

## Úvod

Před časem jsem se probíral krabicí se zásobou starých součástek. I když jsem hledal něco jiného, zastavil jsem se v okamžiku, kdy se mi v ruce objevilo několik digitronů. Kdysi (už je to hodně dávno) jsem je vydoloval z nějaké staré kalkulačky.

I zavzpomínal jsem... Před čtvrtstoletím a dříve byly takové digitrony malým pokladem. A kdo dokázal s jejich pomocí sestavit digitální hodiny, ten už patřil do kategorie "machrů".

Světlo, které digitrony vydávaly bylo nějak teplejší. Po chvíli jsem se naladil do správné nostalgické elektronkové nálady a začalo mne zajímat, zda ještě dnes ty staré lampičky fungují. To, co bylo dříve poměrně složité zařízení, by dnes bylo možné udělat poměrně jednoduše. Stačí jenom jeden mikrořadič...

Protože jsem se zrovna v té době také zajímal o to, na jaké úrovni je v současnosti programování osmibitových mikrořadičů ve vyšších jazycích, rozhodl jsem se trochu si pohrát. Pokusil jsem se sestavit jednoduché hodiny s digitrony.

# Cíl konstrukce

Hodiny by měly být šestimístné s budíkem a možností nastavení pomocí tlačítek. Při konstrukci jsem chtěl vyzkoušet několik mikrořadičů z nejběžnějších rodin nabízených a běžně dostupných pro amatéra. Při programování bude využit jazyk C.

Digitrony pro svou práci vyžadují poměrně vysoké napětí. Protože jsem se chtěl vyhnout zapojení, které by využívalo nebezpečné síťové napětí, musí být hodiny napájeny z bezpečného napětí - nejlépe 12 V.

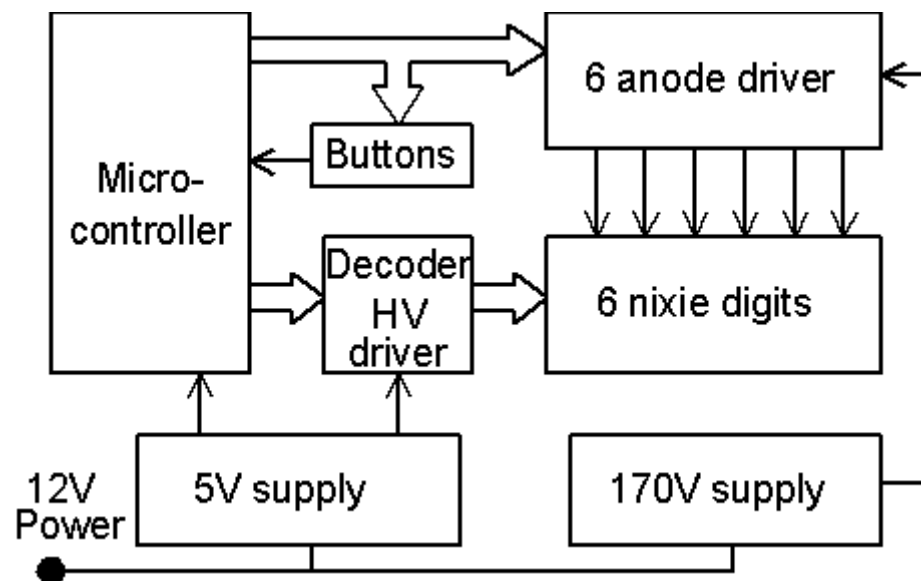
Jelikož šlo spíše o hračku, nepočítal jsem s propracovanou konstrukcí včetně krabičky a mechanického designu. Zveřejněnou konstrukci si však každý může upravit k obrazu svému.

Výsledkem je následující zapojení s těmito parametry:

<b>Zobrazení času:</b>	HH MM SS
<b>Nastavení budíku:</b>	HH MM --
<b>Režim zobrazení:</b>	24 hodin
<b>Přesnost hodin:</b>	+/- 1 sekunda/den (záleží na doladění kmitočtu krystalu)
<b>Napájecí napětí:</b>	12 V
<b>Odběr proudu:</b>	cca 100 mA

## Zapojení

Zcela přirozeně pro jednoduché hodiny s šestimístným displejem vyplynulo zapojení v multiplexním režimu.



V uvedeném blokovém schéma jsou všechny bloky poměrně jasné. Problém by mohl dělat převodník z logických výstupů na vysokonapěťové potřeby digitronu. Pro anodové spínače jsou použity vysokonapěťové tranzistory. Zapojení jsem převzal [zde](#).

Dekódování BCD signálu a vysokonapěťové spínače jednotlivých elektrod pak obstarává klasický TTL obvod 74141. Možná nebude snadné jej sehnat (nevím, zda jej ještě někdo vyrábí), ale pokud jste objevili digitrony, nemusel by tento budič být daleko :-). V dřívějších dobách se k řízení digitronů nepoužívalo téměř nic jiného, tak by se snad měl někde 1 kousek objevit.

