Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 5. Mikrovlnná trouba | | | |
| Dobeš Daniel | |  | 1/3 | Známka: |
| 19. 12. 2018 | | Datum odevzdání: | 30. 1. 2019 | Odevzdáno: |

Zadání:

Zpracujte ovládací program v programovacím jazyce C ovládající model mikrovlnné trouby tak, aby obsahoval nejméně tyto funkce:

1. Simulujte provoz velmi jednoduché mikrovlnné trouby
2. Na vestavěném displeji modelu zobrazuje jak dobu ohřevu, tak i teplotu „pokrmu“
3. Na vestavěné klávesnici modelu umožní nastavit jak dobu ohřevu, tak i požadovanou teplotu.
4. Na vestavěné klávesnici modelu umožní nastavit pracovní otáčky talíře
5. Při „ohřevu“ průběžně zobrazuje metodou „countdown“ zbývající dobu ohřevu
6. Sleduje a zobrazuje provozní a chybové stavy přípravku na monitoru PC

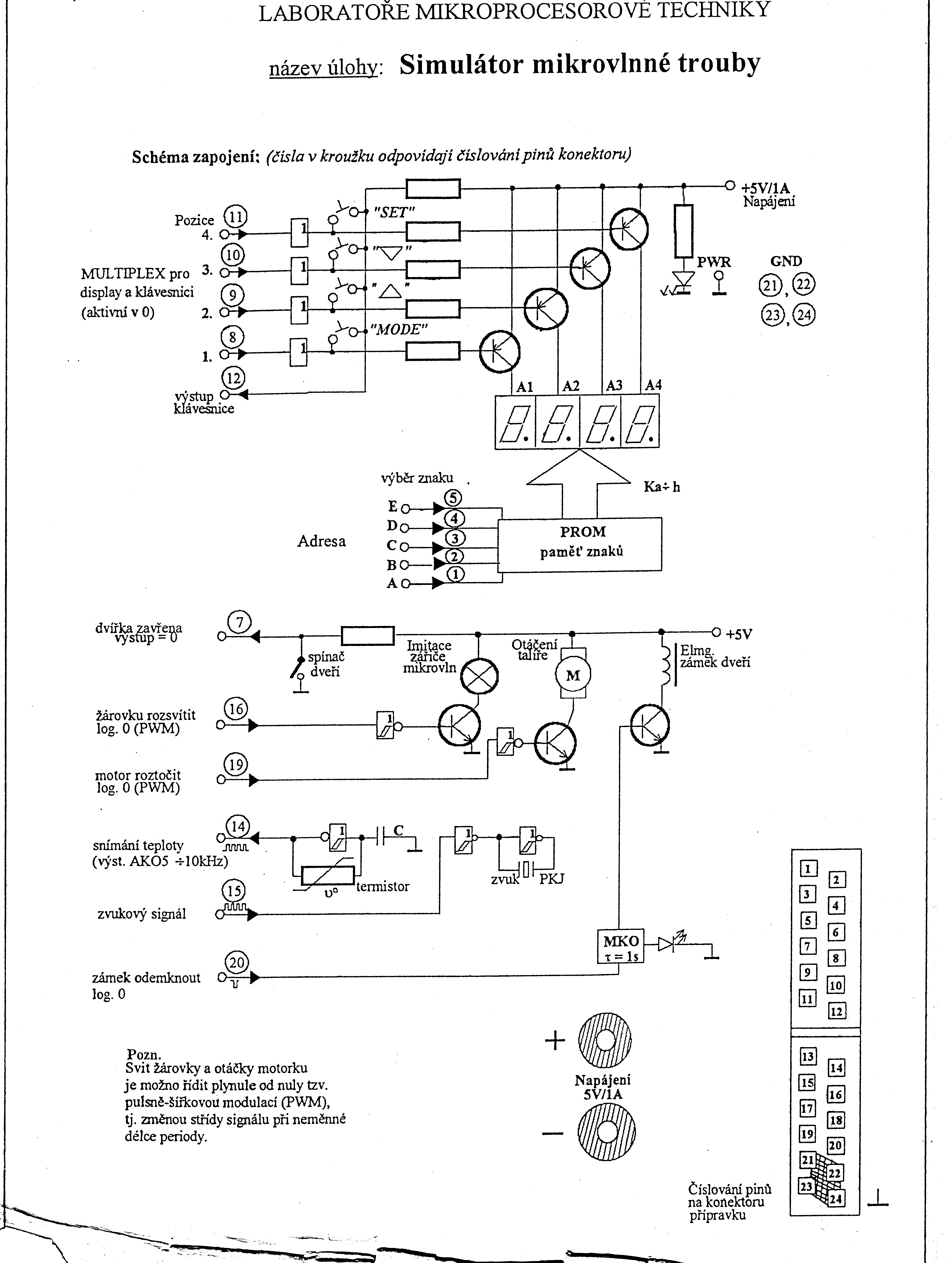
Postup:

1. Stisknout „MODE“
2. Nastavení doby ohřevu pomocí „UP“ a „DOWN“
3. Potvrzení doby „SET“
4. Nastavení teploty ohřevu pomocí „UP“ a „DOWN“
5. Potvrzení teploty „SET“
6. Nastavení rychlosti otáčení volbou 1-4 pomocí „UP“ a „DOWN“
7. Potvrzení otáček „SET“
8. Zapnutí ohřevu „SET“
9. Po skončení ohřevu – 1.

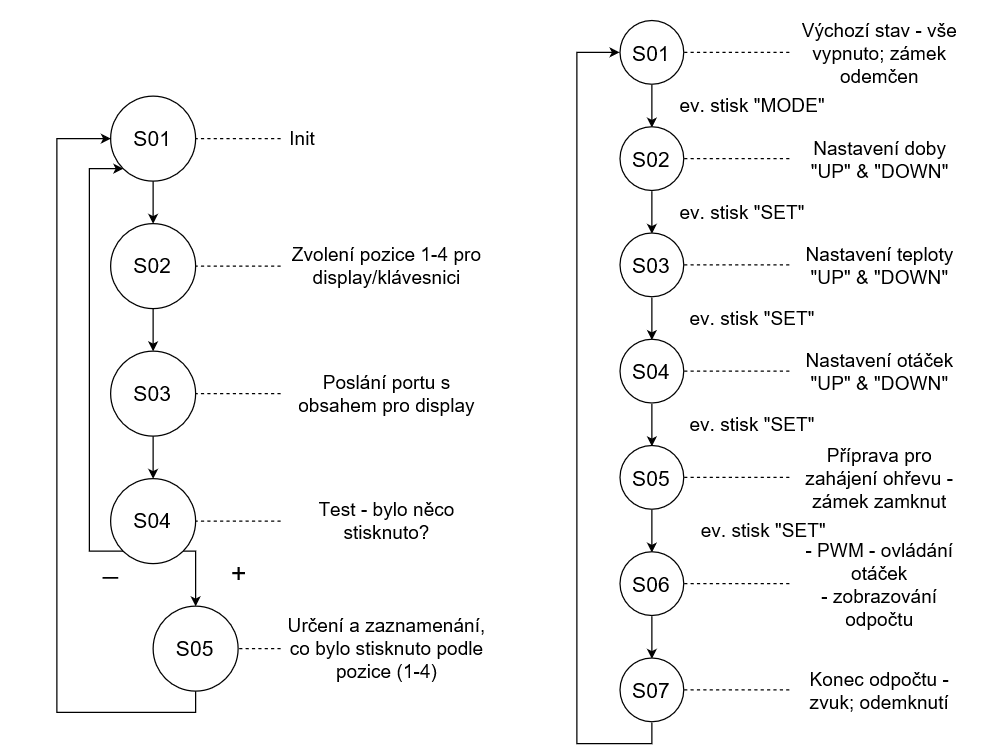
Propojení PC a Mikrovlnné trouby:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0x300 (IN) P1 | | 0x301 (IN) P2 | | 0x301 (OUT) P4 | |
| Bit PC | Pin MT | Bit PC | Pin MT | Bit PC | Pin MT |
| 0 | A | 0 | 8 (digit1) | 0 | - |
| 1 | B | 1 | 9 (digit2) | 1 | - |
| 2 | C | 2 | 10 (digit3) | 2 | - |
| 3 | D | 3 | 11 (digit4) | 3 | - |
| 4 | E | 4 | 16 (bulb) | 4 | - |
| 5 | - | 5 | 19 (motor) | 5 | 14 (temperature) |
| 6 | - | 6 | 15 (tone) | 6 | 7 (door) |
| 7 | - | 7 | 20 (lock) | 7 | 12 (key) |

