**LabVIEW:**

00000685 2024-05-29 12:12:56.3789075 +25.1453458 IRP\_MJ\_CREATE - process 900 (LabVIEW.exe) DOWN 0x00000000

00000686 2024-05-29 12:12:56.3790424 +0.0001349 IRP\_MJ\_CREATE UP 0x00000000

00000695 2024-05-29 12:12:56.3795014 +0.0000116 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE DOWN 0x00000000 80 25 00 00 .%..

00000696 2024-05-29 12:12:56.3799836 +0.0004822 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE UP 0x00000000

00000697 2024-05-29 12:12:56.3799899 +0.0000063 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS DOWN 0x00000000

00000698 2024-05-29 12:12:56.3799909 +0.0000010 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS UP 0x00000000

00000699 2024-05-29 12:12:56.3799943 +0.0000034 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR DOWN 0x00000000

00000700 2024-05-29 12:12:56.3801562 +0.0001619 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR UP 0x00000000

00000701 2024-05-29 12:12:56.3801591 +0.0000029 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL DOWN 0x00000000 00 00 08 ...

00000702 2024-05-29 12:12:56.3806646 +0.0005055 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL UP 0x00000000

00000703 2024-05-29 12:12:56.3806863 +0.0000217 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS DOWN 0x00000000 00 00 00 00 11 13 ......

IOCTL\_SERIAL\_SET\_CHARS: Set special characters

EofChar=0x0

ErrorChar=0x0

BreakChar=0x0

EventChar=0x0

XonChar=0x11

XoffChar=0x13

00000704 2024-05-29 12:12:56.3806890 +0.0000027 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS UP 0x00000000

00000705 2024-05-29 12:12:56.3806904 +0.0000014 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW DOWN 0x00000000 01 00 00 00 40 00 00 00 40 00 00 00 40 00 00 00 ....@...@...@...

00000706 2024-05-29 12:12:56.3806911 +0.0000007 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW UP 0x00000000

00000707 2024-05-29 12:12:56.3806950 +0.0000039 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE DOWN 0x00000000 80 02 00 00 80 02 00 00 ........

IOCTL\_SERIAL\_SET\_QUEUE\_SIZE: Set queue size

InSize=640

OutSize=640

00000708 2024-05-29 12:12:56.3806956 +0.0000006 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE UP 0x00000000

00000709 2024-05-29 12:12:56.3806984 +0.0000028 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS DOWN 0x00000000 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 f4 01 00 00 ....................

IOCTL\_SERIAL\_SET\_TIMEOUTS: Set timeouts

ReadIntervalTimeout=4294967295

ReadTotalTimeoutMultiplier=0

ReadTotalTimeoutConstant=0

WriteTotalTimeoutMultiplier=0

WriteTotalTimeoutConstant=500

00000710 2024-05-29 12:12:56.3806993 +0.0000009 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS UP 0x00000000

00000713 2024-05-29 12:12:56.3807088 +0.0000027 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS DOWN 0x00000000

00000714 2024-05-29 12:12:56.3807093 +0.0000005 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS UP 0x00000000

00000715 2024-05-29 12:12:56.3807115 +0.0000022 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR DOWN 0x00000000

00000716 2024-05-29 12:12:56.3809080 +0.0001965 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR UP 0x00000000

00000725 2024-05-29 12:12:56.3814537 +0.0000066 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE DOWN 0x00000000 80 25 00 00 .%..

00000726 2024-05-29 12:12:56.3820119 +0.0005582 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE UP 0x00000000

00000727 2024-05-29 12:12:56.3820181 +0.0000062 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS DOWN 0x00000000

00000728 2024-05-29 12:12:56.3820190 +0.0000009 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS UP 0x00000000

00000729 2024-05-29 12:12:56.3820215 +0.0000025 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR DOWN 0x00000000

00000730 2024-05-29 12:12:56.3822126 +0.0001911 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR UP 0x00000000

00000731 2024-05-29 12:12:56.3822152 +0.0000026 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL DOWN 0x00000000 00 00 08 ...

