

---

# R-IN32M3, R-IN32M4 シリーズ, RZ/T1 グループ

## TFTP ドライバ編

---

R01AN3939JJ0100  
Rev.1.00  
2017.07.11

### 要旨

本書は、ミドルウェア仕様に関する資料です。TFTP の概要および仕様、また機能実現のためのハードウェア構成、ソフトウェア構成、ドライバ仕様を記載しています。

### 動作確認デバイス

R-IN32M3-EC,  
R-IN32M3-CL  
R-IN32M4-CL2  
RZ/T1

## 目次

1. TFTP 概要.....	3
1.1 パケットフォーマット.....	4
1.2 パケットシーケンス.....	7
1.3 状態遷移表.....	8
1.4 TFTP サーバフロー.....	9
2. TFTPserver ミドルウェア構成.....	10
2.1 ハードウェア構成.....	10
2.2 ソフトウェア構成.....	10
3. 関数仕様.....	11
4. ミドルウェア仕様.....	14
4.1 TFTP port 固定モード設定方法.....	16
5. サンプルソフトチュートリアル.....	17
5.1 使用準備.....	17
5.1.1 ファイルの統合.....	17
5.1.2 ボード~ホスト PC 接続.....	19
5.1.3 IP アドレス設定.....	19
5.1.4 Windows 標準 TFTP クライアント機能有効設定.....	21
5.1.5 サンプルソフト実行.....	23
5.2 動作確認.....	24
6. ホームページとサポート窓口.....	25

## 1. TFTP 概要

TFTP とは、FTP と同様に TCP/IP ネットワークで、ファイルを転送するためのプロトコルのことです。FTP が TCP によりクライアント側とサーバー側とでコネクションを張り、送信処理を行うのに比べ、UDP のコネクションレスで処理を行います。コネクションレスですが、通信の信頼性を確保するために確認応答を行うプロトコルとなっています。用途としてはルータのファームアップデート等の小さなデータの転送に使用されます。

### FTP との違い

- ・FTP のようにユーザ名とパスワード認証を行いません。
- ・FTP では TCP 上で動作しますが TFTP は UDP 上で動作を行います。TFTP の配置を図 1-1 に示します。

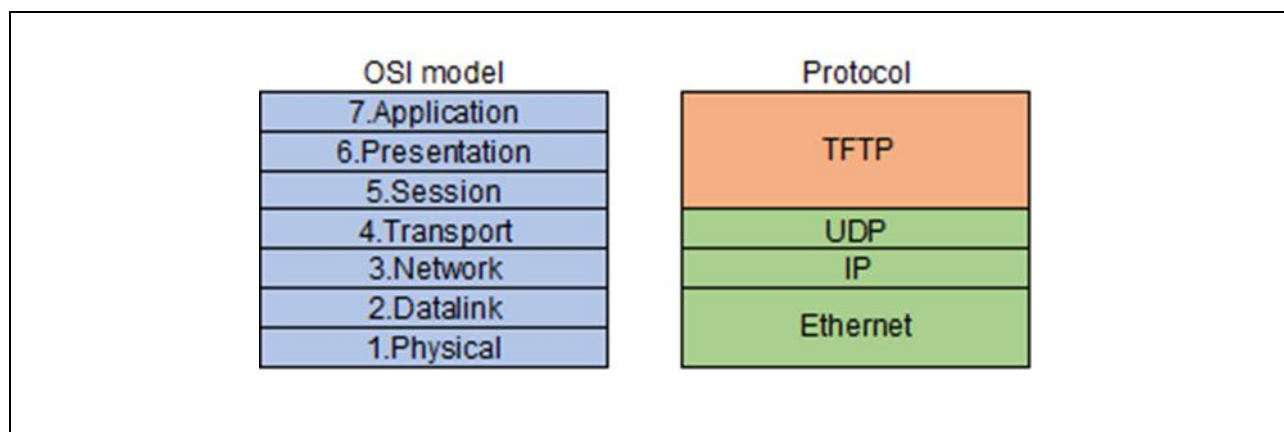


図 1-1 OSI モデルと TFTP プロトコル

## 1.1 パケットフォーマット

TFTP にて扱うパケットを表 1-1 に示します。

本ミドルウェアでは読み出しリクエスト (RRQ) 以外のパケットをサポートします。

表 1-1 TFTP にて扱うパケット一覧

オペコードの値	動作	サポート
1	読み出しリクエスト (RRQ : Read RQuest)	×
2	書き込みリクエスト (WRQ : Write RQuest)	○
3	データ (DATA)	○
4	応答 (ACK : ACKnowledgment)	○
5	エラー (ERROR)	○

図 1-2 に示す通り、UDP ペイロードが TFTP パケットとなります。先頭 2 バイトが TFTP のパケット種別を示すオペレーションコード (オペコード)、以降はパケットごとに書式が異なります。

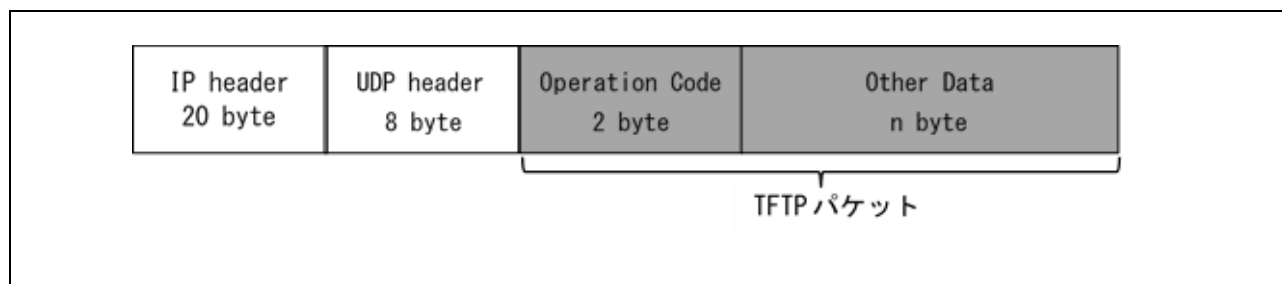


図 1-2 TFTP パケット書式

TFTP パケット別フォーマットを下記に示します。

### ■WRQ パケットフォーマット

クライアントがサーバーにファイルを送る際に使用するパケット。

Operation Code	File name	null	Mode	null
2byte	n byte (string)	1byte	n byte (string)	1byte

- ・Operation Code : オペコード「2」。
- ・File Name : 更新対象のファイル名。
- ・Mode : 転送するファイルの形式。ASCII コードでモード名を指定。  
netascii … FTP の ASCII モード。受信側が自分の使用する文字コードに変換して保存。  
octet … FTP の Binary モード。送信側から無変換で送信。  
mail … netascii モードと同じ。ファイル名の代わりにメールを受信するユーザ名が入る。

**■DATA パケットフォーマット**

ファイルをブロックに分割した転送の際に使用するパケット。

Operation Code	Block No	Data
2byte	2byte	n byte

- ・ Operation Code : オペコード「3」。
- ・ Block No : ブロック番号。
- ・ Data : 512 バイトのデータサイズで構成される。最終ブロックは512 バイト未満となりファイルの終端を表す。

**■ACK パケットフォーマット**

DATA パケットを受信したことを相手に通知する際に使用するパケット。

受信したブロック番号を相手に通知するためにブロック番号を持っている。

Operation Code	Block No
2byte	2byte

- ・ Operation Code : オペコード「4」。
- ・ Block No : DATA パケットを受信した場合は、受信したブロック番号を返す。  
WRQ パケットの場合は、ブロック番号 0 を返す。

