Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 7. Trezor | | | |
| Macháček Daniel | |  | 1/12 | Známka: |
| 1.3.2016 | | Datum odevzdání: | 22.3.2017 | Odevzdáno: |

Zadání:

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

1. Simulujte povoz velmi jednoduché mikrovlnné trouby
2. Na vestavěném displeji modelu zobrazujte 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 zobrazujte metodou „countdown“ zbývající dobu ohřevu
6. Sledujte a zobrazujte provozní a chybové stavy přípravku na monitoru PC

Popis zapojení:

****

Funkce:

* Dvířka mikrovlnky se dají zavřít kdykoli během nastavování parametrů nebo už ve výchozím stavu, zámek zaklapne až po spuštění ohřevu
* Ve výchozím stavu funguje pouze tlačítko MODE
* Během nastavování parametrů fungují pouze tlačítka NAHORU, DOLU a SET (NAHORU a DOLU nastavují hodnoty parametrů a tlačítkem SET se potvrzují)

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, ale odpočítávání a veškerá další signalizace na displeji je splněna. Dále jsem nezprovoznil ovládání žárovky z důvodu, jak jsem již zmínil, nefunkčního čtení teploty. Mým cílem bylo číst teplotu a porovnávat ji s aktuální a tím dosahovat regulace s pomocí PWM.

Přílohy:

* Výpis programu – 9 stran
* Stavové diagramy – 1 strana

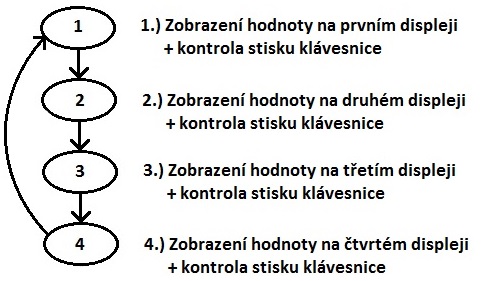
**Stavové diagramy:**

Proces programu:



* Zobrazování na displeji probíhá „paralelně vůči vykonávaným operacím, z toho důvodu jsem nakreslil dva stavové diagramy

Zobrazování na displeji:



**Výpis programu:**

#include <conio.h> //načtení základní knihovmy

#include <stdio.h> //načtení základní knihovmy

#include <dos.h> //načtení knihovmy pro komunikaci s DOS

#include <time.h> //načtení knihovny pro práci s časovačem

int a = 1; //proměnná pro zobrazení pozdravu

int c = 0; //proměnná pro sekundový odpočet

int cas1, cas2, cas3, cas4; //proměnné pro uložení nastaveného času

int teplota = 0; //proměnná pro uložení nastavené teploty

int otacky = 0; //proměnná pro uložení nastavených otáček

int statusd = 0; //proměnná ovládající první switch

int statusp = 0; //proměnná ovládající druhý switch

int i301; //proměnná pro uložení přečteného vstupu

int MODE = 0; //proměnná přiřřazená tlačítku mikrovlnky

int SET = 0; //proměnná přiřřazená tlačítku mikrovlnky

int UP = 0; //proměnná přiřřazená tlačítku mikrovlnky

int DOWN = 0; //proměnná přiřřazená tlačítku mikrovlnky

int PWMM1, PWMM2, PWMZ1, PWMZ2; //proměnné pro hodnoty PWM

unsigned char hodnota1 = 0x1D; //proměnné pro hodnoty zobrazované na displeji (výchozí podtržítko)

unsigned char hodnota2 = 0x1D;

unsigned char hodnota3 = 0x1D;

unsigned char hodnota4 = 0x1D;

unsigned char dis1 = 0x7F; //proměnné pro volbu zobrazovaného displeje

unsigned char dis2 = 0xBF;

unsigned char dis3 = 0xDF;

unsigned char dis4 = 0xEF;

unsigned char maska12 = 0x80; //maska pro čtení výstupu klávesnice

unsigned char maska7 = 0x40; //maska pro čtení výstupu dvířek

void main()

