

# คู่มือการใช้งาน User's manual CP-JR ARM7 LPC2368

## ARM7

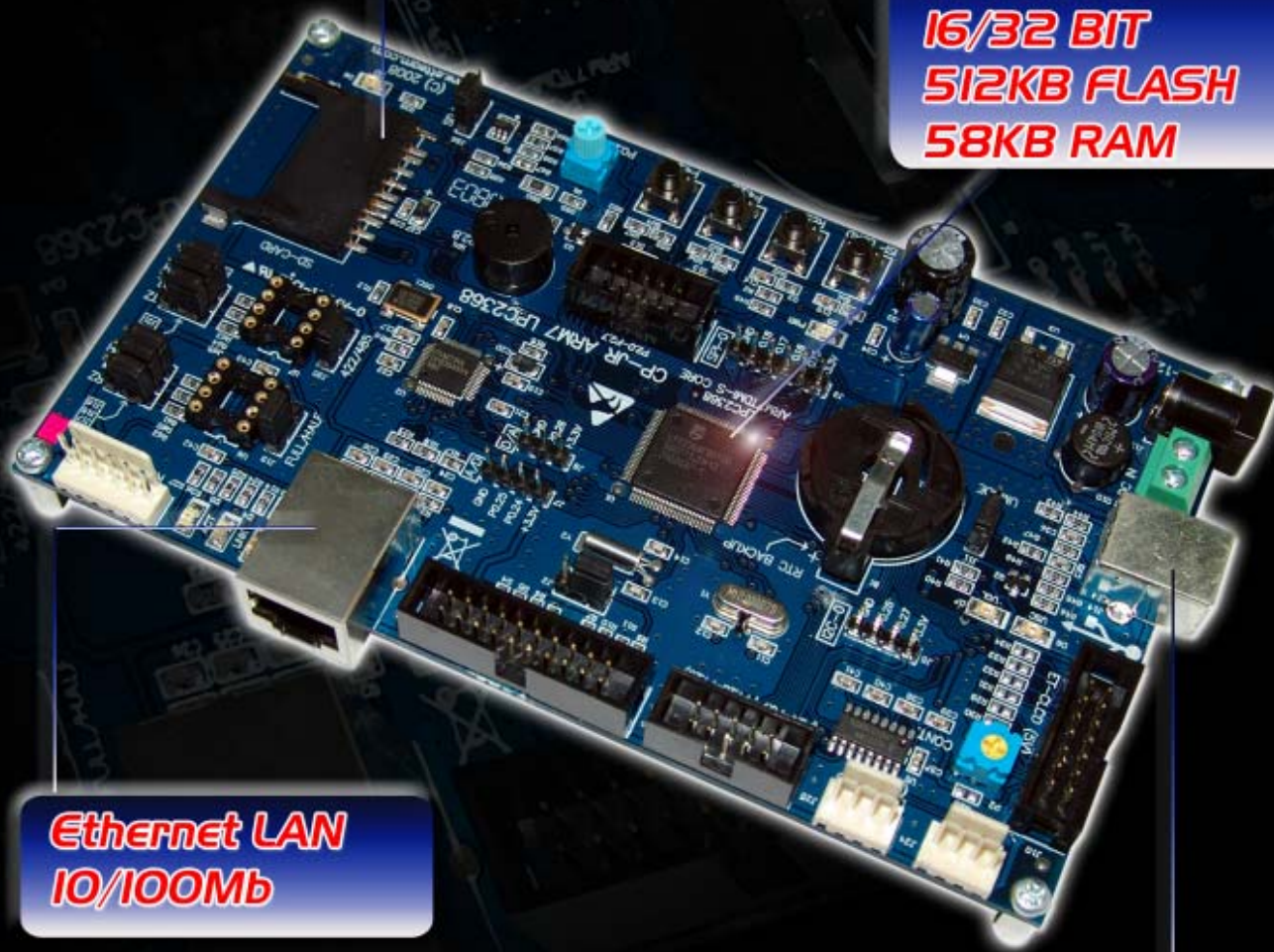


**SD/MMC Card  
Connecotor**

**LPC2368  
16/32 BIT  
512KB FLASH  
58KB RAM**

**Ethernet LAN  
10/100Mb**

**USB 2.0  
Full Speed**



## CP-JR ARM7 LPC2368

CP-JR ARM7 LPC2368 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล ARM7TDMI-S Core ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์ 16/32-Bit ขนาด 100 Pin(LQFP) แบบใช้พลังงานต่ำเป็น MCU ประจำบอร์ด ซึ่งบอร์ดนี้เลือกใช้ MCU เบอร์ LPC2368 ของ Philips (NXP) โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องของการจัดวางอุปกรณ์พื้นฐานที่จำเป็นต่อการนำไปประยุกต์ใช้งาน และ ศึกษาทดลอง ชั้นพื้นฐานรวมไว้อย่างครบถ้วน เช่น LED แสดงสถานะของ Output Logic และ Push Button Switch สำหรับสร้างสัญญาณ Logic เพื่อทดสอบการทำงานของ Input หรือ Volume ปรับค่าแรงดัน เพื่อใช้ทดสอบการทำงานของ A/D รวมถึงวงจรขับเสียงโดยใช้ Mini-Speaker สำหรับสร้างเสียง Beep ต่างๆ เป็นต้น

นอกจากวงจรชั้นพื้นฐานดังกล่าวข้างต้นแล้ว บอร์ด CP-JR ARM7 LPC2368 ยังได้ออกแบบวงจรสำหรับประยุกต์ใช้งานขั้นสูงจัดเตรียมไว้ให้ใช้งานด้วย เช่น

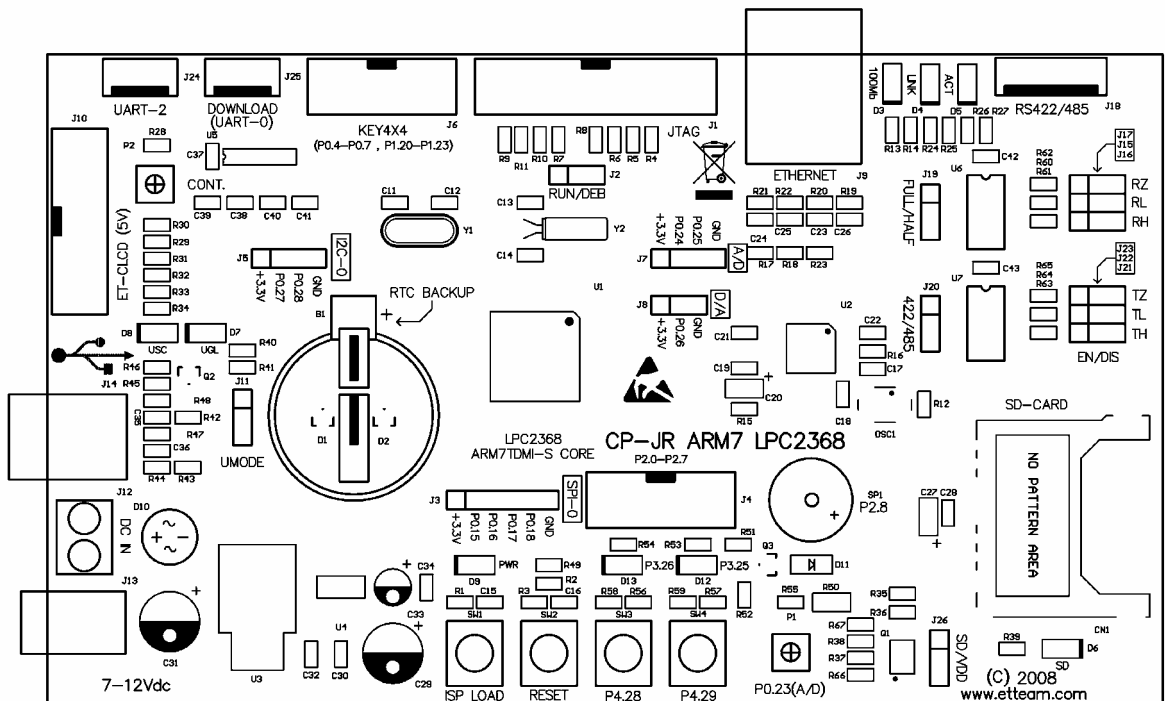
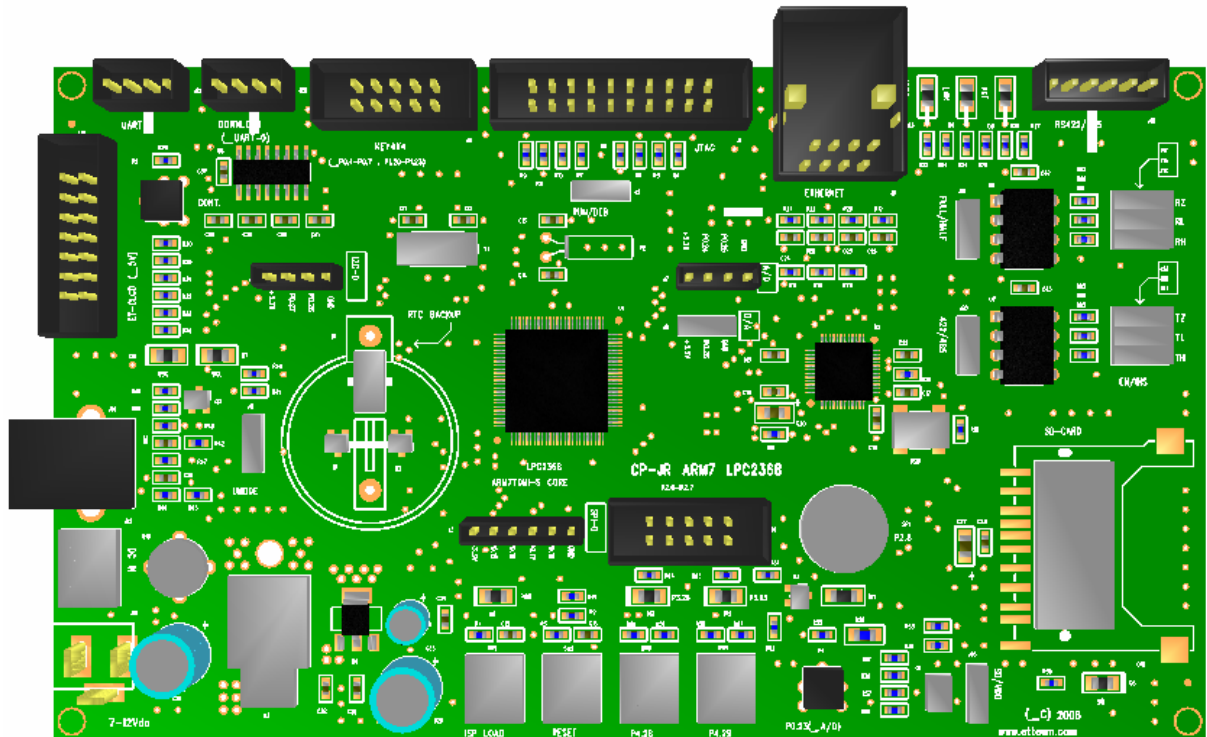
- วงจรเชื่อมต่อกับ USB ซึ่งรองรับการเชื่อมต่อกับ USB 2.0 ได้ ซึ่งผู้ใช้งานสามารถนำบอร์ดไปพัฒนาเป็น USB Device แบบต่างๆได้โดยสะดวก
- วงจรเชื่อมต่อกับการ์ดหน่วยความจำ ซึ่งสามารถใช้งานได้กับการ์ดหน่วยความจำแบบ SD Card และ การ์ดหน่วยความจำแบบ MMC Card
- วงจรเชื่อมต่อ Ethernet LAN แบบ 10/100Mb สามารถเชื่อมต่อกับเครือข่าย LAN แบบ Ethernet ได้ทั้งระบบ 10Mb และ 100Mb
- วงจรเชื่อมต่อกับ Dot-Matrix LCD แบบ Character พร้อมวงจรปรับความสว่าง
- วงจรสื่อสารข้อมูลแบบ RS232 พร้อม Line Driver จำนวน 2 ช่อง
- วงจรสื่อสารข้อมูลแบบ RS422/485 ทั้งแบบ Half-Duplex และ Full-Duplex

