

```

1  /*
2   Name:      FinaliPrograms.ino
3   Created:   12/13/2018 11:11:59 PM
4   Author:    Pavel
5  */
6
7  //Inicializace knihoven
8  #include "IRremote2.h"
9  #include <Wire.h>
10 #include <Eeprom24C04_16.h>
11
12 //inicializace IR vysilani
13 IRsend irSend;
14 IRsend irSend2;
15
16 const int FrekvenceIR_Vysilani = 36; //frekvence vysilani IR LED
17                                     //Prubeh signalu u platna
18 const unsigned int IrSignalPlatnoNahoru[] = { 1300,350, 1300,400, 1250,400, 1250,400,  ↗
19                                               450,1250, 400,1250, 400,1300, 400,1250, 450,1200, 450,1250, 1200,450, 500 }; //nahoru
20 const unsigned int IrSignalPlatnoDolu[] = { 1200,450, 1300,400, 1250,400, 1300,350,  ↗
21                                               450,1250, 500,1150, 450,1200, 450,1250, 450,1200, 1250,450, 400,1250, 450 }; //dolu
22 const unsigned int IrSignalPlatnoStop[] = { 1300,400, 1250,400, 1250,400, 1250,450,  ↗
23                                               400,1250, 400,1250, 450,1250, 450,1200, 400,1300, 400,1250, 450,1200, 1300 };
24
25 //Prubeh signalu u Projektortu
26
27 const unsigned int IrSignalProjektorVipinani[] = { 8950,4450, 600,500, 550,550, 600,500,  ↗
28                                                     600,550, 600,500, 600,500, 600,500, 600,550, 600,500, 600,500, 600,500, 600,550,  ↗
29                                                     600,1600, 600,1650, 550,550, 600,500, 600,550, 600,1600, 600,550, 600,500, 600,500,  ↗
30                                                     600,500, 600,550, 600,500, 600,1650, 550,550, 600,1600, 600,1650, 600,1650, 600,1650,  ↗
31                                                     550,1650, 600,1650, 550 }; //vypinani
32 const unsigned int IrSignalProjektorFreeze[] = { 8950,4400, 650,500, 600,500, 550,550,  ↗
33                                                     550,550, 650,500, 550,550, 650,450, 600,500, 650,500, 550,550, 600,500, 600,500,  ↗
34                                                     600,1650, 550,1700, 600,500, 600,500, 600,1650, 650,1600, 600,500, 550,550, 600,500,  ↗
35                                                     650,500, 600,500, 600,500, 600,550, 600,500, 600,1600, 600,1650, 650,1600, 600,1650,  ↗
36                                                     600,1600, 650,1600, 600 }; //freeze
37 const unsigned int IrSignalProjektorSipkaNahoru[] = { 8900,4450, 600,500, 600,550,  ↗
38                                                         550,550, 550,550, 550,550, 600,550, 550,550, 550,550, 550,550, 600,550, 550,550,  ↗
39                                                         550,550, 550,1700, 550,1650, 600,550, 550,550, 550,1700, 600,1600, 600,550, 550,1650,  ↗
40                                                         600,550, 550,550, 600,500, 600,500, 600,550, 550,550, 550,1700, 550,550, 550,1650,  ↗
41                                                         600,1650, 600,1650, 550,1650, 600 }; //nahoru
42 const unsigned int IrSignalProjektorSipkaDolu[] = { 8900,4450, 600,550, 550,550, 600,500,  ↗
43                                                         550,550, 600,550, 550,550, 600,500, 600,500, 600,550, 600,500, 600,500, 600,500,  ↗
44                                                         600,1650, 550,1700, 600,500, 550,600, 550,550, 550,550, 550,1650, 600,1650, 600,500,  ↗
45                                                         600,550, 550,550, 550,600, 550,1650, 600,1650, 550,550, 550,550, 550,1700, 600,1600,  ↗
46                                                         600,1650, 550,1700, 550 }; //dolu
47 const unsigned int IrSignalProjektorSource[] = { 8900,4450, 550,600, 500,600, 500,550,  ↗
48                                                     650,550, 500,550, 550,600, 550,650, 450,550, 550,600, 550,550, 550,550, 550,550,  ↗
49                                                     600,1650, 550,1650, 600,600, 450,600, 550,600, 550,550, 500,1750, 550,600, 500,550,  ↗
