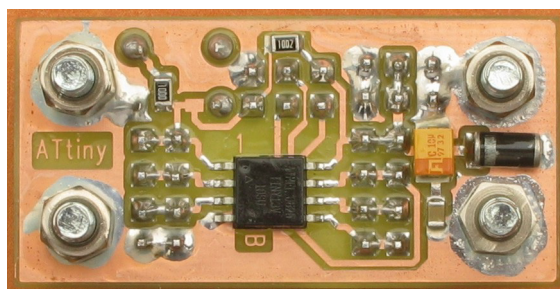
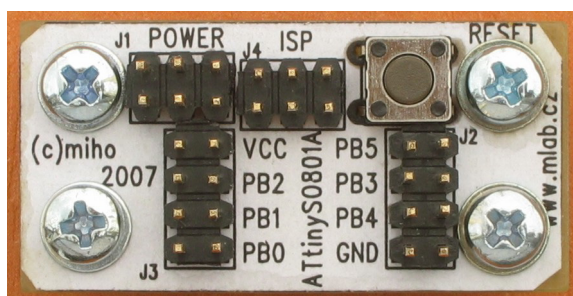


## Modul pro procesory ATMEL ATtiny v pouzdru SO8

Jakub Kákona, Milan Horkel

*Standardní modul pro nejmenší procesory ATMEL rodiny ATtiny. Obsahuje jenom procesor, připojovací hřebínky, programovací konektor a tlačítko reset.*



### 1. Technické parametry

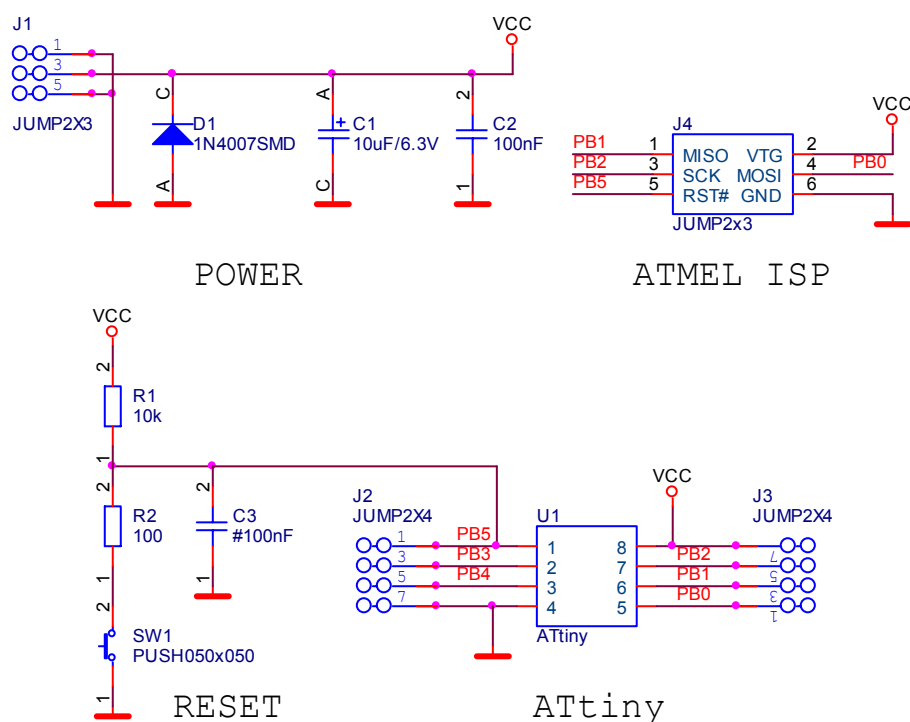
Parametr	Hodnota	Poznámka
Procesor	ATtiny v pouzdru SO8	Například ATtiny13V-10SU
Napájení	1.8V .. 5V	Dle procesoru
Spotřeba	1.2 mA / 5 V 0.4 mA / 2 V	9.6 MHz : 8, interní RC
Programování	ATMEL 6pin ISP	Podpora debugWIRE
Rozměry	40 x 20 x 15 mm	Výška nad základnou

## 2. Popis konstrukce

### 2.1. Úvodem

Standardní modul pro práci s nejmenšími procesory ATMEL ATtiny v pouzdru SO8. Konstrukce je natolik jednoduchá, že nevyžaduje další komentáře.

### 2.2. Zapojení modulu



Dioda D1 slouží jako ochrana před přepólováním napájení. Předpokládá se, že napájecí zdroj má „rozumně“ omezení proudu. Pozor proto napájení z akumulátorů, které dají obrovské proudy.

Programovací konektor je standardní ATMEL šestipinový konektor. Procesory jsou vybaveny ladicím rozhraním debugWIRE, které používá signál RESET na vývodu 1. Proto se kondenzátor C3 standardně neosazuje.

