PANTECH SOLUTIONS®
Technology Beyond the Dreams

ARM HOW-TO GUIDE

Interfacing GSM with LPC2148 ARM

Contents at a Glance

Join the Technical Community Today!
http://www.pantechsolutions.net

ARM7 LPC2148 Primer Board

The ARM7 LPC2148 Primer board is specifically designed to help students to master the required skills in the area of embedded systems. The kit is designed in such way that all the possible features of the microcontroller will be easily used by the students. The kit supports in system programming (ISP) which is done through serial port.

NXP's ARM7 (LPC2148), ARM Primer Kit is proposed to smooth the progress of developing and debugging of various designs encompassing of High speed 32-bit Microcontrollers.

GSM (Global System for Mobile Communication)

GSM is a digital mobile telephony system. GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band.

Join the Technical Community Today!
**http://www.pantechsolutions.net**

Interfacing GSM

Fig. 1 shows how to interface the GSM with microcontroller. The GSM module is communicate the microcontroller with mobile phones through UART. To communicate over UART or USART, we just need three basic signals which are namely, RXD (receive), TXD (transmit), GND (common ground).

GSM modem interfacing with microcontroller for SMS control of industrial equipments. The sending SMS through GSM modem when interfaced with microcontroller or PC is much simpler as compared with sending SMS through UART.

Text message may be sent through the modem by interfacing only three signals of the serial interface of modem with microcontroller i.e., TxD, RxD and GND. In this scheme RTS and CTS signals of serial port interface of GSM Modem are connected with each other.

The transmit signal of serial port of microcontroller is connected with transmit signal (TxD) of the serial interface of GSM Modem while receive signal of microcontroller serial port is connected with receive signal (RxD) of serial interface of GSM Modem.

The SMS message in text mode can contain only 140 characters at the most. It depends upon the amount of information collected from GPS Engine that you need at the base station for tracking vehicle or person.
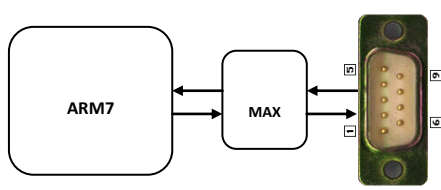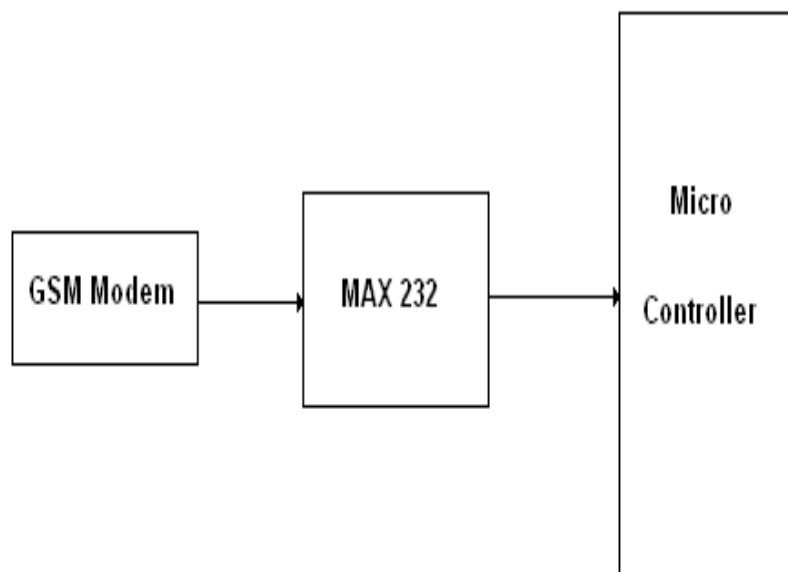