50                                                     550,550, 600,600, 500,600, 650,1550, 550,1700, 500,600, 550,1700, 500,1700, 550,1700,  ↗
51                                                     550,1700, 600,1650, 500 };
52 const unsigned int IrSignalProjektorOk[] = { 9000,4450, 600,500, 600,500, 650,500,  ↗
53                                                600,500, 650,500, 600,500, 600,550, 600,500, 600,550, 600,500, 600,550, 600,500,  ↗
54                                                600,1650, 600,1650, 600,500, 650,500, 600,500, 650,500, 600,500, 650,500, 600,1650,  ↗
55                                                600,500, 600,550, 600,500, 600,1650, 600,1650, 600,1650, 600,1650, 600,500, 650,1600,  ↗
56                                                650,1600, 600,1650, 600 }; // NEC C08F7
57
58 //Cislo tlacitek
59 #define TlacitkoZapVyp 0
60 #define TlacitkoMonitor 1
61 #define TlacitkoKamera 2
62 #define TlacitkoSpecial 3

```

```

35 #define TlacitkoFreeze 4
36 #define TlacitkoPlatno 5
37
38 //Systemove cislo nastavovacich mikrospinacu
39 #define MicroSpinSET 6
40 #define MicroSpinAUTO 7
41 //Vzstup z posuvneho registru pri zjistovani stavu tlacitek a nastavovacich mikrospinacu
42 byte MozPrep[] =
    { B00000001,B00000010,B00000100,B00001000,B00010000,B00100000,B01000000,B10000000,B0000
      0000 };
43
44 //Pini na ATmega328P
45 #define PinLatch 6 //RCLK (12)
46 #define PinClock 7 //SRCLK (11)
47 #define PinData 5 //Data (14)
48 #define PinIrLED 3
49 #define PinTlac 8
50 #define PinKont 2
51
52 //Moznasti specialniho tlacitka
53 #define Jako_canal 0
54 #define Jako_STOP_platna 1
55 #define Jako_bez_funkce 2
56 #define Jako_vypis_Help 3
57 #define Jako_jine 4
58 #define Jako__Max_hodnot 5
59 byte Special_Tlacitko = Jako_STOP_platna;
60
61 //Hlavni promene
62 char vykonavanyPrikaz = 'X'; //Hlavní proměnná pro prepínání canalu: V=Vip/Zap;
    M=Canal_Monitor; C=Kamera; S=Canal_Special; F=Freeze; P=Platno nahoru/dolu;
63 bool detekceZadaniPrikazu = false; //Detekce změni - detekce prikazu, ukonu
64 byte registrTlacitka = B00000000; //Registr na tlacitka - pomoci teho promene arduino
    zjistuje jake je zmačknute tlacitko
65 byte registrZobrazeni = B00000000; //Zobrazení na LED
66 bool StavProjektor = false; //zapinani/vypnuto projektor
67 bool StavPlatno = false; //Vytazene/nasunute platno
68 bool StavFreeze = false; //Zamrazen/odmrazen obraz
69 bool StavSpecial = false;
70 bool DetekceZmenyCanaluProjektoru = false;
71 char zadanyPrikaz = 0; //promena, která se pouziva pri nacistani tlacitek
72 char CanalCil;
73
74 //Pohyb pri posunovani na ve Source
75 char canalNow = 0;
76 char canalOld = 0;
77 char kroky = 0;
78
79 //Vypis konektoru projektoru v Source
80
81 #define KonektorCanalProjektor__Pocet 5
82 #define KonektorCanalProjektor_YBp1Pr1_Pocitac1 0
83 #define KonektorCanalProjektor_YBp2Pr2_Pocitac2 1
84 #define KonektorCanalProjektor_HDMI 2
85 #define KonektorCanalProjekort_Vidio 3
86 #define KonektorCanalProjektor_S_Video 4
87
88
89 //Prizazeni canalu s kterymi pracujeme
90 #define Canal_Monitor KonektorCanalProjektor_YBp1Pr1_Pocitac1
91 #define Canal_Camera KonektorCanalProjekort_Vidio

```

```
92 #define Canal_Special    KonektorCanalProjektor_HDMI
93
94 //Instrukce pro cteni a zapis na Eeprom
95 #define EepromNow         true
96 #define EepromOld         false
97 #define RadekEepromNowOld 20
98 #define RadekEepromProjektor 0
99 #define RadekEepromPlatno 1
100 #define RadekEepromFreeze 2
101 #define RadekEepromKanal 3
102 #define RadekEepromLed 4
103 #define RadekEepromSpecial 5
104 #define RadekEepromSpecialTyp 6
105 #define EEPROM_ADDRESS 0x50
106 static Eeprom24C04_16 eeprom(EEPROM_ADDRESS);
107 #define KontolaEeprom_pocet 5
108 byte KontolaEepromZadano[KontolaEeprom_pocet] = { 1,2,3,4,5 };
109 byte KontolaEepromPrecteno[KontolaEeprom_pocet];
110 #define PocetMistZapisu 7
111 byte DataNactena[PocetMistZapisu];
112 #define PocatecniMistZapisu 10
113
114 //Seznam cekani na ...
