SFR Descriptions**

	Bit Addressable	Non-Bit Addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	UEPINT 0000 0000	CH 0000 0000	CCAP0H XXXX XXXX	CCAP1H XXXX XXXX	CCAP2H XXXX XXXX	CCAP3H XXXX XXXX	CCAP4H XXXX XXXX		FFh
F0h	B 0000 0000	LEDCON 0000 0000							F7h
E8h		CL 0000 0000	CCAP0L XXXX XXXX	CCAP1L XXXX XXXX	CCAP2L XXXX XXXX	CCAP3L XXXX XXXX	CCAP4L XXXX XXXX		EFh
E0h	ACC 0000 0000		UBYCTLX 0000 0000	UBYCTHX 0000 0000					E7h
D8h	CCON 00X0 0000	CMOD 00XX X000	CCAPM0 X000 0000	CCAPM1 X000 0000	CCAPM2 X000 0000	CCAPM3 X000 0000	CCAPM4 X000 0000		DFh
D0h	PSW 0000 0000	FCON (1) XXXX 0000	EECON 0000 0000		UEPCONX 1000 0000	UEPRST 0000 0000			D7h
C8h	T2CON 0000 0000	T2MOD XXXX XX00	RCAP2L 0000 0000	RCAP2H 0000 0000	TL2 0000 0000	TH2 0000 0000	UEPSTAX 0000 0000	UEPDATX 0000 0000	CFh
C0h	P4 XXXX 1111		UEPIEN 0000 0000	SPCON 0001 0100	SPSTA 0000 0000	SPDAT XXXX XXXX	USBADDR 0000 0000	UEPNUM 0000 0000	C7h
B8h	IPL0 X000 000	SADEN 0000 0000	UFNUML 0000 0000	UFNUMH 0000 0000	USBCON 0000 0000	USBINT 0000 0000	USBIEIN 0000 0000		BFh
B0h	P3 1111 1111	IEN1 XXXX X000	IPL1 XXXX X000	IPH1 XXXX X000				IPH0 X000 0000	B7h
A8h	IEN0 0000 0000	SADDR 0000 0000						CKCON1 0000 0000	AFh
A0h	P2 1111 1111		AUXR1 XXXX X0X0	PLLCON XXXX XX00	PLLDIV 0000 0000		WDRST XXXX XXXX	WDTPRG XXXX X000	A7h
98h	SCON 0000 0000	SBUF XXXX XXXX	BRL 0000 0000	BDRCON XXX0 0000	KBLS 0000 0000	KBE 0000 0000	KBF 0000 0000		9Fh
90h	P1 1111 1111			SSCON 0000 0000	SSCS 1111 1000	SSDAT 1111 1111	SSADR 1111 1110		97h
88h	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000	AUXR XX0X 0000	CKCON0 0000 0000	8Fh
80h	P0 1111 1111	SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 00X1 0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Note: 1. FCON access is reserved for the Flash API and ISP software.

 Reserved

วิธีการดูตำแหน่งแอดเดรสในตารางของรีจิสเตอร์ SFRs ต่างๆ นั้น การจัดวางจะได้เรียงลำดับจากด้านล่างซ้ายมือไปทางขวามือ แล้วขึ้นบนไปเรื่อยๆ โดยช่องซ้ายมือสุด และ ช่องทางขวามือสุด จะแสดงตำแหน่งแอดเดรสของรีจิสเตอร์ SFRs เช่น P0 ตรงกับตำแหน่งแอดเดรส 80H, SP ตรงกับตำแหน่งแอดเดรส 81H, PCON ตรงกับตำแหน่งแอดเดรส 87H เป็นต้น ส่วนตัวเลขด้านล่างชื่อรีจิสเตอร์ต่างๆ จะเป็นค่าเริ่มต้นเมื่อ CPU รีเซ็ต

