

```

1 //=====
2 // Z80 dalammcsengő hangkészletének lejátszása...
3 // Konfigurálható rendszerben.
4 // Ver.: 0.3
5 // Szerző: Tcsabi© 2017
6 // Dátum: 2017. augusztus 03. csütörtök
7 // E-mail: toth.csaba.hip@gmail.com
8 // Honlapom: http://www.storage-magic.hu
9 // Mikrovezérlő: ATTINY44A-PU http://www.atmel.com/Images/8183S.pdf
10 // A terv az, hogy a megnevezett mikrovezérlőbe beleférjen, vagy kisebb.
11 // A bemeneti vezérlőket át kell majd irányítani analóg portra.
12 //=====
13
14
15 // A dallamok jelenleg, sztring adatfolyamként vannak konfigurálva...
16 // A sztring adatfolyam meg fog szűnni. Az adatok EEPROM-ból kerülnek majd olvasásra.
17 //
18 1
19 2
20 3
21 4 5 6
22 String dallam =
23 "0CADACAAACADAF6CACAF0CADACAFB1B4B33131B1B3B4B6B4B3B1AF3131B1AFADAFB1B3B4B6B4B638B8B67
24 4733611FE985393538F5193518F53934F8C0AC08A4C8F4F93519351CFD15396569813FE65C8C5C88787C0C
25 A68CDC8CD8A8AC8CA6FF3F1CFD38D8DC0D176F6F4D3D69171FE34727152b4343476b9b614FE80d6d7d6d4d
26 3d4d3d1cf8f6fd3d18f91939436fe809853d1cecfcd18c8ccccacfce9191d1cecfcd193FE606C8ACC6F6F8FD
27 18FCE2F6E91CC2F7396D138FF";
28 //String dallam =
29 "0CADACAAACADAF6CACAF0CADACAFB1B4B33131B1B3B4B6B4B3B1AF3131B1AFADAFB1B3B4B6B4B638B8B67
30 4733611FFF";
31 //String dallam = "985393538F5193518F53934F8C0AC08A4C8F4F93519351CFD15396569813FFF";
32 //String dallam =
33 "65C8C5C88787C0CA68CDC8CD8A8AC8CA6FF3F1CFD38D8DC0D176F6F4D3D69171FFF";
34 //String dallam = "34727152b4343476b9b614FFF";
35 //String dallam =
36 "80d6d7d6d4d3d4d3d1cf8f6fd3d18f91939436fe809853d1cecfcd18c8ccccacfce9191d1cecfcd193FFF";
37 //String dallam =
38 "80cdcd8dcdcd8dcdcab1ef8d80ccccaceccdcadedcfcfcfcfd8f8a80cdcd8dcdcd8dcdcab180cdcd8dcdcd
39 d8dcdcab1ef8d80ccccaceccdcadedcfcfcfcfd8f8a80cdcd8dcdcd8dcdcab1FFF";
40 //String dallam = "80cacb8d4dcacd8f4fcdfcd2d456cacb8d4dcacd8f4fcdfcd2d456FFF";
41
42 //String dallam = "606C8ACC6F6F8FD18FCE2F6E91CC2F7396D138FF";
43
44 // Frekvencia táblázat.
45 //
46 // 0 1 2 3 4 5 6 7 8 9 10 11 12
47 // 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
48 int frekiArr[] = { 0, 988, 932, 880, 830, 784, 740, 698, 658, 622, 588, 554, 522,
49 494, 466, 440, 415, 392, 370, 349, 329, 311, 294, 277, 261, 247, 233, 220, 208, 196 };
50
51 // Hang hosszúság táblázata.
52 int hosszArr[8];
53
54 String xStr; // Általános változó.
55 int xInt; // Általános változó.
56 int dalCim = 0; // Dallam karakter pozíciója.
57 int count; // Számláló...
58 boolean eInd = 1; // A dallamindító gomb előző állapotának megjegyzésére. A
59 folyamatos gombnyomás hatására nem fogja elindítani a következő dallamot.
60
61 // Inicializálás...
62 void setup() {
63 //Serial.begin(38400);
64 pinMode(2, OUTPUT); // A kimenet, azaz itt jelenik meg a dallam.
65 pinMode(8, INPUT_PULLUP); // Dallam indítása. A folyamatos gombnyomást érzékeli
66 és leáll! --> Nyomógomb az előlapon. Analóg érzékelés lesz. Valamint
67 optocsatolóval védett bemenet. Jackdugó_1 jobb, bal +5V, föld GND-re.
68 pinMode(9, INPUT_PULLUP); // Folyamatos lejátszás 1,5-os szünettel. -->
69 Kapcsoló az előlapon.
70 pinMode(10, INPUT_PULLUP); // Aktuális dallam megjegyzése. --> Kapcsoló az
71 előlapon.
