

5 SÍŤ

Tato kapitola se věnuje propojení dvou nebo více micro:bitů pomocí sítě

Co se naučíte

- Základní principy sítí
- Propojit dva micro:bity pomocí drátu a přenést informaci
- Totéž bezdrátově, pomocí rádia

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích

PRŮVODCE HODINOU V-1

Studenti se v této hodině seznámí s některými pojmy z počítačových sítí. Propojí si micro:bity pomocí kabelu a vyzkouší si přenos dat.

Co bude v této hodině potřeba:

- PC s editorem mu.
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

Vysvětlíte studentům pojmy počítačová síť, drátová, bezdrátová. Diskutujte o příkladech.

Vysvětlíte pojmy duplex, half duplex a simplex, síťový protokol.

Nechte studenty ať se domluví, kdo z nich bude Vysílač (bude vysílat signál) a kdo Přijímač (bude přijímat signál).

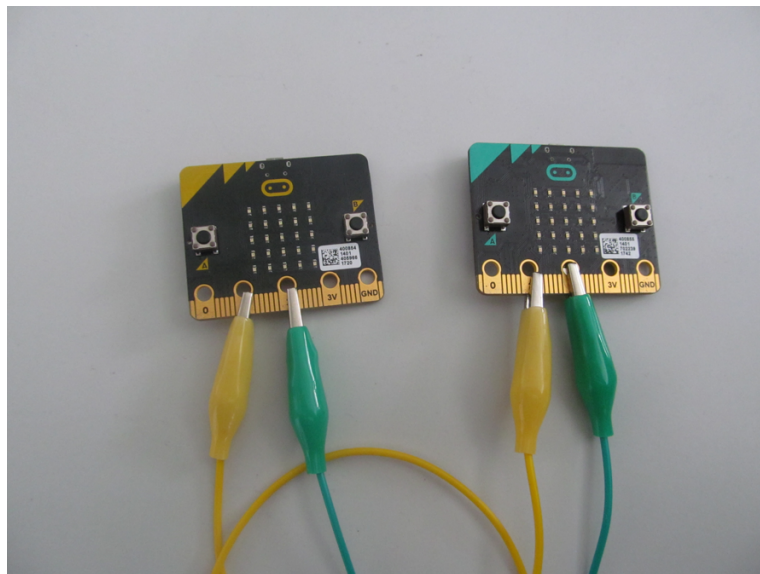
Na micro:bitu označeném *Vysílač* odlaďte následující program:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.     else:
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin2.write_digital(1)
11.    else:
12.        pin2.write_digital(0)
13.    display.clear()
14.    sleep(10)
```

Obdobně na micro:bitu *Přijímač*:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show("A")
5.     elif pin2.read_digital():
6.         display.show("B")
7.     sleep(1000)
8.     display.clear()
```

Propojte po dvou micro:bity, tak že spojíte (nejlépe vodiči s krokodýlky) vzájemně piny 1 na obou stranách a stejně tak piny 2.



Vyzkoušejte přenos signálu.

Vysvětlete studentům, že jaký signál *Vysílače* bude mít u *Přijímače* jaký význam, záleží pouze na předchozí domluvě. Jedná se o tzv. *Síťový protokol*.

2. krok 20 minut

Obraťte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu. Po vyzkoušení s nimi prodiskutujte, že se jedná vlastně o paralelní přenos, neboť můžeme přenášet signál po více vodičích současně. Nechte studenty přijít na to, kolik různých možností přeneseného kódu umožňuje daná kombinace – dvě na druhou – čtyři možnosti.

Dle úrovně studentů a zbylého času zkuste nechat naprogramovat přenos čísel 0 až 3 (1 až 4) nebo pouze přidejte k danému programu možnost zobrazení C při stisku obou kláves současně.

PRACOVNÍ LIST V-1

Co se naučíte

- Co je to počítačová síť, jaké jsou typy sítě
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích

A jděte na to ...

Rozdělte se do dvojic

Popovídejte si s vyučujícím o tom co jsou počítačové sítě a jaké jsou jejich typy.

Nyní se domluvte kdo ve dvojici bude *Vysílač* a kdo *Přijímač*.

