



Welcome

PLCnext Controllerを使った TCP/IPメッセージ (送信)

注意事項

- 本内容は2025年2月時点での資料となります。

本資料並びに本アプリケーションのサポートを将来的に約束するものではありません。

また本内容における動作保証に関して当社は一切の責任を負いません。

- “Node-RED for PLCnext ARM”をPLCnext storeよりダウンロードし、
PLCnext Controllerへインストール及びRUN状態にしておいて下さい。

デモ構成

- AXC F 2152とPCをLANケーブルで接続し、AXC F2152からTCP/IPでテストメッセージをPCに送信します。

```
コマンドプロンプト - python TCPI.py
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Phoenix>cd Documents
C:\Users\Phoenix\Documents>python TCPIP.py
Waiting for connection...
Connection from ('192.168.1.10', 49160) has been established.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
```

```
TCPIP.py
C:\Users\Phoenix\Documents>python TCPIP.py
1 import socket
2
3 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 server_socket.bind(("192.168.1.18", 12345))
5 server_socket.listen(1)
6
7 print("Waiting for connection...")
8
9 client_socket, addr = server_socket.accept()
10 print(f"Connection from {addr} has been established.")
11
12 try:
13     while True:
14         data = client_socket.recv(1024)
15         if not data:
16             break
17         print(f"Received data: {data.decode('ascii')}")
18 except KeyboardInterrupt:
19     print("Interrupted by ctrl + c.")
20 finally:
21     client_socket.close()
22     server_socket.close()
```

“Hello, TCP/IP.”

