



Ethernet スタータキット RX65N

取扱説明書(3)

ルネサス エレクトロニクス社 RX マイコン搭載
HSB シリーズマイコンボード 評価キット

-本書を必ずよく読み、ご理解された上でご利用ください

株式会社 **北斗電子**
REV.1.0.0.0

| | |
|-----------------------------------|----|
| 注意事項 | 1 |
| 安全上のご注意 | 2 |
| 1. マイコン搭載の Ethernet 機能に関して | 4 |
| 2. Ethernet 通信の概要 | 4 |
| 3. ネットワーク接続時の注意点 | 4 |
| 4. Ethernet フレームを送ってみる | 4 |
| 5. ping の応答 | 4 |
| 6. UDP 通信に関して | 4 |
| 7. TCP 通信に関して | 4 |
| 8. LWIP プロトコルスタックの使用 | 5 |
| 8.1. PC(サーバ)に対してのデータ送信(UDP) | 8 |
| 8.2. PC(サーバ)に対してのデータ送信(TCP) | 10 |
| 8.3. PC(クライアント)からのデータ送信(UDP) | 12 |
| 8.4. PC(クライアント)からのデータ送信(TCP) | 12 |
| 8.5. その他の機能 | 13 |
| 9. UDP を使用したファイル転送 | 15 |
| 9.1. PC(サーバ)に対してのデータ送信 | 15 |
| 9.2. PC からマイコンボード(サーバ)に対してのデータ送信 | 19 |
| 9.3. 2 台のマイコンボード間のデータ転送 | 23 |
| 10. TCP を使用したファイル転送 | 27 |
| 10.1. PC(サーバ)に対してのデータ送信 | 27 |
| 10.2. PC からマイコンボード(サーバ)に対してのデータ送信 | 30 |
| 10.3. 2 台のマイコンボード間のデータ転送 | 33 |
| 10.4. TCP のプロジェクト名に関して | 37 |
| 取扱説明書改定記録 | 38 |
| お問合せ窓口 | 38 |

注意事項

本書を必ずよく読み、ご理解された上でご利用ください

【ご利用にあたって】

1. 本製品をご利用になる前には必ず取扱説明書をよく読んで下さい。また、本書は必ず保管し、使用上不明な点がある場合は再読み、よく理解して使用して下さい。
2. 本書は株式会社北斗電子製マイコンボードの使用方法について説明するものであり、ユーザシステムは対象ではありません。
3. 本書及び製品は著作権及び工業所有権によって保護されており、全ての権利は弊社に帰属します。本書の無断複製・複製・転載はできません。
4. 弊社のマイコンボードの仕様は全て使用しているマイコンの仕様に準じております。マイコンの仕様に関しましては製造元にお問い合わせ下さい。弊社製品のデザイン・機能・仕様は性能や安全性の向上を目的に、予告無しに変更することがあります。また価格を変更する場合や本書の図は実物と異なる場合もありますので、御了承下さい。
5. 本製品のご使用にあたっては、十分に評価の上ご使用下さい。
6. 未実装の部品に関してはサポート対象外です。お客様の責任においてご使用下さい。

【限定保証】

1. 弊社は本製品が頒布されているご利用条件に従って製造されたもので、本書に記載された動作を保証致します。
2. 本製品の保証期間は購入戴いた日から1年間です。

【保証規定】

保証期間内でも次のような場合は保証対象外となり有料修理となります

1. 火災・地震・第三者による行為その他の事故により本製品に不具合が生じた場合
2. お客様の故意・過失・誤用・異常な条件でのご利用で本製品に不具合が生じた場合
3. 本製品及び付属品のご利用方法に起因した損害が発生した場合
4. お客様によって本製品及び付属品へ改造・修理がなされた場合

【免責事項】

弊社は特定の目的・用途に関する保証や特許権侵害に対する保証等、本保証条件以外のものは明示・黙示に拘わらず一切の保証は致し兼ねます。また、直接的・間接的損害金もしくは欠陥製品や製品の使用方法に起因する損失金・費用には一切責任を負いません。損害の発生についてあらかじめ知らされていた場合でも保証は致し兼ねます。

ただし、明示的に保証責任または担保責任を負う場合でも、その理由のいかんを問わず、累積的な損害賠償責任は、弊社が受領した対価を上限とします。本製品は「現状」で販売されているものであり、使用に際してはお客様がその結果に一切の責任を負うものとします。弊社は使用または使用不能から生ずる損害に関して一切責任を負いません。

保証は最初の購入者であるお客様ご本人にのみ適用され、お客様が転売された第三者には適用されません。よって転売による第三者またはその為になすお客様からのいかなる請求についても責任を負いません。

本製品を使った二次製品の保証は致し兼ねます。

安全上のご注意

製品を安全にお使いいただくための項目を次のように記載しています。絵表示の意味をよく理解した上でお読み下さい。

表記の意味



取扱を誤った場合、人が死亡または重傷を負う危険が切迫して生じる可能性がある事が想定される



取扱を誤った場合、人が軽傷を負う可能性又は、物的損害のみを引き起こすが可能性がある事が想定される

絵記号の意味

| | | | |
|--|---|--|------------------------------|
| | 一般指示 使用者に対して指示に基づく行為を強制するものを示します | | 一般禁止 一般的な禁止事項を示します |
| | 電源プラグを抜く 使用者に対して電源プラグをコンセントから抜くように指示します | | 一般注意 一般的な注意を示しています |

警告



以下の警告に反する操作をされた場合、本製品及びユーザシステムの破壊・発煙・発火の危険があります。マイコン内蔵プログラムを破壊する場合があります。

1. 本製品及びユーザシステムに電源が入ったままケーブルの抜き差しを行わないでください。
2. 本製品及びユーザシステムに電源が入ったままで、ユーザシステム上に実装されたマイコンまたはIC等の抜き差しを行わないでください。
3. 本製品及びユーザシステムは規定の電圧範囲でご利用ください。
4. 本製品及びユーザシステムは、コネクタのピン番号及びユーザシステム上のマイコンとの接続を確認の上正しく扱ってください。



発煙・異音・異臭にお気づきの際はすぐに使用を中止してください。

電源がある場合は電源を切って、コンセントから電源プラグを抜いてください。そのままご使用すると火災や感電の原因になります。

注意



以下のことをされると故障の原因となる場合があります。

1. 静電気が流れ、部品が破壊される恐れがありますので、ボード製品のコネクタ部分や部品面には直接手を触れないでください。
2. 次の様な場所での使用、保管をしないでください。
ホコリが多い場所、長時間直射日光が当たる場所、不安定な場所、衝撃や振動が加わる場所、落下の可能性がある場所、水分や湿気の多い場所、磁気を発するものの近く
3. 落としたり、衝撃を与えたり、重いものを乗せないでください。
4. 製品の上に水などの液体や、クリップなどの金属を置かないでください。
5. 製品の傍で飲食や喫煙をしないでください。



ボード製品では、裏面にハンダ付けの跡があり、尖っている場合があります。

取り付け、取り外しの際は製品の両端を持ってください。裏面のハンダ付け跡で、誤って手など怪我をする場合があります。



CD メディア、フロッピーディスク付属の製品では、故障に備えてバックアップ（複製）をお取りください。

製品をご使用中にデータなどが消失した場合、データなどの保証は一切致しかねます。



アクセスランプがある製品では、アクセスランプの点灯中に電源を切ったり、パソコンをリセットをしないでください。

製品の故障や、データ消失の原因となります。



本製品は、医療、航空宇宙、原子力、輸送などの人命に関わる機器やシステム及び高度な信頼性を必要とする設備や機器などに用いられる事を目的として、設計及び製造されておりません。

医療、航空宇宙、原子力、輸送などの設備や機器、システムなどに本製品を使用され、本製品の故障により、人身や火災事故、社会的な損害などが生じても、弊社では責任を負いかねます。お客様ご自身にて対策を期されるようご注意ください。

1. マイコン搭載の Ethernet 機能に関して

2. Ethernet 通信の概要

3. ネットワーク接続時の注意点

4. Ethernet フレームを送ってみる

5. ping の応答

上記の内容は、Ethernet スタータキット RX65 取扱説明書(1)で説明しています。

6. UDP 通信に関して

7. TCP 通信に関して

上記の内容は、Ethernet スタータキット RX65 取扱説明書(2)で説明しています。

8. LWIP プロトコルスタックの使用

LWIP は Lightweight IP の略で、マイコンなどの組み込み用に軽量 (CPU 負荷やメモリ使用量を抑えた形) で、TCP/IP のプロトコルの処理を行う事ができるソフトウェアです。

マイコンのソフトの中身は、ソフトウェア編マニュアルで解説していますが、UDP ですとスクラッチ (ゼロから作成) でプログラムを作成しても良いかと思えます。しかし、TCP に関しては、スクラッチでプログラムを作成するのは、ちょっと現実的ではないのかとも思えます。LWIP は、スマート・コンフィグレータで設定して使用可能 (RX65N マイコン向けに用意されたものがある) ですので、使ってみたいと思えます。

マイコンボードには、RX65N_LWIP プロジェクトの mot ファイルを書き込んでください。

マイコンボード側は、TCP サーバ、TCP クライアント、UDP サーバ、UDP クライアントの機能を持たせています。プログラムの動作としては、RX65N_UDP と RX65N_TCP の 2 つのプロジェクトを統合した動作となります。

PC 用のアプリケーションとして、
 PC_APPLI¥UDPTTextServer.exe
 PC_APPLI¥UDPTTextClient.exe
 PC_APPLI¥TCPTTextServer.exe
 PC_APPLI¥TCPTTextClient.exe
 を使用してください。

・IP/MAC アドレス設定



2 台のマイコンボード間で通信を行う場合は、片方の DIP-SW の 3 を OFF、もう一方を ON にしてください。

PC と通信を行う場合は、起動後 10 秒以内に A コマンドを入力してください。

以下、マイコンボード(1)と PC 間で通信を行う内容を説明します。

マイコンボードを起動すると、

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether LwIP sample program.