นอกเหนือจากนี้แล้วยังมี GPIO ต่างที่วางไว้ให้ผู้ใช้ออกแบบใช้งานร่วมกับอุปกรณ์อื่นๆได้เองตามความเหมาะสม สำหรับวิธีการพัฒนาโปรแกรมของบอร์ดนั้นก็มีความอ่อนตัวเป็นอย่างมาก กล่าวคือสามารถใช้การพัฒนาแบบ ISP Download ผ่านทางพอร์ตสื่อสารอนุกรม RS232 และการพัฒนาแบบ JTAG โดยใช้งานร่วมกับ ARM JTAG มาตรฐาน สำหรับ Download และ Debug ได้โดยง่าย ซึ่งจะเห็นได้ว่า โครงสร้างโดยรวมของบอร์ด CP-JR ARM7 LPC2368 จะมีความหลากหลาย และครบถ้วนพอสมควร เหมาะที่จะใช้เป็นบอร์ดทดลองเรียนรู้ และนำไปประยุกต์ดัดแปลงสร้างเป็น Application ใช้งานในด้านต่างๆได้มากมาย

## คุณสมบัติของบอร์ด

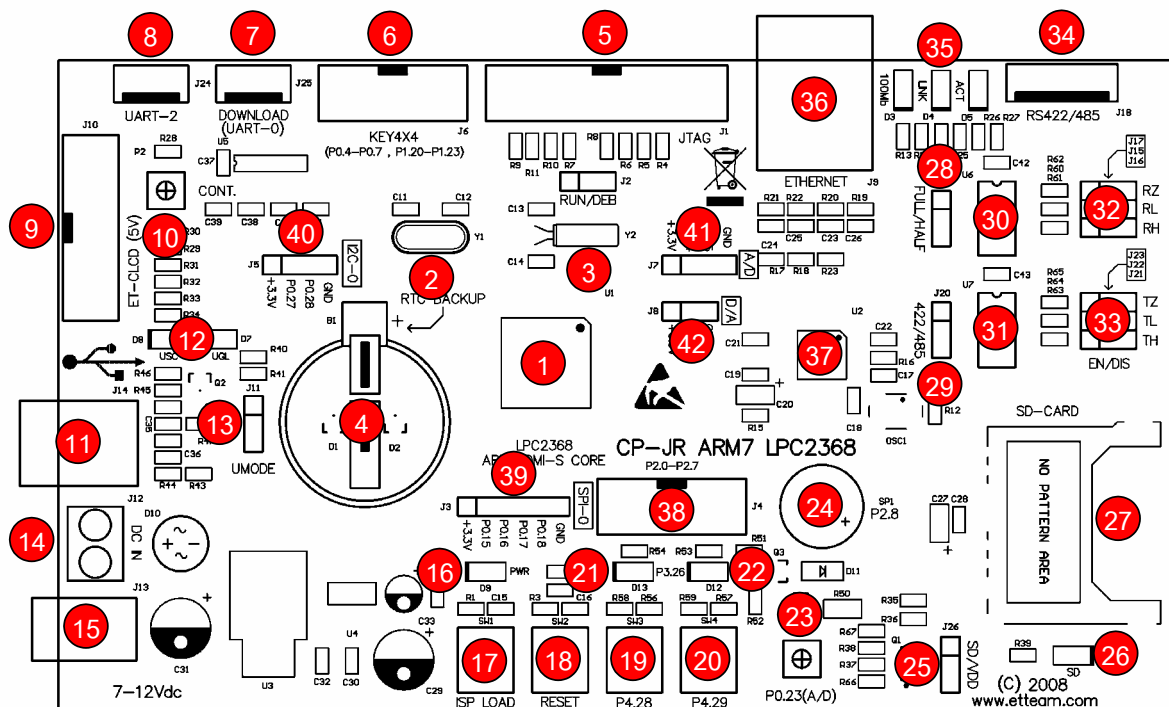
1. ใช้ MCU ตระกูล ARM7TDMI-S เบอร์ LPC2368 ของ Philips(NXP) ซึ่งเป็น MCU ขนาด 16/32Bit
2. ภายใน MCU มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 512KB, Static RAM ขนาด 58KB
3. ใช้ Crystal 12.00 MHz โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 72 MHz เมื่อใช้งานร่วมกับ Phase-Locked Loop (PLL) ภายในตัว MCU เอง
4. มีวงจร RTC(Real Time Clock) พร้อม XTAL ค่า 32.768KHz และ Battery Backup
5. รองรับการโปรแกรมแบบ In-System Programming (ISP) และ In-Application Programming (IAP) ผ่านทาง On-Chip Boot-Loader Software ทางพอร์ต UART-0 (RS232)
6. มีวงจรเชื่อมต่อกับ JTAG ARM ขนาด 20 Pin มาตรฐาน เพื่อทำการ Debug แบบ Real Time ได้
7. Power Supply ใช้แรงดันไฟฟ้า 7-12 VAC/DC โดยใช้ขั้วต่อแบบ Terminal และ DC-Jack พร้อมวงจร Bridge Rectifier และ Regulate +5V/800mA และ +3V3/3A
8. มีวงจร USB มาตรฐาน 2.0 แบบ Full Speed ภายในตัว (USB Function มี 32 End Point)
9. มีวงจรเชื่อมต่อ Ethernet LAN 10/100Mb โดยใช้ขั้วต่อแบบ RJ45 มาตรฐาน จำนวน 1 ช่อง
10. มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card หรือ MMC Card จำนวน 1 ช่อง
11. มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง
12. มีวงจรสื่อสารอนุกรม RS422/485 โดยใช้ขั้วต่อแบบ 6-PIN มาตรฐาน ETT จำนวน 1 ช่อง
13. มีวงจรเชื่อมต่อ Dot-Matrix LCD พร้อมวงจรปรับความสว่าง ใช้ขั้วต่อ 14 Pin มาตรฐาน ETT
14. มีวงจร Push Button Switch จำนวน 3 ชุด พร้อมสวิตช์ RESET
15. มีวงจร LED แสดงสถานะเพื่อทดลอง Output จำนวน 2 ชุด
16. มีวงจร สร้างแรงดัน 0-3V3 โดยใช้ตัวต้านทานปรับค่าได้สำหรับทดสอบ A/D จำนวน 1 ชุด
17. มีวงจรถูกานีตและขับเสียง Beep โดยใช้ Mini Speaker จำนวน 1 ชุด
18. มี 25 Bit GPIO อิสระ สำหรับประยุกต์ต่างๆ เช่น A/D,D/A,I2C,SPI และ Input / Output
  - Header 10Pin IDE (P2[0..7]) สำหรับ GPIO หรือ Full-Duplex Serial UART
  - Header 10Pin IDE (P0[4..7],P1[20..23]) สำหรับ GPIO หรือ Matrix Key ขนาด 4x4
  - 3 Pin Header(P0[26]) สำหรับ GPIO หรือ D/A
  - 4 Pin Header(P0[24..25]) สำหรับ GPIO หรือ A/D
  - 4 Pin Header(P0[27..28]) สำหรับ GPIO หรือ I2C Bus
  - 6 Pin Header(P0[15..18]) สำหรับ GPIO หรือ SPI Bus

## โครงสร้างบอร์ด CP-JR ARM7 LPC2368



รูปแสดง โครงสร้างของบอร์ด CP-JR ARM7 LPC2368





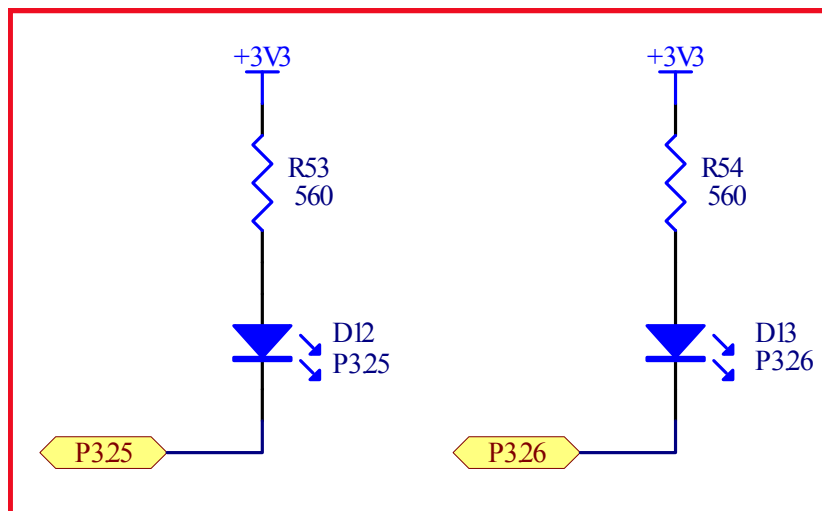
รูปแสดง ตำแหน่งของอุปกรณ์ต่างๆในบอร์ด CP-JR ARM7 LPC2368

- หมายเลข 1 คือ MCU เบอร์ LPC2368 (100Pin LQFP)
- หมายเลข 2 คือ Crystal ค่า 12 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU
- หมายเลข 3 คือ Crystal ค่า 32.768KHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU
- หมายเลข 4 คือ Battery ขนาด 3V สำหรับ Backup ค่าให้กับระบบ RTC
- หมายเลข 5 คือ ขั้วต่อ JTAG ARM สำหรับ Debug แบบ Real Time
- หมายเลข 6 คือ ขั้วต่อ GPIO(P0[4..7],P1[20..23]) สำหรับ Keyboard Matrix 4x4 หรือ GPIO
- หมายเลข 7 คือ ขั้วต่อ UART-0(RS232) สำหรับใช้งาน และ Download Hex File ให้ CPU
- หมายเลข 8 คือ ขั้วต่อ UART-2(RS232) สำหรับใช้งาน
- หมายเลข 9 คือ ขั้วต่อ Character LCD โดยใช้กับ LCD แบบ +5V Supply
- หมายเลข 10 คือ VR สำหรับปรับค่าความสว่างให้ Character LCD
- หมายเลข 11 คือ ขั้วต่อ USB สำหรับเชื่อมต่อกับ USB Hub รุ่น 2.0
- หมายเลข 12 คือ LED แสดงค่าสถานะของการทำงานและการเชื่อมต่อของ USB
- หมายเลข 13 คือ Jumper สำหรับเลือกโหมดการทำงานของ USB
- หมายเลข 14 และ 15 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ 7-12V AC/DC
- หมายเลข 16 คือ LED แสดงสถานะของ Power +VDD(+3V3)
- หมายเลข 17 คือ SW1 เป็น ISP LOAD หรือ P2.10/EINT0