## ■ERROR パケットフォーマット

エラーが発生した場合、相手に通知する際に使用するパケット。

Operation Code	Error Code	Error Message	null
2byte	2byte	n byte	1byte

・ Operation Code : オペコード「5」。

・ Error Code : エラーコード。

・ Error Message : エラーメッセージ。

・ Null : エラーメッセージの終端。

※エラーコード、メッセージについては「表 4-1 マクロ一覧」 参照

## 1.2 パケットシーケンス

1. クライアントよりサーバーへポート番号 69 番で WRQ を送信 ※ポート 69 を使うのは WRQ 受信時のみ
2. サーバーより確認応答を送信。 ※ブロック番号は 0  
※このときポートは 69 ではなくランダムなポートにて応答を返す。以後の処理はこのポートで行う。
3. クライアントより 512 バイトにて DATA 送信開始
4. サーバーは DATA を受信するたびに同じブロックナンバーにて確認応答を返す。
5. クライアントは最後のブロック (0~511 バイト) で DATA 送信
6. クライアントが最後の ACK を受信して処理終了
7. 途中でパケットがロスした場合、リトライ処理を行う。  
一定時間内に DATA パケットが来ない場合、タイムアウト処理は直前の ACK パケットを再送する。  
設定数分リトライし応答がない場合エラーとする。

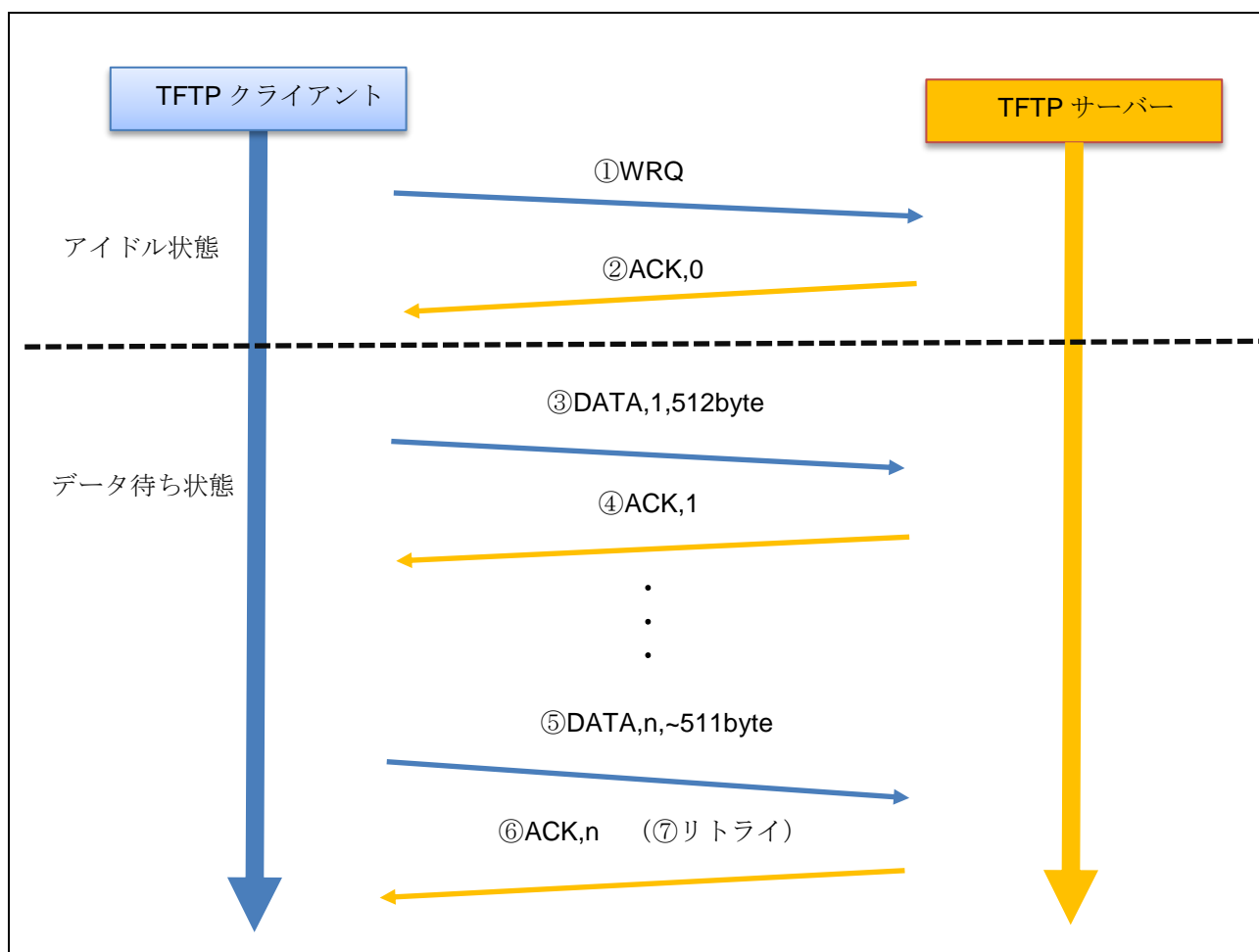


図 1-3 パケットシーケンス

### 1.3 状態遷移表

イベント別の状態遷移を表 1-2 に示します。

※アイドル状態時の DATA 受付、タイムアウト、リトライオーバーについては未処理となります。

想定外のイベントが起きた際はすべてエラーとしています。

表 1-2 状態遷移表

イベント\状態	アイドル状態	DATA 待ち状態
WRQ 受付	確認応答 →DATA 待ち状態	確認応答 →DATA 待ち状態
DATA 受付	-	確認応答 →DATA 待ち状態
DATA 受信完了	-	確認応答 →アイドル状態
タイムアウト	-	確認応答 →DATA 待ち状態
リトライオーバー	-	エラー →アイドル状態
上記以外	エラー →アイドル状態	エラー →アイドル状態



