**Instalación de CircuitPython en el Feather M0 RFM9x**

Las placas Adafruit SAMD21 (M0) y SAMD51 (M4) cuentan con un cargador de arranque (bootloader) mejorado que hace que sea más fácil que nunca flashear código diferente en el microcontrolador. Este gestor de arranque facilita el cambio entre Microsoft MakeCode, CircuitPython y Arduino [3]. Un bootloader es un programa que se ejecuta en el microcontrolador y espera recibir un nuevo programa. Una vez que recibe ese programa, escribe el nuevo programa en la memoria del microcontrolador. Al reiniciar (RESET), el microcontrolador comenzará a ejecutar el nuevo programa en lugar del bootloader (a menos que se le indique específicamente al microcontrolador que ejecute el bootloader en su lugar, por ejemplo en el modo BOOT) [2].

Casi todas las placas CircuitPython se distribuyen con un bootloader llamado UF2 (USB Flasher versión 2) que hace que la instalación y actualización de CircuitPython sea un proceso rápido y fácil. El bootloader es el modo en el que debe estar la placa para que funcione el archivo .uf2 de CircuitPython que se va a descargar. En el caso del Feather M0 RFM9x es necesario instalar el UF2 bootloader primero y luego CircuitPython [1].

El bootloader (UF2) también es compatible con BOSSA, por lo que se puede utilizar con el IDE de Arduino que espera un cargador de arranque BOSSA en placas basadas en ATSAMD. Esto significa que se puede retornar a Arduino IDE luego de haber instalado CircuitPython!

**Primera etapa:** Actualizar el bootloader del Feather M0 RFM9x

El Feather M0 RFM9x por defecto viene para ser utilizado mediante Arduino. La estrategia es utilizar Arduino IDE para cargar el sketch que contiene las direcciones e instrucciones del bootloader para el microcontrolador.

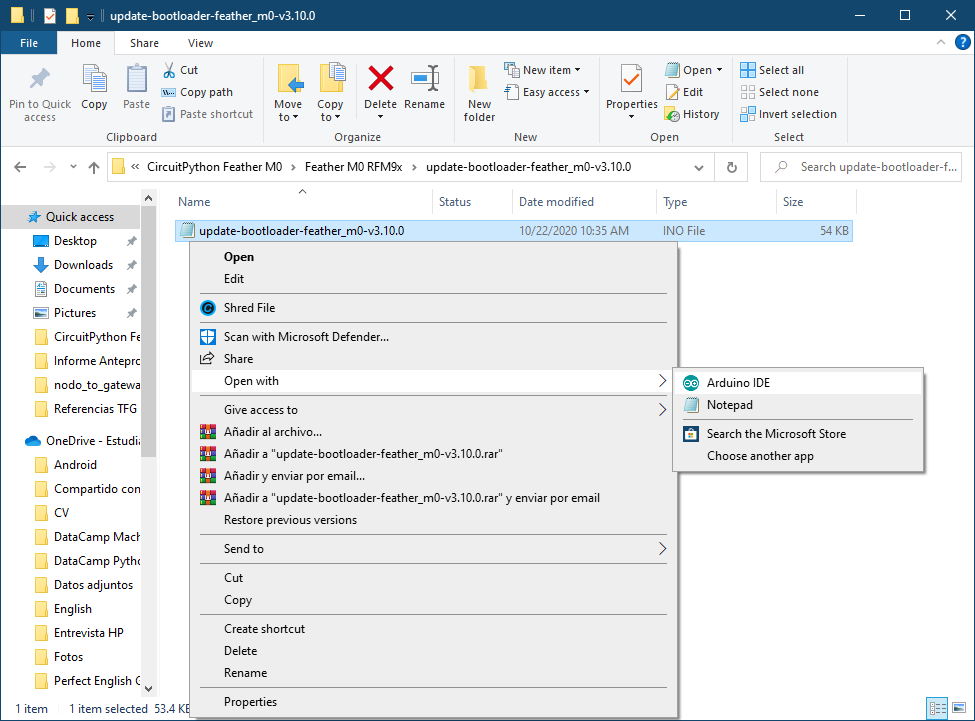
**1.** Descargar el archivo ***update-bootloader-feather\_m0-v3.x.x.ino*** más reciente. Luego abrirlo con Arduino IDE. Obsérvese que el archivo tiene extensión **.ino** como cualquier sketch de Arduino, no confundirlo con el archivo de extensión **.uf2** que aparece a la par de forma similar a la figura 1. El archivo se debe buscar y descargar en el siguiente enlace:

<https://github.com/adafruit/uf2-samdx1/releases>



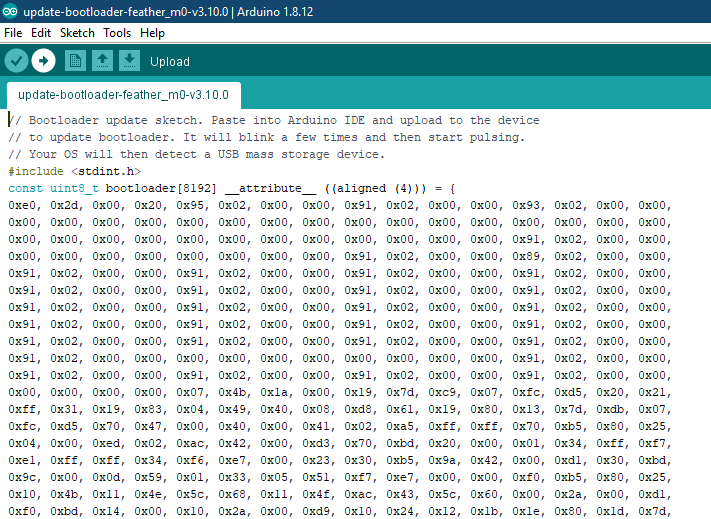
*Figura 1. Lista de bootloader’s para diferentes placas de la marca Adafruit.*

La versión descargada del actualizador en este caso fue ***update-bootloader-feather\_m0-v3.10.0.ino***, tal y como se muestra en la siguiente figura:

****

*Figura 2. Apertura con Arduino IDE del archivo de actualización del bootloader.*

Una vez abierto el archivo mediante Arduino IDE, se visualiza un montón de direcciones hexadecimales y otros parámetros.

