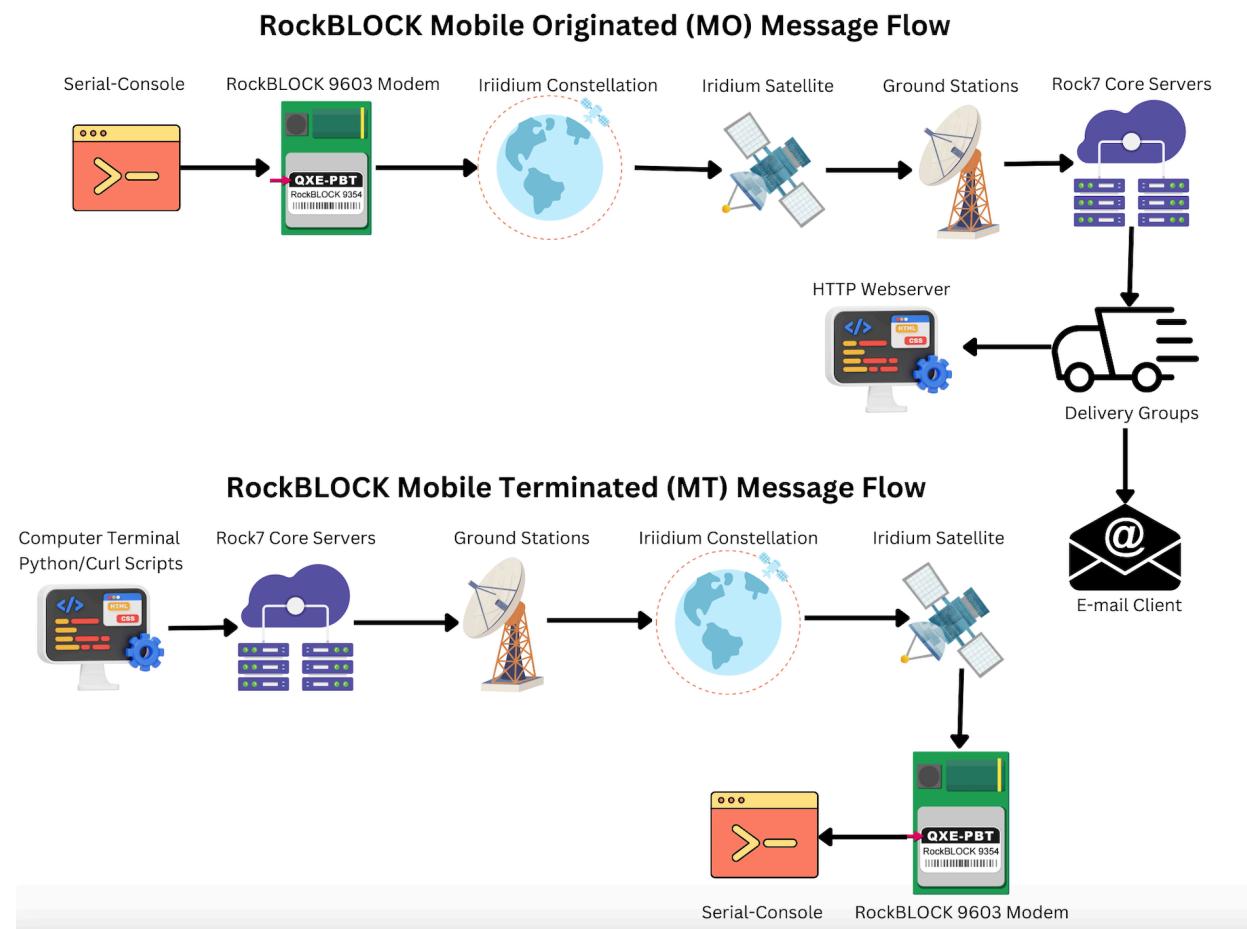


Iridium Satellite Communications Using Ground Control's RockBLOCK 9603 Modem



OBJECTIVE

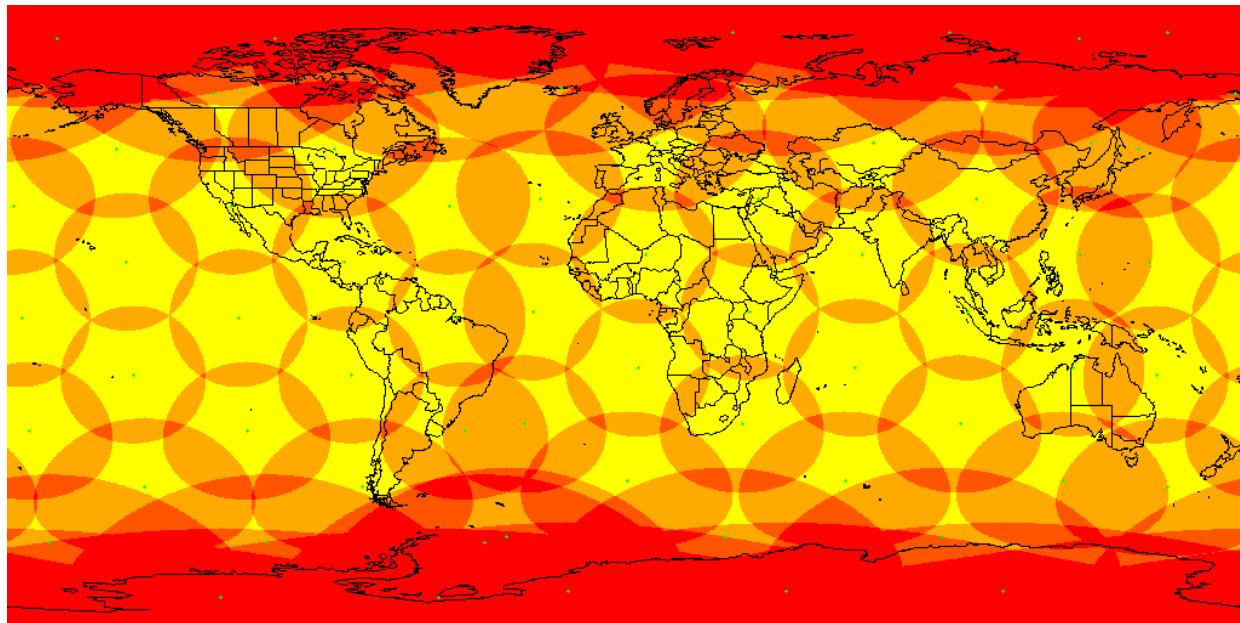
Do you want to be able to communicate short data messages with others from anywhere of planet earth? Or perhaps you have a need to send some sensor data or telemetry data periodically from the middle of nowhere where there is no cell-coverage or any other means to communicate. If you are interested in this, then read on....

This guide provides details regarding how to communicate with the Low Earth Orbit (LEO) Iridium Satellites with the [Ground Control's RockBLOCK 9603 Modem](#) breakout board using Python.

Please Note: The use of this satellite modem does require a [pre-paid \(monthly\) line-rental and credits](#) to send and receive messages. The credits never expire and every 50 bytes that are sent and/or received consumes one credit.

The Iridium Satellite constellation is a collection of 66 satellites that are in low-earth polar orbits around the earth that essentially provides full earth coverage, no matter where on earth. The only requirement is a clear view of the sky for the tiny ceramic antenna on the RockBLOCK 9603 modem.

The animation below attempts to show the Iridium satellite coverage. Check out the coverage at the poles. Impressive isn't it?



GOALS

The goal is to automate the receive of the Mobile Originated (MO), i.e. messages originating from the [RockBLOCK 9603](#) device using a Python HTTP Web-Server that handles HTTP POST messages. Once the message is delivered, the data along with the meta data can be extracted and utilized for any application purposes.

Likewise, the transmit of a message from an application or another computer terminal to the [RockBLOCK 9603](#) device, i.e. Mobile Terminated messages (MT) can also be automated using a Python and/or curl scripts.

