

# PRŮVODCE TEORIÍ

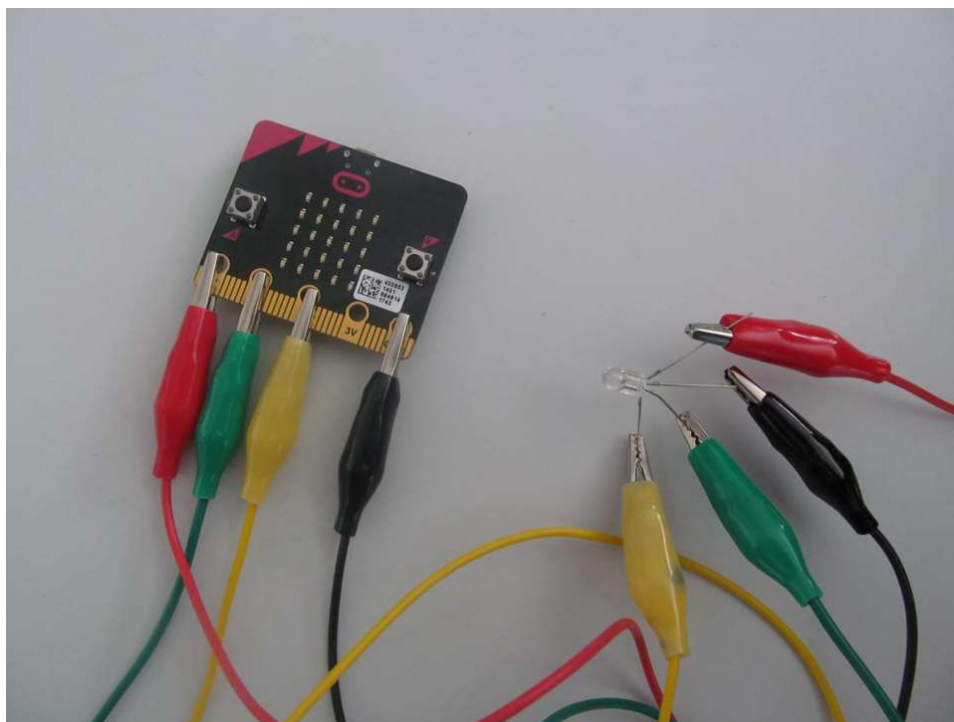
Každý uživatel Micro:bitu po nějaké době zjistí, že mu zcela nepostačují jeho funkce a vlastnosti a začne přemýšlet, jak jej dále rozšířit. To lze dokázat připojením dalších zařízení (periférií). Jako typický příklad si ukážeme připojení trojbarevné diody jako výstupního zařízení a teplotního čidla jako vstupního zařízení. Pokud pochopíte princip připojení těchto zařízení, pak lze říci, že jste schopni připojit cokoliv.

## Tříbarevná dioda

Tříbarevná dioda je taková dioda, která umí svítit modře, zeleně a červeně a kombinací těchto barev. U každé z těchto barev lze nastavit i intenzitu. Takováto dioda má obvykle čtyři vstupy (piny). Nejdelší z nich je společný a připojujeme na něj buď napětí (pak mluvíme o společné anodě) nebo zem (pak mluvíme o společné katodě). Na jedné straně pak je jeden pin (který obvykle ovládá červenou) a na druhé straně dva piny (od středu zelená a modrá).

Pro naše použití se bude hodit dioda se společnou katodou, která se mnohem snadněji zapojuje a ovládá. Dejte si pozor při nákupu, ať si koupíte tento druh diody.

**POZOR** – Jak jistě mnozí z vás zaznamenali není toto zapojení korektní. Správně bychom měli použít předřadný odpor. Nejlépe, pokud máme datasheet diody, spočítat jej pro každou „nožičku zvlášť“. Neumíme-li to nebo nemáme datasheet, poslouží dobře odpor 330 Ohmů na všech vstupních pinech. V opačném případě se takto může přehřát a zničit Micro:bit a pokud ne, tak dioda. Zejména pokud předpokládáte, že lampa bude v provozu delší dobu o této možnosti minimálně uvažovat.



