**Communication protocols:**

XBee3:

* Only supports I2C in Master mode, no Slave mode
* Supports SPI in Slave mode, but only in API mode

ESP8266:

* Only supports I2C in Master mode, no Slave mode
* Only supports SPI in Master mode, no Slave mode

Pros

* Certain ESPs are used with AdaFruit FeatherBoards
* Can communicate with LoRaWan stuff with certain ESPs

Cons

* Does not have many GPIO pins -> have to connect sensors and peripherals to the XBee and ESP rather than just the ESP
  + We are trying to connect everything to the microcontroller since it, rather than the XBee, will be retained in the long run

Solution:

* Operate both in Master mode, but software switch the modes for each board

**SD Card:**

Adafruit MicroSD Card Breakout Board:

* Supports SPI
* Has CircuitPython sample code

CircuitPython vs MicroPython explanation:

* <https://learn.adafruit.com/getting-started-with-raspberry-pi-pico-circuitpython/micropython-or-circuitpython>

**Raspberry Pi Pico:**

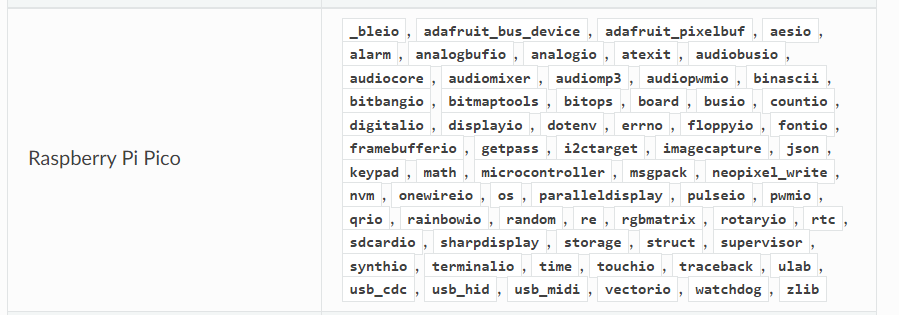


Figure 1: Available CircuitPython modules according to https://docs.circuitpython.org/en/latest/shared-bindings/support\_matrix.html

For a python file to be able to run at power on, save the file onto the Pico from Thonny by:

File 🡪 Save as 🡪 Save onto Pico 🡪 save file with the name “main.py” 🡪 Run the code in Thonny one more time to make sure code runs 🡪 Code should now run on boot

**XBee:**

Calendar

Description automatically generated

Graphical user interface, text, application

Description automatically generated

When setting up UART demo for field testing, you need to press the buttons again to configure them as sensors/coordinator. When the button is pressed and they are connected to a battery, source, they will initiate data transfers.

There is a **minimum sleep time** for the MESH network. Any inputted sleep value below 0.2 will cause hanging issues, or lost data.

**Pycharm:**

**Text

Description automatically generated**

* Install the 2022 Community Version
  + Problems were encountered with the 2018 version

**Reasons Bluetooth was not used:**

* Can do everything we want Bluetooth to do with the MESH network
* Don’t want to juggle Zigbee, Bluetooth, and LoRaWan protocols simultaneously
* The DigiXBeeMobile app doesn’t really help non-technical users initiate Bluetooth beaconing
  + Can't do the following with the app:
    - Turn XBee Bluetooth on with app->XBee scans (beacons) for other XBees->you turn the XBee off while its scanning (the app has trouble accessing the XBee while its scanning with Bluetooth)
    - Turn XBee's BT on when its Bluetooth off (phone can't discover XBee)

**SD Card w/Pico:**

1. Follow the video tutorial on this website: <https://www.digikey.com/en/maker/projects/raspberry-pi-pico-rp2040-sd-card-example-with-micropython-and-cc/e472c7f578734bfd96d437e68e670050>
2. Get the sdcard.py code from this website: <https://github.com/peterhinch/micropython-vs1053/blob/master/sdcard.py>
3. New repo location: <https://github.com/micropython/micropython-lib/tree/d0f97fc218f07c381c835d9f632904c1ae1c9d6b/micropython/drivers/storage/sdcard>
4. SDCard.py should be saved into the Pico in a special folder called “lib” and if no such folder exists on the pico, create it through Thonny.

**XBee Operation Modes:**

* Do we set the data sink to REPL or API
  + REPL mode limits the amount of data that can be moved, leading to potential data loss
  + API mode should be used