WHERE CAN I BUY THE ROCKBLOCK 9603 MODEM?

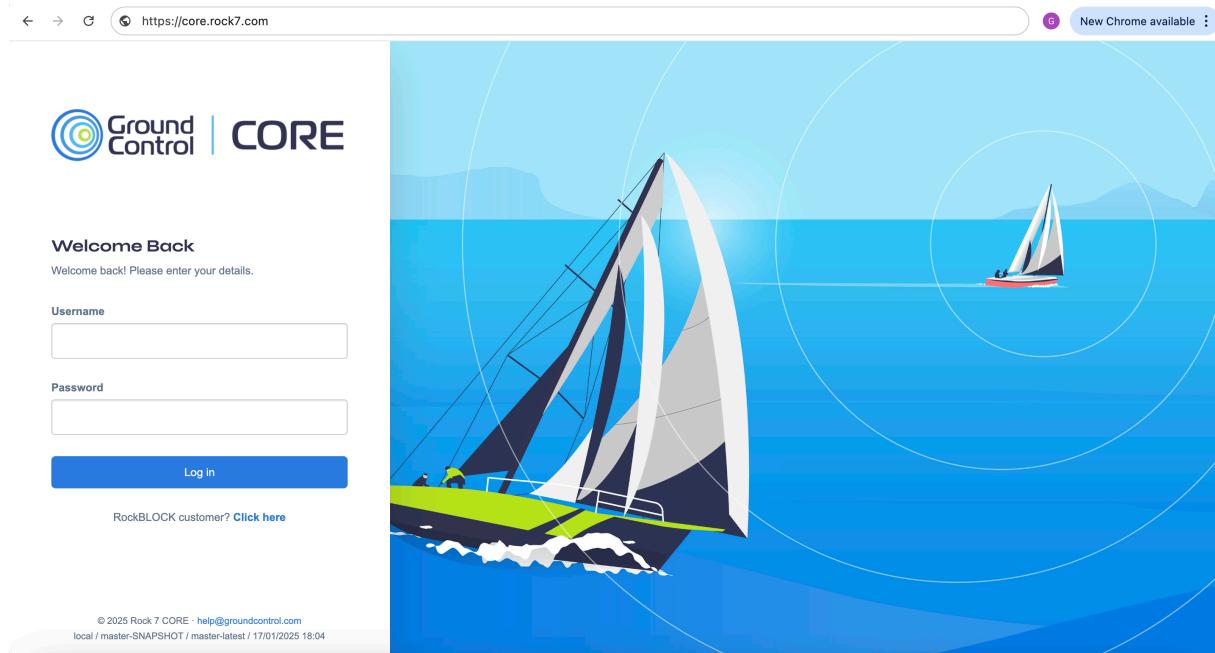
<https://www.groundcontrol.com/product/rockblock-9603-compact-plug-and-play-satellite-transmitter/>

<https://www.adafruit.com/product/4521>

<https://www.sparkfun.com/products/14498>

SIGNUP FOR ROCKBLOCK ACCOUNT

<https://core.rock7.com/>



101 credits remaining, 1 active RockBLOCK

New Chrome available

Support Logout

My RockBLOCKS

Add New RockBLOCK

A registration code (of the form XYZ-XYZ, or ABC-DEF-R) is printed on your RockBLOCK Naked modem (or on the label of the RockBLOCK+). Enter the code below to add the device to your account.