- หมายเลข 18 คือ SW2 หรือ สวิตช์ RESET
- หมายเลข 19 และ 20 คือ SW3 และ SW4 ใช้ทดสอบ Logic Input ของ P4.28 และ P4.29
- หมายเลข 21 และ 22 คือ LED ใช้ทดสอบ Logic Output ของ P3.25 และ P3.26
- หมายเลข 23 คือ VR สำหรับปรับค่าแรงดัน 0-3V3 สำหรับทดสอบ A/D(P0.23/AD0.0)
- หมายเลข 24 คือ Mini Speaker สำหรับใช้กำเนิดเสียงความถี่ต่างๆ
- หมายเลข 25 คือ Jumper สำหรับเลือกแหล่งจ่ายไฟให้กับการ์ดหน่วยความจำ SD/MMC
- หมายเลข 26 คือ LED แสดงสถานะของ แหล่งจ่ายไฟของการ์ดหน่วยความจำ SD/MMC
- หมายเลข 27 คือ ช่องเสียบการ์ดหน่วยความจำสามารถใช้ได้กับ SD Card และ MMC Card
- หมายเลข 28 และ 29 คือ Jumper สำหรับเลือกกำหนดการทำงานของ RS422/485
- หมายเลข 30 คือ IC Line Driver ของ RS422 Receive ใช้ได้กับ 75176 หรือ MAX3088
- หมายเลข 31 คือ IC Line Driver ของ RS422 Transmit และ RS485 Transceiver สามารถใช้ได้กับเบอร์ 75176 หรือ MAX3088
- หมายเลข 32 และ 33 คือ Jumper สำหรับเลือก Enable/Disable Fail-Save Resistor และ Terminate Resistance ของ RS422 Receive, RS422 Transmit และ RS485 Transceiver
- หมายเลข 34 คือ ขั้วต่อสัญญาณ RS422/485
- หมายเลข 35 คือ LED แสดงสถานะของ Ethernet LAN
- หมายเลข 36 คือ ขั้วต่อสัญญาณ Ethernet LAN แบบ RJ45
- หมายเลข 37 คือ IC Physical Ethernet Driver เบอร์ DP83848
- หมายเลข 38 คือ ขั้วต่อ GPIO(P2[0..7] สำหรับ Full-Duplex UART1 หรือ GPIO
- หมายเลข 39 คือ ขั้วต่อ SPI0-Bus หรือ GPIO P0[15..18]
- หมายเลข 40 คือ ขั้วต่อ I2C0-Bus หรือ GPIO P0[27..28]
- หมายเลข 41 คือ ขั้วต่อ A/D หรือ GPIO P0[24..25]
- หมายเลข 42 คือ ขั้วต่อ D/A หรือ GPIO P0.26

## การใช้งานวงจรขับ LED แสดงผล

LED แสดงผลของบอร์ด จะต่อวงจรแบบรับกระแส (Sink Current) โดยใช้กับแหล่งจ่าย +3.3V ทำงานด้วยลอจิก “0” (0V) และหยุดทำงานด้วยลอจิก “1” (+3.3V) โดยควบคุมการทำงานจาก GPIO มีทั้งหมด 2 ชุด คือ P3[25] และ P3[26] โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Output



โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ P3[25] และ P3[26] ทำหน้าที่เป็น Output Port เสียก่อน แล้วจึงควบคุม Logic ให้กับ P3[25] และ P3[26] ตามต้องการ ดังตัวอย่าง

```
// Config Pin GPIO = P3[26:25] Drive LED
PINSEL7  &= 0xFFC3FFFF;      // Config P3[26:25] = GPIO Function
PINMODE7 &= 0xFFC3FFFF;      // Enable Pull-Up on P3[26:25]
FIO3DIR  |= 0x02000000;      // Set P3[25] = Output
FIO3DIR  |= 0x04000000;      // Set P3[26] = Output

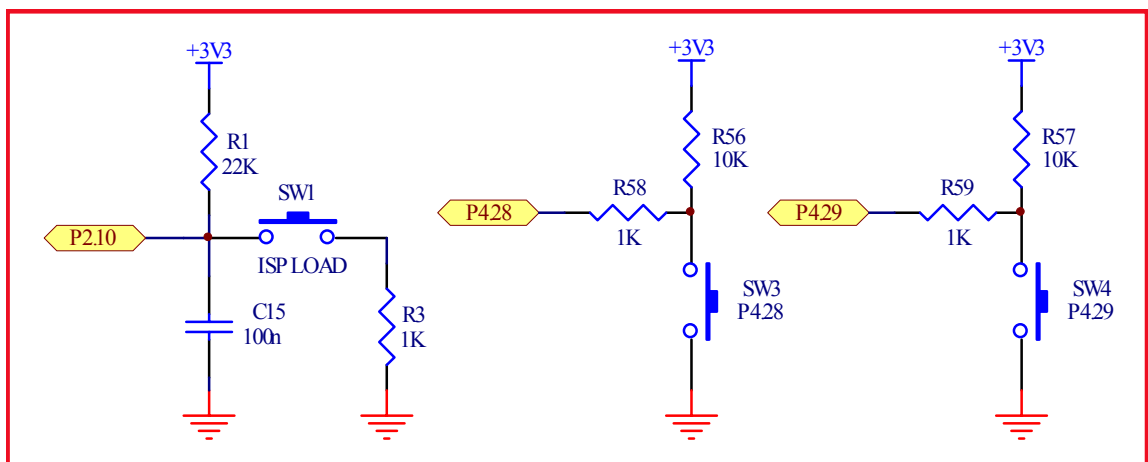
FIO3CLR  = 0x02000000;      // LED(P3[25]) = ON
FIO3CLR  = 0x04000000;      // LED(P3[26]) = ON

FIO3SET  = 0x02000000;      // LED(P3[25]) = OFF
FIO3SET  = 0x04000000;      // LED(P3[26]) = OFF
```

## การใช้งานวงจร Push Button Switch

วงจร Push Button Switch จะใช้วงจร Switch แบบ กดติด-ปล่อยดับ (Push Button) พร้อมวงจร Pull-Up ใช้กับแหล่งจ่าย +3.3V โดยในขณะที่สวิตช์ยังไม่ถูกกดจะให้ค่าสถานะเป็นลอจิก “1” แต่เมื่อสวิตช์ถูกกดอยู่จะให้สถานะเป็นลอจิก “0” ใช้สำหรับทดสอบการทำงานของ Input Logic โดยวงจรส่วนนี้จะมียูต์ด้วยกัน 3 ชุด คือ

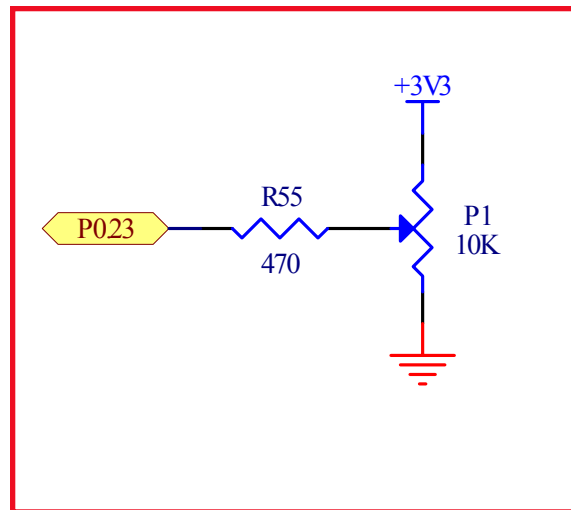
- SW1 (P2[10]) สำหรับ ISP Download และทดสอบ Input หรือ Interrupt(EINT0)
- SW3 (P4[28]) สำหรับ ทดสอบ Logic Input
- SW4 (P4[29]) สำหรับ ทดสอบ Logic Input





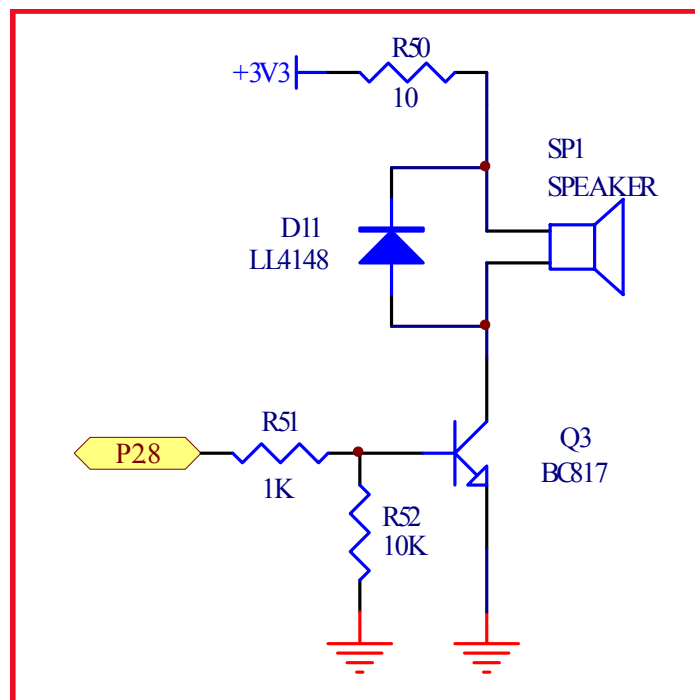
## การใช้งานวงจรปรับแรงดัน (0-3V3)

วงจรปรับแรงดันจะใช้ตัวต้านทานปรับค่าได้แบบเกือกม้า ชนิดมีแกนหมุนสำหรับปรับค่า โดยวงจรนี้ใช้กับแหล่งจ่าย +3.3V โดยจะให้ Output เป็นแรงดันซึ่งมีค่าระหว่าง 0V ถึง +3.3V ตามการปรับค่าของตัวต้านทาน จำนวน 1 ชุด โดย Output ที่ได้จะป้อนให้กับขาสัญญาณ P0[23] สำหรับใช้สร้างแรงดัน Input เพื่อทดสอบการทำงานของวงจร A/D (P0[23])



## การใช้งาน วงจรกำเนิดเสียง

วงจรกำเนิดเสียง จะใช้ลำโพงขนาดเล็ก (Mini Speaker) พร้อมด้วยวงจรทรานซิสเตอร์แบบ NPN สำหรับขับกระแสให้กับลำโพง ใช้กับแหล่งจ่ายขนาด +3.3V ทำงานด้วยลอจิก “1” และหยุดทำงานด้วยลอจิก “0” โดยในการทำงานนั้นต้องส่งสัญญาณลอจิกที่เป็นความถี่ต่างๆให้กับลำโพงเพื่อสร้างเป็นความถี่เสียงย่านต่างๆ ตามต้องการ โดยใช้การควบคุมจาก P2[8]



โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ P2[8] ทำหน้าที่เป็น Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ P2[8] ON/OFF เป็นความถี่ ตามต้องการดังตัวอย่าง

```
// Config Pin GPIO = P2[8] Drive Mini Speaker Generate Beep
PINSEL4  &= 0xFFFFCFFF;           // Config P2[8] = GPIO Function
FIO2DIR  |= 0x00000100;           // Config P2[8] = Output

// Loop Generate Beep on Speaker(P2.8)
while(1)                               // Loop Continue
{
    for (i = 0; i < 500; i++)           // Start Beep Pulse
    {
        FIO2SET  = 0x00000100;         // P2[8] = "1" (ON Speaker)
        delay(5000);
        FIO2CLR  = 0x00000100;         // P2[8] = "0" (OFF Speaker)
        delay(5000);
    }

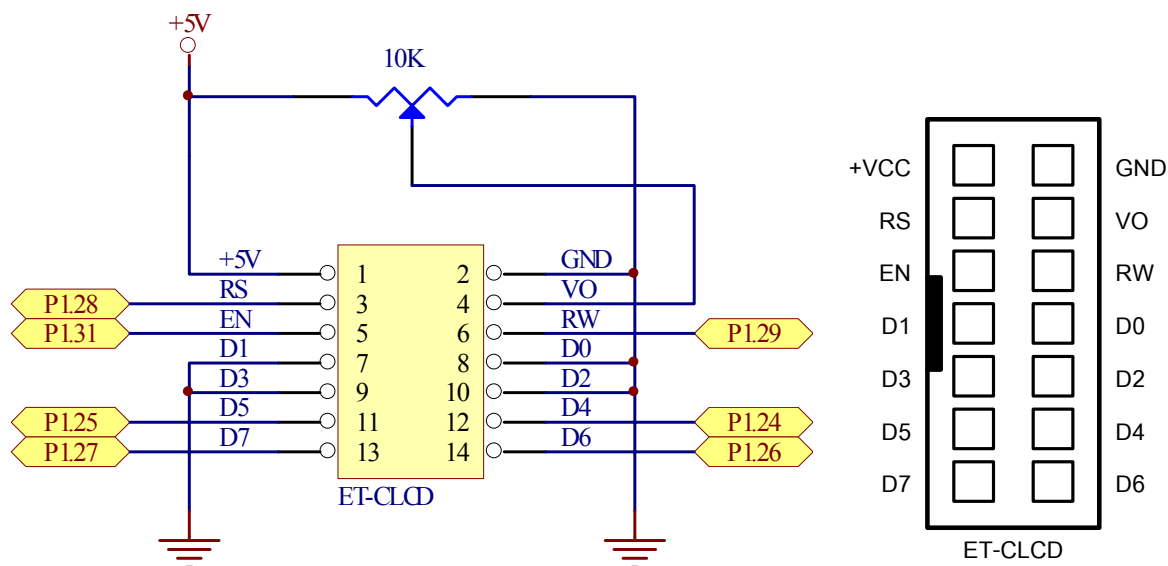
    delay(10000000);                   // Stop Beep Pulse
}
```

## การใช้งาน Character LCD

สำหรับการเชื่อมต่อ LCD นั้นจะสามารถใช้ได้กับ LCD แบบ Character Dot-Matrix เท่านั้น โดยเชื่อมต่อแบบ 4 บิต Data โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะเป็นสัญญาณจาก P1[24..29] และ P1[31] จำนวน 7 บิต โดยในการเชื่อมต่อสายสัญญาณจากขาต่อของ พอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น ดังรูป

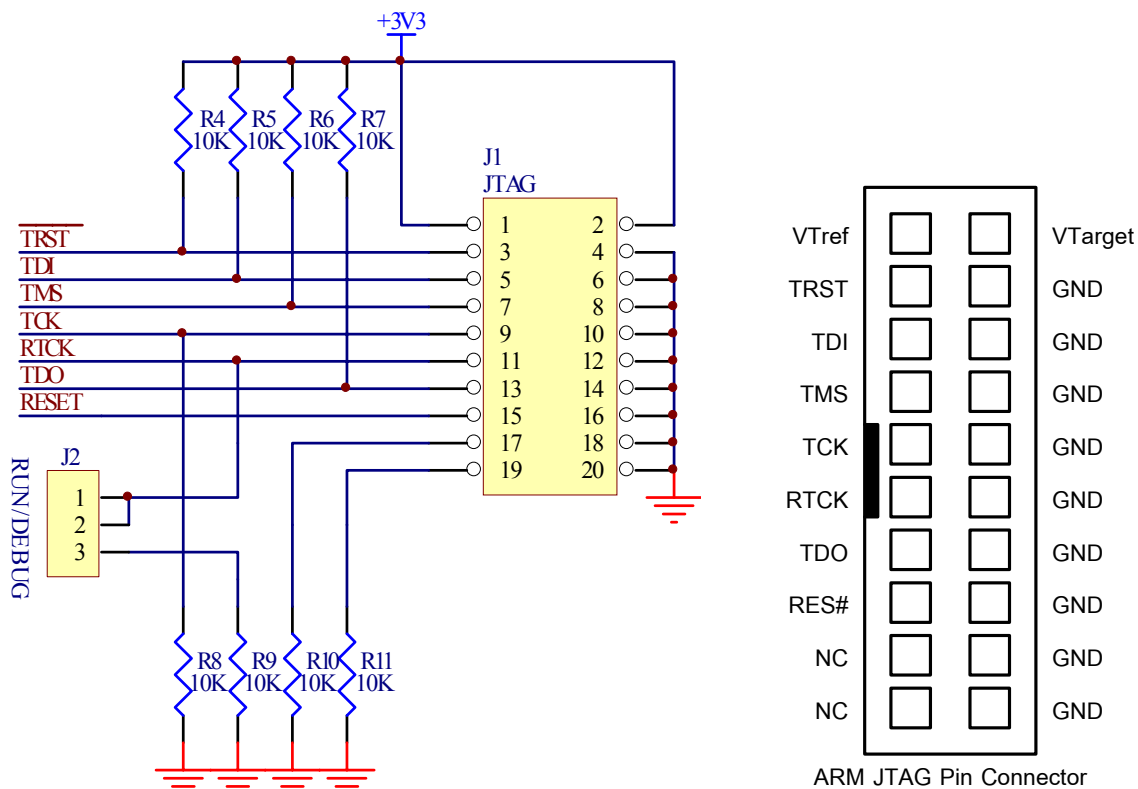
### สัญญาณการเชื่อมต่อกับ LCD

- DB4 = P1[24]
- DB5 = P1[25]
- DB6 = P1[26]
- DB7 = P1[28]
- RS = P1[28]
- RW = P1[29]
- EN = P1[31]



## การใช้งาน JTAG ARM

JTAG หรือ JTAG ARM จะเป็น Connector แบบ IDE 20 Pin สำหรับ Interface กับ JTAG Debugger โดยมีการจัดวงจรและสัญญาณตามมาตรฐานของ JTAG ดังนี้



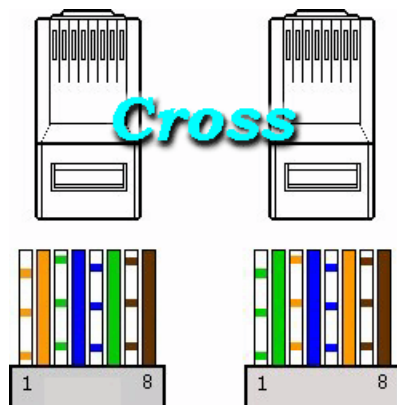
โดยในการใช้งานนั้นเมื่อต้องการเชื่อมต่อกับ JTAG เพื่อ Download Code หรือ Debug ผู้ใช้ต้องทำการเลือกกำหนดตำแหน่ง Jumper J2(RUN/DEB) ไว้ทางด้าน DEB ด้วย แต่หลังจากทำการพัฒนาโปรแกรมเสร็จเรียบร้อยแล้วต้องเลือก Jumper J2(RUN/DEB) กลับมายังด้าน RUN ด้วยเสมอ

## Ethernet LAN

สำหรับการเชื่อมต่อกับเครือข่าย Network ระหว่างบอร์ด CP-JR ARM7 LPC2368 นั้น จะใช้หัวต่อมาตรฐาน Ethernet แบบ RJ45 โดยวงจรส่วนนี้จะใช้ขาสัญญาณ P1[0,1,4,8,9,10,14..17] ในการเชื่อมต่อ โดยใช้ Chips Physical Ethernet เบอร์ DP83848 เป็น Driver ในการเชื่อมต่อ

สำหรับวิธีการเชื่อมต่อสายสัญญาณ Ethernet LAN ของบอร์ดเข้ากับระบบเครือข่ายจะทำได้ 2 แบบด้วยกัน คือการต่อแบบ Direct Line และต่อผ่าน Hub

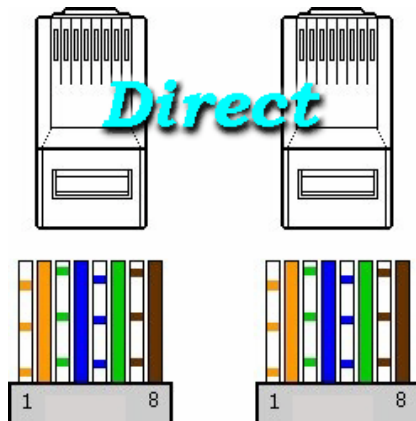
- กรณีที่ 1 คือ การเชื่อมต่อเข้ากับคอมพิวเตอร์โดยตรง สาย LAN จะต้องเข้าสายแบบ Cross



### 10BaseT cross-cable diagram

RJ-45 plug		RJ-45 jack
TD+ 1	↘	1 TD+
TD- 2	↗	2 TD-
RD+ 3	↗	3 RD+
n/c 4	↘	4 n/c
n/c 5	↘	5 n/c
RD- 6	↗	6 RD-
n/c 7		7 n/c
n/c 8		8 n/c

- กรณีที่ 2 คือ การเชื่อมต่อผ่าน Hub ของเครื่องคอมพิวเตอร์ Server จะต้องเข้าสายแบบ Direct



### 10BaseT cross-cable diagram

RJ-45 plug		RJ-45 jack
TD+ 1	=====	1 TD+
TD- 2	=====	2 TD-
RD+ 3	=====	3 RD+
n/c 4	=====	4 n/c
n/c 5	=====	5 n/c
RD- 6	=====	6 RD-
n/c 7	=====	7 n/c
n/c 8	=====	8 n/c