72 pinMode(11, INPUT_PULLUP); // Előző dallam lejátszása. --> Nyomógomb az

```

```

előlapon. Analóg érzékelés lesz.
52 pinMode(12, INPUT_PULLUP); // Dallamlejátszás stop. --> Nyomógomb az előlapon.
Analóg érzékelés lesz. Így felszabadul 3 port. Valamint optocsatolóval védett
bemenet, az ajtónyitás érzékelése érdekében. Jackdugó_2 bal, föld GND-re.
53 pinMode(13, INPUT_PULLUP); // Következő dallam lejátszása. --> Nyomógomb az
előlapon. Analóg érzékelés lesz.
54 //pinMode(xx, INPUT_PULLUP); // EEPROM inicializálás. Nyomógomb a NYÁK-on.
55 //pinMode(xx, INPUT_PULLUP); // Random dallamválasztás. --> Kapcsoló az előlapon.
56 //pinMode(xx, INPUT); // Analóg bemenet, a tempó beállításához! -->
Potméter, az előlapon.
57 //pinMode(xx, OUTPUT); // Üzemállapot tájékoztatás. --> LED az előlapon.
(pld.: EEPROM hiba, gombnyomás elfogadása, stb...)
58
59 // Hang hosszúság táblázata.
60 hosszArr[0] = 1000; // 16/16
61 hosszArr[1] = hosszArr[0] * 3 / 4; // 12/16
62 hosszArr[2] = hosszArr[0] * 1 / 2; // 8/16
63 hosszArr[3] = hosszArr[0] * 3 / 8; // 6/16
64 hosszArr[4] = hosszArr[0] * 1 / 4; // 4/16
65 hosszArr[5] = hosszArr[0] * 3 / 16; // 3/16
66 hosszArr[6] = hosszArr[0] * 1 / 8; // 2/16
67 hosszArr[7] = hosszArr[0] * 1 / 16; // 1/16
68
69 }
70
71 // Főprogram...
72 void loop() {
73
74     boolean start; // Dallamindító gomb értéke.
75     boolean repeat; // Ismétlés gomb értéke.
76     boolean dalStop; // Dallamlejátszás kikapcsolása.
77     boolean dalElozo; // Előző dallam keresése.
78     boolean dalKov; // Következő dallam keresése.
79     int dalHossz; // A dallam sztringjének hossza.
80     byte dalKod; // A sztring 2 karakterének decimális értéke.
81     byte freki; // Dekódolt frekvencia kódja.
82     int hossz; // Dekódolt hanghossz kódja.
83     byte xByt; // Általános bájt típusú változó,
84
85     // Z80 dallam dekódoló...
86     dalHossz = dallam.length();
87     // Serial.println(dallam.length());
88
89     // Dallamindító gomb és a dallamismétlő jumper vizsgálata...
90     do {
91         delay(200);
92         start = digitalRead(8); // Indító gomb olvasása.
93         repeat = digitalRead(9); // Ismétlés kapcsoló olvasása.
94         if (start == HIGH) {
95             eInd = HIGH; // A gombot nem nyomják.
96         }
97         if (start == eInd) {
98             start = HIGH; // Ha a gombot folyamatosan nyomják, akkor nem indítom el a
következő lejátszást.
99         }
100     } while (start == 1 && repeat == 1);
101
102     // A gombot megnyomták. A Következő gomb vizsgálati ciklusban jut szerephez.
103     eInd = LOW;
104
105     // A dallam kezdő pozíciójának betöltése a számlálóba.
106     count = dalCim;
107
108     // Karakterpárok olvasása, azaz dallam lejátszása.
109     do {
110
111         // Sztring feldarabolása...
112         xStr = dallam.substring(count, count + 2);
113         // Sztringből szám konvertálása.
114         dalKod = StrToByte(xStr);
115
116         if (dalKod < 0xFE) {
117             freki = dalKod & 0x1F;

```

```

118     hossz = dalKod & 0xE0;
119     hossz = hossz >> 5;
120     if (freki != 0) {
121         tone(2, frekiArr[freki], hosszArr[hossz]);
122     }
123     int xInt = hosszArr[hossz] * 1;
124     delay(xInt);
125 }
126 // Előző dallam gomb olvasása.
127 dalElozo = digitalRead(11);
128 // Dallam stop gomb olvasása.
129 dalStop = digitalRead(12);
130 // Következő dallam gomb olvasása.
131 dalKov = digitalRead(13);
132 // Ugrás az előző dallamra...
133 if (dalElozo == LOW && dalStop == HIGH) {
134     //Serial.println("Előző...");
135     xByt = 0;
136     do {
137         //Keresés visszafele a dallamban...