Registration Code Add

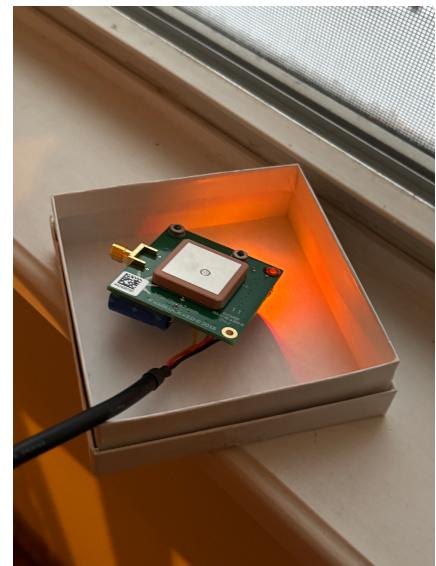
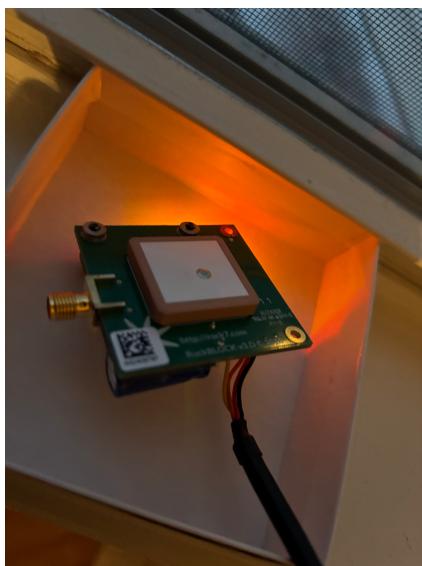
There is currently 1 RockBLOCK device in your account.
Should you require assistance with any RockBLOCK device(s), please view our [support](#) page.

Name	Serial	IMEI	Status
RockBLOCK 9603			Line rental expires 2025-01-25

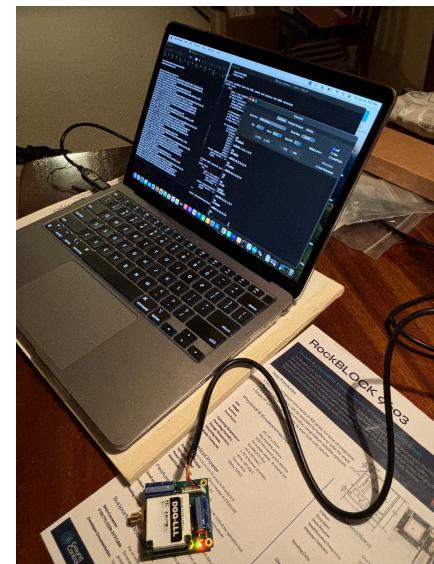
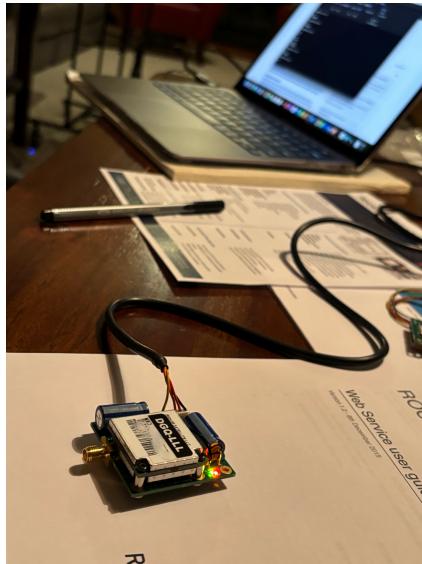
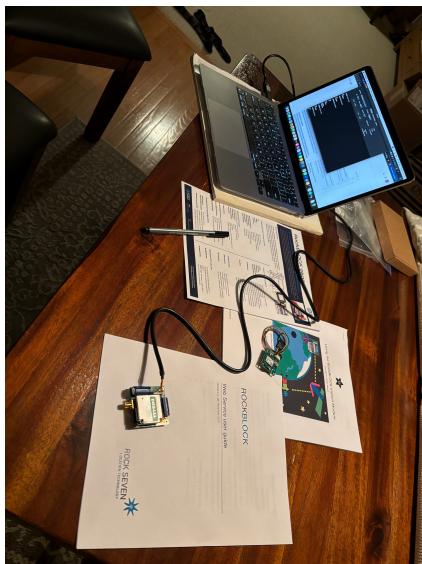
Terminate

PICTURES OF MY SETUP

My RockBLOCK 9603 Modem sitting on a window sill with access to the sky on a super sub-zero cold day. It's impressive that even with this limited (open sky access) setup, it works incredibly well.



