

EXPERIMENT VII



Fall - 2022/2023

MKT3811 - Microprocessors and Programming

Lab 7 Report

Submitted By: Göktuğ Can Şimay

Lab Partners: Ali Doğan, Basel Hadri

Group Number: 22

Student ID: 22067606

Date: 17.12.2022

Descriptions:

The purpose of this experiment is serial communication of two different microcontrollers in C language.

The experiment can be summarized as follows:

- Suppose that you need to control a system with PID controller. However, it is not always easy

to change the PID gain parameters, which are embedded to the code statically. For this reason,

it would be better to alter the parameters with a serial communication in order to see the effect

rapidly in real-time.

- You should send a string array with combination of your student ID via serial communication

(your student ID is represented as “a b c d _ e f g”).

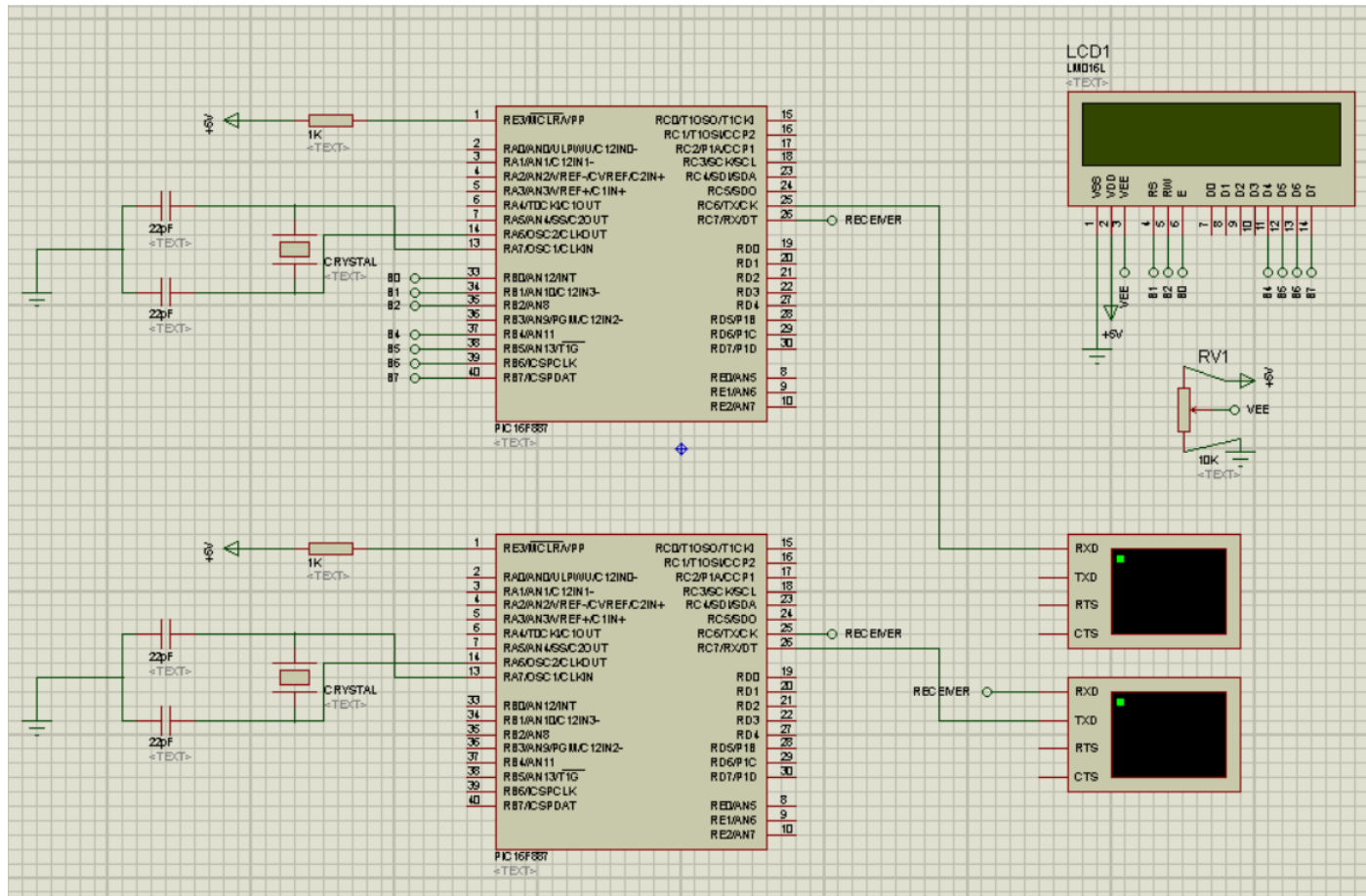
- The message you will send from serial writer to serial reader is defined as “afPdgIdfD”. For