## การ์ดหน่วยความจำ SD/MMC

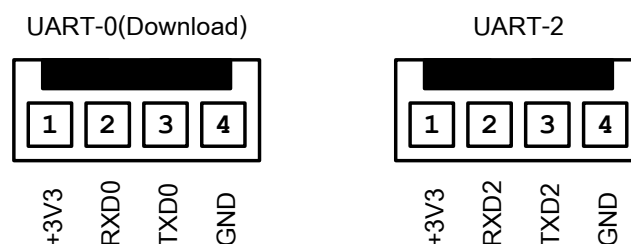
รองรับการเชื่อมต่อกับการ์ดหน่วยความจำแบบ SD Card และ MMC Card โดยในส่วนนี้จะมี LED SD สำหรับแสดง Status ของไฟเลี้ยงสำหรับการ์ดหน่วยความจำด้วย โดยที่แหล่งจ่ายไฟของการ์ดหน่วยความจำ สามารถเลือกได้กำหนดได้จาก Jumper J26 (SD/VDD) เพื่อเลือกใช้แหล่งจ่ายจาก +VDD ของบอร์ดหรือจากการควบคุมของขาสัญญาณ MCIPWR ซึ่งตามปกติจะกำหนด J26 (SD/VDD) ไว้ที่ตำแหน่ง SD เพื่อเป็นการเลือกใช้แหล่งจ่ายไฟของการ์ดหน่วยความจำจากการควบคุมของ MCIPWR โดยวงจรสำหรับเชื่อมต่อกับการ์ดหน่วยความจำทั้งหมดจะเลือกใช้ขาสัญญาณจาก MCU ดังนี้

- MCIDAT0 ใช้ P0.22
- MCIDAT1 ใช้ P2.11
- MCIDAT2 ใช้ P2.12
- MCIDAT3 ใช้ P2.13
- MCICMD ใช้ P0.20
- MCICLK ใช้ P0.19
- MCIPWR ใช้ P0.21
- CD ใช้ P0.8 (GPIO) สำหรับตรวจสอบการ Insert Card การ์ดหน่วยความจำ
- WP ใช้ P0.9 (GPIO) สำหรับตรวจสอบการกำหนด Write Protect ของการ์ดหน่วยความจำ



## การใช้งาน RS232

พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้ว โดยมีจำนวน 2 ช่อง ด้วยกันคือ UART-0 และ UART-2 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูลได้ นอกจากนี้แล้ว UART-0 ยังสามารถใช้งานเป็น ISP Download สำหรับทำการ Download Hex File ให้กับ MCU ได้ด้วย โดยในกรณีนี้ต้องใช้งานร่วมกับ SW1 (ISP LOAD) และ SW2 (RESET) เพื่อ Reset ให้ CPU เริ่มต้นทำงานใน Boot-Loader Mode เพื่อทำการ Download Hex File ให้กับ CPU ได้ด้วย(ดูรายละเอียดเพิ่มเติมเรื่อง “การ Download Hex File ให้กับ MCU ของบอร์ด”)



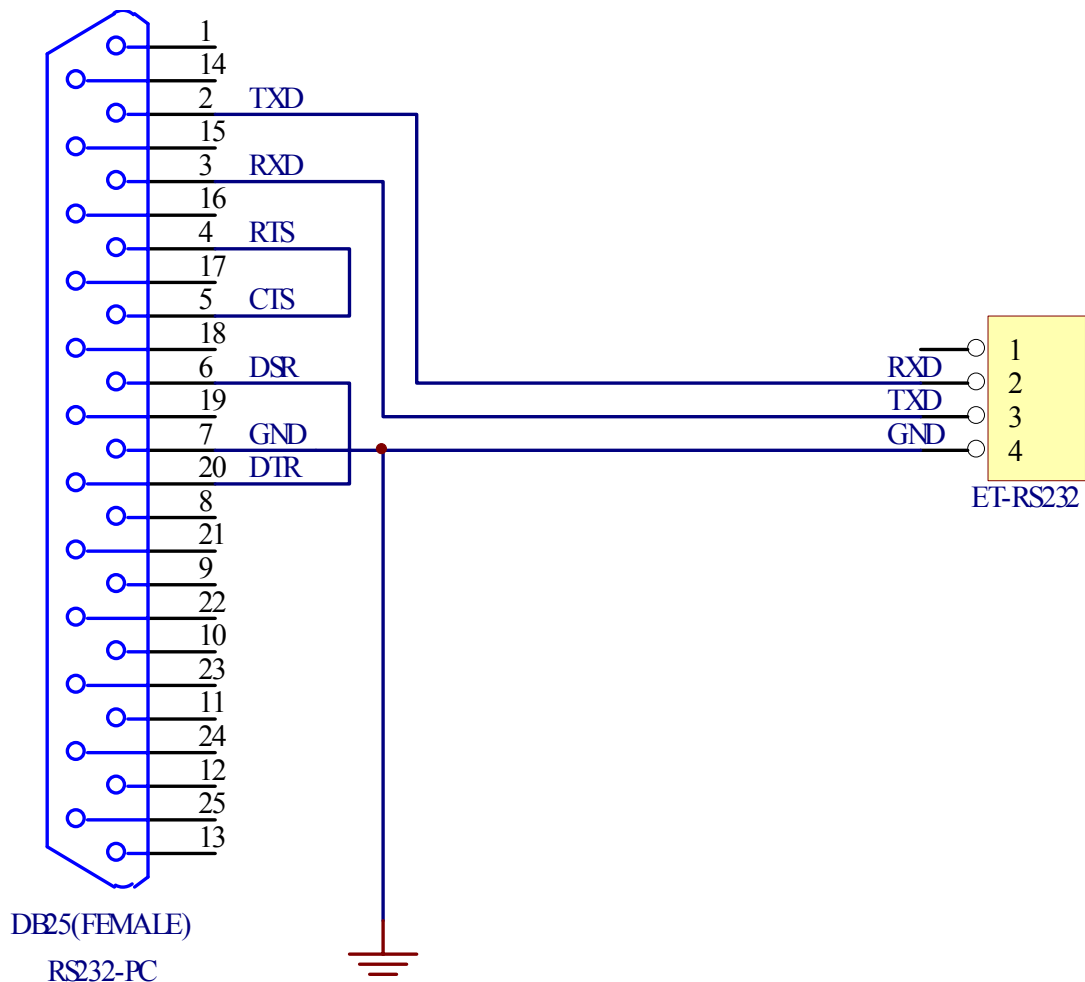
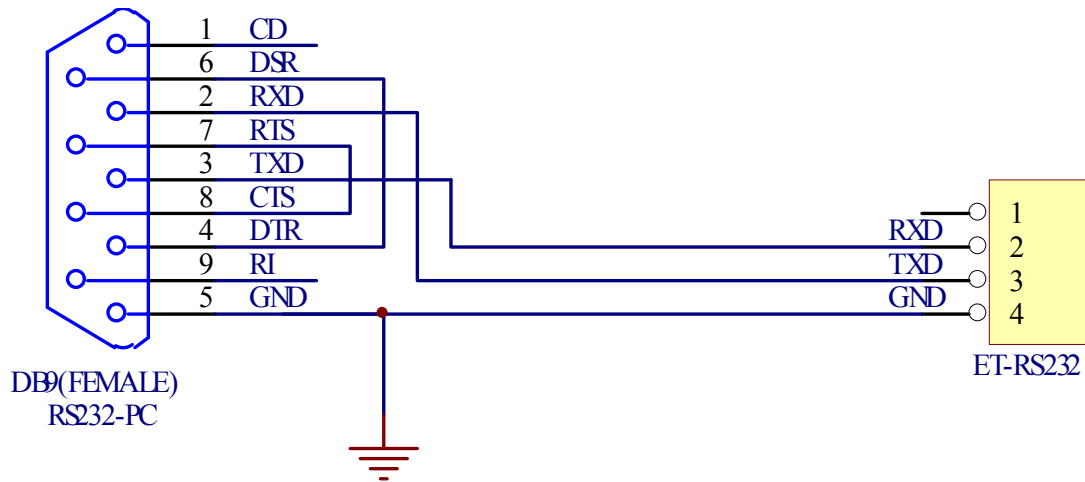
- UART-0 ใช้ขาสัญญาณจาก P0.2(TXD0) และ P0.3(RXD0)
- UART-2 ใช้ขาสัญญาณจาก P0.10(TXD2) และ P0.11(RXD2)

เนื่องจากระบบ Hardware UART ของ LPC2368 นั้นจะสามารถกำหนดขาสัญญาณในการเชื่อมต่อได้หลายจุด ตัวอย่างเช่น UART-2 สามารถเลือกให้ขาสัญญาณ P0[10] กับ P0[11] หรือ P2[8] กับ P2[9] ก็ได้ ซึ่งบอร์ด CP-JR ARM7 LPC2368 นั้นเลือกให้ขาสัญญาณชุด P0[10] กับ P0[11] เป็นจุดเชื่อมต่อกับ UART-2 ดังนั้น ผู้ใช้ต้องกำหนดคำสั่งในการเลือกให้ขาสัญญาณให้ถูกต้องด้วย และข้อควรระวังอีกประการหนึ่งในการใช้งาน UART ก็คือ ค่า Default ของ UART-2 จะถูกปิดการทำงานไว้ ดังนั้นผู้ใช้ต้องสั่งเปิดการทำงานของวงจร UART-2 ก่อนที่จะสั่ง Initial ค่าต่างๆให้กับ UART ด้วย ไม่เช่นนั้นจะไม่สามารถสั่งงาน UART ได้ สำหรับ Code ตัวอย่างการกำหนดค่า UART ในส่วนเริ่มต้นเป็นดังนี้

```
// Config UART-0 Connect to P0[2]:P0[3]
PINSEL0 &= 0xFFFFFFF0F; // Reset P0.2,P0.3 Pin Config
PINSEL0 |= 0x00000010; // Select P0.2 = TxD(UART-0)
PINSEL0 |= 0x00000040; // Select P0.3 = RxD(UART-0)

// Config UART-2 Connect to P0[10]:P0[11]
PINSEL0 &= 0xFF0FFFFFFF; // Reset P0.10,P0.11 Pin Config
PINSEL0 |= 0x00100000; // Select P0.10 = TxD(UART-2)
PINSEL0 |= 0x00400000; // Select P0.11 = RxD(UART-2)
PCONP |= 0x01000000; // UART2 Power-ON
```

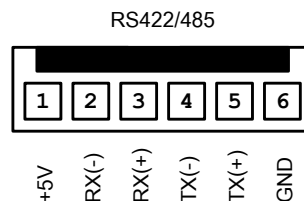
สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้วต่อ UART-0 และ UART-2 ของบอร์ด CP-JR ARM7 LPC2368 นั้น เป็นดังนี้