****

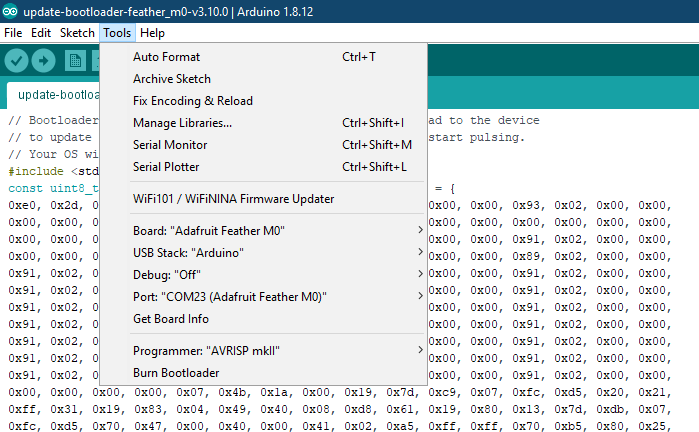
*Figura 3. Contenido del archivo de actualización del bootloader.*

**2.** Ahora se debe seleccionar la board y el puerto correspondiente al Feather M0 al que se le desea instalar el UF2 bootloader. La board aparecerá solo si ya se han instalado las siguientes librerías de Arduino para el microcontrolador ATSAMD21:

* SAMD Support
* Adafruit SAMD

Ver la guía: <https://learn.adafruit.com/adafruit-feather-m0-radio-with-lora-radio-module/using-with-arduino-ide>

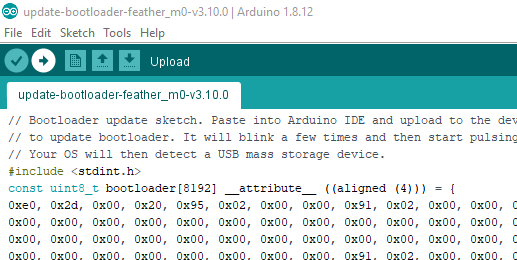
Si ya estas librerías estaban incorporadas en Arduino IDE, entonces en las opciones Board y Port debe aparecer “Adafruit Feather M0”, como se ve en la figura 3.

****

*Figura 4. Asignación de la placa y puerto correspondiente.*

Una vez seleccionada la board y el puerto correspondiente, hacer click en Upload como se ve a continuación. Con esto se comienza a flashear el UF2 bootloader en el microcontrolador ATSAMD21G18 ARM Cortex M0 del Feather M0.

**3.** Cargar el archivo al Feather M0, dar click en Upload. Luego de esto se verá como el LED indicador del Feather comienza a parpadear varias veces. Incluso en Windows el sistema muestra una ventana de error de que no se reconoce la unidad de almacenamiento detectada, es normal.

****

Al aceptar esta ventana de error en Windows, debería mostrarse una nueva unidad de almacenamiento o disco en “Este Equipo”. Si no aparece nada, intente reiniciar el Feather en modo BOOT, para ello pulse rápidamente 2 veces el botón RESET.

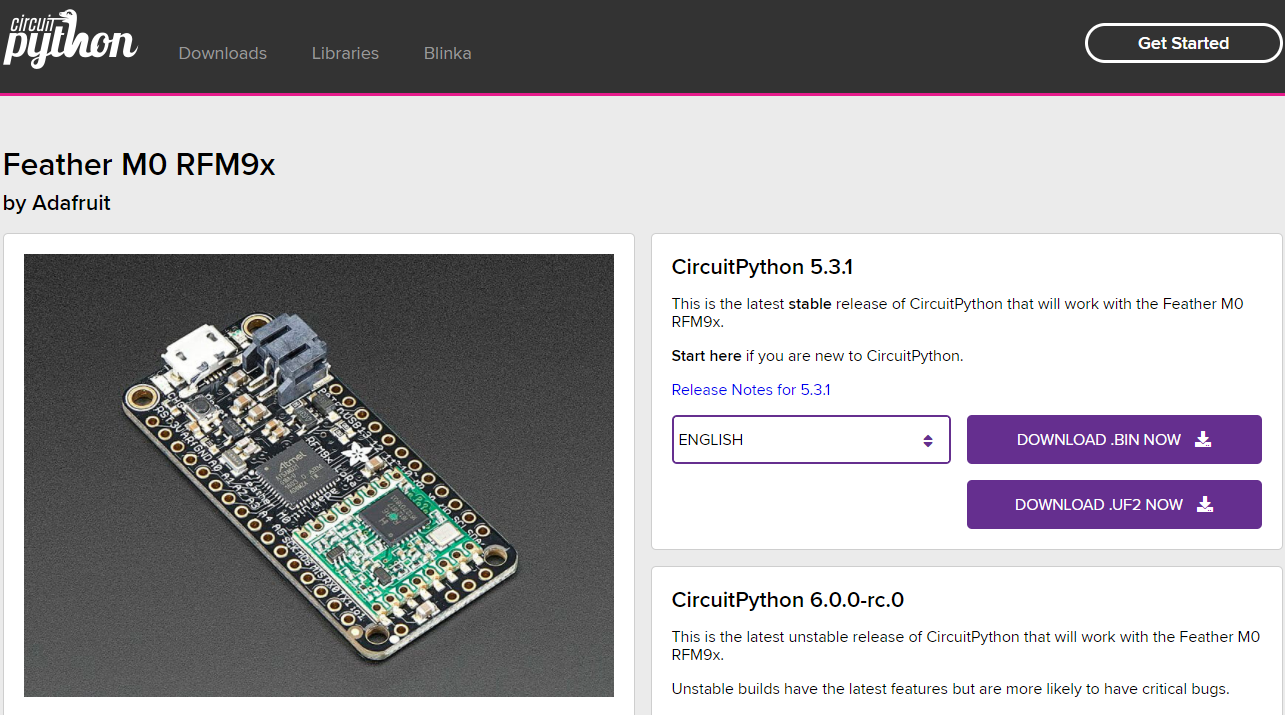
**Segunda etapa:** Instalar CircuitPython en el Feather M0 RFM9x