Vysílač odladí na Micro:bitu následující program:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.     else:
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin2.write_digital(1)
11.    else:
12.        pin2.write_digital(0)
13.    display.clear()
14.    sleep(10)
```

Přijímač odladí následující:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show("A")
5.     elif pin2.read_digital():
6.         display.show("B")
7.     sleep(1000)
8.     display.clear()
```

Propojte nyní Micro:bity dvěma kabely s krokodýlky. Vzájemně propojíte na obou stranách piny1 a piny2. Micro:bity připojte ke zdroji energie a pro jistotu resetujte a vyzkoušejte přenos signálu.

Vyměňte si role a zopakujte si zadání v opačných pozicích.

Jedná se o paralelní přenos signálu – vysvětlete si tento pojem.

- Kolik stavů můžeme přenést při tomto zapojení
- Jak byste upravili programy, abyste přenesli i písmeno C?
- Jak byste upravili programy, abyste přenášeli čísla 0 až 3 nebo 1 až 4?
- Kolik různých stavů je teoreticky možné takto mezi dvěma micro:bity přenášet?

PRŮVODCE HODINOU V-2

Studenti si v této hodině vyzkouší přenos signálu mezi dvěma micro:bity pomocí jednoho vodiče – sériový přenos.

Co bude v této hodině potřeba:

- PC s editorem mu.
- Micro:bit s USB kabelem
- Vodič nejlépe s krokodýlky na obou koncích
- Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

Vysvětlíte studentům pojem sériový přenos. Řekněte jim, že v této hodině se programy budou týkat sériového přenosu.

Nechte studenty ať se domluví, kdo z nich bude *Vysílač* (bude vysílat signál) a kdo *Přijímač* (bude přijímat signál).

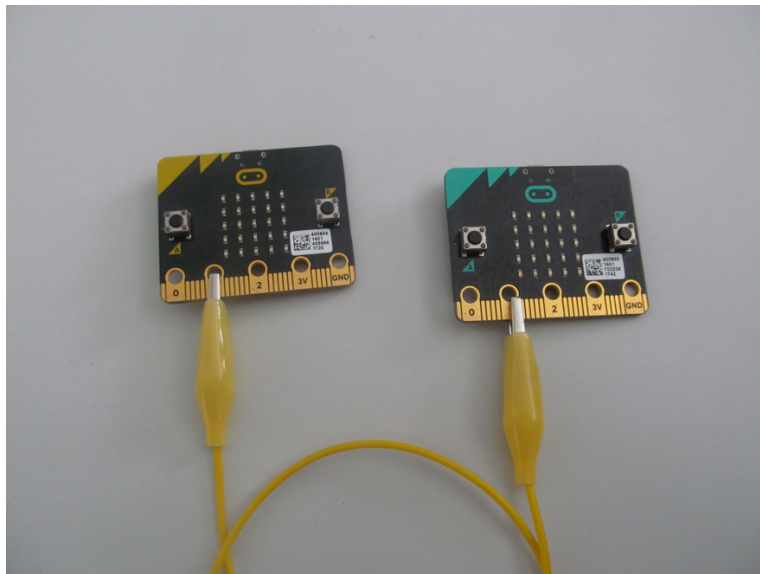
Na micro:bitu označeném *Vysílač* odlaďte následující program:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.         sleep(500)
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin1.write_digital(1)
11.        sleep(2000)
12.        pin1.write_digital(0)
13.        display.clear()
```

Obdobně na micro:bitu *Přijímač*:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         start = running_time()
5.         while pin1.read_digital():
6.             pass
7.         konec = running_time()
8.         cas = konec - start
9.         if cas < 1000:
10.            display.show("A")
11.        else:
12.            display.show("B")
13.        sleep(1000)
14.        display.clear()
```

Propojte po dvou micro:bity, tak že spojte (nejlépe vodiči s krokodýlky) vzájemně piny 1 na obou stranách. Vyzkoušejte přenos signálu.



