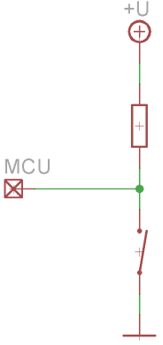
Připojování periferií k MCU

# Periferie

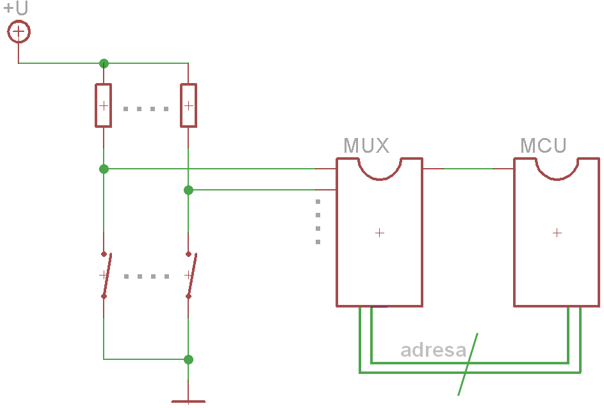
* Vstupní – tlačítko, klávesnice
* Výstupní – LED, displej
* Při připojování je nutné dbát na elektrické vlastnosti (maximální proud 40 mA, dlouhodobé zatížení 20 mA)
* Pull-up/down odpor v obvodu definuje logickou hodnotu, pokud není definována jiným zařízením

# Tlačítka

## Základní zapojení

* Tlačítko vede do MCU, jeden konec na zem, druhý přes pull-up odpor na kladné napětí
* Napětí definuje hodnotu v případě, že tlačítko není stisknuté
* Pokud by se v zapojení nenacházel rezistor, stisknutí tlačítka by zkratovalo obvod (velikost R je obvykle 5/10 KΩ)
* Výhody:
  + Jednoduché zapojení
  + Snadná detekce stisknutí tlačítka (přečtení hodnoty na pinu)
  + Snadné rozlišení stisknutí více tlačítek
* Nevýhody:
  + Více tlačítek zabere hodně pinů

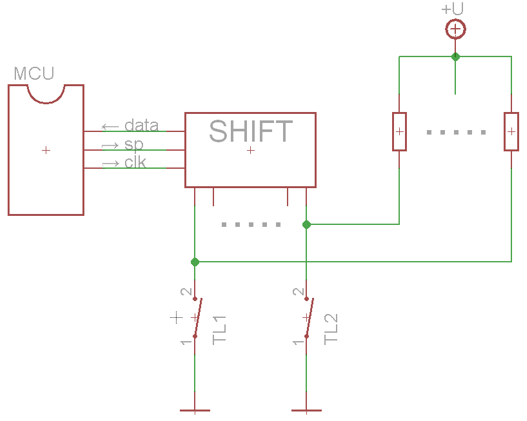
## Zapojení s multiplexorem

* Počet tlačítek je 2^počet adresních bitů
* Výhody:
  + Malá obsazenost pinů (1 + počet adresních bitů)
* Nevýhody:
  + Externí součástka (multiplexor)
  + Počet tlačítek omezen počtem vstupů na MUX
  + Nelze číst více tlačítek najednou

## Zapojení s prioritním dekodérem

* Adresa stisknutého tlačítka má největší prioritu
* Počet tlačítek = 2 ^Počet adresních bitů
* Výstupem dekodéru je "adresa" (pořadí) tlačítka
* Výhody:
  + Menší obsazenost I/O pinů
  + Možnost připojit velké množství tlačítek
* Nevýhody:
  + Externí součástka
  + Nelze detekovat více stisknutých tlačítek najednou

