

Víkendový projekt: **Bezklíčové otevírání domovních dveří**

Cíl: Nemuset u vchodu do domu hledat klíče, ale "nechat si zabzučet" vhodným vyzváněním na svůj zvonek.

Prostředí: Domovní telefon MaT Elektra 8802 s elektrickým otvíračem dveří.

Řešení: Do domácího telefonu vestavět zařízení, které bude vyhodnocovat "vzorec" zvonění a v případě rozpoznání očekávané poslupnosti simuluje stisk tlačítka pro otevírání dveří.

Realizace: Arduino Pro Mini 5V/16MHz, reléový modul, pár pasivních součástek, 9V baterka. Výběr komponent motivován snahou o rychlý HW i SW vývoj i za cenu zjevného přezbrojení (overkillu).

Odkazy: Zapojení telefonu: http://www.matelektra.cz/servis/schemata/1131_8802_nv818.htm, Reléový modul: <http://www.hwkitchen.com/products/electronic-brick-5v-relay-module-digital-/>

Montáž do telefonu

Kupodivu se v telefonu nachází skvělé místo pro zacvaknutí 9V baterie do výlisků v plastu. Ostatní části byly umístěny na vhodná volná místa. Arduino a pasivní součástky byly naleťovány na univerzální plošný spoj. Reléový modul ponechán samostatně.

ToDo

Vzorec zvonění je zadrátován do aplikace. Lepší by bylo jej uvést například jako řetězec pro snadnější parametrizaci. Nebo dokonce zavést učící režim, kdy lze vzorec zařízení naučit.

Ladění aplikace

Seriový port je využit pro výstup ladících informací o délce pulzů a jejich vyhodnocení. Pokud tedy zařízení detekuje vyzvánění, ale neotevírá, pak je možné zjistit, co se mu nelíbí pohledem na sériovou konzoli. Jen pozor - zařízení je galvanicky spojeno se systémem domovního telefonu. Trochu jsem se bojím vyrovnávacích proudů v případě špatně uzeměného napaječe a tak jsem počítač připojoval jen když zařízení není připojeno k telefonu.

Poznámka

Tento popis byl vytvořen pomocí aplikace Notable na iPadu. Kupodivu se moje původní naděje, že kreslení fancy schématu bude rychlé a zábavé a stejně rychlé jako na papír nenaplnila. Kapacitní displeje se pro kreslení něčeho alespoň rámcově přesného nehodí.

As-is disclaimer: Toto je rámcový popis řešení, které mi týden pěkně funguje. Pokud se necháte tímto dokumentem inspirovat a něco si poškodíte, tak za mnou nechoďte.

```
1 //Hardcoded ringing pattern is .-- (short, long, long ringing)
2 #include <avr/sleep.h>
3 #include <avr/power.h>
4
5 #define MAX_PIN_COUNT 22
6 const int pinRing = 2; //Ringing detector
7 const int pinLed = 13; //Ringing indication
8 const int pinLed2 = 5; //Waken up indicator
9 const int pinRele = 6; //Door opener
10
11 //Time constants in milliseconds
12 const unsigned long minShort = 20;
13 const unsigned long maxShort = 200;
14 const unsigned long minLong = 350;
15 const unsigned long maxLong = 700;
16 const unsigned long maxPause = 1000;
17
18
19 void ledOn() {
20     digitalWrite(pinLed, HIGH);
21 }
22
23 void ledOff() {
24     digitalWrite(pinLed, LOW);
25 }
26
27 void led2On() {
28     digitalWrite(pinLed2, HIGH);
29 }
30
31 void led2Off() {
32     digitalWrite(pinLed2, LOW);
33 }
34
35 void releOn() {
36     digitalWrite(pinRele, HIGH);
37 }
38
39 void releOff() {
40     digitalWrite(pinRele, LOW);
41 }
42
43 void wakeUp(){
44 }
45
46 void delay1s() {
47     delay(1000);
48 }
```

```

49
50 void doSleep() {
51     led20ff();
52     attachInterrupt(0, wakeUp, HIGH);
53     set_sleep_mode(SLEEP_MODE_PWR_DOWN);
54     sleep_enable();
55     ADCSRA &= ~(1 << ADEN);
56     power_all_disable();
57     sleep_mode();
58     sleep_disable();
59     detachInterrupt(0);
60     power_all_enable();
61     led20n();
62 }
63
64 unsigned long measureLength() {
65     unsigned long firstTime = millis();
66     unsigned long lastTime = firstTime;
67
68     while( (millis() - lastTime) < 100) {
69         if(digitalRead(pinRing) == HIGH) {
70             lastTime = millis();
71         }
72     }
73     return lastTime - firstTime;
74 }
75
76 void waitForRing() {
77     while(digitalRead(pinRing) == LOW)
78         ;
79 }
80
81 boolean isShortRing(unsigned long howLong) {
82     if( (howLong < maxShort) && (howLong > minShort) ) return
... true;
83     return false;
84 }
85
86 boolean isLongRing(unsigned long howLong) {
87     if( (howLong < maxLong) && (howLong > minLong) ) return
... true;
88     return false;
89 }
90
91 unsigned long measurePause() {
92     unsigned long firstTime = millis();
93
94     while( (millis() - firstTime) < maxPause) {

```

```

95     if(digitalRead(pinRing) == HIGH) {
96         return millis() - firstTime;
97     }
98 }
99 return 0;
100 }
101
102 // SETUP -----
103 void setup() {
104     unsigned short int pin;
105
106     // Put all pins into output mode and low state
107     for (pin = 0; pin < MAX_PIN_COUNT; pin++) {
108         pinMode(pin,OUTPUT);
109         digitalWrite(pin,LOW);
110     }
111     pinMode(pinRing, INPUT);
112
113     // Check outputs
114     ledOn();
115     delay1s();
116     ledOff();
117     led2On();
118     delay1s();
119     led2Off();
120     releOn();
121     delay1s();
122     releOff();
123
124     Serial.begin(9600); //Serial output for debug purposes
125
126 }
127
128 // LOOP -----
129 unsigned long howLong;
130
131 void loop()
132 {
133
134     Serial.println("Try it!" );
135     delay1s();
136
137     doSleep();
138
139     ledOn();
140     howLong = measureLength();
141     Serial.println(howLong);
142     ledOff();

```

```
143
144  if( ! isShortRing(howLong) ) {
145      Serial.println("Not short!" );
146      delay1s();
147      return;
148  }
149
150  howLong = measurePause();
151  Serial.println(howLong);
152
153  if(howLong == 0) {
154      Serial.println("Pause too long!");
155      delay1s();
156      return;
157  }
158
159  ledOn();
160  howLong = measureLength();
161  Serial.println(howLong);
162  ledOff();
163
164  if( ! isLongRing(howLong) ) {
165      Serial.println("Not long!" );
166      delay1s();
167      return;
168  }
169
170  howLong = measurePause();
171  Serial.println(howLong);
172
173  if(howLong == 0) {
174      Serial.println("Pause too long!" );
175      delay1s();
176      return;
177  }
178
179  ledOn();
180  howLong = measureLength();
181  Serial.println(howLong);
182  ledOff();
183
184  if( ! isLongRing(howLong) ) {
185      Serial.println("Not long!" );
186      delay1s();
187      return;
188  }
189
190  if(howLong != 0) {
```

```
191     Serial.println("This ringing is not expected!" );
192     delay1s();
193     return;
194 }
195
196     delay1s();
197
198     releOn();
199     delay(3000);
200     releOff();
201     delay1s();
202 }
203
204
205
206
```