Pokud micro:bit Přijímač zaznamená na pinu1 signál, zjišťuje si jeho délku. Pokud je délka kratší než 1 sekunda považuje to za typ jedna signálu (např. binární 0). pokud je delší, pak za typ dva signálu (např. binární jedna).

2. krok 20 minut

Obráťte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu. Po vyzkoušení s nimi prodiskutujte, jaký význam může mít přenesený signál.

Prodiskutujte možnosti použití daného typu přenosu:

- Morseova abeceda
- ASCII kódy

PRACOVNÍ LIST V-2

Co se naučíte

- Sériový přenos
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Vodiče nejlépe s krokodýlky na obou koncích

A jděte na to ...

Rozdělte se do dvojic a domluvte se kdo ve dvojici bude *Vysílač* a kdo *Přijímač*.

Vysílač odladí na Micro:bitu následující program:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.         sleep(500)
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin1.write_digital(1)
11.        sleep(2000)
12.        pin1.write_digital(0)
13.        display.clear()
```


Přijímač odladí následující:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         start = running_time()
5.         while pin1.read_digital():
6.             pass
7.         konec = running_time()
8.         cas = konec - start
9.         if cas < 1000:
10.            display.show("A")
11.        else:
12.            display.show("B")
13.        sleep(1000)
14.        display.clear()
```

Propojte nyní Micro:bity kabelem s krokodýlky. Vzájemně propojíte na obou stranách piny1. Micro:bity připojte ke zdroji energie a pro jistotu resetujte a vyzkoušejte přenos signálu.

Vyměňte si role a zopakujte si zadání v opačných pozicích. Jak pozná *Přijímač* o jaký signál se jedná?

Jedná se o sériový přenos signálu – vysvětlíte si tento pojem. Napadá vás druh kódování, které lze tímto způsobem přenášet?

PRŮVODCE HODINOU V-3

Studenti si v této hodině vyzkouší oboustranný přenos signálu mezi dvěma micro:bity – duplexní přenos.

Co bude v této hodině potřeba:

- PC s editorem mu.
- Micro:bit s USB kabelem
- Dva vodiče nejlépe s krokodýlky na obou koncích
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 30 minut

Rozdejte studentům micro:bity a kabely. Řekněte jim ať se rozdělí do dvojic.

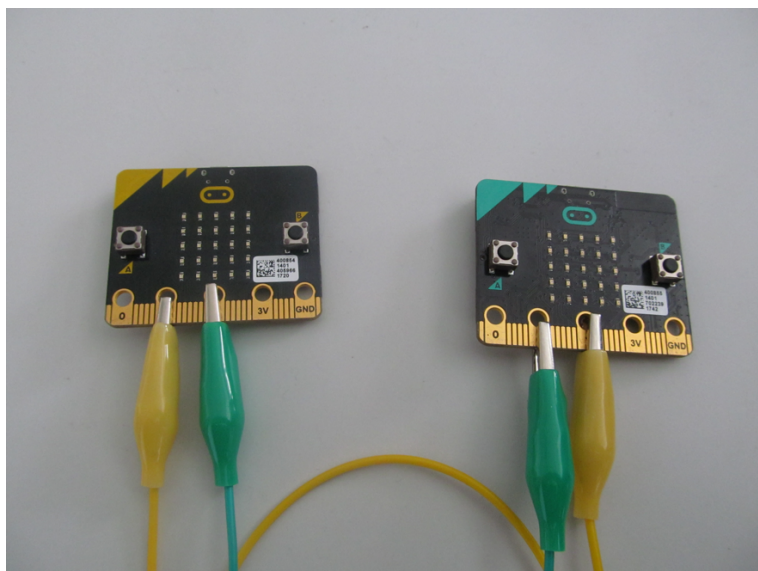
Vysvětlíte studentům pojem duplexní (obousměrný) přenos. Řekněte jim, že v této hodině budeme informaci přenášet oběma směry.