115 #define DobaCekaniTlacitka 5 //ms
116 #define DobaCekaniNaOffProj 1000//ms
117 #define DobaBlikuZakazFunkce 100//ms
118
119 //klavesove prikazy
120 #define ZapVypKlaves0 'V'
121 #define ZapVypKlaves1 'v'
122 #define ZapVypKlaves2 'Z'
123 #define ZapVypKlaves3 'z'
124 #define PlatnoKlaves0 'P'
125 #define PlatnoKlaves1 'p'
126 #define FreezeKlaves0 'F'
127 #define FreezeKlaves1 'f'
128 #define MonitorKlaves0 'm'
129 #define MonitorKlaves1 'M'
130 #define KameraKlaves0 'C'
131 #define KameraKlaves1 'c'
132 #define KameraKlaves2 'K'
133 #define KameraKlaves3 'k'
134 #define SpecialKlaves0 'S'
135 #define SpecialKlaves1 's'
136 #define Special_SET_Klaves0 '#'
137 #define Langue_SET_Klaves0 '@'
138
139 //Zobrazeni LED
140 #define ZobVypOff B00000001
141 #define ZobVypOn B00000010
142 #define ZobMonitor B00000100
143 #define ZobCamera B00001000
144 #define ZobSpecial B00010000
145 #define ZobFreeze B00100000
146 #define ZobPlatNah B01000000
147 #define ZobPlatDol B10000000
148 #define ZobPlat B11000000
149 #define ZobCanalSSpecial B11100011
150 #define ZobCanalBezSpecial B11110011
151 byte ZobCanal = ZobCanalSSpecial;
152
```

```
153
154 void setup()
155 {
156     //Inicilializovani pini
157     pinMode(PinKont, OUTPUT);
158     pinMode(PinTlac, INPUT);
159     pinMode(PinLatch, OUTPUT);
160     pinMode(PinClock, OUTPUT);
161     pinMode(PinData, OUTPUT);
162     //PinIrLED neni nutne nainicializovat
163     digitalWrite(PinKont, HIGH);
164     eeprom.initialize();
165     Serial.begin(9600);    //Inicializace seriove komunikace
166     delay(1000);
167     NacteniEeprom();
168     delay(1000);
169     digitalWrite(PinKont, LOW);
170 }
171
172
173 void loop()
174 {
175     zadanyPrikaz = 0;
176     detekceZadaniPrikazu = false;
177     if (Serial.available() > 0)
178     {
179         zadanyPrikaz = Serial.read();
180         if (zadanyPrikaz >= '!' && zadanyPrikaz <= 'z')
181         {
182             digitalWrite(PinKont, HIGH);
183             detekceZadaniPrikazu = true;
184         }
185     }
186     else
187     {
188         zadanyPrikaz = 0;
189         do
190         {
191             registrTlacitka = MozPrep[zadanyPrikaz];
192             Registr();
193             delay(DobaCekaniTlacitka);
194             if (digitalRead(PinTlac) == HIGH)
195             {
196                 digitalWrite(PinKont, HIGH);
197                 detekceZadaniPrikazu = true;
198             }
199             else
200             {
201                 zadanyPrikaz += 1;
202             }
203         } while (detekceZadaniPrikazu == false && zadanyPrikaz <= 5);
204     }
205
206     if (detekceZadaniPrikazu == true)
207     {
208         delay(1000);
209         PrvotniZpracovani();
210         if (DetekceZmenyCanaluProjektoru == true)
211         {
212             CanalyProjektoru();
213         }
214     }
215 }
```

```
214     do
215     {
216         delay(100);
217         registrTlacitka = MozPrep[zadanyPrikaz];
218         Registr();
219         delay(DobaCekaniTlacitka);
220     } while (digitalRead(PinTlac) == HIGH);
221     EepromUkladani();
222     digitalWrite(PinKont, LOW);
223 }
224 }
225
226
227 void NacteniEeprom()
228 {
229     registrZobrazeni = ZobVypOff | ZobPlatNah;
230     if (NacteniEepromNowOld() == EepromNow)
231     {
232         //Inicializace
233         eeprom.writeBytes(RadekEepromNowOld, KontolaEeprom_pocet, KontolaEepromZadano);
234
235         DataNactena[RadekEepromProjektor] = false;
236         DataNactena[RadekEepromPlatno] = false;
237         DataNactena[RadekEepromKanal] = Canal_Monitor;
238         DataNactena[RadekEepromFreeze] = false;
239         DataNactena[RadekEepromLed] = registrZobrazeni;
240         DataNactena[RadekEepromSpecial] = false;
241         DataNactena[RadekEepromSpecialTyp] = Jako_canal;
242         eeprom.writeBytes(PocatecniMistZapisu, PocetMistZapisu, DataNactena);
243         Chyba(1);
