Bluetooth Remote Control for the Lego Droid Developer Kit

Created by Kevin Walters

https://learn.adafruit.com/bluetooth-remote-for-lego-droid

Last updated on 2024-06-03 02:33:14 PM EDT

Table of Contents

Overview

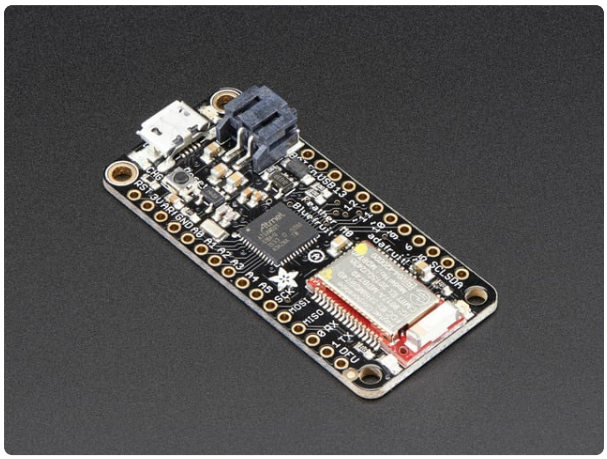
A short time ago in a very nearby galaxy (in 1999)...

Lego® added the Droid Developer Kit (https://adafru.it/Di4) to the Mindstorms (https://adafru.it/Di5) robotic range. This included the Micro Scout brick, a simplified version of the earlier Scout brick with a single, bi-directional motor and a light sensor. The Micro Scout functionality includes control via the Visible Light Link (VLL) protocol.

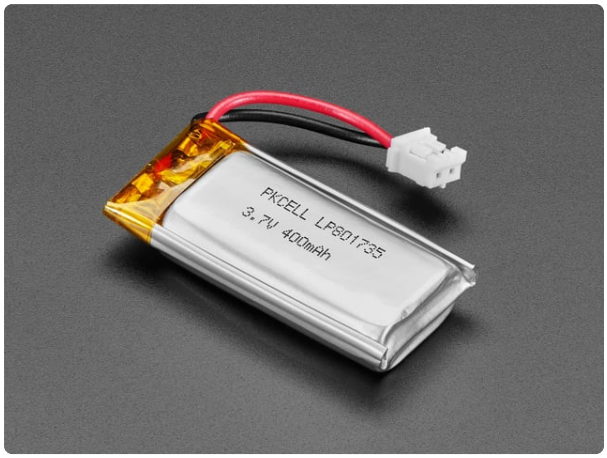
This project revitalises the Droid kit with a Bluetooth Low Energy (LE) (https://adafru.it/Di6) remote control for the Micro Scout brick. A Feather M0 Bluefruit LE (http://adafru.it/2995) is used to receive commands from the Adafruit Bluefruit LE Connect app and send corresponding VLL commands to the brick using an LED. The app is available on both Android (https://adafru.it/f4G) and iOS (https://adafru.it/BYj).

The code is implemented in CircuitPython (https://adafru.it/CgS).

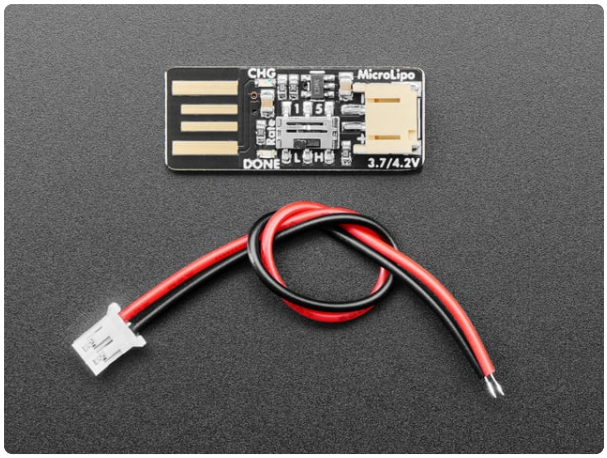
Parts



Adafruit Feather M0 Bluefruit LE
Feather is the new development board from Adafruit, and like its namesake, it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller...
https://www.adafruit.com/product/2995



Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh
Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...
https://www.adafruit.com/product/3898



Adafruit Micro Lipo - USB LiIon/LiPoly charger
Oh so adorable, this is the tiniest little lipo charger, so handy you can keep it any project box! Its also easy to use. Simply plug in the gold plated contacts into any USB port and a...
https://www.adafruit.com/product/1304

1 x White LED
Bright white 5mm LED with narrow beam. Any handy, bright LED is worth testing as a substitute.

