

<div style="text-align: justify;">Toto zapojení využívá? en hodin mikroprocesor od firmy Atmel - ATmega8. Originální realizace na Daqq.eu. Celé zapojení je poměrně jednoduché ale je třeba naprogramovat mikroprocesor. Způsob jak toho dosáhnout? dosáhnout je mnoho. Pokud nechce stavět něco složitě, jeden jednoduchý programátor je zde. My jsme si pak na kroužku postavili tento do USB. Všechny součástky jsou v obvodu a pod mikroprocesor jsme zatím dali raději patičku pro snadnější manipulaci s mikroprocesorem.
</div> <div>..</div> <div>..</div> <div style="text-align: justify;"> Toto zapojení jsme pojali jako motivaci postavit si něco praktického s využitím mikroprocesoru. Celé hodiny jsou tvořeny dvěma DPS, které jsou na sebe navzájem kolmo. V místě kontaktu desek jsou sletovými pločkami a to je zároveň přenos signálu pro displej. Ihned po zapojení na napájení by se na hodinách měl objevit čas 00:00 a již hodiny běžejí (pokud máte samozřejmě správně zapojení a naprogramovaný mikroprocesor). Aktuálně čas nastavíte pomocí tlačítek SW1 (minuty) a SW2 (hodiny). Celé hodiny je pak možné zabudovat do krabičky a lze je napájet 9V baterií nebo lze adaptérem ze síť (spotřeba 100mA). Displej jsme použili CA56-21SRWA (2), na kterém je i nakreslená plošná spoj. Mikroprocesor je ATmega8 - 16PU v pouzdru DIL28.</div> <div style="text-align: justify;">
</div> <div style="text-align: justify;"></div>
 <div>Schéma:
</div> <div></div> <div>..
</div> <div>Plošná spoj:
</div> <div></div> <div></div> <div></div> <div>..
</div> <div>PDF verze: <a target="_blank"

[sch](http://upload.ok1wmmr/clanky/hodiny/hodiny_sch.pdf), [plo](http://upload.ok1wmmr/clanky/hodiny/hodiny_dps.pdf) spoj 1:1, [osazovac](http://upload.ok1wmmr/clanky/hodiny/hodiny_osaz.pdf)

Eagle 5.7: [SCH](http://upload.ok1wmmr/clanky/hodiny/hodiny_1.0.sch) a [BRD](http://upload.ok1wmmr/clanky/hodiny/hodiny_1.0.brd)

Program: [hodiny.c](http://upload.ok1wmmr/clanky/hodiny/hodiny.c), [hodiny.hex](http://upload.ok1wmmr/clanky/hodiny/hodiny.hex) (.hex se nahrádo mikroprocesoru, můžete prohlédnout v editoru txt) vodn.zdroj

[Daqq.eu](http://daq.eu/)

Deska byla vyrobena [metodou naehlen tonneru](http://index.php/technicke-lanky/425-vyroba-plonnych-spoj)

index.php/technicke-lanky/425-vyroba-plonnych-spoj

Seznam součástek:

- R1-R8 - 330R
- R9-R11 - 10K
- C1,C2 - 22p
- C3 - 100nF
- C4, C5 - 100uF/16V
- IC1 - 7805 (TO220)
- IC4 - ATmega8-16PU (DIL28) + patice pro DIL28
- D1 - LED displej

[CA56-21SRWA](http://www.tme.eu/cz/katalog/#cleanParameters%3D1%26searchClick%3D1%26search%3DCA56-21SRWA%26bf_szukaj%3D+)

Q1 - krystal 14.7456 MHz - HC49

SW1, SW2 - mikrotlačka s delhmatnmem

Program v jazyce C

```
Untitled &lt;!-- body { color: #000000; background-color: #FFFFFF; } .cpp1-assembler { } .cpp1-brackets { } .cpp1-comment { color: #008000; font-style: italic; } .cpp1-float { color: #000080; } .cpp1-hexadecimal { color: #000080; } .cpp1-character { } .cpp1-identifier { } .cpp1-illegalchar { } .cpp1-number { color: #000080; } .cpp1-octal { color: #0000FF; } .cpp1-preprocessor { } .cpp1-reservedword { font-weight: bold; } .cpp1-space { color: #008080; } .cpp1-string { color: #800000; } .cpp1-symbol { } -->
#include <avr/io.h> #include <avr/interrupt.h> #include <avr/wdt.h> #include <stdio.h>
#define _s_A 2 #define _s_B 0 #define _s_C 6 #define _s_D 4 #define _s_E 3 #define _s_F 1 #define _s_G 7 #define _s_dot 5
const unsigned char segs[] = {
_BV(_s_A)
_BV(_s_B)
_BV(_s_C)
_BV(_s_D)
```