La versión estable de CircuitPython más reciente a este momento es la versión 5.3.1. Esta es la versión utilizada en este caso, mediante al archivo ***adafruit-circuitpython-feather\_m0\_rfm9x-en\_US-5.3.1.uf2***.

El formato del archivo (UF2) es un poco especial, tiene información adicional para ayudar al bootloader a saber dónde van los datos.

Se puede descargar el archivo **.uf2** en la siguiente página:

<https://circuitpython.org/board/feather_m0_rfm9x/>



*Figura 5. Página web de descarga del archivo instalador de CircuitPython*

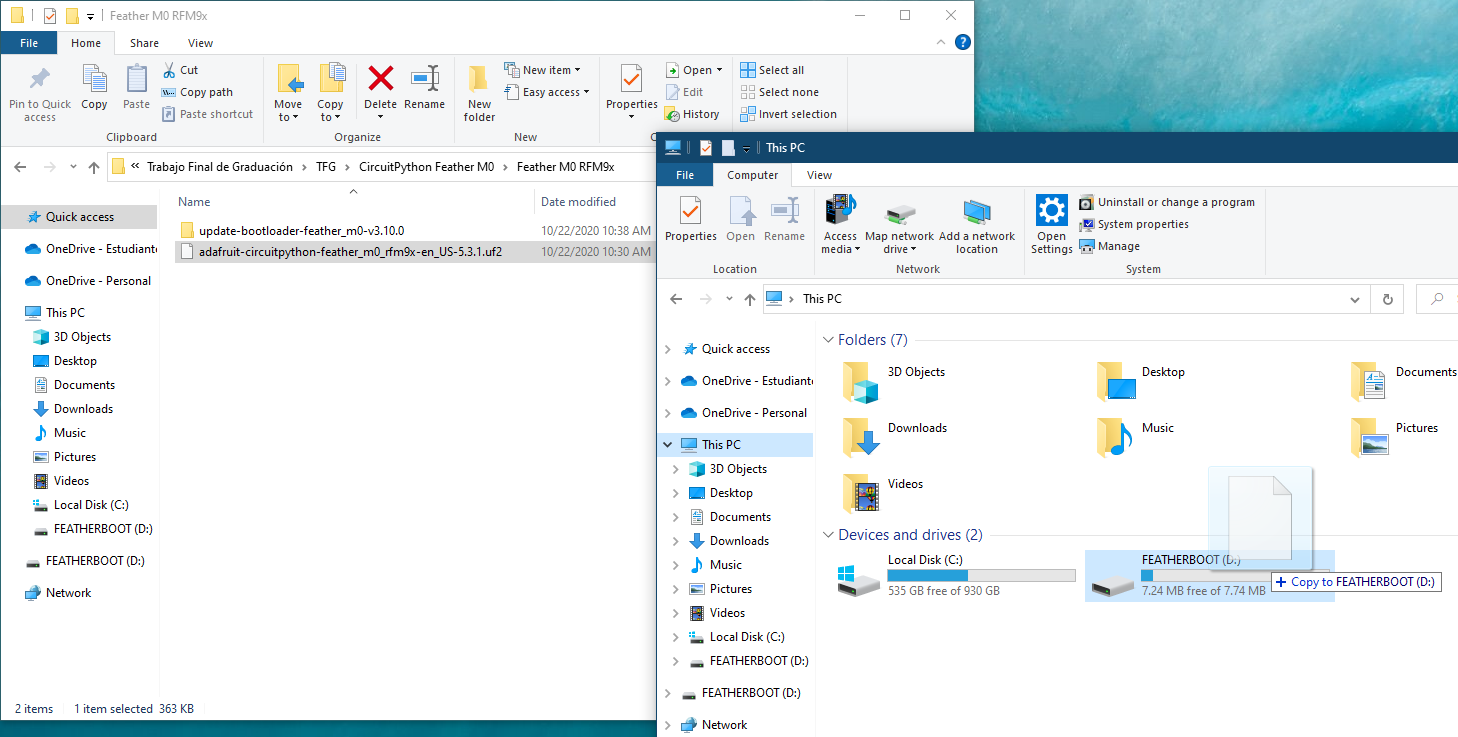
*para el Feather M0 RFM9x.*

El proceso de instalación de CircuitPython se hace solo una vez, luego solo se cargan los archivos **.py** de código dentro del directorio y listo (si están instaladas las librerías necesarias). Para el Feather M0 se recomienda no instalar todas las librerías de CircuitPython (el bundle completo), esto porque no tiene el espacio suficiente para alojar todo, por ello solo se recomienda instalar las librerías específicamente necesarias para la ejecución del código deseado.