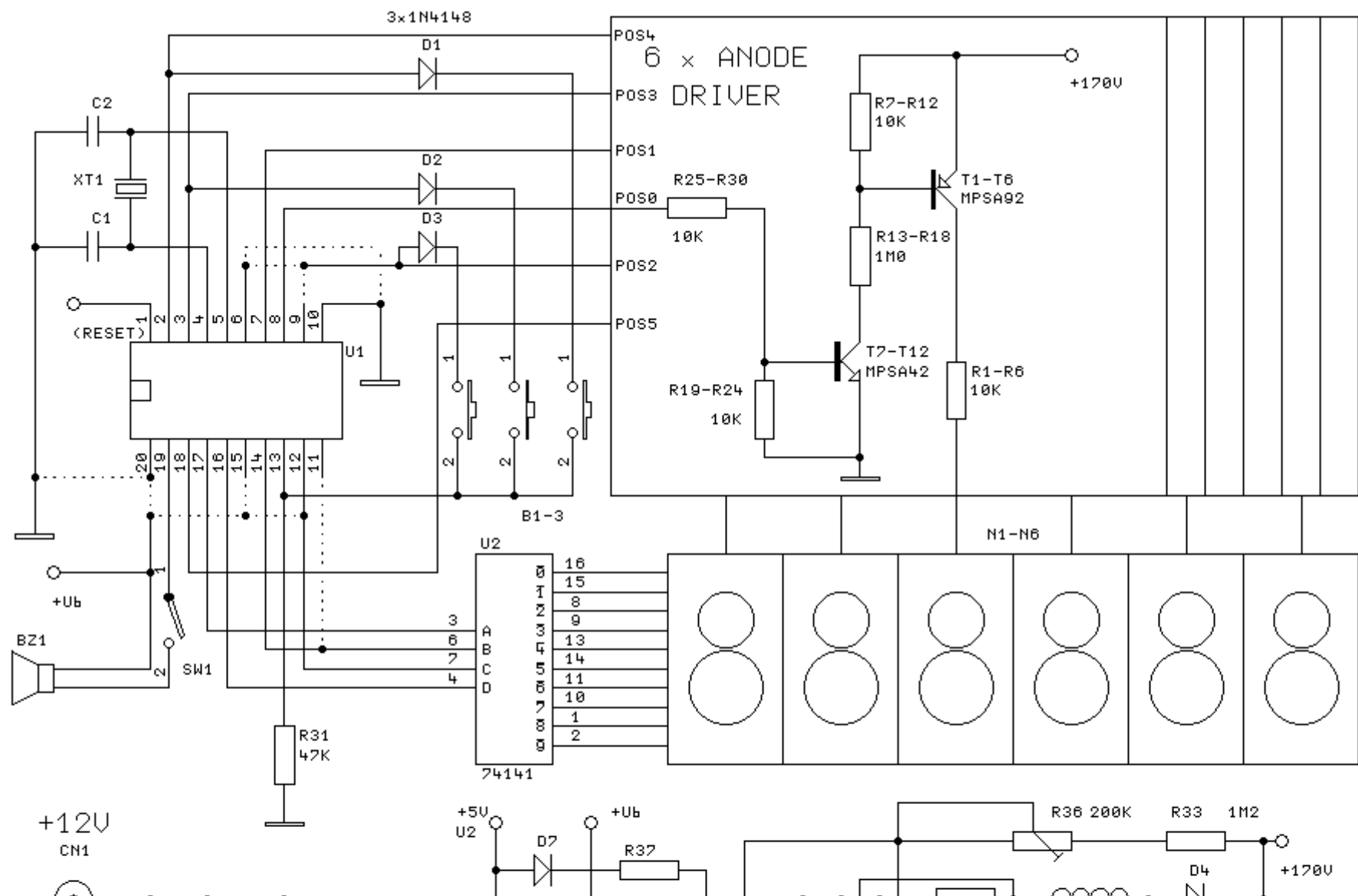
Pro napájení digitronů je obvykle potřeba napětí asi 170 V. Bylo by asi složité konstruovat speciální měničový obvod, když je dostatek integrovaných obvodů připravených pro tento účel. Použil jsem velmi levný a dostupný typ MC34063. Zapojení je téměř katalogové. Pouze výstupní tranzistor bylo nutné posílit externím spínačem T13, protože vnitřní spínač v IO není schopen pracovat s tak vysokým napětím. Jako měničová cívka se osvědčila tlumivka radiálního typu o průměru 8 mm a délce 10 mm - viz obrázek.



