Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | Sériová komunikace | | | |
| Dobeš Daniel | |  | 1/ | Známka: |
| 7. 11. 2018 | | Datum odevzdání: | 12. 12. 2018 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce JSA ATMega128 a C# ovládající sériovou komunikaci mezi jedním přípravkem MB-ATmega128 a PC tak, aby obsahoval nejméně tyto funkce:

1. Stisknuté tlačítko maticové klávesnice připojené k jednomu přípravku MB-ATmega128 se přenese sériovou linkou do P, kde se vhodně zobrazí
2. Zvolte vhodný komunikační protokol
3. Komunikační program v přípravku s maticovou klávesnicí by měla mít funkci „autonegotiation“
4. Využití všech vhodných HW možností přípravku MB-ATmega128.
5. Sledování chybových stavů

Stavové diagramy:

Viz příloha

Výpis programu:

Viz příloha.

Závěr:

Tento program nebyl ověřován na pracovišti, tudíž nemohu tvrdit, že by byl plně funkční. Při psaní programu v JSA jsem se snažil o maximální univerzálnost kódu.

Přílohy:

* Příloha 1 - Stavové diagramy (1strana)
* Příloha 2 - Výpisy programů (9 stran)

**Příloha 1:**

Stavový diagram JSA

Test Port == value from tab

-

-

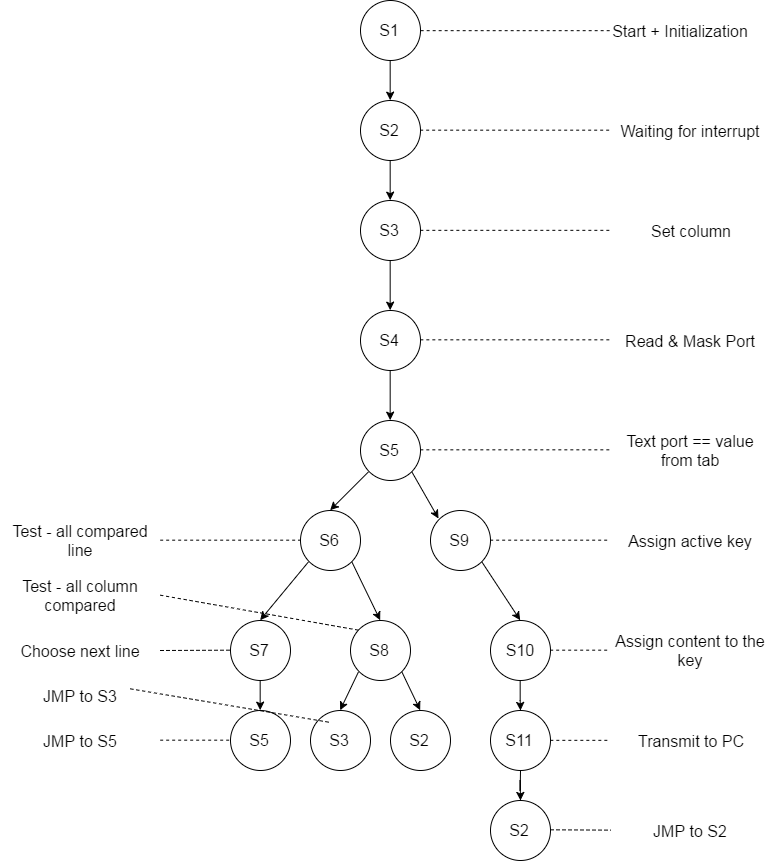
+

-

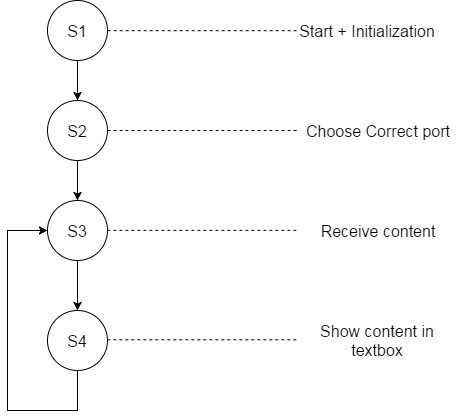
+

+

Ev. - Interrupt



Stavový diagram C#



**Příloha 12:**

Výpis programu JSA

.Nolist

;Include File:

.Include "m128def.inc"

.List

.ORG 0x0000

JMP Initialization

.ORG 0x001E ;Timer0 interrupt

JMP MatrixKeypad

.ORG 0x0046

.CSEG

;Definition register and expression

.DEF temp1 = R16

.DEF temp2 = R17

.DEF index1 = R18 ;cycle counter

.DEF index2 = R19 ;cycle counter

.EQU ddrKey = ddrB

.EQU portKey = PortB

.EQU pinKey = PinB

.EQU Timer0Setting = 0b0000\_0110

.EQU TimskSetting = 0b0000\_0010

.EQU FirstHalf = 0xF0

.EQU SecondHalf = 0x0F

.EQU Column1 = 0b1111\_1110

.EQU Column2 = 0b1111\_1101

.EQU Column3 = 0b1111\_1101

.EQU Column4 = 0b1111\_0111

.EQU Line1 = 0b1110\_1111

.EQU Line2 = 0b1101\_1111

.EQU Line3 = 0b1011\_1111

.EQU Line4 = 0b0111\_1111

.EQU One = 0b1110\_1110

.EQU Two = 0b1110\_1101

.EQU Three = 0b1110\_1011

.EQU Four = 0b1101\_1110

.EQU Five = 0b1101\_1101

.EQU Six = 0b1101\_1011

.EQU Seven = 0b1011\_1110

.EQU Eight = 0b1011\_1101

.EQU Nine = 0b1011\_1011

.EQU Star = 0b0111\_1110

.EQU Zero = 0b0111\_1101

.EQU Hash = 0b0111\_1011

Initialization:

;Stack pointer setting

LDI temp1, LOW(RAMEND)

OUT SPL, temp1

LDI temp1, HIGH(RAMEND)

OUT SPH, temp1

;Port setting

LDI temp1, SecondHalf

OUT ddrKey, temp1

;All Tab definition

CALL AllTab

;USART initialization

;Switch on the transmitter

LDI temp1, (1<<RXEN0)|(1<<TXEN0)

OUT UCSR0B, temp1

;Set frame format: 8data, 2stop bit

LDI temp1, (1<<USBS0)|(3<<UCSZ2)

OUT UCSR0C, temp1

;Set baud rate 9600

LDI temp1, 103

OUT UBRR0L, temp1

;Timer0 & Timsk configuration

LDI temp1, Timer0Setting

OUT TCCR0, temp1

IN temp1, Timsk

ORI temp1, TimskSetting

OUT Timsk, temp1

;Global interrupt enabled

SEI

Program:

RJMP Program

MatrixKeypad: ;interrupt Timer0

LDI XL, LOW(Tab1)

LDI XH, HIGH(Tab1)

LDI index1, 3 ;3 - number of column

SelectColumn:

LDI index2, 4 ;4 - number of line

LD temp1, X+ ;Column from data memory

OUT portKey, temp1 ;Set column

IN temp2, pinKey ;Read port

ANDI temp2, FirstHalf ;Masking

AllocateLine:

LD temp1, X+

CP temp2, temp1 ;Compare port with tab

BREQ EdgeDetected

DEC index2

BRNE AllocateLine ;Next line comparing

DEC index1

BRNE SelectColumn ;Set next column

EndMatrixKeypad: ;Jump to end of interrupt

RETI

EdgeDetected: ;Transmit detected key

AND temp1, temp2 ;Add line and column

LDI XL, LOW(Tab2)

LDI XH, HIGH(Tab2)

