Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 5. Sériová komunikace | | | |
| Macháček Daniel | |  | 1/7 | Známka: |
| 21.12.2016 | | Datum odevzdání: | 1.2.2017 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce JSA MCS51 a C# ovládající sériovou komunikaci mezi jedním přípravkem MCS51 a PC tak, aby obsahoval nejméně tyto funkce:

1. Stisknuté tlačítko klávesnice počítače se sériovou linkou přenese do přípravku MCS51, kde se zobrazí na třísegmentovém displeji
2. Zvolte vhodný komunikační protokol
3. Komunikační program v přípravku MCS51 s třísegmentovým displejem měl mít funkci ,,autonegotiation“
4. Sledování chybových stavů

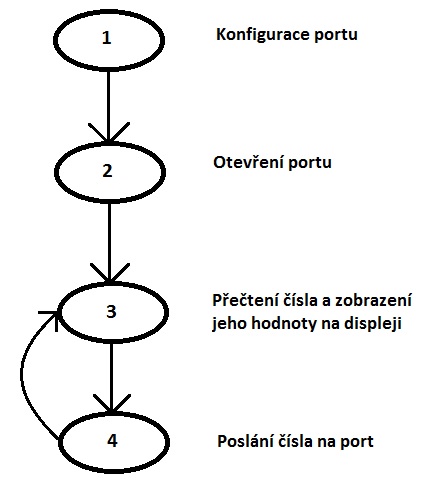
Závěr:

Tento program nebyl ověřován na pracovišti, tudíž nemohu tvrdit, že by byl plně funkční. Program psaný v C# není ošetřen a je proto možné ho snadno splést. Je to z důvodu toho, že jsem věnoval více soustředění JSA a doufám, že to bude mít také nějaké uplatnění.

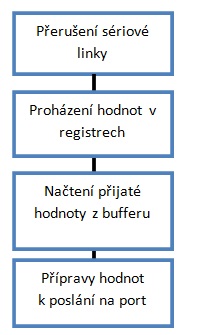
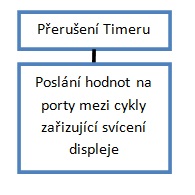
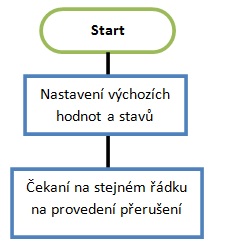
Přílohy:

* Výpisy programů – 4 strany
* Stavový diagram a vývojové diagramy – 1 strana

**Stavový Diagram C#:**



**Vývojové diagramy k JSA:**



**Výpis programu C#:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO.Ports;

namespace Seriovka

{

class Program

{

static SerialPort Port = new SerialPort();

static void Main(string[] args)

{

int status = 0; //Definice proměných

int cislo;

while (true) //Nekonečná smyčka

{

switch(status) //Zvolení činnosti na základě hodnoty status

{

case '0':

{

Port.PortName = "8051"; //Název portu

Port.BaudRate = 4800; //Rychlost přenosu dat

Port.DataBits = 8; //Počet přenášených bitů

status = 1; //Hodnota status se rovná 1

}

break; //Konec činnosti

case '1':

{

Port.Open(); //Otevření portu

status = 2; //Hodnota status se rovná 2

}

break;

case '2':

{

Console.WriteLine("Stiskni číslo od 1 do 9 (včetně): ");

cislo = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Na port 8051 bylo odesláno číslo " + cislo);

status = 3; //Hodnota status se rovná 3

}

break;

case '3':

{

Port.Write(cislo.ToString()); //Poslání čísla na port

status = 2; //Hodnota status se rovná 2

}

break;