example; if your student ID is 12345678, then you should send 17P48I47D.

- These characters should be sent with the second microcontroller that represents the serial input

line for simulation purpose. In your serial reader code, these characters should be parsed to P,

I and D parameters as 17, 48 and 47, respectively.



Proteus Schematic

As you can see, I used PIC16F887 microcontroller. I supplied +5V power to the Vpp terminal. Since I want to use an external oscillator, I connected my oscillator to the 13th and 14th inputs via capacitors.

```
//lab07_code -- Alic1
#include <16f887.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay>(clock=4M)
#FUSES XT
#FUSES NOWDT //No Watch Dog Timer
#FUSES NOPROTECT //Code not protected from reading
#FUSES NOBROWNOUT //No brownout reset
#FUSES NOPUT //No Power Up Timer
#FUSES NOCPD //No EE protection
#FUSES NODEBUG //No Debug mode for ICD
#define use_portb_lcd TRUE
#include <lcd.c>
#include <rs232.h>(baud=9600, xmit=pin_C6, rcv=pin_C7, parity=N, stop=1)

unsigned long inputString;
int counter=0;
int counter2=0;
char strInput[4];

void main() {
    lcd_init();
    delay_ms(10);

    while(1){
        if(kbhit()){
```

```
char i=getc();
if(i=='*'){
    if(counter2==0){
        inputString=atol(strInput);
        printf("Kp:%lu",inputString);
        printf(LCD_PUTC, "\fKp:%lu",inputString);
        counter=0;
        memset(strInput,0,8);
        counter2++;

    }else if(counter2==1){
        inputString=atol(strInput);