00000732 2024-05-29 12:12:56.3827046 +0.0004894 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL UP 0x00000000

00000733 2024-05-29 12:12:56.3827070 +0.0000024 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS DOWN 0x00000000 00 00 00 00 11 13 ......

00000734 2024-05-29 12:12:56.3827081 +0.0000011 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS UP 0x00000000

00000735 2024-05-29 12:12:56.3827094 +0.0000013 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW DOWN 0x00000000 01 00 00 00 40 00 00 00 40 00 00 00 40 00 00 00 ....@...@...@...

00000736 2024-05-29 12:12:56.3827101 +0.0000007 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW UP 0x00000000

00000737 2024-05-29 12:12:56.3827135 +0.0000034 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE DOWN 0x00000000 80 02 00 00 80 02 00 00 ........

00000738 2024-05-29 12:12:56.3827140 +0.0000005 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE UP 0x00000000

00000739 2024-05-29 12:12:56.3827165 +0.0000025 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS DOWN 0x00000000 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 f4 01 00 00 ....................

00000740 2024-05-29 12:12:56.3827173 +0.0000008 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS UP 0x00000000

00000743 2024-05-29 12:12:57.9447867 +0.0000113 IRP\_MJ\_WRITE DOWN 0x00000000 2a 49 44 4e 0d 0a \*IDN..

00000752 2024-05-29 12:12:57.9549852 +0.0000034 IRP\_MJ\_READ UP 0x00000000 2a 49 44 4e 0d 0a 49 53 \*IDN..IS

00000762 2024-05-29 12:12:57.9651564 +0.0000037 IRP\_MJ\_READ UP 0x00000000 42 59 2d 55 43 43 2d 52 65 76 41 2e 31 0d 0a BY-UCC-RevA.1..

00000789 2024-05-29 12:12:58.0773243 +0.0000149 IRP\_MJ\_WRITE DOWN 0x00000000 4d 6f 64 65 30 0d 0a Mode0..

00000798 2024-05-29 12:12:58.0882322 +0.0000019 IRP\_MJ\_READ UP 0x00000000 4d 6f 64 Mod

00000808 2024-05-29 12:12:58.0983868 +0.0000025 IRP\_MJ\_READ UP 0x00000000 65 30 0d 0a 30 0d 0a e0..0..

00000839 2024-05-29 12:12:58.2214303 +0.0109394 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_CLR\_RTS DOWN 0x00000000

00000840 2024-05-29 12:12:58.2214421 +0.0000118 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_CLR\_RTS UP 0x00000000

00000841 2024-05-29 12:12:58.2214503 +0.0000082 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_CLR\_DTR DOWN 0x00000000

00000842 2024-05-29 12:12:58.2217144 +0.0002641 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_CLR\_DTR UP 0x00000000

00000843 2024-05-29 12:12:58.2217344 +0.0000200 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_PURGE DOWN 0x00000000 0f 00 00 00 ....

00000844 2024-05-29 12:12:58.2217409 +0.0000065 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_PURGE UP 0x00000000

00000845 2024-05-29 12:12:58.2259970 +0.0042561 IRP\_MJ\_CLOSE DOWN 0x00000000

00000846 2024-05-29 12:12:58.2261145 +0.0001175 IRP\_MJ\_CLOSE UP 0x00000000

**Python:**

00000847 2024-05-29 12:13:45.3282868 +47.1021723 IRP\_MJ\_CREATE - process 3316 () DOWN 0x00000000

00000848 2024-05-29 12:13:45.3283658 +0.0000790 IRP\_MJ\_CREATE UP 0x00000000

00000849 2024-05-29 12:13:45.3283942 +0.0000284 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE DOWN 0x00000000 00 10 00 00 00 10 00 00 ........

IOCTL\_SERIAL\_SET\_QUEUE\_SIZE: Set queue size

InSize=4096

OutSize=4096

00000850 2024-05-29 12:13:45.3283972 +0.0000030 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_QUEUE\_SIZE UP 0x00000000

00000853 2024-05-29 12:13:45.3284237 +0.0000100 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS DOWN 0x00000000 00 00 00 00 00 00 00 00 b8 0b 00 00 00 00 00 00 00 00 00 00 ....................