My computer setup to interact with the (virtual) serial terminal using the supplied FTDI cable.



SOLUTION

GitHub Repo

<https://github.com/gkrangan/rockblock>

Python HTTP Webserver to Receive Mobile Originated (MO) Messages

```
# Python HTTP Webserver to receive POST messages from Rock7 Servers

import json
from http.server import BaseHTTPRequestHandler, HTTPServer

def hex_to_text(hex_string):
    """Converts a hexadecimal string to text."""
    # Remove the leading '0x' if present
    if hex_string.startswith("0x"):
        hex_string = hex_string[2:]

    # Convert the hex string to bytes
    bytes_object = bytes.fromhex(hex_string)

    # Decode the bytes object to text (assuming ASCII encoding)
    text = bytes_object.decode("utf-8")

    return text

class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):

    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        post_data = self.rfile.read(content_length)

        # Process the POST data here
        print("Received POST data:", post_data)

        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b"POST request received")

        # Process the received message
        decoded_post_data = post_data.decode('utf-8')
        # print(str(decoded_post_data).split('&'))
        output = str(decoded_post_data).split('&')
        print(output)
        print(output[0].split('='))
        print(output[1].split('='))
        print(output[2].split('='))
        print(output[3].split('='))
        print(output[4].split('='))
        print(output[5].split('='))
        print(output[6].split('='))
        print(output[7].split('='))
        print(output[8].split('='))
        print(output[9].split('='))

        # Extract message in Hex:
        # hex_string = output[9].split('=')[1].rstrip("''")
        hex_string = output[9].split('=')[1]
        print("Message Received in Hex:" + hex_string)
        text = hex_to_text(hex_string)
        print("Message Received ASCII: " + text)

if __name__ == '__main__':
    server_address = ('', 8000)
    httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
```

```
print("Starting Python HTTP Webserver to receive POST messages from Rock7 Servers on port 8000...")
httpd.serve_forever()
```

 **Don't forget to whitelist Ground Control's IP Addresses to Receive Data!**

These are the IP addresses you will need to whitelist so data from the Rock7 servers can be through your firewall:

109.74.196.135

212.71.235.32

Mobile Originated (MO) messages sent by RockBLOCK and Received by HTTP Webserver

```
% python rbs.py
Starting Python HTTP Webserver to receive POST messages from Rock7 Servers on port 8000...
Received POST data:
b'imei=301434061666630&device_type=ROCKBLOCK&serial=224733&momsn=35&transmit_time=25-01-27%2022%3A24%3A10&iridium_latitude=40.1006&iridium_longitude=-82.7993&iridium_cep=20.0&iridium_session_status=0&data=4f6e6520536d616c6c205374657020666f72204d616e2c204f6e65204769616e74204c65617020666f72204d416e6b696e64'212.71.235.32 -- [27/Jan/2025 17:24:17] "POST / HTTP/1.1" 200 -
['imei=*****', 'device_type=ROCKBLOCK', 'serial=xxxxxx', 'momsn=35',
 'transmit_time=25-01-27%2022%3A24%3A10', 'iridium_latitude=40.1006', 'iridium_longitude=-82.7993',
 'iridium_cep=20.0', 'iridium_session_status=0',
 'data=4f6e6520536d616c6c205374657020666f72204d616e2c204f6e65204769616e74204c65617020666f72204d416e6b696e64']
['imei', '*****']
['device_type', 'ROCKBLOCK']
['serial', 'xxxxxx']
['momsn', '35']
['transmit_time', '25-01-27%2022%3A24%3A10']
['iridium_latitude', '40.1006']
['iridium_longitude', '-82.7993']
['iridium_cep', '20.0']
['iridium_session_status', '0']
['data',
 '4f6e6520536d616c6c205374657020666f72204d616e2c204f6e65204769616e74204c65617020666f72204d416e6b66e64']
Message Received in
Hex:4f6e6520536d616c6c205374657020666f72204d616e2c204f6e65204769616e74204c65617020666f72204d416e6b696e64
Message Received ASCII: One Small Step for Man, One Giant Leap for Mankind
```

