

第 5 章

网络互连

学习目标

本章重点学习在两个 micro:bit 之间进行通信的方法。

学习要求

- (1) 了解无线通信的基本概念。
- (2) 掌握使用 radio 模块进行无线通信的程序编写方法。

通过网络将设备连接在一起,可以相互发送和接收信息。

如果把网络分为一系列的层,其中最底层是进行沟通的最基础的层面,通过有线或无线的方式让信号从一个设备传送到另一个设备。

在如图 5.1 所示的例子中,只需要使用两根导线,通过鳄鱼夹连接两块 micro:bit,就可以进行通信。两者都使用引脚 1 作为输出,引脚 2 作为输入,将一个设备中的“输出”连接到另一个设备中的“输入”。

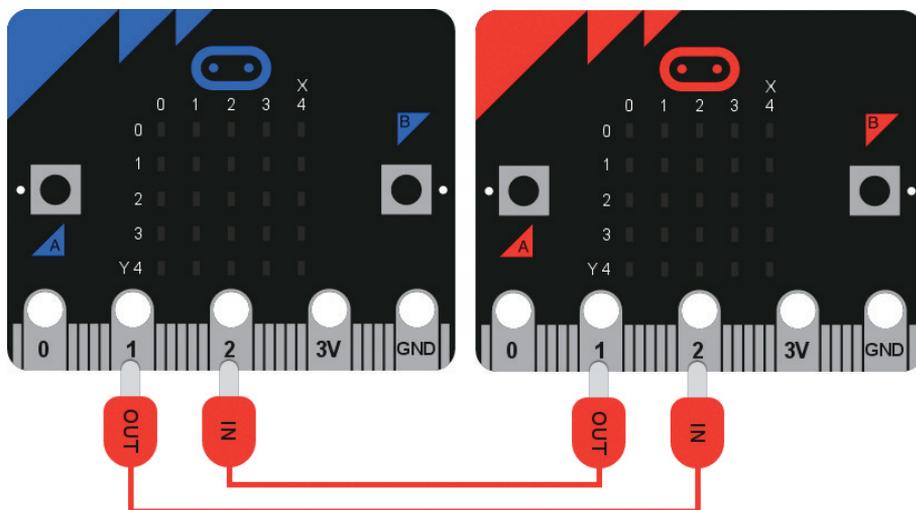


图 5.1 有线通信



5.1 无线通信

在日常生活中,经常会通过手机、笔记本电脑、遥控器等无线方式进行图像、语音、文字等信息的传输,这些设备是如何把信息传输出去的呢?

无线电是指在所有自由空间(包括空气和真空)中传播的电磁波,是其中的一个有限频带,上限频率在 300GHz,下限频率较不统一,常见的有 3kHz~300GHz(国际电信联盟规定)、9kHz~300GHz、10kHz~300GHz。

由于导体中电流强度的改变会产生无线电波,通过调制可以将信息加载于无线电波之上;当电波通过空间传播到达受信端,电波引起的电磁场变化又会在导体中产生电流;通过解调将信息从电流变化中提取出来,就达到了信息传递的目的。

1906 年的圣诞前夜,雷吉纳德·菲森登(Reginald Fessenden)在美国马萨诸塞州采用外差法实现了历史上首次无线电广播,菲森登用小提琴演奏了“平安夜”,以及朗诵了《圣经》片段。位于英格兰切尔姆斯福德的马可尼研究中心在 1922 年开播了世界上第一个定期播出的无线电广播娱乐节目。

无线电经历了从电子管到晶体管再到集成电路,从短波到超短波再到微波,从模拟方式到数字方式,从固定使用到移动使用等各个发展阶段,无线电技术已成为现代信息社会的重要支柱。

micro:bit 主控板内置了 2.4GHz 无线通信模块,能够通过无线电和蓝牙技术与外界进行通信。

5.1.1 一对一通信

想要实现无线电通信的功能,进行通信的两个或多个设备必须在同一个组内。

无线设备只有在同一个组内才能够相互接收到对方发来的信息。发送和接收信息都需要特定的方式。把接收和发射的信息进行比较,看看是否一致,就可以确定通信是否成功。

通过 micro:bit 内置的无线电模块和 MicroPython 的 radio 库,可以不用导线进行设备之间的连接,无线电模块的功能如下。

- (1) 传播的信息具有特定的可配置长度(最多 251B)。
- (2) 接收到的信息是从可配置大小的队列中读取的(队列越大,使用的 RAM 越多),如果队列已满,则会忽略新的信息。
- (3) 信息在预先选择的频道上进行传输和接收(编号为 0~100,默认信道编号为 7,信道 0 的频率为 2400MHz,信道 1 的频率为 2401MHz,信道 2 的频率为 2402MHz,以此类推)。
- (4) 传输基于某特定功率,功率越大范围越广(值为 0~7,默认为 6)。
- (5) 信息通过地址(类似房屋号码)和组(类似指定地址处的指定收件人)进行过滤。
- (6) 发送和接收字节以处理任意数据。