### 2.3. Zapojení použitých IO

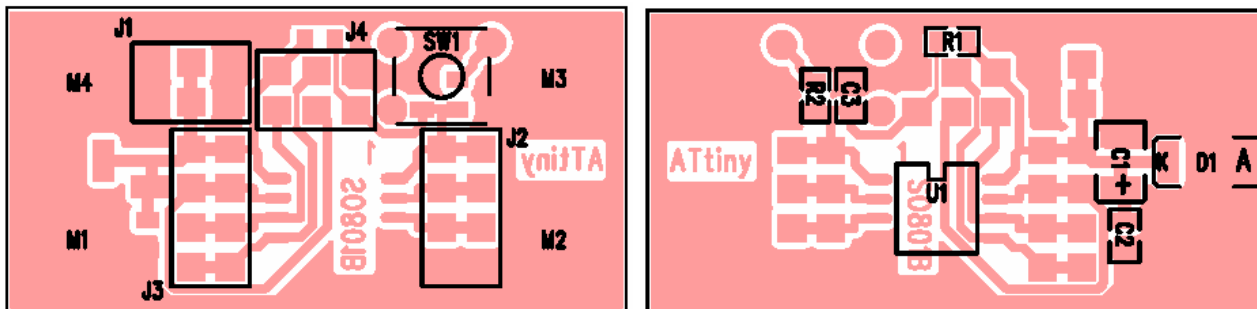
Každý vývod procesoru má obvykle mnoho různých funkcí, proto se nezapomeňte před výběrem procesoru kouknout do příslušného katalogového listu. Modul je obecný a je možné osadit jej různými procesory a postupem času se objevují další a další.

### 2.4. Mechanická konstrukce

Standardní modulek pro stavebnici MLAB s rohovými sloupky k připevnění na nosnou desku.

## 3. Osazení a oživení

### 3.1. Osazení



Reference	Hodnota
<i>Odpor</i>	
R2	100
R1	10k
<i>Keramické kondenzátory</i>	
C2	100nF
C3	#100nF (neosazuje se)
<i>Elektrolytické kondenzátory</i>	
C1	10uF/6.3V
<i>Diody</i>	
D1	1N4007SMD

Reference	Hodnota
<i>Integrované obvody</i>	
U1	ATtiny (pouzdro SO8)
<i>Mechanické součástky</i>	
SW1	P-B1720
J1, J4	JUMP2x3
J2, J3	JUMP2x4
<i>Konstrukční součástky</i>	
4ks	Šroub M3x12
4ks	Podložka M3
4ks	Sloupek M3x5

### 3.2. Oživení

Není co oživovat. Stačí nahrát Vaším oblíbeným programátorem program do procesoru a zkontrolovat funkčnost.

Procesory této řady jsou obvykle nastaveny na interní RC oscilátor s kmitočtem 9.6MHz ale pozor s předděličem 1:8.

## 4. Programové vybavení

### 4.1. Ukázkový program

Procesor bez programu je k ničemu. Základem je rozběhnout i ten nejprimitivnější program a pak už se dá vylepšovat a vylepšovat. Níže uvedený ukázkový program jen bliká LED diodou ale to je základ.

#### 4.1.1. Zapojení

Připojte LED diodu mezi PB3 a zem (nezapomeňte dát do série s diodou odpor tak asi 330Ω). Katoda diody patří na zem. Nezapomeňte připojit napájení.

### 4.1.2. Přeložení programu

Otevřete AVR Studio a založte nový projekt pro Váš procesor ATtiny. Nakopírujte, třeba myši, zdrojový text do hlavního programu a přeložte (volba Build F7). Vše by mělo proběhnout bez chyby.

### 4.1.3. Programování a ladění

Otevřete debugger a nastavte v něm druh debuggeru a cílovou součástku (volba Debug/Select Platform and Device). Pokud nemáte hardwarový debugger nastavte simulátor. Tím se nastaví typ součástky i pro programátor.

*Protože defaultní nastavení frekvence procesoru je 9.6MHz ale je nastaven interní předdělič na 1:8 je třeba u programátoru nastavit maximální komunikační rychlost na cca 250KHz (maximálně 1/4 frekvence procesoru). Tedy pokud to Váš programátor umožňuje.*

Zbývá program naprogramovat do součástky. Pokud máte hardwarový debugger (třeba ATMEL Dragon) můžete program krokovat, zasahovat do registrů procesoru a podobně.

*Ale pozor. Povolení rozhraní debugWIRE má za následek, že přestane fungovat ISP programování dokud debugWIRE nevypnete. Vypínač je totiž v konfiguračním slově procesoru a pamatuje se.*

### 4.1.4. Zdrojový text

```
// Program BLIK pro ATtiny

#define F_CPU 1200000UL // 9.6MHz je defaultni frekvence interniho
                       // RC oscilatoru a default predelic 1:8

#include <avr/io.h>
#include <util/delay.h> // Zpozdeni o libovolny pocet ms

void xDelay_ms(unsigned int Time)
{
    for(;Time!=0;Time--)
        _delay_ms(1); // Knihovni procedura ma velmi omezen
} // maximalni cas zpozdeni

#define MASK 0x08 // Port PB3 (nekoliduje s ISP)

// Hlavni program
int main()
{
    DDRB |= MASK; // Nastav port PBO jako vystup
    // Nekonecna smycka
    for(;;)
    {
        PORTB |= MASK; // Nastav 1
        xDelay_ms(500); // Pockej 1/2 sekundy
        PORTB &= ~MASK; // Nastav 0
        xDelay_ms(500); // Pockej 1/2 sekundy
    }

    return 0;
}
```