

```

1  /*
2  Name:      Ovladač_do_učebny_čísla_120.ino
3  Created:   11/15/2018 11:07:53 PM
4  Author:    Pavel
5  */
6
7
8  //Inicializace knihoven
9  #include "IRremote2.h"
10 #include <Wire.h>
11 #include <Eeprom24C04_16.h>
12
13 //inicializace IR vysilani
14 IRsend irSend;
15 IRsend irSend2;
16
17 const int FrekvenceIR_Vysilani = 36; //frekvence vysilani IR LED
18                                     //Prubeh signalu u platna
19 const unsigned int IrSignalPlatnoNahoru[] = { 1300,350, 1300,400, 1250,400,
20                                               1250,400, 450,1250, 400,1250, 400,1300, 400,1250, 450,1200, 450,1250,
21                                               1200,450, 500 }; //nahoru
22
23 const unsigned int IrSignalPlatnoDolu[] = { 1200,450, 1300,400, 1250,400,
24                                               1300,350, 450,1250, 500,1150, 450,1200, 450,1250, 450,1200, 1250,450,
25                                               400,1250, 450 }; //dolu
26
27 const unsigned int IrSignalPlatnoStop[] = { 1300,400, 1250,400, 1250,400,
28                                               1250,450, 400,1250, 400,1250, 450,1250, 450,1200, 400,1300, 400,1250,
29                                               450,1200, 1300 };
30
31 //Prubeh signalu u Projektortu
32 const unsigned int IrSignalProjektorVipinani[] = { 8950,4450, 600,500,
33                                                     550,550, 600,500, 600,550, 600,500, 600,500, 600,500, 600,550, 600,500,
34                                                     600,500, 600,500, 600,550, 600,1600, 600,1650, 550,550, 600,500, 600,550,
35                                                     600,1600, 600,550, 600,500, 600,500, 600,500, 600,550, 600,500, 600,1650,
36                                                     550,550, 600,1600, 600,1650, 600,1650, 600,1650, 550,1650, 600,1650,
37                                                     550 }; //vypinani
38
39 const unsigned int IrSignalProjektorFreeze[] = { 8950,4400, 650,500, 600,500,
40                                                  550,550, 550,550, 650,500, 550,550, 650,450, 600,500, 650,500, 550,550,
41                                                  600,500, 600,500, 600,1650, 550,1700, 600,500, 600,500, 600,1650, 650,1600,
42                                                  600,500, 550,550, 600,500, 650,500, 600,500, 600,500, 600,550, 600,500,
43                                                  600,1600, 600,1650, 650,1600, 600,1650, 600,1600, 650,1600, 600 }; //freeze
44
45 const unsigned int IrSignalProjektorSipkaNahoru[] = { 8900,4450, 600,500,
46                                                       600,550, 550,550, 550,550, 550,550, 600,550, 550,550, 550,550, 550,550,
47                                                       600,550, 550,550, 550,550, 550,1700, 550,1650, 600,550, 550,550, 550,1700,
48                                                       600,1600, 600,550, 550,1650, 600,550, 550,550, 600,500, 600,500, 600,550,
49                                                       550,550, 550,1700, 550,550, 550,1650, 600,1650, 600,1650, 550,1650, 600 }; //
50                                                       nahoru
51
52 const unsigned int IrSignalProjektorSipkaDolu[] = { 8900,4450, 600,550,
53                                                     550,550, 600,500, 550,550, 600,550, 550,550, 600,500, 600,500, 600,550,
54                                                     600,500, 600,500, 600,500, 600,1650, 550,1700, 600,500, 550,600, 550,550,
