10

CONTROL UNIT IN OPTICAL TOMOGRAPHY

Ruzairi Abdul Rahim
Siti Zarina Mohd Muji
Mohd Hafiz Fazalul Rahiman

10.1 INTRODUCTION

Control unit is important part in any process tomography. This unit will ensure the smoothness of overall system. Knowledge in programming the microcontroller is really needed to ensure the success of any project. This section will give the reader some knowledge about RS232 communication using PIC18F4520 and C language as a software tools to program the code.

10.2 FUNDAMENTAL ABOUT OPTICAL TOMOGRAPHY AND MICROCONTROLLER

The general principal is a set of light sources and the photo detectors are used into obtaining the parallel views of the pipeline. This type of tomography is popular for medical and process tomography. The recent research right now is focusing more on medical side rather than the process. Optical tomography is a “hard field” type tomography where it is sensitive to the parameters that they measure in all position of measurement volume and it is very easy when we want to get the data because the sensitivity is equally in all positions. It is however the opposite with “soft field” where, parameter sensitivity is dependent on the position of the sensors in the measurement volume. In optical, a

beam of light will be projected through some medium from one boundary point and this light will be detected at another boundary point. At the receiving point, the level of voltage will be measured and any decreasing of the value, is proportional to the existing of object in the pipe or vessel. It means the optical tomography detects the attenuation of the signal.

Microcontroller is not new in the design of process tomography. There are many types of microcontroller can be found such as Motorola, Intel, Atmel and Microchip. In this book, microcontroller from Microchip will be used which is has the unique name; Peripheral Interface Controller (PIC). PIC18F4520 will be focused in this design because the characteristics as follows [1]:

- It has 40 pins : More pins is needed in the optical design because usually optical need more projection therefore, need more sensors to operate and controlled by microcontroller
- It has maximum 10 bit analog to digital pin that is important to make the conversion from analog to digital in optical tomography
- It has Rx and Tx pin for communication purpose whether between microntrollers or between microcontroller and Personal Computer.
- It also has huge amount of memory to store programming code and data. For program memory, it has 32 Kbytes of Flash memory and can store up to 16,384 single-word instructions

10.3 RS232 AND ITS CONCEPTS

Before go through, let we see the concept of RS232. RS232 is the interaction using serial communication and it's not needed many wires. The simplest one is only use three wires for transmit, receive and ground. The maximum distance it can goes is 15m and data rates from 50 to 76800 baud (bits per second) [2]. It characteristic is different with parallel where, parallel only used for short distance application and used many wires to transfer data in parallel. But, parallel has the advantages of fast transferring of data compared to serial communication.

10.4 RS232 AND PIC18F4520

In PIC18F4520, there are two pins that is really important for RS232 communication; Tx and Rx pin. Tx pin is shared with RC6 input output pins, whereas Rx pin is shared with RC7 input and output pin. To make a simple communication between microcontroller and any laptop we only need three pins; Rx, Tx and Ground.

Today, laptop has no serial communication; therefore we need to use a converter to make this communication happen.

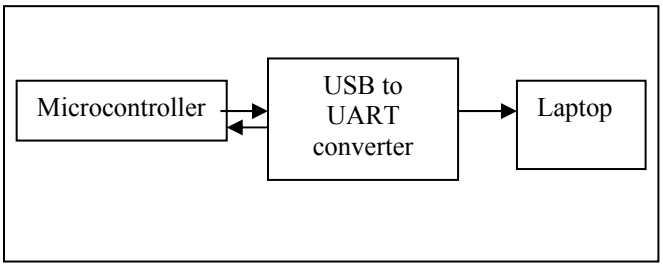

Figure 10.1 Communication between microcontroller and laptop

Before setup the connection in Figure 10.1, we need to test whether the converter is functioning or not, therefore simple steps like below will be implemented:

a) Connect the USB to UART converter to laptop
b) Two wires from USB converter, which is transmitter and receiver, will be touched each other.
c) If any character is typed, it should appear in the hyperterminal in the laptop and it shows the USB converter is working properly.

Only then we can test our first experiment as shown in Figure 10.1.