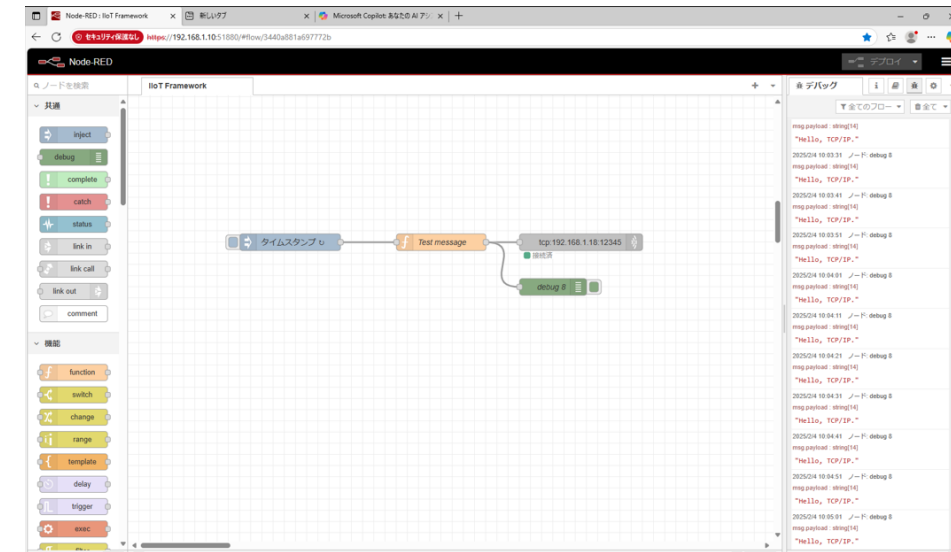
TO 192.168.1.18/24:12345



192.168.1.18/24



192.168.1.10/24

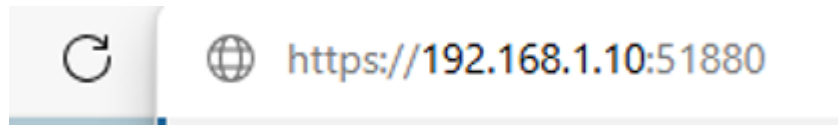


192.168.1.10/24:51880

Node-REDへのアクセス

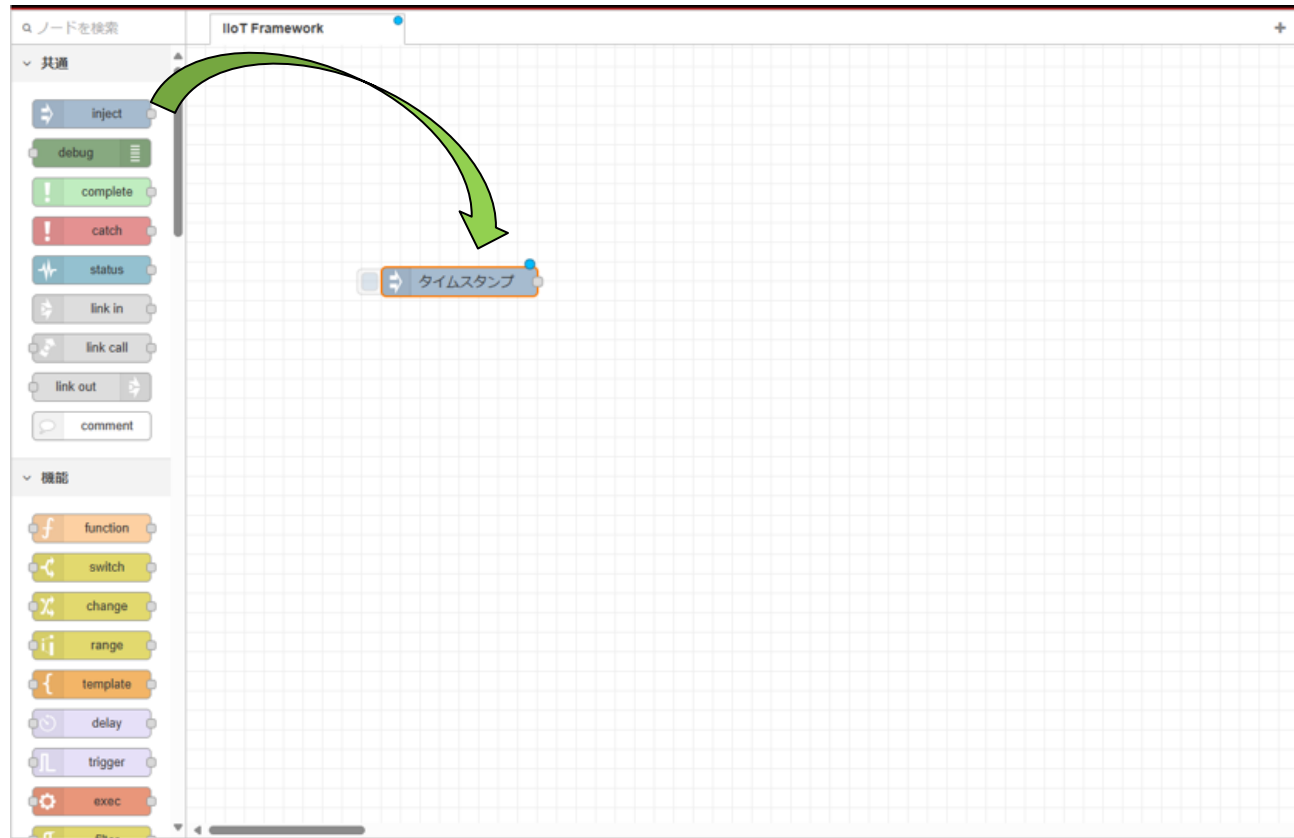
- ブラウザ検索画面で“https://<EPCのIPアドレス>:51880”を入力します。
ここでは<https://192.168.1.10:51880>を入力します。

※EPC内蔵のNode-REDではなく、PLCnext Controller共用のNode-REDのアプリケーションに移行したため、Node-REDの画面へ移行する場合は上記の通り、IPアドレス:51880でアクセスします。