55                                                     550,550, 550,1650, 600,1650, 600,500, 600,550, 550,550, 550,600, 550,1650,
56                                                     600,1650, 550,550, 550,550, 550,1700, 600,1600, 600,1650, 550,1700, 550 }; //
57                                                     dolu
58
59 const unsigned int IrSignalProjektorSource[] = { 8950,4450, 550,600, 500,600,
60                                                  500,550, 650,550, 500,550, 550,600, 550,650, 450,550, 550,600, 550,550,
61                                                  550,550, 550,550, 600,1650, 550,1650, 600,600, 450,600, 550,600, 550,550,
62                                                  500,1750, 550,600, 500,550, 550,550, 600,600, 500,600, 650,1550, 550,1700,

```

```

...dač_do_učebny_číslo_120\Ovladač_do_učebny_číslo_120.ino 2
    500,600, 550,1700, 500,1700, 550,1700, 550,1700, 600,1650, 500 }; // ↗
    UNKNOWN CC57C043
29 const unsigned int IrSignalProjektorOk[] = { 9000,4450, 600,500, 600,500, ↗
    650,500, 600,500, 650,500, 600,500, 600,550, 600,500, 600,550, 600,500, ↗
    600,550, 600,500, 600,1650, 600,1650, 600,500, 650,500, 600,500, 650,500, ↗
    600,500, 650,500, 600,1650, 600,500, 600,550, 600,500, 600,1650, 600,1650, ↗
    600,1650, 600,1650, 600,500, 650,1600, 650,1600, 600,1650, 600 }; // NEC ↗
    C08F7
30
31
//Cislo tlacitek

32 #define TlacitkoZapVyp 0
33 #define TlacitkoMonitor 1
34 #define TlacitkoKamera 2
35 #define TlacitkoSpecial 3
36 #define TlacitkoFreeze 4
37 #define TlacitkoPlatno 5
38
39
//Systemove cislo

nastavovacich mikrospinacu
40 #define MicroSpinSET 6
41 #define MicroSpinAUTO 7
42
//Vzstup z posuvneho
registru pri zjišťování stavu tlacitek a nastavovacich mikrospinacu
43 byte MozPrep[] = ↗
    { B00000001,B00000010,B00000100,B00001000,B00010000,B00100000,B01000000,B100 ↗
        00000,B00000000 };
44
45 //Pini na ATmega328P
46 #define PinLatch 6 //RCLK (12)
47 #define PinClock 7 //SRCLK (11)
48 #define PinData 5 //Data (14)
49 #define PinIrLED 3
50 #define PinTlac 8
51 #define PinKont 2
52
53 //Moznasti specialniho tlacitka
54 #define Jako_canal 0
55 #define Jako_STOP_platna 1
56 #define Jako_bez_funkce 2
57 #define Jako_vypis_Help 3
58 #define Jako_jine 4
59 #define Jako_Max_hodnot 5
60 byte Special_Tlacitko = Jako_STOP_platna;

```

```
61
62 //Hlavní promene
63 char vykonavanyPrikaz = 'X';          //Hlavní proměnná pro prepínání kanálu: ↗
        V=Vip/Zap; M=Canal_Monitor; C=Kamera; S=Canal_Special; F=Freeze; P=Platno ↗
        nahoru/dolu;
64 bool detekceZadaniPrikazu = false;    //Detekce změny - detekce příkazu, ukončení
65 byte registrTlacitka = B00000000;    //Registr na tlačítka - pomocí toho ↗
        promene arduino zjistuje, jaké je stisknuté tlačítko
66 byte registrZobrazení = B00000000;    //Zobrazení na LED
67 bool StavProjektor = false;           //zapínání/vypnutí projektoru
68 bool StavPlatno = false;              //Vytáhne/nasune platno
69 bool StavFreeze = false;              //Zamrazí/odmrazí obraz
70 bool StavSpecial = false;
71 bool DetekceZmenyCanaluProjektoru = false;
72 char zadanyPrikaz = 0;                //proměnná, která se používá při načítání ↗
        tlačítek
73 char CanalCil;
74
75 //Pohyb při posouvání na ve Source
76 char canalNow = 0;
77 char canalOld = 0;
78 char kroky = 0;
79
80 //Vypis konektoru projektoru v Source
81 #define KonektorCanalProjektor__Pocet 5
82 #define KonektorCanalProjektor_YBp1Pr1_Pocitac1 0
83 #define KonektorCanalProjektor_YBp2Pr2_Pocitac2 1
84 #define KonektorCanalProjektor_HDMI 2
85 #define KonektorCanalProjektor_Vidio 3
86 #define KonektorCanalProjektor_S_Video 4
87