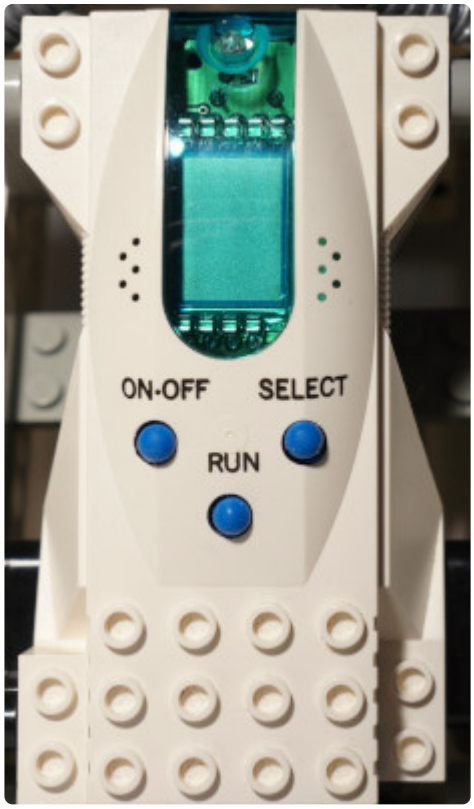
https://www.mouser.com/ProductDetail/Optek-TT-Electronics/OVLEW1CB9?qs=sGAEpiMZZMtmwHDZQCdlqbkQSv8HD28Qn7If7

1 x 330 ohm resistor
330 ohm resistor 1/4W 5%.

https://www.mouser.com/ProductDetail/Yageo/CFR-25JR-52-330R?qs=sGAEpiMZZMsPqMdJzcrNwiPCnpFTGbbhmZU2%

Lego Micro Scout

The Micro Scout (https://adafru.it/Di7) brick is included as part of two kits, the Lego Droid Developer Kit (9748) (https://adafru.it/Di4) and the Dark Side Developer Kit (9754). (https://adafru.it/Di8) The only difference is the colour of the brick: white in the Droid kit, grey in the Dark Side kit.



This electronic brick was intended to be programmed by the Lego Scout brick (https://adafru.it/Di9) or Spybot brick (https://adafru.it/Dia) but anything which can generate the low speed light pulses of the Visible Light Link (VLL) protocol can be used. The RCX brick (https://adafru.it/Dib) can also be used to program it. The Micro Scout has a **P** mode to process VLL commands - the **select** button cycles through the modes.

