```
/*
 * File:   main1509.c
 * Author: valky
 *
 * keyer1509_4.c
 * Created on 2020/02/05
 */


#include <xc.h>

#pragma config FOSC = INTOSC    // Oscillator Selection (INTOSC oscillator: I/O function on CLKIN pin)
#pragma config WDTE = OFF       // Watchdog Timer Enable (WDT disabled)
#pragma config PWRTE = OFF      // Power-up Timer Enable (PWRT disabled)
#pragma config MCLRE = ON      // MCLR Pin Function Select (MCLR/VPP pin function is digital input)
#pragma config CP = OFF         // Flash Program Memory Code Protection (Program memory code protection is disabled)
#pragma config BOREN = ON       // Brown-out Reset Enable (Brown-out Reset enabled)
#pragma config CLKOUTEN = OFF   // Clock Out Enable (CLKOUT function is disabled. I/O or oscillator function on the CLKOUT pin)
#pragma config IESO = OFF       // Internal/External Switchover (Internal/External Switchover mode is disabled)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable (Fail-Safe Clock Monitor is disabled)
// CONFIG2
#pragma config WRT = OFF        // Flash Memory Self-Write Protection (Write protection off)
#pragma config STVREN = OFF     // Stack Overflow/Underflow Reset Enable (Stack Overflow or Underflow will not cause a Reset)
#pragma config BORV = LO        // Brown-out Reset Voltage Selection (Brown-out Reset Voltage (Vbor), low trip point selected.)
#pragma config LVP = OFF        // Low-Voltage Programming Enable (High-voltage on MCLR/VPP must be used for programming)

void dash();
void dot();
void morsea();
void morseb();
void morsec();
void morsed();
void morsee();
void morsef();
void morseg();
void morseh();
void morsei();
void morsej();
```

```
void morsek();
void morsel();
void morsem();
void morsen();
void morseo();
void morsep();
void morseq();
void morser();
void morses();
void morset();
void morseu();
void morsev();
void morsew();
void morsex();
void morsey();
void morsez();
void morse1();
void morse2();
void morse3();
void morse4();
void morse5();
void morse6();
void morse7();
void morse8();
void morse9();
void morse0();
void morseperi();
void morsecomm();
void morseque();
void morsear();
void morsebt();
void morsekn();
void morsesla();
void wait(int n);
void tone(int n);
void output0();

void jh1zvp();
void kamakura();
```

```
void keyer();
void pract();

//
// __delay_ms()用  クロック  1 MH z
#define _XTAL_FREQ 1000000

int n,n3,n4;
int tau=500;
int tau2=1000;

void main() {

    // PIC 設定
    OSCCON = 0b01011010;  // 内部発信器  1MHz
    TRISA  = 0x3E; //0x3E
    TRISB  = 0x30; //0x30
    TRISC  = 0xFF; //0xFF
    ANSELA = 0x2; //0x2
    ANSELB = 0x0; //0x0
    ANSELC = 0x0; //0x0
    WPUA   = 0x3F; //0x3F
    WPUB   = 0xF0; //0xF0
    ADCON0 = 0b00000101;
    ADCON1 = 0b10000000;

   // 初期設定
    LATA = 0b00000000;
    LATB = 0b00000000;
    LATC = 0b00000000;

    GO = 1;
    while(GO);
```

```
    n  = ADRES/8;
    n3 = n+n+n;
    n4 = n3+n;


    // キーヤーモード (RC0=0)  と  練習モード(RC0=1)
    if(RC0==0)       keyer(n,n3);
    if(RC0==1)       pract(n,n3);

}



//
void keyer(n,n3){   // メモリーキーヤー・モード

    if(RA4==0){
        morsea(n,n3);
        morseb(n,n3);
        morsec(n,n3);
        morsed(n,n3);
        morsee(n,n3);
        morsef(n,n3);
        morseg(n,n3);
        morseh(n,n3);
        morsei(n,n3);
        morsej(n,n3);
        morsek(n,n3);
        morsel(n,n3);
        morsem(n,n3);
        morsen(n,n3);
        morseo(n,n3);
        morsep(n,n3);
        morseq(n,n3);
```

```
    morser(n,n3);
    morses(n,n3);
    morset(n,n3);
    morseu(n,n3);
    morsev(n,n3);
    morsew(n,n3);
    morsex(n,n3);
    morsey(n,n3);
    morsez(n,n3);
}

if(RA5==0){
    morse1(n,n3);
    morse2(n,n3);
    morse3(n,n3);
    morse4(n,n3);
    morse5(n,n3);
    morse6(n,n3);
    morse7(n,n3);
    morse8(n,n3);
    morse9(n,n3);
    morse0(n,n3);
    morsecomm(n,n3);
    morseperi(n,n3);
    morseque(n,n3);
    morsear(n,n3);
}

if(RC3==0){
    morsec(n,n3);
    morseq(n,n3);
    wait(n4);
    morsec(n,n3);
```

```
    morseq(n,n3);
    wait(n4);
    morsed(n,n3);
    morsee(n,n3);
    wait(n4);
    jh1zvp(n,n3);
    morsesla(n,n3);
    morse3(n,n3);
    wait(n4);
    jh1zvp(n,n3);
    morsesla(n,n3);
    morse3(n,n3);
    wait(n4);
    morsek(n,n3);
}

if(RC4==0){
    morseu(n,n3);
