Use Serial Ports on M6117D and Vortex86 Series

2004-12-13

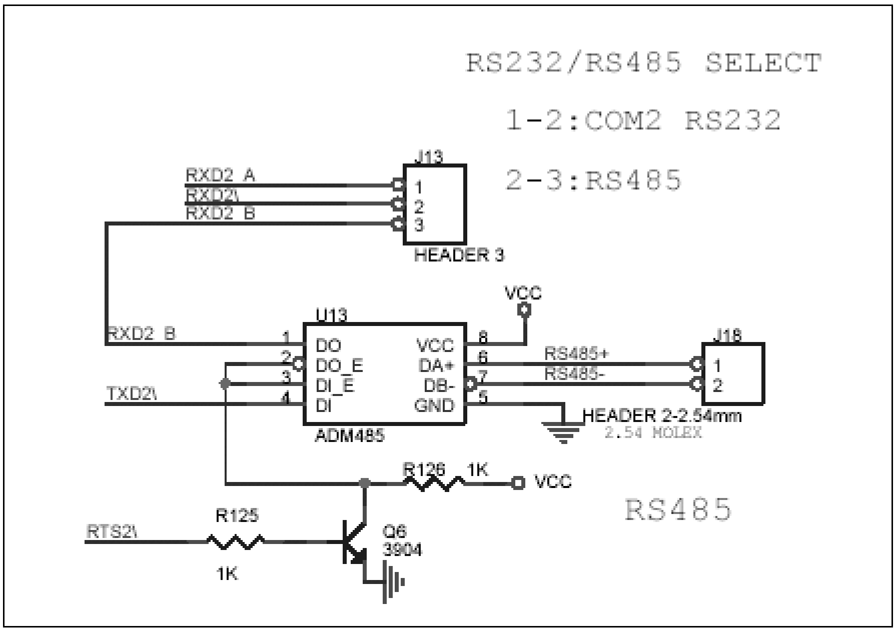
If there are two serial ports on your board, the COM1 will use IRQ 4 and COM2 will use IRQ3. If there are more than 2 serial ports on your board, there will be a lot of sets. See the table:

	I/O Address	IRQ (share)	M6117D (not share IRQ)	Vortex86 (not share IRQ)
COM1	3F8H	4	4	4
COM2	2F8H	3	3	3
COM3	3E8H	4	10	10
COM4	2E8H	3	11	11

In general, the second serial port will be set to RS-485 mode. Before using serial ports, check the RS-485 jumper please. When COM3/4 use IRQ > 7, remember to modify your interrupt vector of program to INT 72H/73H and enable IRQ 2 (IRQ 2 is used to cascade interrupt controller 2 to controller 1). Send EOI to both two 8259 controllers in your ISR.

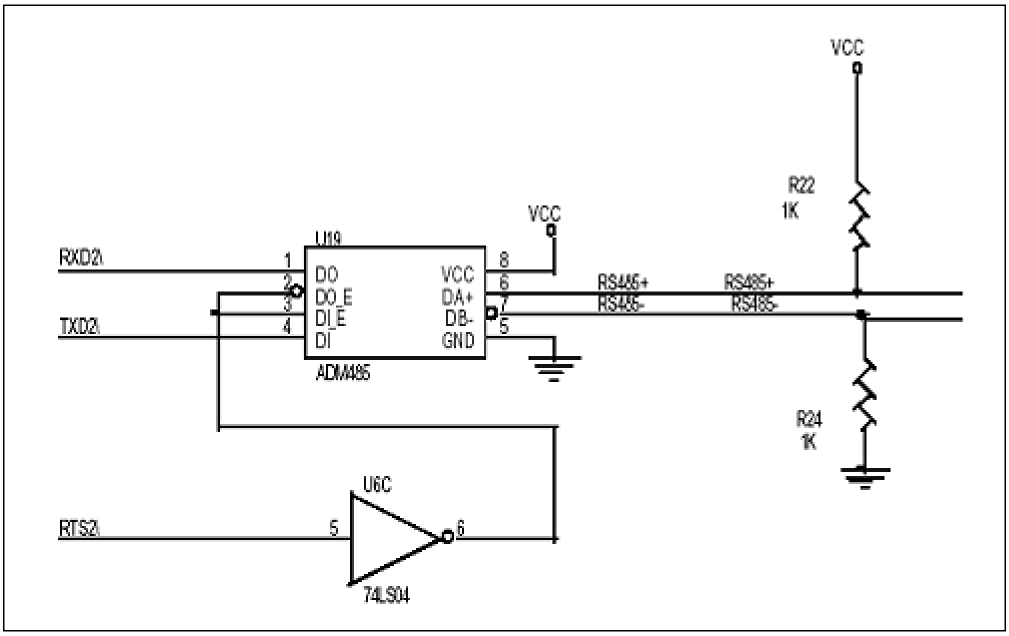
Use RS-485

User can use RS-485 after set jump to enable RS-485. For example, short pin 2 & 3 of JP3 to select RS-485 on ICOP-6026, and then you can send/receive data to/from CONT9. (The jumper and connector number depend on you SBC. They may be different.) Before sending data to RS-485, set RTS=0 to enable input. Set RTS=1 to enable output before receiving data from RS-485. See the example diagram from ICOP-6071 to know why:



Get Wrong from RS-485

If programmer gets error from RS-485, we recommend adding resister to fix it. Add 1K-ohm resister from RS-485 "+" to Vcc and another 1K-ohm resister from RS-485 "-" to GND. If programmer still gets wrong data, use 10K-ohm resister to replace 1K resister. See the diagram:



RS-485 DOS C Example Code

This is the RS-485 example with SerPort library. It will show programmer how to use RTS to control RS-485 transmission direction.

```
/*

Serial port talk example for SerPort, copyright (C) 2004 by DM&P.
This example will show you how to use SerPort library.

*/
#include "serport.h"
#include <conio.h>
#include <stdio.h>

void main()
{
  char  c;

  printf("\nDM&P SerPort Demo Program %s %s.\n\n", __DATE__, __TIME__);
  printf("This program will send/receive key press message via COM1.\n");
  printf("Press ESC to quit.\n\n");

  /* Open COM1 */
  SerPort_Open(COM1, 4096, 0x2e8, 11);

  /* Check COM1 */
  if(SerPort_IsOk(COM1) == 0)
  {
    printf("Unable to initialize COM1.\n");
    return;
  }

  /* Set baud rate 9600 bps, no parity, 8 bits, 1 stop bit */
  SerPort_SetParam(COM1, BAUD_9600, PARITY_NO, LENGTH_8_BIT, STOPBIT_1_BIT);

  /* Set RS-485 as input mode */
  SerPort_SetRTS(COM1, 0);

  printf("Waiting key press...\n");
  while(1)
  {
    /* Detect key press and send it to COM1 */
    if(kbhit())
    {
      c = getch();
      if(c == 27)
        break;

      /* Set RS-485 as output mode */
      SerPort_SetRTS(COM1, 1);

      SerPort_SendByte(COM1, c);

      /* Set RS-485 as input mode */
      SerPort_SetRTS(COM1, 0);

      printf("%c", c);
    }

    /* Is any data avail on COM1 ? */
    if(SerPort_Avail(COM1))
    {
      SerPort_RecvByte(COM1, &c);
      printf("%c", c);
    }
  }

  /* Close COM1 & COM1 */
  SerPort_Close(COM1);
}
```

Technical Support

For more technical support, please visit http://www.dmp.com.tw/tech or mail to tech@dmp.com.tw.