Received POST data:

```
b'imei=*****&device_type=ROCKBLOCK&serial=xxxxxx&momsn=36&transmit_time=25-01-27%2022%3A45%3A39&iridium_latitude=40.0529&iridium_longitude=-82.8517&iridium_cep=6.0&iridium_session_status=0&data=464c5553485f4d54'212.71.235.32 -- [27/Jan/2025 17:45:42] "POST / HTTP/1.1" 200 -
['imei=*****', 'device_type=ROCKBLOCK', 'serial=xxxxxx', 'momsn=36',
 'transmit_time=25-01-27%2022%3A45%3A39', 'iridium_latitude=40.0529', 'iridium_longitude=-82.8517',
 'iridium_cep=6.0', 'iridium_session_status=0', 'data=464c5553485f4d54']
['imei', '*****']
['device_type', 'ROCKBLOCK']
['serial', 'xxxxxx']
['momsn', '36']
['transmit_time', '25-01-27%2022%3A45%3A39']
['iridium_latitude', '40.0529']
['iridium_longitude', '-82.8517']
['iridium_cep', '6.0']
['iridium_session_status', '0']
['data', '464c5553485f4d54']
Message Received in Hex:464c5553485f4d54
Message Received ASCII: FLUSH_MT
```

Received POST data:

```
b'imei=*****&device_type=ROCKBLOCK&serial=xxxxxx&momsn=38&transmit_time=25-01-27%2023%3A03%3A40&iridium_latitude=40.0529&iridium_longitude=-82.8517&iridium_cep=5.0&iridium_session_status=0&data=5468652046616c636f6e20686173206c616e646564'212.71.235.32 -- [27/Jan/2025 18:03:45] "POST / HTTP/1.1" 200 -
['imei=*****', 'device_type=ROCKBLOCK', 'serial=xxxxxx', 'momsn=38',
 'transmit_time=25-01-27%2023%3A03%3A40', 'iridium_latitude=40.0529', 'iridium_longitude=-82.8517',
 'iridium_cep=5.0', 'iridium_session_status=0', 'data=5468652046616c636f6e20686173206c616e646564']
```

```

['imei', '*****']
['device_type', 'ROCKBLOCK']
['serial', 'xxxxxx']
['momsn', '38']
['transmit_time', '25-01-27%2023%3A03%3A40']
['iridium_latitude', '40.0529']
['iridium_longitude', '-82.8517']
['iridium_cep', '5.0']
['iridium_session_status', '0']
['data', '5468652046616c636f6e20686173206c616e646564']
Message Received in Hex:5468652046616c636f6e20686173206c61
Message Received ASCII: The Falcon has landed

```

[Github Repo](#)

<https://github.com/gkrangan/rockblock>

[Python script to Transmit Mobile Terminated \(MT\) Messages](#)

```

# Python Script to SEND Messages to RockBLOCK 9603

import sys
import requests
import binascii
from getpass import getpass

# RockBLOCK 9603 Device IMEI
SN = "xxxxxx"
IMEI = "*****"

USERNAME = input("Enter your RockBLOCK Account Username: ")
if not USERNAME:
    raise Exception("RockBLOCK Username is required!!")
    sys.exit(1)
print("Your RockBLOCK Account Username is:", USERNAME)

PASSWORD = getpass("Enter your RockBLOCK Account Password: ")
if not PASSWORD:
    raise Exception("RockBLOCK Account Password is required!!")
    sys.exit(1)
print("Your RockBLOCK Account Password is:", PASSWORD)

msg = input("Enter your Message: ")
if not msg:
    raise Exception("A Message (to be sent to the RockBLOCK device) is required!!")
    sys.exit(1)
print("Your Message to be sent to the RockBLOCK device is:", msg)

# encoded_msg = msg.encode("utf-8")
encoded_msg = msg.encode()

DATA = binascii.hexlify(encoded_msg)
print(DATA)
print(DATA.decode('utf-8'))
print("DATA Length: " + str(len(DATA.decode('utf-8'))))

# Build URL to HTTP POST send message
# url =https://core.rock7.com/rockblock/MT?imei=IMEI&username=USERNAME&password=PASSWORD&data=DATA
url = "https://core.rock7.com/rockblock/MT?imei={}&username={}&password={}{}".format(IMEI,
USERNAME, PASSWORD, DATA.decode('utf-8'))
print(url)
print("Sending Message to ROCKBLOCK 9603 with IMEI: " + IMEI)
response = requests.post(url=url)
print(response)