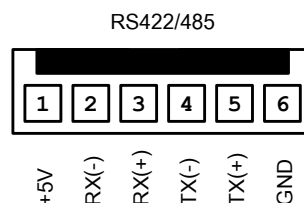
รูป แสดงวงจรสาย Cable สำหรับ RS232

## การใช้งาน RS422/485

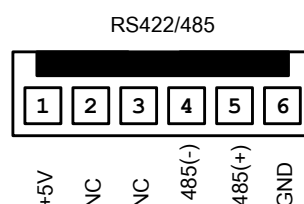
สำหรับวงจรสื่อสารแบบ RS422/485 ของบอร์ด CP-JR ARM7 LPC2368 นั้นจะเลือกใช้ UART-3 โดยใช้ขาสัญญาณชุด P0[0] และ P0[1] เป็นจุดเชื่อมต่อ โดยวงจรในส่วนนี้สามารถเลือกกำหนดการทำงานของวงจร Line Driver ได้ว่าจะใช้งานเป็นแบบ RS422(Full Duplex) หรือ RS485(Half Duplex) ซึ่งในกรณีใช้งานเป็น RS422 นั้นจะสามารถรับส่งข้อมูลได้พร้อมกันทั้ง 2 ทิศทางเหมือนกับ RS232 แต่ได้ระยะทางไกลกว่า แต่ถ้าเลือกเป็น RS485 จะสามารถเลือกกำหนดรูปแบบการสื่อสารเป็น Half-Duplex หรือ Full-Duplex ได้ โดย RS485 แบบ Full-Duplex จะคล้ายกับ RS422 แต่ต้องมีการควบคุมการเปิดและปิดของวงจรด้านส่งด้วย ส่วน RS485 แบบ Half-Duplex จะต้องใช้การสลับหน้าที่เป็น ฝ่ายรับ และ ฝ่ายส่ง โดยใช้สัญญาณ P1[19] ทำหน้าที่เป็น Output Port ในการกำหนดทิศทางข้อมูล โดยถ้า P1[19] เป็น โลจิก “1” จะเป็นการกำหนดทิศทางเป็นฝ่ายส่งข้อมูล และถ้า P1[19] เป็น โลจิก “0” จะเป็นการกำหนดทิศทางเป็นฝ่ายรับข้อมูล โดย IC Line Driver ของวงจร สามารถเลือกใช้เบอร์ 75176 หรือ MAX3088 ก็ได้ โดยถ้าเลือกใช้ 75176 จะสามารถทำการเชื่อมต่อสัญญาณของ RS485 ร่วมกับแบบ Multi-Drop ได้สูงสุด 32 จุด แต่ถ้าใช้ MAX3088 จะสามารถเชื่อมต่อร่วมกันแบบ Multi-Drop ได้สูงสุดมากถึง 256 จุด สำหรับข้อต่อสัญญาณของ RS422/485 นั้นจะเป็นขั้วแบบ CPA-6 ซึ่งมีการจัดเรียงสัญญาณดังรูป



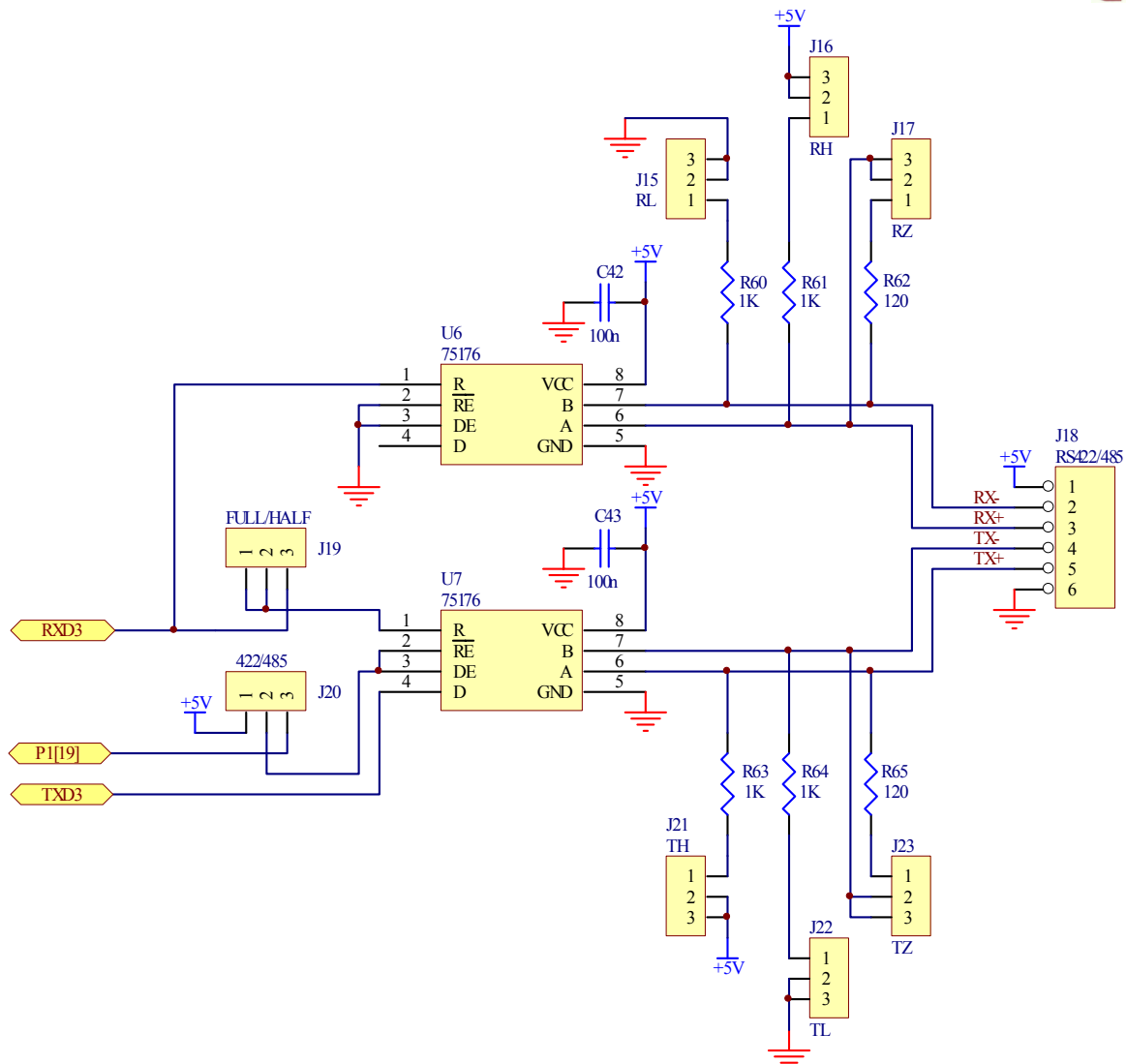
รูปแสดง ตำแหน่งของขาสัญญาณเมื่อใช้งานเป็น RS422



รูปแสดง ตำแหน่งของขาสัญญาณเมื่อใช้งานเป็น RS485 แบบ Full-Duplex



รูปแสดง ตำแหน่งของขาสัญญาณเมื่อใช้งานเป็น RS485 แบบ Half-Duplex



รูปแสดง วงจรของ RS422/485

ในกรณีที่ต้องการใช้งานเป็น RS422 ผู้ใช้ต้องทำการติดตั้ง IC Line Driver จำนวน 2 ตัว โดยในกรณีนี้ต้องกำหนดตำแหน่งของ Jumper J19 และ J20 ให้เป็น RS422 แบบ Full Duplex ด้วยโดยตำแหน่งของ J19 (FULL/HALF) ให้กำหนดไว้ทางด้าน FULL ส่วน J20 (422/485) ให้กำหนดไว้ทางด้าน 422

ในกรณีที่ต้องการใช้งานเป็น RS485 แบบ Full-Duplex ผู้ใช้ต้องทำการติดตั้ง IC Line Driver จำนวน 2 ตัว เช่นเดียวกันกับ RS422 ต่างกันที่กรณีนี้ต้องกำหนดตำแหน่งของ Jumper J19 และ J20 ให้เป็น RS485 แบบ Full Duplex โดย J19 (FULL/HALF) ให้กำหนดไว้ทางด้าน FULL ส่วน J20 (422/485) ให้กำหนดไว้ทางด้าน 485 โดยการทำงานจะมีลักษณะคล้ายกับ RS422 คือสามารถรับส่งข้อมูลได้พร้อมกัน 2 ทิศทางเช่นเดียวกับ RS422 แต่สามารถเชื่อมต่อกันเป็นแบบ Multi-Drop ได้ โดยวงจรสามารถรับข้อมูลได้ตลอดเวลาผ่านทาง IC Line Driver ของ U6 ส่วน IC Line Driver U7 จะใช้ทำหน้าที่เป็นตัวส่งข้อมูล แต่สามารถควบคุมการ เปิด/ปิด การส่งข้อมูลได้ด้วยสัญญาณจาก P1[19] โดยถ้า P1[19] เป็น โลจิก “1” จะเป็นการส่งเปิดการส่งข้อมูล และถ้า P1[19] เป็น โลจิก “0” จะเป็นการส่งปิดการส่งข้อมูล โดย

เมื่อสั่งปิดการส่งข้อมูลจะมีสถานะเหมือนกับการปลดสายสัญญาณด้านส่ง ออกจากวงจร ทำให้สัญญาณด้านส่งไม่ไปชนกับข้อมูลของอุปกรณ์ตัวอื่นๆ โดยผู้ใช้นี้ต้องเขียนโปรแกรมควบคุมให้อุปกรณ์ทั้งหมดทำการส่งข้อมูลออกไปในสายสัญญาณครั้งละ 1 ตัวเท่านั้น

ในกรณีที่ต้องการใช้งานเป็น RS485 แบบ Half Duplex ให้ผู้ใช้งานทำการติดตั้ง IC Line Driver ที่ตำแหน่ง U7 เพียงตัวเดียว โดยในกรณีนี้ต้องกำหนดตำแหน่งของ Jumper J19 และ J20 ให้เป็น RS485 แบบ Half Duplex ด้วย โดย J19 (FULL/HALF) ให้กำหนดไว้ทางด้าน HALF ส่วน J20 (422/485) ให้กำหนดไว้ทางด้าน 485 โดยให้ต่อสายสัญญาณใช้งานของขั้ว CPA-6(J18) ที่ตำแหน่ง TX(-) และ TX(+) ออกไปใช้งาน โดยในกรณีนี้ TX(-) จะทำหน้าที่เป็น RS485(-) และ TX(+) จะทำหน้าที่เป็น RS485(+) ซึ่งทิศทางและหน้าที่ของสัญญาณ RS485 นี้จะขึ้นอยู่กับค่าลอจิกของ P1[19] ซึ่งเป็นขา Output Port ที่ทำหน้าที่เป็น Direction Control

สำหรับการกำหนด Jumper ของ Fail-Save Resistor และ Terminate Resistor นั้น ตามปกติแล้วถ้าบอร์ดถูกติดตั้งใช้งานอยู่ในตำแหน่ง ต้นทาง และ ปลายทาง ของสายสัญญาณ Jumper เหล่านี้ต้องทำการ Enable โดยเลือก Jumper (EN/DIS) ของ RZ, RL, RH, TZ, TL และ TH ไว้ทางด้าน EN หรือ Enable เสมอ โดย Fail-Save Resistor (RL, RH, TL และ TH) จะช่วยทำให้สถานะลอจิกในสายสัญญาณมีสถานะ IDLE ที่ถูกต้องในกรณีที่ไม่มีกรับส่งข้อมูล ส่วน Terminate Resistor (RZ และ TZ) จะช่วยชดเชยค่าความต้านทาน หรือ Impedance ที่เกิดขึ้นในสาย เมื่อสายมีความยาวมากๆ