Schéma zapojení:



Stavový diagram:



Výpis programu:

Viz příloha

Závěr:

Tento program nebyl testován přímo na přípravku, ale nad úlohou jsem se dlouho zamýšlel a věřím, že to nebylo marné. Program neobsahuje funkci čtení teploty, tudíž ji nemůže ani vypisovat na displej. Zároveň jsem nezprovoznil ovládání žárovky z důvodu nefunkčního čtení teploty. Odpočítávání a veškerá další signalizace je splněna.

Přílohy:

* Výpis programu (11 stran)

**Výpis programu:**

**#include <stdio.h> //load the basic library**

**#include <dos.h> //load the library for communication with DOS**

**#include <time.h> //load the library to be able to used timer (delay)**

**#define P1 0x300 //adress of Port1**

**#define P2 0x301 //adress of Port2**

**#define P4 0x301 //adress of Port4**

**#define A 0x0A //Name of value to show**

**#define B 0x0B**

**#define C 0x0C**

**#define D 0x0D**

**#define E 0x0E**

**#define F 0x0F**

**#define G 0x10**

**#define H 0x11**

**#define I 0x01**

**#define J 0x12**

**#define L 0x13**

**#define M 0x14**

**#define N 0x15**

**#define O 0x00**

**#define P 0x16**

**#define R 0x17**

**#define S 0x05**

**#define T 0x18**

**#define U 0x19**

**#define V 0x19**

**#define Y 0x1A**

**#define MIN 0x00**

**#define MAX 0xFF**

**#define nothing 0x1D**

**#define degree 0x1B**

**#define digit1lock 0x7F //adress of digit1 & Door is locked**

**#define digit2lock 0xBF**

**#define digit3lock 0xDF**

**#define digit4lock 0xEF**

**#define digit1unlock 0x7E //adress of digit1 & Door is unlocked**

**#define digit2unlock 0xBE**

**#define digit3unlock 0xDE**

**#define digit4unlock 0xEE**

**#define ButtonMask 0x80 //0b1000\_0000**

**#define DoorMask 0x40 //0b0100\_0000**

**#define MotorON 0xFB //0b1111\_1011**

**#define OFF 0xFF**

**#define Tone 0xFD //0b1111\_1101**

**#define Mode 0 //using to determine pressed button (Part2)**

**#define Up 1**

**#define Down 2**

**#define Set 3**

**void main()**

**{**

**//Init**

**outportb(P1, MAX); //Full reset all status**

**outportb(P2, MAX);**

**//create new field with value to display**

**unsigned char value[] = {value1, value2, value3, value4};**

**//create new field with digit adress**

**unsigned char digit[] = {digit1, digit2, digit3, digit4};**

**//when something is "1" -> something is pressed**

**int button[] = {0, 0, 0, 0}; //[Mode, Up, Down, Set]**

**int index1 = 0; //field position selection in Part1**

**int temp1; //object to save read port**

**int ActiButton = 0; //if "1" -> something is pressed, if "0" nothing is pressed**

**int Step = 0; //used in Part 2 - Define each step**

**unsigned char digit1;**

**unsigned char digit2;**

**unsigned char digit3;**

**unsigned char digit4;**

**unsigned char value1 = nothing;**

**unsigned char value2 = nothing;**

**unsigned char value3 = nothing;**

**unsigned char value4 = nothing;**

**int Time1, Time2, Time3, Time4;**

**int temperature = 0;**

**int rotation = 1;**

**int PWMrotON, PWMrotOFF;**

**int temp2;**

**while (true) //Endless cycle**

**{**

**{ //Part 1 - Displaying one of digit an control one button**

**if (index1 == 4) index1 = 0; //Eeset if index1 != avilable value in position in field**

**outportb(P2, digit[index1]); //Selecting digit**

**outportb(P1, value[index1]); //Value