IOCTL\_SERIAL\_SET\_TIMEOUTS: Set timeouts

ReadIntervalTimeout=0

ReadTotalTimeoutMultiplier=0

ReadTotalTimeoutConstant=3000

WriteTotalTimeoutMultiplier=0

WriteTotalTimeoutConstant=0

00000854 2024-05-29 12:13:45.3284251 +0.0000014 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS UP 0x00000000

00000855 2024-05-29 12:13:45.3284305 +0.0000054 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_WAIT\_MASK DOWN 0x00000000 80 00 00 00 ....

00000856 2024-05-29 12:13:45.3284320 +0.0000015 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_WAIT\_MASK UP 0x00000000

00000873 2024-05-29 12:13:45.3297792 +0.0000042 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE DOWN 0x00000000 80 25 00 00 .%..

00000874 2024-05-29 12:13:45.3303541 +0.0005749 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_BAUD\_RATE UP 0x00000000

00000875 2024-05-29 12:13:45.3303654 +0.0000113 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS DOWN 0x00000000

00000876 2024-05-29 12:13:45.3303672 +0.0000018 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_RTS UP 0x00000000

00000877 2024-05-29 12:13:45.3303701 +0.0000029 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR DOWN 0x00000000

00000878 2024-05-29 12:13:45.3306145 +0.0002444 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_DTR UP 0x00000000

00000879 2024-05-29 12:13:45.3306191 +0.0000046 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL DOWN 0x00000000 00 00 08 ...

00000880 2024-05-29 12:13:45.3311032 +0.0004841 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_LINE\_CONTROL UP 0x00000000

00000881 2024-05-29 12:13:45.3311079 +0.0000047 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS DOWN 0x00000000 ef 01 01 00 11 13 ......

IOCTL\_SERIAL\_SET\_CHARS: Set special characters

EofChar=0xad

ErrorChar=0x2

BreakChar=0x2

EventChar=0x0

XonChar=0x11

XoffChar=0x13

00000882 2024-05-29 12:13:45.3311099 +0.0000020 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_CHARS UP 0x00000000

00000883 2024-05-29 12:13:45.3311114 +0.0000015 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW DOWN 0x00000000 01 00 00 00 40 00 00 00 40 00 00 00 40 00 00 00 ....@...@...@...

00000884 2024-05-29 12:13:45.3311123 +0.0000009 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_HANDFLOW UP 0x00000000

00000885 2024-05-29 12:13:45.3311642 +0.0000519 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_PURGE DOWN 0x00000000 0f 00 00 00 ....

00000886 2024-05-29 12:13:45.3311677 +0.0000035 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_PURGE UP 0x00000000

00000887 2024-05-29 12:13:45.3315105 +0.0003428 IRP\_MJ\_WRITE DOWN 0x00000000 2a 49 44 4e 0d 0a \*IDN..

00000892 2024-05-29 12:13:45.3319708 +0.0000024 IRP\_MJ\_READ UP 0x00000000 2a \*

00000896 2024-05-29 12:13:45.3319953 +0.0000006 IRP\_MJ\_READ UP 0x00000000 49 I

00000900 2024-05-29 12:13:45.3320074 +0.0000005 IRP\_MJ\_READ UP 0x00000000 44 D

00000904 2024-05-29 12:13:45.3320194 +0.0000005 IRP\_MJ\_READ UP 0x00000000 4e N

00000908 2024-05-29 12:13:45.3320325 +0.0000004 IRP\_MJ\_READ UP 0x00000000 0d .

00000912 2024-05-29 12:13:45.3320423 +0.0000005 IRP\_MJ\_READ UP 0x00000000 0a .

00000913 2024-05-29 12:13:45.3322906 +0.0002483 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS DOWN 0x00000000 ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 f4 01 00 00 ....................