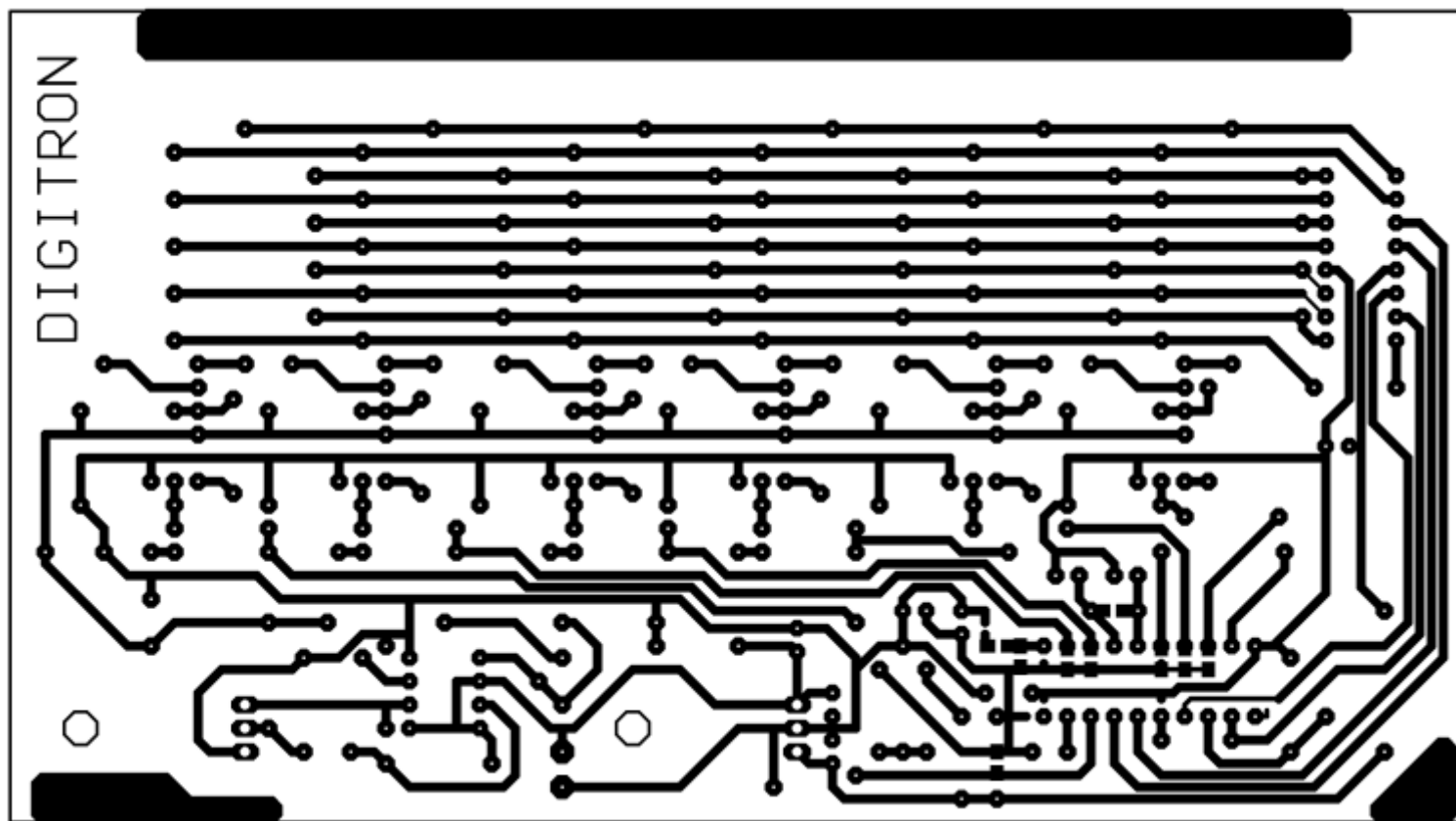
Měnič má vyhovující účinnost a měl by být i relativně bezpečný. Při pokusném zatížení 5 mA pokleslo výstupní napětí na cca 60 V. Omezení výstupního proudu určuje R32.

Napájecí napětí pro logiku je vytvořeno analogovým stabilizátorem U4. Ve schématu i na plošném spoji je připraveno zapojení pro připojení záložního akumulátoru 3,6 V. Diody D7 a D8 by měly být Schottkyho typu a R37 musí omezit dobíjecí proud podle použitého akumulátoru. Pokud si postavíte hodiny pouze pro potěchu oka, akumulátor, D7, D8 a R37 nemusíte samozřejmě osazovat.