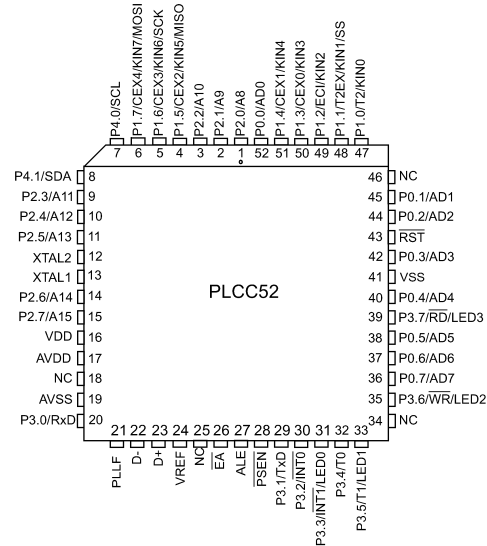
## ภาคผนวก ข.

### PINOUT & BLOCK DIAGRAM

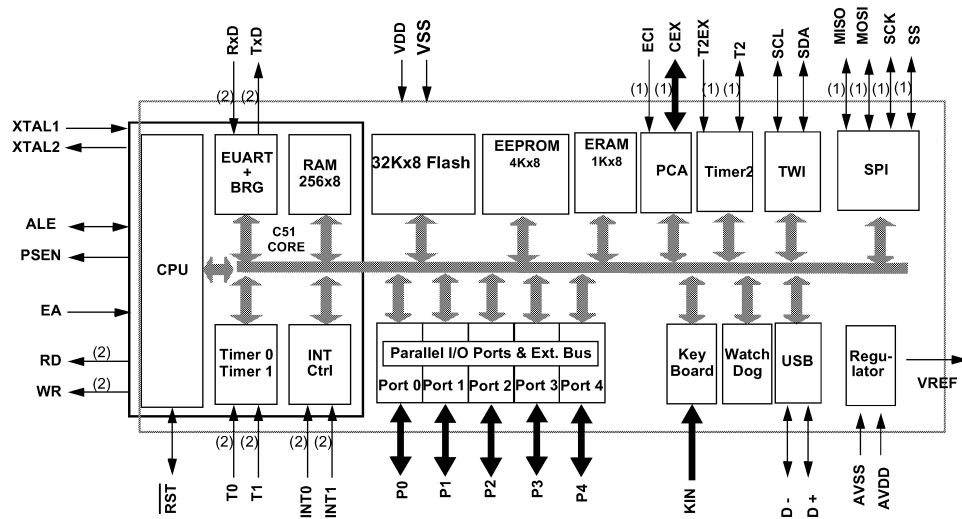
## Pinout Description

### Pinout

Figure 1. AT89C5131 52-pin PLCC Pinout



## Block Diagram



- Notes:
1. Alternate function of Port 1
  2. Alternate function of Port 3
  3. Alternate function of Port 4

## ภาคผนวก ค.

CP-JR51USB V1.0 SCHEMATICs.