## 1.4 TFTP サーバフロー

本ミドルウェアの簡易フローを図 1-4 に示します。

初期化を行った後、状態の判定を行い DATA 受付、WRQ 受付を行います。

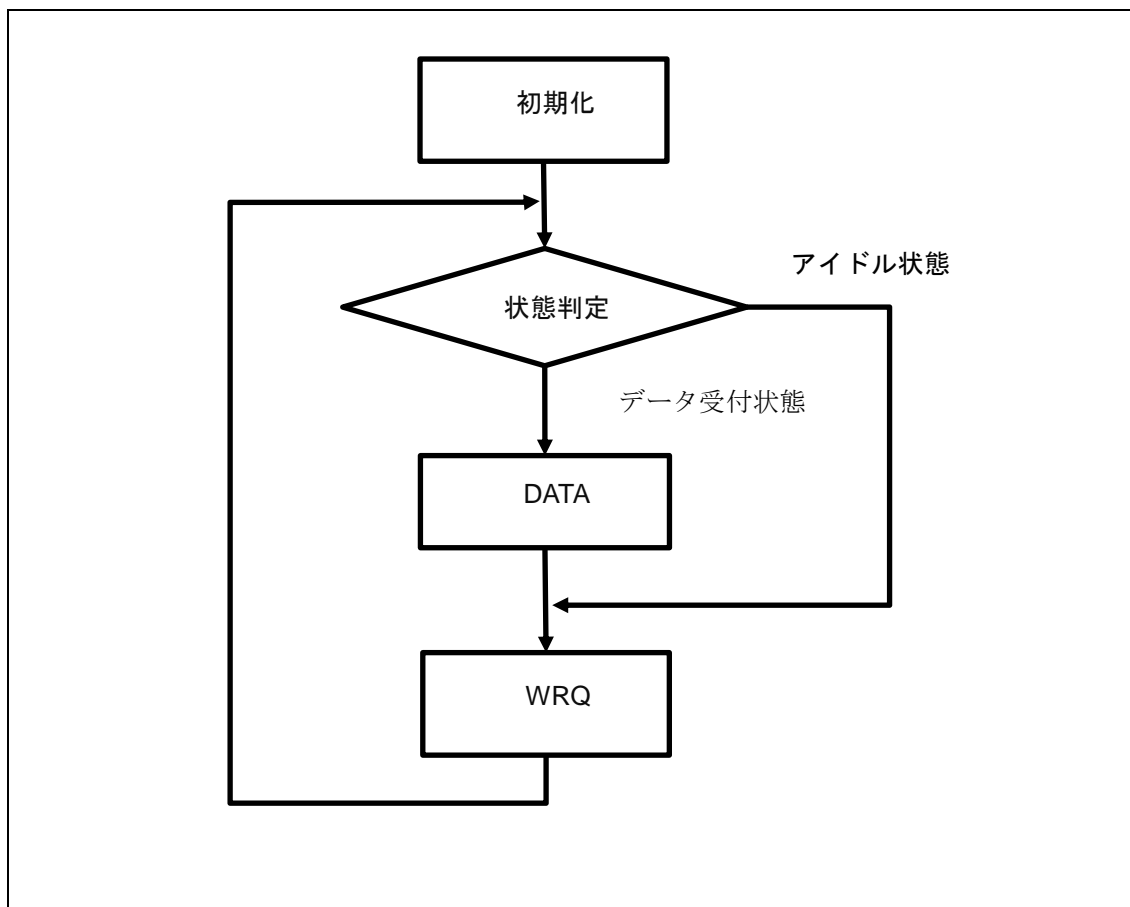


図 1-4 TFTP サーバフロー

## 2. TFTPserver ミドルウェア構成

### 2.1 ハードウェア構成

Ethernet 通信でクライアント PC とターゲットボード間を接続します。

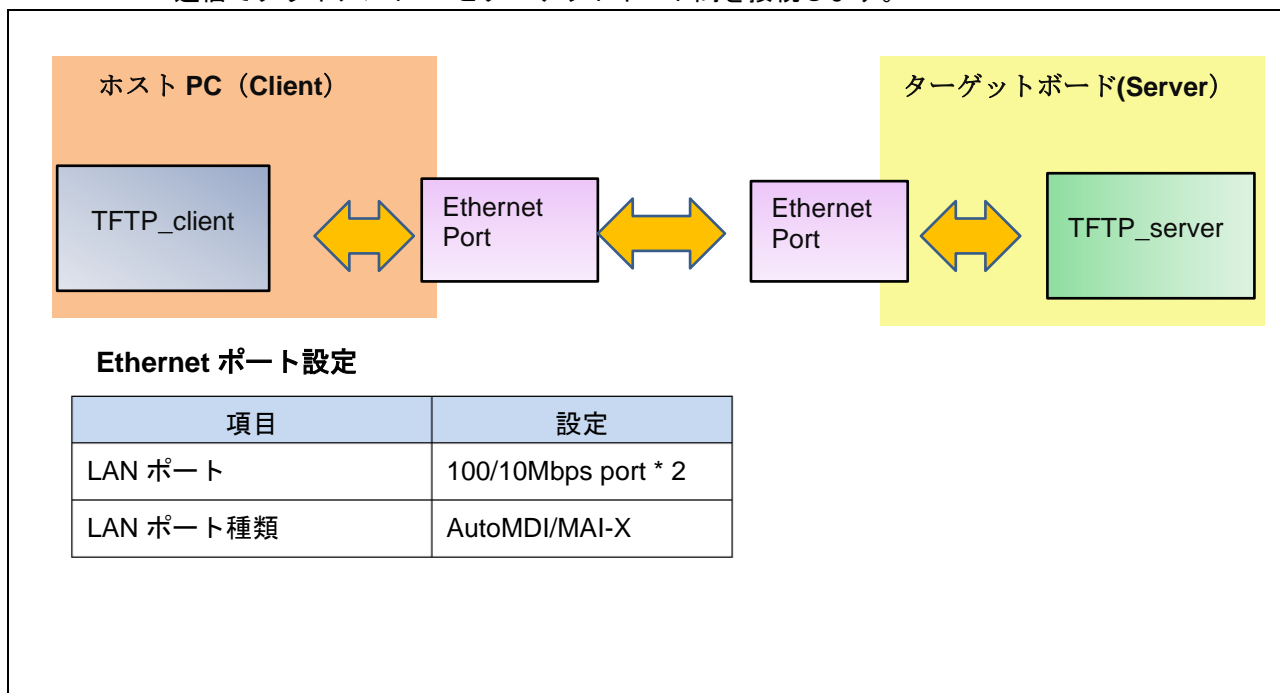


図 2-1 ハードウェア構成とポート設定

### 2.2 ソフトウェア構成

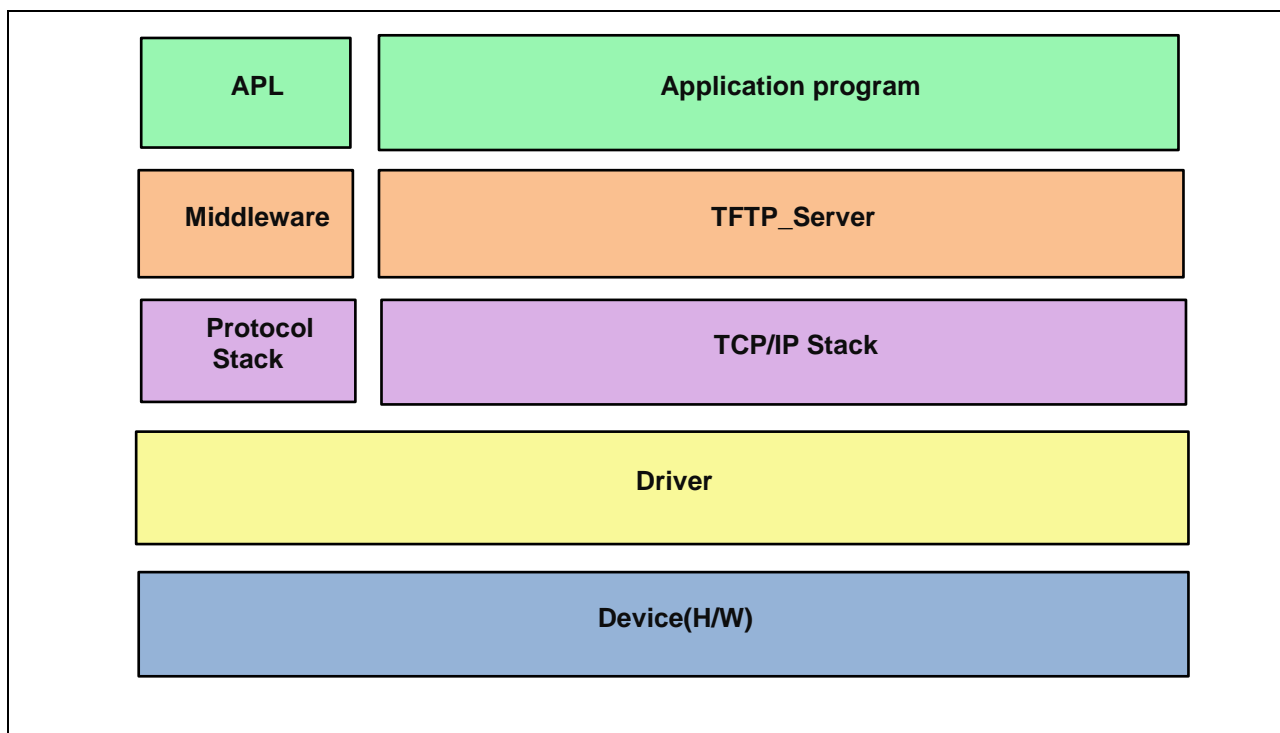


図 2-2 ソフトウェア構成

## 3. 関数仕様

## ・ ack 構造体

```
typedef struct t_ack {
    uint16_t    op;           /*オペレーションコード*/
    uint16_t    block;       /*ブロックナンバー*/
}ack
```

## ・ err 構造体

```
typedef struct t_err {
    uint16_t    op;           /*オペレーションコード*/
    uint16_t    errcd;       /*エラーコード*/
    char        msg[1500-4]  /*メッセージ*/
}err
```

---



---

```
tftp_wrq_socket_init    WRQ用ソケットの初期化
```

---



---

## 【書式】

```
int tftp_wrq_socket_init(void);
```

## 【パラメータ】

なし

## 【戻り値】

OK	正常終了
API_ERR	APIエラー

## 【解説】

WRQ 用ソケットの初期化及びクライアントからのデータを受信するバッファの初期化を行います。どのアドレスからも接続を受け入れるポート 69 のソケットを作成します。