3 / 10

4 / 10

5 / 10

```

class="cpp1-identifier">_BV(_s_E)</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">|</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">_BV(_s_F)</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">|</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">_BV(_s_G)</span><span class="cpp1-space"> </span><span
class="cpp1-comment">>//F </span><span class="cpp1-brackets">}; </span><span
class="cpp1-space"> </span><span class="cpp1-preprocessor">#define _ms(n) (17*n)
</span><span class="cpp1-reservedword">void</span><span class="cpp1-space">
</span><span class="cpp1-identifier">wait(</span><span
class="cpp1-reservedword">unsigned</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">int</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">a)</span><span class="cpp1-space"> </span><span
class="cpp1-comment">>//basic wait </span><span class="cpp1-brackets">{</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">volatile</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">unsigned</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">int</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">b,c; </span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">for</span><span class="cpp1-brackets">(b=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;b!=</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">a;</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">b++</span><span
class="cpp1-reservedword">for</span><span class="cpp1-brackets">(c=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;c!=</span><span
class="cpp1-space"> </span><span class="cpp1-number">50</span><span
class="cpp1-symbol">;c++</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">return</span><span class="cpp1-symbol">; } </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">unsigned</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">char</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">prescale=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">; </span><span
class="cpp1-reservedword">unsigned</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">char</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">sec=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-reservedword">unsigned</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">char</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">min_1=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">; </span><span
class="cpp1-reservedword">unsigned</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">char</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">min_10=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-reservedword">unsigned</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">char</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">hour_1=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">; </span><span
class="cpp1-reservedword">unsigned</span><span class="cpp1-space"> </span><span

```

```

class="cpp1-reservedword">char</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">hour_10=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-reservedword">unsigned</span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">char</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">show_t=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">; </span><span
class="cpp1-space"> </span><span class="cpp1-identifier">ISR(TIMER1_OVF_vect) {
</span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">if</span><span class="cpp1-brackets">(++prescale</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">==</span><span
class="cpp1-space"> </span><span class="cpp1-number">225</span><span
class="cpp1-brackets">){prescale</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">=</span><span class="cpp1-space"> </span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;sec++;}; </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">if</span><span
class="cpp1-brackets">(sec</span><span class="cpp1-number">59</span><span
class="cpp1-brackets">){min_1++;sec=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">;}; </span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">if</span><span class="cpp1-brackets">(min_1></span><span
class="cpp1-number">9</span><span class="cpp1-brackets">){min_1=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;min_10++;}; </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">if</span><span
class="cpp1-brackets">(min_10></span><span class="cpp1-number">5</span><span
class="cpp1-brackets">){min_10=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">;hour_1++;}; </span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">if</span><span class="cpp1-brackets">(hour_1></span><span
class="cpp1-number">9</span><span class="cpp1-brackets">){hour_1=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;hour_10++;}; </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">if</span><span
class="cpp1-brackets">(hour_10></span><span class="cpp1-number">1</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">&&</span><span
class="cpp1-space"> </span><span class="cpp1-identifier">hour_1></span><span
class="cpp1-number">3</span><span class="cpp1-brackets">){hour_1=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;hour_10=</span><span
class="cpp1-number">0</span><span class="cpp1-symbol">;}; </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">if</span><span
class="cpp1-brackets">(++show_t==</span><span class="cpp1-number">4</span><span
class="cpp1-brackets">)</span><span class="cpp1-space"> </span><span
class="cpp1-identifier">show_t=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-space"> </span><span
class="cpp1-reservedword">switch</span><span class="cpp1-brackets">(show_t)
</span><span class="cpp1-space"> </span><span class="cpp1-brackets">{ </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">case</span><span
class="cpp1-space"> </span><span class="cpp1-number">0</span><span
class="cpp1-symbol">:</span><span class="cpp1-space"> </span><span
class="cpp1-comment">//show minutes </span><span class="cpp1-space">