ETT





## ETT

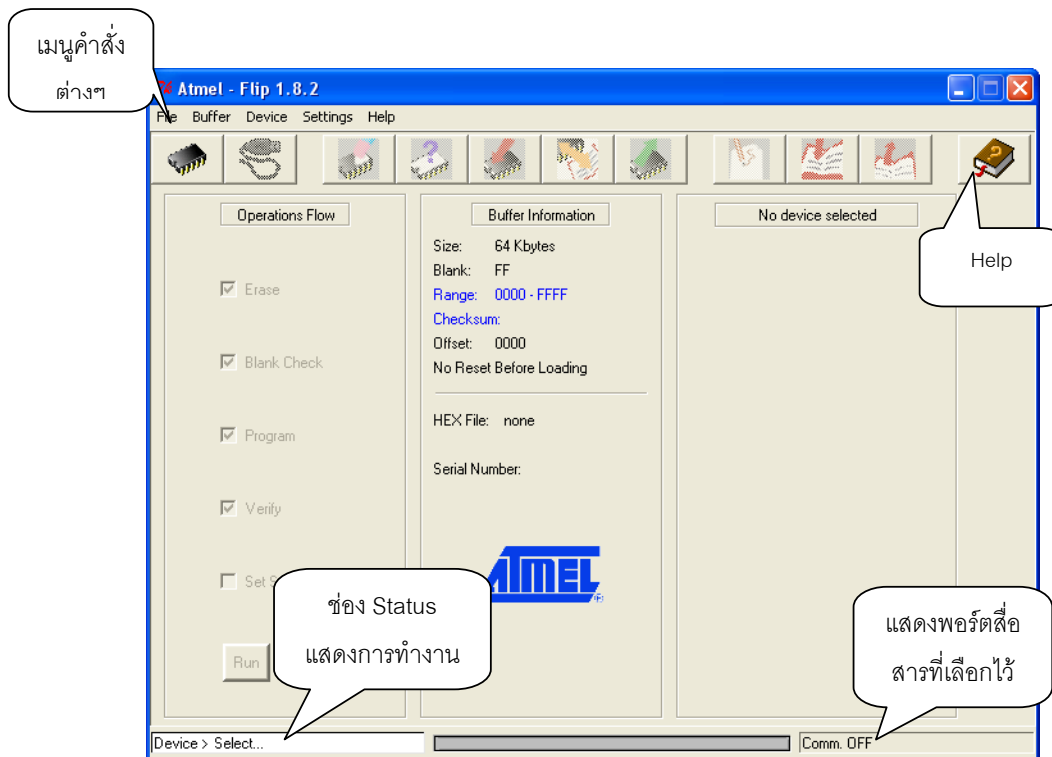


## ภาคผนวก ง.

โปรแกรม Flip1.8.2

## การใช้งานโปรแกรม FLIP (Flexible In-system Programmer)

โปรแกรม FLIP (Flexible In-system Programmer) เป็นโปรแกรมที่ใช้สำหรับดาวน์โหลดไฟล์โปรแกรมที่มีนามสกุล “.Hex “ ให้กับหน่วยความจำ Flash memory ที่อยู่ภายในตัว CPU ของบริษัท ATMEL โดยสามารถใช้สนับสนุนการพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS51 ในกลุ่มที่ใช้การพัฒนาแบบ ISP ซึ่งรวมถึงเบอร์ AT89C5131 ด้วย โดยโปรแกรมจะทำงานภายใต้ระบบปฏิบัติการของ Windows98/Me/NT/2000 และ Windows XP โดยสนับสนุนการเชื่อมต่อกับระบบฮาร์ดแวร์ที่ใช้การเชื่อมต่อแบบ RS232 หรือ CAN หรือ USB ซึ่งวิธีการเชื่อมต่อของโปรแกรม FLIP กับระบบฮาร์ดแวร์ของไมโครคอนโทรลเลอร์นั้น จะขึ้นอยู่กับความสามารถของตัวไมโครคอนโทรลเลอร์ที่จะนำมาพัฒนาว่าสามารถใช้การติดต่อสื่อสารด้วยวิธีใดได้บ้าง แต่สำหรับไมโครคอนโทรลเลอร์เบอร์ AT89C5131 นั้นจะสามารถใช้การเชื่อมต่อผ่านทางพอร์ต USB ได้ โดยทำงานใน Monitor Mode ( โหมดดาวน์โหลด ) เพื่อให้ผู้ใช้สั่งจัดการกับหน่วยความจำภายในตัว CPU ไม่ว่าจะเป็นการ ล้างข้อมูล(Erase) สังเกตตรวจสอบข้อมูลในหน่วยความจำ(Blank Check) ส่งโปรแกรมข้อมูลให้กับหน่วยความจำโปรแกรมของ CPU (Program) ส่งเปรียบเทียบข้อมูลจาก Buffer กับหน่วยความจำในตัว CPU (Verify) หรือส่งอ่านข้อมูลจากหน่วยความจำของ CPU (Read) เป็นต้น