Node-RED側(AXC F2152)のセットアップ

- “inject”ノードをドラッグ&ドロップします。



Node-RED側(AXC F2152)のセットアップ°

- “inject”ノードをクリックし、下記の通りの設定を行います。

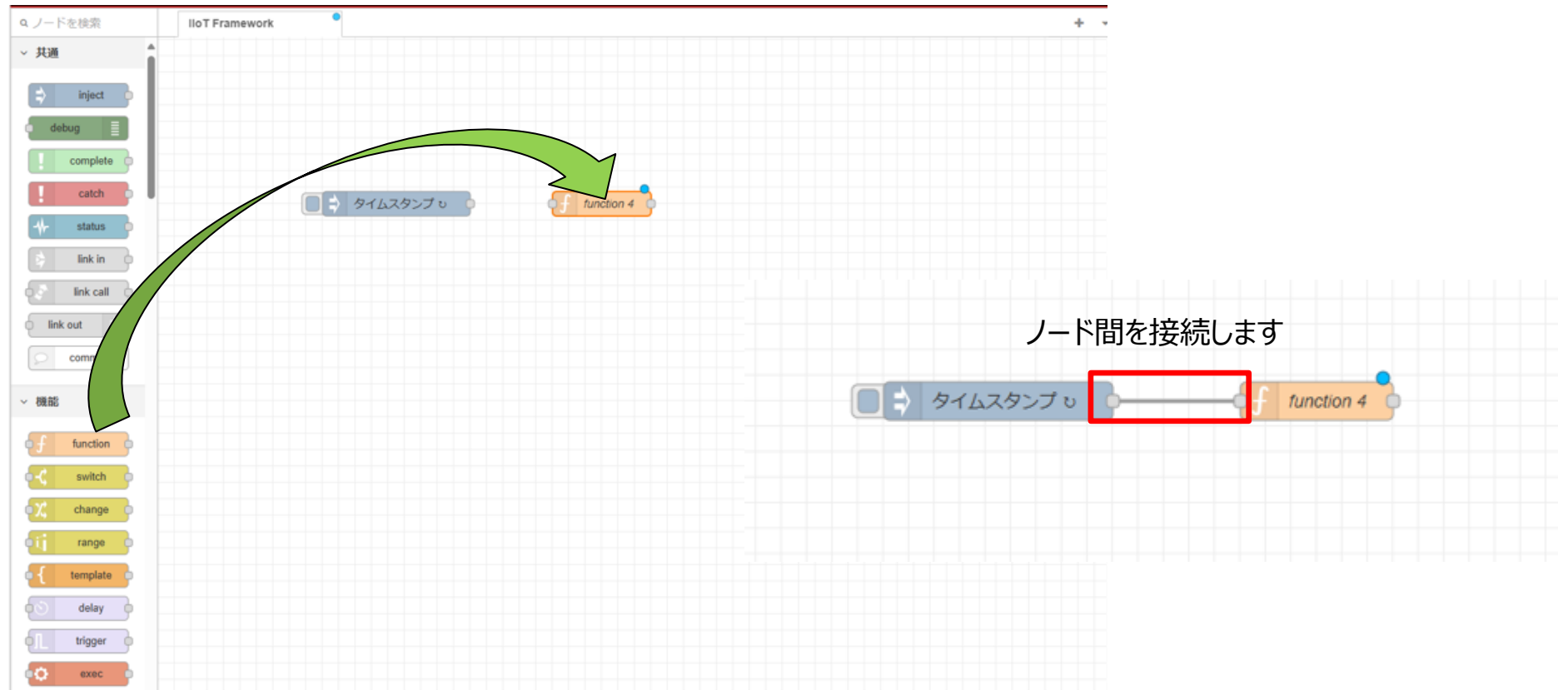
“繰り返し”: “指定した時間間隔”

