

# S1S60000

## アプリケーションノート

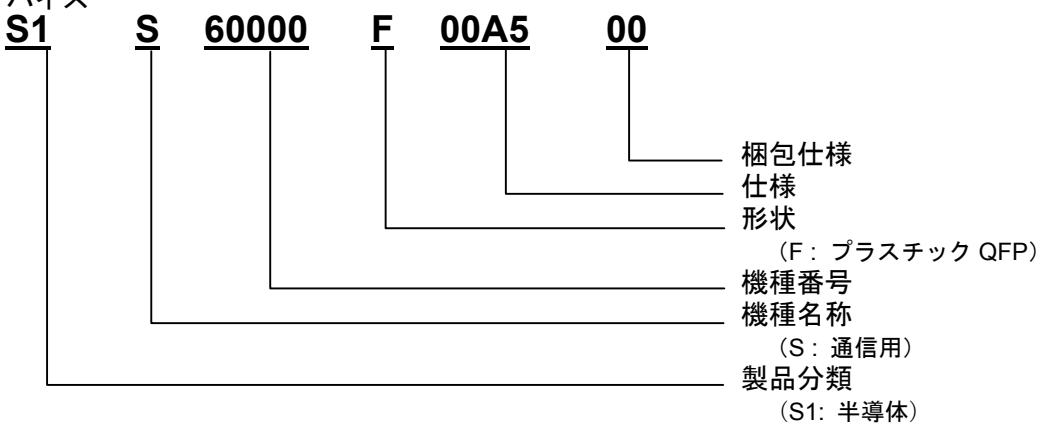
No.4 TCP/IP 通信端点使用方法

本資料のご使用につきましては、次の点にご留意願います。

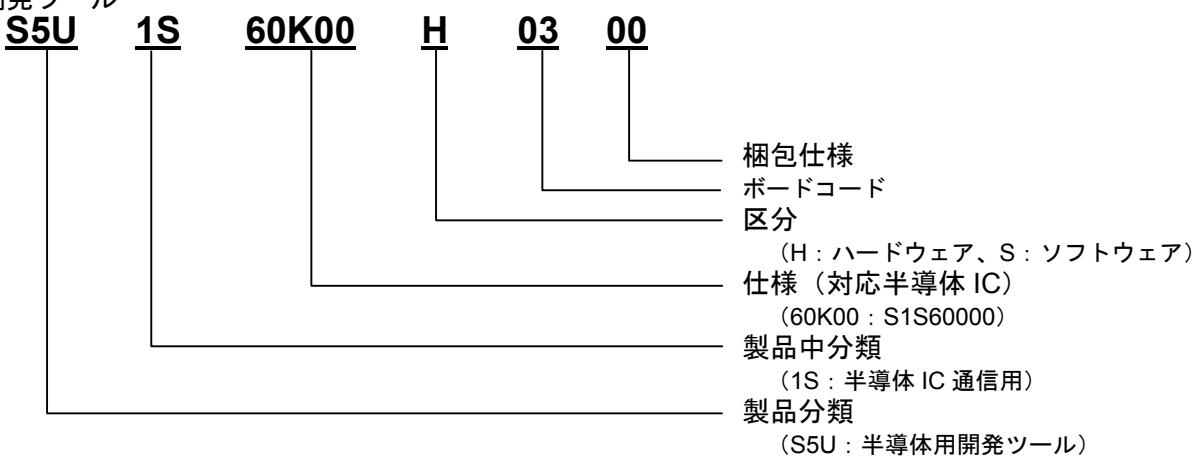
- 
1. 本資料の内容については、予告無く変更することがあります。
  2. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。
  3. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これら起因する第三者の権利（工業所有権を含む）侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の工業所有権の実施件の許諾を行うものではありません。
  4. 特性値の数値の大小は、数直線上の大小関係で表しています。
  5. 本資料に掲載されている製品のうち「外国為替及び外国貿易法」に定める戦略物資に該当するものについては、輸出する場合、同法に基づく輸出許可が必要です。
  6. 本資料に掲載されている製品は、一般民生用です。生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。