}

}

}

}

}

**Výpis programu JSA:**

org 0 ;Začátek programu

jmp setintr ;Skok na počáteční nastavení

org 0Bh ;Adresa přerušení

jmp intr ;Skok na provedení přerušení

org 23h ;Adresa přerušení sériové linky

jmp sintr ;Skok na provedení přerušení

cislad1: DB 10110000b, 10110001b, 10110010b, 10110011b, 10110100b, 10110101b, 10110110b, 10110111b, 10111000b, 10111001b

cislad2: DB 11010000b, 11010001b, 11010010b, 11010011b, 11010100b, 11010101b, 11010110b, 11010111b, 11011000b, 11011001b

cislad3: DB 11100000b, 11100001b, 11100010b, 11100011b, 11100100b, 11100101b, 11100110b, 11100111b, 11101000b, 11101001b

casintr EQU 60000

setintr: ;Počáteční nastavení

mov DPTR, #8000h ;Instrukce pro komunikaci s HardWarem

movc A, @A+DPTR ;Instrukce pro komunikaci s HardWarem

mov TMOD, #00100001b ;Nastavení časovačů

mov SCON, #01010000b ;Nastavení sériové komunikace

setb EA ;Povolení přerušení

setb ET0 ;Povolení přerušení Timeru

setb ES ;Povolení přerušení sériové linky

mov R0, #0FFh ;Defaultní hodnoty

mov R1, #0FFh ;Defaultní hodnoty

mov R2, #0FFh ;Defaultní hodnoty

mov R3, #1d ;Defaultní hodnoty

mov TH0, #high(casintr) ;Nastaví horní část časovače hodotou casintr

mov TL0, #low(casintr) ;Nastaví spodní část časovače hodotou casintr

setb TR0 ;Spustí časovač

mov TH1, #0FAh ;Nastaví horní část časovače hodotou casintr

setb TR1 ;Spustí časovač

jmp $ ;Opakuje skok na tento řádek

intr: ;Přerušení časovače

mov TH0, #high(casintr) ;Nastaví horní část časovače hodotou casintr

mov TL0, #low(casintr) ;Nastaví spodní část časovače hodotou casintr

jedna:

Cjne R3, #1d, dva ;Skočí na návěští dva pokud se R3 nerovná 1

mov P3, R0 ;Poslání hodnoty z R0 na port

inc R3 ;Zvětší hodnotu v R3 o 1

jmp konec ;skočí na návěští konec

dva:

Cjne R3, #2d, tri ;Skočí na návěští tri pokud se R3 nerovná 2

mov P3, R1 ;Poslání hodnoty z R1 na port

inc R3 ;Zvětší hodnotu v R3 o 1

jmp konec ;skočí na návěští konec

tri:

Cjne R3, #3d, konec ;Skočí na návěští konec pokud se R3 nerovná 3

mov P3, R2 ;Poslání hodnoty z R2 na port

mov R3, #1d ;Reset hodnoty

konec:

reti ;Ukončení přerušení

sintr: ;Přerušení sériov linky

mov B, R6 ;Přemístí hodnotu z R6 do R7

mov R7, B

mov B, R5 ;Přemístí hodnotu z R5 do R6

mov R6, B

mov R5, SBUF ;Přesune do R5 hodnotu přijetou bufferem

mov A, R5 ;Přesune hodnotu z R5 do A

mov DPTR, #cislad1 ;Vybere použití tabulky cislad1

movc A, @A+DPTR ;Vybere číslo podle indexu

mov R0, A ;Číslo uloží opět do A

mov A, R6 ;Přesune hodnotu z R6 do A

mov DPTR, #cislad2 ;Vybere použití tabulky cislad2

movc A, @A+DPTR ;Vybere číslo podle indexu

mov R1, A ;Číslo uloží opět do A

mov A, R7 ;Přesune hodnotu z R7 do A

mov DPTR, #cislad3 ;Vybere použití tabulky cislad3

movc A, @A+DPTR ;Vybere číslo podle indexu

mov R2, A ;Číslo uloží opět do A

clr RI ;Vyčistí buffer pro další příjem dat

reti ;Ukončení přerušení

end