

Přípravy na kroužek Mechatronika – KA6(A2d)

4. lekce – obvod s rezistorem, napětí v obvodu, spínače, pull-up rezistory, sestavování robota

Vyhodnocování stavu spínačů pomocí mikrokontrolérů AVR

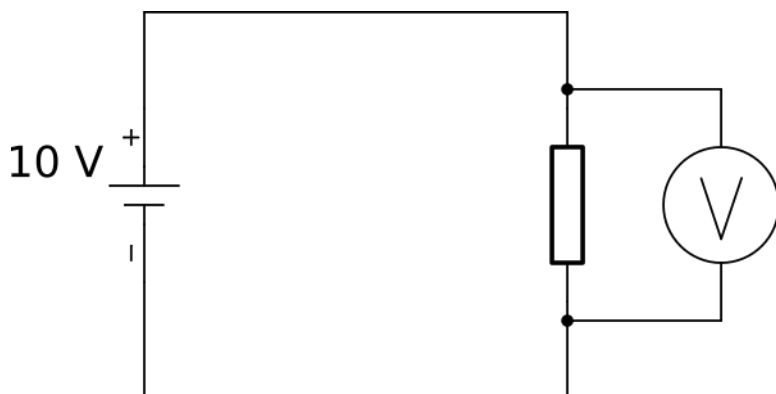
Spínače a sensory můžou dle konstrukce a konkrétního použití signalizovat mikrokontroléru (MCU) svůj stav sepnutí (uzavření obvodu):

1. **Posláním signálů** (napětí) na příslušný pin MCU.
2. **Spojením příslušného pinu MCU**, který je použitý v obvodu spínače, s nulovým potenciálem (se zemí).

Následně pomocí MCU vyhodnocujeme stav na příslušném **PINu** MCU. V prvním případě jde změna stavu **PINu** vyhodnotit bez problémů, jelikož se stav z 0 (**nulové napětí**) změní sepnutím spínače na stav 1 (**přítomné napětí na PINu MCU**).

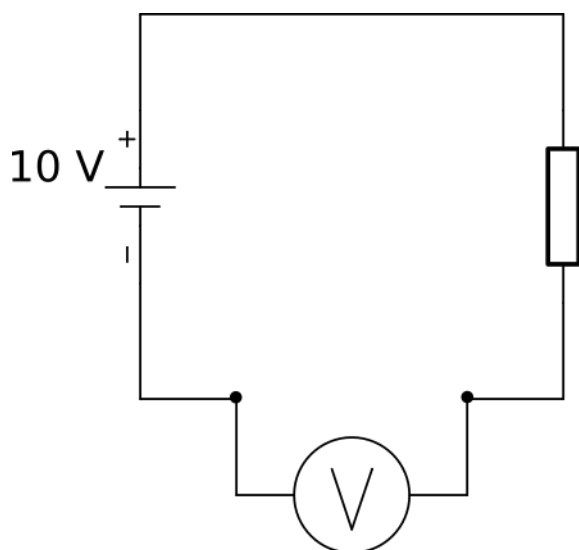
Ve druhém případě musíme použít speciální techniku tzv. **PULL-UP rezistoru**, kdy naopak ve stavu rozepnutí spínače je na příslušném **PINu** MCU napětí přítomno a ve stavu sepnutí spínače je rovno 0 V (**ve skutečnosti se pouze blíží 0 V**). Tato technika se používá v případě různých klávesnic, ovládacích tlačítek apod.

Pro lepší pochopení principu funkce **PULL-UP rezistoru** si popíšeme jednoduchý obvod se zapojeným zdrojem napětí a rezistorem.

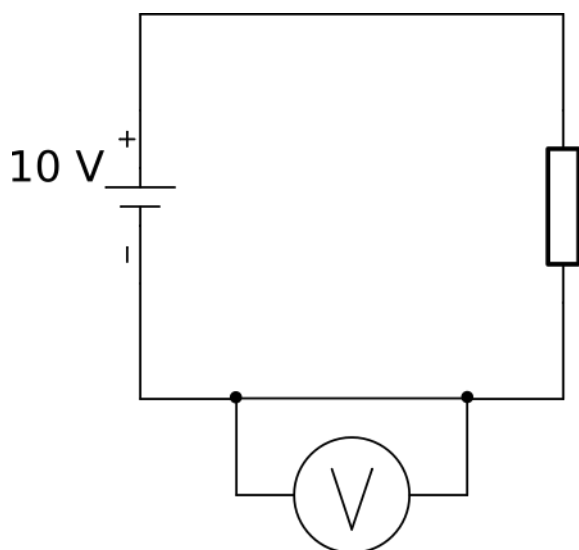


V tomto obvodu naměříme na svorkách rezistoru **napětí 10 V**, stejně jako na samotném zdroji.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