244     }
245     if (NacteniEepromNowOld() == EepromOld)
246     {
247         eeprom.readBytes(PocatecniMistZapisu, PocetMistZapisu, DataNactena);
248         StavProjektor = DataNactena[RadekEepromProjektor];
249         StavPlatno = DataNactena[RadekEepromPlatno];
250         StavFreeze = DataNactena[RadekEepromFreeze];
251         canalOld = DataNactena[RadekEepromKanal];
252         StavSpecial = DataNactena[RadekEepromSpecial];
253         registrZobrazeni = DataNactena[RadekEepromLed];
254         Special_Tlacitko = DataNactena[RadekEepromSpecialTyp];
255     }
256     else
257     {
258         StavProjektor = true;
259         StavPlatno = false;
260         StavFreeze = false;
261         canalOld = Canal_Monitor;
262         Chyba(2);
263     }
264     Registr();
265     delay(100);
266     Registr();
267 }
268
269
270 bool NacteniEepromNowOld() //Old = 0   Now = 1
271 {
272     bool Potvrzeni = EepromOld;
273     eeprom.readBytes(RadekEepromNowOld, KontolaEeprom_pocet, KontolaEepromPrecteno);
274     for (byte i = 0; i < KontolaEeprom_pocet; i++)
```

```
275     {
276         if (KontolaEepromPrecteno[i] != KontolaEepromZadano[i])
277         {
278             Potvrzeni = EepromNow;
279         }
280     }
281     return Potvrzeni;
282 }
283
284
285 void EepromUkladani()
286 {
287     DataNactena[RadekEepromProjektor] = StavProjektor;
288     DataNactena[RadekEepromPlatno] = StavPlatno;
289     DataNactena[RadekEepromKanal] = canalOld;
290     DataNactena[RadekEepromFreeze] = StavFreeze;
291     DataNactena[RadekEepromLed] = registrZobrazeni;
292     DataNactena[RadekEepromSpecial] = StavSpecial;
293     DataNactena[RadekEepromSpecialTyp] = Special_Tlacidko;
294     eeprom.writeBytes(PocatecniMistZapisu, PocetMistZapisu, DataNactena);
295 }
296
297
298 void PrvotniZpracovani()
299 {
300     switch (zadanyPrikaz)
301     {
302     case TlacidkoZapVyp:
303     case ZapVypKlaves0:
304     case ZapVypKlaves1:
305     case ZapVypKlaves2:
306     case ZapVypKlaves3://Zapnuti/vypnuti
307         vykonavanyPrikaz = 'Z';//zapinani
308         if (StavProjektor == true)
309         {
310             //vypinani
311             Serial.println(F("Vypinani projektoru"));
312             StavProjektor = false;
313             registrZobrazeni = registrZobrazeni & ZobPlat;//nemenit, funkce jsou
314                 nefunkcni
315             registrZobrazeni = registrZobrazeni | ZobVypOff;
316             StavFreeze = false;
317             VysilaniIR('R', 'V');
318             delay(DobaCekaniNaOffProj); //kvuli casove oddeleni potvrzrni vipinaciho
319                 signalu
320         }
321     else
322     {
323         //zapinani
324         Serial.println(F("Zapinani projektoru"));
325         StavProjektor = true;
326         registrZobrazeni = registrZobrazeni & ~ZobVypOff;
327         registrZobrazeni = registrZobrazeni | ZobVypOn;
328         switch (canalOld)
329         {
330         case Canal_Monitor:
331             registrZobrazeni = registrZobrazeni | ZobMonitor;
332             Serial.println(F("Obraz z monitoru (pocitace)"));
333             break;
334         case Canal_Camera:
335             registrZobrazeni = registrZobrazeni | ZobCamera;
```

```
334     Serial.println(F("Obraz z kamery"));
335     canalNow = Canal_Camera;
336     break;
337     case Canal_Special:
338         if (Special_Tlacitko == Jako_canal)
339         {
340             registrZobrazeni = registrZobrazeni | ZobSpecial;
341             Serial.println(F("Obraz ze stavebnice"));
342             canalNow = Canal_Special;
343         }
344         break;
345     default:
346         break;
347 }
348 }
349 VysilaniIR('R', 'V');//vzdi to bude vyslano
350 if (AUTO() && StavProjektor != StavPlatno)
351 {
352     delay(DobaCekaniNaOffProj);
353     PlatnoOvladani();
354 }
355 break;//Platno nic neudela pokud je mod SET nebo pokud Zap a Plat se rovnaj...