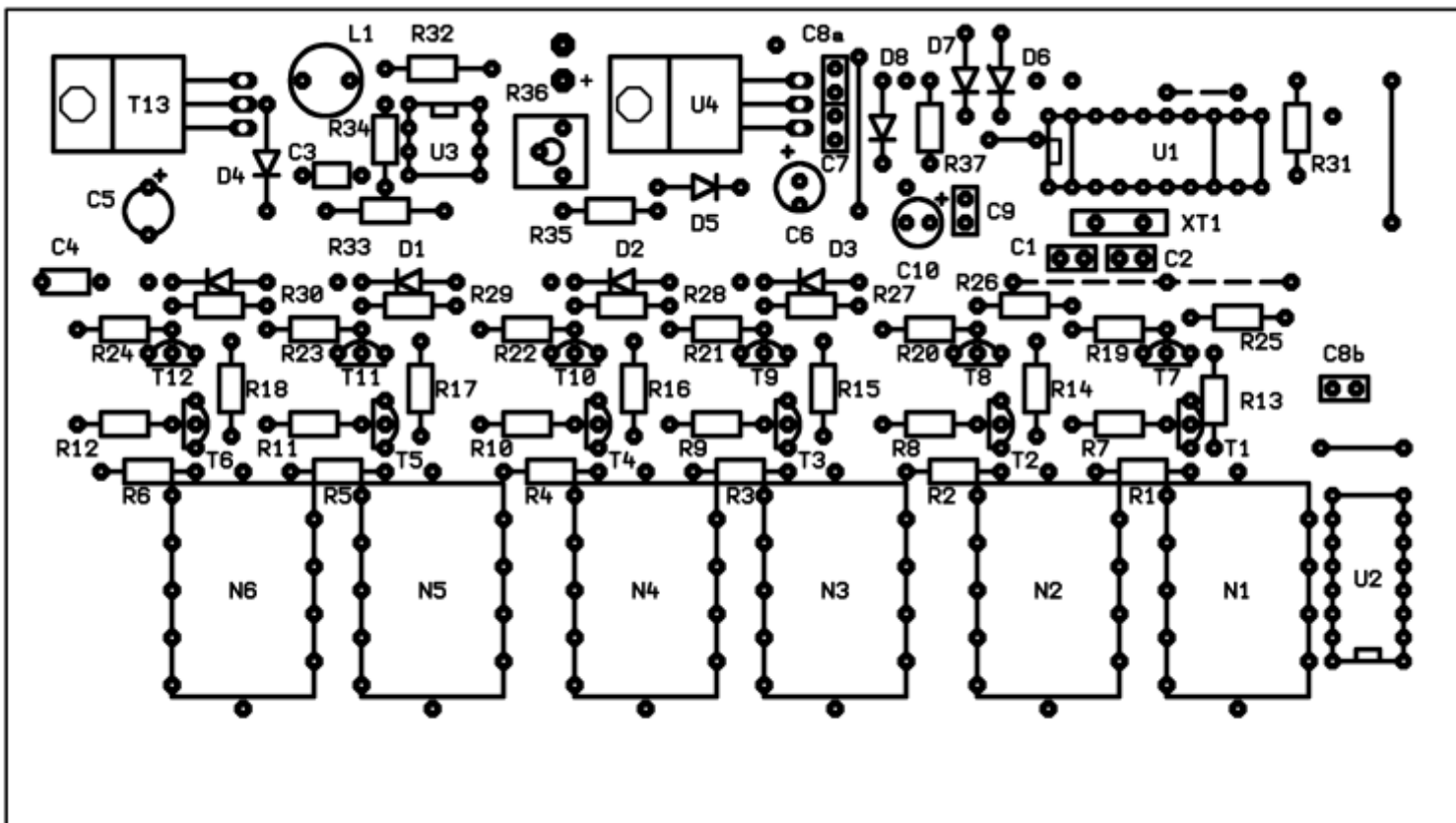
Konečnou verzí bylo zapojení podle obrázku:







... a zde rozmístění součástek.



Kliknutím na obrázek získáte předlohu pro přímý tisk na tiskárně s rozlišením 600 dpi.

## Procesor

Pro řízení těchto hodin by postačil prakticky jakýkoliv jednoduchý mikrořadič s dostatečným počtem vývodů. Pro minimální potřeby je potřeba zajistit tento počet vývodů:

Funkce	vývody
Napájení	2
Krystal	2

Buzení anod	6
Vstupy 74141	4
Vstup tlačítek	1
Siréna	1
<i>Celkem</i>	<i>16</i>

Každý z velkých výrobců mikrořadičů si zapojení vývodů pro procesory určuje sám a navzájem se mezi sebou liší. Pokusil jsem se navrhnout jednoduchý plošný spoj, u kterého by s minimálním množstvím propojek bylo možné použít různé procesory. Na zkušební desce jsem osadil 20kolíkovou patici a pouhou záměnou několika propojek jsem upravoval zapojení pro příslušný mikrořadič.

