

# Výuková laboratorní sestava - seminář pro učitele

## Použití UART a radia na platformě micro:Bit

Jakub Vodsed'álek

Katedra měření  
Fakulta elektrotechnická  
České vysoké učení v Praze

30. listopadu 2017



# Obsah

## ① Úvod

## ② UART

- UART - úvod

- UART - výstup

  - Prostý výpis

  - Voltmeter

- UART - vstup

  - Blokovací čtení

  - Neblokovací čtení

- UART - přesměrování na piny

## ③ Radio

- Radio - komunikace mezi kity

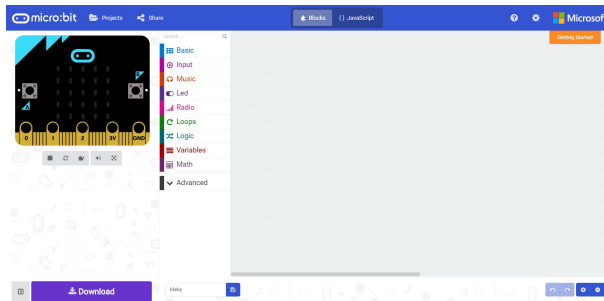
  - Bez řešení kolizí

  - S řešením kolizí

## ④ Generátor

# Úvod I

- Zaměřeno na grafické programovací prostředí dostupné na <https://makecode.microbit.org>
  - Po prvním začtení dostupné i bez připojení k internetu
  - Zároveň umožňuje přepínání mezi grafickým jazykem a JavaScriptem



Obrázek: Grafické programovací prostředí pro microBit.

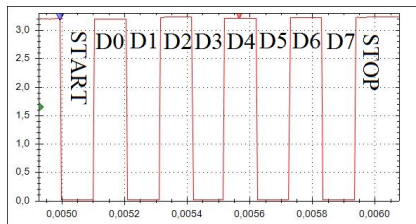
# Úvod II

- Dokumentace k programovacímu prostředí a různé ukázky na <https://makecode.microbit.org/docs>
- Ukázkové programy použité v této prezentaci dostupné na <http://embedded.fel.cvut.cz/platformy/bbc/seminar/materialy>

# UART

## Úvod


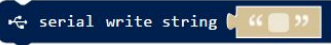

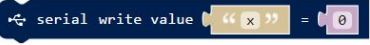
- Výchozí stav:
  - Na USB
  - baud rate 115200 Bd
  - 8 datových bitů
  - 1 stop bit
  - Bez parity
- Možné přesměrovat na piny a zde měnit baud rate
- Možné blokovací i neblokovací čtení



Obrázek: Záznam odesílání znaku 'U' přes UART.


# UART

## Výstup

-  - pro výpis čísla
-  - pro výpis textu
-  - pro výpis textu zakončeného odřádkováním
-  - pro výpis textu a hodnoty zakončeného odřádkováním
- Odřádkování pomocí znaků "`\r\n`" - CR a LF
- Při výpisu speciálních znaků jako `'\'` automaticky připíše do textového řetězce další `\` - není nutné ošetřovat, ale zároveň znemožňuje ruční zalamování pomocí "`\r\n`" v blokovém editoru

# UART I

## Prostý výpis

- Ukázka výpisu informace o stisknutí tlačítka na UART
- ① Zjišťování stisknutí tlačítka v hlavní programové smyčce - ukázka
  - Ukázkový program UART\_vypis1.hex
  - Při stisknutí tlačítka je na UART vyslána informace o jeho stisknutí
  - Zpráva je vypisována dokud je tlačítko stisknuto
  - Využití 
  - Funguje na principu pollingu - periodického čtení stavu tlačítek

# UART II

## Prostý výpis

### ② Zjišťování stisknutí tlačítka pomocí odchytnutí události - ukázka

- Ukázkový program UART\_vypis2.hex
- Při stisknutí tlačítka je na UART vypsána informace o jeho stisknutí
- Zpráva je vypsána pouze jednou pro jedno stisknutí



- Využití
- Funguje na principu odchytnutí události stisknutí tlačítka



# UART I

## Voltmeter

- Praktická ukázka využití výpisu pro realizaci jednoduchého voltmetru



- Použití bloků
- První slouží ke čtení analogové hodnoty na pinu (A/D převodník)
- Druhý slouží k přemapování rozsahu proměnné/čísla
- Pro ukázku budeme měřit napětí generované pomocí LEO

# UART II

## Voltmeter

### 1 Výpis na více řádků - ukázka

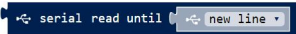
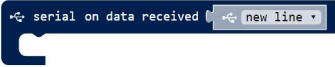
- Ukázkový program UART\_voltmeter1.hex
- S periodou jedné sekundy vypíše na UART napětí v mV na pinech P0, P1 a P2
- Každé napětí je vypsáno na jednotlivý řádek

### 2 Výpis na jeden řádek - ukázka

- Ukázkový program UART\_voltmeter2.hex
- S periodou jedné sekundy vypíše na UART napětí v mV na pinech P0, P1 a P2
- Napětí jsou vypsána na jeden řádek
- Jednotlivé hodnoty jsou odděleny pomocí středníku (';')

# UART

## Vstup

-  - čte vstup UARTu dokud nepřijde ukončovací znak
-  - po přijetí vybraného znaku se spustí vykonávání daného bloku
- Pozor na čtení po "new line" - momentálně nefunguje pro "\r\n" (CR a LF), které jinak platforma standartě používá (nefunguje ani pouze pro "\n" LF)
- Pokud chceme číst po odřádkování (ENTER) musíme kód přepsat v JS - v blocích nelze
- Při přepisování dát pozor na nastavení terminálu na PC (odřádkování může být LF nebo CR a LF)