V této hodině budeme na obou micro:bitech používat stejný program. Nechte studenty, ať si naprogramují a odladí následující program:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show(Image.HAPPY)
5.     else:
6.         display.clear()
7.     if button_a.is_pressed():
8.         pin2.write_digital(1)
9.     else:
10.        pin2.write_digital(0)
11.        sleep(100)
```

Při úvodním ladění si mohou studenti vystačit i sami. Programy jsou připravené tak aby vysílali na pinu2 a naslouchali na pinu1. Pokud si studenti propojí vodičem pin2 s pinem1, bude jim program fungovat i v rámci jednoho micro:bitu.

Nyní propojte dva micro:bity tak, že pin2 na jednom připojíte k pinu1 na druhém a naopak. Oba micro:bity připojte ke zdroji energie a zresetujte. Vyzkoušejte přenos signálu.



2. krok 15 minut

Nechte studenty upravit druhy signálu atd. Můžete upravit program i tak, že bude rozlišovat druh signálu dle jeho délky. Viz minulá hodina.

Pohovořte o výhodách a nevýhodách duplexního přenosu.

PRACOVNÍ LIST V-3

Co se naučíte

- Obousměrný přenos
- Propojit dva micro:bity drátovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit
- Dva vodiče nejlépe s krokodýlky na obou koncích

A jděte na to ...

Rozdělte se do dvojic. Každý si odlaďte na svém micro:bitu následující program:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show(Image.HAPPY)
5.     else:
6.         display.clear()
7.     if button_a.is_pressed():
8.         pin2.write_digital(1)
9.     else:
10.        pin2.write_digital(0)
11.        sleep(100)
```

Při testování můžete jednotlivé micro:bity testovat tak, že propojíte vodičem pin2 a pin1.

Je-li program odladen, můžete propojit dva micro:bity, tak že pin1 na jednom propojíte s pinem2 na druhém a naopak. Na pin2 se vysílá na pinu1 naslouchá.

Vyzkoušejte obousměrný přenos.

PRŮVODCE HODINOU V-4

Studenti si v této hodině vyzkouší radiový přenos signálu mezi dvěma micro:bity.

Co bude v této hodině potřeba:

- PC s editorem mu.
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor – v této hodině jsou doporučeny dva dataprojektory, je třeba promítat dva různé programy současně (anebo se raději spokojte pouze s pracovními listy)
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 25 minut

Rozdějte studentům micro:bity. Řekněte jim ať se rozdělí do dvojic.

Vysvětlíte studentům pojem radiový přenos.

Nechte studenty ať se domluví, kdo z nich bude *Vysílač* (bude vysílat signál) a kdo *Přijímač* (bude přijímat signál). Můžete je nechat i domluvit si kanál, viz dále.

Na micro:bitu označeném *Vysílač* odlaďte následující program:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     if button_a.is_pressed():
7.         radio.send("Zprava")
8.         sleep(1000)
9. radio.off()
```

Obdobně na micro:bitu *Přijímač*:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     zprava = radio.receive()
7.     if (zprava):
8.         display.scroll(zprava)
9.         zprava = ""
10. radio.off()
```

Můžete studenty nechat nejprve vyzkoušet programy bez nastavení kanálu. Až si budou vzájemně posílat zprávy, vysvětlíte jim, že pomocí nastavení kanálu (0-83) si mohou vybrat kdo s kým si bude povídat. Můžete při tom pohovořit o bezpečnosti a typu útoku man in the middle.

2. krok 20 minut

Obraťte nyní role v týmu, aby si studenti vyzkoušeli oba směry přenosu.

PRACOVNÍ LIST V-4

Co se naučíte

- Sériový přenos
- Propojit dva micro:bity rádiovou sítí
- Odeslání i příjem signálu

Co budete potřebovat

- PC s nainstalovaným editorem mu
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit

A jděte na to ...

Rozdělte se do dvojic a domluvte se kdo ve dvojici bude *Vysílač* a kdo *Přijímač*.

Vysílač odladí na Micro:bitu následující program:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     if button_a.is_pressed():
7.         radio.send("Zprava")
8.         sleep(1000)
9. radio.off()
```

Přijímač odladí následující:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     zprava = radio.receive()
7.     if (zprava):
8.         display.scroll(zprava)
9.         zprava = ""
10. radio.off()
```

Vyzkoušejte přenos signálu. K čemu slouží nastavení kanálu (channel)?