88 //Prizazeni kanálu s kterými pracujeme
89 #define Canal_Monitor KonektorCanalProjektor_YBp1Pr1_Pocitac1
90 #define Canal_Camera KonektorCanalProjektor_Vidio
91 #define Canal_Special KonektorCanalProjektor_HDMI
92
93 //Instrukce pro čtení a zápis na Eeprom
94 #define EepromNow true
95 #define EepromOld false
96 #define RadekEepromNowOld 20
97 #define RadekEepromProjektor 0
98 #define RadekEepromPlatno 1
99 #define RadekEepromFreeze 2
100 #define RadekEepromKanal 3
101 #define RadekEepromLed 4
102 #define RadekEepromSpecial 5
103 #define RadekEepromSpecialTyp 6
104 #define EEPROM_ADDRESS 0x50
105 static Eeprom24C04_16 eeprom(EEPROM_ADDRESS);
106 #define KontolaEeprom_pocet 5
107 byte KontolaEepromZadano[KontolaEeprom_pocet] = { 1,2,3,4,5 };
108 byte KontolaEepromPrecteno[KontolaEeprom_pocet];
109 #define PocetMistZapisu 7
110 byte DataNactena[PocetMistZapisu];
111 #define PocatecniMistZapisu 10
112
```

```
113 //Ovladani signalizacni LED
114 #define LedKontOn    digitalWrite(PinKont, HIGH);    //Zapne se Signalizacni  ↗
115 #define LedKontOff   digitalWrite(PinKont, LOW);     //Vipne se Signalizacni  ↗
116
117 //Seznam cekani na ...
118 #define DobaCekaniTlacitka    5 //ms
119 #define DobaCekaniNaOffProj    1000//ms
120 #define DobaBlikuZakazFunkce  100//ms
121
122 //klavesove prikazy
123 #define ZapVypKlaves0 'V'
124 #define ZapVypKlaves1 'v'
125 #define ZapVypKlaves2 'Z'
126 #define ZapVypKlaves3 'z'
127 #define PlatnoKlaves0 'P'
128 #define PlatnoKlaves1 'p'
129 #define FreezeKlaves0 'F'
130 #define FreezeKlaves1 'f'
131 #define MonitorKlaves0 'm'
132 #define MonitorKlaves1 'M'
133 #define KameraKlaves0 'C'
134 #define KameraKlaves1 'c'
135 #define KameraKlaves2 'K'
136 #define KameraKlaves3 'k'
137 #define SpecialKlaves0 'S'
138 #define SpecialKlaves1 's'
139 #define Special_SET_Klaves0 '#'
140 #define Langue_SET_Klaves0 '@'
141
142 //Zobrazeni LED
143 #define ZobVypOff    B00000001
144 #define ZobVypOn    B00000010
145 #define ZobMonitor  B00000100
146 #define ZobCamera   B00001000
147 #define ZobSpecial  B00010000
148 #define ZobFreeze   B00100000
149 #define ZobPlatNah  B01000000
150 #define ZobPlatDol  B10000000
151 #define ZobPlat     B11000000
152 #define ZobCanalSSpecial  B11100011
153 #define ZobCanalBezSpecial B11110011
154 byte ZobCanal = ZobCanalSSpecial;
155
156
157 void setup()
158 {
159     //Inicilizovani pini
160     pinMode(PinKont, OUTPUT);
161     pinMode(PinTlac, INPUT);
162     pinMode(PinLatch, OUTPUT);
163     pinMode(PinClock, OUTPUT);
164     pinMode(PinData, OUTPUT);
165     //PinIrLED neni nutne nainicializovat
166     LedKontOn
```

```
167     eeprom.initialize();
168     Serial.begin(9600);      //Inicializace seriové komunikace
169     delay(1000);
170     NacteniEeprom();
171     delay(1000);
172     LedKontOff
173 }
174
175
176 void loop()
177 {
178     zadanyPrikaz = 0;
179     detekceZadaniPrikazu = false;
180     if (Serial.available() > 0)
181     {
182         LedKontOn
183         zadanyPrikaz = Serial.read();
184         detekceZadaniPrikazu = true;
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                 LedKontOn
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         do
215         {
216             delay(100);
217             registrTlacitka = MozPrep[zadanyPrikaz];
218             Registr();
219             delay(DobaCekaniTlacitka);
220         } while (digitalRead(PinTlac) == HIGH);
221         EepromUkladani();
222         LedKontOff
```

```
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,
234                             KontolaEepromZadano);
235