COMMAND:
--message setting command
m : send text message set(TCP)
r : response text message set(TCP)
M : send text message set(UDP)
p : send text message print
--TCP command
s : send text message(TCP)
c : connect tcp session
d : disconnect tcp session
D : force disconnect tcp session
--UDP command
S : send text message(UDP)
X : udp listen end
Z : udp listen restart

-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )
netmask                    -> 255.255.255.0
gateway IP address        -> 192.168.0.1
communication target IP address -> 192.168.0.81
---

address setting change -> Please input 'A' within 10 seconds.
>
```

上記メッセージが出ます。マイコンボードをサーバとして通信を行わせる場合は、マイコンボードの IP アドレスが 192.168.0.80 (サブネットマスク 255.255.255.0) で問題なければ、特に変更は不要です。

マイコンボードをクライアントとして動かす場合は、通信相手の IP アドレス(communication target IP address -> 192.168.0.81)を変更します。

キーボードから A を入力します。(起動後 10 秒以内)

```
-- IP address settings change --

COMMAND:
1 : this board
2 : netmask
3 : gateway
4 : communication target
5 : TCP server port (default = 20000)
6 : UDP servrt port (default = 30000)
7 : TCP client port (default = 20000)
8 : UDP client port (default = 30000)
p : print setting
e : exit setting
```

本ボードの IP アドレスを変更する場合 1

サブネットマスクの変更 2

ゲートウェイアドレスの変更 3

※同一サブネット内の通信を行う場合は、ゲートウェイアドレスは使用しません

ゲートウェイは、192.168.0.???以外の IP アドレスのホストと通信を行う場合に使用します

(マイコンボードをクライアントとして使用する場合)接続先サーバの IP アドレス 4
(マイコンボードがサーバとして動作する際の)TCP サーバの待ち受けポート番号 5
(マイコンボードがサーバとして動作する際の)UDP サーバの待ち受けポート番号 6
(マイコンボードをクライアントとして使用する場合)接続先サーバの TCP の待ち受けポート番号 7
(マイコンボードをクライアントとして使用する場合)接続先サーバの UDP の待ち受けポート番号 8
現在の設定を確認する場合 p
設定終了 e

の各コマンドで、IP アドレスなどを適切に設定してください。ここでは、通信相手として 192.168.0.97 と通信したいので、4 を入力して接続先の IP アドレスを設定します。

キーボードから 4 を入力します。(※本プログラムでは、通信相手の設定は 4 番です)

```
communication target IP set(please input 3 letters(0-9) [or 1,2 letters(0-9) + Enter] number x 4)
IP[0](b31-24) >192
IP[1](b23-16) >168
IP[2](b15-8) >0
IP[3](b7-0) >97
IP address set to -> 192.168.0.97
```

キーボードから p を入力して設定を確認する。

```
>-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )
netmask                    -> 255.255.255.0
gateway IP address         -> 192.168.0.1
communication target IP address -> 192.168.0.97
---
-setting port-
---
TCP server listen port -> 20000
UDP server listen port -> 30000
TCP client target port -> 20000
UDP client target port -> 30000
---
```

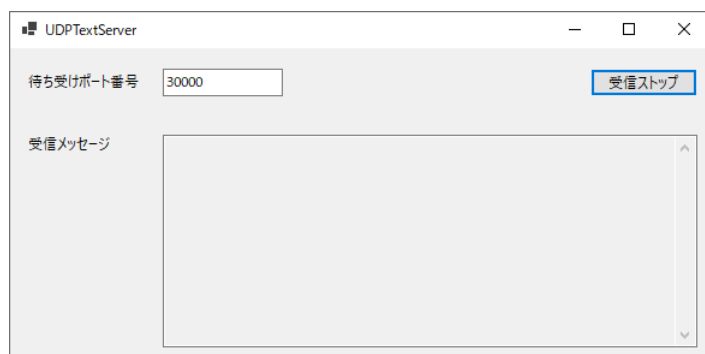
キーボードから e を入力して設定を終了。

```
network operation start!
-- TCP SERVER LISTEN START (port = 20000) --
-- UDP SERVER LISTEN START (port = 30000) -----
```

ネットワークの動作が開始され、マイコンボードはポート番号 20000 で TCP の接続待ち。ポート番号 30000 で UDP のデータ受信待ちとなります。

8.1. PC(サーバ)に対してのデータ送信(UDP)

PC_APPLI\UDPTextServer.exe を起動して、「スタート」ボタンを押します。



マイコンボード側は、

```
COMMAND:
--message setting command
m : send text message set(TCP)
r : response text message set(TCP)
M : send text message set(UDP)
p : send text message print
--TCP command
s : send text message(TCP)
c : connect tcp session
d : disconnect tcp session
D : force disconnect tcp session
--UDP command
S : send text message(UDP)
X : udp listen end
Z : udp listen restart
```

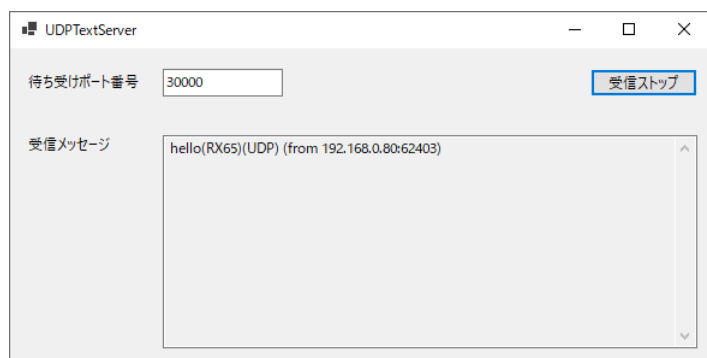
上記のコマンドが使用できます。

UDP の場合、

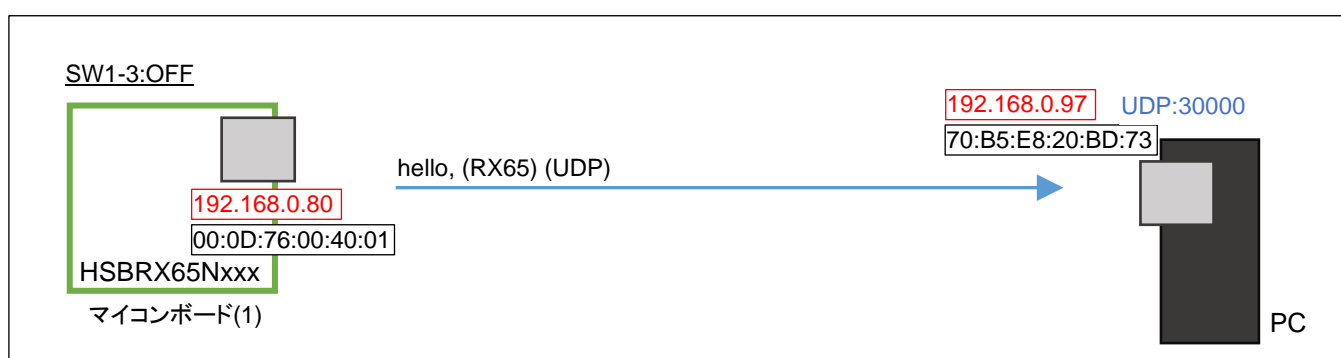
S コマンド(大文字 S)でメッセージの送信
です。

s コマンド

```
>command=S
UDP send text message to 192.168.0.97 port 30000
```



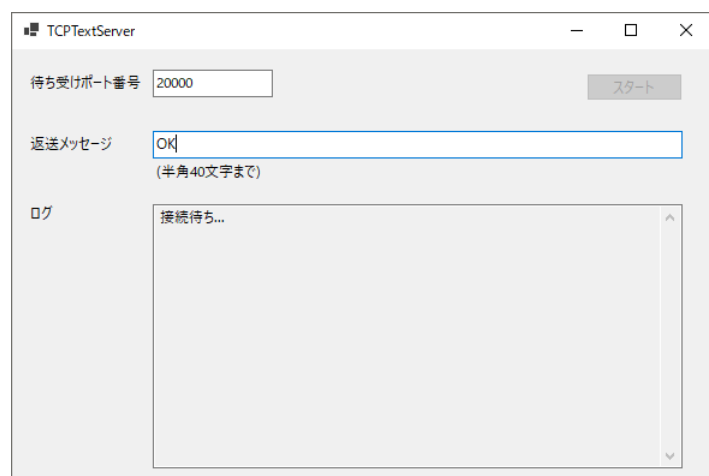
PC 側のアプリケーションで、受信メッセージが確認できるはずですが。



ここでは、マイコンボード側から UDP でテキストメッセージを送っています。

8.2. PC (サーバ) に対してのデータ送信(TCP)

PC_APPLI¥TCPTextServer.exe を起動して、「スタート」ボタンを押します。



マイコンボード側は、

```
COMMAND:
--message setting command
m : send text message set(TCP)
r : response text message set(TCP)
M : send text message set(UDP)
p : send text message print
--TCP command
s : send text message(TCP)
c : connect tcp session
d : disconnect tcp session
D : force disconnect tcp session
--UDP command
S : send text message(UDP)
X : udp listen end
Z : udp listen restart
```

上記のコマンドが使用できます。

TCP の場合、

c コマンドで接続(セッションの確立)
s コマンドでメッセージの送信
です。

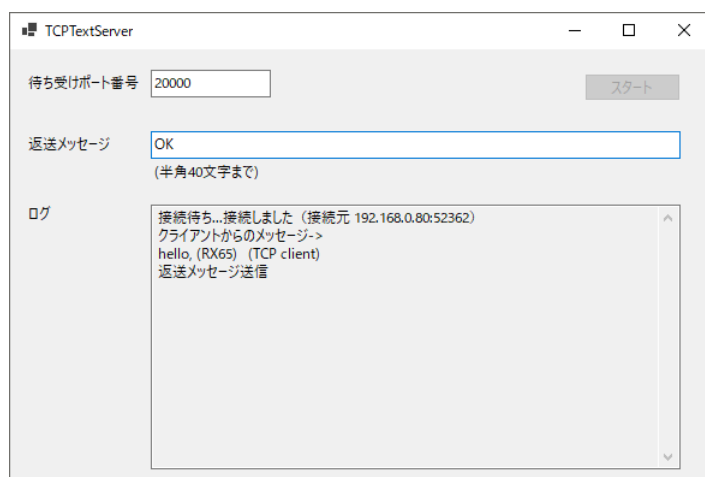
c コマンド

```
>command=c
connect 192.168.0.97 port 20000
-- TCP CONNECTION ESTABLISHED --
```

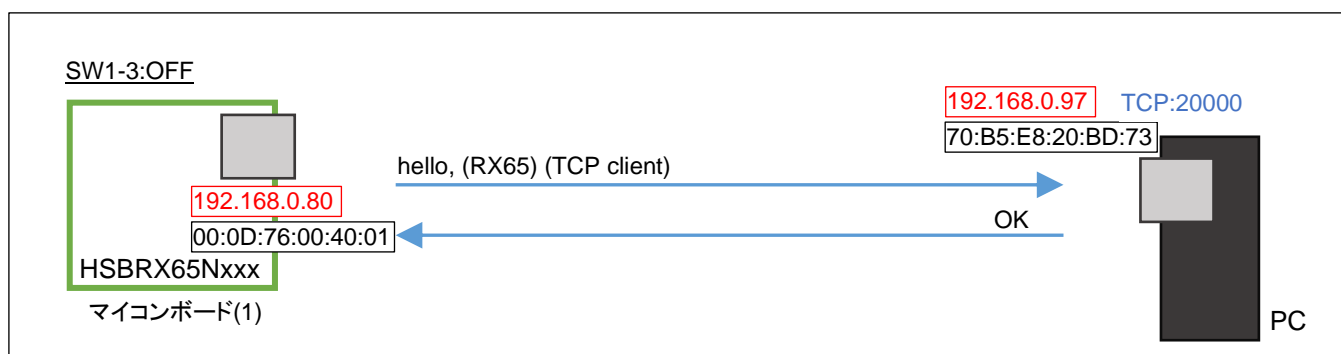
上記表示となった場合は、接続成功です。(TCP CONNECTION ESTABLISHED が出ない場合は、PC や通信相手のマイコンボード側で 20000 で待ち受けしているかを確認してください。

s コマンド