10.5 MICROCONTROLLER AND LAPTOP COMMUNICATION

In this section, we will see how to make communication as shown in Figure 10.2. At this experiment we will send 'A' character to Laptop and see the result at hyperterminal.

a) Connect the USB to UART converter to laptop
b) Connect the microcontroller to USB converter using three pins which are transmitter pin, receiver pin and ground as shown in Figure 10.2.

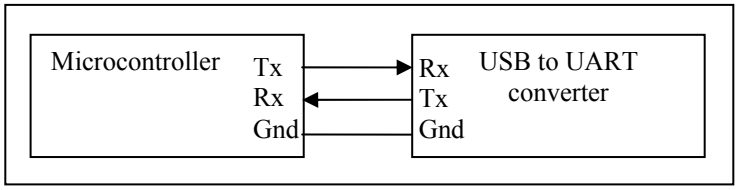

Figure 10.2 Connection between microntroller and USB converter

c) Program the microcontroller to send the 'A' character.
d) Hyperterminal will shows 'A' character.

10.5.1 Programming Using C Languange

There are two types of C language that always been used by researchers, which are Micro C and MCC18 (MPLAB C Compiler for PIC18). For Micro C the difficulty occurs when we want to burn the chip. Sometime it would not work correctly and will make the PIC become hot. It happens because there is a conflict in the burning process because MPLAB burner is used instead of the burner from Micro C itself. Therefore, we will not use Micro C to prevent the problem as mentioned. MCC18 will be used because it is compatible with the MPLAB burner.

10.5.2 Step to program the microcontroller:

a) Initialization
The initialization is important to make sure the program setting the correct PIC microcontroller and other configuration. The source code below is the ordinary setting for the PIC microcontroller.

```
#include <p18f4520.h>       //Select PIC18F4580
#pragma config OSC=HS //select High Speed Oscillator
#pragma config WDT=OFF //disable watchdog timer
#pragma config LVP=OFF //disable low voltage programming
```

b) Set the bit in the specific register and the recommendation to ease the setting

i) Set 3 bits in Transmit Status and Control (TXSTA) register
a) Select the baudrate using High Baud Rate Select (BRGH) bit. Select low baudrate
b) Select how many bit to transfer using 9-bit Transmit Enable (TX9) bit. Select 8 bit.
c) Enable the transmit prcocess bye set the Transmit Enable bit (TXEN).

ii) Set the baudrate and clock in EUSART Baud Rate Generator Register, Low Byte (SPBRG:). It will set for baudrate and the clock. Table 1.1 shows how to set the baudrate and clock. If we use 9600 baudrate and crystal clock equal to 20MHz. Therefore, the 31 is being selected. 31 is in decimal, and when it is converted to hexadecimal we will get 1F.

iii) when it is converted to hexadecimal we will get 1F.

Table 10.1 The way to set the SPBRG register

BAUD RATE (K)	SYNC = 0, BRGH					
	Fosc = 40.000 MHz			Fosc = 20.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255
2.4	2.441	1.73	255	2.404	0.16	129
9.6	9.615	0.16	64	9.766	1.73	31
19.2	19.531	1.73	31	19.531	1.73	15
57.6	56.818	-1.36	10	62.500	8.51	4
115.2	125.000	8.51	4	104.167	-9.58	2

iv) Enable the serial port communication by setting 1 in Serial Port Enable (SPEN) bit. This bit is in Receive Status and Control (RCSTA) register.

The style to write is same as C language where the right side is the value to be transferred to left side as shown in the source code below.

```
void main (void)
{
    TXSTA = 0x20;
    SPBRG = 0x1F;
```

(c) Program the code to transmit A character;

i) Insert C character into EUSART Transmit Register (TXREG) as shown below

```
TXREG = 'C';
```

ii) Monitor the changing of EUSART Transmit Interrupt Flag (TXIF) bit in the Peripheral Interrupt Request (Flag) 1 (PIR1) register. If TXIF is equal to '0', the data is not finishing transmitting.

```
wait:
    if(PIR1bits.TXIF == 0)
        goto wait;
```

After finish the code programming, test whether C character is appeared in the Hyperterminal. If the programming is succes you can see 'C' character in the hyperterminal display continuously.