356
357 case TlacitkoPlatno:
358 case PlatnoKlaves0:
359 case PlatnoKlaves1:
360     PlatnoOvladani();
361     break;
362
363 case TlacitkoFreeze:
364 case FreezeKlaves0:
365 case FreezeKlaves1:
366     vykonavanyPrikaz = 'F';//free
367     FreezeOvladani();
368     break;
369
370 case TlacitkoMonitor:
371 case MonitorKlaves0:
372 case MonitorKlaves1:
373     vykonavanyPrikaz = 'M';//pocitac
374     DetekceZmenyCanaluProjektoru = true;
375     registrZobrazeni = registrZobrazeni & ZobCanal;
376     registrZobrazeni = registrZobrazeni | ZobMonitor;
377     Registr();
378     Serial.println(F("Obraz z monitoru (pocitace)"));
379     canalNow = Canal_Monitor;
380     break;
381
382 case TlacitkoKamera:
383 case KameraKlaves0:
384 case KameraKlaves1:
385 case KameraKlaves2:
386 case KameraKlaves3:
387     vykonavanyPrikaz = 'C';//kamera
388     DetekceZmenyCanaluProjektoru = true;
389     registrZobrazeni = registrZobrazeni & ZobCanal;
390     registrZobrazeni = registrZobrazeni | ZobCamera;
391     Registr();
392     Serial.println(F("Obraz z kamery"));
393     canalNow = Canal_Camera;
394     break;
```

```
395
396     case TlacitkoSpecial:
397     case SpecialKlaves0:
398     case SpecialKlaves1://Bez funkce:
399         vykonavanyPrikaz = 'S';
400         switch (Special_Tlacitko)
401         {
402             case Jako_canal:
403                 DetekceZmenyCanaluProjektoru = true;
404                 registrZobrazeni = registrZobrazeni & ZobCanal;
405                 registrZobrazeni = registrZobrazeni | ZobSpecial;
406                 Registr();
407                 Serial.println(F("Obraz ze stavebnice"));
408                 canalNow = Canal_Special;
409                 break;
410             case Jako_STOP_platna:
411                 if (StavSpecial == true)
412                 {
413                     StavSpecial = false;
414                     registrZobrazeni = registrZobrazeni & ~ZobSpecial;
415                     if (StavPlatno == true)
416                     {
417                         StavPlatno = false;
418                     }
419                     else
420                     {
421                         StavPlatno = true;
422                     }
423                     PlatnoOvladani();
424                 }
425                 else
426                 {
427                     StavSpecial = true;
428                     registrZobrazeni = registrZobrazeni | ZobSpecial;
429                     Serial.println(F("STOP platnu."));
430                     VysilaniIR('L', 'S');
431                 }
432                 break;
433             case Jako_bez_funkce:
434                 break;
435             case Jako_vypis_Help:
436                 registrZobrazeni = registrZobrazeni | ZobSpecial;
437                 Registr();
438                 Help();
439                 registrZobrazeni = registrZobrazeni & ~ZobSpecial;
440                 Registr();
441                 break;
442             case Jako_jine:
443                 break;
444             default:
445                 break;
446         }
447         break;
448
449     case Langue_SET_Klaves0:
450         break;
451
452     case Special_SET_Klaves0:
453         if (Special_Tlacitko == Jako__Max_hodnot - 1)
454         {
455             Special_Tlacitko = 0;
```



```
456     }
457     else
458     {
459         Special_Tlacitko += 1;
460     }
461     if (Special_Tlacitko == Jako_canal)
462     {
463         ZobCanal = ZobCanalSSpecial;
464     }
465     else
466     {
467         ZobCanal = ZobCanalBezSpecial;
468     }
469     break;
470
471     default:
472         Chyba(0);
473 }
474 }
475
476
477 void CanalyProjektoru()
478 {
479     if (StavProjektor == true)
480     {
481         CanalCil = vykonavanyPrikaz;