    morser(n,n3);
    wait(n4);
    morse5(n,n3);
    morsen(n,n3);
    morsen(n,n3);
    wait(n4);
    morse7(n,n3);
    morse3(n,n3);
    wait(n4);
    morset(n,n3);
    morseu(n,n3);
}

 if(RC5==0){
    morseq(n,n3);
```

```
    morset(n,n3);
    morseh(n,n3);
    wait(n4);
    morseh(n,n3);
    morser(n,n3);
    wait(n4);
    morsei(n,n3);
    morses(n,n3);
    wait(n4);
    kamakura(n,n3);
    wait(n4);
    kamakura(n,n3);
}

if(RC6==0){
    morseo(n,n3);
    morsep(n,n3);
    wait(n4);
    morseh(n,n3);
    morser(n,n3);
    wait(n4);
    morsei(n,n3);
    morses(n,n3);
    wait(n4);
    morsem(n,n3);
    morsea(n,n3);
    morset(n,n3);
    morses(n,n3);
    morseu(n,n3);
    morsed(n,n3);
    morsea(n,n3);
    wait(n4);
    morsem(n,n3);
```

```
        morsea(n,n3);
        morset(n,n3);
        morses(n,n3);
        morseu(n,n3);
        morsed(n,n3);
        morsea(n,n3);
    }

     if(RC7==0){
        morseh(n,n3);
        morsew(n,n3);
        morseque(n,n3);
        wait(n4);
        morsed(n,n3);
        morsee(n,n3);
        wait(n4);
        jh1zvp(n,n3);
        morsesla(n,n3);
        morse3(n,n3);
        wait(n4);
        morsekn(n,n3);
    }

    if((RB4==0)&&(RA2==0))
            dot(n,n3);
    if((RB4==0)&&(RA2==1))
            dash(n,n3);
    if((RB5==0)&&(RA2==0))
            dash(n,n3);
    if((RB5==0)&&(RA2==1))
            dot(n,n3);
}
```

```
void pract(n,n3){   // 練習モード
    int i,x1;
    int a=11;
    int b= 3;
    int M=1000;     // 1000 個の（周期 1000 個の）乱数発生
    int x=8;

    morseh(n,n3);
    morser(n,n3);
    wait(n3);
    morseh(n,n3);
    morser(n,n3);
    wait(n3);
    morsebt(n,n3);
    wait(n4);

    for(i=0; i<M; i++){
        x1 = (a*x+b)%M+1;   // 線形合同法による疑似乱数の発生
        if(x1>0 && x1<=20)      morse1(n,n3);
        if(x1>20 && x1<=40)     morse2(n,n3);
        if(x1>40 && x1<=60)     morse3(n,n3);
        if(x1>60 && x1<=80)     morse4(n,n3);
        if(x1>80 && x1<=100)    morse5(n,n3);
        if(x1>100 && x1<=120)   morse6(n,n3);
        if(x1>120 && x1<=140)   morse7(n,n3);
        if(x1>140 && x1<=160)   morse8(n,n3);
        if(x1>160 && x1<=180)   morse9(n,n3);
        if(x1>180 && x1<=200)   morse0(n,n3);
        if(x1>200 && x1<=220)   morsea(n,n3);
        if(x1>220 && x1<=240)   morseb(n,n3);
        if(x1>240 && x1<=260)   morsec(n,n3);
        if(x1>260 && x1<=280)   morsed(n,n3);
        if(x1>280 && x1<=300)   morsee(n,n3);
```

```
if(x1>300 && x1<=320)   morsef(n,n3);
if(x1>320 && x1<=340)   morseg(n,n3);
if(x1>340 && x1<=360)   morseh(n,n3);
if(x1>360 && x1<=380)   morsei(n,n3);
if(x1>380 && x1<=400)   morsej(n,n3);
if(x1>400 && x1<=420)   morsek(n,n3);
if(x1>420 && x1<=440)   morsel(n,n3);
if(x1>440 && x1<=460)   morsem(n,n3);
if(x1>460 && x1<=480)   morsen(n,n3);
if(x1>480 && x1<=500)   morseo(n,n3);
if(x1>500 && x1<=520)   morsep(n,n3);
if(x1>520 && x1<=540)   morseq(n,n3);
if(x1>540 && x1<=560)   morser(n,n3);
if(x1>560 && x1<=580)   morses(n,n3);
if(x1>580 && x1<=600)   morset(n,n3);
if(x1>600 && x1<=620)   morseu(n,n3);
if(x1>620 && x1<=640)   morsev(n,n3);
if(x1>640 && x1<=660)   morsew(n,n3);
if(x1>660 && x1<=680)   morsex(n,n3);
if(x1>680 && x1<=700)   morsey(n,n3);
if(x1>700 && x1<=720)   morsez(n,n3);

if(x1>720 && x1<=740)   morsee(n,n3);
if(x1>740 && x1<=760)   morset(n,n3);
if(x1>760 && x1<=780)   morsea(n,n3);
if(x1>780 && x1<=800)   morseo(n,n3);
if(x1>800 && x1<=820)   morsei(n,n3);
if(x1>820 && x1<=840)   morseperi(n,n3);
if(x1>840 && x1<=860)   morsecomm(n,n3);
if(x1>860 && x1<=880)   morseque(n,n3);
if(x1>880 && x1<=900)   morsesla(n,n3);
if(x1>900 && x1<=1000)  wait(n4);
x = x1;
```

```
    }
    wait(n4);
    morsear(n,n3);
    wait(n3);
    wait(n3);
}


void output0(){
     LATA=0b00000000;
     LATB=0b00000000;
}


void dash(n,n3){
        tone(n3);
        output0();
        wait(n);
}

void dot(n,n3){
        tone(n);
        output0();
        wait(n);
}

void morsea(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morseb(n,n3){
```

```
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsec(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsed(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
```

```
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsee(n,n3){