138         count = count - 2;
139         if (count < 0 ) {
140             count = dalHossz - 2;
141         }
142         xStr = dallam.substring(count, count + 2 );
143         dalKod = StrToByte(xStr);
144         /*
145             Serial.print("Cím:");
146             Serial.print(count);
147             Serial.print(" => ");
148             Serial.print(xStr);
149             Serial.print(":");
150             Serial.println(dalKod);
151         */
152         if (dalKod > 0xFD) {
153             xByt = xByt + 1;
154         }
155     } while (xByt < 2 );
156     dalStop = LOW; // Stop, hogy ki tudjon lépni a ciklusból.
157 }
158 // Ugrás az következő dallamra...
159 if (dalKov == LOW && dalStop == HIGH) {
160     //Serial.println("Következő...");
161     xByt = 0;
162     do {
163         //Keresés előre a dallamban...
164         count = count + 2;
165         xStr = dallam.substring(count, count + 2 );
166         dalKod = StrToByte(xStr);
167         /*
168             Serial.print("Cím:");
169             Serial.print(count);
170             Serial.print(" => ");
171             Serial.print(xStr);
172             Serial.print(":");
173             Serial.println(dalKod);
174         */
175         if (dalKod > 0xFD) {
176             xByt = xByt + 1;
177         }
178     } while (xByt < 1 );
179     dalStop = LOW; // Stop, hogy ki tudjon lépni a ciklusból.
180 }
181 // Következő karakterpáros pozíciója.
182 count = count + 2;
183 // Ha a dallamnak még nincs vége és nem nyomták meg a dallam stop gombot, akkor
184 // a következő hang lejátszása...
185 } while (dalKod < 0xFE && dalStop == HIGH);
186 delay(200);
187 // Kikapcsolja a dallam kimenetét.
188 noTone(2);
189 // Ha nem kell megjegyezni és nem nyomták le a stop gombot, akkor a dallamcím

```

```

190     mentése...
191     if (digitalRead(10) == HIGH && dalStop == HIGH) {
192         dalCim = count;
193     }
194
195     if (dalElozo == LOW || dalKov == LOW) {
196         dalCim = count;
197     }
198
199     // Dallamkészlet vége. Ugrás a készlet elejére.
200     if (dalKod == 0xFF) {
201         dalCim = 0;
202     }
203     /*
204     Serial.print("Cím:");
205     Serial.println(dalCim);
206     */
207     // Ha ismétlés van beállítva, a dallamok közé egy 1,5-os szünet kerül.
208     if (repeat == 0) {
209         delay(1500);
210     }
211 }
212
213 // 2 karakteres sztringből szám konvertáló függvény...
214 byte StrToByte(String kodBe) {
215     byte result;
216     String fStr;
217     fStr = kodBe.substring(0, 1);
218     if (fStr == "0") {
219         result = 0;
220     }
221     if (fStr == "1") {
222         result = 1;
223     }
224     if (fStr == "2") {
225         result = 2;
226     }
227     if (fStr == "3") {
228         result = 3;
229     }
230     if (fStr == "4") {
231         result = 4;
232     }
233     if (fStr == "5") {
234         result = 5;
235     }
236     if (fStr == "6") {
237         result = 6;
238     }
239     if (fStr == "7") {
240         result = 7;
241     }
242     if (fStr == "8") {
243         result = 8;
244     }
245     if (fStr == "9") {
246         result = 9;
247     }
248     if (fStr == "A" || fStr == "a") {
249         result = 10;
250     }
251     if (fStr == "B" || fStr == "b") {
252         result = 11;
253     }
254     if (fStr == "C" || fStr == "c") {
255         result = 12;
256     }
257     if (fStr == "D" || fStr == "d") {
258         result = 13;
259     }
260     if (fStr == "E" || fStr == "e") {
261         result = 14;
262     }
263 }

```

```
262     if (fStr == "F" || fStr == "f") {
263         result = 15;
264     }
265     result = result * 16;
266     fStr = kodBe.substring(1, 2);
267     if (fStr == "1") {
268         result = result + 1;
269     }
270     if (fStr == "2") {
271         result = result + 2;
272     }
273     if (fStr == "3") {
274         result = result + 3;
275     }
276     if (fStr == "4") {
277         result = result + 4;
278     }
279     if (fStr == "5") {
280         result = result + 5;
281     }
282     if (fStr == "6") {
283         result = result + 6;
284     }
285     if (fStr == "7") {
286         result = result + 7;
287     }
288     if (fStr == "8") {
289         result = result + 8;
290     }
291     if (fStr == "9") {
292         result = result + 9;
293     }
294     if (fStr == "A" || fStr == "a") {
295         result = result + 10;
296     }
297     if (fStr == "B" || fStr == "b") {
298         result = result + 11;
299     }
300     if (fStr == "C" || fStr == "c") {
301         result = result + 12;
302     }
303     if (fStr == "D" || fStr == "d") {
304         result = result + 13;
305     }
306     if (fStr == "E" || fStr == "e") {
307         result = result + 14;
308     }
309     if (fStr == "F" || fStr == "f") {
310         result = result + 15;
311     }
312     return result;
313 }
314
315
```