รูปแสดง ลักษณะหน้าต่างโปรแกรมของ FLIP Version 1.8.2

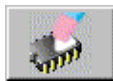
การทำงานของปุ่มคำสั่งต่างๆในโปรแกรม FLIP



Select Device ใช้สำหรับเลือกกำหนดเบอร์ CPU ที่จะใช้งานร่วมกับโปรแกรม FLIP ( กด F2 )



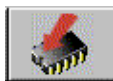
Select Communication ใช้สำหรับกำหนดรูปแบบการเชื่อมต่อโปรแกรม FLIP กับ CPU



Erase Device ใช้สำหรับล้างข้อมูลจากหน่วยความจำของ CPU



Blank Check Device ใช้สำหรับตรวจสอบว่าข้อมูลในหน่วยความจำของ CPU ว่างอยู่หรือไม่



Program Device ใช้สำหรับส่ง Program ข้อมูลให้กับหน่วยความจำของ CPU



Verify Device ใช้สำหรับส่งเปรียบเทียบข้อมูลในหน่วยความจำของ CPU กับข้อมูลที่อยู่ใน Buffer ของโปรแกรม FLIP ในเครื่องคอมพิวเตอร์ PC ว่ามีค่าตรงกันทั้งหมดหรือไม่



Read Device ใช้สำหรับส่งอ่านข้อมูลจากหน่วยความจำของ CPU ไปเก็บไว้ใน Buffer



Edit Buffer ใช้สำหรับส่งแก้ไขข้อมูลใน Buffer



Load HEX File ใช้สำหรับส่งโหลดข้อมูลแบบ HEX File มาเก็บไว้ใน Buffer



Save HEX File ใช้สำหรับส่งบันทึกข้อมูลใน Buffer เป็น HEX File เก็บไว้



Help ใช้สำหรับเรียกไฟล์ Help สำหรับดูรายละเอียดการใช้งานโปรแกรม ( กดปุ่ม F1 )

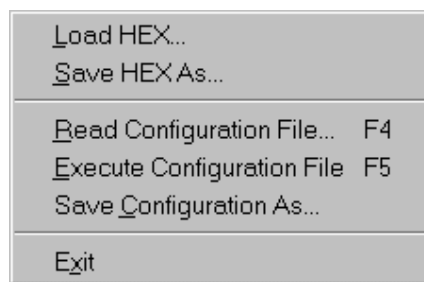
*Note* : คำว่า Buffer ในที่นี้ หมายถึง พื้นที่สำหรับใช้พักข้อมูลชั่วคราวของโปรแกรม FLIP ซึ่งโปรแกรม FLIP จะใช้พื้นที่ส่วนนี้ในการพักข้อมูลและใช้เป็นตัวกลางในการทำงานของโปรแกรม เช่น Load HEX File ข้อมูลใน HEX File ก็จะถูกพักรอไว้ใน Buffer เมื่อสั่ง Program Device โปรแกรม FLIP จะนำข้อมูลใน Buffer นี้เขียนเข้าไปยังหน่วยความจำของ CPU เป็นต้น

## คำสั่งต่างๆ ของเมนู

สำหรับการเรียกใช้งานคำสั่งต่างๆของโปรแกรม FLIP อีกวิธีหนึ่งคือการเรียกใช้งานผ่านทางเมนูคำสั่ง โดยเมื่อต้องการใช้งานคำสั่งใดก็ให้ทำการคลิกเมาส์ไปยังเมนูคำสั่งนั้นแล้วเลื่อนเมาส์ให้แถบสว่างไปปรากฏอยู่เหนือคำสั่งที่ต้องการแล้วคลิกเมาส์โปรแกรม FLIP ก็จะทำงานตามคำสั่งนั้นๆทันที ซึ่งโปรแกรม FLIP จะจัดกลุ่มหรือหมวดหมู่ของเมนูคำสั่งต่างๆไว้ด้วยกัน 5 กลุ่มดังนี้คือ