Fig. 1 Interfacing UART to Microcontroller
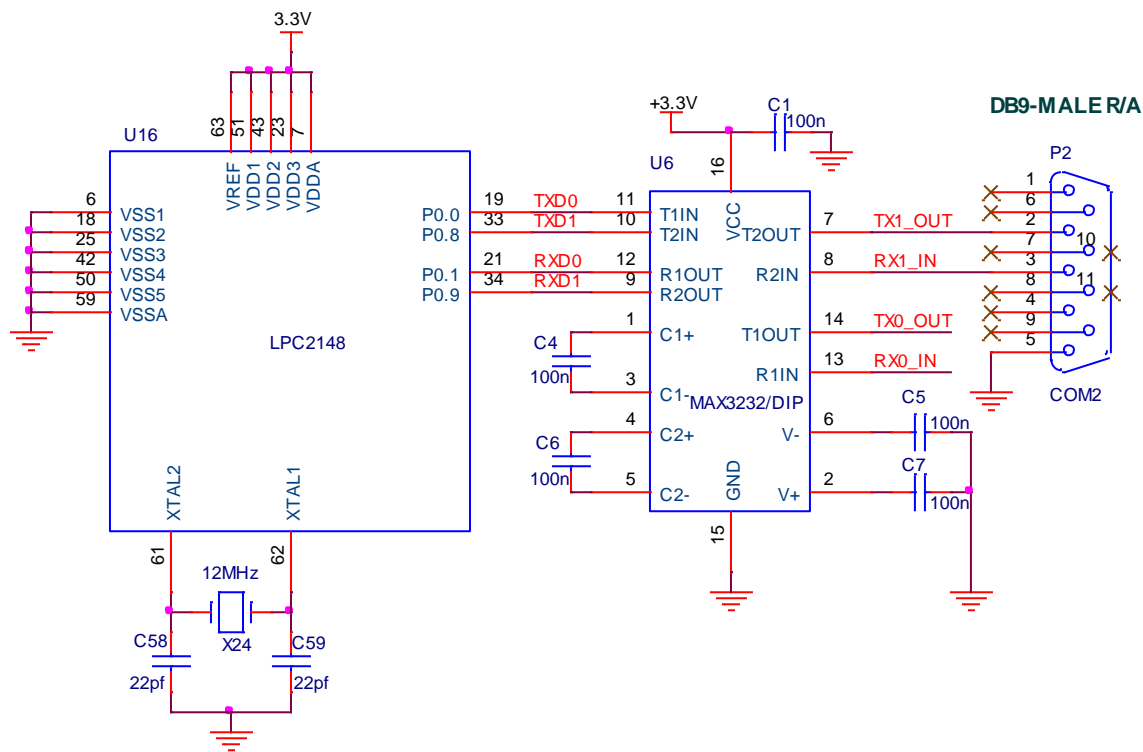
Interfacing GSM with LPC2148

     We now want to display a text in mobile from LPC2148 Primer Board by using GSM module through UART. In LPC2148 Primer Board contains two serial interfaces that are UART0 & UART1. Here we are using UART0. The GSM modem is being interfaced with the microcontroller LPC2148 Primer Board for SMS communication. The SMS can be sending and receiving for the data sharing and situation information and control.

Pin Assignment with LPC2148

	UART DB-9 Connector	LPC2148 Processor Lines	Serial Port Section
UART0(P1) ISP PGM	TXD-0	P0.0	
	RXD-0	P0.1	
UART1 (P2)	TXD-1	P0.8	
	RXD-1	P0.9	

Circuit Diagram to Interface GSM with LPC2148

Source Code

The Interfacing GSM module with LPC2148 program is very simple and straight forward, which send a message to mobiles from LPC2148 Primer Board through GSM module by using UART0. Some delay is occurring when a single data is sent to mobile through UART. C programs are written in Keil software. The baud rate of microcontroller is 9600.