สำหรับ Code การเขียนโปรแกรมสำหรับกำหนดการใช้งาน UART-3 ของ RS422/485 จะต้องกำหนดให้ใช้ P0[0] และ P0[1] เป็นขาเชื่อมต่อสัญญาณ และใช้ P1[19] ควบคุมทิศทางการรับส่งข้อมูลของ RS485 โดยมีข้อควรระวังอย่างหนึ่งคือ ค่า Default ของวงจร UART-3 ใน LPC2368 นั้น ตามปกติแล้ววงจรจะถูกปิดการทำงานไว้ ดังนั้นเมื่อต้องการเปิดการใช้งานวงจรส่วนนี้ ผู้ใช้ต้องไม่ลืมเขียนคำสั่งเพื่อสั่งเปิดการทำงานของ UART-3 ก่อนที่จะเริ่มกำหนดค่าต่างๆ ให้กับวงจร โดยการสั่งเปิดการทำงานของ UART-3 นั้นจะควบคุมจากบิตในรีจิสเตอร์ PCONP ดังตัวอย่าง

```
// Config UART-3(RS422/485) Connect to P0[0]:P0[1]
PINSEL0 &= 0xFFFFFFFF; // Reset P0.0,P0.1 Pin Config
PINSEL0 |= 0x00000002; // Select P0.0 = TxD(UART3)
PINSEL0 |= 0x00000008; // Select P0.1 = RxD(UART3)
PCONP |= 0x02000000; // UART3 Power-ON

// Config P1.19 = Output Control Direction RS485
// P1.19 = "0" = Received RS485
// P1.19 = "1" = Transmit RS485
PINSEL3 &= 0xFFFF3FFF; // P1.19 = GPIO
IODIR1 = 0x00080000; // Pin Control Direction RS485 = Output

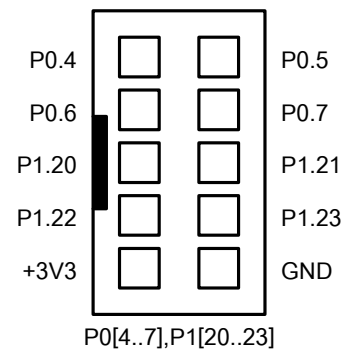
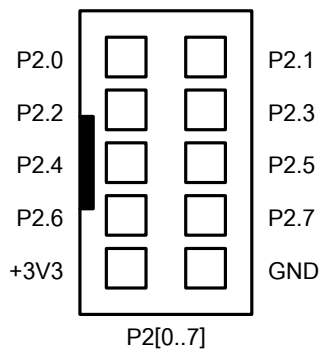
// Select Direction RS485 = Receive
IOCLR1 = 0x00080000; // RS485 Direction = 0 (Receive)

// Select Direction RS485 = Transmit
IOSET1 = 0x00080000; // RS485 Direction = 1 (Transmit)
```

## ขั้วต่อ Port I/O ต่างๆของบอร์ด

สำหรับขั้วต่อ Port I/O ของ CPU นั้น จะจัดเรียงออกมารอไว้ยังขั้วต่อแบบต่างๆ สำหรับให้ผู้ใช้เลือกต่อออกไปใช้งานตามต้องการ โดยมีด้วยกัน 6 ชุดดังนี้

- ขั้วต่อ IDE 10 Pin จำนวน 2 ชุด ชุดละ 8 บิต คือ P2[0..7] และ KEY4X4 (P0[4..7],P1[20..23]) โดยมีการจัดเรียงสัญญาณไว้ดังนี้



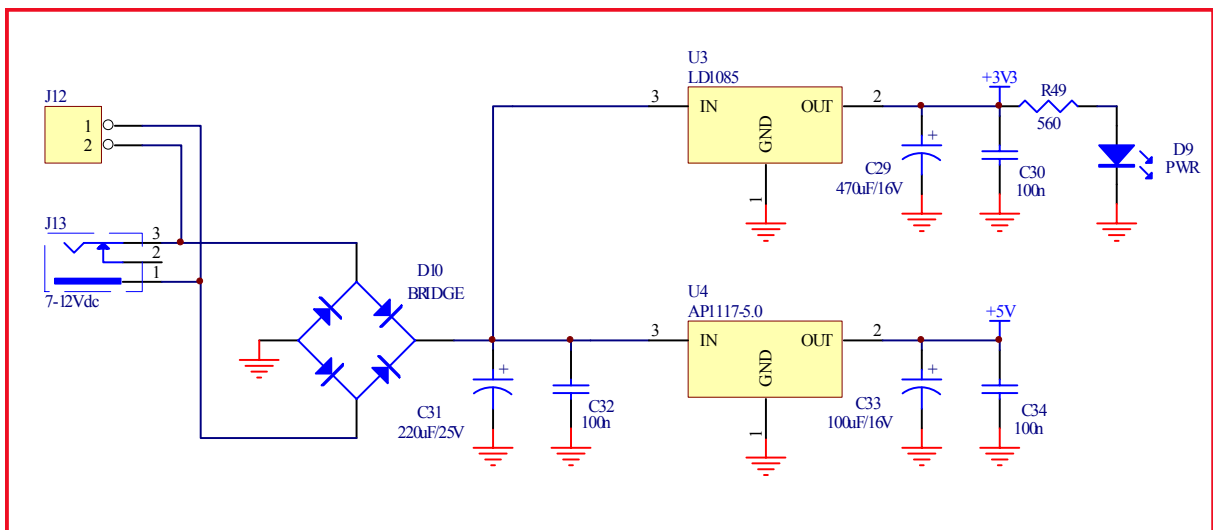
- ขั้วต่อ SPI-0 เป็น Header ขนาด 1x6 ใช้เป็นจุดเชื่อมต่อ P0[15..18] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น SPI Bus ได้ตามต้องการ
  - P0.15 = SCK/SCK0
  - P0.16 = SSEL/SSEL0
  - P0.17 = MISO/MISO0
  - P0.18 = MOSI/MOSI0
- ขั้วต่อ I2C-0 เป็น Header ขนาด 1x4 ใช้เป็นจุดเชื่อมต่อ P0[27..28] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น I2C Bus ได้ตามต้องการ
  - P0.27 = SDA0
  - P0.28 = SCL0
- ขั้วต่อ A/D เป็น Header ขนาด 1x4 ใช้เป็นจุดเชื่อมต่อ P0[24..25] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น A/D ได้ตามต้องการ
  - P0.24 = AD0.1
  - P0.25 = AD0.2
- ขั้วต่อ D/A เป็น Header ขนาด 1x3 ใช้เป็นจุดเชื่อมต่อ P0[26] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น D/A(Aout) ได้ตามต้องการ
  - P0.26 = AOUT หรือ D/A



## วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟสามารถใช้งานได้กับไฟ AC/DC ขนาด 7-12V ได้ ซึ่งสามารถต่อไฟเลี้ยงให้บอร์ดได้ทั้งจุดต่อที่เป็น Terminal และ Jack-DC ก็ได้ตามต้องการ โดยไฟที่ต่อให้นี้จะถูกส่งต่อไปเข้าวงจร Bridge Rectifier และ Regulate ขนาด +3V3/3A และ +5V/800mA

โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจร Regulate ขนาด 3.3V นั้นจะจ่ายให้กับ CPU และวงจร I/O ของบอร์ดทั้งหมด ยกเว้น LCD แบบ Character และวงจร Line Driver ของ RS422/485 ซึ่งจะใช้แหล่งจ่ายไฟขนาด +5VDC จากวงจร Regulate

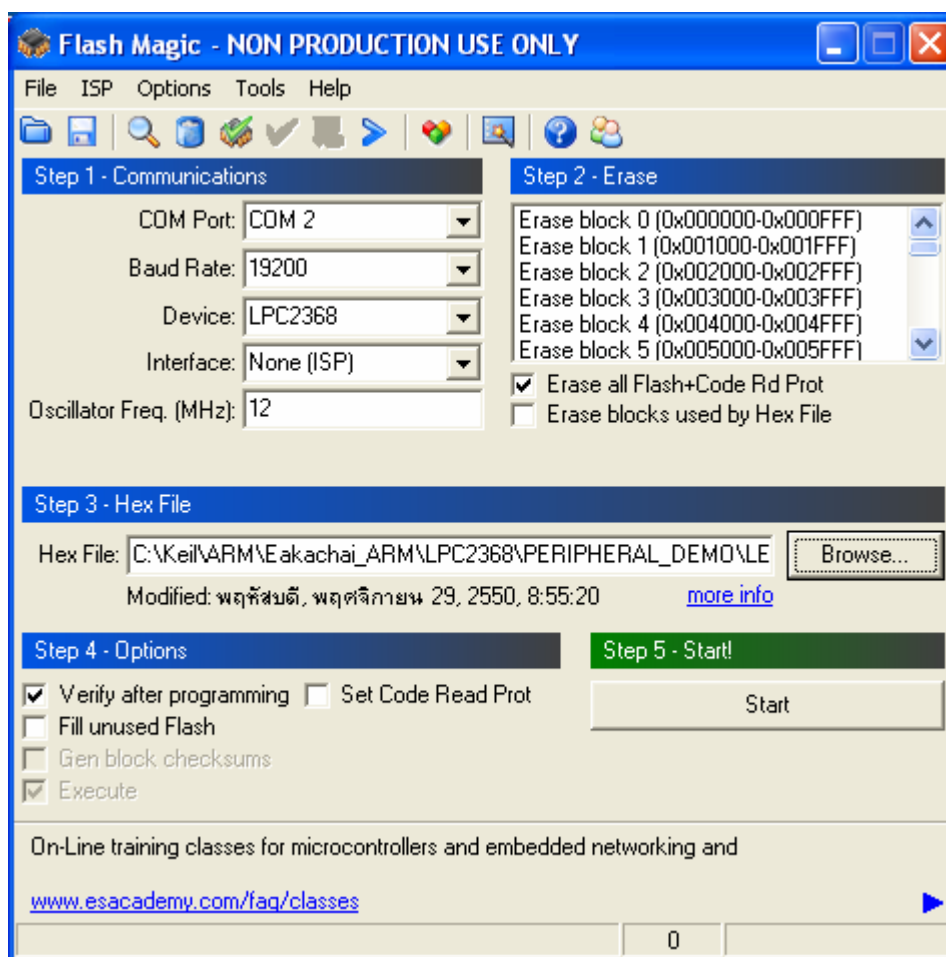


## การ Download Hex file ให้กับ MCU ของบอร์ด

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ Flash Magic ของ “Embedded System Academy, Inc” ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดได้ที่ [www.esacademy.com](http://www.esacademy.com)

### ขั้นตอนการ Download HEX File ให้กับ MCU

1. ต่อสายสัญญาณ RS232 ระหว่างพอร์ตสื่อสารอนุกรม RS232 ของ PC และบอร์ด UART-0
2. จ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งจะสังเกตเห็น LED PWR ติดสว่างให้เห็น
3. สั่ง Run โปรแกรม Flash Magic ซึ่งถ้าเป็น Version 4.02.260 จะได้ผลดังรูป