---



---

```
tftp_data_socket_init    DATA用ソケットの初期化
```

---



---

## 【書式】

```
int tftp_data_socket_init(void);
```

## 【パラメータ】

なし

## 【戻り値】

OK	正常終了
API_ERR	APIエラー

## 【解説】

DATA 用ソケットの初期化及びクライアントからのデータを受信するバッファの初期化を行います。どのアドレスからも接続を受け入れるランダムポートのソケットを作成します。

---



---

tftp\_cmd\_wrq WRQ受信処理

---



---

## 【書式】

```
int tftp_cmd_wrq(TFTP_PARAM_TABLE *param);
```

---

## 【パラメータ】

TFTP_PARAM_TABLE	*param	パラメータテーブル
------------------	--------	-----------

---

## 【戻り値】

OK	正常終了
ERR	エラー
API_ERR	APIエラー
NO_DATA	データ受信なし

---

## 【解説】

この関数は WRQ 用ソケットに対し受信があった際の処理を行います。

受信したパケットの判定を行い、WRQ でない場合、エラーパケットを返します。

WRQ の場合、ACK 応答を返しパケットよりファイル名とモード名を取得しコールバック関数へ渡します。

正常に処理終了した場合、DATA 用ソケットを作成し、データ受付状態になります。

---



---

tftp\_cmd\_data DATA受信処理

---



---

## 【書式】

```
int tftp_cmd_data(TFTP_PARAM_TABLE *param);
```

---

## 【パラメータ】

TFTP_PARAM_TABLE	*param	パラメータテーブル
------------------	--------	-----------

---

## 【戻り値】

OK	正常終了
ERR	エラー
API_ERR	APIエラー

## 【解説】

この関数は DATA 用ソケットに対し受信があった際の処理を行います。

受信したパケットの判定を行い、DATA でない場合、エラーパケットを返します。DATA の場合、パケットよりブロック番号とデータを取得しコールバック関数へ渡します。

また、一定時間内に DATA パケットが来ない場合、タイムアウト処理として直前の ACK パケットを再送します。

サイズの判定を行い DATA パケットのサイズが 512 バイト未満の場合、データ受信を完了としコールバック関数を呼び出します。

DATA 受信時、WRQ の受信があった場合、一旦 WRQ の判定へと処理が移ります。

---



---

tftp\_send\_ack            ACK応答処理

---



---

## 【書式】

```
int tftp_send_ack(int block);
```

---

## 【パラメータ】

int	block	ブロックナンバー
-----	-------	----------

---

## 【戻り値】

OK	正常終了
API_ERR	APIエラー

---

## 【解説】

この関数はクライアントへ確認応答をする際に使用する関数です。パラメータよりパケット作成しクライアントへ送信を行います。

また、状態判定により送信の際に使用するソケットを切り替えます。

---



---

tftp\_send\_err            ERR応答処理

---



---

## 【書式】

```
int tftp_send_err(int errcd, const char *errmsg);
```

---

## 【パラメータ】

int	errcd	エラーコード
const char	*errmsg	エラーメッセージポインタ

---

## 【戻り値】

OK	正常終了
API_ERR	APIエラー

---

## 【解説】

この関数はクライアントへエラー応答をする際に使用する関数です。パラメータよりパケット作成しクライアントへ送信を行います。また、状態判定により送信の際に使用するソケットを切り替えます。

---



---

tftp\_end                終了処理

---



---

## 【書式】

```
void tftp_end(void);
```

---

## 【パラメータ】

なし

## 【戻り値】

なし

## 【解説】

この関数はミドルウェア処理をエラーで抜ける際に使用する関数です。現在作成されているソケットをクローズします。

#### 4. ミドルウェア仕様

本ミドルウェアではクライアントからの要求に応じて動作するコールバック関数を定義しています。これによりファームウェアアップデート以外にも汎用的に利用できる仕様としています。  
※決まったパラメータを渡し、任意の処理を行い仕様に従い戻り値を返します。

##### ・ TFTP パラメータ構造体

```
typedef struct tftp_param_table {
    Int (*wrq_cbk)(char *,char *)           WRQ受信コールバック関数ポインタ
    Int (*data_cbk)(int ,uint8_t *,int)     DATA受信コールバック関数ポインタ
    void(*end_cbk)(void)                   受信完了コールバック関数ポインタ
}TFTP_PARAM_TABLE
```

##### 【解説】

この構造体に TFTP サーバーで使用する関数を登録します。

表 4-1 マクロ一覧

マクロ名	値	メッセージ	説明
TFTP_OK	0		正常終了
TFTP_ERR0	1	Not defined, see error message (if any).	0 未定義。エラーメッセージを参照。
TFTP_ERR1	2	File not found.	1 ファイルが見つからない。
TFTP_ERR2	3	Access violation.	2 アクセス違反。
TFTP_ERR3	4	Disk full or allocation exceeded.	3 ディスクに書き込み領域が無い。
TFTP_ERR4	5	Illegal TFTP operation.	4 不正なTFTPコマンド。
TFTP_ERR5	6	Unknown transfer ID.	5 不明な転送ID。
TFTP_ERR6	7	File already exists.	6 ファイルが既にある。
TFTP_ERR7	8	No such user.	7 ユーザ未定義。
TFTP_PORT_FIX_MODE	0	-	tftpポート固定モード切替 0 = disable , 1 = enable

---



---

wq\_cbk                                      WRQコールバック関数

---



---

##### 【書式】

```
int wq_cbk(char *filename ,char *mode);
```

##### 【パラメータ】

char	*filename	WRQパケットパラメータ"filename"へのポインタ
char	*mode	WRQパケットパラメータ"mode"へのポインタ

##### 【戻り値】

マクロ一覧に従い戻り値を選択。

##### 【解説】

この関数は WRQ パケット受信時のコールバック関数処理となります。  
パラメータについて任意の処理を行います。

---



---

data_cbk	DATAコールバック関数
----------	--------------

---

**【書式】**

```
void data_cbk(int blockno ,uint8_t *buf,int size);
```

**【パラメータ】**

int	blockno	DATAパケットパラメータ"block"
uint8_t	*buf	DATAパケットパラメータ"data"へのポインタ
int	size	"data"の有効サイズ

**【戻り値】**

なし

**【解説】**

この関数は DATA パケット受信時のコールバック関数処理となります。  
パラメータについて任意の処理を行います。

---



---

end_cbk	ENDコールバック関数
---------	-------------

---

**【書式】**

```
void end_cbk(void);
```

**【パラメータ】**

なし

**【戻り値】**

なし

**【解説】**

この関数は DATA 受信完了時のコールバック関数処理となります。

---



---

tftp_server	TFTPサーバー
-------------	----------

---

**【書式】**

```
int tftp_server(TFTP_PARAM_TABLE *param);
```

**【パラメータ】**

TFTP_PARAM_TABLE	*param	コールバック関数構造体
------------------	--------	-------------

**【戻り値】**

なし

**【解説】**

この関数は TFTP サーバーを初期化し TFTP クライアントからの要求を受付します。  
アイドル状態とデータ受付状態を判定し、受付処理の切り替えをします。

```

実装例
int wrq_cbk(char *filename , char *mode){
    return 0;
}

int data_cbk(int blockno , uint8_t *buf,int size){
    return 0;
}

int end_cbk(void){
    return 0;
}

int main()
{
    TFTP_PARAM_TABLE param;
    int result;

    param.wrq_cbk = wrq_cbk;
    param.data_cbk = data_cbk;
    param.end_cbk = end_cbk;

    re = tftp_server(&param);
    if(re == OK){
        /*Successful completion*/
    }
}

```

#### 4.1 TFTP port 固定モード設定方法

ポート固定モード指定マクロ”TFTP\_PORT\_FIX\_MODE”にてクライアントへ応答を返す際のポート設定を切り替えることができます。デフォルトでランダムポート設定”0”となっています。