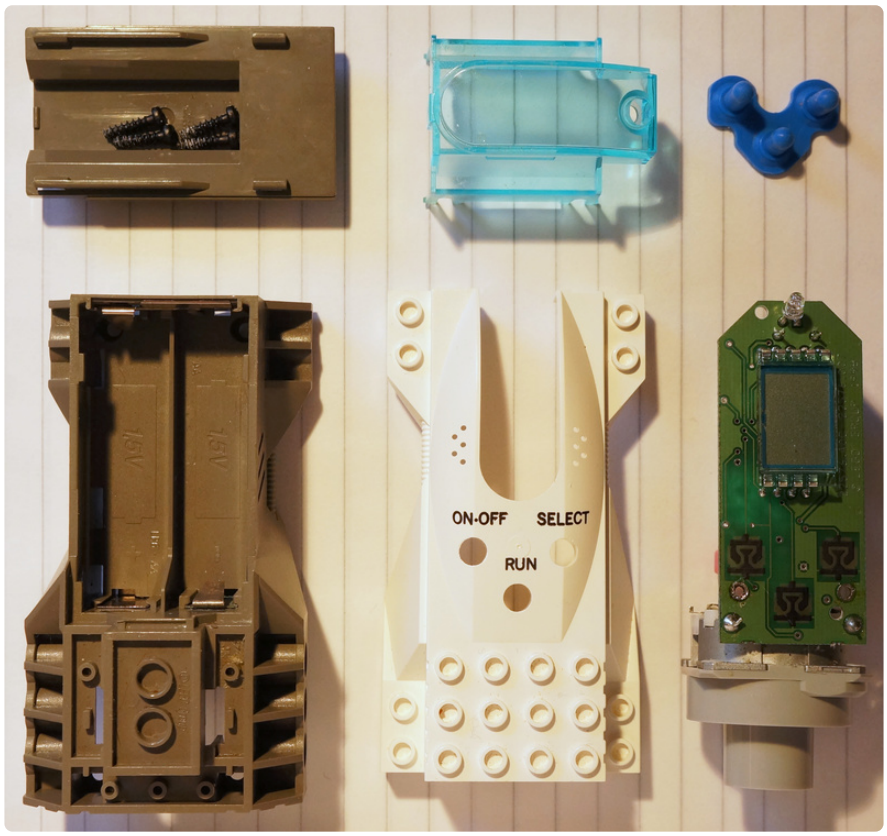
The Micro Scout is powered by 2 AA batteries. If you are digging one out of the attic or buying one second-hand from Jawas then it's important to check that there have been no battery leaks. A caustic leak can damage the printed circuit board (https://adafru.it/D5E) (PCB) particularly if the Micro Scout has not been stored face-up. The original manual contains some sage advice:

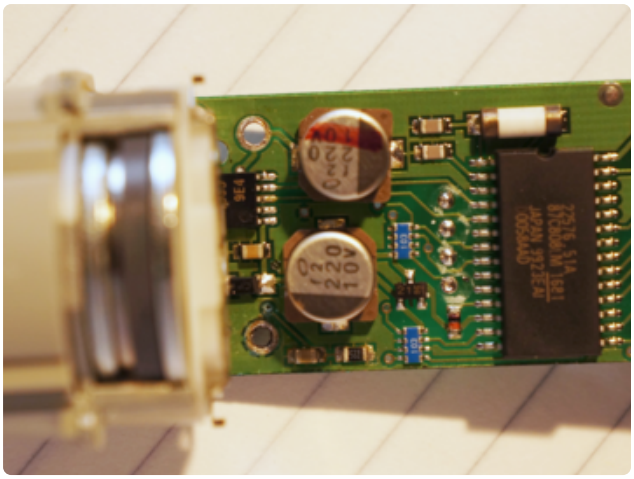
Always remove the batteries from the battery box for long-term storage or if they have reached the end of their life. Liquid leaking from dead batteries will damage the battery box.

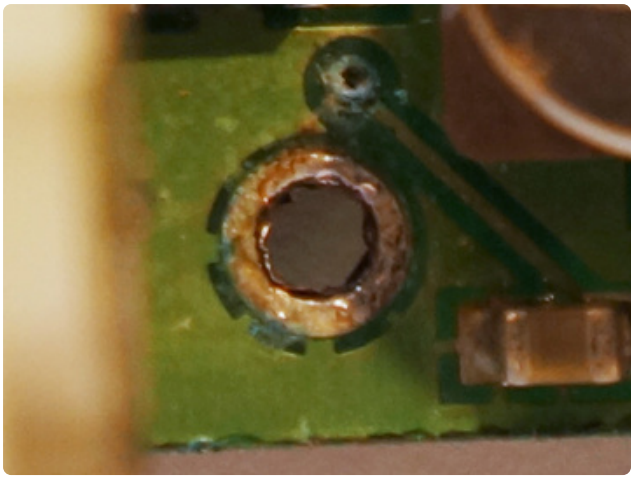
Inside the Micro Scout

If you need to fix a Micro Scout brick this section may help.