### [ เมนู File ]

ใช้สำหรับสั่งจัดการเกี่ยวกับไฟล์ ไม่ว่าจะเป็นการสั่งเปิด HEX ไฟล์ การสั่งบันทึก HEX ไฟล์ การสั่งอ่าน Configuration File การสั่งบันทึก Configuration File หรือการปิดโปรแกรมเป็นต้น โดยลักษณะเมนูคำสั่งของ File จะเป็นดังรูป



แสดง คำสั่งในเมนู File

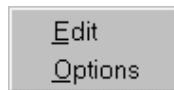
- **Load HEX** ใช้สำหรับสั่งอ่านข้อมูลแบบ Intel HEX มาพักรอไว้ใน Buffer ของโปรแกรม FLIP เพื่อรอการสั่ง Program Device
- **Save HEX As..** ใช้สำหรับสั่งบันทึกข้อมูลใน Buffer เก็บเป็น File แบบ Intel HEX ไว้
- **Read Configuration File** ใช้สำหรับสั่งอ่าน Configuration File เพื่อกำหนดคุณสมบัติการทำงานของโปรแกรม FLIP เหมือนกับที่กำหนดไว้ใน Configuration File ซึ่งวิธีการนี้จะช่วยอำนวยความสะดวกต่อการพัฒนาโปรแกรมมาก ในกรณีที่มีการเรียกใช้งานโปรแกรม FLIP เพื่อพัฒนาโปรแกรมให้กับบอร์ด ซึ่งในบางครั้งอาจจำเป็นต้องกระทำซ้ำๆ กันหลายครั้งกว่าจะใช้งานได้ ซึ่งการกระทำดังกล่าวตามปกติแล้วมักจะใช้เงื่อนไขหรือข้อกำหนดต่างๆ ที่เหมือนกันด้วย ไม่ว่าจะเป็น การกำหนดเบอร์ CPU การกำหนดรูปแบบการติดต่อสื่อสาร หรือชื่อ HEX ไฟล์ ที่จะนำมาใช้เป็นต้น ซึ่งโปรแกรม FLIP สามารถให้ผู้ใช้ทำการบันทึกคุณสมบัติต่างๆ เหล่านี้ไว้ในรูปแบบของ Configuration File ซึ่งเมื่อทำการเปิดโปรแกรม FLIP ขึ้นมาใช้งานในครั้ง

ต่อไปก็ไม่จำเป็นต้องเสียเวลาไปกำหนดตัวเลือกหรือเงื่อนไขต่างๆ ให้กับโปรแกรมให้ยุ่งยาก สามารถสั่งอ่าน Configuration File ที่สั่งบันทึกหรือสร้างไว้ก่อนหน้านี้ขึ้นมา ซึ่งโปรแกรม FLIP จะทำการกำหนดคุณสมบัติและเงื่อนไขการทำงานต่างๆ ของโปรแกรมให้เหมือนกับที่บันทึกไว้ใน Configuration File ทั้งหมดในทันที ซึ่งจะช่วยประหยัดเวลาและลดความยุ่งยากไปได้มากเลยทีเดียว