10.6 MICROCONTROLLER AND MICROCONTROLLER COMMUNICATION

After we success transmit 'C' character to the laptop, we can continue with transmit a character to other microcontroller a shown in Figure 10.3

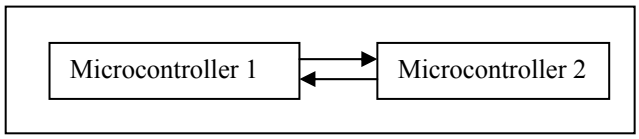

Figure 10.3 The connection using two microcontrollers

Here, there are two programming code must be written; transmitter and receiver code. For the transmiter code, the program is the same as explain before. Now, we will look on how to program the receiver side. The step to program the receiver is stated below:

a) Initialization
   The initialization is same as in the transmitter side.
b) Set the bit in the specific register and the recommendation to ease the setting
   i) Set 2 bits in Receive Status and Control (RCSTA) register
      a) Enable the serial port by enabling the Serial Port Enable (SPEN) bit.
      b) Enable the Continuous Receive Enable (CREN) bit.
   ii) Set SPBRG register same as the transmitter's setting.

c) Monitor whether the receiver (microcontroller) has received the data or not using EUSART Receive Interrupt Flag (RCIF) bit in the PIR1 register.

d) To test whether the correct data has been received or not, EUSART Receive Register (RCREG) will be used with if statement as shown below.

```
if (RCREG == 'C')
PORTB = 0x23;
else if (RCREG == 'B')
PORTB = 0xff;
```

10.7 PROTEUS

To make sure the programming is correct; the connection in the Proteus will be established as shown in Figure 10.4.

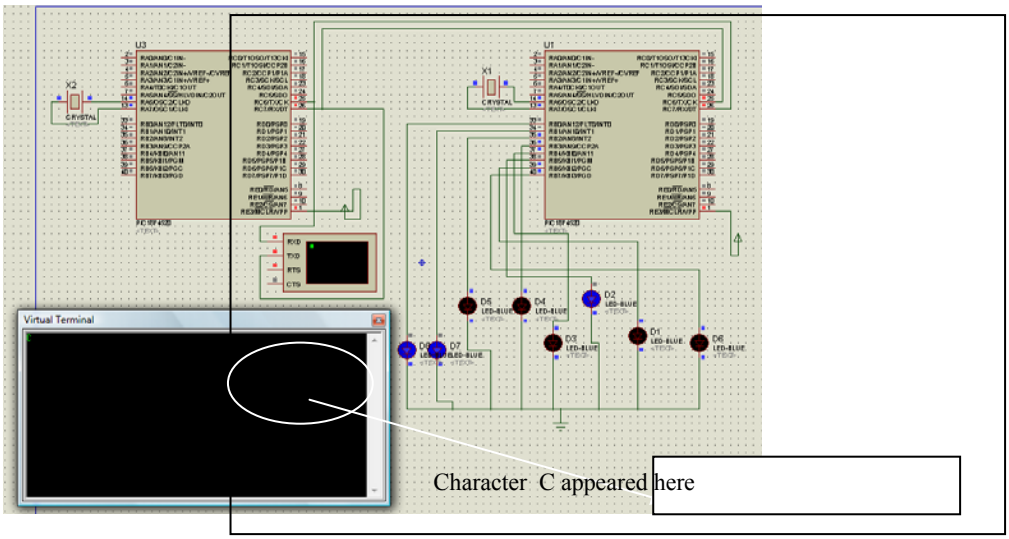

Figure 10.4 Overall connection in PROTEUS

Virtual terminal has been added to replace the hyperterminal function in the laptop. As the transmitter transmits the character C, the virtual terminal will show the C at the black screen. It shows that the character

has been successfully transmitted from the transmitter. Then at the receiver side, if the character C is successfully received, the binary 00100011 will appear as set in the receiver part. These LEDs are to make sure the microcontroller receives the character.

REFERENCES

[1] Microchip (2001). *"PIC18F2420/2520/4420/4520 Data Sheet"*

[2] Mike James. (2001). *"Microcontroller Cookbook. PIC & 8051"* . 2nd Edition. Newness. Great Britain.