## 製品型番体系

### ●デバイス



### ●開発ツール



## 目 次

1. はじめに.....	1
2. TCP端点オープン前の設定.....	1
3. TCP通信端点のオープン .....	1
3.1 TCP通信端点 0 (TCP0) のオープン .....	2
3.2 接続の確立の確認 .....	3
4. データの受信 .....	6
5. データの送信 .....	9
6. TCP接続の切断.....	12
6.1 ホストCPUからの接続の切断.....	12
6.2 リモートマシンからの接続の切断 .....	14

## 1. はじめに

この文書は、S1S60000 の TCP 通信端点オープン方法とデータ送受信方法の基本手順を説明します。S1S60000 の TCP 通信端点をオープンする事により TCP/IP による S1S60000 とネットワーク機器とのデータ送受信が行えます。

## 2. TCP 通信端点オープン前の設定

あらかじめ S1S60000 の SYSTEM 通信端点をオープンして下さい。SYSTEM 通信端点のオープン方法については、“アプリケーションノート ping 応答方法”を参照して下さい。ここでは IP アドレス 192.168.0.254 で SYSTEM 通信端点をオープンします。

## 3. TCP 通信端点のオープン

S1S60000 は、4 つの TCP 用の通信端点 (TCP0~TCP3) を持っています。ここでは TCP0 (端点番号 2) を使用して S1S60000 がリモートマシンから接続を待つ passive open (サーバ動作) による TCP 接続を行います。ホスト CPU と S1S60000 とのコマンド、ステータス、データの送受信については “ping 応答方法”を参照して下さい。

この章ではホスト CPU からの TCP 通信端点オープン、接続確認を行います。手順は以下のようになります。

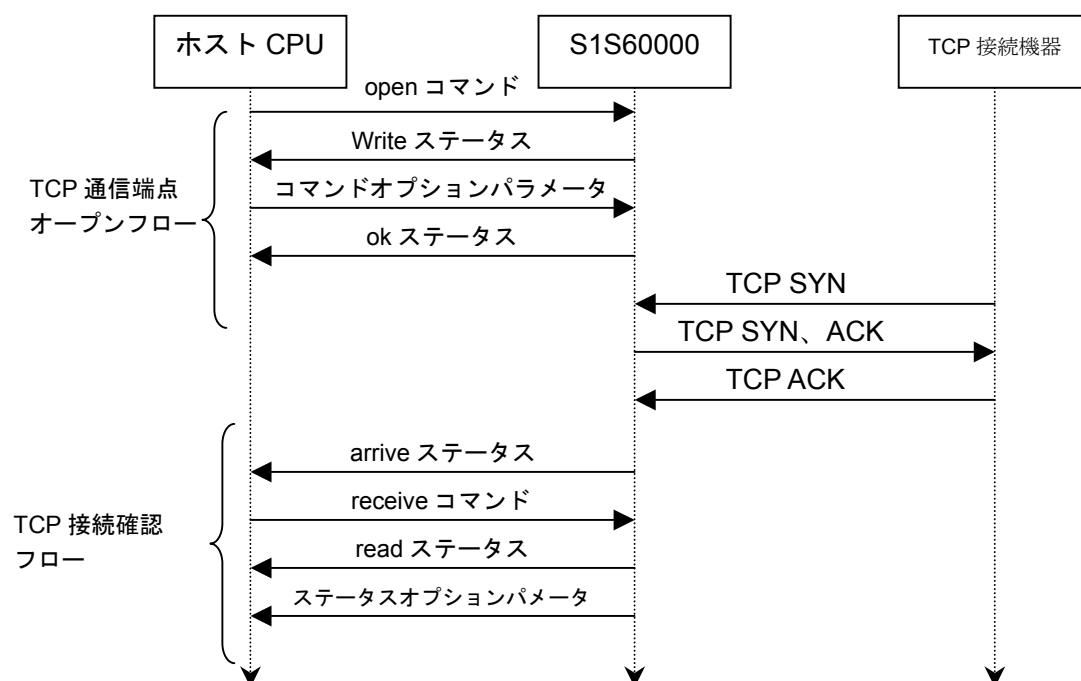


図 3-1 TCP 通信端点オープンフロー

### 3. TCP 通信端点のオープン

#### 3.1 TCP 通信端点 0 (TCP0) のオープン

ホスト CPU は、S1S60000 のコマンドポートに open コマンド (0x0120 : シーケンス番号 01、通信端点番号 2=TCP0、コマンド番号 0=open) を発行して下さい。