【例 5.1】 发送信息。

编写发射端程序,代码如下:

```
from microbit import *
import radio

while True:
    radio.on()
    message="Hello,World!"
    radio.send(message)
    sleep(500)
```

【例 5.2】 接收信息。

编写接收端程序,代码如下:

```
from microbit import *
import radio

radio.on()

while True:
    incoming=radio.receive()
    if incoming is not None:
        display.show(incoming)
        sleep(500)
```

**小贴士**

其中,import radio 用于引用 radio;radio.on()用于打开无线电;radio.off()用于关闭无线电;radio.send()用于发送消息字符串;radio.receive()用于接收消息字符串。

将例 5.2 程序下载到第一块 micro:bit 板卡中,LED 点阵没有任何信息显示。

将例 5.1 程序下载到第二块 micro:bit 板卡中,看到第一块 micro:bit 板卡的 LED 点阵显示“Hello,World!”,如图 5.2 所示。

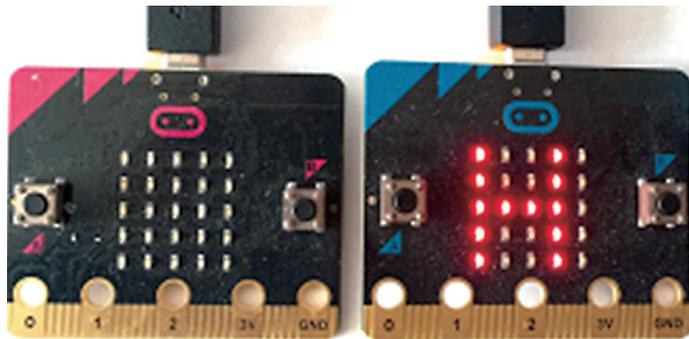


图 5.2 无线通信



复习思考题

- (1) 如何发送一个 0~9 的随机数并显示出来?
- (2) 如何通过按钮来实现发送、接收的控制?

【例 5.3】莫尔斯码。

电报(telegram)是通信业务的一种,是最早使用电进行通信的方法,它是 19 世纪 30 年代在英国和美国发展起来的。

19 世纪 30 年代,由于铁路迅速发展,迫切需要一种不受天气影响和时间限制,又比火车跑得快的通信工具。此时,发明电报的基本技术条件(电池、铜线、电磁感应器)也已具备。1837 年,英国人库克和惠斯通设计制造了第一个有线电报,经过不断改进,使发报速度不断提高。这种电报很快在铁路通信中获得了应用。这种电报系统的特点是电文直接指向字母。

电报是利用电流(有线)或电磁波(无线)作为载体,通过编码和相应的电处理技术,实现人类远距离传输与交换信息的一种通信方式。

电报信息通过专用的交换线路以电信号的方式进行发送,该信号用编码代替文字和数字,通常使用的编码是摩尔斯编码。随着电话、传真等的普及,目前电报已很少使用。

电报大大加快了信息的流通,是工业社会的一项重要发明。早期的电报只能在陆地上通信,后来使用了海底电缆,开通了越洋服务。到了 20 世纪初,开始使用无线电拍发电报,电报业务基本上已能抵达地球上绝大部分地区。电报主要用作传递文字信息,使用电报技术传送图片的技术称为传真。

虽然早在 19 世纪初就有人开始研制电报,但可实用的电磁电报的发明,主要归功于英国科学家约翰·库克、惠斯通和美国科学家莫尔斯。1836 年,约翰·库克制成电磁电报机并于次年申请了首个电报专利,惠斯通则是约翰·库克的合作者。莫尔斯原本是美国的一流画家,出于兴趣,他在 1835 年研制出电磁电报机的样机,后又根据电流接通、断掉时分别出现电火花和没有电火花两种信号,于 1838 年发明了“莫尔斯码”。