There are four screws which are easily unscrewed to open up the brick. The blue screen clips into the battery compartment and is easy to unclip. The circuit board is more difficult to remove as there are two terminals which need to be desoldered.



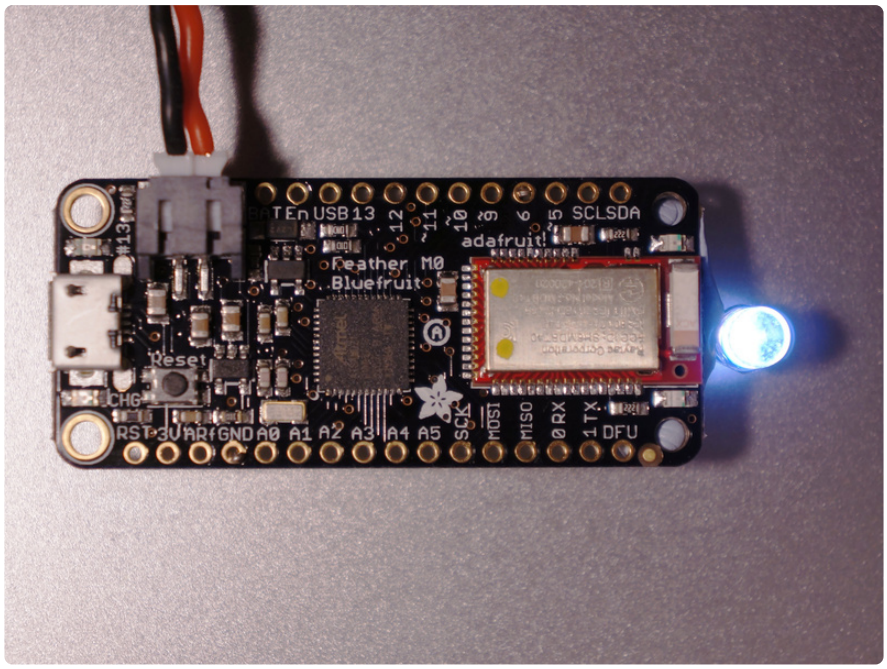




This shows the PCB which has been cleaned to remove the alkaline deposits from a substantial battery leak at the negative terminal. The through-hole plating is damaged and partially missing leading to poor connectivity for the ground. This damage caused the Micro Scout to work intermittently.

There are a few options to repair this, the easiest approach for this particular case is to scrape away some lacquer and solder a small wire (not shown) from one side of the PCB to the other to ensure the negative terminal connects to the ground and ground plane on the other side of the PCB. Inelegant but effective.

Feather M0 Bluefruit LE

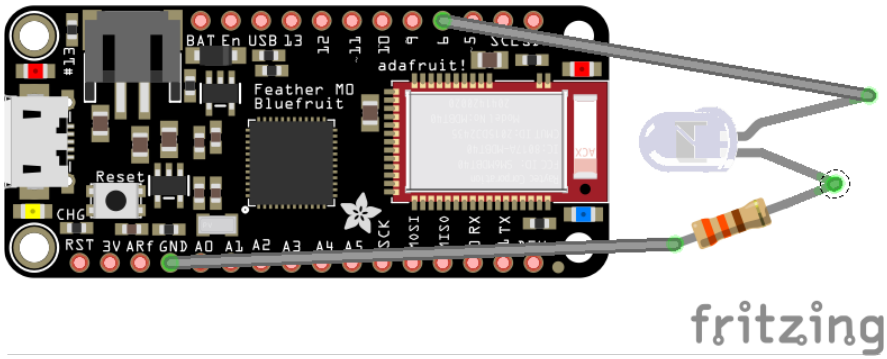


The Feather M0 Bluefruit LE (http://adafru.it/2995) is a compact board which in essence is a Feather M0 Basic Proto (http://adafru.it/2772) with an on-board Bluefruit LE SPI Friend (http://adafru.it/2633). It's small and has a Lithium Polymer (LiPo) battery charger making it ideal for this project.