S1S60000 は write ステータス (0x0121:シーケンス番号 01、通信端点番号 2=TCP0、ステータス番号 1=write) を返します。ステータスポートから write ステータスを読み出して下さい。ここでは TCP 通信端点を passive open、相手の IP アドレス無効、相手のポート番号無効、自身のポート番号有効 (49152)、タイムアウト値無効でオープンします。オープンするパラメータを次のようにホスト CPU のメモリ上に作成しておき、データポートへ書き込んで下さい。

表 3-1 コマンドオプションパラメータ設定値

BYTE	書き込み値	内容	コメント
0	0x00	固定値	
1	0x00		
2	0x10	フラグ	Bit7(passive open)=0、 Bit6(相手の IP アドレス無効)=0、 Bit5(相手のポート番号無効)=0、 Bit4(自身のポート番号有効)=1、 Bit3(予約)=0、 Bit2(タイムアウト値無効)=0、 Bit1(予約)=0、 Bit0(予約)=0
3	0x00	固定値	
4	0x00	相手のIPアドレス	無効 指定しません
5	0x00		
6	0x00		
7	0x00		
8	0x00	相手のポート番号	無効 指定しません
9	0x00		
10	0xc0	自身のポート番号	49152(0xc000)
11	0x00		
12	0x00	タイムアウト値	タイムアウト値無効 0
13	0x00		
14	0x00	固定値	
15	0x00		

オプションパラメータをメモリに格納すると表 3-1 の BYTE0 がメモリの下位アドレスになります。

ホスト CPU タイプ別の書き込みデータ列は次のようにになります。

表 3-2 ホスト CPU タイプ別データ列

ホスト CPU タイプ	データ列
8bit 接続 LittleEndian	0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xc0,0x00,0x00,0x00,0x00,0x00,0x00
8bit 接続 BigEndian	0x00,0x00,0x10,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xc0,0x00,0x00,0x00,0x00,0x00,0x00
16bit 接続 LittleEndian	0x0000,0x0010,0x0000,0x0000,0x0000,0x00c0,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000
16bit 接続 BigEndian	0x0000,0x1000,0x0000,0x0000,0x0000,0xc000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

ok ステータス (0x0123 : シーケンス番号 01、通信端点番号 2=TCP0、コマンド番号 3=ok) が返れば open コマンドの処理は終了です。ok のかわりに working ステータス (0x0124) が返る場合、この通信端点は open コマンド発行済みか、close コマンド発行後の非同期処理の途中です。この通信端点に対して status コマンド (0x0227 : シーケンス番号 02、通信端点番号 2=TCP0、コマンド番号 7=status) を発行して、端点の状態が time-wait か closed であることを確認してください。

### 3.2 接続の確立の確認

TCP 通信端点の場合、open コマンドに対して ok ステータスが返っても、まだ TCP の接続が確立していません。TCP 接続の確立は S1S60000 が自動で処理します。接続が確立すると S1S60000 は arrive ステータス (0x002d : シーケンス番号 00、通信端点番号 2=TCP0、ステータス番号 d=arrive) を出力します。ホスト CPU は receive コマンド (0x0322 : シーケンス番号 03、通信端点番号 2=TCP0、コマンド番号 2=receive) を発行してください。receive コマンドに対して S1S60000 が read ステータス 0x0322 (シーケンス番号 3、通信端点番号 2=TCP 通信端点 0、ステータス番号 2=read) を返しますので、データポートから read ステータスオプションパラメータを読み出してください。

接続を確立した相手が IP アドレス 192.168.0.2、ポート番号 1039 の場合、次のようなデータ列が読み出せます。

### 3. TCP 通信端点のオープン

---

表 3-3 receive-read ステータスオプションパラメータ読み出し値

BYTE	書き込み値	内容	コメント
0	0x00	受信データ長	TCP 制御フラグの受信なので通常はデータがありません
1	0x00		
2	0x6a	フラグ	Bit7(データ終了)=0、 Bit6(相手の IP アドレス有効)=1、 Bit5(相手のポート番号有効)=1、 Bit4(unicast)=0、 Bit3(制御フラグ有効)=1、 Bit2(予約)=0、 Bit1(データオフセット有効)=1、 Bit0(全データ長 無効)=0
3	0x00	固定値	
4	0xc0	相手のIPアドレス	192.168.0.2 (0xc0.0xa8.0x00.0x02)
5	0xa8		
6	0x00		
7	0x02		
8	0x04	相手のポート番号	1039 (0x040f)
9	0x0f		
10	0x12	制御フラグ	TCP 制御フラグ Bit4(ACK フラグ)=1、 Bit1(SYN フラグ)=1
11	0x00		
12	0x00	データオフセット	
13	0x00		
14	0x00	データ全長	
15	0x00		