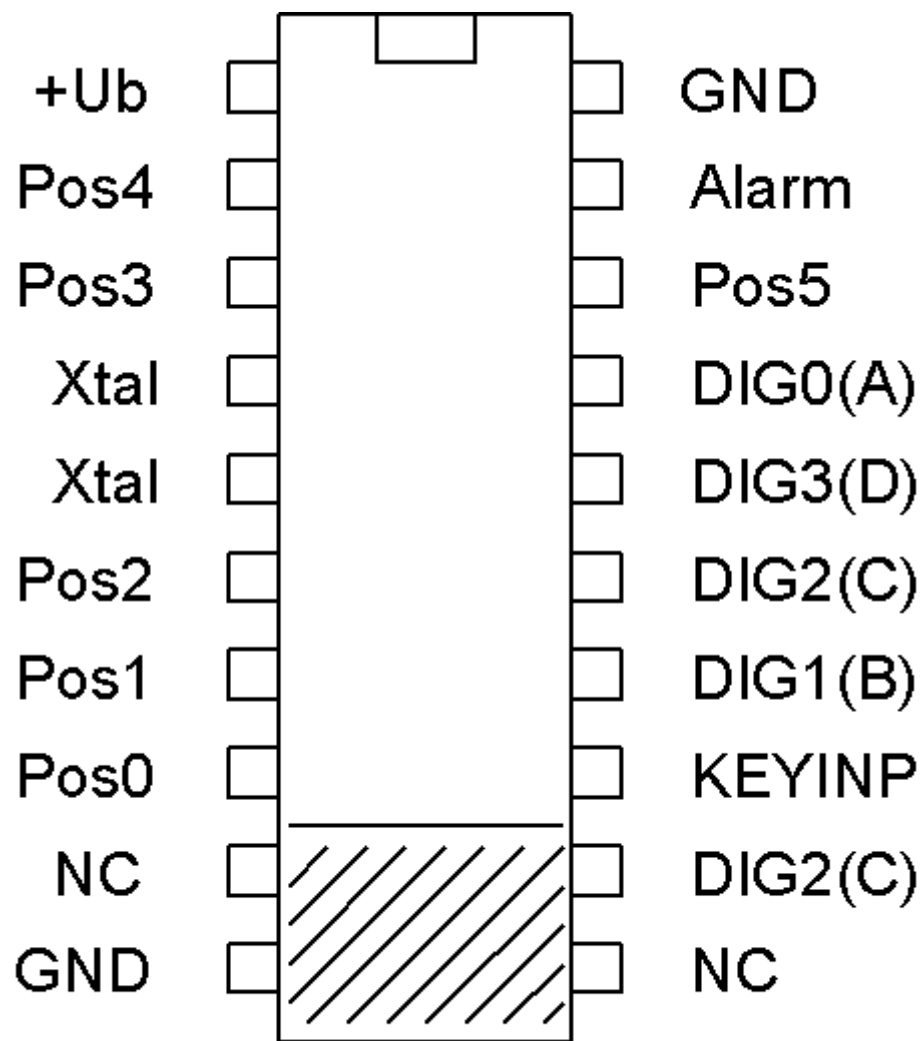
Dále jsou uvedeny typy, které jsem osobně otestoval. Vy však můžete případně vyzkoušet i jiné. Variabilita procesorů má výhodu i v tom, že amatér má obvykle k dispozici pouze jednoduchý programátor na určité typy procesorů a může pro něj být obtížné naprogramovat jiný typ. Vy si můžete vybrat to, na co jste zvyklí nebo vybaveni.

Připravil jsem vždy obrázek se zapojením procesoru, zdrojové soubory a použité nástroje pro naprogramování.

## Freescale

Použitý typ:	<b>MC68HC908QY1</b>
Krystal:	12 MHz
Kondenzátory C1,C2:	22 pF
Software:	<a href="#">freescale.zip</a>
Nastavení:	--
Programátor:	<a href="#">Janus Kit</a>
Vývojový nástroj:	<a href="#">CodeWarrior</a>
Poznámka:	ke krystalu připojen 10M SMD rezistor paralelně





MC68HC908QY1

## Microchip

Použitý typ:

PIC16F628A

Krystal:

32,768 kHz

Kondenzátory C1,C2: 22 pF

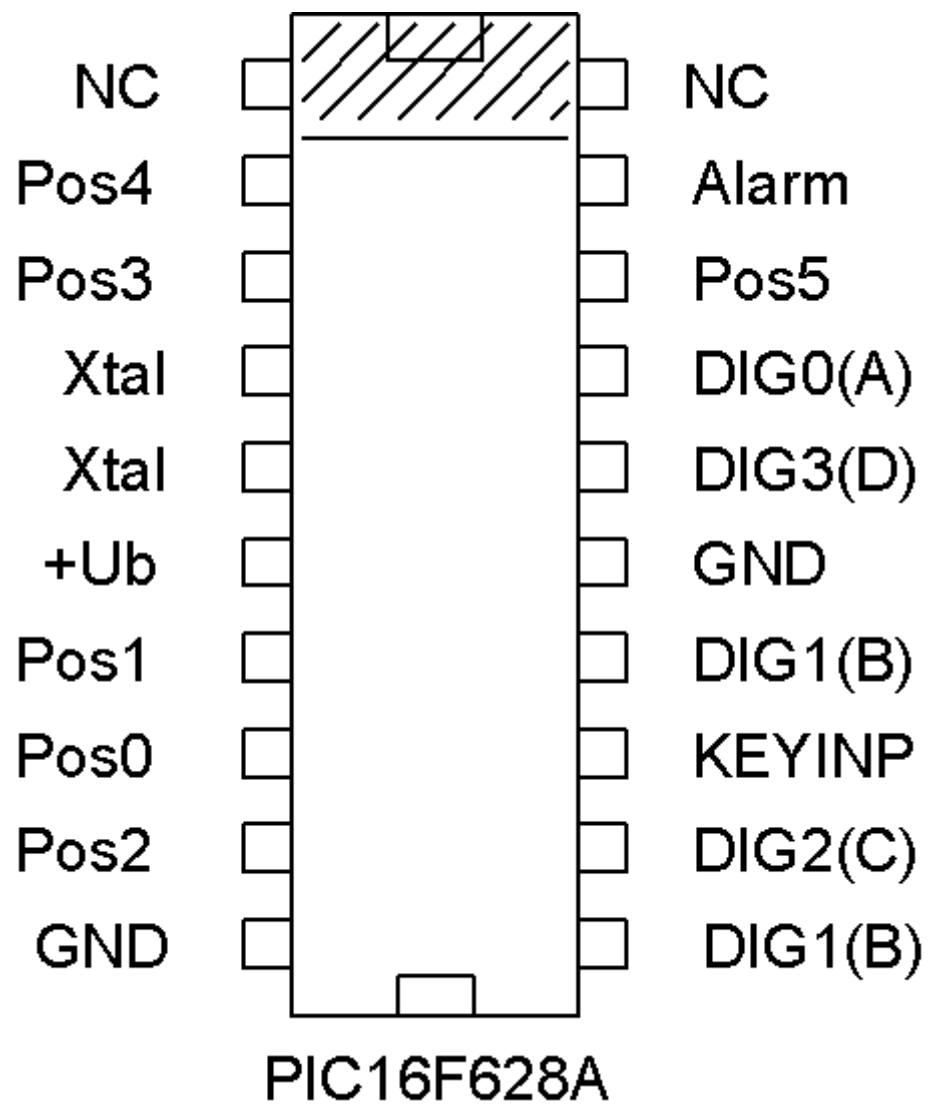
Software: [pic628.zip](#)

Nastavení: Vnitřní oscilátor 4 MHz - I/O RA6, MCLR OFF, WDT OFF, LVP OFF, BROUT OFF, CP OFF, PWRUP OFF

Programátor: Např. [zde](#)

Vývojový nástroj: [HI-TECH PICC-Lite™ Compiler](#) + [PSPad](#)

Poznámka: Pozor! Obvod je v patici obráceně



## Atmel

Použitý typ: **ATtiny2313**

Krystal: 12 MHz

Kondenzátory C1,C2: 15 pF

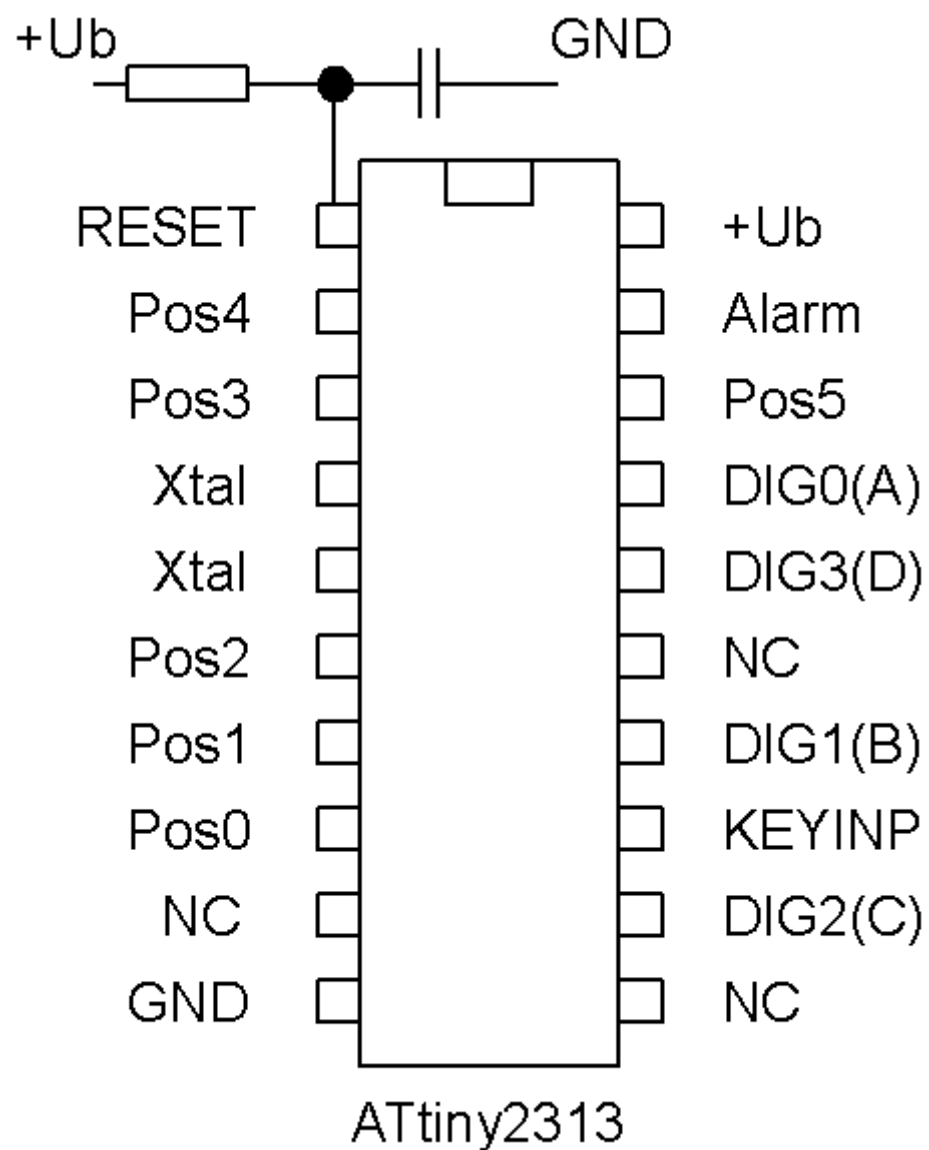
Software: [attiny.zip](#) + doplněk (viz dále)