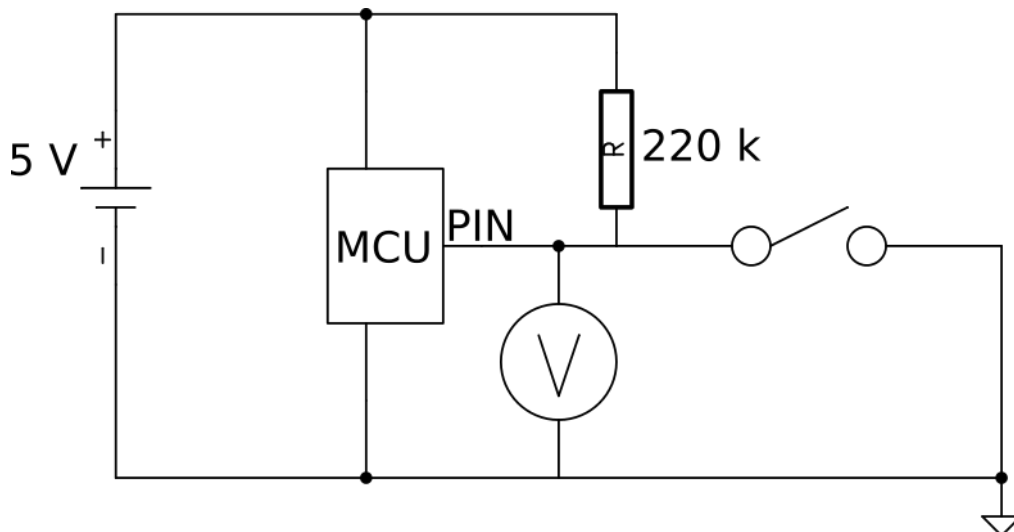
Pokud obvod v nějakém místě přerušíme a připojíme zde voltmetr, naměřené napětí bude opět odpovídat napětí na zdroji, tj. **10 V**.



Zůstane-li obvod v místě měření propojený (v tomto případě **představuje sepnutí spínače**), naměříme mezi oběma body napětí **0 V**.

Na podobném principu pracují **PULL-UP rezistory** v případě MCU.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Na **PIN** mikrokontroléru je přivedeno napětí přes připojený **PULL-UP rezistor**, na příslušném **PINu** je tedy přítomno napětí a MCU vyhodnocuje hodnotu na tomto **PINu** jako **log 1** – odtud pochází název **PULL-UP** - v překladu **vytáhnout nahoru** (napětí). Stejnou hodnotu by nám vykázal i připojený voltmetr.

Při propojení kontaktů spínače, **jsou PIN mikrokontroléru a záporný pól baterie na stejném potenciálu (napětí = rozdíl potenciálů)**, voltmetr ukáže hodnotu přibližně **0 V** a stejně to vyhodnotí i samotný MCU – **log 0**. Vzhledem k tomu, že **PULL-UP rezistor** má dostatečně vysokou hodnotu elektrického odporu (220 kΩ), bude protékající proud minimální.

Mikrokontroléry AVR mají **PULL-UP rezistory** již vestavěné, stačí je pouze aktivovat v našem programu. Pro aktivaci nastavíme:

- příslušný kontakt MCU na **vstup** (jako **PIN – DDR registr** bude mít příslušný bit nastavený na **log 0**)
- **registr PORT** na příslušném kontaktu na **log 1**

Příklad:

```
DDRB=0b00000000
PORTB=0b00000001
```

Tímto nastavíme **PINB0** na vstup a aktivujeme pro něj **PULL-UP rezistor** pomocí **PORTB0**. Pro lepší názornost je uvedeno v binární soustavě.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Program rozsvícení LED pomocí MCU a spínače

Na základě předchozích informací použijeme řídicí desku ze stavebnice Merkur k rozsvícení LED po stisku tlačítka. Tlačítko bude na řídicí desce simulováno **modrým 4řadým DIP přepínačem**, kde každá řada jde nastavit do polohy **ON** nebo **OFF**. V poloze **ON** se spojí kontakt na MCU, se kterým je příslušný přepínač propojen, se zemí (záporným pólem baterie).

Založíme nový projekt v Atmel studiu a do zdrojového souboru vložíme kód:

```
#include <avr/io.h>

int main(void)
{
    DDRC|=(1<<PORTC0);
    PORTB|=(1<<PORTB3);
    while(1)
    {
        if (bit_is_clear(PINB,PINB3)) {
            PORTC|=(1<<PC0);
        } else {
            PORTC&=~(1<<PC0);
        }
    }
    return 0;
}
```

Nastavíme **DDRC0** na **log 1** – zde je výstup, kam připojíme LED.

Registr **POTRB3** použijeme k nastavení **PULL-UP** rezistoru, na tento vývod MCU je připojený spínač řídicí desky ze 4. řady DIP.

Pro vyhodnocení stisku tlačítka použijeme funkci **bit_is_clear**. Ta vrátí v podmínce **if** hodnotu **TRUE** v případě, že je tlačítko stisknuté – logická hodnota na příslušném **PINu** je **0** a provede rozsvícení LED pomocí registru **PORTC0**.

Opakem funkce, jenž by šlo využít, je **bit_is_set**. Princip je stejný jako u předchozí funkce ale inverzní, tj. hlídá, kdy je na příslušném kontaktu MCU logická hodnota 1.

Ve zbylé části lekce budeme pokračovat v sestavování robota Merkur dle návodu z lekce 3.

Vypracoval Radek Zvěřina.