“時間間隔”: “10”秒



Node-RED側(AXC F2152)のセットアップ

- “function”ノードをドラッグ&ドロップします。



Node-RED側(AXC F2152)のセットアップ

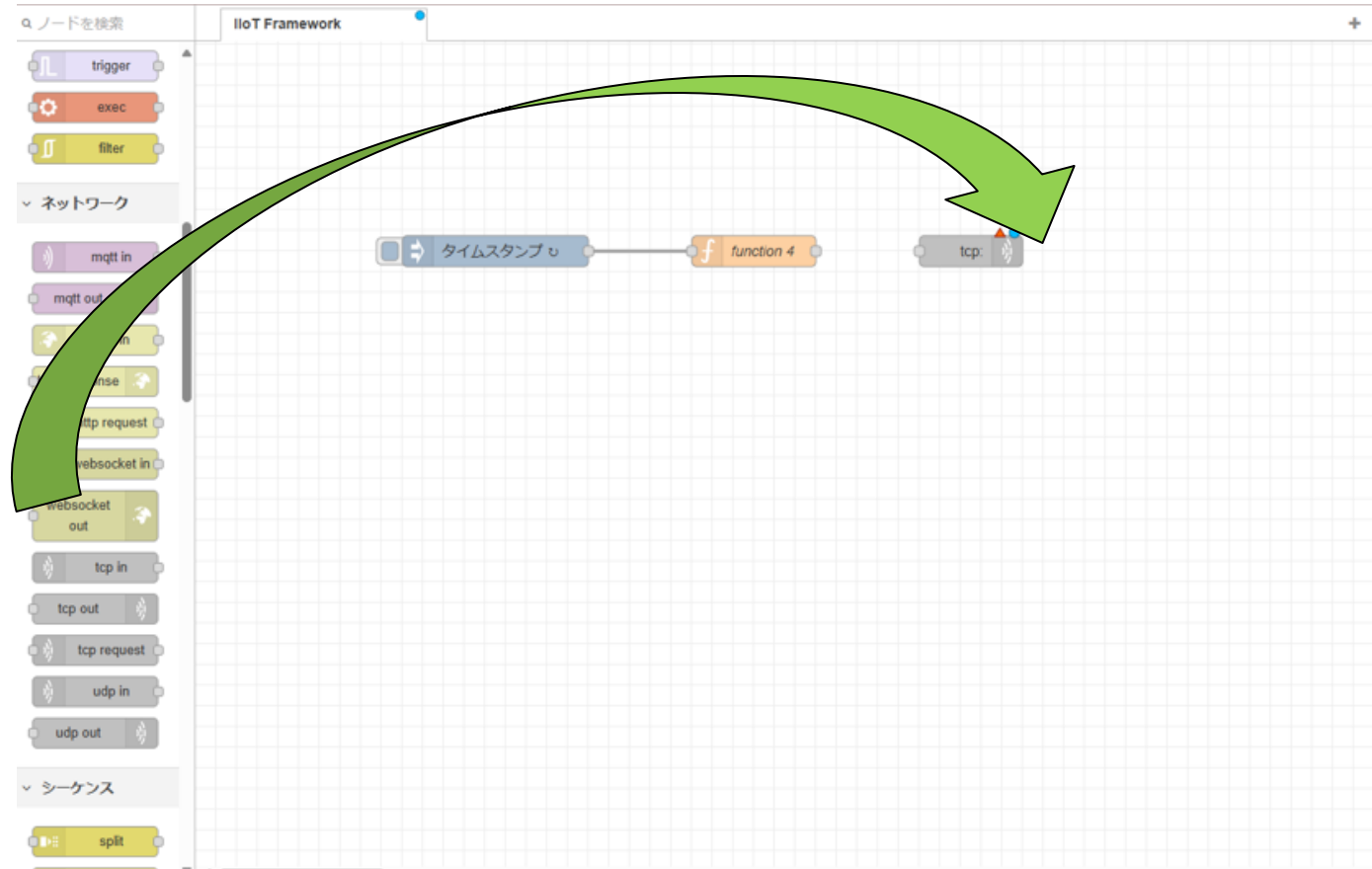
- “function”ノードをクリックし、下記の通りの設定を行います。

```
1 msg.payload = "Hello, TCP/IP.";
2 Return msg;
```