{

outportb(0x300, 0xFF); //vypnutí všech stavů mikrovlnky

outportb(0x301, 0xFF);

while (1) //nekonečná smyčka

{

i301 = inportb(0x301); //načtení hodnot z portu

switch (statusd) //zvolení operace dle hodnoty statusd

{

case '0': //rozsvícení prvního displeje a zobrazení hodnoty

{

outportb(0x301, dis1); //rosvícení prvního displeje

outportb(0x300, hodnota1); //zobrazení chtěné hodnoty

delay(18); //zpožděné mezi jednotlivými displeji

if (i301 & maska12 == 0) //stisknuto tlačítko MODE

{

MODE = 1;

}

statusd = 1; //posunutí k dalšímu procesu

}

break; //ukončení procesu

case '1': //rozsvícení druhého displeje a zobrazení hodnoty

{

outportb(0x301, dis2);

outportb(0x300, hodnota2);

delay(18);

if (i301 & maska12 == 0) //stisknuto tlačítko NAHORU

{

UP = 1;

}

statusd = 2;

}

break;

case '2': //rozsvícení třetího displeje a zobrazení hodnoty

{

outportb(0x301, dis3);

outportb(0x300, hodnota3);

delay(18);

if (i301 & maska12 == 0) //stisknuto tlačítko DOLU

{

DOWN = 1;

}

statusd = 3;

}

break;

case '3': //rozsvícení čtvrtého displeje a zobrazení hodnoty

{

outportb(0x301, dis4);

outportb(0x300, hodnota4);

delay(18);

if (i301 & maska12 == 0) //stisknuto tlačítko SET

{

SET = 1;

}

statusd = 0; //návrat k prvnímu procesu

}

break;

}

switch (statusp)

{

case '0': //pozdrav na displeji po 3s

{

hodnota1 = 0x0A; //A

hodnota2 = 0x11; //H

hodnota3 = 0x00; //O

hodnota4 = 0x12; //J

a = a + 1; //navyšování o jedna kvůli cyklu

if (a == 50) //zajištění zobrazení po 3s cyklem

{

statusp = 1; //posunutí k dalšímu procesu

}

else //při nekodončení cyklu...

{

statusp = 0; //opakuje tento proces

}

}

break;

case '1': //výchozí stav mikrovlnky s otevřením dveří

{

hodnota1 = 0x00; //zobrazení nul na displejích

hodnota2 = 0x00;

hodnota3 = 0x00;

hodnota4 = 0x00;

dis1 = 0x7E; //změna hodnoty, kvůli otevření dvířek

dis2 = 0xBE;

dis3 = 0xDE;

dis4 = 0xEE;

if (MODE == 1) //tlačítko MODE zahájí nastavování parametrů ohřevu

{

statusp = 2;

}

else

{

statusp = 1;

}

}

break;

case '2': //nastavení času ohřevu

{

if (UP == 1) //v případě stisku tlačítka NAHHORU...

{

hodnota4++; //navýší počet sekund o 1

if (hodnota4 == 0x0A) //v případě maximálního počtu sekund...

{

hodnota3++; //navýší počet desítek sekund o 1

hodnota4 = 0x00; //reset sekund

if (hodnota3 == 0x07) //v případě maximálního počtu desítek sekund

{

hodnota2++; //navýší počet minut o 1

hodnota3 = 0x00; //reset desítek sekund

if (hodnota2 == 0x0A)//v případě maximálního počtu minut

{

printf("Vyšší čas nelze nastavit!");

hodnota2 = 0x09; //navrácení hodnot

hodnota3 = 0x05;

hodnota4 = 0x09;

}

}

}

}

statusp = 3;

}

break;

case '3': //nastavení času ohřevu

{

if (DOWN == 1) //v případě stisku tlačítka DOLU...

{

if (hodnota4 == 0x00)

{

if (hodnota3 == 0x00) //kontrola zda už byla hodnota nastavena

{

if (hodnota2 = 0x00) //pokud ne

{

printf("Nižší čas nelze nastavit!");

}

else //pokud ano

{

hodnota2--; //ubrání minut

hodnota3 = 0x05; //reset desítek sekund

hodnota4 = 0x09; //reset sekund

}

}

else //pokud ano

{

hodnota3--; //ubrání desítek sekund

hodnota4 = 0x09; //reset sekund

}

}

else //pokud ano

{

hodnota4--; //ubrání sekund

}

}

if (SET == 1) //stiskem tlačítka SET uložení nastavení

{ //a pokračování k dalším parametrům

statusp = 4;

cas4 = hodnota4;

cas3 = hodnota3;

cas2 = hodnota2;

cas1 = hodnota1;