00000914 2024-05-29 12:13:45.3322921 +0.0000015 IRP\_MJ\_DEVICECONTROL: IOCTL\_SERIAL\_SET\_TIMEOUTS UP 0x00000000

00000915 2024-05-29 12:13:45.3366053 +0.0043132 IRP\_MJ\_CLOSE DOWN 0x00000000

00000916 2024-05-29 12:13:45.3367646 +0.0001593 IRP\_MJ\_CLOSE UP 0x00000000

Email 30 – 5 – 2024

HI Rob,

I did a Serial debugging session comparing Python and LabVIEW.

Here is what I found:

The commands used in setting up the port are not identical. Many are, but not all. I have attached the full document showing the comparison, but here is a summary.

There are three commands that are different:

1. IOCTL\_SERIAL\_SET\_CHARS (this is where I believe the problem most likely is)

* The challenge here is to figure out how to set these parameters in either LabVIEW or Python. Both are opaque in comparison to C.
* Given that the EofChar (end of file), and break characters are different, I could imagine this causing a problem.
  1. LabVIEW:

IOCTL\_SERIAL\_SET\_CHARS: Set special characters

  EofChar=0x0

  ErrorChar=0x0

  BreakChar=0x0

  EventChar=0x0

  XonChar=0x11

  XoffChar=0x13

* 1. Python

IOCTL\_SERIAL\_SET\_CHARS: Set special characters

  EofChar=0xad

  ErrorChar=0x2

  BreakChar=0x2

  EventChar=0x0

  XonChar=0x11

  XoffChar=0x13

1. IOCTL\_SERIAL\_SET\_QUEUE\_SIZE (I think it is highly unlikely that this is the problem)
   1. LabVIEW

IOCTL\_SERIAL\_SET\_QUEUE\_SIZE: Set queue size

  InSize=640

  OutSize=640

* 1. Python

IOCTL\_SERIAL\_SET\_QUEUE\_SIZE: Set queue size

  InSize=4096

  OutSize=4096

1. IOCTL\_SERIAL\_SET\_TIMEOUTS (the problem could also be here)

* This one is strange, since when I change the timeout value in LabVIEW, it doesn’t change these parameters.
* It seems that the LabVIEW timeout might be part of the LabVIEW code, rather than something that defines the serial port communication. I will need to look further into this.
  1. LabVIEW

IOCTL\_SERIAL\_SET\_TIMEOUTS: Set timeouts

  ReadIntervalTimeout=4294967295

  ReadTotalTimeoutMultiplier=0

  ReadTotalTimeoutConstant=0

  WriteTotalTimeoutMultiplier=0

  WriteTotalTimeoutConstant=500

* 1. Python

IOCTL\_SERIAL\_SET\_TIMEOUTS: Set timeouts

  ReadIntervalTimeout=0

  ReadTotalTimeoutMultiplier=0

  ReadTotalTimeoutConstant=3000

  WriteTotalTimeoutMultiplier=0

  WriteTotalTimeoutConstant=0

In the document, I have expanded the information for where LabVIEW and Python disagreed.

Note also, that in LabVIEW used a linefeed and carriage return character, while Python initially none – I added “\r\n” to get the linefeed + carriage return but that didn’t help.

My Python code was simply the following:

# Configure the serial port (example configuration)

port = 'COM4'

baud\_rate = 9600  # Set the baud rate

try:

    # Open the serial port

    ser = serial.Serial(port, baud\_rate, timeout=3, stopbits=serial.STOPBITS\_ONE)

    # Check if the serial port is open

    if ser.is\_open:

        print(f"Serial port {port} opened successfully.")

    else:

        print(f"Failed to open serial port {port}.")

    # Example: Write data to the serial port

    num = ser.write(b'\*IDN\r\n')

    print(f"Bytes written: {num}:")

   # Example: Read data from the serial port

    response = " "

    response = ser.readline()

    print(f"Received: {response}")

Update,

I solved the timeout issue with the following Python line of code:

    ser = serial.Serial(port, baud\_rate, timeout = 0, write\_timeout = 0.5, stopbits=serial.STOPBITS\_ONE)

Now that discrepancy between LabVIEW and Python has been eliminated, but things still don’t work.

This gets me back to the character definitions, where I believe the problem lies.

HI Rob,

I think we are good to go with Python.

It turns out that readline fails, but read doesn’t.