El bundle de CircuitPython para la versión correspondiente se puede descargar de:

<https://circuitpython.org/libraries>

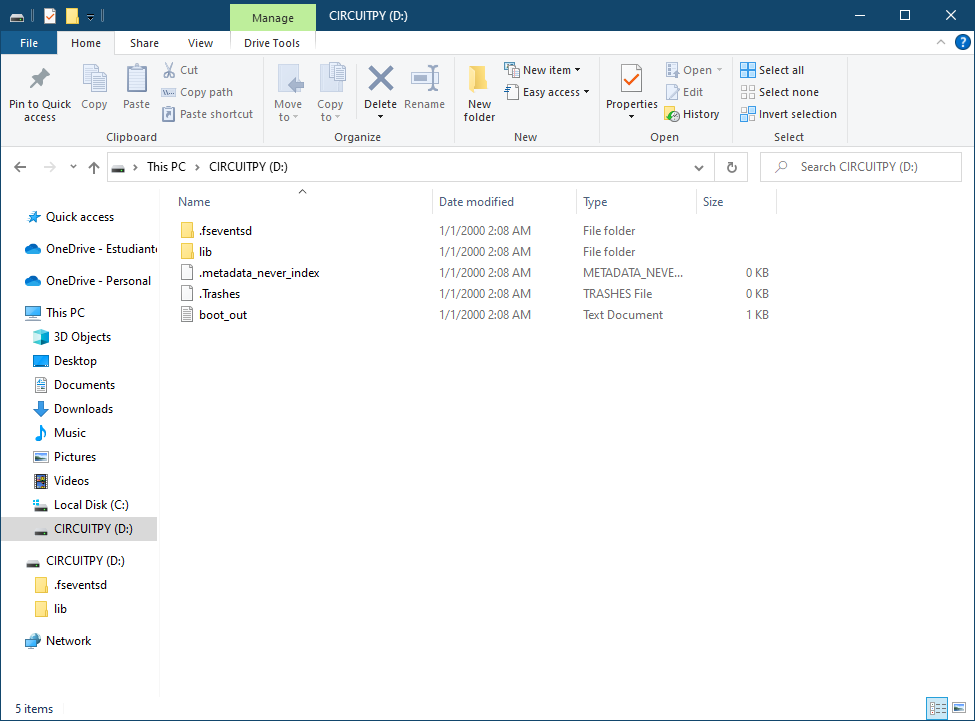
**4.** Arrastrar (mover o copiar y pegar) el archivo **.uf2** instalador de CircuitPython en el directorio FEATHERBOOT que apareció tras actualizar el bootloader mediante Arduino IDE.

****

*Figura 6. Ejemplo de cómo incorporar el instalador de CircuitPython a la unidad.*

Las placas que no tienen SPI flash necesitan poner el sistema de archivos en flash, eso significa que solo obtiene alrededor de 48 KB de espacio en el sistema de archivos. Esa asignación adicional del sistema de archivos significa menos espacio para el firmware CircuitPython [2].

Una vez incorporado el archivo a la unidad FEATHERBOOT, el Feather M0 se reiniciará y deberá aparecer una nueva unidad que ahora sí es reconocida por Windows. Esta unidad se llamará CIRCUITPY.