        tone(n);
        output0();
        wait(n3);
}

void morsef(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morseg(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
```

```
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morseh(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsei(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsej(n,n3){
        tone(n);
        output0();
        wait(n);
```

```
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsek(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsel(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
```

```
        output0();
        wait(n3);
}

void morsem(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsen(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morseo(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0;
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsep(n,n3){
```

```
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morseq(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morser(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
```

```
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morses(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morset(n,n3){
        tone(n3);
        output0();
        wait(n3);
}

void morseu(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
```

```
        wait(n3);
}

void morsev(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
       tone(n3);
        output0();
        wait(n3);
}

void morsew(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsex(n,n3){
        tone(n3);
        output0();
        wait(n);
```

```
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsey(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsez(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
```

```
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morse1(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morse2(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
```

```
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morse3(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morse4(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
```

```
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morse5(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morse6(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
```

```
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();

        wait(n3);
}

void morse7(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morse8(n,n3){
        tone(n3);
```

```
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morse9(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}
```

```
void morse0(n,n3){
	tone(n3);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n3);
}

void morseperi(n,n3){
	tone(n);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n);
	tone(n);
	output0();
	wait(n);
	tone(n3);
	output0();
	wait(n);
	tone(n);
	output0();
	wait(n);
```

```
        tone(n3);
        output0();
        wait(n3);
}

void morsecomm(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morseque(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
```

```
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsear(n,n3){
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsebt(n,n3){
        tone(n3);
        output0();
```

```
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n3);
}

void morsekn(n,n3){
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void morsesla(n,n3){
```

```
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n);
        tone(n3);
        output0();
        wait(n);
        tone(n);
        output0();
        wait(n3);
}

void wait(n){
    int i;
    for(i=0; i<n; i++){
      __delay_us(1000);
    }
}

void tone(n){
    int i;
    for(i=0; i<n; i++){
    LATA = 0b00000001;
    LATB = 0b11000000;
        __delay_us(500);
    LATA = 0b00000000;
    LATB = 0b11000000;
        __delay_us(500);
    }
```

```
}

void jh1zvp(n,n3){
    morsej(n,n3);
    morseh(n,n3);
    morse1(n,n3);
    morsez(n,n3);
    morsev(n,n3);
    morsep(n,n3);
}

void kamakura(n,n3){
    morsek(n,n3);
    morsea(n,n3);
    morsem(n,n3);
    morsea(n,n3);
    morsek(n,n3);
    morseu(n,n3);
    morser(n,n3);
    morsea(n,n3);
}
```