display**

**temp1 = inportb(P4); //Load value from port - button**

**if (temp1 && ButtonMask == 0) //Control pressed button**

**{**

**button [] = {0, 0, 0, 0}; //Clear, dont do "something in 2nd part" again**

**if (ActiButton == 0) //something is pressed -> nothing happen**

**{**

**button[index1] = 1; //[Mode, Up, Down, Set]**

**ActiButton = 1; //Set that something is pressed**

**}**

**}**

**else**

**ActiButton = 0; //Set that nothing is pressed**

**index1 ++; //inc to display next digit and control next button**

**} //end of Part 1**

**{ //Part 2 - Each Step to manage microwave**

**switch (Step)**

**{**

**case '0': //Default state - waiting for press "Mode"**

**{**

**value1 = nothing; //nothing display**

**value2 = nothing;**

**value3 = nothing;**

**value4 = nothing;**

**digit1 = digit1unlock; //unlock door**

**digit2 = digit2unlock;**

**digit3 = digit3unlock;**

**digit4 = digit4unlock;**

**if (button[Mode] == 1) Step++; //next if is "Mode" pressed**

**}**

**break;**

**case '1': //Set heating time**

**{**

**value1 = T;**

**if (button[Up] == 1) //If "Up" is pressed**

**{**

**value4++; //set time + 1 sec**

**if (value4 > 9) //max unit seconds**

**{**

**value3++; //tens seconds + 1**

**value4 = 0; //reset unit seconds**

**if (value3 > 6) //max tens seconds**

**{**

**value2++; //minutes + 1**

**value3 = 0; //reset tens seconds**

**if (value2 > 9) //max minutes**

**{**

**printf("You have reached the maximum heating time!");**

**value2 = 9; //set last value again**

**value3 = 5;**

**value4 = 9;**

**}**

**}**

**}**

**}**

**Step++;**

**}**

**break;**

**case '2': //Set heating time**

**{**

**if (button[Down] == 1) //If "Down" is pressed**

**{**

**if (value4 == 0)**

**{**

**if (value3 == 0)**

**{**

**if (value2 == 0)**

**{**

**printf("You have reached the minimum heating time!");**

**}**

**else**

**{**

**value2--; //decrement minutes**

**value3 = 5; //reset tens seconds**

**value4 = 9; //reset units seconds**

**}**

**}**

**else**

**{**

**value3--; //decrement tens seconds**

**value4 = 9; //reset units seconds**

**}**

**}**

**else**

**{**

**value4--; //decrement units seconds**

**}**

**}**

**Step++;**

**}**

**break;**

**case '3': //Confirmation of choice**

**{**

**if (button[Set] == 1) //If "Set" is pressed**

**{**

**Time1 = value1; //Store the selected time**

**Time2 = value2;**

**Time3 = value3;**

**Time4 = value4;**

**value1 = 0; //Clear value**

**value2 = 0;**

**value3 = 0;**

**value4 = 0;**

**Step++;**

**}**

**else**

**{**

**Step = 1;**

**}**

**}**

**break;**

**case '4': //Set heating temperature**

**{**

**value4 = C;**

**value3 = degree;**

**if (button[Up] == 1) //If "Up" is pressed**

**{**

**value2++;**

**temperature++;**

**if (value2 > 9) //quite same as a time setting**

**{**

**value1++;**

**value2 = 0;**

**if (value1 > 9)**

**{**

**printf("You have reached the maximum heating temperature!");**

**value1 = 9;**

**value2 = 9;**

**temperature--;**

**}**

**}**

**}**

**Step++;**

**}**

**break;**

**case '5': //Set heating temperature**

**{**

**if (button[Down] == 1) //If "Down" is pressed**

**{**

**if (value2 == 0) //quite same as a time setting**

**{**

**if (value1 == 0)**

**{**