4. เริ่มต้นกำหนดค่าตัวเลือกต่างๆให้กับโปรแกรมตามต้องการ ซึ่งในกรณีนี้ใช้กับ LPC2368 ของบอร์ด CP-JR ARM7 LPC2368 ของ อีทีที ให้เลือกกำหนดค่าต่างๆให้โปรแกรมหาดังนี้
  - 2.1 เลือก COM Port ให้ตรงกับหมายเลข COM Port ที่ใช้งานจริง (ในตัวอย่างใช้ COM2)
  - 2.2 ตั้งค่า Baud Rate อยู่ระหว่าง 2400 - 115200 ซึ่งถ้าเลือกใช้ค่า Baud rate สูงๆ แล้วเกิด Error ให้ลดค่า Baud rate ให้ต่ำลง จากตัวอย่างใช้ค่า 19200

2.3 กำหนด Device เป็น LPC2368

2.4 กำหนด Interface เป็น None ISP

2.5 กำหนดค่าคริสตอล ออสซิลเลเตอร์ ให้ตรงกับที่ใช้ในจริงภายในบอร์ด โดยกำหนดให้มีหน่วยเป็น MHz ในที่นี้ใช้ค่า 12.000MHz ซึ่งต้องกำหนดเป็น 12

2.6 ให้กดสวิตช์ ISP LOAD และ RESET ที่บอร์ด “CP-JR ARM7 LPC2368” เพื่อทำการ Reset ให้ MCU ทำงานใน Boot Loader ตามขั้นตอนดังต่อไปนี้

- กดสวิตช์ ISP LOAD ค้างไว้
- กดสวิตช์ RESET โดยที่สวิตช์ ISP LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ RESET โดยที่สวิตช์ ISP LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ ISP LOAD เป็นลำดับสุดท้าย

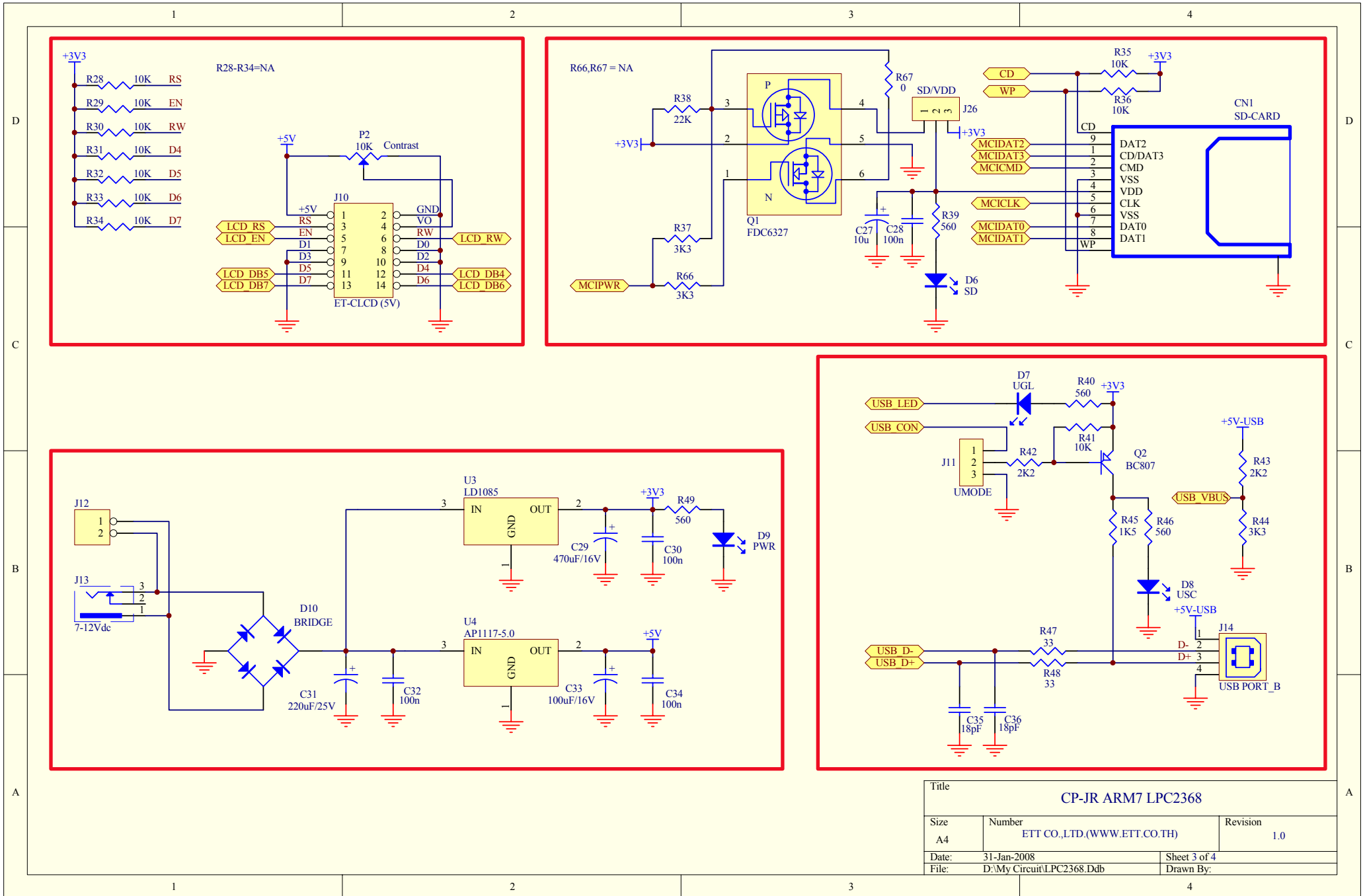
5. เลือกรูปแบบการลบข้อมูลเป็น “Erase all Flash + Code Rd Prot”

6. เลือกกำหนด Option เป็น “Verify after programming”

7. ให้คลิกเมาส์ที่ “Browse” เพื่อทำการเลือกกำหนด HEX File ที่จะทำการสั่ง Download

8. ให้ทำการคลิกเมาส์ที่ “Start” ซึ่งโปรแกรม Flash Magic จะเริ่มต้นทำการ Download ข้อมูลให้กับ MCU ทันที โดยสังเกตการทำงานที่ Status bar โดยในขั้นตอนนี้ให้รอจนกว่าการทำงานของโปรแกรมจะเสร็จสมบูรณ์

9. เมื่อทำงานของโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดสวิตช์ Reset ที่บอร์ด ซึ่ง MCU จะเริ่มต้นทำงานตามโปรแกรมที่สั่ง Download ให้ทันที

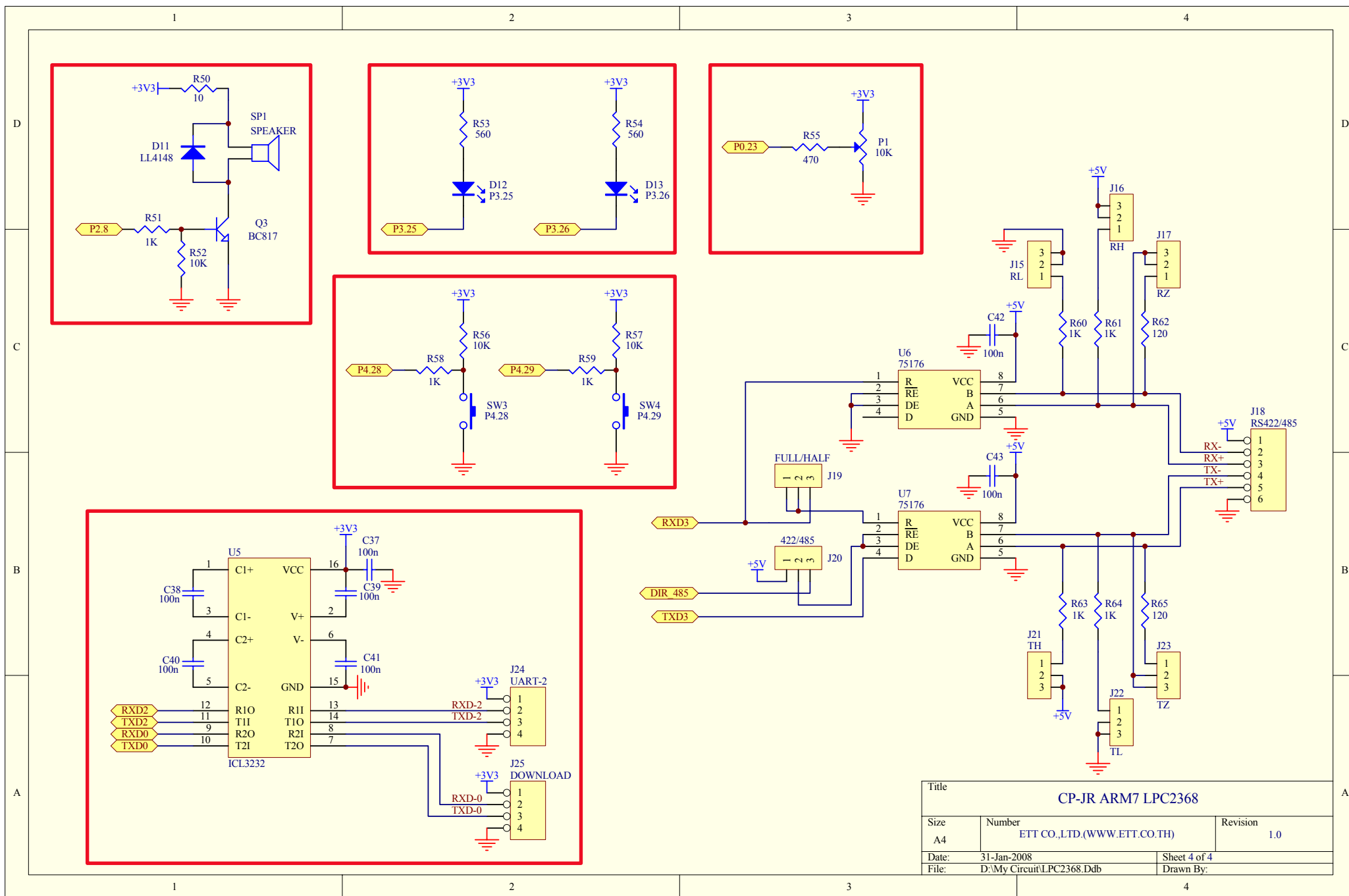


Title		
CP-JR ARM7 LPC2368		
Size	Number	Revision
A4	ETT CO.,LTD.(WWW.ETT.CO.TH)	1.0
Date:	31-Jan-2008	
File:	D:\My Circuit\LPC2368.Ddb	
	Sheet 3 of 4	Drawn By:









Title		
CP-JR ARM7 LPC2368		
Size	Number	Revision
A4	ETT CO.,LTD.(WWW.ETT.CO.TH)	1.0
Date:	31-Jan-2008	Sheet 4 of 4
File:	D:\My Circuit\LPC2368.Ddb	Drawn By: