**信号发生器**

一、任务要求

（1）参考例4-3、例5-5、例6-1、例7-1、例8-1，设计一个信号发生器；

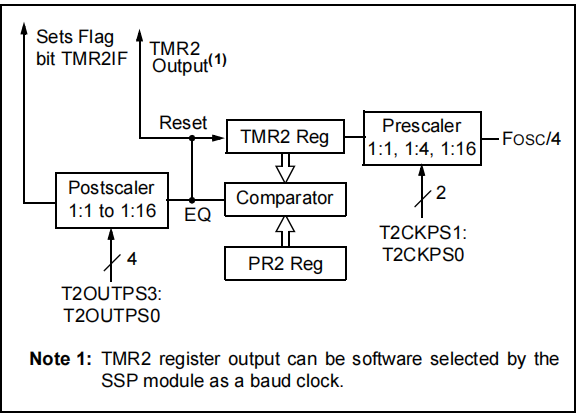
（2）通过4X4键盘选择输出信号的频率，LCD显示按键号和当前输出信号的频率值，在仿真软件中用示波器及蜂鸣器能分辨出不同频率信号的输出，下载到开发板，验证显示和声音；

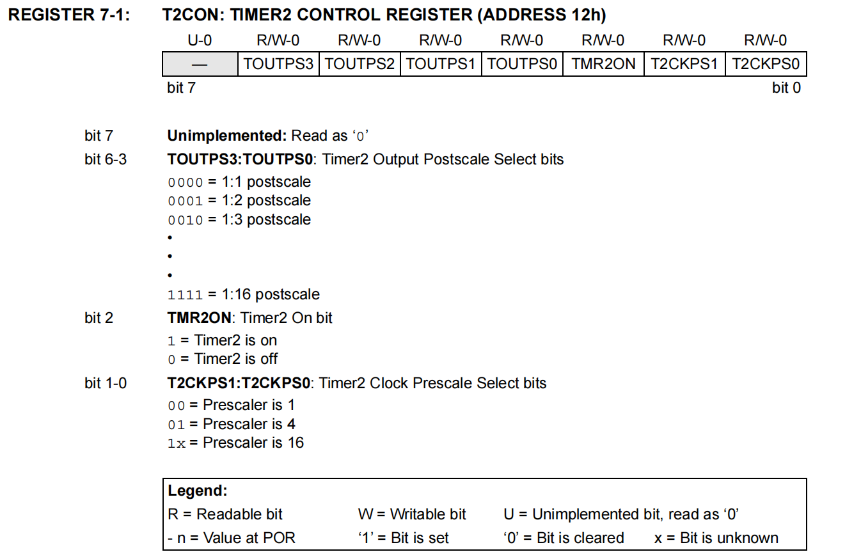
（3）要求输出信号在0.1~1kHz之间，必须具备的频率包括0.1、1、10、40、100、1k等，可利用仿真软件中的频率计（Counter Timer）检测频率。

二、实验原理

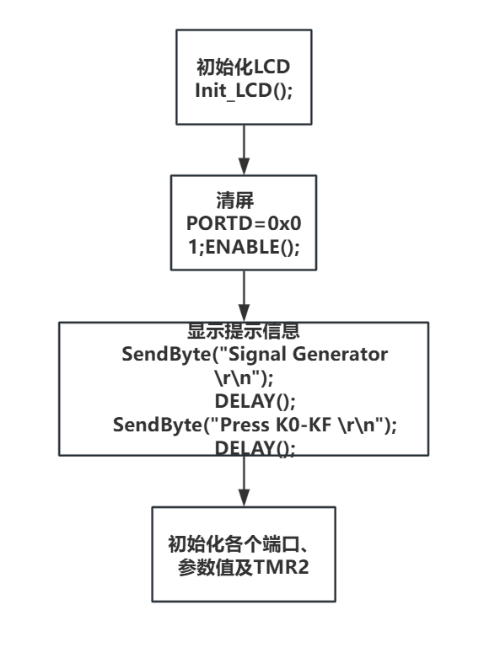
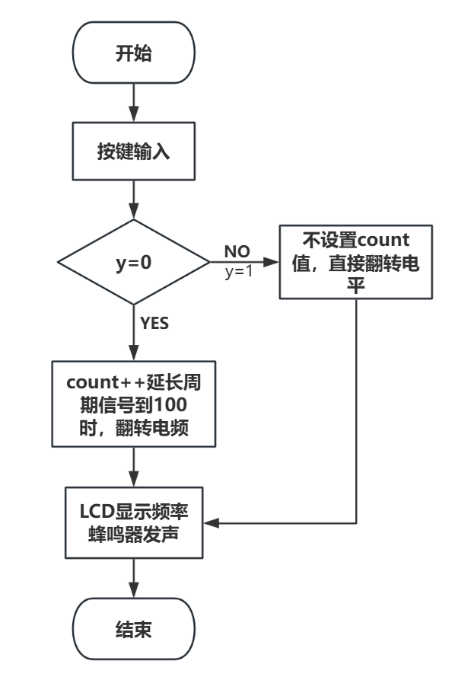
本次实验使用的是TMR2，且开发板内部设计蜂鸣器对应的引脚为RE0，所以利用该模块没有与之相关的引脚，便于使用TMR2产生信号驱动蜂鸣器。

由于需要产生0.1Hz~1kHz这样既包含低频由又包含高频的信号，需要采用合适的分频。本次实验采用的是预分频和后分频都为16，以满足要求。如图所示，为TMR2的结构图和控制寄存器。



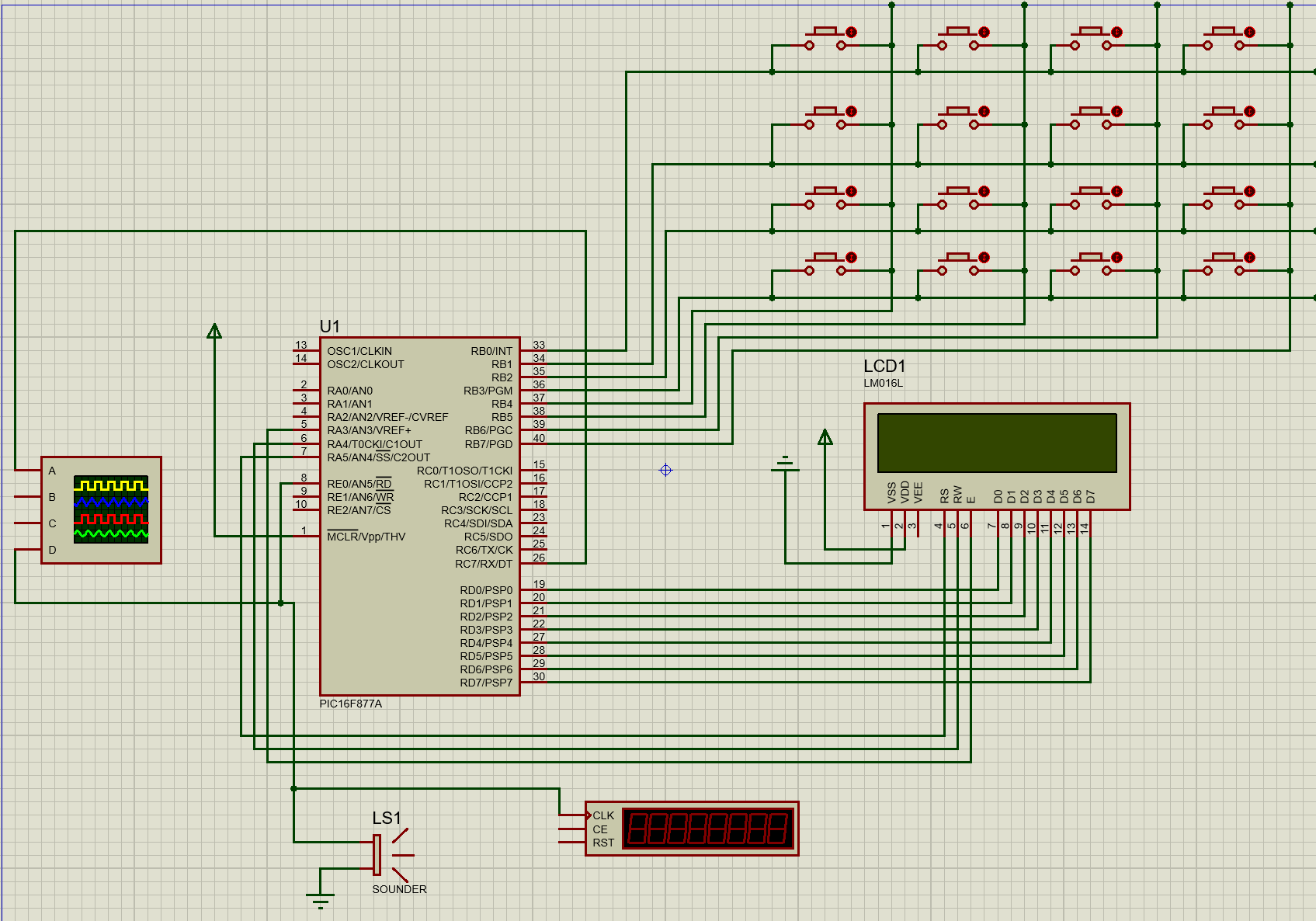


主程序流程及低频/高频信号处理框图：

三、工程实现

（1）电路图实现



（2）代码实现

#include<pic.h>

\_\_CONFIG(0xFF29);

*/\**

    \*  PIC16F877AA  4MHz  8位单片机

    \*  LCD1602显示模块

    \*  按键输入，产生不同频率的方波，使蜂鸣器发出不同的声音

    \*  波形频率范围：MIN：0.1Hz，MAX：1kHz

\*/

#define SIGNAL\_PIN RE0    *// 信号输出/蜂鸣器驱动引脚（RE0）*

#define SIGNAL\_DDR TRISE0 *// 方向控制：0=输出*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LCD显示模块 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

#define rs RA5

#define rw RA4

#define e RA3

static volatile char TABLE[16]={0x30,0x31,0x32,0x33,0x34,

                                0x35,0x36,0x37,0x38,0x39,

                                0x41,0x42,0x43,0x44,0x45,0x46}; *// 0-9 A-F*

unsigned char string1;

unsigned char string2;

char adh,adl,a,b,result,preres;

unsigned int dat,y;

unsigned char count;

*// 延时子程序*

void DELAY() {

    unsigned int i;

    for(i=999;i>0;i--);

}

*// 写入控制命令的子程序*

void ENABLE() {

    rs=0;

    rw=0;

    e=0;

    DELAY();

    e=1;

}

*// 写入字的子程序*

void ENABLE1() {

    rs=1;

    rw=0;

    e=0;

    DELAY();

    e=1;

}

*// LCD初始化函数*

void Init\_LCD() {

    TRISD=0;

    ADCON1=7;

    TRISA=0;

    RA1=0;

    PORTD=0;

    e=1;

*// 调用延时，刚上电LCD复位不一定有PIC快*

    DELAY();

}

*// 字符显示函数*

void SendByte(unsigned char \*str) {

*//PORTD =1;ENABLE(); //清屏，调延时，因为LCD是慢速器件*

    PORTD=0x38;ENABLE();*//8位2行5x7点阵*

    PORTD=0x0C;ENABLE(); *//显示器开、光标关、闪烁关*

    PORTD=0x06;ENABLE(); *//文字不动，光标自动右*

    PORTD=0x80;ENABLE(); *//光标指回第一行*

    while(\*str != '\0') {

        PORTD = \*str++;

        ENABLE1();

    }

}

*// LCD显示函数*

void Display\_LCD() {

    switch(result) {

        case 0xEE: dat = (5000/0.1);y=0; SendByte("K0:0.1Hz \r\n");break; *//K0*

        case 0xED: dat = (5000/1);y=0; SendByte("K1:1Hz \r\n");break; *//K1*

        case 0xEB: dat = (5000/10);y=0; SendByte("K2:10Hz \r\n");break; *//K2*

        case 0xE7: dat = (500000/40); y=1; SendByte("K3:40Hz \r\n");break; *//K3*

        case 0xDE: dat = (500000/50); y=1; SendByte("K4:50Hz \r\n");break; *//K4*

        case 0xDD: dat = (500000/1000);y=1;SendByte("K5:1kHz \r\n");break; *//K5*

        case 0xDB: dat = (5000/0.5);y=0;SendByte("K6:0.5Hz \r\n");break; *//K6*

        case 0xD7: dat = (5000/2);y=0;SendByte("K7:2Hz \r\n");break; *//K7*

        case 0xBE: dat = (5000/4);y=0;SendByte("K8:4Hz \r\n");break; *//K8*

        case 0xBD: dat = (5000/5);y=0;SendByte("K9:5Hz \r\n");break; *//K9*

        case 0xBB: dat = (500000/20);y=1;SendByte("KA:20Hz \r\n");break; *//KA*

        case 0xB7: dat = (500000/80);y=1;SendByte("KB:80Hz \r\n");break; *//KB*

        case 0x7E: dat = (500000/100);y=1;SendByte("KC:100Hz \r\n");break; *//KC*

        case 0x7D: dat = (500000/200);y=1;SendByte("KD:200Hz \r\n");break; *//KD*

        case 0x7B: dat = (500000/400);y=1;SendByte("KE:400Hz \r\n");break; *//KE*

        case 0x77: dat = (500000/500);y=1;SendByte("KF:500Hz \r\n");break; *//KF*

    }

    ENABLE1();

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 按键模块 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

void SweepKeyboard() {

    PORTB = 0X7f; *//RB7输出低电平，其他三位输出高电平*

    asm("nop"); *// 插入一定延时，确保电平稳定*

    result = PORTB; *// 读回B口低4位结果*

    result = result & 0x0f; *// 清除高4位*

    if (result != 0x0f) { *// 判断低4位是否为全1(全1代表没按键按下)*

        result = result | 0x70; *// 否，加上高4位0x70，做为按键扫描的结果*

    } else {

        PORTB = 0Xbf; *//RB6输出低电平，其他三位输出高电平*

        asm("nop"); *// 插入一定延时，确保电平稳定*

        result = PORTB; *// 读回B口高低4位结果*

        result = result & 0x0f; *// 清除高4位*

        if (result != 0xf) { *// 判断低4位是否为全1(全1代表没按键按下)*

            result = result | 0xb0; *//否，加上高4位0xb0，做为按键扫描的结果*

        } else {

            PORTB = 0Xdf; *//RB5输出低电平，其他三位输出高电平*

            asm("nop"); *// 插入一定延时，确保电平稳定*

            result = PORTB; *// 读回B口低4位结果*

            result = result & 0x0f; *// 清除高4位*

            if (result != 0x0f) { *// 判断低4位是否为全1(全1代表没按键按下)*

                result = result | 0xd0; *//否，加上高4位0xd0，做为按键扫描的结果*

            } else {

                PORTB = 0Xef; *//B4输出低电平，其他三位输出高电平*

                asm("nop"); *// 插入一定延时，确保电平稳定*

                result = PORTB; *// 读回B口低4位结果*

                result = result & 0x0f; *//清除高4位*

                if (result != 0x0f) { *//判断低四位是否为全1(全1代表没有按键按下)*

                    result = result | 0xe0; *//否，加上高4位0x0e，做为按键扫描的结果*

                } else { *//是，全部按键扫描结束，没有按键按下，置无按键按下标志位*

                    result = 0xff; *//扫描结果为0xff，做为没有按键按下的标志*

                }

            }

        }

    }

    if(result==0xff)*//无键按下显示‘X’*

        result=preres;

    else

        preres=result;

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 定时器模块 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*// 初始化TMR2函数*

void Init\_tmr2() {

*//TRISC7=0;*

    SIGNAL\_PIN = 0;       *// 初始低电平*

    SIGNAL\_DDR = 0;

    GIE=1;

    PEIE=1;

    TMR2IE=1;

    TMR2IF=0;

    T2CON = 0B01111111;

    PR2 = dat/256;

}

void interrupt tmr2() {

    TMR2IF =0;

    PR2 = dat/256;

    if(y==0) {

        count++;

        if(count == 100) {

*//RC7 = !RC7;*

            SIGNAL\_PIN = !SIGNAL\_PIN; *// 翻转RE0引脚，驱动蜂鸣器/信号输出*

            count =0;

        }

    } else {

*//RC7 = !RC7;*

        SIGNAL\_PIN = !SIGNAL\_PIN; *// 翻转RE0引脚，驱动蜂鸣器/信号输出*

    }

}

void main() {

*// 初始化LCD*

    Init\_LCD();

*// 清屏*

    PORTD=0x01;ENABLE();

    DELAY();

*// 显示提示信息*

    SendByte("Signal Generator \r\n");

    DELAY();

    SendByte("Press K0-KF \r\n");

    DELAY();

    TRISB=0X0F;

    nRBPU=0;

    result=0x00;

    preres=0x00;

*//TRISC=0B01111111;//RC7为输出口，其他为输入口*

    dat = 50000;

*//初始化RE0端口*

    SIGNAL\_DDR = 0;       *// RE0设为输出*

    ADCON1 = 7; *// 设置为数字输入*

    TRISE0 = 0; *// 设置RE0为输出模式*

    Init\_tmr2();

*// 主循环*

    while(1) {

        SweepKeyboard();

        Display\_LCD();

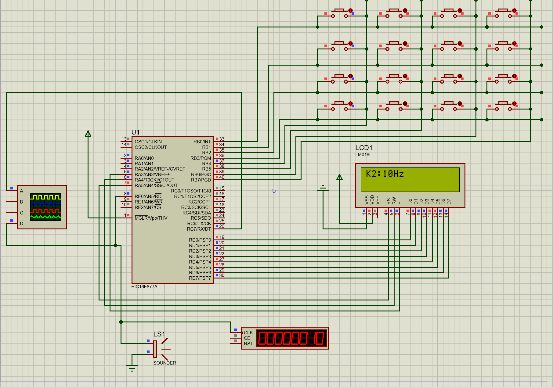
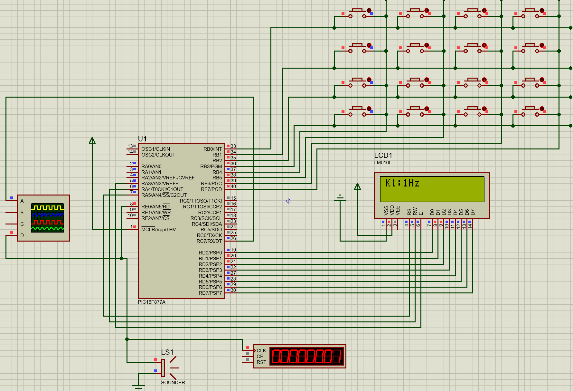
    }

}

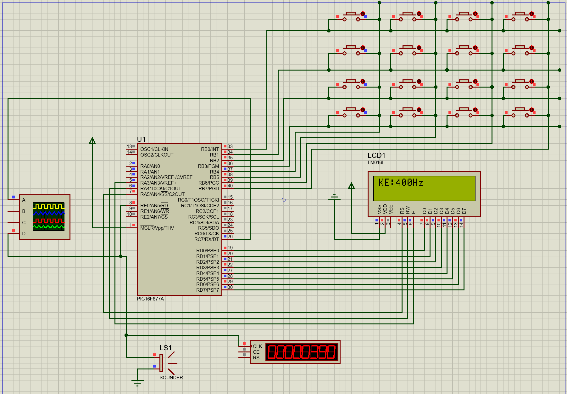
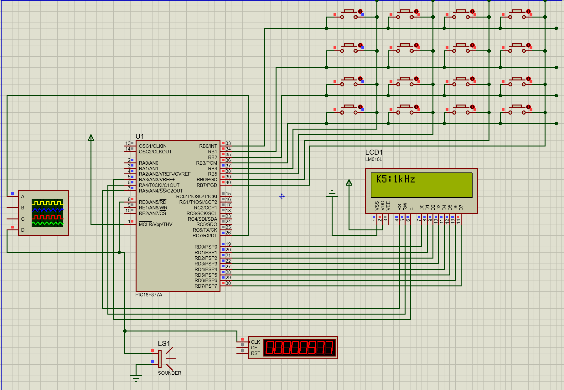
四、实验现象

（1）电路仿真

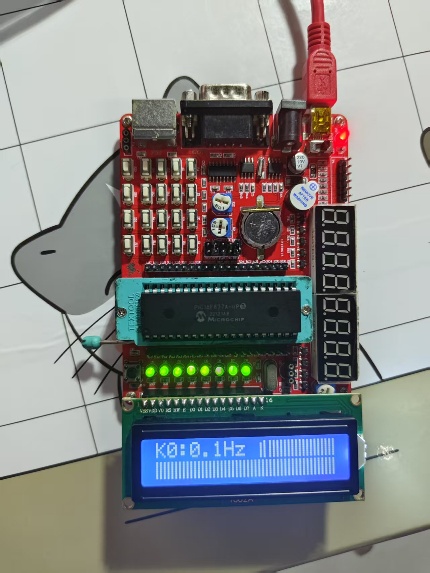
如图所示分别为仿真软件中频率计所测得的1Hz和10Hz的显示：

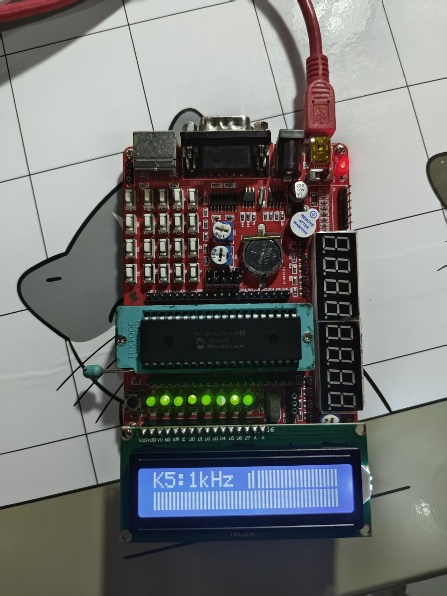
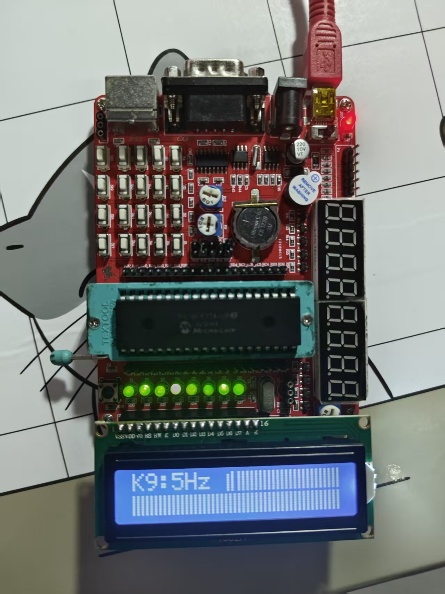


如图所示分别为仿真软件中频率计所测得的1kHz和400Hz的显示：



（2）硬件现象





五、实验总结及心得

（1）实验总结

本次实验成功设计了基于PIC16F877A单片机的信号发生器，实现了 0.1Hz~1kHz 频率范围内的方波输出。通过 4×4 键盘选择频率，LCD显示按键号及当前频率，经仿真软件验证，在误差允许范围内，示波器和频率计准确反映信号特性，硬件调试中蜂鸣器能分辨不同频率声音。核心利用 TMR2 定时器，通过预分频和后分频均为 16 的配置，结合计数器逻辑，有效处理了高低频信号的生成问题。

（2）实验心得

通过本次实验，进一步理解了单片机定时器的配置。TMR2 的预分频和后分频配置是实现宽频率范围的关键，而计数器与中断的配合解决了低频信号周期过长的问题。此外，代码的模块化设计有利于提高其可读性，以及后续的拓展开发。