Node-RED側(AXC F2152)のセットアップ

- “tcp out”ノードをドラッグ&ドロップします。



Node-RED側(AXC F2152)のセットアップ

- “tcp out”ノードをクリックし、下記の通りの設定を行います。

送信したい相手先のIPアドレスとポート番号を設定します。
ここでは下記を設定しています。

“ポート”: 12345

“ホスト”: 192.168.1.18

tcp out ノードを編集

削除 中止 完了

プロパティ

種類 接続 ポート 12345

ホスト 192.168.1.18

☐ SSL/TLS接続を有効化

☐ メッセージを送信するたびに接続を切断

☐ Base64メッセージの復号

名前

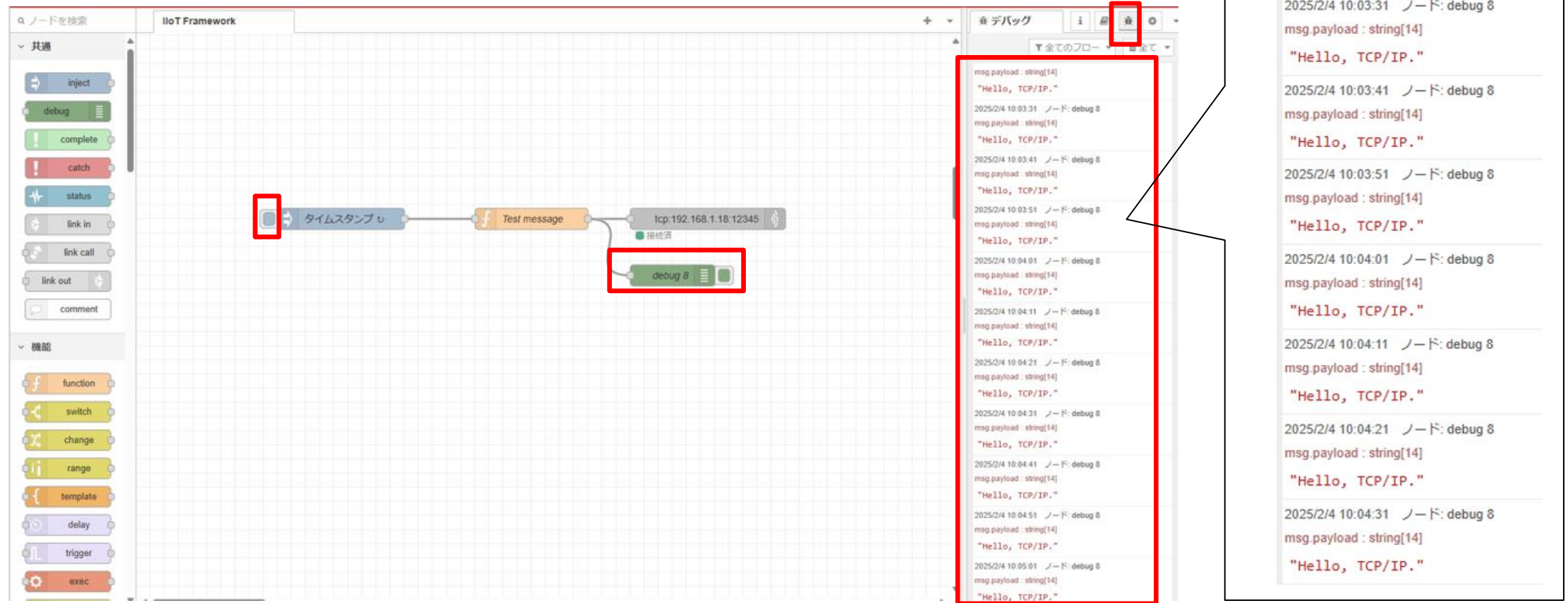
Node-RED側(AXC F2152)のセットアップ

- ノード間を接続します。 問題なければデプロイで保存します。



Node-RED側(AXC F2152)のセットアップ

- “inject(タイムスタンプ)”ノードの左をクリックすることで10秒ごとにメッセージを送信することが確認できます。(debugノード)



The screenshot displays the Node-RED web interface. On the left, the 'inject' node is highlighted with a red box. In the center workspace, a flow is visible: an 'inject' node (labeled 'タイムスタンプ') is connected to a 'function' node (labeled 'Test message'), which is then connected to a 'tcp' node (labeled 'tcp:192.168.1.18:12345'). The 'debug' node (labeled 'debug 8') is also highlighted with a red box. On the right, the 'debug console' shows a list of messages received from the debug node, each containing the text 'Hello, TCP/IP.' and a timestamp.

Timestamp	Node	Message
2025/2/4 10:03:31	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:03:41	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:03:51	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:01	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:11	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:21	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:31	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:41	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:04:51	debug 8	msg.payload : string[14] "Hello, TCP/IP."
2025/2/4 10:05:01	debug 8	msg.payload : string[14] "Hello, TCP/IP."