```
>command=s
TCP send text message to 192.168.0.97 port 20000
send message.
message received (TCP) ->
OK
```



PC 側のアプリケーションで、受信メッセージが確認できるはずですが。



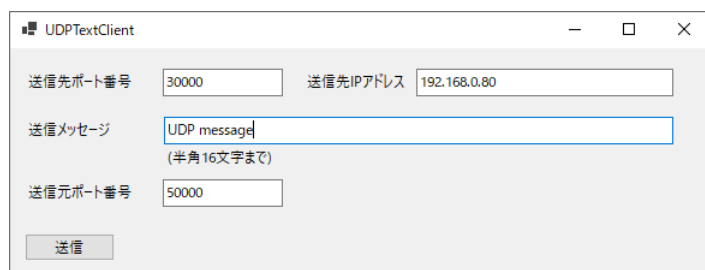
データとしては、上記の様なメッセージのやり取りを行っています。

7章のプログラムでは、マイコンボードを TCP クライアントとして動作させた場合、送信元ポート番号は、50000 としていましたが、本プログラムでは送信元のポート番号はプログラムで明示的に指定はしていません。(ランダムな番号が使われます)

一般的には、サーバ側はポート番号が決まっていなくて、通信相手(クライアント側)がどのポートに接続を行うかを決められないので、サーバ側の待ち受けポート番号は固定。それに対しクライアントの使用ポート番号は、接続を確立する際に、TCP ヘッダ内に送信元ポート番号の情報が含まれます。サーバ側は「IP ヘッダ内の送信元 IP アドレス:TCP ヘッダ内の送信元ポート番号」に対して接続を確立すれば良いので、クライアント側のポート番号は固定せずにランダムな番号を使用するのが一般的です。

8.3. PC(クライアント)からのデータ送信(UDP)

PC_APPLI¥UDPTextClient.exe を起動して、送信メッセージを入力して「送信」ボタンを押します。



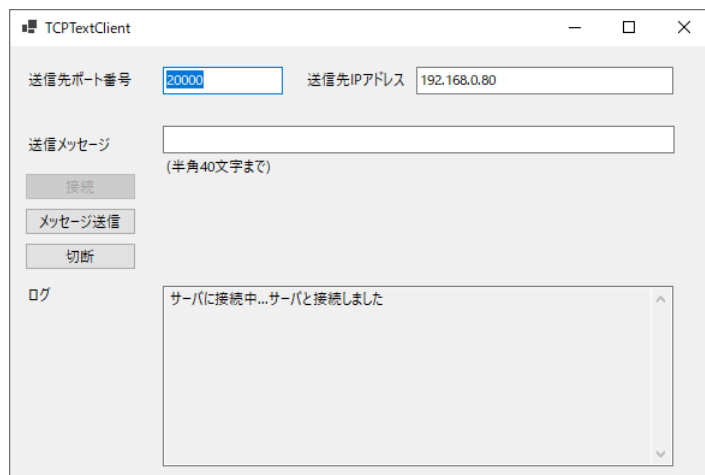
マイコンボード側は、

```
message received (UDP) ->
UDP message      (from 192.168.0.97:50000)
```

メッセージを受信した際に端末に表示します。

8.4. PC(クライアント)からのデータ送信(TCP)

PC_APPLI¥TCPTTextClient.exe を起動して、「接続」ボタンを押します。



マイコンボード側は、

```
-- TCP CONNECTION ACCEPTED --
(connection from 192.168.0.97:60183)
```

TCP の接続を受け付けた際には、上記表示が出ます。



送信メッセージを入力して、「メッセージ送信」ボタンを押します。

マイコンボード側は、

```
message received (TCP) ->
TCP client message
send response.
```

メッセージを受信して、レスポンス"OK (RX65)"を返送します。

6章で説明したUDP通信と7章で説明したTCP通信の、両方の機能を持たせているのが、RX65N_LWIPのプロジェクトです。ソフトウェアの説明は、ソフトウェア編マニュアルに記載していますが、LWIPのプロトコルスタックを使用すると、7章で説明しているTCPのプログラムに対して、比較的容易にTCP通信を実現する事が可能です。

8.5. その他の機能

マイコンボードのSW2を押すと、

```
ping request send to 192.168.0.97
-> ping reply received from 192.168.0.97
```

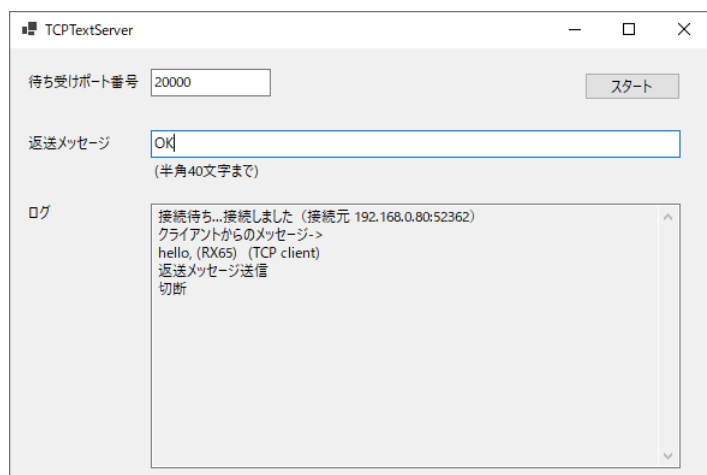
接続先に設定したIPアドレスにpingを実行します。上記は、pingの応答が返ってきた場合の表示です。

dコマンドは、

```
>command=d
disconnect
-- TCP CONNECTION CLOSED --
```

マイコンボードがTCPクライアントとして接続しているサーバから切断します。

TCPTextServer 側では、「切断」の表示が出ます。



この状態では、TCP のセッションは終了しているので、再度 c コマンドで接続する事が可能です。

X コマンドは、

```
>command=X
udp listen -> end
-- UDP SERVER LISTEN END --
```

UDP サーバとしての動作を終了します。

Z コマンドは、

```
udp listen -> restart
-- UDP SERVER LISTEN START (port = 30000) --
```

UDP サーバとしての動作を再開します。

p コマンドは、

```
>command=p
send text message is ->
TCP client : "hello, (RX65) (TCP client)          "
TCP server : "OK, (RX65) (TCP server)           "
UDP : "hello(RX65)(UDP)"
```

TCP クライアントで送信するメッセージ。TCP サーバで返送するメッセージ。UDP クライアントで送信するメッセージの設定を表示します。それぞれ、

m TCP クライアントで送信するメッセージを設定

r TCP サーバで返信するメッセージを設定

M UDP クライアントで送信するメッセージを設定

m, r, M のコマンドでメッセージの内容は変更可能です。

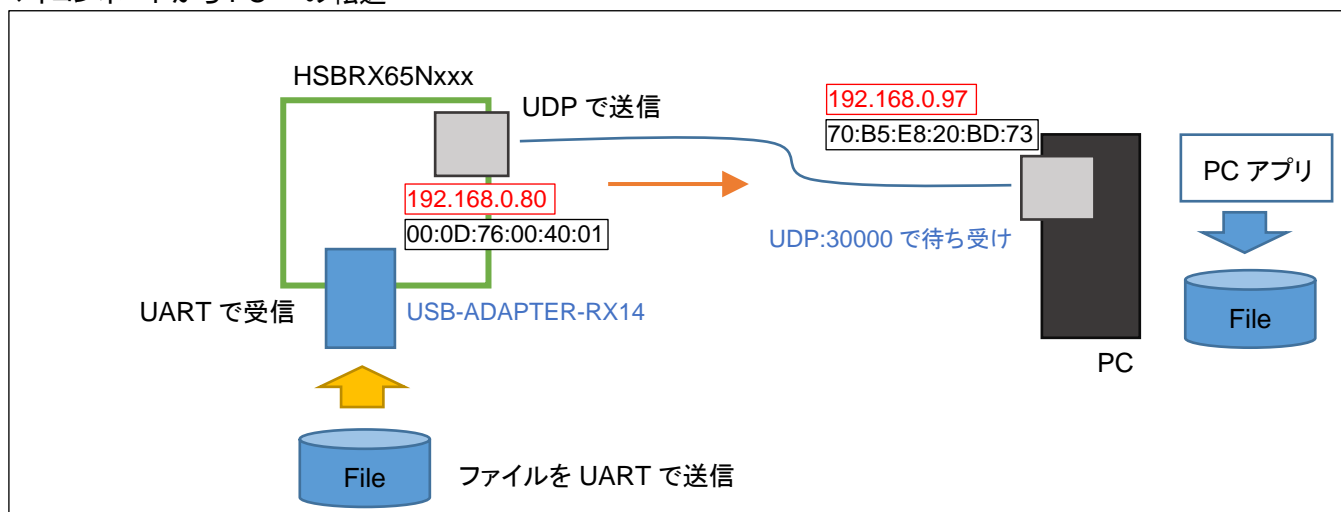
9. UDP を使用したファイル転送

マイコンボードには、RX65N_UART_ETHER_UDP プロジェクトの mot ファイルを書き込んでください。

9.1. PC (サーバ) に対してのデータ送信

PC 側では、PC_APPLI¥UDPFileReceive.exe を実行してください。

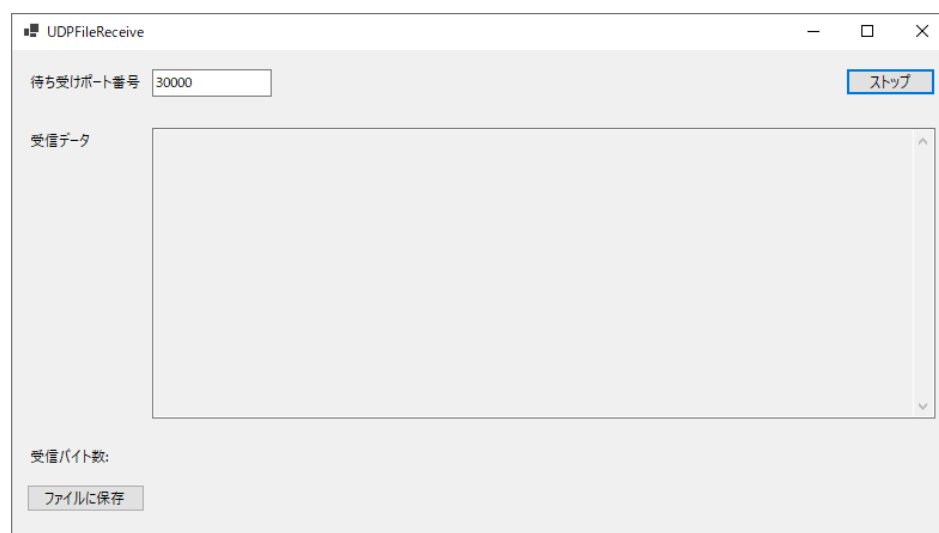
・マイコンボードから PC への転送



マイコンボードの UART で受信したデータを、Ethernet 経由で UDP を使って通信を行い、PC 上でファイルに保存します。

マイコンボードの UART にデータを送る際は、ここでは teraterm の File – Send file 機能を使います。

PC 側で UDPFileReceive.exe を実行し、「スタート」を押します。



マイコンボード側は、起動後デフォルトでは通信先は、192.168.0.81 です。通信先を変える場合は、マイコンボード上の SW3 を押しながりセットを押します。

(起動したら、SW3 は離して良いです。数秒は押したままにしてください。)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.  
RX65N Ether UART Ether UDP sample program.
```

```
-- IP address settings change --
```

```
COMMAND:
```

```
1 : this board  
2 : netmask  
3 : gateway  
4 : communication target  
5 : UDP servrt port (default = 30000)  
6 : UDP client port (default = 30000)  
p : print setting  
e : exit setting
```

IP アドレス変更メニューが出ますので、通信先を変える場合は 4 を押して、IP アドレスの設定を行ってください。ここでは、192.168.0.97 に設定します。

```
>communication target IP set(please input 3 letters(0-9) [or 1,2 letters(0-9) + Enter] number x 4)
```