表 4-2 TFTPport 固定モード

マクロ名	定義ファイル	値	説明
TFTP_PORT_FIX_MODE	tftp_server.h	0	ランダムポート
		1	ポート69固定モード

**補足 Windows 標準 TFTP クライアントを使用する場合はポート固定モードとしてください。**



## 5. サンプルソフトチュートリアル

本パッケージは R-IN32M3 向けのサンプルソフトとなります。

TFTP ミドルウェアについては R-IN32M3, R-IN32M4 シリーズ, RZ/T1 グループ共通でご使用できます。

本章では Windows 標準の TFTP クライアントを使用した実行・確認手順を記載しています。

### 動作概要

サンプルソフトを実行すると TFTP サーバーとしてファイル転送を受け付けます。

下記のタイミングで任意の処理を実行することができます。

- ・ WRQ パケット受信時
- ・ DATA パケット受信時
- ・ DATA パケット受信完了時

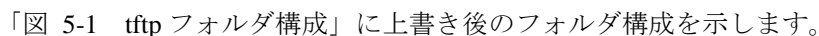
## 5.1 使用準備

### 5.1.1 ファイルの統合

本パッケージは「R-IN32M3 シリーズ TCP/IP スタック」からの差分で構成されています。

「R-IN32M3 シリーズ TCP/IP スタック」と統合することで TFTP ミドルウェア、サンプルソフトが使用可能となります。

R-IN32M3 製品ページのサンプルコードより「R-IN32M3 シリーズ TCP/IP スタック」を入手し、本パッケージを上書きしてください。