482         kroky = canalOld - canalNow;
483         if (kroky != 0)
484         {
485             if (abs(kroky) > (KonektorCanalProjektor__Pocet / 2))
486             {
487                 if (kroky < 0)
488                 {
489                     kroky += KonektorCanalProjektor__Pocet;
490                 }
491                 else
492                 {
493                     kroky -= KonektorCanalProjektor__Pocet;
494                 }
495             }
496             if (StavFreeze == true)
497             {
498                 FreezeOvladani();
499                 delay(DobaCekaniNaOffProj);
500             }
501             VysilaniIR('R', 'S');
502             if (kroky > 0)
503             {
504                 for (char i = 0; i < kroky; i++)
505                 {
506                     VysilaniIR('R', 'N');
507                     delay(DobaCekaniNaOffProj);
508                 }
509             }
510             else
511             {
512                 for (char i = 0; i < ((-1)*kroky); i++)
513                 {
514                     VysilaniIR('R', 'D');
515                     delay(DobaCekaniNaOffProj);
516                 }
517             }
518         }
519     }
520 }
```

```
517     }
518     delay(DobaCekaniNaOffProj);
519     VysilaniIR('R', 'O'); //VysilaniIR('R', 'S'); VysilaniIR('R', 'S');
520 }
521 canalOld = canalNow;
522 }
523 else
524 {
525     registrZobrazeni = registrZobrazeni & ZobCanal;
526     Registr();
527     ZakazFunkce();
528 }
529 DetekceZmenyCanaluProjektoru = false;
530 }
531
532
533 void PlatnoOvladani()
534 {
535     vykonavanyPrikaz = 'P'; //platno
536     if (StavPlatno == true)
537     {
538         //zasunovani
539         StavPlatno = false;
540         registrZobrazeni = registrZobrazeni & ~ZobPlatDol;
541         registrZobrazeni = registrZobrazeni | ZobPlatNah;
542         Serial.println(F("Platno se zasunuje"));
543         VysilaniIR('L', 'N');
544     }
545     else
546     {
547         //vytahovani
548         StavPlatno = true;
549         registrZobrazeni = registrZobrazeni & ~ZobPlatNah;
550         registrZobrazeni = registrZobrazeni | ZobPlatDol;
551         Serial.println(F("Platno se vysunuje"));
552         VysilaniIR('L', 'D');
553     }
554 }
555
556
557 void FreezeOvladani()
558 {
559     if (StavProjektor == true)
560     {
561         if (StavFreeze == true)
562         {
563             //odmrazit
564             StavFreeze = false;
565             registrZobrazeni = registrZobrazeni & ~ZobFreeze;
566             Serial.println(F("Obraz je spusten"));
567             VysilaniIR('R', 'F');
568         }
569         else
570         {
571             //zamrazit
572             StavFreeze = true;
573             registrZobrazeni = registrZobrazeni | ZobFreeze;
574             Serial.println(F("Obraz je zamrazen"));
575             VysilaniIR('R', 'F');
576         }
577     }
```

```
578     else
579     {
580         ZakazFunkce();
581     }
582 }
583
584
585 void ZakazFunkce()
586 {
587     Serial.println(F("Projektor je vipnuty. Funkce mimo provoz."));
588
589     Blikani(DobaBlikuZakazFunkce, DobaBlikuZakazFunkce, 8);
590 }
591
592
593 void VysilaniIR(char Typ, char Pokyn)
594 {
595     Registr();
596     if (WORK())
597     {
598         delay(100);
599         if (Typ == 'R')//platno
600         {
601             switch (Pokyn)
602             {
603                 case 'V'://vypinani
604                     irSend.sendRaw(IrSignalProjektorVipinani, sizeof
605                                     (IrSignalProjektorVipinani) / sizeof(IrSignalProjektorVipinani[0]),
606                                     FrekvenceIR_Vysilani);
607                     break;
608                 case 'F':