Nastavení: Krystalový oscilátor nad 8 MHz, RESET zapnut

Programátor: [PonyProg](#), [Biprog](#)

Vývojový nástroj: [AVR Studio 4](#) + [WinAVR](#)

Poznámka: Doplnit R a C v SMD provedení na RESET (např. 10k a 100n)



## Atmel

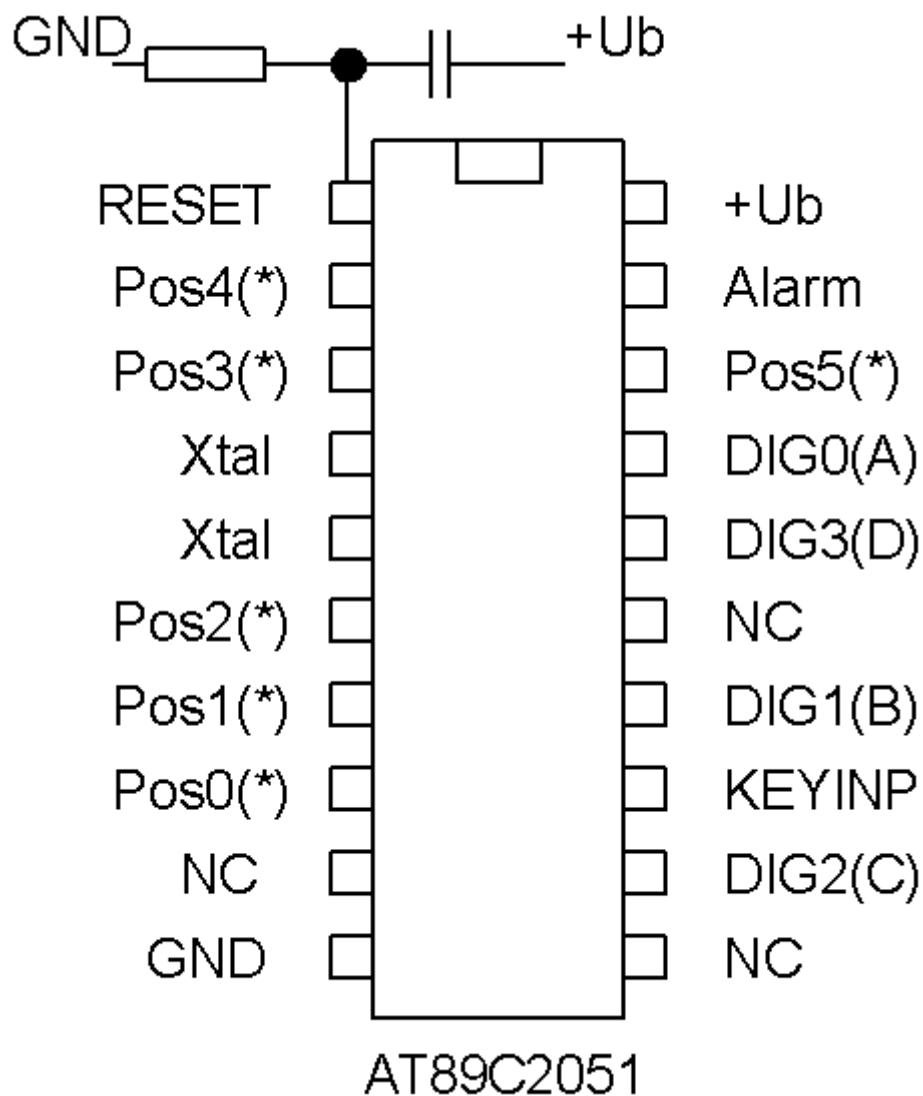
Použitý typ:

**AT89C2051**

Krystal:

12 MHz

Kondenzátory C1,C2: 22 pF  
Software: [at2051.zip](#)  
Nastavení: --  
Programátor: [PAtmel](#) nebo [zde](#)  
Vývojový nástroj: [SDCC](#) + [PSPad](#)  
Poznámka: Doplnit R a C v SMD provedení na RESET (např. 10k a 100n), hvězdičkou označené vývody připojit přes SMD rezistor 3k3 na napětí +Ub



Porovnáním zdrojového programu pro jednotlivé procesory zjistíte, že program je mezi jednotlivými platformami téměř přenositelný. Liší se pouze přístupem k portům, definicí funkce pro přerušení a ovládáním a nastavením HW části.

Zdrojový program má dvě hlavní části. Funkce *main()* nastaví porty a spustí časovač pro generování přerušení. Dále se pak v nekonečné smyčce stará o obsluhu tlačítek a případné nastavení časů. Také je zde hlídáno dosažení času alarmu.

Druhou částí je obsluha přerušení od časovače. Podle možností časovače v mikrořadiči je přerušení generováno přibližně po 1 milisekundě. V přerušení je zajištěna inkrementace času a obsluha multiplexu displeje. Při změně číslice je současně testováno i zmáčknutí tlačítka.

## Oživení

Osazování a oživování začněte od napájecí části. Po osazení obvodu kolem stabilizátoru U4 zkontrolujte, je-li na U2 napětí 5V a na procesor jde asi 4,6 V. Pak osad'te obvod měniče. Při zapojení naprázdno nastavte pomocí R36 výstupní napětí 170 V. Pokud by byl rozsah trimru malý, můžete změnit hodnotu R33 na nejbližší z řady. Osad'te U2, anodové budiče a digitrony. Po přizemnění vstupů obvodu U2 zkuste pokusně připojovat rezistory R25 až R30 na +Ub. Mělo by vždy dojít k rozsvícení číslice na odpovídající pozici. Před osazením procesoru ještě můžete překontrolovat sirénu, jestli píská.

V příslušném ZIP souboru naleznete ke každému procesoru jak zdrojový, tak přeložený program. Po naprogramování do procesoru zkontrolujte podle obrázku, zda máte zapojené odpovídající drátové propojky a cínové můstky, aby měl procesor zapojené všechny vývody správně. Pokud je U1 správně naprogramován a zapojen, již by měly hodiny pracovat a měly by jít nastavit. Pozor na zapojení tlačítek. Na desce je o 1 "náhradní" diodu pro další tlačítko navíc.

Na závěr byste měli jemnou změnou kapacit u krystalu doladit co nejpřesněji kmitočet. Hodiny pak mohou dosahovat uváděnou přesnost.

## Závěr

Jednoduché 8bitové procesory je možné již programovat ve vyšším programovacím jazyku. Jazyk C není sice primárně pro tyto procesory určen, pro jednoduché aplikace jej však lze velmi dobře použít. U složitých případů, kde jsou komplikované časově kritické vazby nebo je maximálně využíván výkon procesoru, bude asi vhodnější assembler. I s freewarovými nástroji (nebo omezenými komerčními verzemi) lze vytvořit zcela funkční program.

Při programování v C je potřeba dát především pozor na specifický přístup k jednotlivým bitům, jak se ukázalo např. u překladače pro ATtiny.

Pokud se vám vše podařilo, pak už jenom zbývá naladit se do správné elektronkově nostalgické nálady... :-)

## Úprava č.1

Uvedená konstrukce umožňuje vlastní úpravy. Toho se chopil i **Čestmír Hýbl** a program pro jednoduché hodiny vylepšil a doplnil o nové funkce. Popis úpravy programu pro ATtiny2313 najdete [zde](#). Ten, kdo používá jiný procesor, se může alespoň inspirovat a upravit si program podle svých potřeb a schopností.



## Úprava č.2

Pokud dáváte přednost zobrazení času v 12hodinovém cyklu (prostě máte sklony k anglicko-americkému počítání času) a chcete mít zobrazenou část dne "před obědem - AM" a "po obědě - PM", můžete použít úpravu, kterou připravil **Adam Jacobs**. Máte zde připravené podklady pro hodiny se čtyřmi digitrony [zde](#) nebo hodiny se šesti digitrony [zde](#) . Nebo se opět můžete alespoň inspirovat...

## Úprava č.3

**Adam Jacobs** si myslí, že nejenom digitrony mají své kouzlo. A proto upravil hodiny i pro jiný typ displeje. [Zde](#) si můžete stáhnout jeho úpravu hodin pro vakuový fluorescenční displej.