C Program to send a message from LPC2148

```
*******************************************************************************
         Title   : Program to send a message from LPC2148 to mobile through GSM
*******************************************************************************

         #define CR       0x0D
         #include <LPC21xx.H>
         #include <stdio.h>

         void getstring(unsigned char *);
         int getchar (void)  /* Read character from Serial Port */
         void status_ok(void);
         void Serial_Init(void);
         void delay(unsigned int n);

         void main(void)
         {
            unsigned int cnt=0x80,m;
            char xx;
            Serial_Init();
            delay(50);

            while(1)
            {
              printf("AT\r");   // AT COMMAND FOR INITIALING
              status_ok();
              printf("AT+IPR=9600\r"); // AT COMMAND FOR BAUD RATE
              status_ok();

              printf("AT+CMGR=2\r"); // Reading the message detail
                 // at Index 1 with phone number, data and time
              status_ok();
              delay(250);

              printf("ATD9790550124;\r");//AT COMMAND FOR CALL DIALING
              delay(250);
```

```c
            status_ok();
            delay(500);
            delay(500);
            delay(500);
            delay(500);
            delay(500);
            delay(500);

            printf("ATH\r"); // AT COMMAND FOR CALL DISCONNECTING
            delay(250);
            status_ok();
            delay(500);
            delay(500);

            printf("ATDL\r"); // AT COMMAND FOR REDIALING
            delay(250);
            status_ok();
            delay(500);
            delay(500);

            printf("ATH\r"); // AT COMMAND FOR ANSWERING THE CALL
            delay(250);
            status_ok();
            delay(500);
            delay(500);
        }
}

void getstring(unsigned char *array)
{
    unsigned char temp=0, i=0;
    do
    {
        temp = getchar();
       *array++ = temp;
    }
    while((temp != '\r') && (temp != '\n'));
    *array = '\0';

}
```

```c
int getchar (void)  /* Read character from Serial Port */
{

  while (!(U0LSR & 0x01));
  return (U0RBR);
}



void status_ok(void)
{
      getstring(y);
      while(!(strstr(y,"OK"))) getstring(y);
      pointr = strstr(y,"OK");
      lcd_cmd(0xc0);
      lcd_data(*pointr++);
      lcd_data(*pointr);
      delay(500);
      lcd_cmd(0x01);

}

void Serial_Init(void)
{
   PINSEL0 |= 0X00000005; //Enable Txd0 and Rxd0
   U0LCR    = 0x00000083; //8-bit data, no parity, 1-stop bit
   U0DLL   = 0x00000061;  //for Baud rate=9600,DLL=82
   U0LCR   = 0x00000003;  //DLAB = 0;
}

void delay(unsigned int n)
{
   unsigned int i,j;
   for(i=0;i<n;i++)
   {
      for(j=0;j<12000;j++)
      {;}
   }
}
```

To compile the above C code you need the KEIL software. They must be properly set up and a project with correct settings must be created in order to compile the code. To compile the above code, the C file must be added to the project.

In Keil, you want to develop or debug the project without any hardware setup. You must compile the code for generating HEX file. In debugging Mode, you want to check the port output without LPC2148 Primer Board.

The Flash Magic software is used to download the hex file into your microcontroller IC LPC2148 through UART0.

Testing the GSM with LPC2148

Give +3.3V power supply to LPC2148 Primer Board; connect the +5V adapter with GSM module which is connected with LPC2148 Primer Board through UART0. Open the Hyper Terminal screen, select which port you are using and set the default settings. Now the screen should show some text messages.

The following Commands and sequence of events performed for sending text message to a mobile phone through GSM Modem interfaced with microcontroller:

1. First select the text mode for SMS by sending the following AT Command to GSM Modem : AT+CMGF = 1 . This command configures the GSM modem in text mode.
2. Send the following AT Command for sending SMS message in text mode along with mobile number to the GSM Modem : AT+CMGS =+923005281046 . This command sends the mobile number of the recipient mobile to the GSM modem.

3. Send the text message string ("hello!") to the GSM Modem This is a test message from UART".

4. Send ASCII code for CTRL+Z i.e., 0x1A to GSM Modem to transmit the message to mobile phone. After message string has been sent to the modem, send CTRL+Z to the micro-controller, which is equivalent to 0x1A (ASCII value).

If you not reading any text from UART0, then you just check the jumper connections & just check the serial cable is working. Otherwise you just check the code with debugging mode in Keil. If you want to see more details about debugging just see the videos in below link.

➢ How to Create & Debug a Project in Keil.

General Information

- For proper working use the components of exact values as shown in Circuit file. Wherever possible use new components.
- Solder everything in a clean way. A major problem arises due to improper soldering, solder jumps and loose joints.
- Use the exact value crystal shown in schematic.
- More instructions are available in following articles,

  ➢ User Manual of LPC2148 Primer Board.
  ➢ Tutorial of how to create & Debug a project in KEIL.

Did you enjoy the read?

Pantech solutions creates information packed technical documents like this one every month. And our website is a rich and trusted resource used by a vibrant online community of more than 1,00,000 members from organization of all shapes and sizes.

Join the Technical Community Today!
http://www.pantechsolutions.net

What do we sell?

Our products range from Various Microcontroller development boards, DSP Boards, FPGA/CPLD boards, Communication Kits, Power electronics, Basic electronics, Robotics, Sensors, Electronic components and much more . Our goal is to make finding the parts and information you need easier and affordable so you can create awesome projects and training from Basic to Cutting edge technology.

Join the Technical Community Today!
http://www.pantechsolutions.net