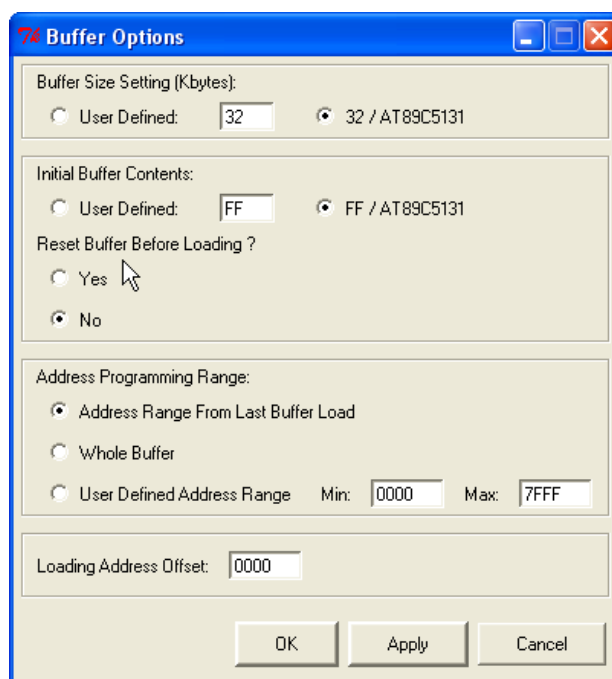
- **Execute Configuration File** ใช้สำหรับสั่งให้โปรแกรม FLIP ทำงานตามคำสั่งที่บันทึกไว้แล้วใน Configuration File ที่ถูกสั่งอ่านออกมาครั้งสุดท้าย(ปัจจุบัน)
- **Save Configuration As...** ใช้สำหรับสั่งบันทึกค่าตัวเลือกและคุณสมบัติต่างๆ ที่กำหนดให้กับโปรแกรม FLIP ไว้ในปัจจุบัน ไม่ว่าจะเป็นการกำหนดเบอร์ CPU การกำหนดการสื่อสาร หรือการสั่งเปิดไฟล์ HEX สำหรับนำมาโปรแกรมให้กับ CPU เป็นต้น โดยคุณสมบัติต่างๆ ทั้งหมดจะถูกบันทึกเป็น Configuration File ไว้ให้ตามชื่อที่กำหนดตอนสั่งบันทึก
- **EXIT** ใช้สำหรับปิดโปรแกรม FLIP หรือจบการทำงานของโปรแกรม FLIP

#### [ เมนู Buffer ]

ใช้สำหรับสั่งจัดการเกี่ยวกับ Buffer ในโปรแกรม Flip ซึ่งมีเพียง 2 คำสั่งเท่านั้นดังแสดงในรูปด้านล่าง คือ



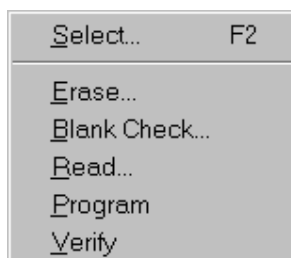
- **Edit** ใช้สำหรับสั่งแก้ไขข้อมูลใน Buffer ซึ่งเมื่อเรียกใช้คำสั่งนี้จะปรากฏหน้าต่างการทำงานของ Buffer พร้อมทั้งการแสดงค่าของข้อมูลในตำแหน่งแอดเดรสต่างๆ ใน Buffer ให้ทราบ ซึ่งผู้ใช้สามารถเข้าไปทำการแก้ไขค่าข้อมูลในตำแหน่งแอดเดรสต่างๆ ได้ตามต้องการ
- **Options** ใช้สำหรับกำหนดคุณสมบัติของ Buffer ซึ่งเมื่อเรียกใช้งานคำสั่งนี้จะปรากฏหน้าต่างสำหรับให้กำหนดค่า Option ต่างๆ ของ Buffer ซึ่งในกรณีที่ใช้งานกับบอร์ด CP-JR51USB ตัวเลือกต่างๆ แสดงดังรูปด้านล่าง คือ



แสดง คำสั่ง Options ในเมนู Buffer

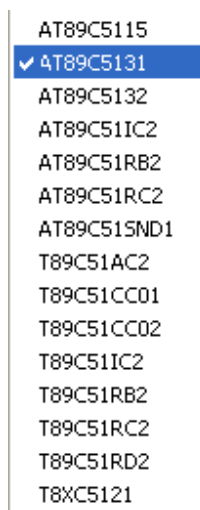
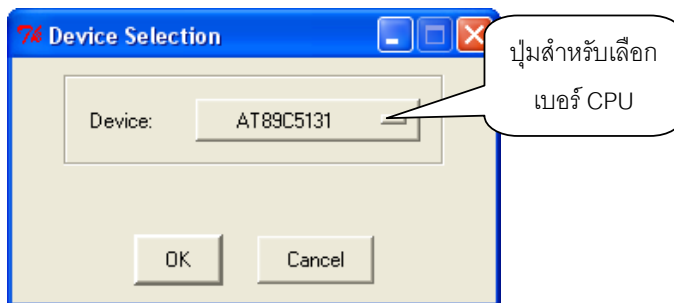
## [ เมนู Device ]