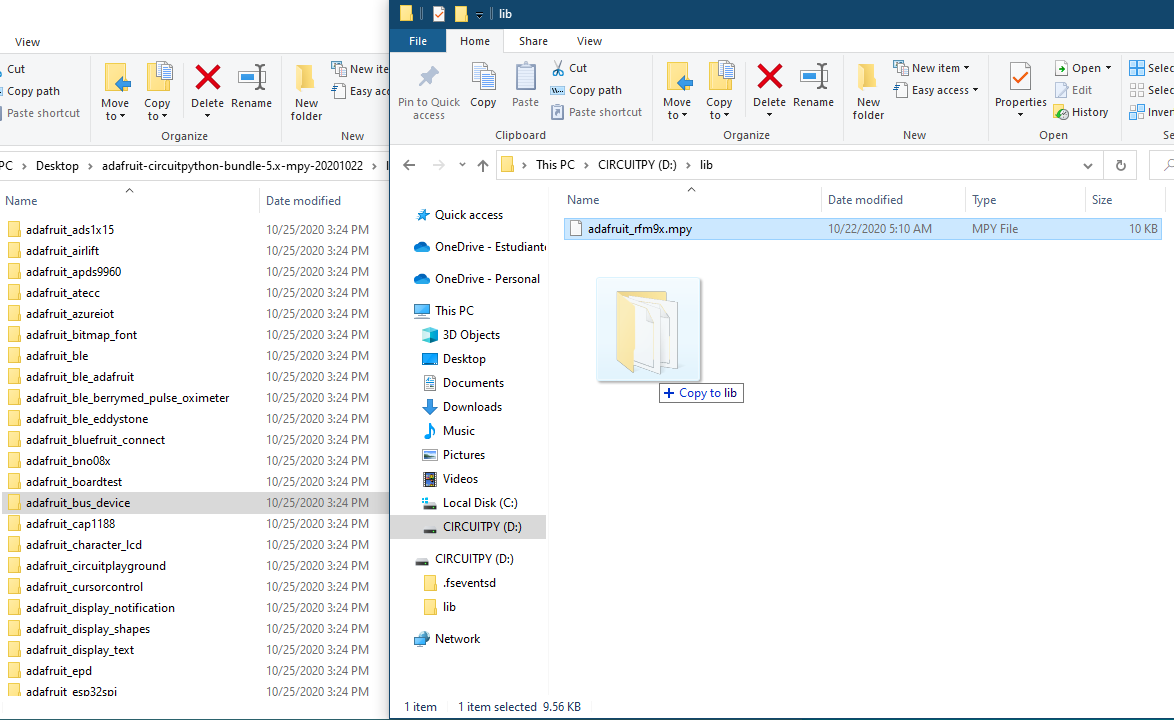
****

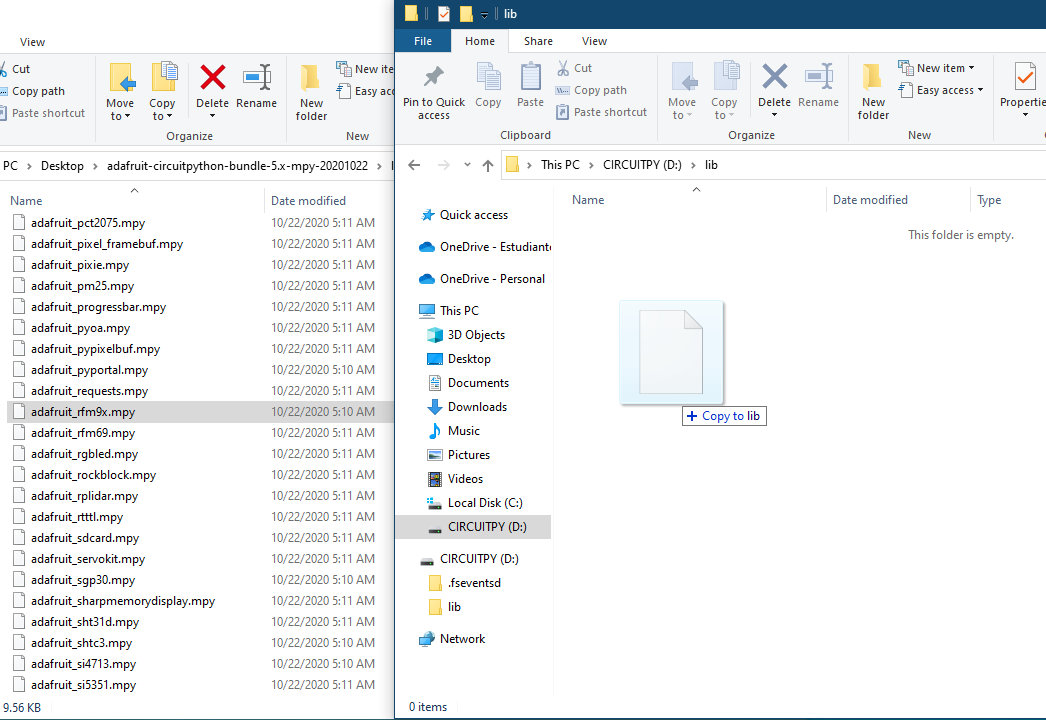
*Figura 7. Unidad de almacenamiento CIRCUITPY correspondiente al Feather M0.*

En este directorio CIRCUITPY es donde se coloca el archivo **.py** del código que se quiere ejecutar en el Feather M0, este archivo debe tener el nombre **main.py** o **code.py** para que sea autoejecutado por el bootloader del Feather M0.

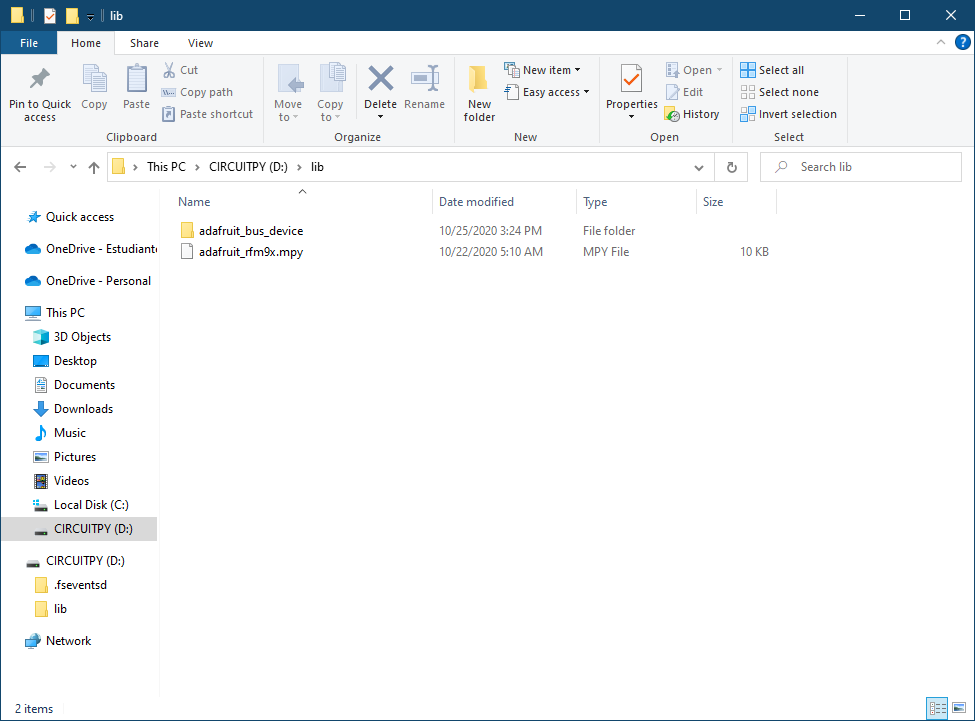
**Incorporación de librerías CircuitPython**

Una vez instalado CircuitPython, en el caso del Feather M0 es mejor agregar una a una las librerías según se requiera. Para el uso del módulo LoRa RFM9x incorporado en el Feather M0, conviene añadir en la carpeta **lib** las librerías del bundle de CircuitPython llamadas: ***adafruit\_bus\_device*** (carpeta) y ***adafruit\_rfm9x.mpy***