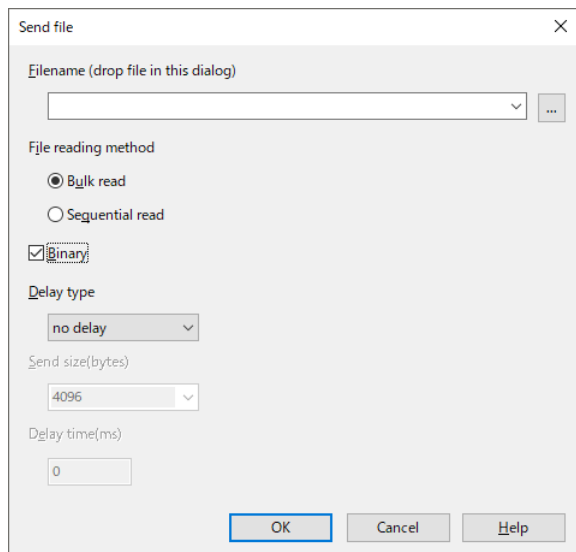
```
IP[0](b31-24) >192  
IP[1](b23-16) >168  
IP[2](b15-8) >0  
IP[3](b7-0) >97  
IP address set to -> 192.168.0.97
```

e を押して設定終了です。

```
-setting address-  
---  
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )  
netmask                    -> 255.255.255.0  
gateway IP address        -> 192.168.0.1  
communication target IP address -> 192.168.0.97  
---  
-- UDP SERVER LISTEN START (port = 30000) --
```

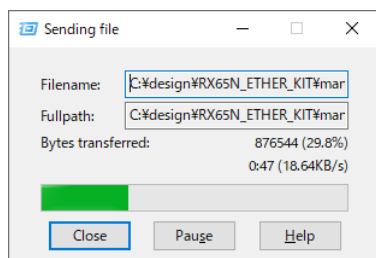
この状態で通信の準備が完了しています。

マイコンボード側では、teraterm で端末を開いているので、File-Send file を選択します。teraterm のバージョンによって表示は異なりますが、



テキストファイルやソースファイル以外のファイルを送る場合は、Binary にチェックを入れてください。
 (teraterm には transfer として、XMODEM や ZMODEM でファイルを送信する機能もありますが、ここではデータ変換せずにバイナリデータを扱いますので、Send file の機能を使ってください。)

teraterm では、ファイル送信のダイアログが表示されます。

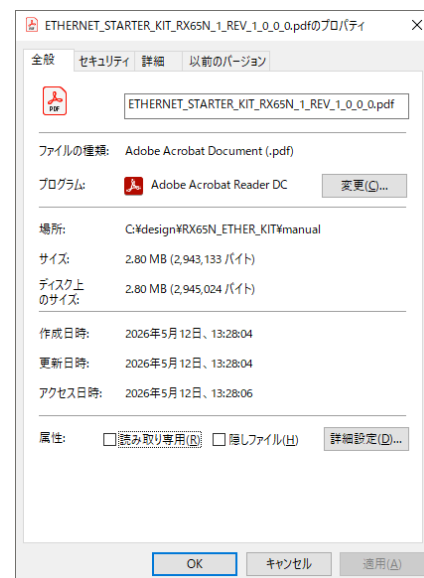
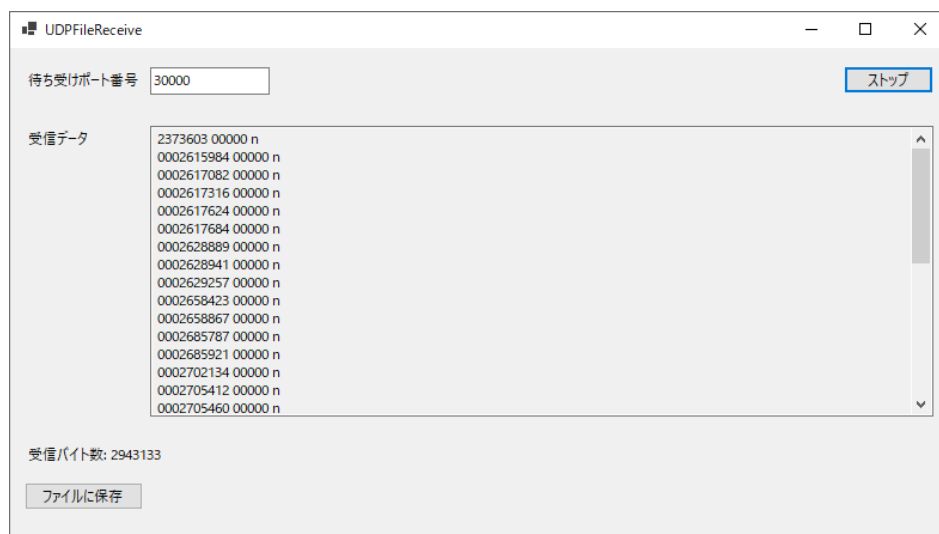


マイコンボード側では

```
-- 1146 bytes send --
-- 1147 bytes send --
-- 1146 bytes send --
-- 1146 bytes send --
```

上記の様な表示が出ます。100ms 毎に受信したデータを、UDP のパケットに乗せて送り出しています。

PC 側のアプリケーションでは、



現在受信しているデータと受信バイト数が表示されます。(今回送信したのは PDF ファイルですが、たまたま PDF の最後の方がテキストだったので文字化けせずに見えています。バイナリの場合は、表示は文字化けします。)

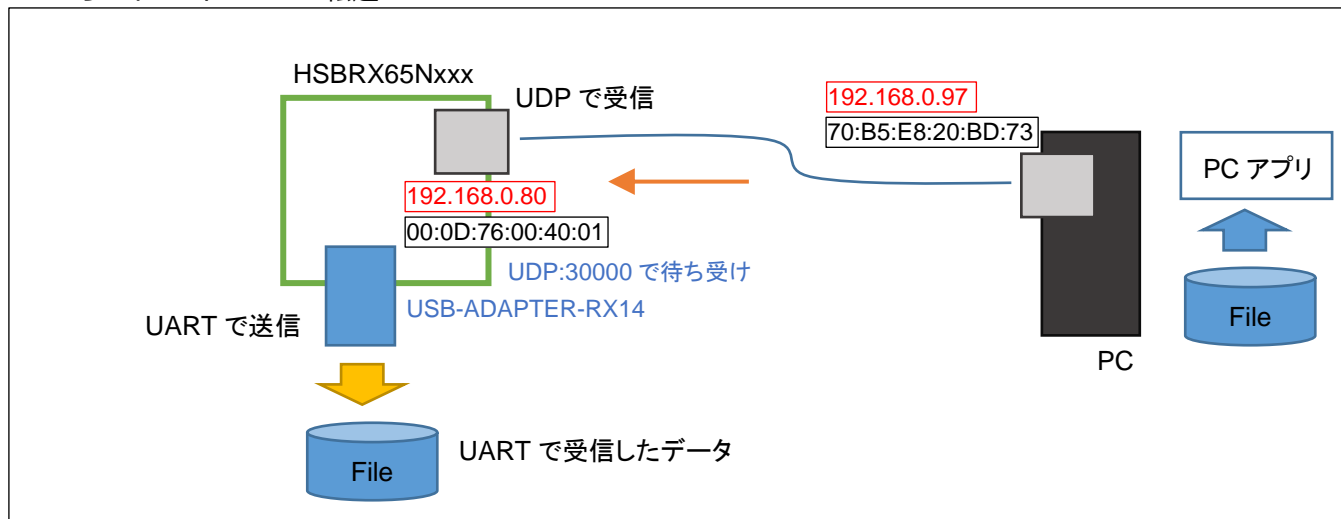
送ったのが右にプロパティを表示した PDF ファイルですが、ファイルサイズは同じになっています。PC のアプリで、「ファイルに保存」を押して、適当なファイル名で保存します。

(ファイルに保存後、PDF として開くことが出来ました。また、バイナリでコンペアを取りましたが、一致しています。)

9.2. PC からマイコンボード(サーバ)に対してのデータ送信

PC 側では、PC_APPLI¥UDPFileSend.exe を実行してください。

・PC からマイコンボードへの転送



PC 側でファイルを Ethernet 経由で UDP を使って通信を行い、マイコンボード側から UART で送信します。teraterm を使用して受信したデータをファイルに保存します。

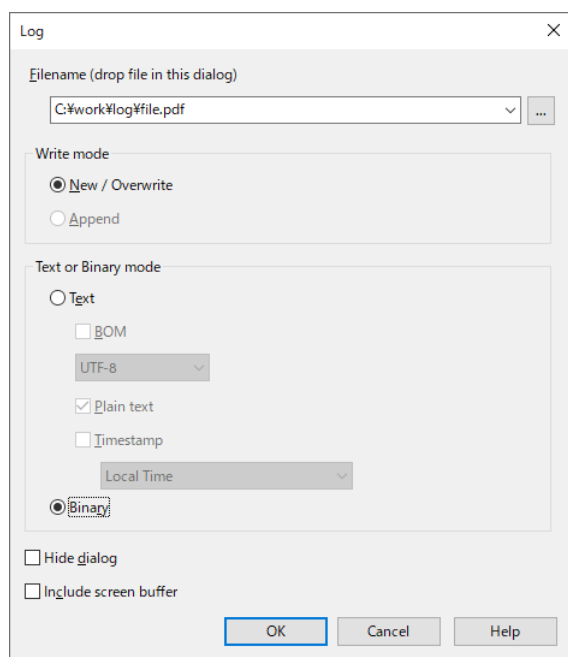
このケースでは、マイコンボード側が UDP のサーバ動作となりますので、マイコンボードの IP アドレスが 192.168.0.80 で問題なければ、変更の必要はありません。

