USB Serial Instructions

The usbRFID file works for serial. The only change that needs to be made is possible the usb0 to the correct usb. If there is only one usb device plugged in 0 is likely the correct one. In Linux open terminal in directory and run command `test.py`. If there are permission issues you may need to run command `chmod +x test.py` in the directory to make file executable. This will, set up the device and run start continuous read with off and on times that can be set in the test.py file.

for uart

uartRFID are the files that contain the code for uart instead of usb. The main difference is using the uart library instead of pyserial.

instructions for loading files onto esp32 with terminal.

1. make sure the raspberry pi is up to date. In terminal run `sudo apt update; sudo apt upgrade -y`
2. install pip python package manager `sudo apt install python3-pip`
3. install esptool for loading files onto esp32 `sudo pip3 install esptool`
4. in terminal run `dmesg | grep ttyUSB` this will let you know what port usb is attached to (probably usb0)
5. test connection run `esptool.py --port /dev/ttyUSB* flash_id` *** is the usb number from step 4** should bring up info about esp32
6. erase flash on esp32 `esptool.py --port /dev/ttyUSB* erase_flash`
7. go to https://micropython.org/download/esp32/ and download micropython most stable version OR just use the following direct link download download to /Downloads folder
8. now install micropython onto esp32 `esptool.py --port /dev/ttyUSB0 write_flash 0x1000 /Downloads/esp32-idf3-20210202-v1.14.bin`
9. install rshell `sudo pip3 install rshell` this allows connecting and uploading files to esp32
10. go to directory where RFID project files are and run `rshell --buffer-size=30 -p /dev/ttyUSB*`
11. command `ls /pyboard` will show you files loaded on the esp32, it should only be a file called boot.py right now
12. `cp *.py /pyboard` will copy all of the necessary files from the zip file to the esp32, **note** in this case * is only a wild card not a replacement for usb number like the rest of this instruction file.
13. type `repl` this will run a shell like python command line. enter `import test.py` and test.py will run. Any file named main.py will run on startup of the esp32. If test.py is renamed main.py it will run on boot of esp32. For testing I'm naming it test.py for now and just importing it (which runs it) to eliminate the need to restart the esp32.