****

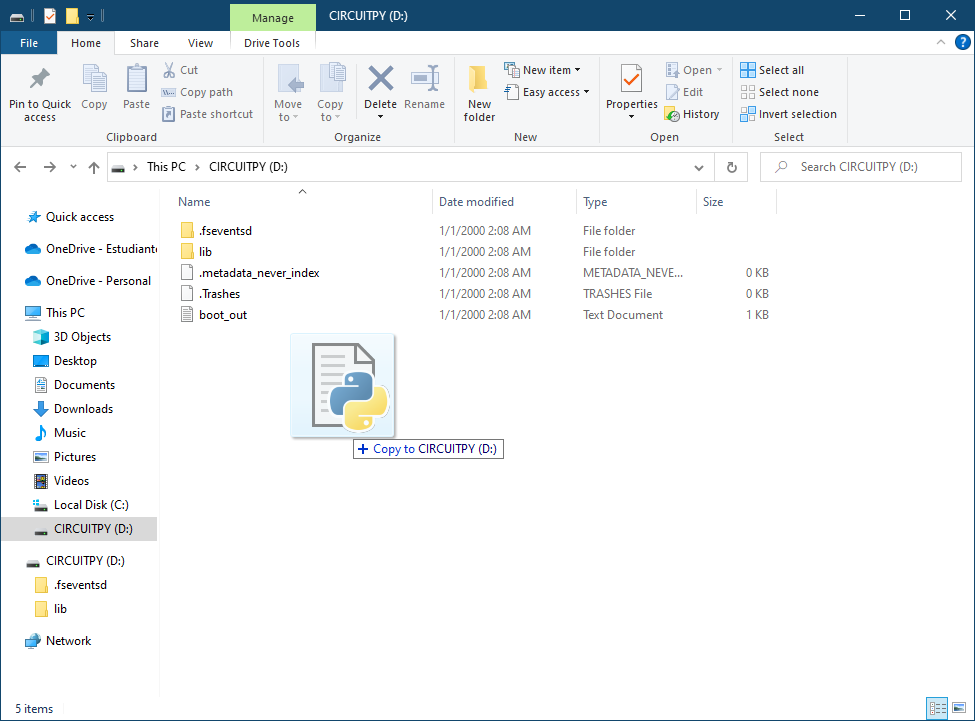
****

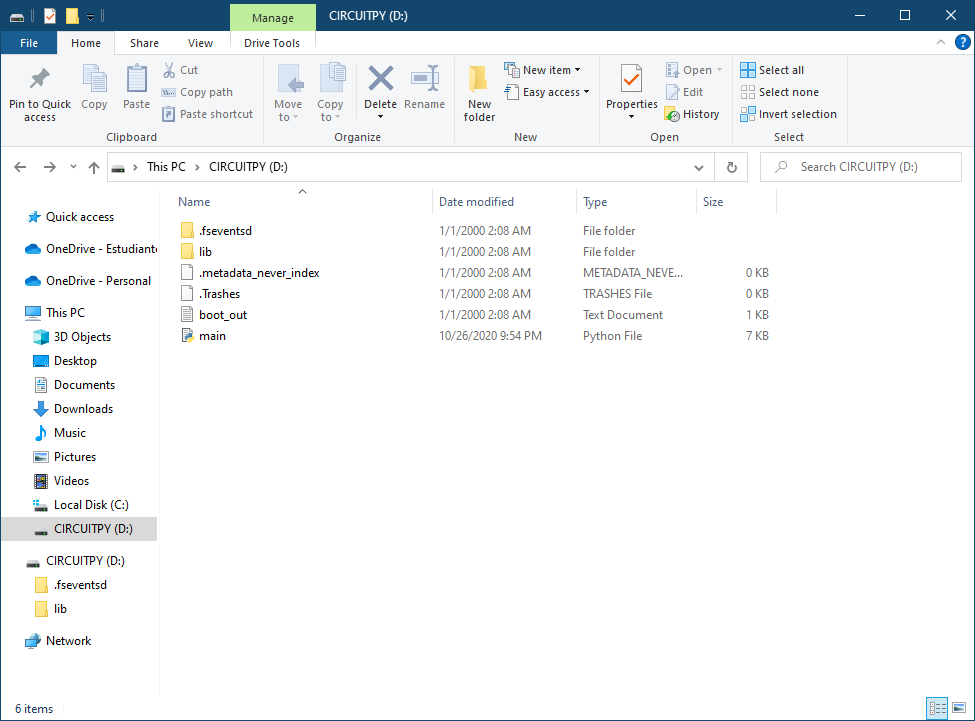
*Figura 8. Ejemplos de cómo añadir dos librerías en el directorio CIRCUITPY/lib.*

****

*Figura 9. Directorio CIRCUITPY/lib únicamente con dos librerías necesarias para el uso del módulo LoRa RFM9x.*

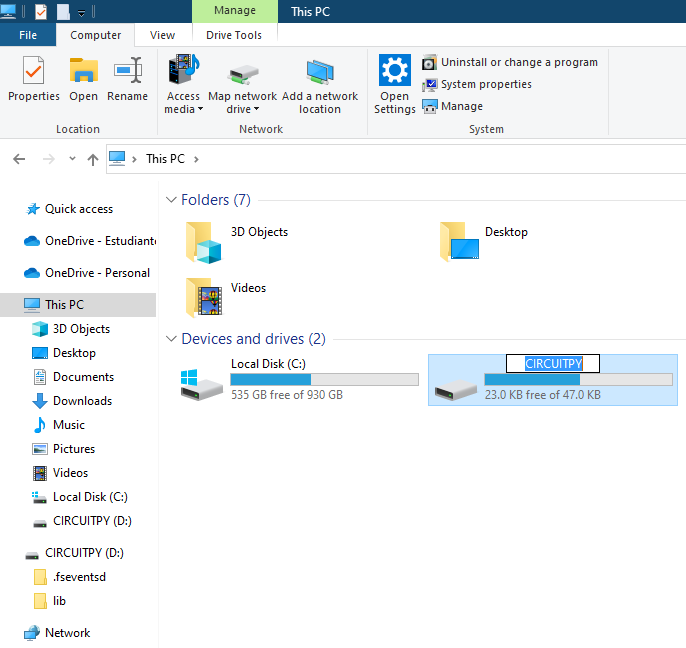
Finalmente, en la figura 10 se muestra cómo se puede añadir el código que se quiera ejecutar con el Feather M0. El código se puede guardar desde algún editor como Mu Editor (con consola serial y REPL), muy recomendado. El código también se puede editar desde cualquier editor, solo que hay que tener cuidado con los tiempos de escritura de algunos ya que si se desconecta el Feather M0 cuando no se ha guardado bien el código se puede corromper la unidad y tener que volver a instalar todo de nuevo.