小贴士

1858 年 7 月, *Scientific American* 杂志报道: 众所周知, 英国人一向宣称, 电磁式电报(magnetic telegraph)是由他们的同胞惠斯通教授发明的。而在大西洋彼岸, 电报公司的成立, 则让更多的欧洲人开始讨论, 谁才是电报的真正发明者。在法国巴黎发行的《通报》认为, 莫尔斯虽不是电报原理的创立者, 却是第一个将该原理用于实践的人。

莫尔斯码在早期无线电上举足轻重, 是每个无线电通信者必须掌握的。由于通信技术的进步, 各国已于 1999 年停止使用了莫尔斯码。

莫尔斯码由两种基本信号和不同的间隔时间组成: 短促的点信号“.”和保持一定时间



的长信号“—”。表 5.1 是字母对应的莫尔斯码基本码表。

表 5.1 字母的莫尔斯码基本码表

字符	电码符号	字符	电码符号	字符	电码符号	字符	电码符号
A	.—	H	O	— — —	V	...—
B	—...	I	..	P	.— — .	W	.— —
C	— . — .	J	. — — —	Q	— — . —	X	— . —
D	— ..	K	— . —	R	. — .	Y	— . — —
E	.	L	. — ..	S	...	Z	— — ..
F	.. — .	M	— —	T	—		
G	— — .	N	— .	U	.. —		

通过 micro:bit 实现莫尔斯码发送与接收的代码如下：

```

from microbit import *
import radio

dash=Image("00000:00000:99999:00000:00000")
dot=Image("00000:00000:00900:00000:00000")
word=Image("00900:00090:99999:00090:00900")
over=Image.HAPPY

radio.on()

while True:
    incoming=radio.receive()
    gesture=accelerometer.current_gesture()

    if button_a.is_pressed():
        display.show(dot)
        radio.send(str("dot"))
    elif button_b.is_pressed():
        display.show(dash)
        radio.send(str("dash"))
    elif gesture=="shake":
        display.show(word)
        radio.send(str("word"))
    elif gesture=="face down":
        display.show(over)
        radio.send(str("over"))
    elif incoming=="dot":
        display.show(dot)

```



```
elif incoming=="dash":
    display.show(dash)
elif incoming=="over":
    display.show(over)
elif incoming=="word":
    display.show(word)
sleep(200)
display.clear()
```

将程序下载到第二块板卡上,按下第一块板卡的按钮 A 发送“.”,按下按钮 B 发送“—”,摇晃 micro:bit 表示完成一个字符的编码,开始下一个字符,将 micro:bit 正面朝下表示完成整个电报的发送。



复习思考题

完成某个字符莫尔斯码的发送后(摇晃 micro:bit),如何实现在第二块板卡上显示莫尔斯码对应的字符?

5.1.2 一对多通信

以上是两块板卡之间的一对一通信,下面是 3 块板卡之间的一对多通信。

【例 5.4】 一对多通信。

编写程序,代码如下:

```
from microbit import *
import radio

radio.config(group=1)
radio.on()

while True:
    if button_a.was_pressed():
        radio.send('Hello from A!')
    if button_b.was_pressed():
        radio.config(group=2)
        display.scroll('Switching to Group 2')
    message=radio.receive()
    if message !=None:
        display.scroll(str(message))
```



小贴士

radio.config()用于配置与无线电相关的各种基于关键字的设置,包括用于过滤信息时使用的 group(范围为 0~255,默认值为 0),如果没有调用 config,则使用默认值。

`radio.config(group=1)` 用于将两块板卡设置为同一组 group 1, 同组之间可以进行通信。

按下任何一块板卡上的按钮 A, 发送信息 Hello from A!, 在其他板卡上显示。

按下任何一块板卡上的按钮 B, 切换到 group 2, 非同组则之间无法进行通信。

如果再次按下另一块板卡的按钮 B, 它们又都属于 group 2, 相互之间又可以进行通信了。

运行程序, 验证效果。

复习思考题

(1) 修改程序, 实现按下第二块板卡的按钮 A 时, 在第一块板卡上显示“Hello from B!”。

(2) 修改程序, 实现按下第三块板卡的按钮 A 时, 在其他两块板卡上显示“Hello from C!”。

程序运行时, 3 块板卡 A、B、C 都属于 group 1, 按下板卡 A 的按钮 A, 板卡 B 和 C 均显示信息“Hello from A!”。

如果按下某一块板卡(如板卡 C)的按钮 B, 只有其他两块属于 group 1 的板卡 A 和 B 相互之间可以进行通信, 如图 5.3 所示。

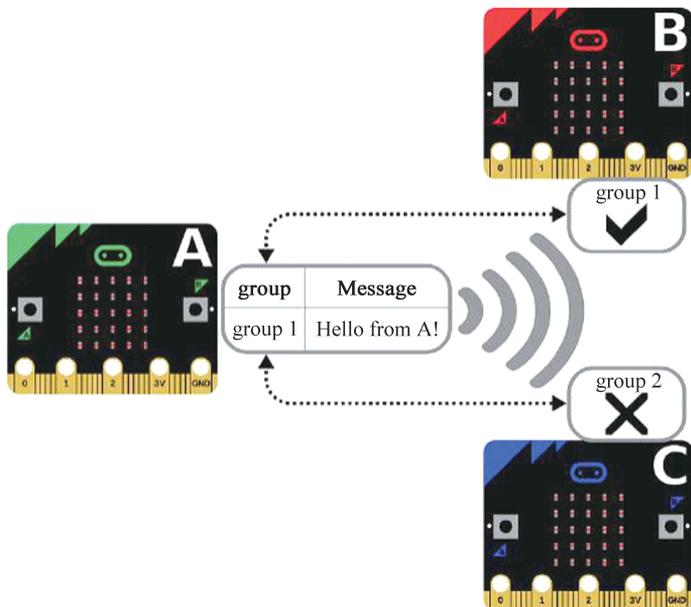


图 5.3 group 1 的板卡 A 和 B 可以相互通信



如果再按下另一块板卡(如板卡 A)的按钮 B,则按下按钮 B 的同属于 group 2 的两块板卡 A 和 C 之间可以相互通信,如图 5.4 所示。

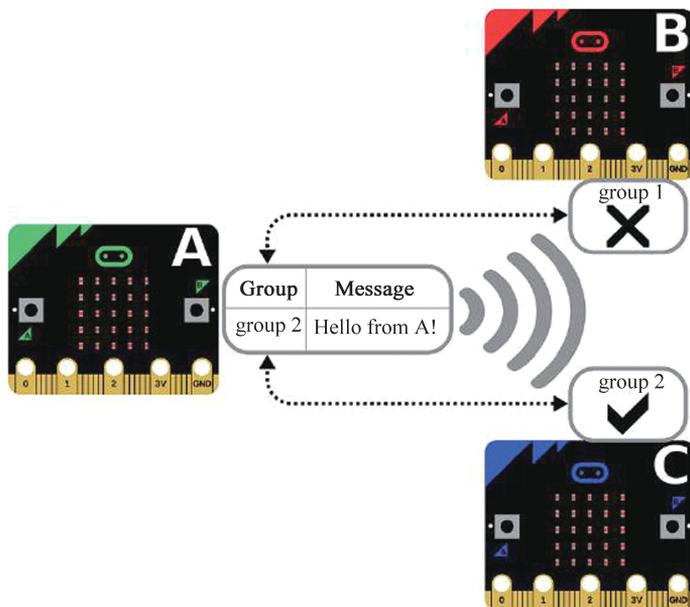


图 5.4 group 2 的板卡 A 和 C 可以相互通信

如果按下第三块板卡(板卡 B)的按钮 B,则它们都属于 group 2,3 块板卡之间又可以进行通信,如图 5.5 所示。

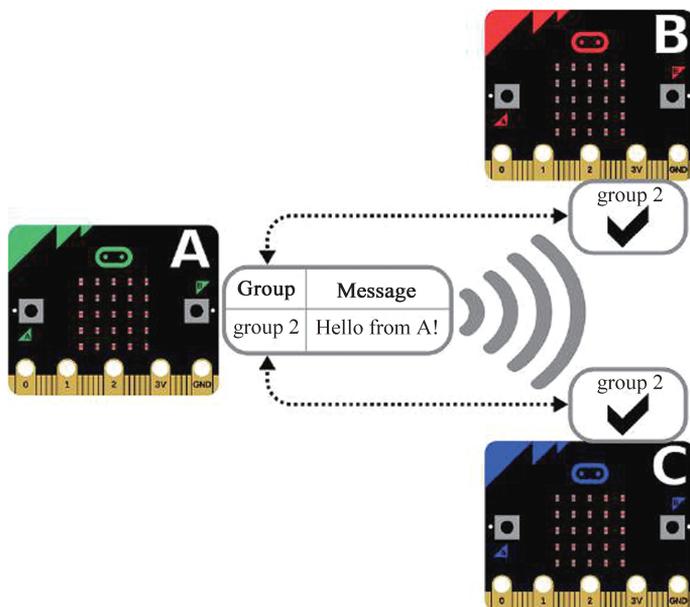


图 5.5 group 2 的 3 块板卡可以相互通信

【例 5.5】 “萤火虫”。

编写程序,代码如下:

```
from microbit import *
import radio
import random

flash=[Image().invert() * (i/9) for i in range(9, -1, -1)]

radio.on()

while True:
    if button_a.was_pressed():
        radio.send('flash')
    incoming=radio.receive()
    if incoming=='flash':
        sleep(random.randint(50, 350))
        display.show(flash, delay=100, wait=False)
        if random.randint(0, 9)==0:
            sleep(500)
            radio.send('flash')
```

小贴士