Vyměňte si role a zopakujte si zadání v opačných pozicích.

Pohovořte si o možné bezpečnosti přenosu.

PRŮVODCE TEORIÍ

Počítačové sítě

Než se pustíme do práce se sítí uvedeme si pár teoretických definic, abychom později lépe porozuměli tomu, co stavíme a programujeme.

Počítačové sítě dělíme dle přenosového média na:

- Drátové
- Bezdrátové
 - Wi-Fi
 - Bluetooth
 - Radio
 - Infra

My budeme micro:bity propojovat jak *drátově*, tak *bezdrátově* pomocí radia. Micro:bit sice obsahuje i Bluetooth přijímač a vysílač, ale jak bude dále vysvětleno v MicroPythonu jej nelze použít.

Dle směru vysílání dělíme sítě na:

Simplexové – vysílání jde pouze jedním směrem, na jedné straně se nachází vysílač (sender nebo transmitter) a na druhé přijímač (receiver).

Half duplex (poloviční duplex) – vysílání může jít oběma směry, ale v daném okamžiku pouze jedním směrem.

Duplex – vysílání může jít v daném okamžiku oběma směry současně

Síťový protokol – soubor předem domluvených pravidel, kterými se řídí daný síťový přenos.

Drátový přenos

Pro následující příklad budeme potřebovat dva micro:bity. Oba propojíme pomocí dvou kabelů tak, že propojíme vzájemně piny 1 a piny 2 na obou micro:bitech. (Pin1 na prvním micro:bitu s pinem1 na druhém micro:bitu a stejně i pin2.) Opět můžeme s výhodou použít kabely s „krokodýlky“ na koncích.

Je třeba stanovit, který micro:bit bude „Vysílač“ a který „Přijímač“. Jedná se tedy o simplexový přenos. Funkce programů je následující na prvním micro:bitu stiskneme tlačítko A nebo B a na druhém micro:bitu se vzápětí rozsvítí symbol A nebo B. Jedná se tedy o binární stav, jeden ze symbolů může reprezentovat jedničku (pravdu – true) a druhý nulu (nepravdu – false). Pomocí síťového protokolu lze domluvit, co v daném případě, který symbol znamená a o jakou informaci se jedná.

Na micro:bitu „Vysílač“ nahrajte a odlad'te následující program:

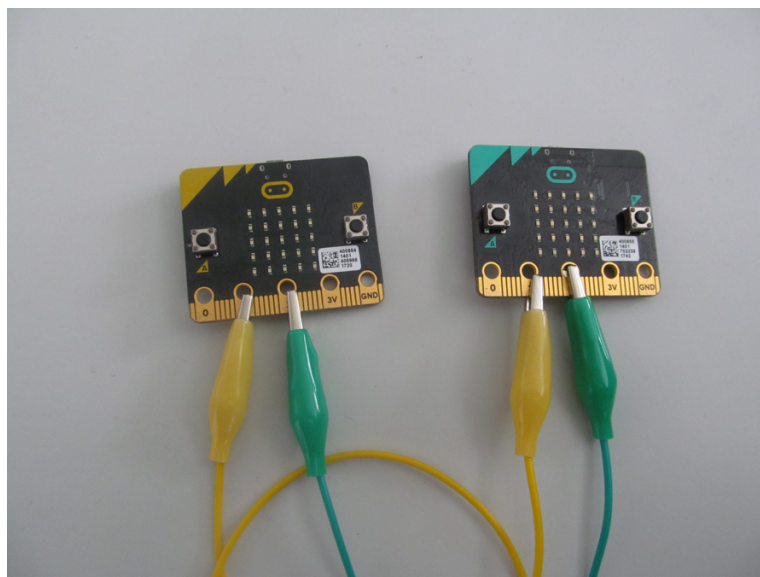
```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.     else:
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin2.write_digital(1)
11.    else:
12.        pin2.write_digital(0)
13.    display.clear()
14.    sleep(10)
```

Program v nekonečném cyklu sleduje, zda bylo stisknuté tlačítko a, nebo tlačítko b a podle toho vysílá jedničku na daném propojení. Pro kontrolu zobrazuje rovněž písmeno dle stisknutého tlačítka.