ใช้สำหรับสั่งจัดการกับ CPU ไม่ว่าจะเป็นการกำหนด หรือ เลือกเบอร์ CPU รวมทั้งการสั่งจัดการหน่วยความจำของ CPU ไม่ว่าจะเป็นการล้าง (Erase) สั่งตรวจสอบข้อมูลในตัว CPU (Blank Check) สั่งอ่านข้อมูลภายในหน่วยความจำของ CPU มาไว้ใน Buffer (Read) สั่งเขียนข้อมูลจาก Buffer ให้กับหน่วยความจำภายในตัว CPU (Program) หรือสั่งเปรียบเทียบข้อมูลในตัว CPU กับข้อมูลใน Buffer (Verify) เป็นต้น



แสดง คำสั่งในเมนู Device

- **Select** ใช้สำหรับกำหนดเบอร์ CPU ที่จะนำมาใช้กับโปรแกรม FLIP ซึ่งเมื่อเรียกใช้คำสั่งนี้จะปรากฏหน้าต่างสำหรับเลือกเบอร์ของ CPU ให้เห็นพร้อมกับแสดงเบอร์ของ CPU ที่กำหนดไว้แล้วในช่อง Device ซึ่งถ้าเบอร์ที่ต้องการถูกต้องแล้วให้กดปุ่ม OK แต่ถ้ายังไม่ตรงกับความต้องการให้คลิกเมาส์ที่ปุ่มเบอร์ของ CPU ที่แสดงในรูปด้านล่าง ซึ่งโปรแกรมจะแสดงเมนูสำหรับเลือกเบอร์ขึ้นมาให้เลือก ซึ่งในขั้นตอนนี้ให้ทำการเลื่อนเมาส์ให้แถบสว่างเลื่อนไปยังเบอร์ CPU ที่ต้องการแล้วคลิกเมาส์เพื่อเลือกเบอร์ดังรูป



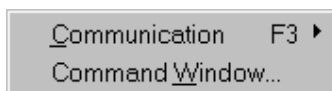
รูปแสดง การเลือกกำหนดเบอร์ CPU สำหรับใช้กับบอร์ด CP-JR51USB v1.0

- **Erase** ใช้สำหรับล้างข้อมูลจากหน่วยความจำของ CPU
- **Blank Check** ใช้สำหรับสั่งตรวจสอบหน่วยความจำโปรแกรมของ CPU ว่ามีข้อมูลอยู่หรือไม่
- **Read** ใช้สำหรับสั่งอ่านข้อมูลจากหน่วยความจำของ CPU มาพักรอไว้ใน Buffer
- **Program** ใช้สำหรับสั่งโปรแกรมข้อมูลใน Buffer ให้กับหน่วยความจำของ CPU
- **Verify** ใช้สำหรับสั่งเปรียบเทียบข้อมูลใน Buffer และ หน่วยความจำของ CPU ว่าตรงกันหรือไม่

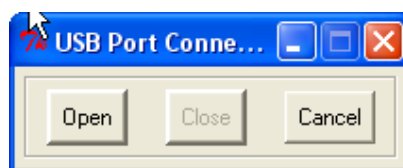


## [ เมนู Setting ]

ใช้สำหรับกำหนดรูปแบบการเชื่อมต่อของโปรแกรม FLIP กับ CPU



- **Communication** ใช้สำหรับกำหนดรูปแบบการสื่อสารของโปรแกรม FLIP กับ CPU ซึ่งในกรณีที่ใช้กับบอร์ด CP-JR51USB v1.0 นั้นจะสามารถเลือกกำหนดรูปแบบการสื่อสารแบบ USB ได้