Funkci a pořadí barev si nyní můžete vyzkoušet pomocí jednoduchého programu:

```
from microbit import *
pin0.write_digital(1)
sleep(2000)
pin0.write_digital(0)
pin1.write_digital(1)
sleep(2000)
pin1.write_digital(0)
pin2.write_digital(1)
sleep(2000)
pin2.write_digital(0)
```

Který můžeme významně zjednodušit a zpřehlednit takto:

```
from microbit import *
A = [pin0, pin1, pin2]
for I in range(0, 3):
    A[I].write_digital(1)
    sleep(2000)
    A[I].write_digital(0)
```

Ten postupně na dvě vteřiny rozsvítí barvy dle pinů připojených na pin0, pin1 a nakonec na pin2 micro:bitu. Takto si můžete nastavit pořadí barev dle svého přání.

V tomto příkladu se jedná o digitální dvoustavový (binární) výstup. Na výstup zapisujeme vždy jedničku nebo nulu, barva buď svítí naplno nebo nesvítí. Můžete si vyzkoušet rozsvítit také kombinaci barev nebo všechny tři barvy najednou.

Nyní si vyzkoušejte i analogový výstup, když budete postupně rozsvěcet a zhasínat jednu barvu. Ve skutečnosti se jedná o diskretizaci analogového prostoru. Máme k dispozici 1024 úrovně (0 až 1023).

```
from microbit import *
while True:
    for I in range(0, 1024):
        pin0.write_analog(I)
        sleep(2)
    sleep(1000)
    for I in range(1023, 0, -1):
        pin0.write_analog(I)
        sleep(2)
```

Následuje ukázkový program, který je nazván „Magická lampa“. Tento program diodu plynule rozsvěcí a zhasíná náhodnou barvou, přičemž při zhasínání se již rozsvěcí barva následující. Je ošetřeno, aby se barvy neopakovaly.

Tentokrát je použit analogový zápis, který na příslušný pin posílá hodnotu od 0 (nesvítí) do 1023 (svítí naplno). Tak dosáhneme postupně všech poloh mezi úplným zhasnutím a úplným rozsvícením dané barvy.

Proměnná `minula` obsahuje informaci o naposledy nastavené barvě (0 až 3). Na počátku je nastavena na např. na 2, takže začneme tím, že se zhasí barva 2 a současně rozsvěcí nově vybraná barva. Doba svitu po rozsvícení je stanovena opět náhodně. Na závěr cyklu současnou barvu

nastavíme jako minulou. Můžete příklad různě upravovat, např. hodnotu měňte po větším kroku, třeba po 10.

```
from microbit import *
import random
A = [pin0, pin1, pin2]
minula = 2
while True:
    barva = random.randint(0, 2)
    while (barva == minula):
        barva = random.randint(0, 2)
    delka = random.randint(1000, 5000)
    for I in range(0, 1024):
        A[barva].write_analog(I)
        A[minula].write_analog(1023-I)
        sleep(2)
    sleep(delka)
    minula = barva
```

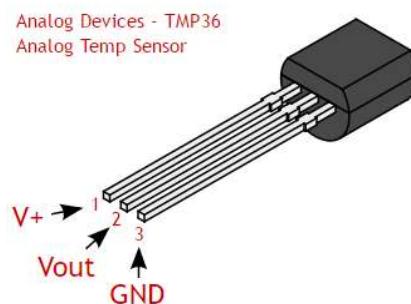
Lampu si můžete skutečně vyrobit. Lze použít papírovou konstrukci, lampion anebo nějakou lampu vyrobenou na 3D tiskárně.

**Pozor:** Pokud chcete tuto lampu nechat svítit delší dobu je nutné zapojit předřadný odpor (dle charakteristik diody), aby nedošlo k poškození diody. Napětí na pinech při plném zatížení je přibližně 2,8 V.

