

```

/*
Slidersteuerung.ino
hmr-video.de
25.05.2013
*/

// Eingesetzte librarys:
#include <Arduino.h>
#include <LiquidCrystal.h>
#include <sainsmartkeypad.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);           // Auswahl der LCD Kontakte
SainsmartKeypad keypad(0);

uint8_t key;
byte Menu=1;
int i=1;
int myArray[]={380, 240, 160, 100, 70, 60, 50, 45, 40, 35}; // Ermittelte Laufzeiten in s für 1 m bei entsprechender Vorschubeinstellung
byte Vorschub=1; // Vorschubeinstellung am Poti
long Schritt=1000; // Schrittvorstellung der Intervallzeit in Millisekunden
int z_faktor=1000; // Zeitfaktor Umrechnung von Millisekunden in Sekunden
int motor_z=250; // Motorlaufzeit
int kamera_z=150; // Kameralaufzeit
int Motor=12;
int Kamera=11;

void setup() {
  pinMode(Motor, OUTPUT);
  pinMode(Kamera, OUTPUT);
  byte downpfeil[8] = {B00000, B11111, B11111, B01110, B01110, B00100, B00100, B00000};
  lcd.createChar (1, downpfeil); // Erstellt ein nach unten gerichtetes Pfeildreieck
  lcd.begin(16, 2); // Start der LCD library
  lcd.setCursor(0,0); // Cursor position 1. Zeile, 1. Feld
  lcd.print("Slidersteuerung");
  lcd.setCursor(3,1); // Cursor position 2. Zeile, 3. Feld
  lcd.print("hmr-video");
  delay(2000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Einstellungen");
  lcd.setCursor(4,1);
  lcd.print("press ");
  lcd.write(1); // zeigt das erstellte Pfeildreieck an
}

void loop() {
  key = keypad.getKey_fastscroll(); // Anzeige wird nur nach einer Tastenbetätigung aktualisiert

  if(key != SAMPLE_WAIT) {
    switch(key)

```

```

{
case UP_KEY:
  Menu--;
  if (Menu<2) Menu=2;
  lcd.clear();
  break;

case DOWN_KEY:
  Menu++;
  if (Menu>4) Menu=4;
  lcd.clear();
  break;
}

switch(Menu) {

case 1:
  lcd.setCursor(1,0);
  lcd.print("Einstellungen");
  lcd.setCursor(4,1);
  lcd.print("press ");
  lcd.write(1);
  break;

case 2:
  lcd.setCursor(4,0);
  lcd.print("Vorschub");

  if(key == LEFT_KEY) {
    Vorschub=Vorschub-1;
    lcd.setCursor(0,1);
    lcd.print(" ");
  }

  if(key == RIGHT_KEY) {
    Vorschub=Vorschub+1;
    lcd.setCursor(0,1);
    lcd.print(" ");
  }

  if(Vorschub<1) { // Vorschub kann nicht unter 1 eingestellt werden
    Vorschub=1;
    lcd.setCursor(7,1);
    lcd.print(Vorschub);
    lcd.print(" >");
  }
  else if(Vorschub>10) { // Vorschub kann nicht über 10 eingestellt werden
    Vorschub=10;
    lcd.setCursor(5,1);

```

```

lcd.print("< ");
lcd.print(Vorschub);
}
else {
lcd.setCursor(7,1);
lcd.print(Vorschub);
}
break;

```

case 3:

```

lcd.setCursor(1,0);
lcd.print("Intervallzeit");

```

```

if(key == LEFT_KEY) {
Schritt=Schritt-1000;
}

```

```

if(Schritt>1000) {
lcd.setCursor(0,1);
lcd.print(" ");
}

```

// Zeit kann nicht unter 1s eingestellt werden

```

else {
Schritt=1000;
lcd.setCursor(0,1);
lcd.print(" ");
}

```

```

if(key == RIGHT_KEY) {
Schritt=Schritt+1000;
lcd.setCursor(0,1);
lcd.print(" ");
}

```

```

if(Schritt>1000) {
lcd.setCursor(6,1);
lcd.print(Schritt/z_faktor);
lcd.print(" s");
}

```

// Bei eingestellter Zeit von 1s

// wird das < Zeichen nicht mehr angezeigt

```

else {
Schritt=1000;
lcd.setCursor(6,1);
lcd.print(Schritt/z_faktor);
lcd.print(" s >");
}
break;

```

case 4:

```

lcd.setCursor(0,0);
lcd.print("Zeitd.: ");

```

// Von den gewählten Einstellungen wird die Zeitdauer und die Anzahl der Intervalle berechnet

```

if (((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1]))/3600<10) lcd.print("0");

```

// damit die Zahl immer zweistellig angezeigt wird

```

lcd.print(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])/3600); // Zeitdauer Umrechnung in h
lcd.print("-");
if (((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])-(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])/3600)*3600)/60<10) lcd.print("0"); // damit die Zahl immer zweistellig angezeigt wird
lcd.print(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])-(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])/3600)*3600)/60); // Zeitdauer Umrechnung in m
lcd.print("-");
if ((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])-(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])/60)*60<10) lcd.print("0"); // damit die Zahl immer zweistellig angezeigt wird
lcd.print((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])-(((1000/motor_z)*(Schritt/z_faktor)*(myArray[Vorschub-1])/60)*60); // Zeitdauer Umrechnung in s
lcd.setCursor(0,1);
lcd.print("Interv.: ");
lcd.print((myArray[Vorschub-1]*(1000/motor_z)); // Anzahl der Intervalle
break;
}

if(key == SELECT_KEY) {
for (int i=0; i<(myArray[Vorschub-1]*(1000/motor_z); i++) // Schleife wird mit eingestellter Sekundenzeit x i_faktor durchlaufen; Erhoehung um 1
{
digitalWrite(Kamera, HIGH); // schaltet Kamera ein
delay(kamera_z); // Laufzeit der Kamera
digitalWrite(Kamera, LOW); // schaltet Kamera aus
delay((Schritt/2)-(motor_z+kamera_z)); // 1/2 gewählte Zeit minus Motor- und Kameralaufzeit
digitalWrite(Motor, HIGH); // schaltet Motor ein
delay(motor_z); // Laufzeit des Motors
digitalWrite(Motor, LOW); // schaltet Motor aus
delay(Schritt/2); // 1/2 gewählte Zeit

if(((myArray[Vorschub-1]*(1000/motor_z)-i)>1) { // Intervalle werden gezählt
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Intervalle");
lcd.setCursor(0,1);
lcd.print((myArray[Vorschub-1]*(1000/motor_z)-(i+1)); // Restintervalle von
lcd.print(" von ");
lcd.print((myArray[Vorschub-1]*(1000/motor_z)); // Gesamtintervalle werden angezeigt
}
else {
lcd.clear(); // Sind alle Intervalle abgearbeitet wird Anzeige gelöscht
lcd.setCursor(1,0); // und Programm Ende für 5 s angezeigt
lcd.print("Programm Ende");
lcd.setCursor(3,1);
lcd.print("hmr-video");
delay(5000);
lcd.clear(); // dann wird wieder die Zeitdauer und Intervallzeit angezeigt
}
}
}
} //Ende von void loop

```