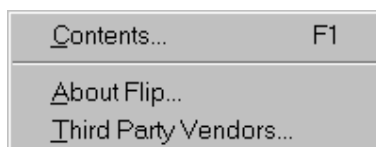


รูปแสดง การใช้งานพอร์ต USB

- **Command Window** ใช้สำหรับสั่งเปิดหน้าต่างการแสดงผลการทำงานของคำสั่งต่างๆที่สั่งงานผ่านเมนูคำสั่งหรือปุ่มคำสั่งต่างๆ ซึ่งไม่มีความจำเป็นต้องใช้งาน

## [ เมนู Help ]

ใช้สำหรับเรียก Help File เพื่อดูคำอธิบายการใช้งานโปรแกรม

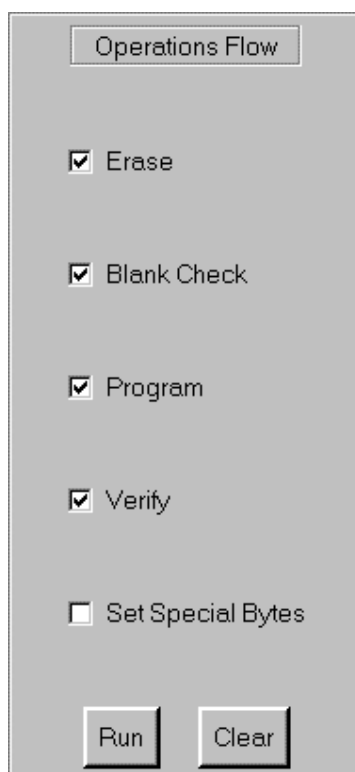


- **Contents** เป็นการเรียกดูรายละเอียดการใช้งานโปรแกรม FLIP โดยสามารถเลือกหัวข้อที่ต้องการจะดูรายละเอียดต่างๆได้ตามต้องการ ซึ่งเมื่อเลือกหัวข้อนี้จะเกิดหน้าต่าง Help พร้อมกับแสดงหัวข้อต่างๆให้เลือกตามต้องการ
- **About Flip** ใช้สำหรับแสดง รุ่น หรือ Version ของโปรแกรม FLIP
- **Third Party Vendor** ใช้สำหรับแสดงรายชื่อบริษัทที่ร่วมกันพัฒนา Driver เพื่อใช้งานร่วมกับโปรแกรม FLIP

## การสั่งงานโปรแกรม FLIP แบบอัตโนมัติ

สำหรับในกรณีที่ต้องการสั่งให้โปรแกรม FLIP จัดการโปรแกรมข้อมูลให้กับหน่วยความจำของ CPU นั้น ถ้าใช้วิธีการสั่งงานทีละคำสั่ง อาจเกิดความล่าช้า และ ต้องเสียเวลาคลิกเมาส์เพื่อสั่งงานแต่ละคำสั่งหลายๆครั้ง ซึ่งโปรแกรม FLIP ได้ออกแบบคำสั่ง Operation Flow เพื่อให้สามารถเลือกกำหนดการทำงานของคำสั่งต่างๆตามลำดับขั้นและขั้นตอน ซึ่งตามปกติแล้วเมื่อจะสั่งโปรแกรมข้อมูล ให้กับ CPU จะต้องสั่งงาน CPU เป็นลำดับขั้นคือ

- Erase เพื่อลบข้อมูลเดิมที่อยู่ในตัว CPU ออกเสียก่อน
- Blank Check เพื่อตรวจสอบค่าข้อมูลในหน่วยความจำว่าลบหมดแล้วหรือยัง
- Program เพื่อเขียนข้อมูลจาก Buffer ไปยังหน่วยความจำของ CPU
- Verify เพื่อตรวจสอบดูว่าข้อมูลที่เขียนให้กับหน่วยความจำของ CPU นั้นถูกต้องเหมือนใน Buffer หรือไม่



แสดง ลักษณะของหน้าต่าง Operations Flow สำหรับสั่งงานโปรแกรม FLIP แบบต่อเนื่อง