**printf("You have reached the minimum heating temperature!");**

**}**

**else**

**{**

**value1--;**

**value2 = 9;**

**temperature--;**

**}**

**}**

**else**

**{**

**value2--;**

**temperature--;**

**}**

**}**

**Step++;**

**}**

**break;**

**case '6': //Confirmation of choice**

**{**

**if (button[Set] == 1) //If "Set" is pressed**

**{**

**value1 = 0; //Clear value**

**value2 = 0;**

**value3 = 0;**

**value4 = 0;**

**Step++;**

**}**

**else**

**{**

**Step = 4;**

**}**

**}**

**break;**

**case '7': //Set plate rotation**

**{ //Choice of 4 speeds**

**if (button[Up] == 1) //If "Up" is pressed**

**{**

**value4 = T;**

**value3 = 0;**

**value2 = nothing;**

**value1++;**

**rotation++;**

**if (value1 > 4)**

**{**

**printf("You have reached the maximum rotation speed");**

**value1 = 4;**

**rotation--;**

**}**

**}**

**Step++;**

**}**

**break;**

**case '8': //Set plate rotation**

**{ //Choice of 4 speeds**

**if (button[Down] == 1) //If "Down" is pressed**

**{**

**if (value1 == 0)**

**{**

**printf("You have reached the minimum rotation speed");**

**value1++;**

**rotation++;**

**}**

**else**

**{**

**value1--;**

**rotation--;**

**}**

**}**

**Step++;**

**}**

**break;**

**case '9': //Confirmation of choice**

**{**

**if (button[Set] == 1) //If "Set" is pressed**

**{**

**value1 = nothing; //Write "Set"**

**value2 = S;**

**value3 = E;**

**value4 = T;**

**if (rotation == 1) //Assign PWM mode of rotation**

**{ PWMrotON = 10; PWMrotOFF = 40; }**

**if (rotation == 2)**

**{ PWMrotON = 20; PWMrotOFF = 30; }**

**if (rotation == 3)**

**{ PWMrotON = 30; PWMrotOFF = 20; }**

**if (rotation == 4)**

**{ PWMrotON = 40; PWMrotOFF = 10; }**

**Step++;**

**}**

**else**

**{**

**Step = 7;**

**}**

**}**

**break;**

**case '10': //Confirmation to start heating**

**{**

**if (temp1 && DoorMask == 0) //If door is closed**

**{**

**if (button[Set] == 1) //If "Set" is pressed**

**{**

**digit1 = digit1lock; //lock door**

**digit2 = digit2lock;**

**digit3 = digit3lock;**

**digit4 = digit4lock;**

**value1 = Time1;**

**value2 = Time2;**

**value3 = Time3;**

**value4 = Time4;**

**Step++;**

**}**

**}**

**}**

**break;**

**case '11': //Rotating motor - turn on**

**{**

**outportb(P2, MotorON);**

**delay(PWMrotON); //delay of active PWM**

**Step++;**

**}**

**break;**

**case '12': //Rotating motor - turn off**

**{**

**outportb(P2, MotorOFF);**

**delay(PWMrotOFF); //delay of inactive PWM**

**Step++;**

**}**

**break;**

**case '13': //Count Down**

**{**

**temp2++; //used as a deduction 1s**

**if (temp2 == 5) //5x = cca 1 sec**

**{**

**if (value4 == 0)**

**{**

**if (value3 == 0)**

**{**

**if (value2 == 0)**

**{**

**Step++;**

**}**

**else**

**{**

**value2--;**

**value3 = 5;**

**value4 = 9;**

**Step = 11;**

**}**

**}**

**else**

**{**

**value3--;**

**value4 = 9;**

**Step = 11;**

**}**

**}**

**else**

**{**

**value4--;**

**Step = 11;**

**}**

**temp2 = 0; //Reset - next compare 1 sec**

**}**

**}**

**break;**

**case '14': //End**

**{**

**printf("The heating is done!");**

**outportb(P2, Tone);**

**delay(200);**

**outportb(P2, OFF);**

**Step = 0;**

**}**

**break;**

**} //end of Part 2**

**} //end of While**

**} //end of void main()**