609                     irSend.sendRaw(IrSignalProjektorFreeze, sizeof(IrSignalProjektorFreeze) /
610                                     sizeof(IrSignalProjektorFreeze[0]), FrekvenceIR_Vysilani);
611                     break;
612                 case 'D':
613                     irSend.sendRaw(IrSignalProjektorSipkaDolu, sizeof
614                                     (IrSignalProjektorSipkaDolu) / sizeof(IrSignalProjektorSipkaDolu[0]),
615                                     FrekvenceIR_Vysilani);
616                     break;
617                 case 'N':
618                     irSend.sendRaw(IrSignalProjektorSipkaNahoru, sizeof
619                                     (IrSignalProjektorSipkaNahoru) / sizeof(IrSignalProjektorSipkaNahoru
620                                     [0]), FrekvenceIR_Vysilani);
621                     break;
622                 case 'S':
623                     irSend.sendRaw(IrSignalProjektorSource, sizeof(IrSignalProjektorSource) /
624                                     sizeof(IrSignalProjektorSource[0]), FrekvenceIR_Vysilani);
625                     break;
626                 case 'O':
627                     irSend.sendRaw(IrSignalProjektorOk, sizeof(IrSignalProjektorOk) / sizeof
628                                     (IrSignalProjektorOk[0]), FrekvenceIR_Vysilani);
629                     break;
630                 default;;
631             }
632         }
633     }
634     else
635     {
636         switch (Pokyn)
637         {
638             case 'D':
639                 irSend2.sendRaw(IrSignalPlatnoDolu, sizeof(IrSignalPlatnoDolu) / sizeof
```

```
(IrSignalPlatnoDolu[0]), FrekvenceIR_Vysilani);
630     break;
631     case 'N':
632         irSend2.sendRaw(IrSignalPlatnoNahoru, sizeof(IrSignalPlatnoNahoru) /
        sizeof(IrSignalPlatnoNahoru[0]), FrekvenceIR_Vysilani);
633     break;
634     case 'S':
635         irSend2.sendRaw(IrSignalPlatnoStop, sizeof(IrSignalPlatnoStop) / sizeof
        (IrSignalPlatnoStop[0]), FrekvenceIR_Vysilani);
636     break;
637     default::
638     }
639 }
640 }
641 }
642
643
644 bool WORK()
645 {
646     registrTlacitka = MozPrep[MicroSpinSET];
647     Registr();
648     delay(DobaCekaniTlacitka);
649     if (digitalRead(PinTlac) == HIGH)
650     {
651         Serial.println(F("Nastaven SET mod."));
652         return false; //Mod SET
653     }
654     else
655     {
656         return true; //Mos Work
657     }
658 }
659
660
661 bool AUTO()
662 {
663     registrTlacitka = MozPrep[MicroSpinAUTO];
664     Registr();
665     delay(DobaCekaniTlacitka);
666     if (digitalRead(PinTlac) == HIGH)
667     {
668         Serial.print(F("Vypnutá automatika platna."));
669         return false; //Vypla Automatika platna
670     }
671     else
672     {
673         return true; //Zapla Automatika platna
674     }
675 }
676
677
678 void Registr() //Zobrazeni obrazu
679 {
680     digitalWrite(PinLatch, LOW);
681     shiftOut(PinData, PinClock, MSBFIRST, registrTlacitka);
682     shiftOut(PinData, PinClock, MSBFIRST, registrZobrazeni);
683     digitalWrite(PinLatch, HIGH);
684 }
685
686
687 void Chyba(byte UsekProg)
```

```

688 {
689     switch (UsekProg)
690     {
691     case 0:
692         Serial.println(F("ERROR_0"));
693         Serial.println(F("Muze indikovat:"));
694         Serial.println(F("Zmackli jste klavesu, ke ktere neni prirazena funkce."));
695         Serial.println(F("Tlacitka maji poruchu nebo ma poruchu pridavna konzole."));
696         Serial.println(F("Chyba softweru."));
697         Serial.println(F("Jina chyba."));
698         ZakazFunkce();
699         break;
700     case 1:
701         Serial.println(F("ERROR_EEPROM_1"));
702         Serial.println(F("Muze indikovat:"));
703         Serial.println(F("Nova EEPROM."));