オプションパラメータをメモリに格納すると表 3-3 の BYTE0 がメモリの下位アドレスになります。

ホスト CPU タイプ別の読み出しデータ列は次のようにになります。

表 3-4 ホスト CPU タイプ別データ列

ホスト CPU タイプ	データ列
8bit 接続 LittleEndian	0x00,0x00,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x12,0x00,0x00,0x00,0x00,0x00
8bit 接続 BigEndian	0x00,0x00,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x12,0x00,0x00,0x00,0x00,0x00
16bit 接続 LittleEndian	0x0000,0x006a,0xa8c0,0x0200,0x0f04,0x0012,0x0000,0x0000
16bit 接続 BigEndian	0x0000,0x6a00,0xc0a8,0x0002,0x040f,0x1200,0x0000,0x0000

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

ここで BYTE10 で SYN フラグが存在し (Bit1=1)、FIN フラグが存在しない (Bit0=0) ことが、TCP 接続が成立したことを示しています。受信データ長が 0 なので、read ステータスオプションに続く受信データはありません。

## 4. データの受信

### 4. データの受信

この章ではネットワークから S1S60000 がデータを受信した場合のホスト CPU による読み出し手順を説明します。手順は以下のようになります。

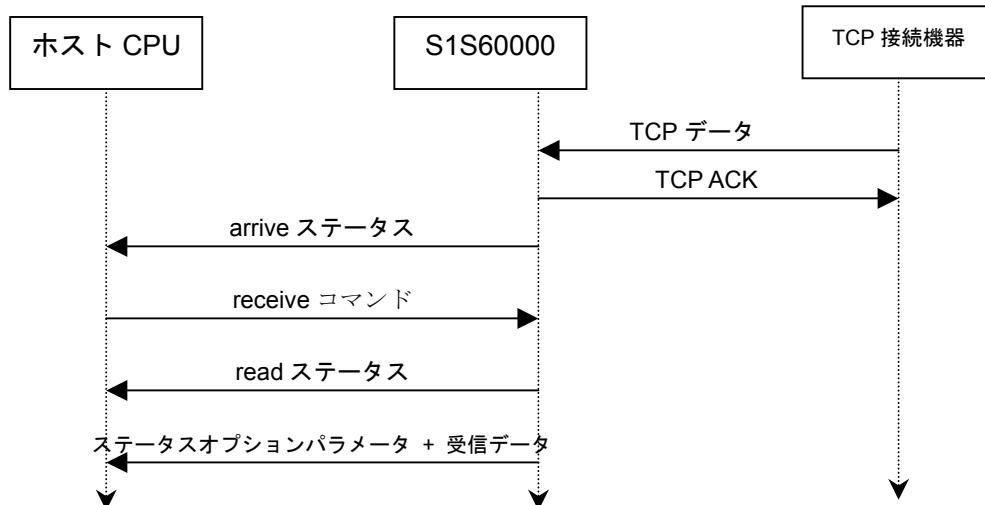


図 4-1 ホスト CPU データ受信フロー

S1S60000 はネットワークからデータを受信すると、ホスト CPU に arrive ステータス (0x002d : シーケンス番号 00、通信端点番号 2=TCP0、ステータス番号 d=arrive) を出力します。ホスト CPU は receive コマンド (0x0422 : シーケンス番号 04、通信端点番号 2=TCP0、コマンド番号 2=receive) を発行してください。receive コマンドに対応する read ステータス (0x0422:シーケンス番号 04、通信端点番号 2=TCP0、ステータス番号 2=read) が返りますので、データポートから受信データを読み出してください。  
read ステータスオプションパラメータは TCP 接続が確立したときと同じ形式です。

read ステータスオプションの先頭 2 バイトが受信データ長です。単位はバイトですので、受信データ長を n とすると、データポートから 16 ビット単位で受信データを読み出す回数は n が偶数の時は  $n/2$  回、n が奇数のときは  $(n+1)/2$  回になります。8 ビット単位で受信データを読み出す回数は、n が偶数であれば n 回、奇数であれば  $(n+1)$  回になります。必要な回数だけデータポートを読み出してください。