マイコンボード側では

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether UDP sample program.
-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-1)
netmask                    -> 255.255.255.0
gateway IP address         -> 192.168.0.1
communication target IP address -> 192.168.0.81
---
-- UDP SERVER LISTEN START (port = 30000) --
```

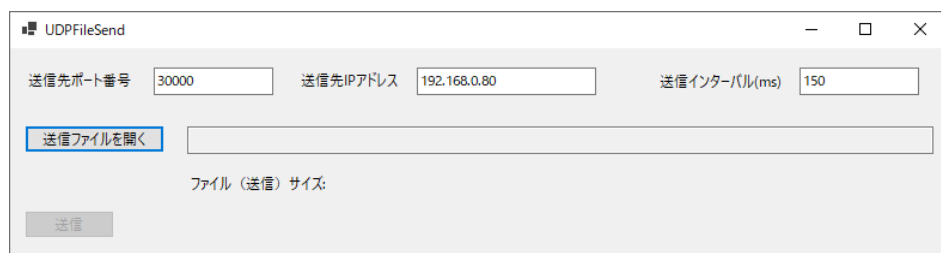
UDP で、30000 番のポートでデータを待ち受けしていますので、外部から 192.168.0.80:30000 にデータを送信すれば良いです。

受信データをファイルに落とすのに、teraterm の File – Log 機能を使います。

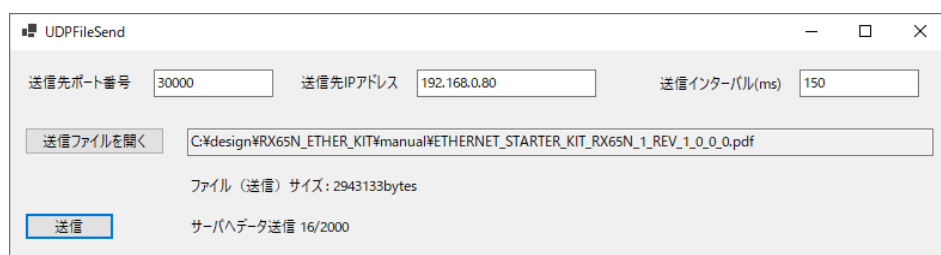


適当なファイル名を設定し、バイナリファイルの場合は、Binary をチェックして、OK を押してください。

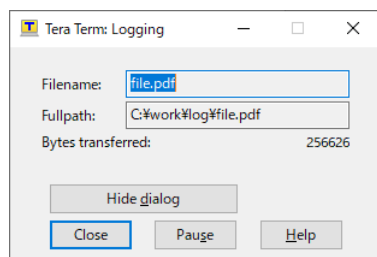
PC 側では、PC_APPLI\UDPFileSend.exe を起動して、



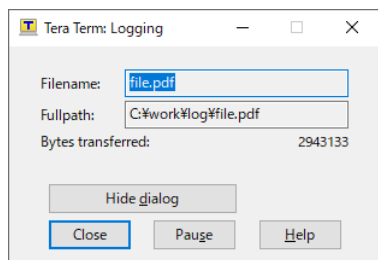
「送信ファイルを開く」ボタンで適当なファイルを開いて「送信」ボタンを押してください。



teraterm の方は、



上記の様なダイアログが表示され、Bytes transferd:の数値が変わっていれば、データを受信しています。



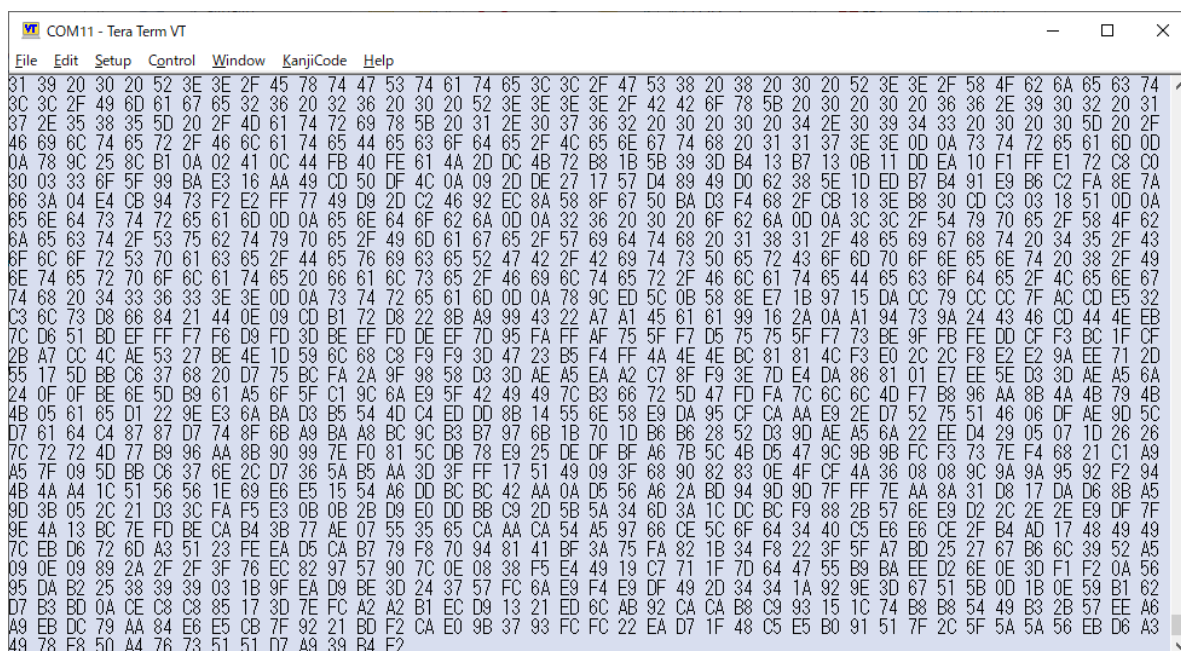
ファイルの送信が終わると、Bytes transferd:の数値も止まりますので、Close を押してください。

なお、teraterm でバイナリファイルの受信を行った場合、



上記の様に表示が文字化けして、(画面表示の負荷が大きく?)結果的に受信データを取りこぼす様です。

teraterm で、バイナリデータを受信する際は、「デバッグモードを有効化」、「Shift-ESC でデバッグモードに入れる (hex フォーマットで表示)」、



とすると、受信データの取りこぼしがなくなり、送信したファイルとバイナリレベルで一致する様になります。

※必ずしも、「送信ファイル」=「受信ファイル」となるとは限りません。PC 側で、SSD のプチフリーズが起こったり、何か重たい処理が走ったりすると、データが欠落する事があります。本プログラムでは、単純にバイナリデータの送信 (UDP を使った一方的な送り付け) としているのでデータの信頼性という観点では低いです。実用的なアプリケーションでは、ブロック単位で CRC やチェックサムを付加した形で送信し、受信側で一致しない場合は再送するなどの工夫が必要になると考えます。

UDPFileSend.exe には、送信先 IP アドレスの設定の他に、送信インターバル「150ms」という項目があります。このタイミングで、データを送信しています。1 回に 1472 バイト送って 150ms 待つという動作です。

マイコンボード側は、データを UART で送信しているので、UART の設定速度(115,200bps)を考慮したタイミングとしています。(送信インターバルは「送信」ボタンを押したタイミングで読み込むので、変更したい場合は送信前に変えてください。)

なお、UDPFileSend/UDPFileReceive で取り扱うファイルサイズは 10MB までとしています。この値は、C#のソース内に書かれている値なので、変更する場合はソースを 1 行変更してビルドすると変えられます。

※teraterm でデバッグモードを有効化する方法

(1)TERATERM.INI を書き換える

TERATERM.INI

```
; Display all characters (debug mode)
; Debug=off
Debug=on
; Debug mode type which can be selected by user.
; on|all = All types
; off|none = Disabled debug mode
; normal = usual teraterm debug mode
; hex = hex output
; noout = disable output completely
; DebugModes=all
DebugModes=hex
```

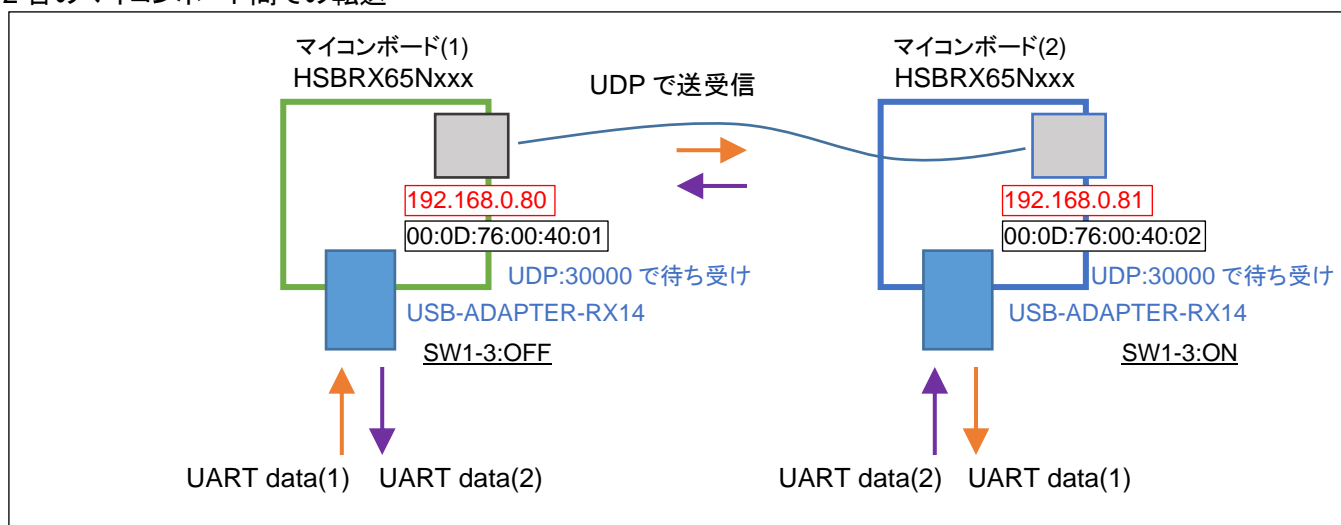
Debug=on, DebugModes=hex(推奨)としてください。

(2)teraterm の画面を選択し「Shift+ESC」を押す

Shift+ESC を押す度にデバッグモードがトグルで切り替わります。DebugModes=hex とした場合は、Shift+ESC で通常←→HEX 表示が切り替わります。複数のデバッグモードを有効化した場合は、Shift+ESC を押す度に次の表示モードに切り替わります。

9.3. 2 台のマイコンボード間のデータ転送

・2 台のマイコンボード間での転送



2 台のボードに同じプログラムを書き込み、DIP-SW の 3 を一方を OFF、もう一方を ON に設定します。その場合は、UART-Ethernet-UART のブリッジとして動作します。

○マイコンボード(1)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether UDP sample program.
-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )
netmask                    -> 255.255.255.0
gateway IP address        -> 192.168.0.1
communication target IP address -> 192.168.0.81
---
-- UDP SERVER LISTEN START (port = 30000) --
```

○マイコンボード(2)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether UDP sample program.
-setting address-
---
this board IP address      -> 192.168.0.81 (MAC address -> 00-0D-76-00-40-02 )
netmask                    -> 255.255.255.0
gateway IP address        -> 192.168.0.1
communication target IP address -> 192.168.0.80
---
-- UDP SERVER LISTEN START (port = 30000) --
```

マイコンボード(1)側の端末から、キーボードで abcdef を入力してみます。

○マイコンボード(1)

```
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --
```

○マイコンボード(2)

```
abcdef
```

今度は、マイコンボード 2 から、123 を入力してみます。

○マイコンボード(2)

```
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --
```

○マイコンボード(1)