236         DataNactena[RadekEepromProjektor] = false;
237         DataNactena[RadekEepromPlatno] = false;
238         DataNactena[RadekEepromKanal] = Canal_Monitor;
239         DataNactena[RadekEepromFreeze] = false;
240         DataNactena[RadekEepromLed] = registrZobrazeni;
241         DataNactena[RadekEepromSpecial] = false;
242         DataNactena[RadekEepromSpecialTyp] = Jako_canal;
243         eeprom.writeBytes(PocatecniMistZapisu, PocetMistZapisu, DataNactena);
244         Chyba(1);
245     }
246     if (NacteniEepromNowOld() == EepromOld)
247     {
248         eeprom.readBytes(PocatecniMistZapisu, PocetMistZapisu, DataNactena);
249         StavProjektor = DataNactena[RadekEepromProjektor];
250         StavPlatno = DataNactena[RadekEepromPlatno];
251         StavFreeze = DataNactena[RadekEepromFreeze];
252         canalOld = DataNactena[RadekEepromKanal];
253         StavSpecial = DataNactena[RadekEepromSpecial];
254         registrZobrazeni = DataNactena[RadekEepromLed];
255         Special_Tlacitko = DataNactena[RadekEepromSpecialTyp];
256     }
257     else
258     {
259         StavProjektor = true;
260         StavPlatno = false;
261         StavFreeze = false;
262         canalOld = Canal_Monitor;
263         Chyba(2);
264     }
265     Registr();
266     delay(100);
267     Registr();
268 }
269
270 bool NacteniEepromNowOld() //Old = 0   Now = 1
271 {
272     bool Potvrzeni = EepromOld;
273     eeprom.readBytes(RadekEepromNowOld, KontolaEeprom_pocet,
274                     KontolaEepromPrecteno);
275     for (byte i = 0; i < KontolaEeprom_pocet; i++)
276     {
277         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:case ZapVypKlaves0:case ZapVypKlaves1:case ZapVypKlaves2:case ZapVypKlaves3://Zapnutí/vypnutí
303         vykonavanyPrikaz = 'Z';//zapínání
304         if (StavProjektor == true)
305         {
306             //vypínání
307             Serial.println(F("Vypínání projektoru"));
308             StavProjektor = false;
309             registrZobrazeni = registrZobrazeni & ZobPlat;//nemenit, funkce
310             registrZobrazeni = registrZobrazeni | ZobVypOff;
311             StavFreeze = false;
312             VysilaniIR('R', 'V');
313             delay(DobaCekaniNaOffProj); //kvůli časové oddělení potvrzení
314             vipinaciho signalu
315         }
316     else
317     {
318         //zapínání
319         Serial.println(F("Zapínání projektoru"));
320         StavProjektor = true;
321         registrZobrazeni = registrZobrazeni & ~ZobVypOff;
322         registrZobrazeni = registrZobrazeni | ZobVypOn;
323         switch (canalOld)
324         {
325         case Canal_Monitor:
326             registrZobrazeni = registrZobrazeni | ZobMonitor;
327             Serial.println(F("Obraz z monitoru (pocítace)"));
328             break;
329         case Canal_Camera:
330             registrZobrazeni = registrZobrazeni | ZobCamera;
```

```
330     Serial.println(F("Obraz z kamery"));
331     canalNow = Canal_Camera;
332     break;
333     case Canal_Special:
334         if (Special_Tlacitko == Jako_canal)
335         {
336             registrZobrazeni = registrZobrazeni | ZobSpecial;
337             Serial.println(F("Obraz ze stavebnice"));
338             canalNow = Canal_Special;
339         }
340         break;
341     default:
342         break;
343     }
344 }
345 VysilaniIR('R', 'V');//vzdi to bude vyslano
346 if (AUTO() && StavProjektor != StavPlatno)
347 {
348     delay(DobaCekaniNaOffProj);
349     PlatnoOvladani();
350 }
351 break;//Platno nic neudela pokud je mod SET nebo pokud Zap a Plat se  ↗
    rovnaj...