## Zapojení s posuvným registrem

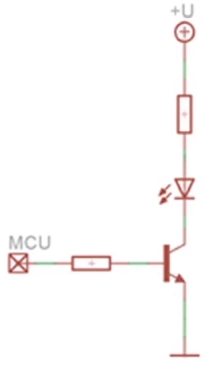
* Na paralelní vstup jsou připojena tlačítka
* Na pinu sp se nastaví parallel a pošle se jeden pulz na clk, tím se stav všech tlačítek zapíše do registru
* Poté se na pinu sp nastaví serial a na clk se pošle tolik pulzů, kolik je tlačítek
* Po každém hodinovém pulzu se přečte datový vstup
* => hodnoty se „vytáhnou“ z PISO do MCU
* Výhody:
  + Konstantní počet pinů (3)
  + Detekce současně stisknutých tlačítek
* Nevýhody:
  + Sériové čtení – rychlý stisk nemusí být zachycen
  + Počet tlačítek omezen velikostí registru, rostoucí počet tlačíte bude mít za následek pomalejší čtení

## Zapojení do matice (klávesnice)

* Horní čtyři piny jsou vstupy, dolní výstupy
* MCU zná pin, na kterém je připojeno tlačítko (jeden pin ze čtyř spodních)
* Programovou detekcí se zjistí, se kterou za čtyř horních větví je spodní spojena a dojde k detekci stisknutí
* Maximální počet tlačítek = vstupy × výstupy
* Výhody:
  + Jednoduchá aplikace
  + Střední obsazenost pinů
  + Rychlé čtení
* Nevýhody:
  + Detekce a rozlišení jsou složítější

# LED

## Přímé připojení na PIN

* Jak vypočítat odpor:
  + Známe proud procházející diodou (typicky 20 mA – úbytek 2 V zjistíme z charakteristiky)
  + Vezmeme celkové napětí (5 V), odečteme úbytek (2 V) a vydělíme proudem (20 mA)
  + (5-2)/(20×10^-3) = 150 Ω
* Výhody:
  + Jednoduché
* Nevýhody:
  + Počet obsazených pinů
  + Proudová náročnost

## Připojení přes tranzistor

* Menší proudové zatížení, stejná obsazenost pinů
* Lze spínat jiné napětí, než které je na výstupních pinech MCU

# Displeje

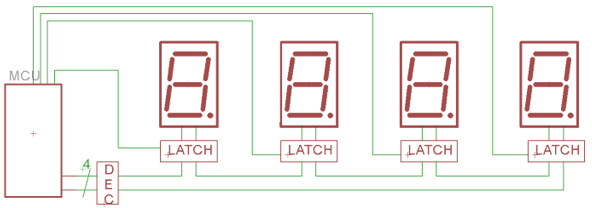
## Přímé zapojení

* Velká obsazenost IO pinů

## Zapojení přes dekodér

* Poloviční obsazenost pinů
* Omezený počet zobrazení, nevyužitá desetinná tečka

## Zapojení na společnou sběrnici

* V jeden moment svítí pouze jeden displej
* Hodnota je vysílána na všechny displeje a odpovídá právě rozsvícenému displeji
* displeje se v zobrazování rychle střídají (obnovovací frekvence)
* Počet pinů = počet displejů + počet segmentů
* Nevýhoda: MCU musí stále přepínat displeje a nastavovat hodnoty
* Hodnoty určené pro jednotlivé displeje lze ukládat do záchytných registrů (hodnota se zapíše na společnou sběrnici a na příslušný LATCH se pošle zápisový pulz)

# Inteligentní displeje

* maticové displeje
* Mají vlastní řadič, do kterého se z MCU posílají příkazy

# Alfanumerický displej

* Nemůžeme pracovat přímo s jednotlivými body
* Pracuje s vnitřní ASCII tabulkou

# Grafický displej

* Lze adresovat přímo jednotlivé body
* Monochromatické (jedna LED na bod)
* Barevné (tři LED na bod)

# Zákmity a jejich odstranění

* způsobují je mechanické nedokonalosti
* změna stavu na opačnou hodnotu je doprovázena opakovanými přechody
* Použití softwarových (opakované čtení) a hardwarových metod (Schmittův klopný obvod + kapacitor)