}

else

{

statusp = 2;

}

}

break;

case '4': //obnova proměnných

{

hodnota1 = 0x00;

hodnota2 = 0x00;

hodnota3 = 0x00;

hodnota4 = 0x00;

statusp = 5;

}

case '5': //nastavení teploty ohřevu

{

if (UP == 1)

{

hodnota4 = 0x0C;

hodnota3 = 0x1B;

hodnota2++;

teplota++;

if (hodnota2 == 0x0A)

{

hodnota1++;

hodnota2 = 0x00;

if (hodnota1 == 0x0A)

{

printf("Vyšší teplota nelze nastavit!");

hodnota1 = 0x09;

hodnota2 = 0x09;

teplota--;

}

}

}

statusp = 6;

}

break;

case '6':

{

if (DOWN == 1)

{

if (hodnota2 == 0x00)

{

if (hodnota1 == 0x00)

{

printf("Nižší teplota nelze nastavit!");

}

else

{

hodnota1--;

hodnota2 = 0x09;

teplota--;

}

}

else

{

hodnota2--;

teplota--;

}

}

if (SET == 1)

{

statusp = 7;

}

else

{

statusp = 5;

}

}

break;

case '7':

{

hodnota1 = 0x00;

hodnota2 = 0x00;

hodnota3 = 0x00;

hodnota4 = 0x00;

statusp = 8;

}

break;

case '8': //nastavení otáček talíře

{ //nastavováno ve čtyřech rychlostech

if (UP == 1)

{

hodnota4 = 0x18;

hodnota3 = 0x00;

hodnota2 = 0x1D;

hodnota1++;

otacky++;

if (hodnota2 == 0x05)

{

printf("Vyšší otáčky nelze nastavit!");

hodnota1 = 0x04;

otacky--;

}

}

statusp = 9;

}

break;

case '9':

{

if (DOWN == 1)

{

if (hodnota1 == 0x00)

{

printf("Nižší otáčky nelze nastavit!");

}

else

{

hodnota1--;

otacky--;

}

}

if (SET == 1)

{

statusp = 10;

}

else

{

statusp = 8;

}

}

break;

case '10': //obnova proměnných + příprava na odpočet

{

if (i301 & maska7 == 0) //pokud jsou dvířka zavřena

{

dis1 = 0x7F; //změna hodnoty kvůli zamčení dvířek

dis2 = 0xBF;

dis3 = 0xDF;

dis4 = 0xEF;

hodnota1 = cas1; //nahrání uloženého času

hodnota2 = cas2;

hodnota3 = cas3;

hodnota4 = cas4;

if (otacky == 4) { PWMM1 = 10; PWMM2 = 40; } //výběr módu otáček

if (otacky == 3) { PWMM1 = 20; PWMM2 = 30; }

if (otacky == 2) { PWMM1 = 30; PWMM2 = 20; }

if (otacky == 1) { PWMM1 = 40; PWMM2 = 10; }

statusp = 11;

}

else //pokud nejsou dvířka zavřena

{

statusp = 10;

}

}

break;

case '11': //Zapnutí pohonu

{

outportb(0x301, 0xFB); //aktivní část PWM

delay(PWMM1); //zpoždění před vypnutím

statusp = 12;

}

break;

case '12': //Vypnutí pohonu

{

outportb(0x301, 0xFF); //vypnutá část PWM

delay(PWMM2); //zpoždění před zapnutím

statusp = 13;

}

break;

case '13':

{

c++; //navýšení hodnoty pro odpočet 1s

if (c == 5) //po uplynutí 1s...

{

if (hodnota4 == 0x00)

{

if (hodnota3 == 0x00) //kontrola zda není ohřev dokončen

{

if (hodnota2 = 0x00) //pokud ano

{

printf("Ohřev dokončen!");

statusp = 1;

}

else //pokud ne

{

hodnota2--;

hodnota3 = 0x05;

hodnota4 = 0x09;

statusp = 11;

}

}

else //pokud ne

{

hodnota3--;

hodnota4 = 0x09;

statusp = 11;

}

}

else //pokud ne

{

hodnota4--;

statusp = 11;

}

c = 0; //reset proměnné pro 1s odpočet

}

}

break;

}

}

}