Image().invert()用于通过反转源图像中像素的亮度来产生新图像。
flash=[Image().invert() * (i/9) for i in range(9, -1, -1)]用于实现动画效果。
sleep(random.randint(50,350))用于随机进行短时间的休息(多块 micro:bit 错开时间,“萤火虫”显示效果更佳)。
if random.randint(0,9)==0 用于显示动画的每块 micro:bit 有 1/10 的概率将动画传递给其他 micro:bit,这就使得“萤火虫”在多个设备间闪光成为可能。

运行程序,9 块 micro:bit 的运行效果如图 5.6 所示。

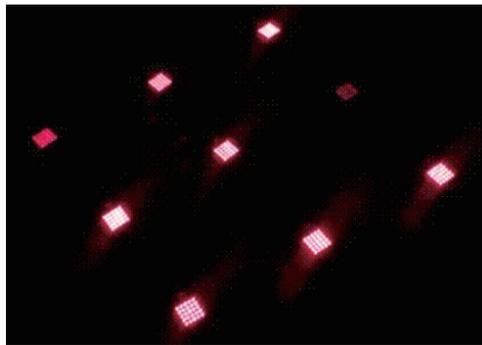


图 5.6 9 块 micro:bit 的运行效果



5.2 实践：简易 POS 机

科技发展日新月异,技术不断升级,给人们的生活带来了翻天覆地的变化,也让生活越来越方便。

在付款方面,古代都是以物换物,之后出现了作为等价物的金属币,再之后更加轻便的纸币慢慢出现。现在,人们出门基本不需要带现金,需要付款的时候,直接刷卡或者使用微信、支付宝等进行电子支付即可。

电子支付具有方便、快捷、高效、经济的优势,用户可以随时随地完成整个支付过程,支费用仅相当于传统支付的几十分之一,甚至几百分之一。

电子支付的发展经历了 5 个阶段。

第 1 阶段是银行利用计算机处理银行之间的业务,办理结算。

第 2 阶段是银行计算机与其他机构计算机之间资金的结算,如代发工资等业务。

第 3 阶段是利用网络终端向客户提供各项银行服务,如自助银行。

第 4 阶段是利用银行销售终端向客户提供自动的扣款服务。

第 5 阶段是基于 Internet 的电子支付,它将第 4 阶段的电子支付系统与 Internet 进行整合,实现随时随地通过 Internet 进行直接转账结算,形成了电子商务交易支付平台。

下面,模拟制作一个简单的电子支付系统——简易 POS 机。

首先,需要选择扣款的值,因为 micro:bit 主控板只有两个按钮,所以没有办法像键盘这样直接输入。这里选择使用变量:按钮 A 每按下一次,变量的值加 1,到需要扣款的数字(假设扣款的数在 10 以内);按下按钮 B,把扣款的值发送出去。

【例 5.6】 发送需要扣款的值。

编写程序,代码如下:

```
from microbit import *
import radio

radio.on()
radio.config(channel=1)
shu=0

while True:
    display.show(shu)
    if(button_a.was_pressed()):
        shu=shu+1
        if shu==10:
            shu=0

    if (button_b.was_pressed()):
        message=str(shu)
        radio.send(message)
```