受信データが文字列 “0123456789A” で 11 バイトのキャラクタコード (16 進) を受信した場合、次のようなデータ列が読み出せます。

表 4-1 receive-read ステータスオプションと受信データの読み出し値

BYTE	書き込み値	内容	コメント
0	0x00	受信データ長	受信データ 11 バイト
1	0x0b		
2	0x6a	フラグ	Bit7(データ終了)=0、 Bit6(相手の IP アドレス有効)=1、 Bit5(相手のポート番号有効)=1、 Bit4(unicast)=0、 Bit3(制御フラグ有効)=1、 Bit2(予約)=0、 Bit1(データオフセット有効)=1、 Bit0(予約)=0
3	0x00	固定値	
4	0xc0	相手の IP アドレス	192.168.0.2 (0xc0.0xa8.0x00.0x02)
5	0xa8		
6	0x00		
7	0x02		
8	0x04	相手のポート番号	1039 (0x040f)
9	0x0f		
10	0x18	制御フラグ	TCP 制御フラグ Bit4(ACK フラグ)=1、 Bit3(PSH フラグ)=1
11	0x00		
12	0x00	データオフセット	TCP 通信端点では意味はありません
13	0x00		
14	0x00	データ全長	
15	0x00		
16	0x30	受信データ	文字列 “0123456789A” の 11 バイトのキャラクタコード（16 進）を受信
17	0x31		
18	0x32		
19	0x33		
20	0x34		
21	0x35		
22	0x36		
23	0x37		
24	0x38		
25	0x39		
26	0x41		
27	0xXX	padding	データが奇数バイトなのでパディングします

オプションパラメータをメモリに格納すると表 4-1 の BYTE0 がメモリの下位アドレスになります。

## 4. データの受信

---

ホスト CPU タイプ別の読み出しデータ列は次のようにになります。

表 4-2 ホスト CPU タイプ別データ列

ホスト CPU タイプ	データ列
8bit 接続 LittleEndian	0x00,0x0b,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x18,0x00,0x00,0x00,0x00,0x00 0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x41,0xXX
8bit 接続 BigEndian	0x00,0x0b,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x18,0x00,0x00,0x00,0x00,0x00 0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x41,0xXX
16bit 接続 LittleEndian	0x0b00,0x006a,0xa8c0,0x0200,0x0f04,0x0018,0x0000,0x0000, 0x3130,0x3332,0x3534,0x3736,0x3938,0xXX41
16bit 接続 BigEndian	0x000b,0x6a00,0xc0a8,0x0002,0x040f,0x1800,0x0000,0x0000, 0x3031,0x3233,0x3435,0x3637,0x3839,0x41XX