****

****

*Figura 10. Carga del código en el directorio CIRCUITPY.*

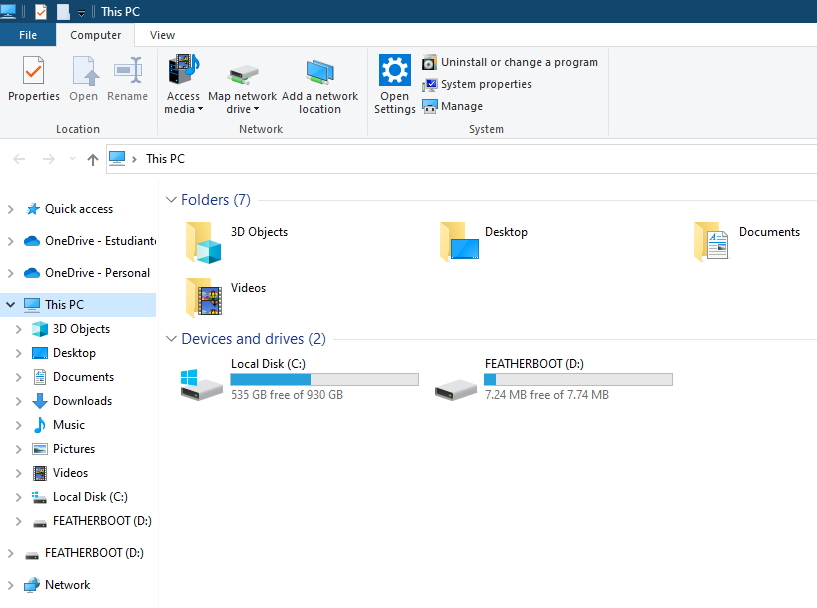
Por otro lado, si se desea cambiar el nombre del dispositivo Feather M0 simplemente basta con tocar el nombre y cambiárselo. Opcionalmente a este se le puso CIRCUITPY2. Renombrar la unidad puede ser útil si se tienen varios Feather M0 conectador al mismo computador a la vez.



*Figura 11. Ejemplo del cambio de nombre de la unidad CIRCUITPY.*

**Entrar en modo BOOT**

Esto se logra presionando dos veces rápidamente el botón de RESET. La unidad detectada por el sistema tendrá otro nombre en el modo BOOT, en este caso se llama FEATHERBOOT (independientemente del nombre que se le haya puesto a la unidad CIRCUITPY en el modo normal). Un ejemplo se muestra en la siguiente figura:

****

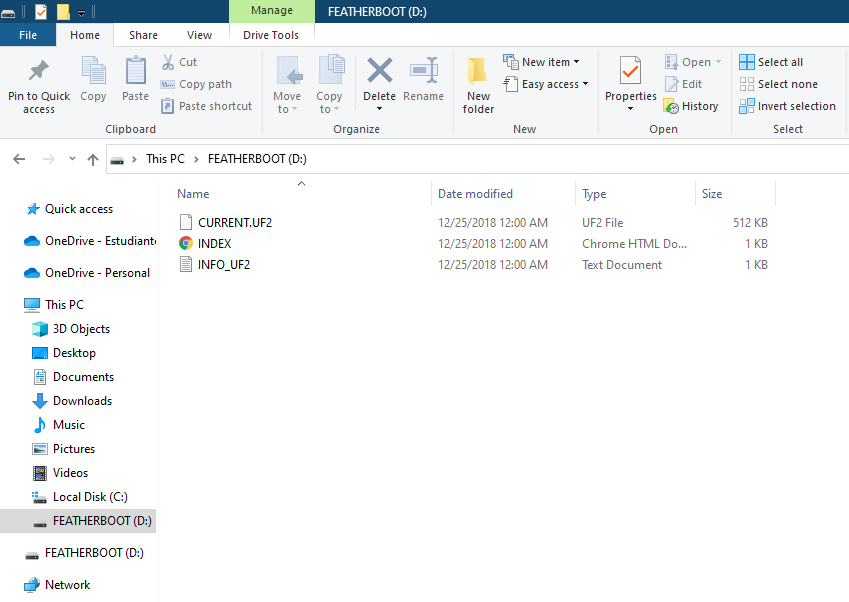
*Figura 12. Unidad FEATHERBOOT detectada tras entrar en modo BOOT.*

Una vez que el bootloader ha finalizado correctamente, se puede abrir la unidad y explorar el sistema de archivos virtual. Este no es el mismo sistema de archivos de la unidad CIRCUITPY que se usa con CircuitPython o Arduino [3]. Debe tener tres archivos:

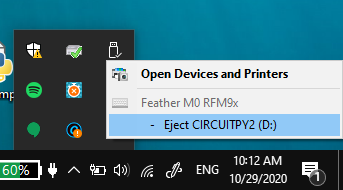
**CURRENT.UF2:** el contenido actual de la memoria flash del microcontrolador.