352 case TlacitkoPlatno:case PlatnoKlaves0:case PlatnoKlaves1:
353     PlatnoOvladani();
354     break;
355 case TlacitkoFreeze:case FreezeKlaves0:case FreezeKlaves1:
356     vykonavanyPrikaz = 'F';//free
357     FreezeOvladani();
358     break;
359 case TlacitkoMonitor:case MonitorKlaves0:case MonitorKlaves1:
360     vykonavanyPrikaz = 'M';//pocitac
361     DetekceZmenyCanaluProjektoru = true;
362     registrZobrazeni = registrZobrazeni & ZobCanal;
363     registrZobrazeni = registrZobrazeni | ZobMonitor;
364     Registr();
365     Serial.println(F("Obraz z monitoru (pocitace)"));
366     canalNow = Canal_Monitor;
367     break;
368 case TlacitkoKamera:case KameraKlaves0:case KameraKlaves1:case  ↗
    KameraKlaves2:case KameraKlaves3:
369     vykonavanyPrikaz = 'C';//kamera
370     DetekceZmenyCanaluProjektoru = true;
371     registrZobrazeni = registrZobrazeni & ZobCanal;
372     registrZobrazeni = registrZobrazeni | ZobCamera;
373     Registr();
374     Serial.println(F("Obraz z kamery"));
375     canalNow = Canal_Camera;
376     break;
377 case TlacitkoSpecial:case SpecialKlaves0:case SpecialKlaves1://Bez funkce:
378     vykonavanyPrikaz = 'S';
379     switch (Special_Tlacitko)
380     {
381     case Jako_canal:
382         DetekceZmenyCanaluProjektoru = true;
383         registrZobrazeni = registrZobrazeni & ZobCanal;
```



```
384     registrZobrazeni = registrZobrazeni | ZobSpecial;
385     Registr();
386     Serial.println(F("Obraz ze stavebnice"));
387     canalNow = Canal_Special;
388     break;
389 case Jako_STOP_platna:
390     if (StavSpecial == true)
391     {
392         StavSpecial = false;
393         registrZobrazeni = registrZobrazeni & ~ZobSpecial;
394         if (StavPlatno == true)
395         {
396             StavPlatno = false;
397         }
398         else
399         {
400             StavPlatno = true;
401         }
402         PlatnoOvladani();
403     }
404     else
405     {
406         StavSpecial = true;
407         registrZobrazeni = registrZobrazeni | ZobSpecial;
408         Serial.println(F("STOP platnu."));
409         VysilaniIR('L', 'S');
410     }
411     break;
412 case Jako_bez_funkce:
413     break;
414 case Jako_vypis_Help:
415     registrZobrazeni = registrZobrazeni | ZobSpecial;
416     Registr();
417     Help();
418     registrZobrazeni = registrZobrazeni & ~ZobSpecial;
419     Registr();
420     break;
421 case Jako_jine:
422     break;
423 default:
424     break;
425 }
426 case Langue_SET_Klaves0:
427     break;
428 case Special_SET_Klaves0:
429     if (Special_Tlacitko == Jako__Max_hodnot - 1)
430     {
431         Special_Tlacitko = 0;
432     }
433     else
434     {
435         Special_Tlacitko += 1;
436     }
437     if (Special_Tlacitko == Jako_canal)
438     {
439         ZobCanal = ZobCanalSSpecial;
```

```
440     }
441     else
442     {
443         ZobCanal = ZobCanalBezSpecial;
444     }
445
446     break;
447 default:
448     Chyba(0);
449 }
450 }
451
452
453 void CanalyProjektoru()
454 {
455     if (StavProjektor == true)
456     {
457         CanalCil = vykonavanyPrikaz;
458         kroky = canalOld - canalNow;
459         if (kroky != 0)
460         {
461             if (abs(kroky) > (KonektorCanalProjektor__Pocet / 2))
462             {
463                 if (kroky < 0)
464                 {
465                     kroky += KonektorCanalProjektor__Pocet;
466                 }
467                 else
468                 {
469                     kroky -= KonektorCanalProjektor__Pocet;
470                 }
471             }
472             if (StavFreeze == true)
473             {
474                 FreezeOvladani();
475                 delay(DobaCekaniNaOffProj);
476             }
477             VysilaniIR('R', 'S');
478             if (kroky > 0)
479             {
480                 for (char i = 0; i < kroky; i++)
481                 {
482                     VysilaniIR('R', 'N');
483                     delay(DobaCekaniNaOffProj);
484                 }
485             }
486             else
487             {
488                 for (char i = 0; i < ((-1)*kroky); i++)
489                 {
490                     VysilaniIR('R', 'D');
491                     delay(DobaCekaniNaOffProj);
492                 }
493             }
494             delay(DobaCekaniNaOffProj);
495             VysilaniIR('R', 'O'); //VysilaniIR('R', 'S'); VysilaniIR('R',
```

```
        'S');
    }
    canalOld = canalNow;
}
else
{
    registrZobrazeni = registrZobrazeni & ZobCanal;
    Registr();
    ZakazFunkce();
}
DetekceZmenyCanaluProjektoru = false;
}

void PlatnoOvladani()
{
    vykonavanyPrikaz = 'P'; //platno
    if (StavPlatno == true)
    {
        //zasunovani
        StavPlatno = false;
        registrZobrazeni = registrZobrazeni & ~ZobPlatDol;
        registrZobrazeni = registrZobrazeni | ZobPlatNah;
        Serial.println(F("Platno se zasunuje"));
        VysilaniIR('L', 'N');
    }
    else
    {
        //vytahovani
        StavPlatno = true;
        registrZobrazeni = registrZobrazeni & ~ZobPlatNah;
        registrZobrazeni = registrZobrazeni | ZobPlatDol;
        Serial.println(F("Platno se vysunuje"));
        VysilaniIR('L', 'D');
    }
}

void FreezeOvladani()
{
    if (StavProjektor == true)
    {
        if (StavFreeze == true)
        {
            //odmrazit
            StavFreeze = false;
            registrZobrazeni = registrZobrazeni & ~ZobFreeze;
            Serial.println(F("Obraz je spusten"));
            VysilaniIR('R', 'F');
        }
        else
        {
            //zamrazit
            StavFreeze = true;
            registrZobrazeni = registrZobrazeni | ZobFreeze;
            Serial.println(F("Obraz je zamrazen"));
        }
    }
}
```

```
551     VysilaniIR('R', 'F');
552   }
553 }
554 else
555 {
556     ZakazFunkce();
557 }
558 }
559
560
561 void ZakazFunkce()
562 {
563     Serial.println(F("Projektor je vipnuty. Funkce mimo provoz.));
564
565     Blikani(DobaBlikuZakazFunkce, DobaBlikuZakazFunkce, 8);
566 }
567
568
569 void VysilaniIR(char Typ, char Pokyn)
570 {
571     Registr();
572     if (WORK())
573     {
574         delay(100);
575         if (Typ == 'R')//platno
576         {
577             switch (Pokyn)
578             {
579                 case 'V'://vypinani
580                     irSend.sendRaw(IrSignalProjektorVipinani, sizeof
581                                     (IrSignalProjektorVipinani) / sizeof
582                                     (IrSignalProjektorVipinani[0]), FrekvenceIR_Vysilani);
583                     break;
584                 case 'F':
585                     irSend.sendRaw(IrSignalProjektorFreeze, sizeof
586                                     (IrSignalProjektorFreeze) / sizeof(IrSignalProjektorFreeze
587                                     [0]), FrekvenceIR_Vysilani);
588                     break;
589                 case 'D':
590                     irSend.sendRaw(IrSignalProjektorSipkaDolu, sizeof
591                                     (IrSignalProjektorSipkaDolu) / sizeof
592                                     (IrSignalProjektorSipkaDolu[0]), FrekvenceIR_Vysilani);
593                     break;
594                 case 'N':
595                     irSend.sendRaw(IrSignalProjektorSipkaNahoru, sizeof
596                                     (IrSignalProjektorSipkaNahoru) / sizeof
597                                     (IrSignalProjektorSipkaNahoru[0]), FrekvenceIR_Vysilani);
598                     break;
599                 case 'S':
600                     irSend.sendRaw(IrSignalProjektorSource, sizeof
601                                     (IrSignalProjektorSource) / sizeof(IrSignalProjektorSource
602                                     [0]), FrekvenceIR_Vysilani);
603                     break;
604                 case 'O':
605                     irSend.sendRaw(IrSignalProjektorOk, sizeof
606                                     (IrSignalProjektorOk) / sizeof(IrSignalProjektorOk[0]),
```

```
        FrekvenceIR_Vysilani);
596         break;
597     default::
598     }
599 }
600 else
601 {
602     switch (Pokyn)
603     {
604     case 'D':
605         irSend2.sendRaw(IrSignalPlatnoDolu, sizeof
606             (IrSignalPlatnoDolu) / sizeof(IrSignalPlatnoDolu[0]),
607             FrekvenceIR_Vysilani);
608         break;
609     case 'N':
610         irSend2.sendRaw(IrSignalPlatnoNahoru, sizeof
611             (IrSignalPlatnoNahoru) / sizeof(IrSignalPlatnoNahoru[0]),
612             FrekvenceIR_Vysilani);
613         break;
614     case 'S':
615         irSend2.sendRaw(IrSignalPlatnoStop, sizeof
616             (IrSignalPlatnoStop) / sizeof(IrSignalPlatnoStop[0]),
617             FrekvenceIR_Vysilani);
618         break;
619     default::
620     }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 bool WORK()
638 {
639     registrTlacitka = MozPrep[MicroSpinSET];
640     Registr();
641     delay(DobaCekaniTlacitka);
642     if (digitalRead(PinTlac) == HIGH)
643     {
644         Serial.println(F("Nastaven SET mod."));
645         return false; //Mod SET
646     }
647     else
648     {
649         return true; //Mos Work
650     }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
```

```
645     return false;//Vypla Automatika platna
646 }
647 else
648 {
649     return true;//Zapla Automatika platna
650 }
651 }
652
653
654 void Registr();//Zobrazení obrazu
655 {
656     digitalWrite(PinLatch, LOW);
657     shiftOut(PinData, PinClock, MSBFIRST, registrTlacitka);
658     shiftOut(PinData, PinClock, MSBFIRST, registrZobrazení);
659     digitalWrite(PinLatch, HIGH);
660 }
661
662
663 void Chyba(byte UsekProg)
664 {
665     switch (UsekProg)
666     {
667     case 0:
668         Serial.println(F("ERROR_0"));
669         Serial.println(F("Muze indikovat:"));
670         Serial.println(F("Zmackli jste klavesu, ke které není přiřazena  ↗
        funkce."));
671         Serial.println(F("Tlačítka mají poruchu nebo má poruchu přidavná  ↗
        konzole."));
672         Serial.println(F("Chyba softwaru."));
673         Serial.println(F("Jiná chyba."));
674         ZakazFunkce();
675         break;
676     case 1:
677         Serial.println(F("ERROR_EEPROM_1"));
678         Serial.println(F("Muze indikovat:"));
679         Serial.println(F("Nová EEPROM."));
680         Serial.println(F("Automatická oprava:"));
681         Serial.println(F("System nahrává na EEPROM předpřipravený defaultní  ↗
        záznam o stavu ovladače."));
682         Serial.println(F("Pokud byly na EEPROM data, budou vymazány."));
683         Blikani(200, 200, 4);
684         break;
685     case 2:
686         Serial.println(F("ERROR_EEPROM_2"));