```
123
```

双方、どちらからでも通信が通る様になっています。

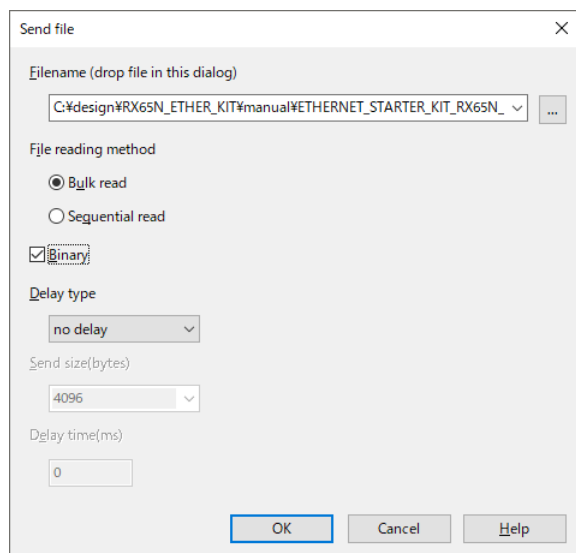
※--- xx bytes send ---表示はデバッグ用に表示する様になっています。

ether_common.h 内の#define SCI_DEBUG をコメントアウトしてプログラムをビルドすると、メッセージ類は一切表示されず、純粋な UARTブリッジとして動作します。(プログラムの変更に関しては、ソフトウェア編マニュアルに記載しています。)

teraterm を使って、バイナリファイル転送を試してみます。

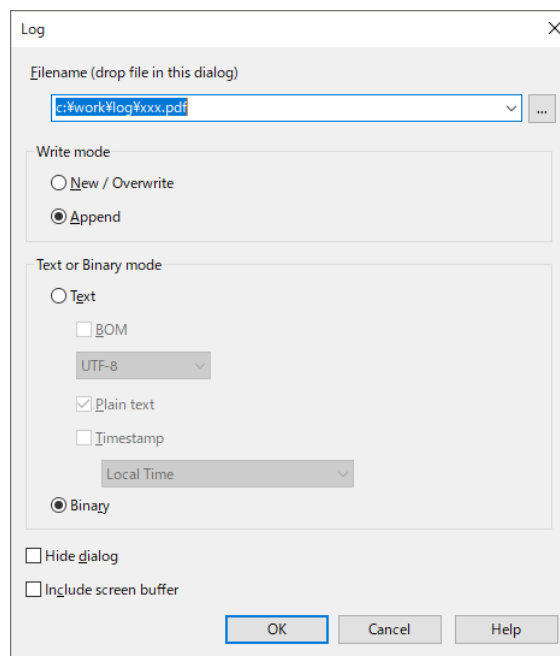
マイコンボード(1)(送信側)

File – Send File



マイコンボード(2)(受信側)

File – log



(バイナリファイルの送受信を行う場合は、どちらも Binary にチェックを入れる)

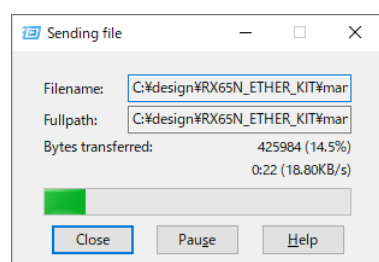
○マイコンボード(1)(送信側)

```
-- 1147 bytes send --
-- 1146 bytes send --
-- 1146 bytes send --
```

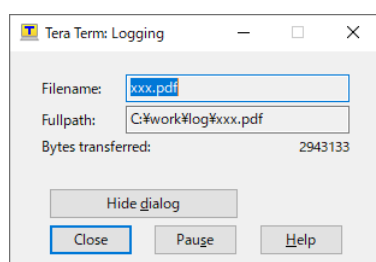
○マイコンボード(2) (受信側) ※hex 表示モードに設定(重要)

```
32 35 37 20 30 30 30 30 30 20 6E 0D 0A 30 30 30 32 36 35 38 34 32 33 20 30 30 30 30 30 20 6E 0D 0A 30 30 30
32 36 35 38 38 36 37 20 30 30 30 30 30 20 6E 0D 0A 30 30 30 32 36 38 35 37 38 37 20 30 30 30 30 30 20 6E 0D
0A 30 30 30 32 36 38 35 39 32 31 20 30 30 30 30 30 20 6E 0D 0A 30 30 30 32 37 30 32 31 33 34 20 30 30 30 30
30 20 6E 0D 0A 30 30 30 32 37 30 35 34 31 32 20 30 30 30 30 30 20 6E 0D 0A 30 30 30 32 37 30 35 34 36 30 20
30 30 30 30 30 20 6E 0D 0A 74 72 61 69 6C 65 72 0D 0A 3C 3C 2F 53 69 7A 65 20 31 30 38 33 36 2F 52 6F 6F 74
20 31 20 30 20 52 2F 49 6E 66 6F 20 35 31 30 20 30 20 52 2F 49 44 5B 3C 31 31 33 31 30 38 43 30 31 43 37 38 35
33 34 32 42 33 33 31 39 34 38 45 43 38 39 42 39 39 37 42 3E 3C 31 31
```

送信中



受信完了



この場合も、バイナリレベルの一致が確認できました。

上記では、マイコンボード(1)→マイコンボード(2)のファイル転送を行いました。逆も動作します。

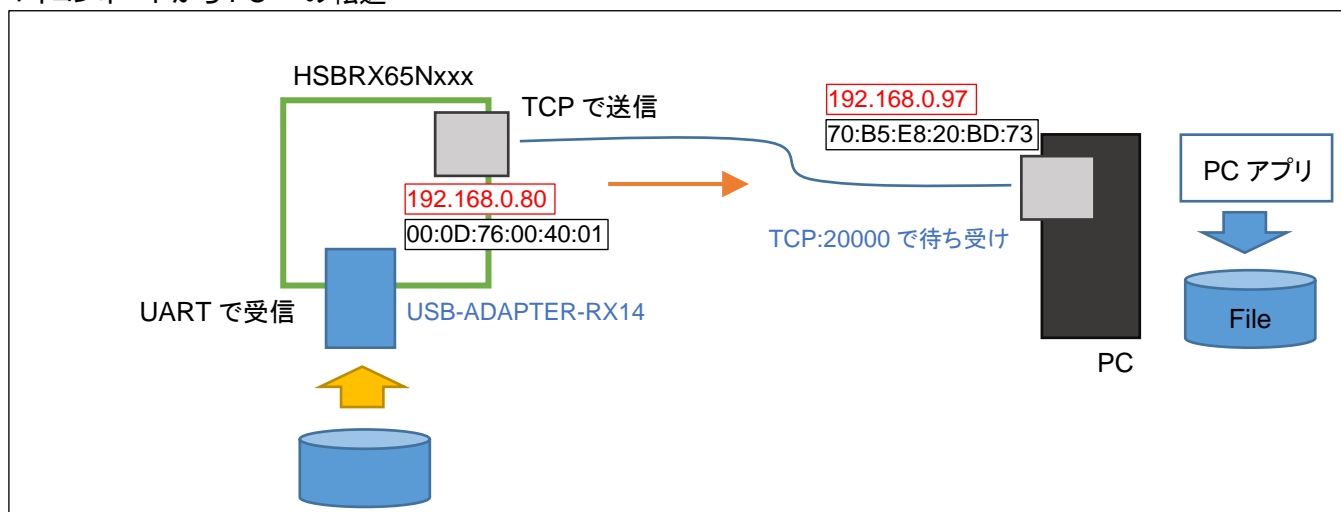
10.TCP を使用したファイル転送

マイコンボードには、RX65N_UART_ETHER_TCP_DMACH プロジェクトの mot ファイルを書き込んでください。

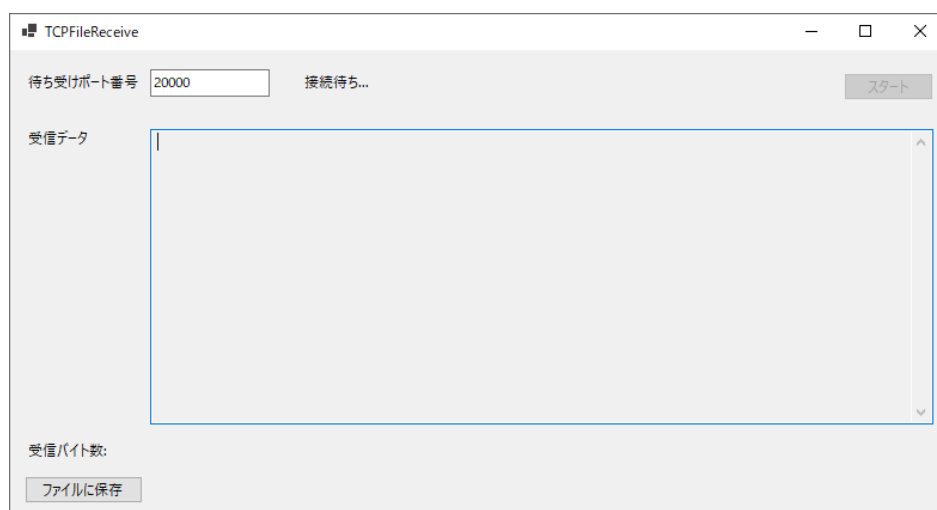
10.1.PC(サーバ)に対してのデータ送信

PC 側では、PC_APPLI¥TCPFileReceive.exe を実行してください。

・マイコンボードから PC への転送



PC 側で TCPFileReceive.exe を実行し、「スタート」を押します。



マイコンボード側はデフォルトは、通信先 IP アドレス=192.168.0.81 です。変更する場合は、SW3 を押しながリセ
ットを押してください。(SW3 は数秒間押したまま)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether TCP sample program (with SCI1-TX->DMAC).
```

```
-- IP address settings change --
```

```
COMMAND:
```

```
1 : this board
2 : netmask
3 : gateway
4 : communication target
5 : TCP servrt port (default = 20000)
6 : TCP client port (default = 20000)
p : print setting
e : exit setting
```

4 で通信先の IP アドレスを設定してください。

e で設定を終了すると、以下の様な表示となります。

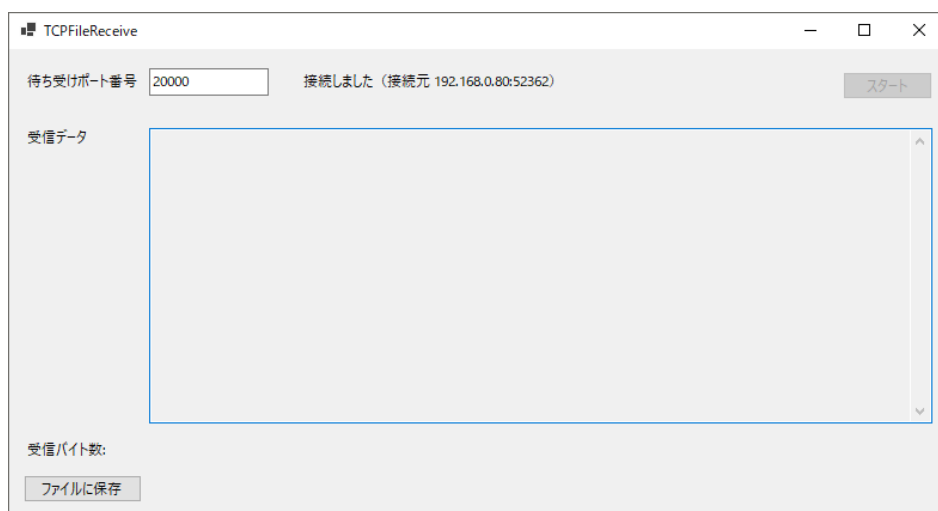
```
-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )
netmask                   -> 255.255.255.0
gateway IP address        -> 192.168.0.1
communication target IP address -> 192.168.0.97
---
-- TCP SERVER LISTEN START (port = 20000) --
```

従来は c コマンドで接続でしたが
本プログラムでは SW3 で接続です

ここで、マイコンボードから PC で動いているサーバに接続するために、マイコンボード上の SW3 を押します。

```
-- TCP CONNECTION ESTABLISHED --
```