**INDEX.HTM:** enlace a Microsoft MakeCode.

**INFO\_UF2.TXT:** incluye la versión del bootloader.



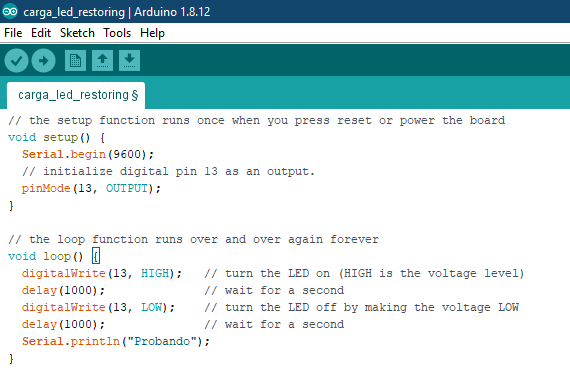
*Figura 13. Archivos dentro de la unidad FEATHERBOOT.*

Finalmente, si ya no se va a utilizar el Feather M0, se recomienda expulsar de forma segura la unidad;  
  


**Regresando a Arduino:** volver al inicio!

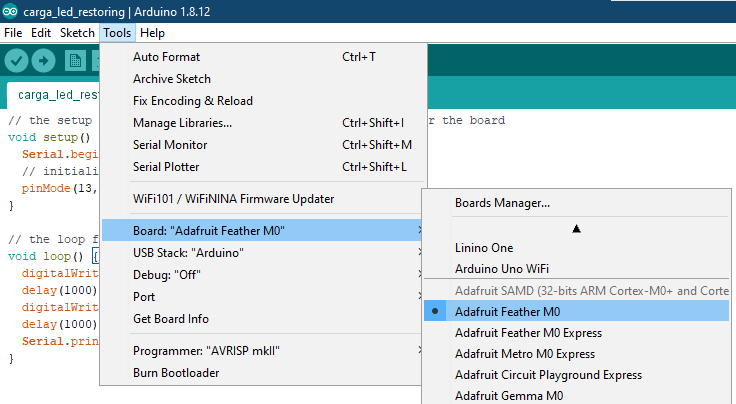
Tal y como se mencionó antes, el bootloader UF2 con el que se actualizó el Feather M0 es compatible con Arduino, por lo tanto, es capaz de detectar cuando se desea programar con Arduino o CircuitPython. Eso sí, TENGA EN CUENTA QUE AL VOLVER A ARDUINO SE REMOVERÁ CIRCUITPYTHON, por lo tanto, habrá que instalarlo de nuevo cuando se desee.

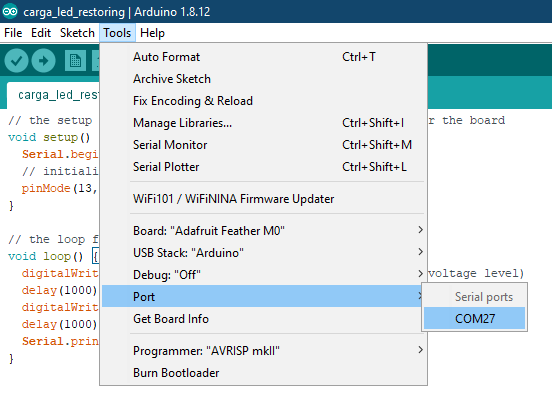
**1.** Para regresar a Arduino, simplemente cargue un código al Feather M0. El bootloader se encargará de preparar el microcontrolador para que a partir de este momento vuelva a ser programable mediante Arduino. Por ejemplo se puede utilizar el siguiente código que hace parpadear el LED del Feather M0 cada 1 s.

****

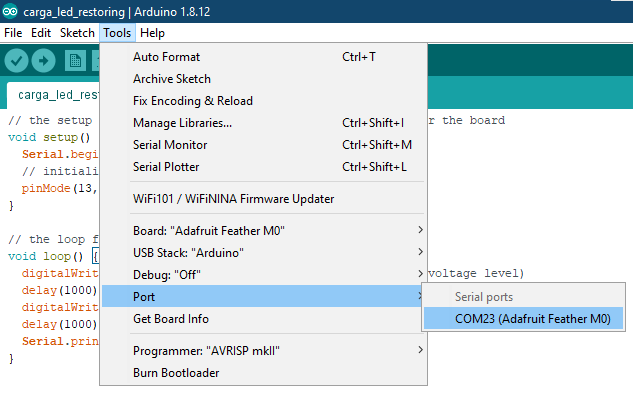
*Figura 14. Sketch de Arduino para cargarlo en el Feather M0.*

**2.** Ahora se debe seleccionar la board y el puerto correspondiente al Feather M0 que se esté utilizando. El puerto aparecerá como COM#, se recomienda buscar el número del puerto (#) en el Administrador de Dispositivos (Device Manager) de Windows. Asegúrese de que el puerto seleccionado corresponde al Feather M0.

****

****

Con esto se carga el código tomado de [3] luego de compilar y el LED comienza a parpadear cada 1 s. Luego de esto se verá que el Feather M0 es reconocido como un dispositivo compatible con Arduino. A partir de acá el Feather se puede utilizar mediante Arduino IDE como si fuera la primera vez (por defecto)! ¡Igualmente si se necesita se puede volver a instalar CircuitPython como se indicó antes!



*Figura 15. Detección del Feather M0 como una unidad compatible con Arduino.*

**Referencias:**

[1] <https://learn.adafruit.com/using-lorawan-and-the-things-network-with-circuitpython/circuitpython-installation>

[2] <https://learn.adafruit.com/installing-circuitpython-on-samd21-boards/installing-circuitpython>

[3] <https://learn.adafruit.com/adafruit-feather-m0-express-designed-for-circuit-python-circuitpython/uf2-bootloader-details>

[4] <https://learn.adafruit.com/welcome-to-circuitpython/uninstalling-circuitpython>