Na micro:bitu „Přijímač“ nahrajte a odlad'te následující program:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show("A")
5.     elif pin2.read_digital():
6.         display.show("B")
7.     sleep(1000)
8.     display.clear()
```

Tento program v nekonečné smyčce kontroluje, zda je na některém z připojených vodičů signál a pak zobrazuje odpovídající písmeno. Je třeba si také uvědomit, že v daném případě nelze říci co se stane, když bude signál na obou vodičích. To jaké písmeno (zpráva) se zobrazí záleží na naprogramování, tedy na předchozí domluvě. Tomu se říká *Síťový protokol*.



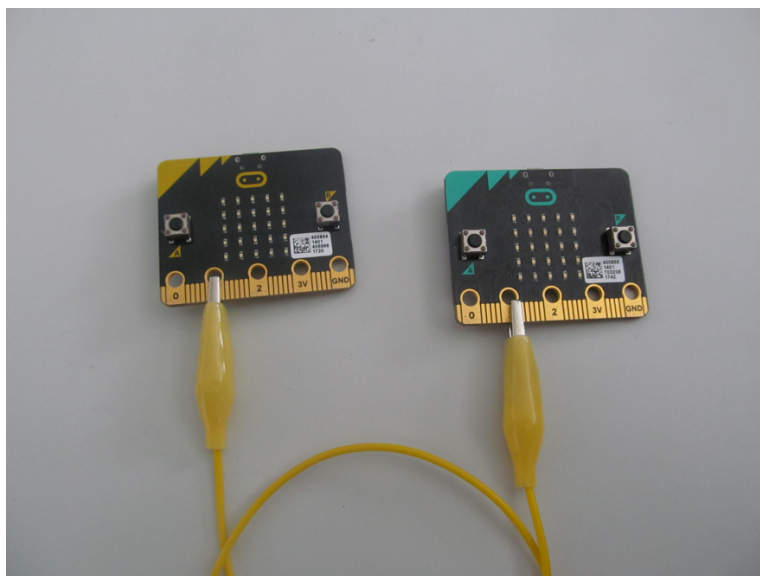
Tomuto případu, kdy signál putuje po více kabelech se říká *paralelní přenos* – signály posíláme vedle sebe. Bylo by možné využít tohoto zapojení k přenosu 4 znaků, s následujícími možnostmi:

1. žádný signál
2. signál na prvním vodiči
3. signál na druhém vodiči
4. signál na obou vodičích

Kolik znaků bychom mohli přenést při maximálním možném počtu vodičů, který umožňuje micro:bit? Odpověď závisí na tom, zda použijeme tři vodiče v běžném režimu (pomocí krokodýlků) anebo až 17 vodičů, pokud použijeme datový shield. Vždy je to však 2^n .

Pokud chceme propojit micro:bity pouze jedním kabelem, a i pak přenášet jiný signál, než zapnuto vypnuto, je nutno se domluvit na nějakém protokolu. V následujícím příkladu odlišujeme přenesený signál dle jeho délky – 500 ms nebo 1500 ms. Měříme jeho délku pomocí funkce `running.time()` a pokud je signál kratší než jedna sekunda, považujeme jej za první stav a pokud je delší než jedna sekunda za druhý stav. Takovémuto způsobu přenosu se říká *sériový* – signály posíláme za sebou.

Micro:bity propojíme pouze jedním kabelem mezi piny 1.



Na micro:bit „Vysílač“ nahrajeme následující program:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show("A")
5.         pin1.write_digital(1)
6.         sleep(500)
7.         pin1.write_digital(0)
8.     if button_b.is_pressed():
9.         display.show("B")
10.        pin1.write_digital(1)
11.        sleep(2000)
12.        pin1.write_digital(0)
13.        display.clear()
```

Program v nekonečné smyčce hlídá stisk kláves A a B a při stisku vyšle signál odpovídající délky a pro kontrolu zobrazí i kód stisknuté klávesy. Na micro:bit „Přijímač“ nahrajeme následující program:

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         start = running_time()
5.         while pin1.read_digital():
6.             pass
7.         konec = running_time()
8.         cas = konec - start
9.         if cas < 1000:
10.            display.show("A")
11.        else:
12.            display.show("B")
13.            sleep(1000)
14.            display.clear()
```

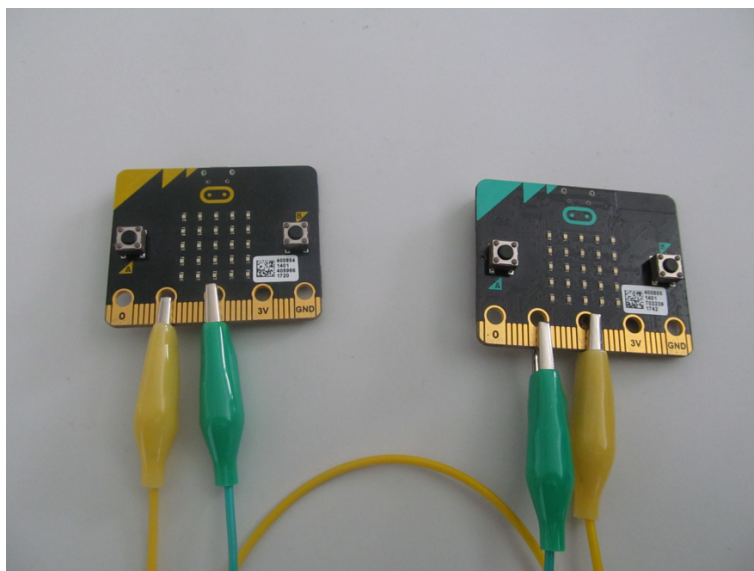
Program v nekonečné smyčce hlídá, zda se objeví signál na pin1. Pokud ano zaznamená si jeho čas. Funkce `running.time()` vrací čas v milisekundách od spuštění micro:bitu. Nyní se čeká dokud je na pin1 signál a po jeho ukončení opět změříme čas. Spočítáme dobu signálu odečtením začátku od konce. Je-li signál kratší než jedna sekunda považujeme jej za první typ znaku – A. Je-li delší pak za B.

Tímto způsobem si můžeme přenášet binární signál, namísto znaků A a B, můžeme použít 1 a 0 (nebo naopak). Tímto způsobem je možné například přenášet ASCII kód, když si řekneme, že osm za sebou přenesených znaků je kód jednoho ASCII znaku. Lze takto například přenášet i Morseovu abecedu, pokud si například řekneme, že kratší znak je tečka, delší znak je čárka.

Zájemci si mohou zkusit obě tyto úlohy řešit. Vyřešení úlohy s přenosem kódu Morseovy abecedy naleznete na internetových stránkách MicroPythonu.

Nyní opět změníme zapojení. Propojíme micro:bity tak, že kabel bude na jednom micro:bitu připojen na pin1 a na druhém na pin2. S druhým kabelem to uděláme naopak. Micro:bity budou tedy zapojeny do kříže na pinech 1 a 2. Oba micro:bity budou nyní současně „Přijímač“ i „Vysílač“ na oba tedy nahrajeme stejný kód.