「 5-1 tftp フォルダ構成」に上書き後のフォルダ構成を示します。

「R-IN32M3 シリーズ TCP/IP スタック」に対して、赤字が新規に追加される本パッケージのフォルダ・ファイルとなります。

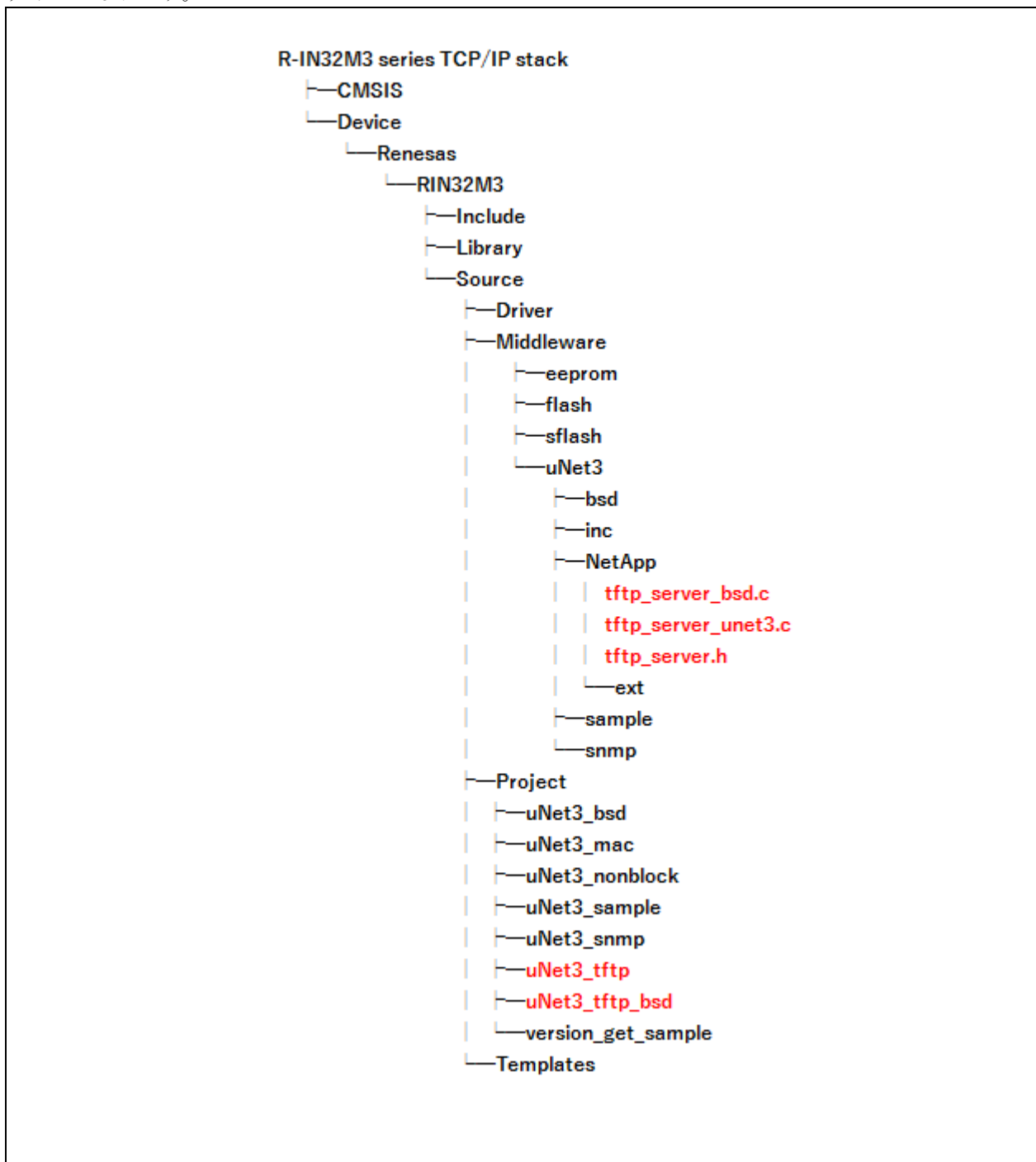


図 5-1 tftp フォルダ構成

### 5.1.2 ボード～ホスト PC 接続

使用ボード : R-IN32M3-EC Evaluation Board TS-R-IN32M3-EC

「2.1 ハードウェア構成」に従い Ethernet ケーブルを LAN ポート 1 へ接続してください。

### 5.1.3 IP アドレス設定

R-IN32M3ボードのIPアドレスについて下記の設定となっています。

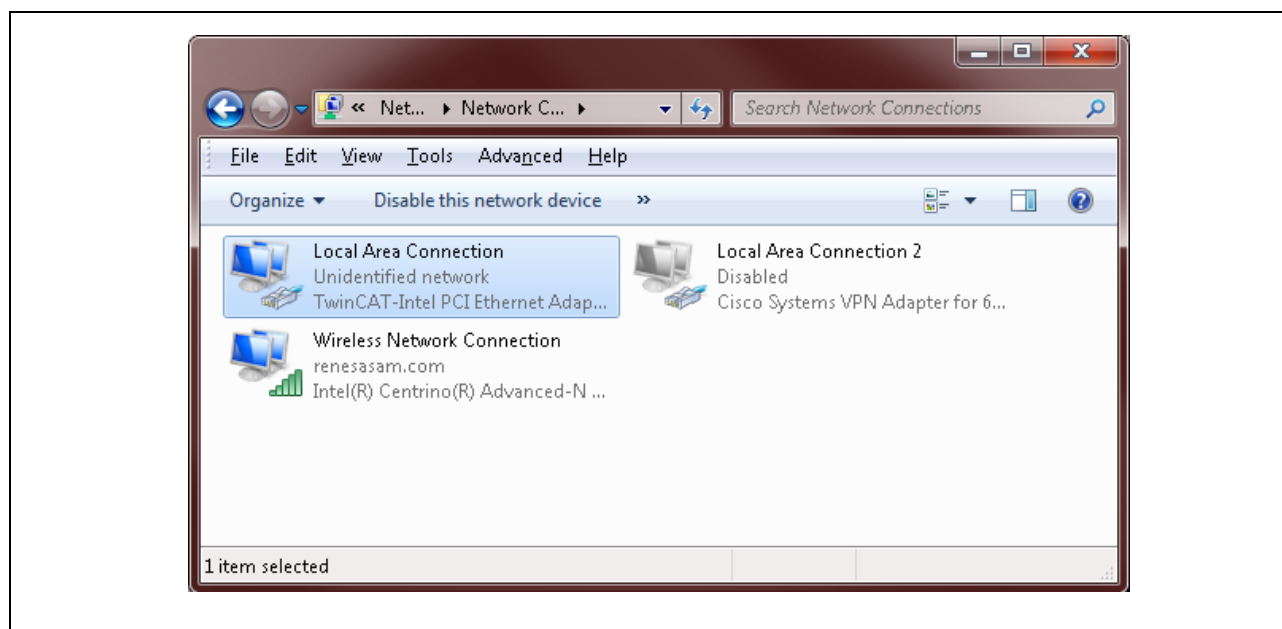
サブネットマスク : 255.255.255.0

IPアドレス : 192.168.1.100

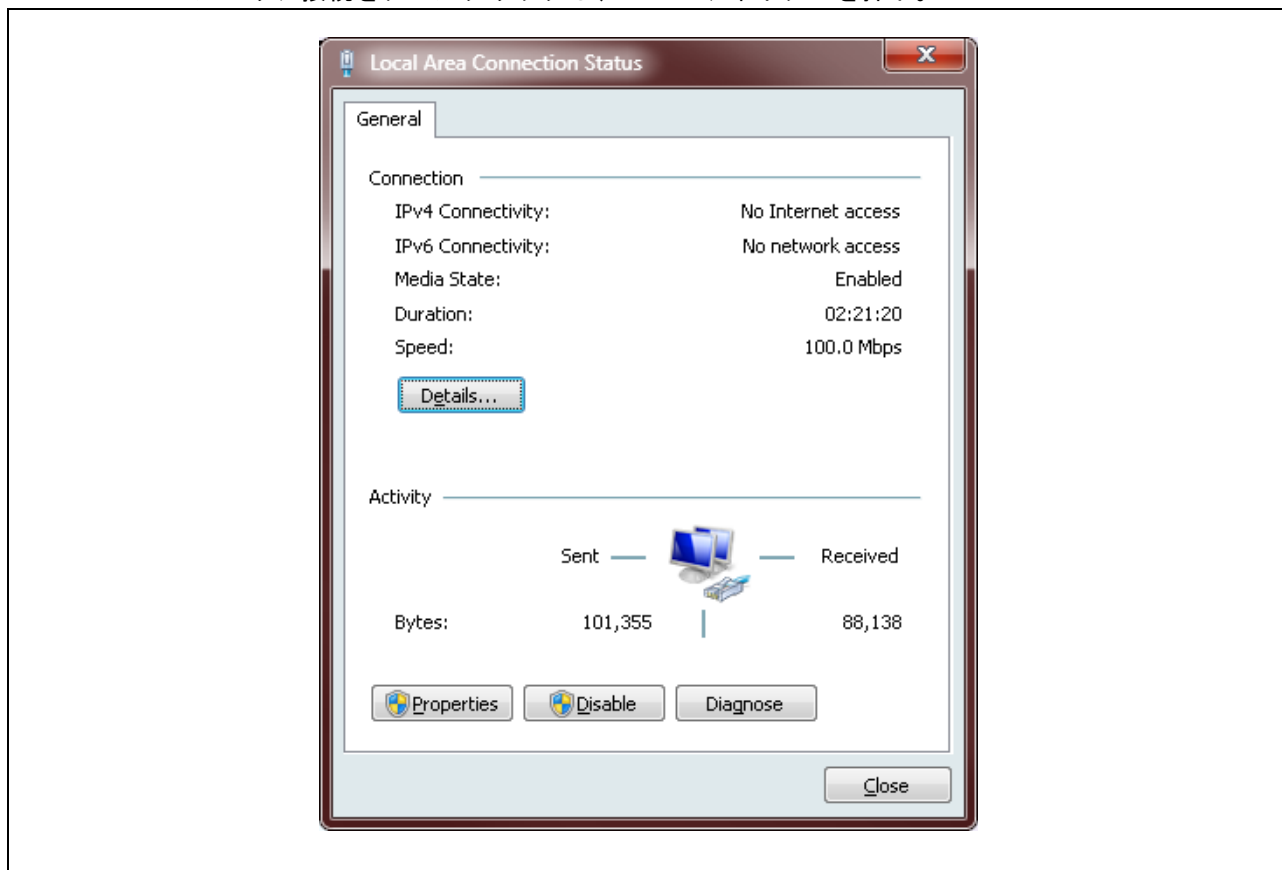
ホストPCのIPアドレスをR-IN32M3ボードと同じネットワーク部に設定してください。

下記にIPアドレス設定方法を示します。

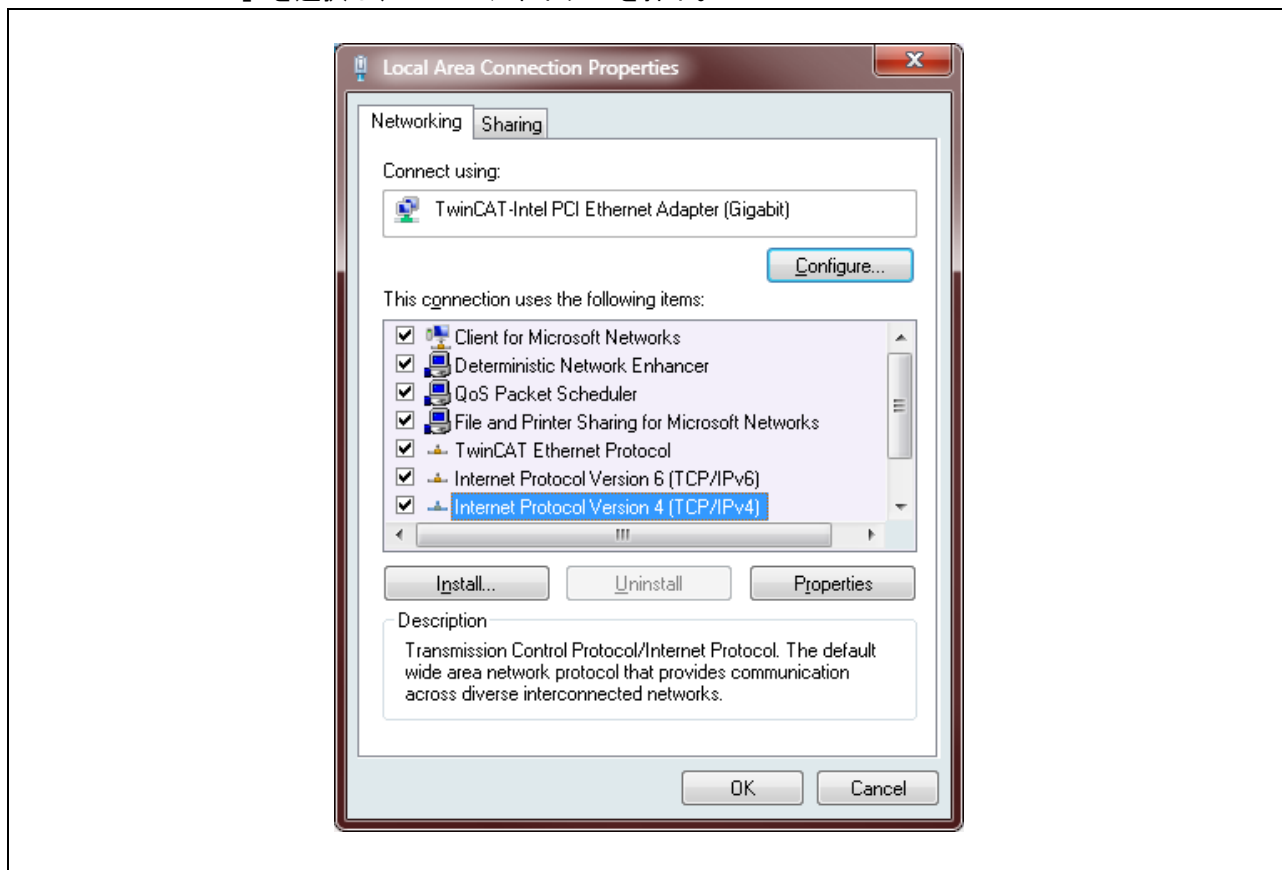
- ・ 「ネットワーク設定」を開く。  
Windows7 では、Control panel->Network and Sharing Center->Change adapter settings。