704         Serial.println(F("Automaticka oprava:"));
705         Serial.println(F("System nahrava na EEPROM predpripraveny defoltni zaznam o stavu ➤
            ovladace."));
706         Serial.println(F("Pokut byly na EEPROM data, budou vymazany."));
707         Blikani(200, 200, 4);
708         break;
709     case 2:
710         Serial.println(F("ERROR_EEPROM_2"));
711         Serial.println(F("Muze indikovat:"));
712         Serial.println(F("Potize s komunikaci s EEPROM."));
713         Serial.println(F("Oprava:"));
714         Serial.println(F("Prosim zkontrolujte EEPROM."));
715         Serial.println(F("Pri castem opakovani, vložte novou EEPROM do zarizeni."));
716         Serial.println(F("Softwerova oprava:"));
717         Serial.println(F("System si sytvoril imaginarni EEPROM, jejiz hodnoty jsou ➤
            defoltne nastaveny."));
718         Blikani(200, 400, 4);
719         break;
720     default:
721         break;
722     }
723     Help();
724 }
725
726
727 void Help()
728 {
729     Serial.println(F("HELP"));
730     Serial.println(F("Pozadali jste o pomoc, zadali jste spatny prikaz, nebo nestala ➤
        chyba."));
731     Serial.println(F("Prikazova tabulka je nasledujici:"));
732     Serial.println(F(" Tlacitko | Prikaz | Popis"));
733     Serial.println(F("=====|=====|=====>>"));
734     Serial.println(F("   ?   | Z,z,V,v | Vypne nebo zapne projektor. Pokud je ➤
        Automatizace zapnuta, pak se plato samo sroluje nebo roztahne."));
735     Serial.println(F("  MON   | M,m     | Obraz prepne na monitor (pocitac u ➤
        katedry)."));
736     Serial.println(F("   C-   | C,c,K,k | Obraz prepne na kameru."));
737
738     switch (Special_Tlacitko)
739     {
740     case Jako_canal:
741         Serial.println(F("   St   | S,s     | Obraz prepne na Stavebnici."));
742         break;
743     case Jako_STOP_platna:

```

```

744     Serial.println(F("    St    | S,s    | Zastavi pohyb platna."));
745     break;
746     case Jako_bez_funkce: case Jako_jine:
747         Serial.println(F("    St    | S,s    | Nema funkci."));
748         break;
749     case Jako_vypis_Help:
750         Serial.println(F("    St    | S,s    | Vipise hlasku HELP."));
751         break;
752     default:
753         break;
754 }
755
756 Serial.println(F("  -F-    | F,f    | Obraz se zamrazi/odmrazi."));
757 Serial.println(F("[ (90°otoc) | P,p    | Vysouva/zasouva platno."));
758
759 Serial.println(F("\nDalsi klavesove zkratky, které nemají tlačítka."));
760 Serial.println(F(" Prikaz | Popis"));
761 Serial.println(F("=====|=====>>"));
762 Serial.println(F("  H,h    | Vipise hlasku HELP.\n"));
763 Serial.print(F("    #    | Zmeni funkce tlačítka special. Ted ma funkci pod  ↗
    cislem:")); Serial.println(Special_Tlacitko);
764 Serial.println("0: Canal; 1: STOP platnu; 2: bez funkce; 3: vypis Help; 4: Jine.\n");
765
766 Serial.println(F("Nastavovací microspinace:"));
767 Serial.println(F("Cislo | Nazev | funkce"));
768 Serial.println(F("====|=====|=====>>"));
769 Serial.println(F("  1 | Napajeni ovladace | Napajeci ovladani ovladace.\n"));
770 Serial.println(F("  2 | Nastaveni modu | V SET poloze se vypne IR LED =>  ↗
    Pouzivejte, jen pri dezynchronizaci."));
771 Serial.println(F("  3 | PIEZO | Spusti zvukovou signalizaci."));
772 Serial.println(F("  4 | Automatizace platna | V Off poloze se zapinani a vipinani  ↗
    projektoru vysunuje zasunuje platno. V On nema zapinani a vypinani projektoru na  ↗
    platno vliv."));
773 }
774
775
776 void Blikani(int CasSviceni, int CasZhasnuti, byte Kolikrat)
777 {
778     for (byte i = 0; i < Kolikrat; i++)
779     {
780         digitalWrite(PinKont, LOW);
781         delay(CasZhasnuti);
782         digitalWrite(PinKont, HIGH);
783         delay(CasSviceni);
784     }
785 }

```