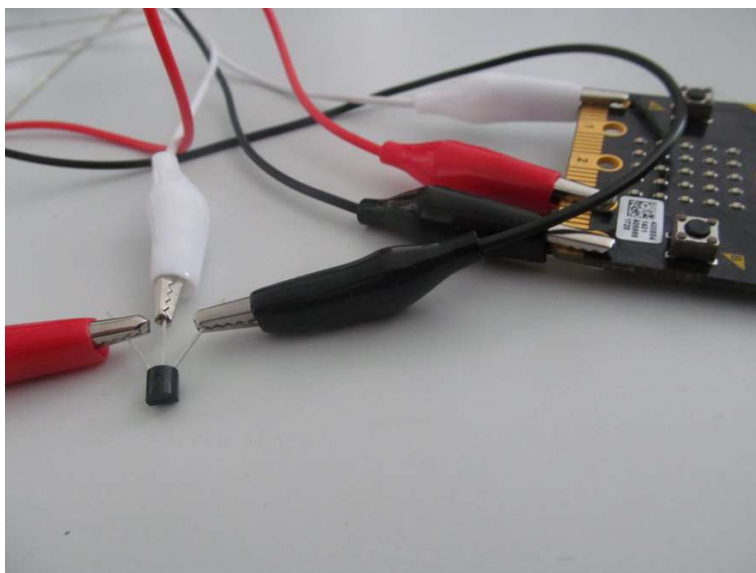
## Teplotní čidlo

Pro tuto kapitolu potřebujete nějaké levné teplotní čidlo, pracující s napětím 3 V. Autoři použili čidlo TMP-36, ale je samozřejmě možné použít libovolné které je k dispozici. K použitému čidlu, potřebujete dokumentaci (zvanou datasheet), kterou buď dostanete spolu s čidlem nebo si jí stáhnete z webu prodejce nebo odjinud z internetu.

První věc, kterou si musíte zjistit je zapojení čidla. Například čidlo TMP-36 se zapojuje dle následujícího schématu:



Zde V+ je napájení, připojte na něj 3V, GND (zem) připojte na GND a Vout je výstup, který zapojte na libovolný pin, například na pin nula:



Všimněte si na fotografii, že plochá hrana čidla je dole. Dejte si pozor abyste nespětli zapojení napájení a země, mohli byste snadno teplotní čidlo zničit (autoři s tím již mají zkušenosti).

Čidlo po připojení napájení a země začne měřit teplotu a výsledek sděluje úrovní napětí na výstupním pinu (Vout). Je zde napětí od 0 do 1023 mV. Toto napětí ukazuje procento ze vstupního napětí, které je u micro:bitu 3.18 V.

Proto je výpočet napětí:

$$napeti = \frac{V_{out} \cdot 3.18}{1024}$$

Odtud pak již vypočtete teplotu (ve stupních celsia):

$$teplota = \frac{napeti - 500}{10}$$

Vzorec je převzatý z datasheetu (manuálu) k čidlu TMP 36 a může se lišit, pokud máte jiné čidlo než popisované TMP 36. V takovém případě si naleznete potřebné vzorce v odpovídajícím datasheetu.

Pokud vše přepíšete do programu, dostanete následující kód:

```
from microbit import *
while True:
    hodnota = pin0.read_analog()
    napeti = hodnota * (3170 / 1024)
    teplota = (napeti - 500) / 10
    display.scroll(round(teplota, 1))
    sleep(10000)
```

Jak je vidět mezi jednotlivými měřeními je pauza 10 sekund. Tu si samozřejmě můžete upravit, dle vlastního přání.

Počítejte s tím, že po zapojení chvíli trvá než se teplotní čidlo srovná na teplotu měřeného okolí. Zejména pokud jste jej před tím drželi delší dobu v ruce. První dva až tři výsledky doporučujeme ignorovat. Všimněte si, jak se teplota postupně ustaluje na určité hodnotě.

Ověřte si výsledek vůči nějakému teploměru, kterému důvěřujete. Pokud se hodnoty významně liší, pokuste se zjistit, kde je problém v tomto pořadí:

- ověřte program, zejména výpočty
- ověřte zapojení
- pokud se liší u všech žáků od spolehlivého teploměru pak, ověřte zda máte správný datasheet a používáte správný vzorec pro výpočet
- ověřte voltmetrem napětí

Pro úplnost je třeba dodat, že samotný micro:bit obsahuje teplotní čidlo, které však není zcela spolehlivé, kvůli tomu, že jej ovlivňuje teplota procesoru. Teplota může být při delším měření ovlivněna teplotou procesoru.

Program, který využívá tohoto čidla je velmi jednoduchý:

```
from microbit import *
while True:
    teplota = temperature()
    display.scroll(teplota)
    sleep(10000)
```

Máte-li více micro:bitů můžete si srovnat změřené teploty oběma čidly.