When I changed my Python code to:

    response = ser.read(100)

    print(f"Received: {response}")

everything now works.

Give it a shot and let me know it works for you.

My main code is:

   num = ser.write(b'\*IDN\r\n')

    print(f"Bytes written: {num}:")

    num = ser.out\_waiting

    print(f"Bytes available: {num}")

   # Example: Read data from the serial port

    response = " "

    response = ser.read(100)

    print(f"Received: {response}")

The response was:

Bytes written: 6:

Bytes available: 0

Received: b'\*IDN\r\nISBY-UCC-RevA.1\r\n'

So this now appears to be largely working.

It would be good if I could ge the ser.out.waiting function to also work, but I am happy for some progress.

-Frank



FP

This is even nicer:

    num = ser.write(b'Mode0\r\n')

    print(f"Bytes written: {num}:")

    num = ser.write(b'Read1:100\r\n')

    print(f"Bytes written: {num}:")

    num = ser.out\_waiting

    print(f"Bytes available: {num}")

   # Example: Read data from the serial port

    response = " "

    response = ser.read\_until('\n',size=None)

    print(f"Received: {response}")

The read\_until function makes it easy to read the required number of bytes when many are expected.

Here I read 100 values cleanly.

Note that the Mode0 (more Mode1) will need to be sent to the IBM4 so that it knows from then on if a read is based on the B2U or not.

Otherwise an error will result.

-Frank

This code will autolocate an IBM4 and then has some example code.

Changing the timeout (making it small) while checking the ports makes sense, and is something I should do with LabVIEW if we would continue with it.

This might be the easiest way going forward.

-Frank

import sys

import glob

import serial

baud\_rate = 9600  # Set the baud rate

def FindIBM4():

    """ Goes through all serial ports looking for an IBM4

        once found, return the port

    """

    if sys.platform.startswith('win'):

        ports = ['COM%s' % (i + 1) for i in range(256)]

    elif sys.platform.startswith('linux') or sys.platform.startswith('cygwin'):

        # this excludes your current terminal "/dev/tty"

        ports = glob.glob('/dev/tty[A-Za-z]\*')

    elif sys.platform.startswith('darwin'):

        ports = glob.glob('/dev/tty.\*')

    else:

        raise EnvironmentError('Unsupported platform')

    for port in ports:

        try:

            s = serial.Serial(port, baud\_rate, timeout = 0.05, write\_timeout = 0.1, inter\_byte\_timeout=0.1, stopbits=serial.STOPBITS\_ONE)

            s.write(b'\*IDN\r\n')

            response = s.read\_until('\n',size=None)

            Code = response.rsplit(b"\r\n")

            size = len(Code)

            if (len(Code) > 2) :

                if (Code[1] == b"ISBY-UCC-RevA.1") :

                    print(f"IBM4 found at {port}")

                    IBM4Port = port

            s.close()

        except (OSError, serial.SerialException):

            pass

    return IBM4Port

try:

    # Open the serial port

    port = FindIBM4()

    ser = serial.Serial(port, baud\_rate, timeout = 3, write\_timeout = 0.1, inter\_byte\_timeout=0.1, stopbits=serial.STOPBITS\_ONE)

    # Check if the serial port is open

    if ser.is\_open:

        print(f"Serial port {port} opened successfully.")

    else:

        print(f"Failed to open serial port {port}.")

    # Example: Write data to the serial port

    # num = ser.write(b'\*IDN\r\n')

    num = ser.write(b'Mode0\r\n')

    print(f"Bytes written: {num}:")

    num = ser.write(b'Read0:1000\r\n')

    print(f"Bytes written: {num}:")

    num = ser.out\_waiting

    print(f"Bytes available: {num}")

   # Example: Read data from the serial port

    response = " "

    response = ser.read\_until('\n',size=None)

    print(f"Received: {response}")

    # response = ser.get\_settings()

    # print(f"Settings: {response}")

except serial.SerialException as e:

    print(f"Error opening serial port {port}: {e}")

finally:

    # Close the serial port

    if ser.is\_open:

        ser.close()

        print(f"Serial port {port} closed.")