```

[Mobile Terminated \(MT\) messages sent by RockBLOCK](#)

```

% python send2rb.py
Enter your RockBLOCK Account Username: john doe@yahoo.com
Your RockBLOCK Account Username is: john doe@yahoo.com

```

```
Enter your RockBLOCK Account Password:  
Your RockBLOCK Account Password is: *****  
Enter your Message: Apollo11, This is Houston  
Your Message to be sent to the RockBLOCK device is: Apollo11, This is Houston  
b'41706f6c6c6f31312c205468697320697320486f7573746f6e'  
41706f6c6c6f31312c205468697320697320486f7573746f6e  
DATA Length: 50  
https://core.rock7.com/rockblock/MT?  
imei=301434061666630&username=johndoe@yahoo.com&password=*****&data=41706f6c6c6f31312c205468697320697320486f7573746f6e  
7320486f7573746f6e  
Sending Message to ROCKBLOCK 9603 with IMEI: *****  
<Response [200]
```

AT Commands on Serial Terminal connected to RockBLOCK 9603

```
AT  
OK
```

```
AT+CGMI  
Iridium  
OK
```

```
AT+CGMM  
IRIDIUM 9600 Family SBD Transceiver  
OK
```

```
AT+SBDWT=One Small Step for Man, One Giant Leap for Mankind  
OK
```

```
AT+SBDIX  
+SBDIX: 0, 34, 1, 20, 24, 1  
OK
```

```
AT+SBDRT  
+SBDRT:  
Apollo11, This is Houston  
OK
```

```
AT+SBDWT=FLUSH_MT  
OK
```

```
AT+SBDIX  
+SBDIX: 0, 36, 0, 0, 0, 0  
OK
```

```
AT+SBDRT  
+SBDRT:  
OK
```

```
AT+SBDIX  
+SBDIX: 32, 38, 2, 0, 0, 0  
OK
```

```
AT+SBDWT=The Falcon has landed  
OK
```

```
AT+SBDIX  
+SBDIX: 32, 38, 2, 0, 0, 0  
OK
```

```
AT+SBDIX  
+SBDIX: 0, 38, 1, 23, 17, 0  
OK
```

ROCKBLOCK 9603 AT COMMAND REFERENCE

Category	Command	Description
General	AT	ATtention Code
	AT+CGMI	Manufacturer Identification
	AT+CGMM	Model Identification
	AT+CGMR	Request Code Revision Information
	AT+CGSN	Request IMEI Number
	AT+MSSTM	Request System Time
	AT+MSGEO	Request Geo Location
	AT+MSGEOS	Request Geo Location, spherical coordinates
Transmit	AT+SBDWT=<msg>	Insert ASCII message in Mobile Originated (MO) Buffer
	AT+SBDIX	Execute Send/Receive. Initiate Extended SBD (Short Burst Data) Session
Receive	AT+SBDIX	Execute Send/Receive. Initiate Extended SBD (Short Burst Data) Session
	AT+SBDRT	Transfer ASCII (text) Received Message to Controller
	AT+SBDWT=FLUSH_MT	Clear Mobile Terminated (MT) Queue

REFERENCES

<https://docs.groundcontrol.com/iot/rockblock>

<https://learn.adafruit.com/using-the-rockblock-iridium-modem?view=all>

<https://core.rock7.com/>

<https://github.com/gkrangan/rockblock>