;Compare active key with tab

LD temp2, X+

CP temp1, temp2

BREQ KeyOne

LD temp2, X+

CP temp1, temp2

BREQ KeyTwo

LD temp2, X+

CP temp1, temp2

BREQ KeyThree

LD temp2, X+

CP temp1, temp2

BREQ KeyFour

LD temp2, X+

CP temp1, temp2

BREQ KeyFive

LD temp2, X+

CP temp1, temp2

BREQ KeySix

LD temp2, X+

CP temp1, temp2

BREQ KeySeven

LD temp2, X+

CP temp1, temp2

BREQ KeyEight

LD temp2, X+

CP temp1, temp2

BREQ KeyNine

LD temp2, X+

CP temp1, temp2

BREQ KeyStar

LD temp2, X+

CP temp1, temp2

BREQ KeyZero

LD temp2, X+

CP temp1, temp2

BREQ KeyHash

;Assign number to Key

KeyOne:

LDI temp1, 1

RJMP Transmit

KeyTwo:

LDI temp1, 2

RJMP Transmit

KeyThree:

LDI temp1, 3

RJMP Transmit

KeyFour:

LDI temp1, 4

RJMP Transmit

KeyFive:

LDI temp1, 5

RJMP Transmit

KeySix:

LDI temp1, 6

RJMP Transmit

KeySeven:

LDI temp1, 7

RJMP Transmit

KeyEight:

LDI temp1, 8

RJMP Transmit

KeyNine:

LDI temp1, 9

RJMP Transmit

KeyStar:

LDI temp1, 10

RJMP Transmit

KeyZero:

LDI temp1, 0

RJMP Transmit

KeyHash:

LDI temp1, 11

RJMP Transmit

Transmit:

OUT UDR1, temp1

JMP EndMatrixKeypad

;SubProgram

AllTab:

;Reserve Place For Tab1

.DSEG

Tab1: .BYTE 15

.CSEG

;Fill Tab1

LDI XL, LOW(Tab1)

LDI XH, HIGH(Tab1)

;Storage value in tab in same position as we read

LDI temp1, Column1 ;1

ST X+, temp1

LDI temp1, Line1

ST X+, temp1

LDI temp1, Line2

ST X+, temp1

LDI temp1, Line3

ST X+, temp1

LDI temp1, Line4

ST X+, temp1

LDI temp1, Column2 ;2

ST X+, temp1

LDI temp1, Line1

ST X+, temp1

LDI temp1, Line2

ST X+, temp1

LDI temp1, Line3

ST X+, temp1

LDI temp1, Line4

ST X+, temp1

LDI temp1, Column3 ;3

ST X+, temp1

LDI temp1, Line1

ST X+, temp1

LDI temp1, Line2

ST X+, temp1

LDI temp1, Line3

ST X+, temp1

LDI temp1, Line4

ST X+, temp1

;Reserve Place For Tab2

.DSEG

Tab2: .BYTE 12

.CSEG

;Fill Tab2

LDI XL, LOW(Tab2)

LDI XH, HIGH(Tab2)

;Storage value in tab in same position as we read

LDI temp1, One

ST X+, temp1

LDI temp1, Two

ST X+, temp1

LDI temp1, Three

ST X+, temp1

LDI temp1, Four

ST X+, temp1

LDI temp1, Five

ST X+, temp1

LDI temp1, Six

ST X+, temp1

LDI temp1, Seven

ST X+, temp1

LDI temp1, Eight

ST X+, temp1

LDI temp1, Nine

ST X+, temp1

LDI temp1, Star

ST X+, temp1

LDI temp1, Zero

ST X+, temp1

LDI temp1, Hash

ST X+, temp1

;End of Tabs

RET

Výpis programu C#

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO.Ports;

using System.Windows.Forms;

namespace Seriál\_Communication

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent(); //Initialization

static bool \_continue;

static SerialPort \_serialPort;

textBox\_SerialPort.Text = "";

\_serialPort = new SerialPort(); //create a new SerialPort

\_serialPort.DataBits = SetPortDataBits(\_serialPort.DataBits);

}

private void Timer1\_Tick(object sender, EventArgs e)

{

//try all allowed SerialPort

foreach (string s in System.IO.Ports.SerialPort.GetPortNames())

{

//read all data bits

string dataBits = dataBits + defaultPortDataBits.ToString();

textBox\_SerialPort.Text = dataBits;

}

}

}

}