```

8 / 10

```

class="cpp1-space">          </span><span class="cpp1-identifier">show_t</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">=</span><span
class="cpp1-space"> </span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-space">          </span><span
class="cpp1-reservedword">break</span><span class="cpp1-symbol">; </span><span
class="cpp1-space"> </span><span class="cpp1-brackets">}</span><span
class="cpp1-space">          </span><span class="cpp1-reservedword">return</span><span
class="cpp1-symbol">; } </span><span class="cpp1-space"> </span><span
class="cpp1-preprocessor">#define B1() (bit_is_clear(PINB,3)) #define B2()
(bit_is_clear(PINB,4)) #define B_WAIT 300 #define nop() asm volatile ("nop;")
</span><span class="cpp1-reservedword">int</span><span class="cpp1-space">
</span><span class="cpp1-identifier">main(</span><span
class="cpp1-reservedword">void</span><span class="cpp1-brackets">) { </span><span
class="cpp1-space"> </span><span class="cpp1-identifier">TIMSK</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">=</span><span
class="cpp1-space"> </span><span class="cpp1-hexadecimal">0x04</span><span
class="cpp1-symbol">; </span><span class="cpp1-space"> </span><span
class="cpp1-identifier">TCCR1B</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">=</span><span class="cpp1-space"> </span><span
class="cpp1-hexadecimal">0x01</span><span class="cpp1-symbol">; </span><span
class="cpp1-space"> </span><span class="cpp1-identifier">DDRD</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">=</span><span
class="cpp1-space"> </span><span class="cpp1-hexadecimal">0xFF</span><span
class="cpp1-symbol">; </span><span class="cpp1-space"> </span><span
class="cpp1-identifier">DDRC</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">=</span><span class="cpp1-space"> </span><span
class="cpp1-hexadecimal">0x3F</span><span class="cpp1-symbol">; </span><span
class="cpp1-space"> </span><span class="cpp1-identifier">DDRB</span><span
class="cpp1-space"> </span><span class="cpp1-symbol">=</span><span
class="cpp1-space"> </span><span class="cpp1-hexadecimal">0x00</span><span
class="cpp1-symbol">; </span><span class="cpp1-space"> </span><span
class="cpp1-identifier">PORTB</span><span class="cpp1-space"> </span><span
class="cpp1-symbol">=</span><span class="cpp1-space"> </span><span
class="cpp1-hexadecimal">0xFF</span><span class="cpp1-symbol">; </span><span
class="cpp1-space">          </span><span class="cpp1-identifier">sei(); </span><span
class="cpp1-space"> </span><span class="cpp1-reservedword">while</span><span
class="cpp1-brackets">(</span><span class="cpp1-number">1</span><span
class="cpp1-brackets">)</span><span class="cpp1-space"> </span><span
class="cpp1-brackets">{ </span><span class="cpp1-space">          </span><span
class="cpp1-reservedword">if</span><span class="cpp1-brackets">(B1()) </span><span
class="cpp1-space">          </span><span class="cpp1-brackets">{ </span><span
class="cpp1-space">          </span><span class="cpp1-identifier">wait(_ms(B_WAIT));
</span><span class="cpp1-space">          </span><span class="cpp1-identifier">min_1++;
</span><span class="cpp1-space">          </span><span
class="cpp1-identifier">sec=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-space">          </span><span

```

```

class="cpp1-brackets">} </span><span class="cpp1-space">    </span><span
class="cpp1-reservedword">if</span><span class="cpp1-brackets">(B2()) </span><span
class="cpp1-space">    </span><span class="cpp1-brackets">{ </span><span
class="cpp1-space">        </span><span class="cpp1-identifier">wait(_ms(B_WAIT));
</span><span class="cpp1-space">            </span><span class="cpp1-identifier">hour_1++;
</span><span class="cpp1-space">        </span><span
class="cpp1-identifier">sec=</span><span class="cpp1-number">0</span><span
class="cpp1-symbol">; </span><span class="cpp1-space">    </span><span
class="cpp1-brackets">} </span><span class="cpp1-space">    </span><span
class="cpp1-brackets">} } </span></span> <div> <hr /> </div> <div><span style="color:
#ffffff;">.</span></div> <div><span style="color: #ffffff;">.</span></div> <div><span
style="color: #ffffff;">.</span></div> <div><span style="color: #ffffff;">.</span></div>
<div><span style="color: #ffffff;">.</span></div> <div style="text-align: center;"><span
style="color: #3366ff;"><a target="_blank"
href="http://krouzek.radioklub.cz">http://krouzek.radioklub.cz</a></span><span style="color:
#ffffff;"><br /></span></div> <div><span style="color: #ffffff;">.</span></div> <div><span
style="color: #ffffff;">.</span></div> <div><span style="color: #ffffff;">.</span></div>
<div><span style="color: #ffffff;">.</span></div> <div><span style="color:
#ffffff;">.</span></div> <div><span style="color: #ffffff;">.</span></div> <div><span
style="color: #ffffff;">.</span></div> <div><span style="color: #ffffff;">.</span><br /> <hr />
</div> <div style="position: absolute; left: 0px; top: 20px; width: 100px; height: 100px;">New
layer...</div> <div style="position: absolute; left: 0px; top: 20px; width: 100px; height:
100px;">New layer...</div> <div style="position: absolute; left: 0px; top: 20px; width: 100px;
height: 100px;">New layer...</div>

```