**Name :**

**Roll no :**

**Group B Lab Assignment:** 11

**Subject :**PSDL

**Title :** PC to PC serial communication using UART.

**Assignment No: 11**

**Title :** Embedded C program for PC to PC serial communication using UART.

**Aim :** To write a C program to interface PIC18F4550 to PC using serial communication and transmit / receive characters over it.

**Experimental Setup:** MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Objective:**

* To Perform interfacing of real-world input and output devices to PIC18FXXX microcontroller.
* To write and execute an Embedded C program to PC using serial communication and transmit / receive characters over it.

**Theory:**

**1). UART Communication using PIC Microcontroller**

**“Addressable Universal Synchronous Asynchronous Receiver and Transmitter” shortly known as USART**.  USART is a two wire communication system in which the data flow serially. USART is also a full-duplex communication, means you can send and receive data at the same time which can be used to communicate with peripheral devices, such as CRT terminals and personal computers. The **USART** can be configured in the following modes:

* Asynchronous (full-duplex)
* Synchronous – Master (half-duplex)
* Synchronous – Slave (half-duplex)

There are also two different modes namely the 8-bit and 9-bit mode, in this tutorial we will configure the USART module to work in **Asynchronous mode with 8-bit communication system**, since it is the most used type of communication. As it is asynchronous it doesn't need to send clock signal along with the data signals. UART uses two data lines for sending (Tx) and receiving (Rx) data. The ground of both devices should also be made common. This type of communication does not share a common clock hence a common ground is very important for the system to work.

PIC18F4550 microcontroller has one (1) USART (Universal Synchronous/Asynchronous Receive/Transmit) module. This module can work in USRT mode or UART mode. In this topic we are going to use the USART module as UART (Universal Asynchronous Receive/Transmit) to transmit and receive data between the microcontroller and the computer.

PIC18F4550 has an in-built USART module which is useful for serial communication. With the help of USART, we can send/receive data to a computer or other devices. USART is also used in interfacing PIC with various modules like Wi-Fi (ESP8266), Bluetooth, GPS, GSM, etc.

We will see how the communication is established between PIC microcontroller and PC through USART using RS232 protocol. We will also see how to communicate with laptops, which do not have an RS232 DB9 port, and instead use a USB port.

Let us start with the serial communication using PIC18F4550.

**Asynchronous Communication:**PIC18F4550 has a built-in asynchronous receiver-transmitter. Asynchronous means each character (data byte) is placed in between the start and stop bits. The start bit is always 0 (low) and the stop bit is always 1 (high).

**Bit Rate & Baud Rate:**The rate of data transfer in serial data communication is stated in bps (bits per second). Another widely used terminology for bps is baud rate; means, a number of changes in signal per second. Here the signal is in bits, therefore bit rate = baud rate.

**Interface**: Although there are many pins in the DB9 connector, we do not need all. We only use pins RX, TX, and GND..

**PIC18F4550 UART Module Configuration Registers**

Now let’s start with Internal configuration registers and how to use these register to configure the UART communication module of PIC18F4550 microcontroller. There are three main control  registers associated with the PIC18F4550 microcontroller UART module:

* Transmit Status and Control (TXSTA)
* Receive Status and Control (RCSTA)
* Baud Rate Control (BACON)

We can control the complete data receiving and transmission operation of the UART module with these three registers. But if you want to use UART with interrupt, we need to configure some interrupt control registers also. bUT We see interrupt example in later

**UART Transmit Status and Control (TXSTA)**

This is an 8-bit register that is used to control and enable transmission features. Firstly, it selects data transfer type either synchronous or asynchronous. Secondly, it chooses the data frame size ( 8-bit or 9-bit).

