#define F\_CPU 16000000UL

#define BAUD 9600

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<avr/io.h>

#include<util/setbaud.h>

#include<util/delay.h>

#include<avr/interrupt.h>

#include<avr/eeprom.h>

#define DIVIDER\_DID 0x03

#define OVRFLW\_DIS 0x01

#define PORT\_X PORTE

#define DDR\_X DDRE

// DEKLARACE PROMĚNNÝCH PRO PŘERUŠENÍ

volatile int N=0; // INDEX POLE DEKODOVANÝCH HODNOT

volatile int VALUE\_POOL [15]; // POLE DEKODOVANÝCH HODNOT

volatile int Krtecek=0; // URČUJE INDEX POSLEDNÍ ZMÁČKLÉ HODNOTY

volatile int Lojza=0; // URČUJE INDEX HODNOT V POLI ZMÁČKLÝCH HODNOT V CYKLU

ZOBRAZENÍ

volatile int M=0; // URČUJE ŘÁDEK V POLI ČÍSLIC

volatile int MaxiPesFik=0; // 0: NEZMÁČKL SE ENTER PRO PŘEHRÁVÁNÍ HODNOT 1: ZMÁČKL SE ENTER PRO PŘEHRÁVÁNÍ HODNOT

volatile int Pat=0; // POČÍTÁ POČET PŘERUŠNÍ

volatile int interval=2000; // INTERVALUJE

volatile uint8\_t NUMBER\_POOL [10][7]=

{

{ /\*0\*/

0b11001000,

0b10110001,

0b10110010,

0b10110011,

0b10110100,

0b10110101,

0b11001110,

},

{ /\*1\*/

0b11101000,

0b11001001,

0b10101010,

0b11101011,

0b11101100,

0b11101101,

0b11101110,

},

{ /\*2\*/

0b11001000,

0b10110001,

0b11110010,

0b11101011,

0b11011100,

0b10111101,

0b10000110,

},

{ /\*3\*/

0b11001000,

0b10110001,

0b11110010,

0b11101011,

0b11110100,

0b10110101,

0b11001110,

},

{ /\*4\*/

0b10111000,

0b10101001,

0b10000010,

0b11101011,

0b11101100,

0b11101101,

0b11101110,

},

{ /\*5\*/

0b10000000,

0b10111001,

0b10001010,

0b11110011,

0b11110100,

0b10110101,

0b11001110,

},

{ /\*6\*/

0b11001000,

0b10111001,

0b10111010,

0b10001011,

0b10110100,

0b10110101,

0b11001110,

},

{ /\*7\*/

0b11111000,

0b10000001,

0b11110010,

0b11101011,

0b11011100,

0b10111101,

0b11111110,

},

{ /\*8\*/

0b11001000,

0b10110001,

0b10110010,

0b11001011,

0b10110100,

0b10110101,

0b11001110,

},

{ /\*9\*/

0b11001000,

0b10110001,

0b10110010,

0b11000011,

0b11110100,

0b10110101,

0b11001110,

}

};

void TIME(void)

{

TCCR0 = DIVIDER\_DID; // DĚLIČKA

TIMSK |= OVRFLW\_DIS; // PŘETEČENÍ

sei();

}

ISR (TIMER0\_OVF\_vect) // PŘETEČNÍ

{

Krtecek=N-1;

if (MaxiPesFik == 0)

{

PORT\_X = NUMBER\_POOL [VALUE\_POOL [Krtecek]] [M];

M++;

if (M>7)

M=0;

}

else

{

PORT\_X = NUMBER\_POOL [VALUE\_POOL [Lojza]] [M];

M++;

Pat++;

if (Pat == interval)

{

Pat=0;

Lojza++;

if (Lojza>Krtecek)

Lojza=0;

}

if (M>7)

M=0;

}

}

void hw\_init(void)

{

DDRB = 0x0F;

DDRD = 0x0F;

DDR\_X = 0xFF;

}

uint8\_t GET\_KEY(void) //ZJIŠTĚNÍ STIKLÉ KLÁVESY

{

uint8\_t COLUMN=0; //SLOUPEC

uint8\_t ROW=0; //ŘÁDEK

uint8\_t COMPARE=0; //POROVNÁNÍ

for (COLUMN=0;COLUMN<4;COLUMN++)

{

PORTB=~(1<<COLUMN); //BITOVÝ POSUN PODLE SLOUPCE

\_delay\_us(20); //ZPOŽDĚNÍ V MIKROSEKUNDÁCH

for (ROW=4;ROW<8;ROW++)

{

COMPARE=~(1<<ROW) & 0xF0; //BITOVÝ POSUN PODLE ŘÁDKU A MASKOVÁNÍ

if ((PINB & 0xF0)==COMPARE) //POROVNÁNÍ

return (((ROW-4)\*4)+COLUMN); //VRÁCENÍ HODNOTY

}

}

return(128);

}

uint8\_t KEY\_DECODER (uint8\_t SCAN\_CODE)

{

// ROZVRŽENÍ REÁLNÉ KLÁVESNICE

uint8\_t KEY\_POOL[16]={1,2,3,10,4,5,6,11,7,8,9,12,15,0,14,13};

return(KEY\_POOL[SCAN\_CODE]);

}

int main (void)

{

hw\_init();

int LAST\_BTTN = 0; // HODNOTA POSLEDNÍHO ZMÁČKLÉHO TLAČÍTKA

int KEY\_PAST = 128;

int DECODER\_VALUE = 0; // DEKODOVANÁ HODNOTA

while (1)

{

//PORT\_X = 0b00000100;

LAST\_BTTN = GET\_KEY ();

// PODMÍNKA, KTERÁ SE VYKONÁ, JEN KDYŽ JE TAM NÁBĚŽNÁ HRANA

if ((KEY\_PAST == 128) && (LAST\_BTTN != 128))

{

// JESTLI-ŽE V PŘEDCHOZÍM KROKU NEBYLO ZMÁČKLÉ TLAČÍTKO A TEĎ JE

VYKONEJ TOTO:

DECODER\_VALUE = KEY\_DECODER (LAST\_BTTN); // DEKÓDOVÁNÍ HODNOTY

if ((DECODER\_VALUE >= 0) && (DECODER\_VALUE <= 9))

{

VALUE\_POOL [N] = DECODER\_VALUE;

N++;

}

// ZMÁČKL SE ENTER A V POLI JE ULOŽENO MIN 5 A MAX 15 HODNOT

if ((DECODER\_VALUE == 13) && (N>4) && (N<16))

{

if (MaxiPesFik==0)

MaxiPesFik=1;

else

MaxiPesFik=0;

}

}

KEY\_PAST = LAST\_BTTN;

}

}