Connecting an LED

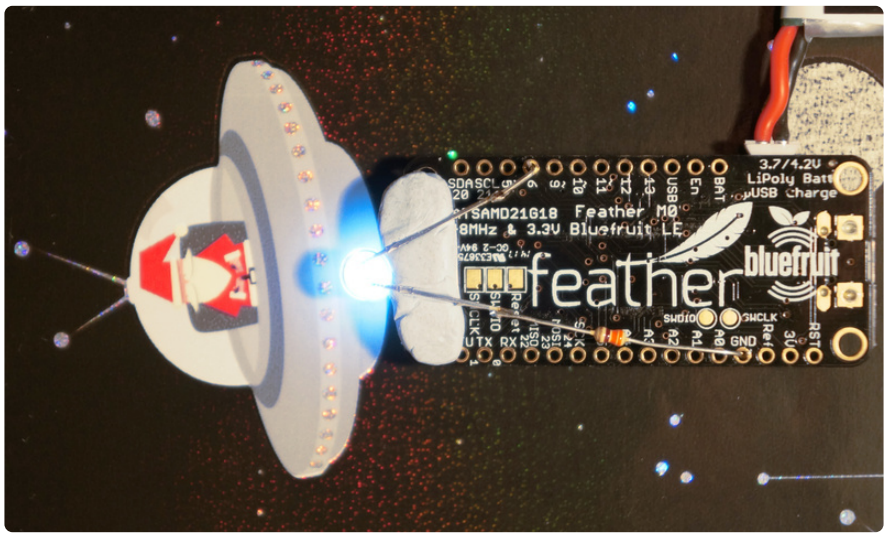
The VLL protocol requires a light source to transmit the pulses to the Micro Scout. Surprisingly, the Feather's small, on-board, red LED can be used in dim ambient light. It struggles in brighter surroundings so an external LED is a better option.

For the project, the wires were routed beneath the board.



The diagram above and the picture below show a brighter, white LED with a 330 ohm resistor (https://adafru.it/Dic) connected between **D6** and **GND**. In this case the resistor has been soldered (https://adafru.it/Did) to the LED and the other lead has been extended a little with the offcut from the resistor. This could be done by twisting the wires together tightly for a quick, temporary, solderless prototype. The leads are bent back at the ends to double them up to make them the fit and grip the holes removing the need for soldering to the Feather.

Care needs to be taken to ensure the wires stay clear of the other connections on the Feather board. The Blu Tack (https://adafru.it/Die) used here is holding the LED in place and also keeping the wires clear above the board. A multimeter can be used to confirm that similar putty/tak products are non-conductive.



The white LED is showing up as blue partly due to the photograph being colour-balanced for the warm ambient light.

Be sure if you wire your LED this way, the bare wires do not touch any other copper pads on the back of the Feather.

CircuitPython

If you are new to CircuitPython, see Welcome to CircuitPython! (https://adafru.it/cpy-welcome)

Adafruit suggests using the Mu editor to edit your code and have an interactive REPL in CircuitPython. You can learn about Mu and its installation in this tutorial (https://adafru.it/ANO).

The code should also work on any CircuitPython compatible board using an on-board or separate SPI bus connected Bluefruit friend board.

Installing CircuitPython on Feather M0 Bluefruit LE

For this project, the Feather M0 Adalogger variant of the firmware must be used as it defines **board.D8** which is used by the code.

The non-express Feather M0 boards did not originally ship with the UF2 bootloader. If **CIRCUITPY** drive disappears but no **FEATHERBOOT** appears when reset is double-clicked:

- then check CircuitPython Troubleshooting (https://adafru.it/Den) first and if **FEATHERBOOT** is still missing then use CircuitPython Non-UF2 Installation (https://adafru.it/Bed),
- else use Installing CircuitPython (UF2) (https://adafru.it/Amd).