PC側のセットアップ

- PCでテキストエディタ(ここではVisual Studio Code)を開き、Pythonコードを入力して保存してください。

4 server_socket.bind(("IP address", port No))

“IP address”: ここではPCのIPアドレス
(Node-REDから送信するIPアドレス)

“port No”: Node-REDから送信するポート番号

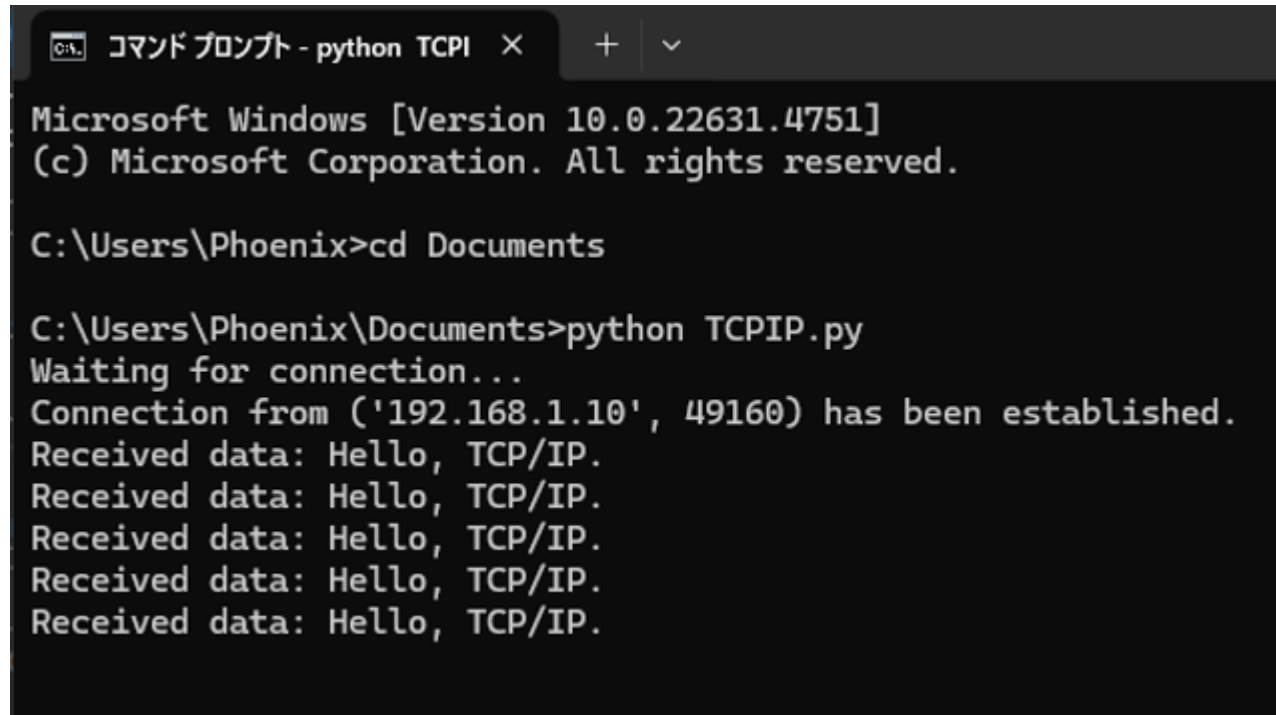
```
TCPIP.py x リリース ノート: 1.96.4
C: > Users > Phoenix > Documents > TCPIP.py > ...
1  import socket
2
3  server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  server_socket.bind(("192.168.1.18", 12345))
5  server_socket.listen(1)
6
7  print("Waiting for connection...")
8
9  client_socket, addr = server_socket.accept()
10 print(f"Connection from {addr} has been established.")
11
12 try:
13     while True:
14         data = client_socket.recv(1024)
15         if not data:
16             break
17         print(f"Received data: {data.decode('ascii')}")
18 except KeyboardInterrupt:
19     print("Interrupted by ctrl + c.")
20 finally:
21     client_socket.close()
22     server_socket.close()
```

デモの実行

- コマンドプロンプトを開き、先ほど記述したスクリプトを実行します。

Python3の実行は"Python "ファイル名".py"になります。

AXC F 2152コントローラからメッセージが送信され、TCP/IP経由でPC側が受け取れていることが分かります。



```
コマンド プロンプト - python TCPI x + v
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Phoenix>cd Documents

C:\Users\Phoenix\Documents>python TCPIP.py
Waiting for connection...
Connection from ('192.168.1.10', 49160) has been established.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
Received data: Hello, TCP/IP.
```

Thank you