```
1. from microbit import *
2. while True:
3.     if pin1.read_digital():
4.         display.show(Image.HAPPY)
5.     else:
6.         display.clear()
7.     if button_a.is_pressed():
8.         pin2.write_digital(1)
9.     else:
10.        pin2.write_digital(0)
11.        sleep(100)
```



Význam kódu je velice jednoduchý. Program v nekonečné smyčce provádí následující. Pokud je na pinu 1 signál, zobrazí se smajlík. Pokud je stisknutá klávesa A vyšleme signál na pin 2. To samé se děje i na druhé straně. Jedná se o *duplexní signál*, současně přenášíme informaci oběma směry.

Možná vás nyní napadlo, že by mohl stačit jen jeden kabel. V tom případě musíme řešit následující dva problémy:

1. Pokud by micro:bit vyslal signál na vodič, současně by se na něm při testu signálu na témže vodiči načetla jednička a měl by za to, že signál i přijímá. Této situaci se dá zabránit tak, že v případě vysílání signálu se netestuje stav signálu na vodiči.
2. Další problém by vznikne pokud oba micro:bity vysílají současně. Takovémuto případu se říká *kolize*. Řešením je, že micro:bit před vysláním signálu nejprve ověří, zda na kabelu již není signál a pak teprve začne vysílat. Je třeba si uvědomit, že to není dokonalé řešení, oba micro:bity mohou testovat a začít vysílat ve stejném okamžiku. Je to málo pravděpodobné, ale ne zcela vyloučené. I tento případ má řešení, po začátku vysílání micro:bit po náhodné době na malou chvilku přeruší signál a ověří zda na vodiči není signál z druhé strany a pak buď počká nebo vysílá dál. Doba po, které se to testuje, nesmí být vždy stejná, aby opět obě strany nedělaly totéž.

Takovémuto přenosu jak již víme by se říkal *half duplex*. Můžete si jej zkusit naprogramovat, včetně řešení kolize. Toto je zjednodušení způsobu jakým je přenášen internet po běžných ethernetových kabelech.

Bluetooth přenos

Ačkoliv Micro:bit obsahuje přijímač i vysílač bluetooth signálu, nelze jej v MicroPythonu využít. Na vině je to, že do paměti micro:bitu nelze současně umístit překladač MicroPythonu i kód pro obsluhu bluetooth kvůli velikosti.

Problémem je i to, že pokud jsme na micro:bitu pracovali s MicroPythonem, nelze jej pomocí bluetooth spárovat s jiným zařízením. Jediné možné řešení je nahrát na micro:bit libovolný program vytvořený grafickým rozhraním na stránkách microbit.org. Pak již micro:bit můžeme spárovat s jiným zařízením prostřednictvím bluetooth.

Rádiový přenos

Velice zajímavou možností jak mohou spolu dva Micro:boty komunikovat je bezdrátový rádio přenos. Je možné naladit celkem 84 kanálů označených 0 až 83. Jedná se o frekvenci 2400 MHz (odpovídá kanálu 0), každý kanál má rozsah cca. 1 MHz.

Práce s tímto přenosem je velmi jednoduchá, MicroPython obsahuje knihovnu, která má v sobě metody umožňující přímo přenos textového řetězce (nebo čísla). Ukázka je v následujícím příkladě. Na straně odesílatele:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     if button_a.is_pressed():
7.         radio.send("Zprava")
8.         sleep(1000)
9. radio.off()
```

A na straně příjemce:

```
1. from microbit import *
2. import radio
3. radio.on()
4. radio.config(channel=23)
5. while True:
6.     zprava = radio.receive()
7.     if (zprava):
8.         display.scroll(zprava)
9.         zprava = ""
10. radio.off()
```

Parametr channel je nepovinný, pokud jej neuvedete, pak předvolený je kanál 7. Na druhou stranu, pokud si má spolu povídat větší množství micro:bitů, je nutné, aby ty které k sobě patří používaly nezávislý kanál a nepletly se tak ostatním.

Na tomto příkladě je rovněž vidět potenciální nebezpečí. Pokud útočník ví na kterém kanále si naše micro:bity povídají, může je buď odposlouchávat anebo podstrkovat vlastní zprávy. Jedná se o typ útoku Man in the middle.

Možné použití této technologie se nabízí v následujících možnostech:

- *Dálkový ovladač.* Jeden z micro:bitů můžeme ovládat pomocí gest a přenášet pokyny k druhému, který může něco řídit. Je takto možné například ovládat autíčko, například nějakou stavebnici, kterou lze ovládat pomocí micro:bitu
- *Zabezpečení.* Jeden z micro:bitů může sledovat např. pohyb, intenzitu světla, magnetického pole atd. a při neobvyklém stavu poslat signál jinému, který je umístěn někde, kde snadno vyhlásí poplach.
- Pro přímou komunikaci se micro:bity příliš nehodí, protože není jednoduché zadat zprávu, kterou chceme odeslat. Snad nějakým způsobem výběr z nabídky připravených zpráv a naopak druhý micro:bit může vybírat ze seznamu připravených odpovědí.