687         Serial.println(F("Muze indikovat:"));
688         Serial.println(F("Potíže s komunikací s EEPROM."));
689         Serial.println(F("Oprava:"));
690         Serial.println(F("Prosím zkontrolujte EEPROM."));
691         Serial.println(F("Při častém opakování, vložte novou EEPROM do  ↗
        zařízení."));
692         Serial.println(F("Softwarová oprava:"));
693         Serial.println(F("System si vytvořil imaginární EEPROM, její hodnoty  ↗
        jsou defaultně nastaveny."));
694         Blikani(200, 400, 4);
695         break;
```

```

696     default:
697         break;
698     }
699     Help();
700 }
701
702
703 void Help()
704 {
705     Serial.println(F("HELP"));
706     Serial.println(F("Pozadali jste o pomoc, zadali jste spatny prikaz, nebo  ↗
       nestala chyba."));
707     Serial.println(F("Prikazova tabulka je nasledujici:"));
708     Serial.println(F(" Tlacitko | Prikaz | Popis"));
709     Serial.println(F("=====|=====|=====>>"));
710     Serial.println(F("      ?      | Z,z,V,v | Vypne nebo zapne projektor. Pokud  ↗
       je Automatizace zapnuta, pak se plato samo sroluje nebo roztahne."));
711     Serial.println(F("    MON    |  M,m    | Obraz prepne na monitor (pocitac  ↗
       u katedry)."));
712     Serial.println(F("    C-    |  C,c,K,k | Obraz prepne na kameru."));
713
714     switch (Special_Tlacitko)
715     {
716     case Jako_canal:
717         Serial.println(F("    St    |  S,s    | Obraz prepne na  ↗
       Stavebnici."));
718         break;
719     case Jako_STOP_platna:
720         Serial.println(F("    St    |  S,s    | Zastavi pohyb platna."));
721         break;
722     case Jako_bez_funkce: case Jako_jine:
723         Serial.println(F("    St    |  S,s    | Nema funkci."));
724         break;
725     case Jako_vypis_Help:
726         Serial.println(F("    St    |  S,s    | Vipise hlasku HELP."));
727         break;
728     default:
729         break;
730     }
731
732     Serial.println(F("    -F-    |  F,f    | Obraz se zamrazi/odmrazi."));
733     Serial.println(F("[90°otoc] |  P,p    | Vysouva/zasouva platno."));
734
735     Serial.println(F("\nDalsi klavesove zkratky, ktere nemaji tlacitko."));
736     Serial.println(F(" Prikaz | Popis"));
737     Serial.println(F("=====|=====>>"));
738     Serial.println(F("    H,h    | Vipise hlasku HELP.\n"));
739     Serial.print(F("    #      | Zmeni funkce tlacitka special. Ted ma funkci  ↗
       pod cislem:")); Serial.println(Special_Tlacitko);
740     Serial.println(F("0: Canal; 1: STOP platnu; 2: bez funkce; 3: vypis Help; 4:  ↗
       Jine.\n"));
741
742     Serial.println(F("Nastavovaci microspinace:"));
743     Serial.println(F("Cislo | Nazev | funkce"));
744     Serial.println(F("=====|=====|=====>>"));
745     Serial.println(F("  1  | Napajeni ovladace | Napajeci ovladani ovladace.  ↗

```

```
    \n"));
746   Serial.println(F(" 2 |   Nastavení modu   | V SET poloze se vypne IR   ↗
      LED => Používejte, jen při dezinchronizaci."));
747   Serial.println(F(" 3 |           PIEZO           | Spustí zvukovou       ↗
      signalizaci."));
748   Serial.println(F(" 4 | Automatizace platna | V Off poloze se zapínání a ↗
      vpinání projektoru vysunuje zasunuje platno. V On nema zapínání a ↗
      vypínání projektoru na platno vliv."));
749 }
750
751
752 void Blikani(int CasSviceni, int CasZhasnuti, byte Kolikrat)
753 {
754     for (byte i = 0; i < Kolikrat; i++)
755     {
756         LedKontOff
757         delay(CasZhasnuti);
758         LedKontOn
759         delay(CasSviceni);
760     }
761 }
```