**Bit7 CSRC Clock Source Select Bit**

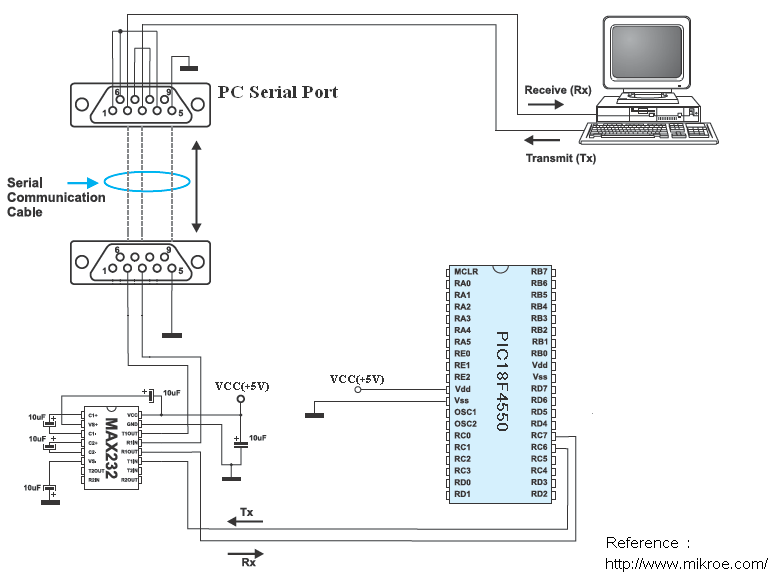
This bit selects the clock source input for the UART module. PIC18F4550 supports both synchronous and asynchronous modes of communication. Asynchronous serial

**Data Registers**

Other than control registers, four other registers are used for data storage, data transmission and for Baud rate generation. Like control registers, Data registers are also of 8-bit size.

* TXREG USART Transmit Register: It holds the data that the transmitter device wants to send to a receiver device.
* RCREG EUSART Receive Register: It stores data that PIC18F4550 microcontroller on RC7 pin.
* SPBRGH and SPBRG: These 8-bit registers are used to calculate the baud rate.

**PIC18F4550 UART connection circuit connection schematic:**  
Pin RC6 (TX) and pin RC7 (RX) are used for the UART (serial) communication between the microcontroller and the computer. To change between TTL logic levels (5V) and RS232 signals (+/-12V), an IC is needed which is max232.  
Don’t connect TX and RX pins directly to an RS232 serial port which may damage your microcontroller.



**PIC18F4550 UART connection circuit schematic diagram**

**Procedure:**

**Step1:** Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2:** Write the program in C language for interfacing PC to PIC18F4550 and sending ascii characters over serial communication. (in program properties make sure to add the 0x800 offset).

**Step3:** Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4:** Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC**.**

**Step5:** Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6:** Press reset button and execute the program.

**.**

**Source code :**

/\*Baud Rate GENERATION

\* n => required baudrate

\* BRGH = 0

\* SPBRG = (Fosc / (64 \* n)) -1

\* For 9600 baudrate, SPBRG ~=77

\*/

#include<p18F4550.h>

#include<stdio.h>

#define Fosc 48000000UL

void InitUART(unsigned int baudrate)

{

TRISCbits.RC6 = 0; //TX pin set as output

TRISCbits.RC7 = 1; //RX pin set as input

SPBRG = (unsigned char)(((Fosc /64)/baudrate)-1);

BAUDCON = 0b00000000; //Non-inverted data; 8-bit baudrate generator

TXSTA = 0b00100000; //Asynchronous 8-bit; Transmit enabled; Low speed baudrate select

RCSTA = 0b10010000; //Serial port enabled; 8-bit data; single receive enabled

}

void SendChar(unsigned char data)

{

while(TXSTAbits.TRMT == 0); //Wait while transmit register is empty

TXREG = data; //Transmit data

}

void putch(unsigned char data)

{

SendChar(data);

}

unsigned char GetChar(void)

{

while(!PIR1bits.RCIF); //Wait till receive buffer becomes full

return RCREG; //Returned received data

}

void main(void)

{

InitUART(9600);

printf("\r\nHello MicroPIC-18F: Enter any Key from Keyboard\r\n");

while(1)

{

printf("%c! ",GetChar()); //Receive character from PC and echo back

}

while(1);

}

**Result:** Connect the PC to the board using the USB cable. Start a terminal program on the PC (tera Term, Putty, hyperterminal) with the specified baud rate (9600). Check if you are getting the transmitted characters from the board and back

**Conclusion:** Thus, we have studied C program to interface PIC18F4550 to PC using serial communication (UART) and transmit / receive characters over it.