という表示が出れば、PC で動いているアプリ(TCPFileReceive.exe)に接続完了です。



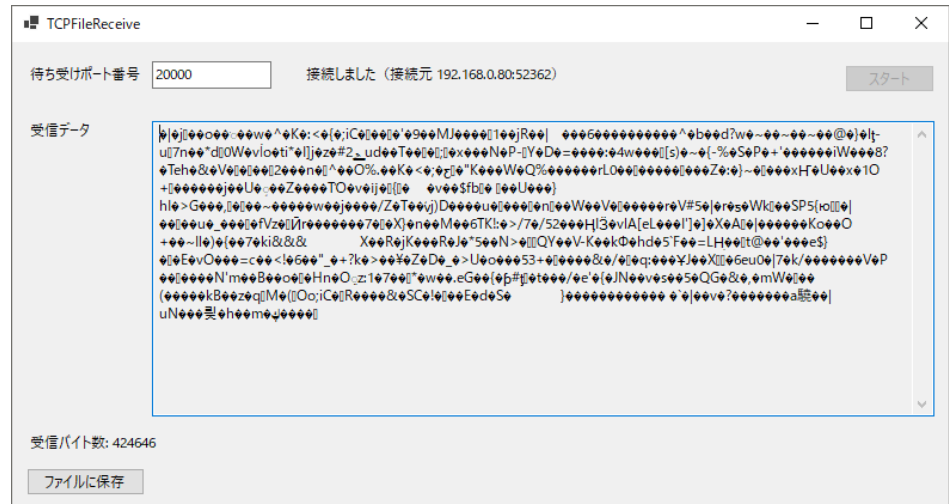
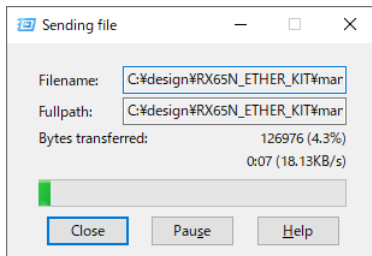
上の方に「接続しました」という表示が出ます。

マイコンボード側の端末の、teraterm で、File – Send file を実行します。

(バイナリファイルを送る場合は、Binary にチェックを入れてください)

送信側

受信側

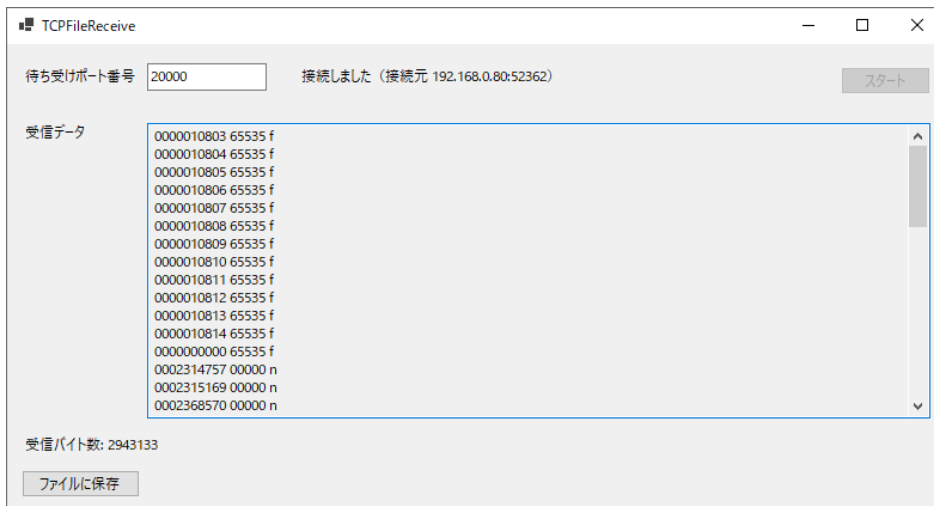


バイナリファイルの場合は、表示は化けますが、データは受信されています。

マイコンボードの端末には、

```
-- 1146 bytes send --
-- 1147 bytes send --
```

の表示が出ます。(デバッグモードをオフにするとメッセージは表示されなくなります。デフォルトではデバッグモード=オンです)



転送が終わると、PC アプリ側で、受信バイト数の値が増えなくなります。「ファイルに保存」で受信データをファイルに保存する事ができます。

動作としては、UDP を使った場合と同じです。データを送る際にどちらのプロトコルを使っているかだけの違いで、アプリケーションレベルでの動作に相違はありません。

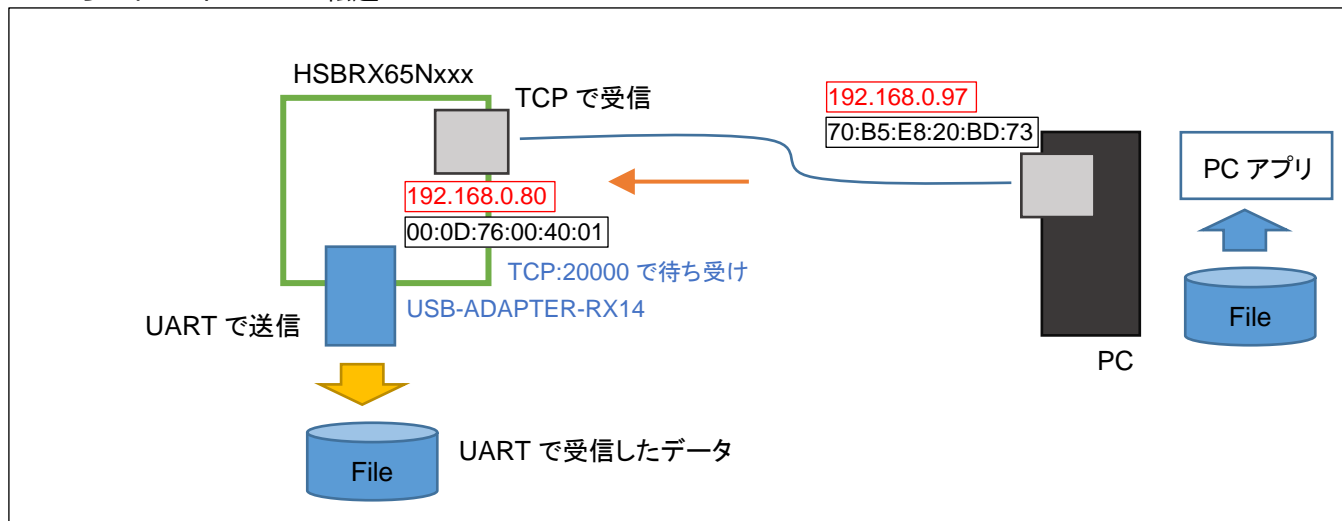
但し、マイコンボードを TCP クライアントとして動作させる場合は、接続(コネクションを張る)動作として、SW3 を最初に 1 度押す必要があります(PC 側で接続待ち状態にした後)。

この点は、コネクションレスで動作する UDP のプログラムと異なる点です。

10.2.PC からマイコンボード(サーバ)に対してのデータ送信

PC 側では、PC_APPLI¥TCPFileSend.exe を実行してください。

・PC からマイコンボードへの転送



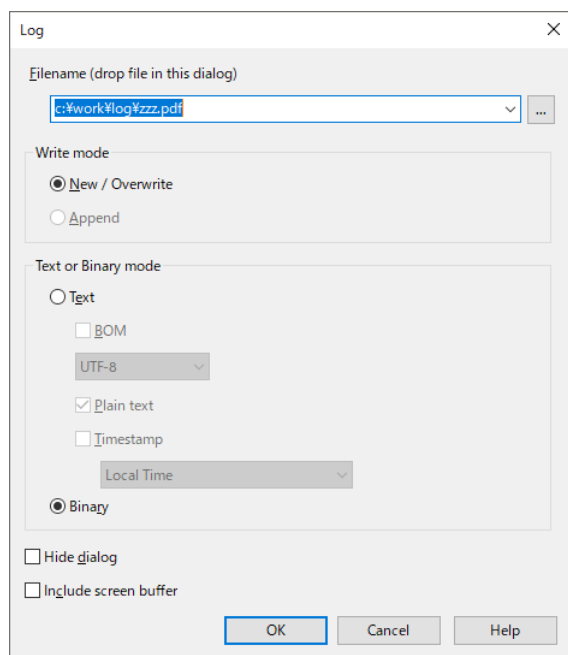
PC 側で、TCPFileSend.exe を起動して、「接続」ボタンを押す。



マイコンボード側では、PC アプリケーションで接続ボタンを押した際に、端末に表示が出ます。

```
-- TCP CONNECTION ACCEPTED --
(connection from 192.168.0.97:60210)
```

上記メッセージをログに含めない様に、メッセージ表示後に、マイコンボード側では、teraterm の File - log でログを取ってください。tetaterm の端末表示をデバッグモード設定して、Shift-ESC を押して、hex 表示に切り替えてください。

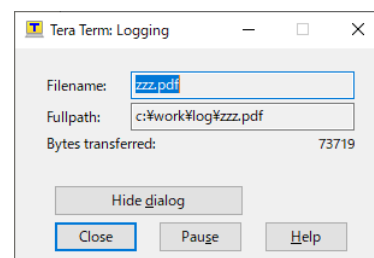


(バイナリファイルを受信する場合は、Binary を選択)

PC 側では、「送信ファイルを開く」でファイルを開いた後で、「送信」ボタンを押してください。

送信側

受信側



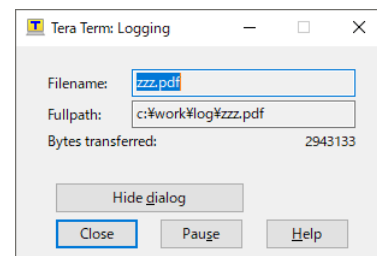
マイコンボード側では、端末に受信データの表示が出ます。

```
A3 06 A5 57 A2 B4 32 74 ED 01 BA 90 31 F6 AB 95 91 98 B6 18 DC 4A 4B 4B 4B 4A 4A 8A 8B 8B 8B 8A 8A FA E6
9B 6F E8 BD EE 74 4C 09 0A 1C 3F 7E 1C 08 47 CA 08 4D 4A EC 6D 57 F9 F9 F9 A1 A1 A1 50 31 23 23 E3 FA F5 EB
57 AF 5E 4D 48 48 80 B6 F6 EF DF 8F EF 3B F3 CA D3 70 73 0A 0B 0B 3B 79 F2 64 6A 6A EA 2D 73 02 6B E0 27 E0
16 78 75 EC D8 31 61 75 20 2E 29 CB AB 0E B7 01 6E CD CE 51 0F 75 0E 1A 51 E7 20 6A EC A6 56 60 4B 1A D0 AB
BE D9 4C 2C 74 F9 FA CA 85 DC 9A D7 0B 46 00 B7
```

送信完了すると、

送信側

受信側



となりますので、teraterm のログのダイアログの「Close」を押してください。(ログのクローズ→ファイルを閉じる)

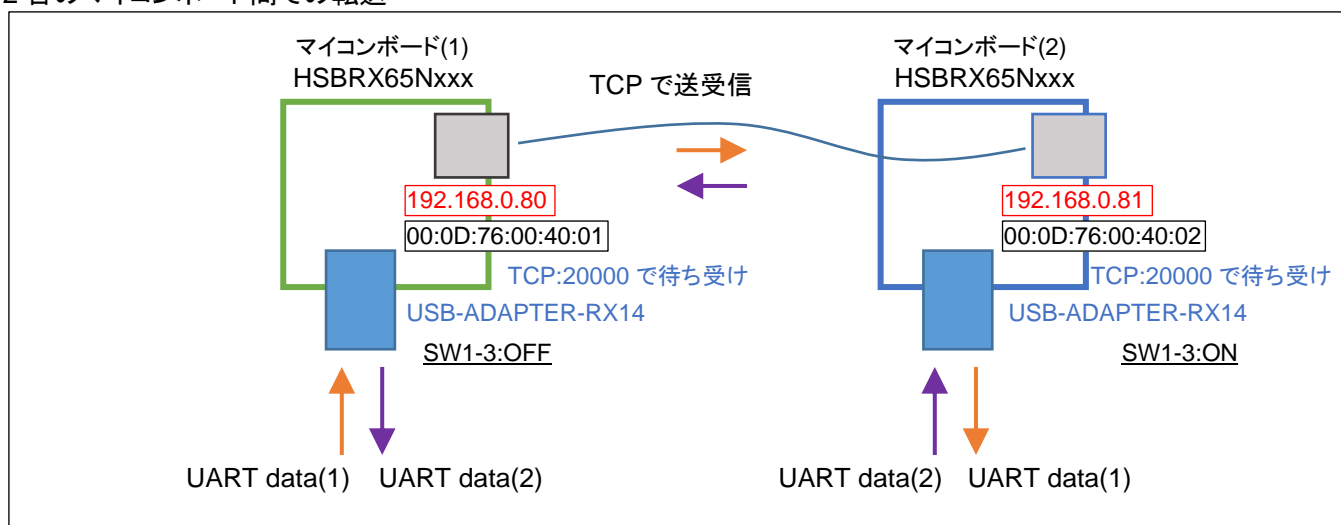
TCPFileSend.exe には、送信先 IP アドレスの設定の他に、送信インターバル「150ms」という項目があります。このタイミングで、データを送信しています。1 回に 1460 バイト送って 150ms 待つという動作です。

