```c
#include "Functions.h"

int main(void)
{
    // define variables
    int N_step, N_delay, flag, N, sum, k;
    unsigned long V, V1, V2;

    // ADC&DAC setting
    ADCpoweron(20000);  // power on ADC
    REFCON = 0x01;  // internal 2.5V reference
    DAC0CON = 0x12; // AGND-ARef range 0x12 2.5V
    ADCCP = 0x00;   // conversion on ADC0
    ADCCON = 0x6E4; // continuous conversion

    // IO setting
    GP1CON = 0x00000000;    // IO initialization
    GP1DAT = 0xFF000000;    // set P1.n as digital output
    GP0CON = 0x00000000;    // IO initialization
    GP0DAT = 0x00000000;    // set P0.n as digital input

    // locking parameters initialization
    N_step = 5; //  step size
    N_delay = 500; // wait for certain time
    flag = 1; // indicator for searching direction
    N = 100; // accumulation number for each locking step
    V1 = DATtoADC(0); // initialize the voltage of first step

    // main loop for the locking
    while(1){
        Sweep();
        // switch between sweep mode and locking mode;
        // note that the sweep mode is just for the convenience of experiment;
        // for locking the phase, it is not needed.
        Write_Digital(0,1); // Output a high level digital output on pin P1.0 for indicating in the locking mode.

        V = V + flag * DATtoADC(N_step); // calculate the voltage for next step
        V = Bound(V); // limit the range of V
        DAC0DAT = V; // output voltage on DAC0
        Delay(N_delay); //wait for a certain time for the response of PZT

        sum = 0; // initialization
        for(k = 1; k <= N; k++){
            while(!ADCSTA){}    // wait for end of ADC conversion
            sum += ADCtoDAT(ADCDAT); // read voltage from ADC0
        }
        V2 = DATtoADC(sum/N);   // calculate average value for the voltage of second step

        if(V2 < V1) flag = -1 * flag; // change maximum searching direction if V2 < V1

        V1 = V2; // update the voltage of first step
    }
}
```

```c
#include<aduc7020.h>

// declaration of functions
void ADCpoweron(int);
void Delay(int);
int Read_Digital(int n);
void Write_Digital(int n, int state);
int ADCtoDAT(unsigned long ADC);
unsigned long DATtoADC(int DAT);
unsigned long Bound(unsigned long N);
void Sweep(void);
void Max_Min(unsigned long *y_max, unsigned long *y_min);

// flag for sweeping direction
struct bs
{
    unsigned c0:1;
    unsigned c1:1;
    unsigned c2:1;
    unsigned c3:1;
    unsigned c4:1;
    unsigned c5:1;
    unsigned c6:1;
    unsigned c7:1;
};
struct bs flag = {0,0,0,0,0,0,0,0};

// read digital value from the pin P0.n
int Read_Digital(int n)
{
    return ((GP0DAT&0x000000FF) >> n) & 0x1;
}

// write digital value to the pin P1.n
void Write_Digital(int n, int state)
{
    if(state == 1)
        GP1DAT = (0x00000001<<(n+16))|GP1DAT;
    else
        GP1DAT = ~((0x00000001<<(n+16))|(~GP1DAT));
}

// wait for ADC to be fully powered on
void ADCpoweron(int time)
{
    ADCCON = 0x620; // power-on the ADC
    while (time >= 0)   time--;
}

// convert ADC/DAC format to integer format
int ADCtoDAT(unsigned long ADC)
{
    return (ADC&0xFFF0000)>>16;
}

// convert integer format to ADC/DAC format
unsigned long DATtoADC(int DAT)
{
    unsigned long ADC;
    ADC = DAT;
    return ADC<<16;
}

//  full scanning for finding the max and min value
void Max_Min(unsigned long *y_max, unsigned long *y_min)
{
```

```
    int i;
    unsigned long V, Ymax, Ymin;

    Ymax = 0;
    Ymin = 0xFFF0000;
    V = 0;

    if(flag.c0 == 0){   // make a ramp sweeping
        for(i = 10; i <= 4080; i += 10){
            DAC0DAT = DATtoADC(i);
            Delay(100);  // wait for PZT
            V = ADCDAT;
            if(V > Ymax) Ymax = V;
            if(V < Ymin) Ymin=V;
        }
    }
    else {
        for(i = 4080; i >= 10; i -= 10){
            DAC0DAT = DATtoADC(i);
            Delay(100);  // wait for PZT
            V = ADCDAT;
            if(V > Ymax) Ymax = V;
            if(V < Ymin) Ymin = V;
        }
    }
flag.c0++;
*y_max = Ymax;
*y_min = Ymin;
}

// sweeping mode
void Sweep(void)
{
    unsigned long y_max, y_min;
    if(Read_Digital(3) == 0){ // read digital value from pin P0.3
        while(Read_Digital(3) == 0){
            Write_Digital(0,0); //write a low level digital value on pin P1.0 for indicating in the sweeping mode.
            Max_Min(&y_max, &y_min);
        }
    }
}

// wait for a certain time
void Delay(int cnt)
{
    while(cnt>0) cnt--;
}

// Bound the output value of DAC
// when V reaches the limit of DAC range, it will be shifted to the center
unsigned long Bound(unsigned long V)
{
  if (V > 0xFCE0000 || V < 0x320000)
        return 0x8000000;
    else
        return V;
}
```