        printf("Ki:%lu",inputString);
        printf(LCD_PUTC, "\fKi:%lu",inputString);
        counter=0;
        memset(strInput,0,8);
        counter2++;

    }else if(counter2==2){
        inputString=atol(strInput);
        printf("Kd:%lu",inputString);
        printf(LCD_PUTC, "\fKd:%lu",inputString);
        counter=0;
        memset(strInput,0,8);
        counter2=0;

    }
}
```

```
    }else{
        strInput[counter]=i;
        counter++;
    }
}
}
}
```

```
//lab07_code -- Verici
#include <16f887.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay>
#define delay(clock=4M)
#define FUSES XT
#define FUSES NOWDT //No Watch Dog Timer
#define FUSES NOPROTECT //Code not protected from reading
#define FUSES NOBROWNOUT //No brownout reset
#define FUSES NOPUT //No Power Up Timer
#define FUSES NOCPD //No EE protection
#define FUSES NODEBUG //No Debug mode for ICD
#define use_portb_lcd TRUE
#include <lcd.c>
#include <rs232.h>
#define rs232 (baud=9600, xmit=pin_C6, rcv=pin_C7, parity=N, stop=1)

void main() {

    while (1) {

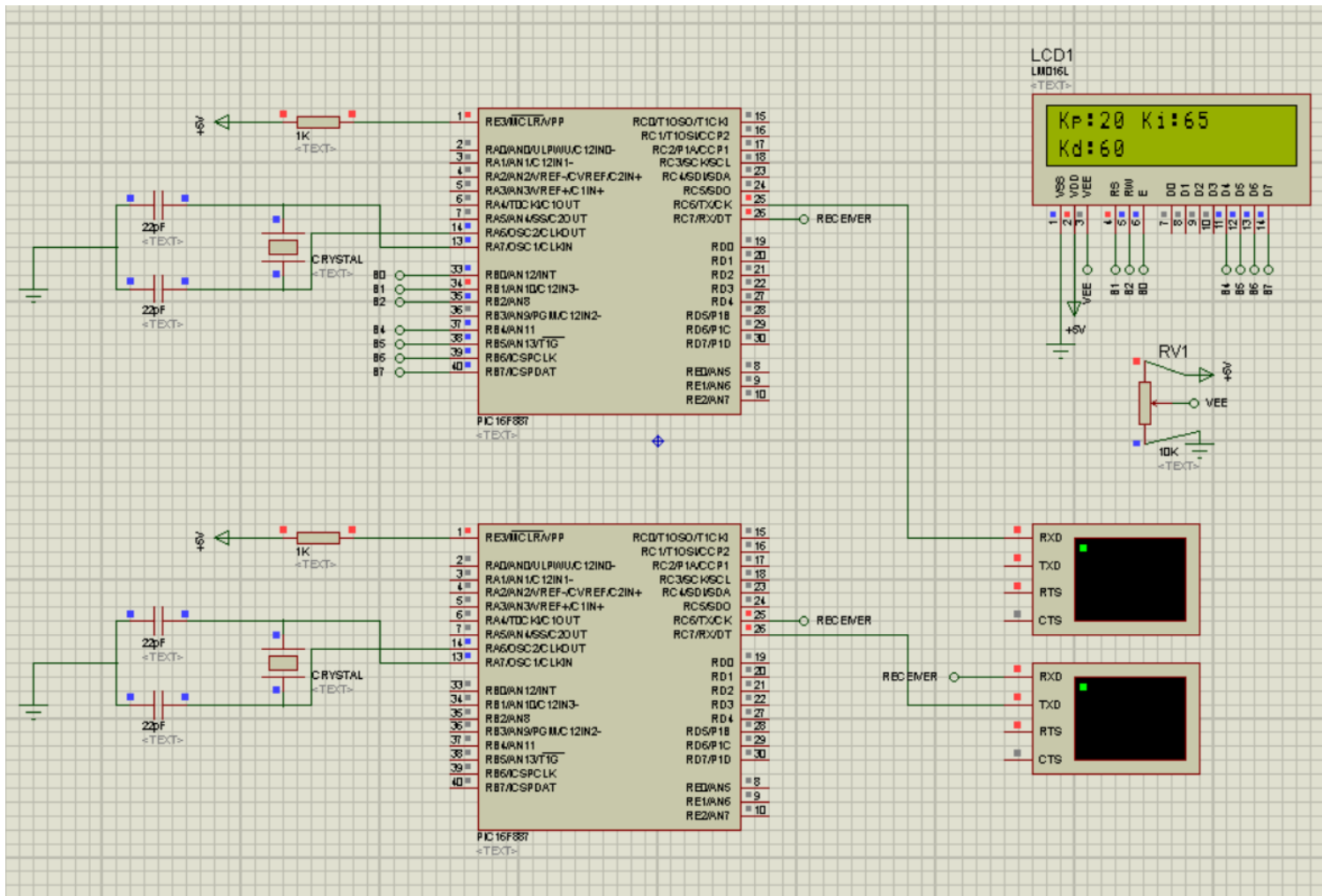
        printf("2");
        delay_ms(500);
        printf("0");
        delay_ms(500);
        printf("*");
        delay_ms(500);
    }
}
```

```
printf("6");  
delay_ms(500);  
printf("5");  
delay_ms(500);  
printf("*");  
delay_ms(500);  
printf("6");  
delay_ms(500);  
printf("0");  
delay_ms(500);  
printf("*");  
delay_ms(2000);
```

```
}
```

```
}
```

```
}
```

Proteus Simulation