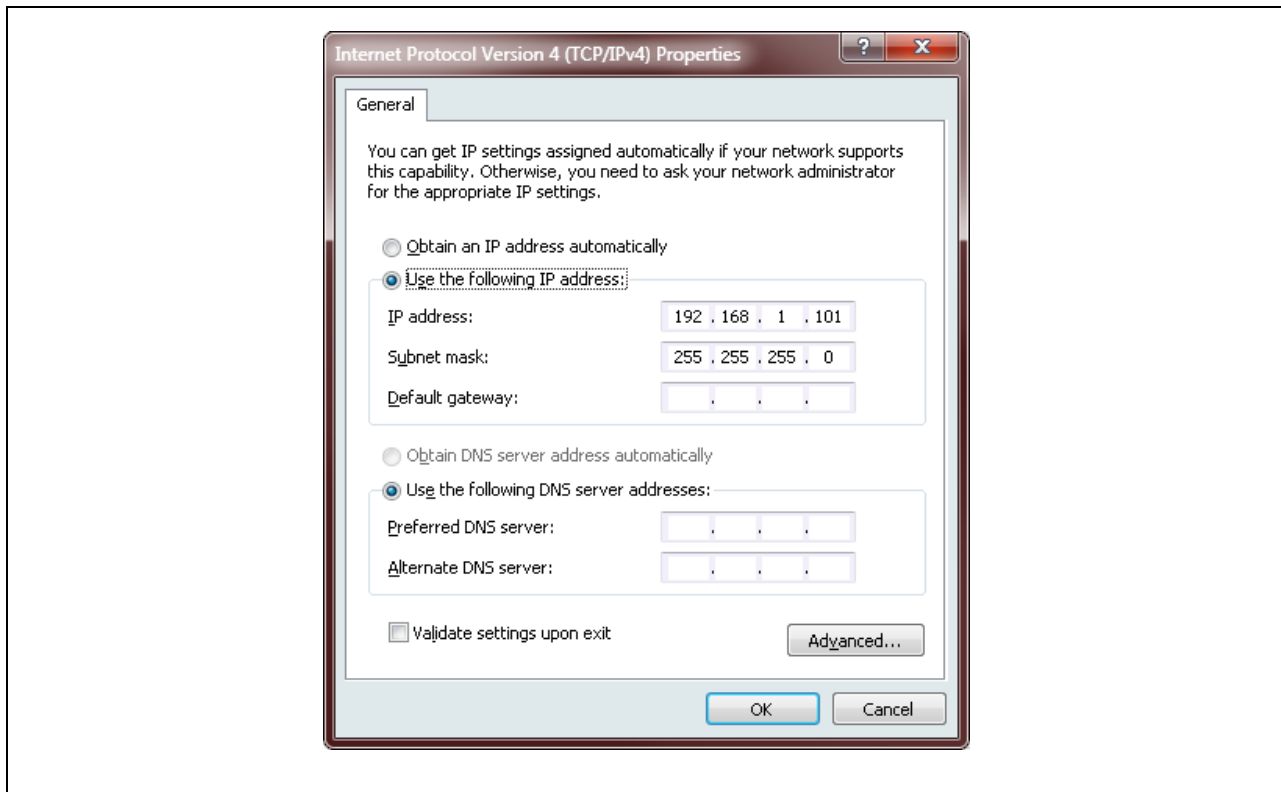
- ローカルエリア接続をダブルクリックし、プロパティボタンを押下。



- 「TCP/IPv4」を選択し、プロパティボタンを押下。



- ・ IP アドレスを 192.168.1.101 に設定、サブネットマスクを 255.255.255.0 に設定。



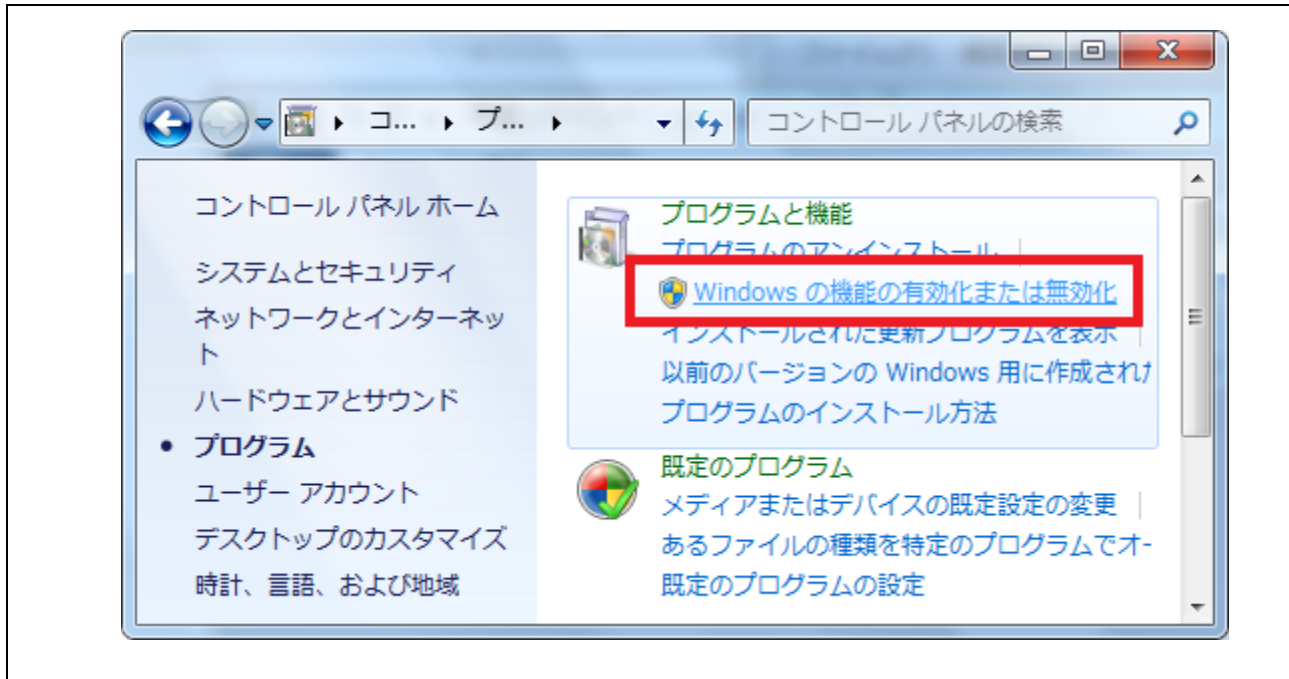
#### 5.1.4 Windows 標準 TFTP クライアント機能有効設定

端末により無効となっている場合があるため下記に有効にする設定方法を示します。

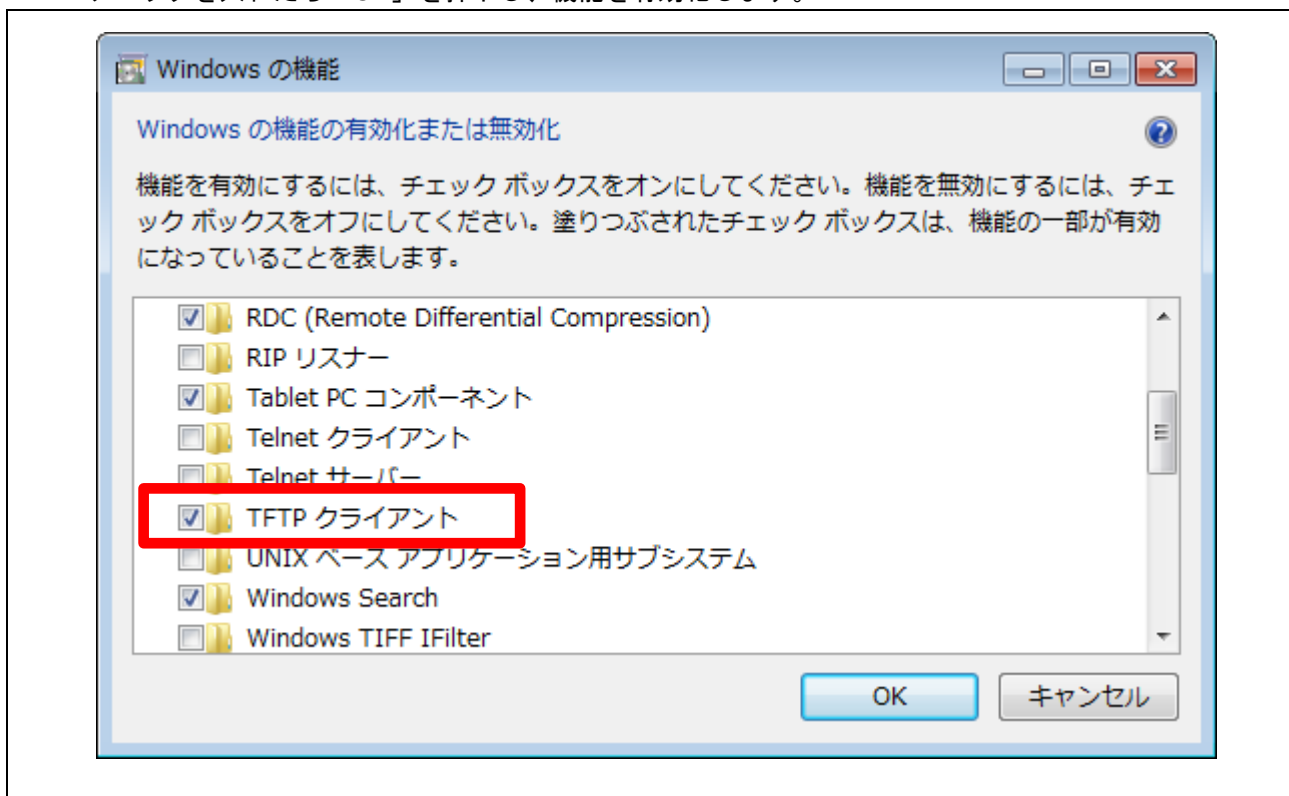
- ・ コントロールパネルを開き、「プログラム」を選択。



- ・「プログラムと機能」より「windows の機能の有効化または無効化」を選択。



- ・表示項目の中から「TFTP クライアント」にチェックを入れます。  
チェックを入れたら「OK」を押下し、機能を有効化します。



### 5.1.5 サンプルソフト実行

...¥Device¥Renesas¥RIN32M3¥Source¥Middleware¥uNet3¥NetApptftp\_server.h にて定義されているマクロ TFTP\_PORT\_FIX\_MODE の値を”1” ポート固定モードに設定してください。

下記いずれかのワークベンチファイルよりサンプルソフトを実行してください。

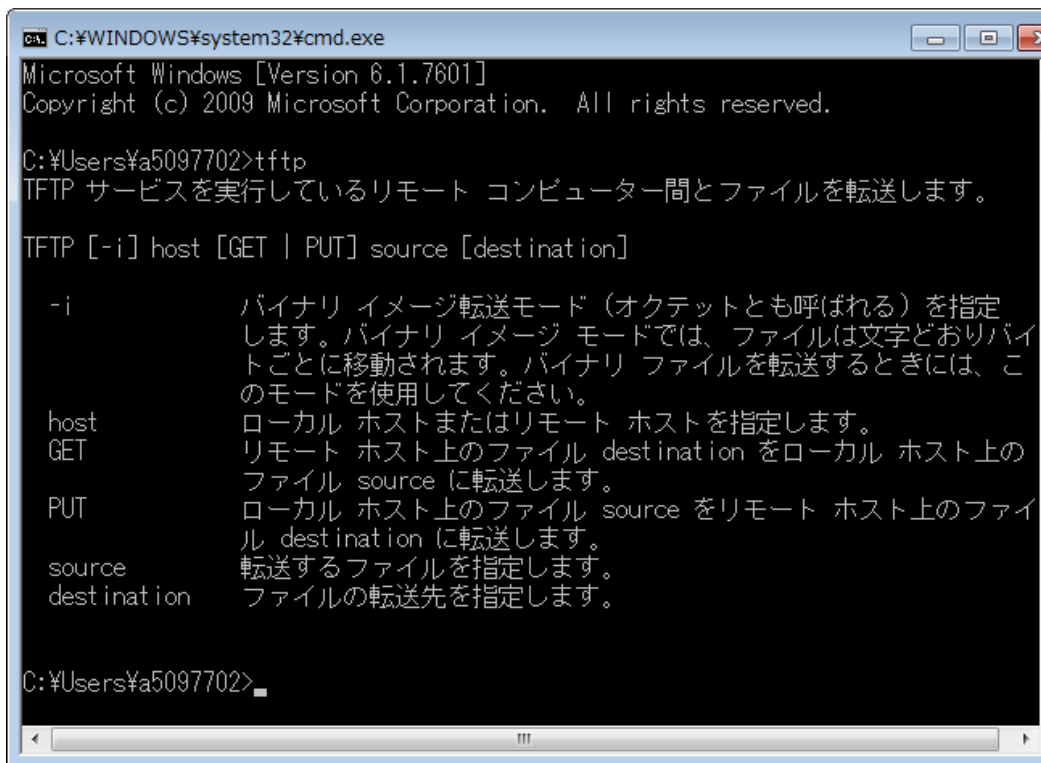
ワークベンチファイル :

- ・ uNet3\_tftp ...¥Device¥Renesas¥RIN32M3¥Source¥Project¥uNet3\_tftp¥IAR¥main.eww
- ・ uNet3\_tftp\_bsd ...¥Device¥Renesas¥RIN32M3¥Source¥Project¥uNet3\_tftp\_bsd¥IAR¥main.eww

## 5.2 動作確認

windows 標準でインストールされている TFTP クライアントを使用した確認手順を示します。

- ・ コマンドプロンプトより TFTP クライアントを起動します。  
「tftp」とオプションなしで入力するとコマンドヘルプが表示されます。



```

C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\%a5097702>tftp
TFTP サービスを実行しているリモート コンピューター間とファイルを転送します。

TFTP [-i] host [GET | PUT] source [destination]

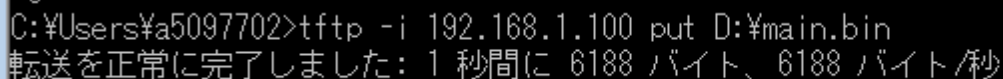
-i          バイナリ イメージ転送モード（オクテットとも呼ばれる）を指定
           します。バイナリ イメージ モードでは、ファイルは文字どおりバイト
           ごとに移動されます。バイナリ ファイルを転送するときには、こ
           のモードを使用してください。
host       ローカル ホストまたはリモート ホストを指定します。
GET        リモート ホスト上のファイル destination をローカル ホスト上の
           ファイル source に転送します。
PUT        ローカル ホスト上のファイル source をリモート ホスト上のファイル
           destination に転送します。
source     転送するファイルを指定します。
destination ファイルの転送先を指定します。

C:\Users\%a5097702>

```

- ・ バイナリ形式のファイルを TFTP サーバーへ送信する例を示します。

host = 192.168.1.100 : TFTP サーバーアドレス  
source = main.bin : 転送対象ファイル



```

C:\Users\%a5097702>tftp -i 192.168.1.100 put D:\main.bin
転送を正常に完了しました: 1 秒間に 6188 バイト、6188 バイト/秒

```



## 6. ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.07.11	-	日本語版 初版発行

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子

（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違くと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

○ARM および Cortex は、ARM Limited(またはその子会社)の EU またはその他の国における登録商標です。  
All rights reserved.

○Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。

○TRON は” The Real-time Operation system Nucleus” の略称です。

○ITRON は” Industrial TRON” の略称です。

○ $\mu$ ITRON は” Micro Industrial TRON” の略称です。

○TRON、ITRON、および $\mu$ ITRON は、特定の商品ないし商品群を指す名称ではありません。

○その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれが生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
  5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
  10. 当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  11. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
  12. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 注1. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。  
注2. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

(Rev.3.0-1 2016.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>