# UART

## Blokovací čtení

- Blokovací čtení - program se neposune do další části dokud na vstup nepřijdou požadovaná data (typicky ukončovací znak)
- ① Blokovací čtení - ukázka
  - Ukázkový program UART\_cteni1.hex
  - Po přijmutí ukončovacího znaku odešle přijaté znaky zpět na výstup ve formátu "Přijato: xxxx"
  - Jako ukončovací znak je pro výchozí stav zvolen znak čárky ','

# UART

## Neblokovací čtení

- Neblokovací čtení - čtení probíhá mimo hlavní programovou smyčku a hlavní programová smyčka tak může nezávisle běžet
- Funguje na principu spuštění části programu událostí (přijetí ukončovacího znaku)

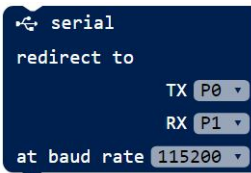
### 1 Neblokovací čtení - ukázka

- Ukázkový program UART\_cteni2.hex
- Po přijmutí ukončovacího znaku odešle přijaté znaky zpět na výstup ve formátu "Přijato: xxxx"
- Jako ukončovací znak je pro výchozí stav zvolen znak čárky ','

# UART

## Přesměrování

- UART je možné přesměrovat z USB na piny kitu
- V případě přesměrování je možné měnit baud rate UARTu



- - blok zajišťující přesměrování UARTu na piny

### 1 Přesměrování UART - ukázka

- Ukázkový program UART\_vypis\_presmerovani.hex
- TX přesměrováno na P0, RX přesměrováno na P1
- Každou 0.5 sekundu vypíše číslo
- Vypisované číslo je po spuštění programu 0
- Při stisknutí tlačítka A je číslo o 1 zvětšeno
- Při stisknutí tlačítka B je číslo o 1 zmenšeno

# Radio

## Komunikace mezi kity

- Vlastní protokol na komunikaci mezi kity
- Pro uživatele se bloky chovají totožně jako bloky pro UART



- - odešle přes radio textový řetězec (existují ještě bloky send number a send value, které mají stejnou funkcionalitu jako v případě UARTu)



- - po přijetí vybraného typu dat přes radio se spustí proces



- - nastavení komunikační skupiny kitů (aby spolu mohly kity komunikovat musejí být ve stejné skupině)
- Kity mohou komunikovat přes radio pouze pokud jsou přiřazeny do stejné skupiny

# Radio

## Komunikace mezi kity bez řešení kolizí

- ① Komunikace mezi kity bez řešení kolizí - ukázka
  - Ukázkový program Radio1.hex
  - Po stisknutí tlačítka se všem ostatním kitům v dosahu signálu zobrazí na matici LED písmeno stisknutého tlačítka (A/B)
  - Neřešíme kolize - "křičí jeden přes druhého"



# Radio

## Komunikace mezi kity s řešením kolizí

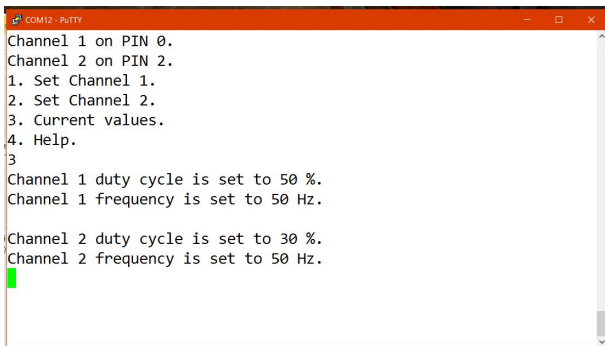
### ① Komunikace mezi kity s řešením kolizí - ukázka

- Ukázkový program Radio2.hex
- Ve výchozím stavu jsou všechny kity nastaveny jako přijímače
- Po stisknutí libovolného tlačítka na libovolném kitu se daný kit nastaví jako vysílač
- Zároveň zprávou zablokuje ostatní kity ve stavu přijímače
- Vysílač nyní může provádět, to co v minulém případě všechny kity (Po stisknutí tlačítka se všem ostatním kitům v dosahu signálu zobrazí na matici LED písmeno stisknutého tlačítka (A/B))
- Přijímače pouze přijímají a nic jiného nemohou
- Vysílač se své role vzdá stisknutím obou tlačítek zároveň
- Tím uvolní pozici vysílače a libovolný kit ho může zastoupit stisknutím tlačítka a celý proces se opakuje

# Ukázka komplexnějšího programu

## 1 Generátor

- Ukázkový program SquarGenerator.hex
- Na dvou kanálech generuje obdélníkový signál
- Komunikace s PC pomocí UARTu
- Přes terminálové prostředí umožňuje nastavování střídy a frekvence signálu



```
COM12 - PuTTY
Channel 1 on PIN 0.
Channel 2 on PIN 2.
1. Set Channel 1.
2. Set Channel 2.
3. Current values.
4. Help.
3
Channel 1 duty cycle is set to 50 %.
Channel 1 frequency is set to 50 Hz.

Channel 2 duty cycle is set to 30 %.
Channel 2 frequency is set to 50 Hz.
█
```

**Děkuji za pozornost!**

**Jakub Vodseďálek**

`vodsejak@fel.cvut.cz`