マイコンボード側は、データを UART で送信しているので、UART の設定速度(115,200bps)を考慮したタイミングとしています。(送信インターバルは「送信」ボタンを押したタイミングで読み込むので、変更したい場合は送信前に変えてください。)

なお、TCPFileSend/TCPFileReceive で取り扱うファイルサイズは 10MB までとしています。この値は、C#のソース内に書かれている値なので、変更する場合はソースを 1 行変更してビルドすると変えられます。

10.3.2 台のマイコンボード間のデータ転送

・2 台のマイコンボード間での転送



2 台のボードに同じプログラムを書き込み、DIP-SW の 3 を一方を OFF、もう一方を ON に設定します。その場合は、UART-Ethernet-UART のブリッジとして動作します。

UDP と異なり、TCP の場合はデータ送信前にセッションを確立する必要があるため、データを送信したい側(クライアント側)のボードの SW3 を押して TCP の接続を行ってください。なお、どちらが送信側になるか意識せずに、単純に起動後に両方のボードの SW3 を押す事で問題ありません。その場合は、双方向で通信が通る形となります。

○マイコンボード(1)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether TCP sample program (with SCI1-TX->DMAC).
-setting address-
---
this board IP address      -> 192.168.0.80 (MAC address -> 00-0D-76-00-40-01 )
netmask                    -> 255.255.255.0
gateway IP address         -> 192.168.0.1
communication target IP address -> 192.168.0.81
---
-- TCP SERVER LISTEN START (port = 20000) --
```

○マイコンボード(2)

```
Copyright (C) 2026 HokutoDenshi. All Rights Reserved.
RX65N Ether UART Ether TCP sample program (with SCI1-TX->DMAC).
-setting address-
---
this board IP address      -> 192.168.0.81 (MAC address -> 00-0D-76-00-40-02 )
netmask                    -> 255.255.255.0
gateway IP address         -> 192.168.0.1
communication target IP address -> 192.168.0.80
---
-- TCP SERVER LISTEN START (port = 20000) --
```

マイコンボード(1)の SW3 を押す。

○マイコンボード(1)

```
-- TCP CONNECTION ESTABLISHED --
```

○マイコンボード(2)

```
-- TCP CONNECTION ACCEPTED --  
(connected from 192.168.0.80:52362)
```

マイコンボード(2)の SW3 を押す。

○マイコンボード(1)

```
-- TCP CONNECTION ACCEPTED --  
(connected from 192.168.0.81:52362)
```

○マイコンボード(2)

```
-- TCP CONNECTION ESTABLISHED --
```

この段階で、双方向でデータのやり取りができる状態となります。

マイコンボード(1)側の端末から、キーボードで abcdef を入力してみます。

○マイコンボード(1)

```
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --  
-- 1 bytes send --
```

○マイコンボード(2)

```
abcdef
```

今度は、マイコンボード 2 から、123 を入力してみます。

○マイコンボード(2)

```
-- 1 bytes send --
-- 1 bytes send --
-- 1 bytes send --
```

○マイコンボード(1)

```
123
```

双方、どちらからでも通信が通る様になっています。

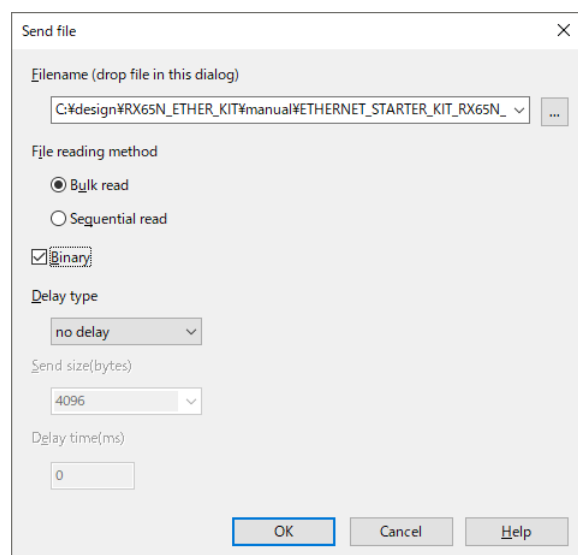
※--- xx bytes send ---表示はデバッグ用に表示する様になっています。

ether_common.h 内の#define SCI_DEBUG をコメントアウトしてプログラムをビルドすると、メッセージ類は一切表示されず、純粋な UART ブリッジとして動作します。(プログラムのビルドに関しては、ソフトウェア編マニュアルに記載しています。)

teraterm を使って、バイナリファイル転送を試してみます。

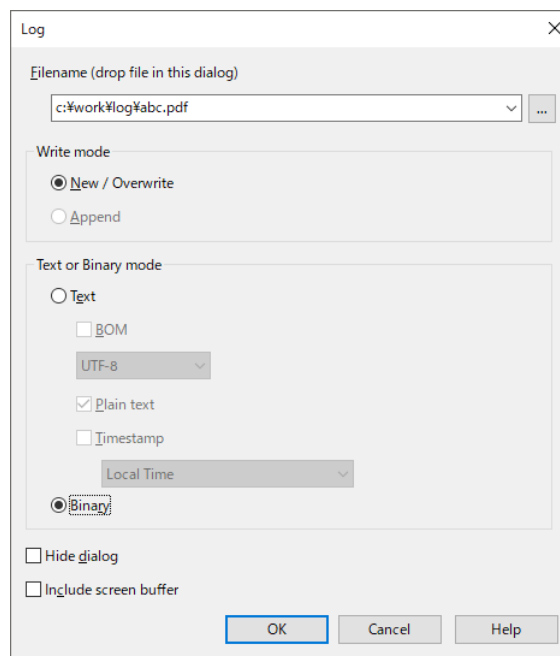
マイコンボード(1)(送信側)

File – Send File



マイコンボード(2)(受信側)

File – log



(バイナリファイルの送受信を行う場合は、どちらも Binary にチェックを入れる)

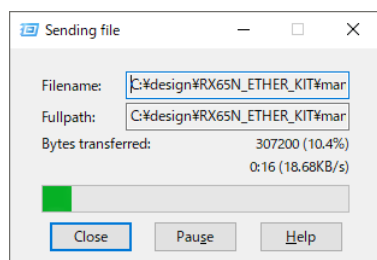
○マイコンボード(1)(送信側)

```
-- 1146 bytes send --
-- 1147 bytes send --
-- 1146 bytes send --
```

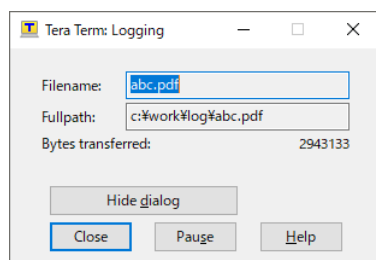
○マイコンボード(2) (受信側) ※hex 表示モードに設定(重要)

```
20 52 2F 49 6E 66 6F 20 35 31 30 20 30 20 52 2F 49 44 5B 3C 31 31 33 31 30 38 43 30 31 43 37 38 35 33 34 32 42
33 33 31 39 34 38 45 43 38 39 42 39 39 37 42 3E 3C 31 31 33 31 30 38 43 30 31 43 37 38 35 33 34 32 42 33 33 31
39 34 38 45 43 38 39 42 39 39 37 42 3E 5D 20 3E 3E 0D 0A 73 74 61 72 74 78 72 65 66 0D 0A 32 37 32 36 30 35
38 0D 0A 25 25 45 4F 46 0D 0A
```

送信中



受信完了



この場合も、バイナリレベルの一致が確認できました。

上記では、マイコンボード(1)→マイコンボード(2)のファイル転送を行いました。逆も動作します。

10.4.TCP のプロジェクト名に関して

TCP を使用したファイル転送で使用するプロジェクトは、

RX65N_UART_ETHER_TCP_DMAC

です。UDP を使用したバージョンは、

RX65N_UART_ETHER_UDP

でした。TCP の方はプロジェクト名に"DMAC"が付いています。TCP の方は、SCI(UART)の送信処理に、DMAC を使用しています。

| プロジェクト名 | IP プロトコル | SCI(UART)送信 |
|---------------------------|----------|-------------|
| RX65N_UART_ETHER_UDP | UDP | 割り込み |
| RX65N_UART_ETHER_TCP_DMAC | TCP | DMAC |

SCI の送信を行うのは、TCP/IP のデータを受信した側のボードです。UDP に対し TCP を取り扱う場合は、CPU の負荷が大きくなるために、UDP と TCP で同じ処理(UART-Ether ブリッジ)を行わせた場合、TCP の方が CPU リソースの余裕が少なく、データを取りこぼすケースが見受けられました。CPU 負荷低減のために、TCP 側のプロジェクトでは、SCI の送信に DMAC を使用しています。

(本プロジェクトで使用してる SCI のルーチンは、間欠的なデバッグメッセージの表示や低速なキーボードからの入力を想定して作成されています。バイナリデータを連続で送受信する事に最適化されていないという事情もあります。SCI のデータ送受信をブロック単位で行うようにすれば、SCI のボトルネックは解消されるかとも思います。(元々は、1 バイト単位で SCI のデータを取り扱っています。))

取扱説明書改定記録

| バージョン | 発行日 | ページ | 改定内容 |
|-------------|----------|-----|------|
| REV.1.0.0.0 | 2026.6.5 | — | 初版発行 |

お問合せ窓口

最新情報については弊社ホームページをご活用ください。

ご不明点は弊社サポート窓口までお問合せください。

株式会社 **北斗電子**

〒060-0042 札幌市中央区大通西 16 丁目 3 番地 7

TEL 011-640-8800 FAX 011-640-8801

e-mail: support@hokutodenshi.co.jp (サポート用)、order@hokutodenshi.co.jp (ご注文用)

URL: <https://www.hokutodenshi.co.jp>

商標等の表記について

- ・ 全ての商標及び登録商標はそれぞれの所有者に帰属します。
- ・ パーソナルコンピュータを PC と称します。

ルネサス エレクトロニクス RX マイコン搭載
HSB シリーズマイコンボード 評価キット

Ethernet スタータキット RX65N 取扱説明書(3)

株式会社 **北斗電子**

©2026 北斗電子 Printed in Japan 2026 年 6 月 5 日改訂 REV.1.0.0.0 (260605)