Ensure you have the **CIRCUITPY** drive appear when you plug your Feather into your computer via a known good USB cable. The version of CircuitPython you are running can be seen either in the Mu editor Serial/REPL or by the text in the file **boot_out.txt** on the **CIRCUITPY** drive.

Libraries

Download the latest set of libraries for CircuitPython to match the version of CircuitPython you are running. There is one library package for CircuitPython 3.x, 4.x, and so on. Click the box below and download the library bundle to your computer.

Click to go to the latest Adafruit CircuitPython Library Bundle Release Page

https://adafru.it/y8E

Two libraries are needed for the code. Open the library Zip file and copy the following files onto the Feather **CIRCUITPY** drive in a directory called **/lib**

- **adafruit_bus_device**
- **adafruit_bluefruitspi**

See the CircuitPython Libraries (https://adafru.it/Cqa) guide for additional details on how to add libraries.

Code

Adafruit recommends copying this file to the board using the target filename **code.py** on the Feather M0 Bluefruit LE.

```
# SPDX-FileCopyrightText: 2018 John Edgar Park for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import busio
from digitalio import DigitalInOut, Direction
from adafruit_bluefruitspi import BluefruitSPI

# Setup SPI bus and 3 control pins for Nordic nRF51822 based Raytec MDBT40
spi_bus = busio.SPI(board.SCK, MOSI=board.MOSI, MISO=board.MISO)
cs  = DigitalInOut(board.D8)
irq = DigitalInOut(board.D7)
rst = DigitalInOut(board.D4)
bluefruit = BluefruitSPI(spi_bus, cs, irq, rst, debug=False)

boardled = DigitalInOut(board.D6)
#boardled = DigitalInOut(board.D13) # small onboard red LED
boardled.direction = Direction.OUTPUT

def vllchksum(n):
    return 7-((n+(n>>2)+(n>>4))&7)

# Lego Micro Scout commands
MS_FWD  = 0
MS_REV  = 1
MS_STOP = 2
MS_BEEP = 4

PAUSE = 0.15

ADVERT_NAME = b'BlueMicroScout'

# Note: incompatibile with ZX Spectrum cursors
BUTTON_1     = 1
BUTTON_2     = 2
BUTTON_3     = 3
BUTTON_4     = 4
BUTTON_UP    = 5
BUTTON_DOWN  = 6
BUTTON_LEFT  = 7
BUTTON_RIGHT = 8

# From https://github.com/JorgePe/mindstorms-vll/blob/master/vll-atat.py
# https://github.com/JorgePe/mindstorms-vll
def vll1():
    boardled.value =  True
    time.sleep(0.02)
    boardled.value = False
    time.sleep(0.04)

def vll0():
    boardled.value = True
    time.sleep(0.04)
    boardled.value = False
    time.sleep(0.02)

def vllinit():
    boardled.value = True
    time.sleep(0.4)

def vllstart():
    boardled.value = False
    time.sleep(0.02)

def vllstop():
    boardled.value = True
```

```
    time.sleep(0.02)
    boardled.value = False
    time.sleep(0.06)
    boardled.value = True
    time.sleep(0.12)

def send(command):
    vllstart()
    v = (vllchksum(command) << 7 ) + command
    i = 0x200
    while i>0 :
        if v & i:
            vll1()
        else:
            vll0()
        i = i >> 1
    vllstop()

def pause():
    boardled.value = True
    time.sleep(PAUSE)

boardled.value = False

vllinit()

def init_bluefruit():
    # Initialize the device and perform a factory reset
    print("Initializing the Bluefruit LE SPI Friend module")
    bluefruit.init()
    bluefruit.command_check_OK(b'AT+FACTORYRESET', delay=1)
    # Print the response to 'ATI' (info request) as a string
    print(str(bluefruit.command_check_OK(b'ATI'), 'utf-8'))
    # Change advertised name
    bluefruit.command_check_OK(b'AT+GAPDEVNAME='+ADVERT_NAME)

def wait_for_connection():
    print("Waiting for a connection to Bluefruit LE Connect ...")
    # Wait for a connection ...
    dotcount = 0
    while not bluefruit.connected:
        print(".", end="")
        dotcount = (dotcount + 1) % 80
        if dotcount == 79:
            print("")
        time.sleep(0.5)

# This code will check the connection but only query the module if it has been
# at least 'n_sec' seconds. Otherwise it 'caches' the response, to keep from
# hogging the Bluefruit connection with constant queries
connection_timestamp = None
is_connected = None
def check_connection(n_sec):
    # pylint: disable=global-statement
    global connection_timestamp, is_connected
    if (not connection_timestamp) or (time.monotonic() - connection_timestamp >
n_sec):
        connection_timestamp = time.monotonic()
        is_connected = bluefruit.connected
    return is_connected

# Borrowed from MUNNY code

# Unlike most circuitpython code, this runs in two loops
# one outer loop manages reconnecting bluetooth if we lose connection
# then one inner loop for doing what we want when connected!
while True:
    # Initialize the module
    init_bluefruit()
```

```
    try:        # Wireless connections can have corrupt data or other runtime
failures
                # This try block will reset the module if that happens
        while True:
            # Once connected, check for incoming BLE UART data
            if check_connection(3):  # Check our connection status every 3 seconds
                # OK we're still connected, see if we have any data waiting
                resp = bluefruit.read_packet()
                if not resp:
                    continue  # nothin'
                print("Read packet", resp)
                # Look for a 'B' for Button packet
                if resp[0] != 'B':
                    continue
                button_num = resp[1]
                button_down = resp[2]
                # For now only look for the down events
                if button_down:
                    if button_num == BUTTON_UP:
                        send(MS_FWD)
                    elif button_num == BUTTON_DOWN:
                        send(MS_REV)
                    elif button_num == BUTTON_1:
                        send(MS_BEEP)
                    else:
                        # some other key pressed
                        pass
            else:  # Not connected
                pass

    except RuntimeError as e:
        print(e)  # Print what happened
        continue  # retry!
```

This code makes use of Jorge Pereira's VLL python code (https://adafru.it/Dif).

This code has been tested on a Feather M0 Bluefruit LE running CircuitPython 3.1.1.

Remote Control

Install the App

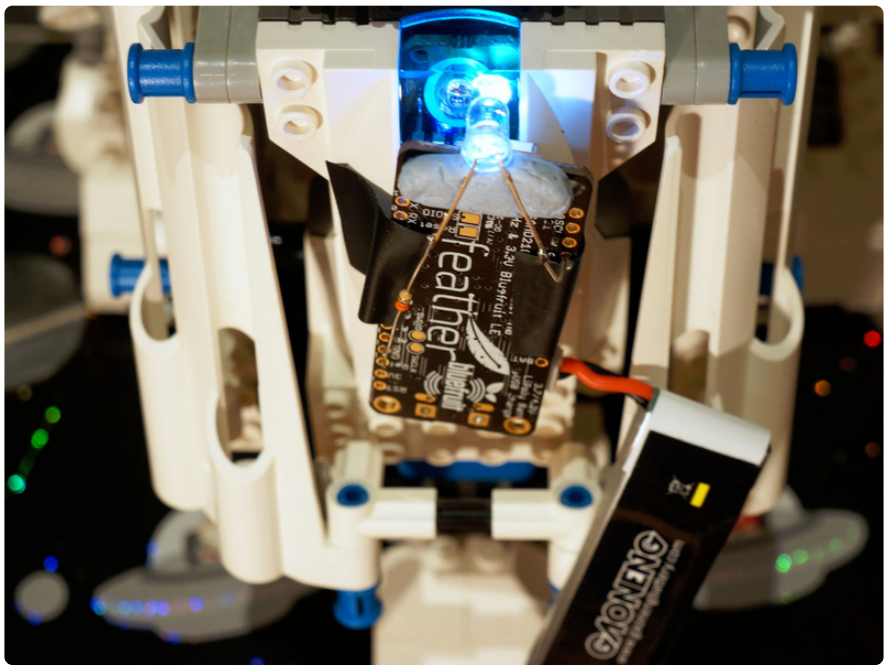
Download the Adafruit Bluefruit LE Connect app on either an Android (https://adafru.it/f4G) or iOS (https://adafru.it/BYj) device with Bluetooth.

Attach the Feather Board to the Droid

There are lots of different ways to attach the Feather board:

- extra Lego to hold the board in place,
- a custom-made 3d printed holder,
- two pieces of black insulating tape.

The last option was used with the battery hanging off precariously.



The LED must be aligned with the photodiode (https://adafru.it/Dig) light sensor. In bright conditions it may help to cover/shroud this area to stop ambient light interfering with the light pulse communication.

The LiPo battery shown in the pictures comes with the correct connector (JST PH) but the wrong polarity. The pins have been swapped around so the polarity matches the Adafruit board. This is best done with non-conductive tools to avoid the risk of shorting the battery. The recommended battery does NOT have this issue.

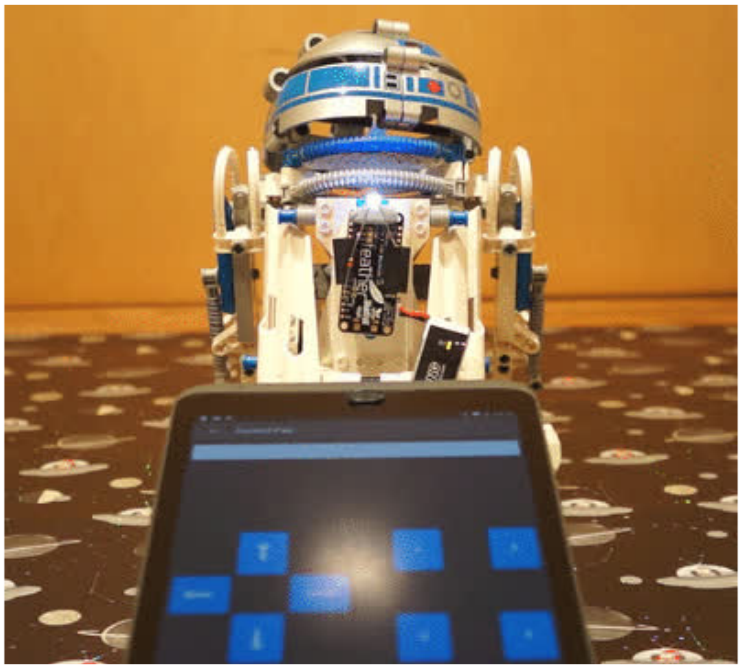
Droid Remote Control

Ensure the Micro Scout brick is in **P** mode, tap connect in the tablet for the **BlueMicroScout** and start sending commands to the droid:

Button -> Action

- Up -> Reverse
- Down -> Forward
- 1 -> Beep

It's possible the droid in this example is built incorrectly with the direction mechanically reversed. The intention of the code is for the direction to be the other way around!



If the droid starts misbehaving by following a different command, this is likely to be due to a reported bug (https://adafru.it/Dih).

Going Further

Ideas for Areas to Explore

- Implement "bump and turn" functionality with an accelerometer or an alternative board which contains an accelerometer like the (larger) Circuit Playground Express (http://adafru.it/3333).
- Add additional servos (https://adafru.it/Bei) to rotate the head and control the foot direction.
- Compare with the more advanced bluetooth remote control addition to the Hasbro R2-D2 by Chris Lydgate: The Resurrection of R2-D2 (https://adafru.it/EO-) (YouTube).

Related Projects

- MUNNY Glowing Friend with Bluetooth Control! (https://adafru.it/Dii)

Further Reading

- Introducing Adafruit Feather (https://adafru.it/Dij)
- All the Internet of Things - Epsiode One: Bluetooth & BTLE (https://adafru.it/Dik)
- Adafruit Bluetooth Low Energy (BLE) FAQ (https://adafru.it/Dil).