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

## 5. データの送信

この章ではネットワークへS1S60000がデータを送信する場合のホストCPUの手順を説明します。手順は以下のようになります。

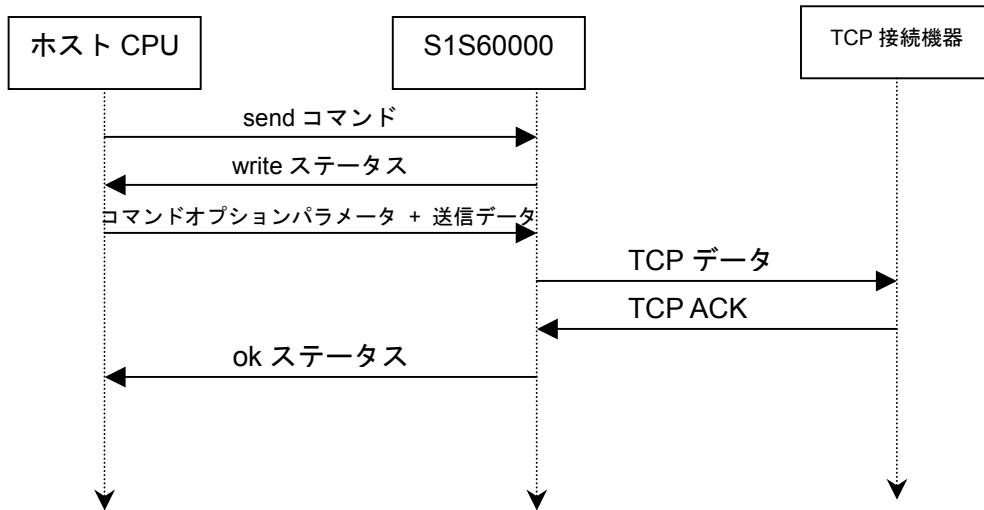


図 5-1 ホスト CPU データ送信フロー

ホストCPUはsendコマンド(0x0521:シーケンス番号05、通信端点番号2=TCP0、コマンド番号1=send)を発行してください。S1S60000がwriteステータス(0x0521:シーケンス番号05、通信端点番号2=TCP0、ステータス番号1=write)を返しますので、データポートにコマンドオプションパラメータと、それにつづけて送信データを書き込んでください。ここでは文字列“0123456789A”の11バイトのキャラクタコード(16進)を送信します。

## 5. データの送信

---

表 5-1 コマンドオプションパラメータと送信データ値

BYTE	書き込み値	内容	コメント
0	0x00	送信データ長	送信データ 11 バイト
1	0x0b		
2	0x00	フラグ	Bit7-6(予約)=0、 Bit3(制御フラグ無効)=0、 Bit2-0(予約)=0
3	0x00	固定値	
4	0x00		
5	0x00		
6	0x00		
7	0x00		
8	0x00		
9	0x00		
10	0x00	制御フラグ	TCP 制御フラグ フラグ設定無し
11	0x00		
12	0x00	固定値	
13	0x00		
14	0x00		
15	0x00		
16	0x30	送信データ	文字列 “0123456789A” の 11 バイトのキャラクタコード（16 進）を送信
17	0x31		
18	0x32		
19	0x33		
20	0x34		
21	0x35		
22	0x36		
23	0x37		
24	0x38		
25	0x39		
26	0x41		
27	0xXX	padding	データが奇数バイトなのでパディングします

オプションパラメータをメモリに格納すると表 5-1 の BYTE0 がメモリの下位アドレスになります。

ホスト CPU タイプ別の書き込みデータ列は次のようにになります。

表 5-2 ホスト CPU タイプ別データ列

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

16 ビット単位で送信データを書き込む回数は、送信データ長  $n$  (byte) が偶数の時は  $n/2$  回、 $n$  が奇数のときは  $(n+1)/2$  回になります。

ここでは送信データ長  $n=11$  なので、データポートへ 16 ビット単位で送信データを書き込む回数は  $(n+1)/2 = 6$  (コマンドオプション 16 バイト/2=8 と合わせて 14 回の書き込み) となります。

8 ビット単位で送信データを書き込む回数は、n が偶数であれば n 回、奇数であれば (n+1) 回になります。ホスト CPU は必要な回数だけデータポートに書き込みを行って下さい。TCP の再送は S1S60000 が自動で処理します。通信相手先にデータが受け取られたことを確認できると S1S60000 は ok ステータス (0x0523: シーケンス番号 5、通信端点番号 2=TCP0、コマンド番号 3=ok) を返します。

再送を繰り返したにもかかわらず send コマンドに失敗すると、S1S60000 は ok ではなく cancel ステータス (0x0528: シーケンス番号 05、通信端点番号 2=TCP0、ステータス番号 8=cancel) を返します。cancel ステータスを受け取った場合は以後の通信ができないと思われますので接続を終了してください。通常の接続終了には close コマンドを利用しますが、cancel ステータスが返った場合は相手先との通信ができないと思われますので、abort コマンドを利用してください。接続の終了についての詳細は、次章の TCP 接続の切断にて説明します。

### 6. TCP 接続の切断

この章では TCP 接続をホスト CPU から切断要求を行い切断する手順とリモートマシンから切断が要求された場合の手順を説明します。

#### 6.1 ホスト CPU からの接続の切断

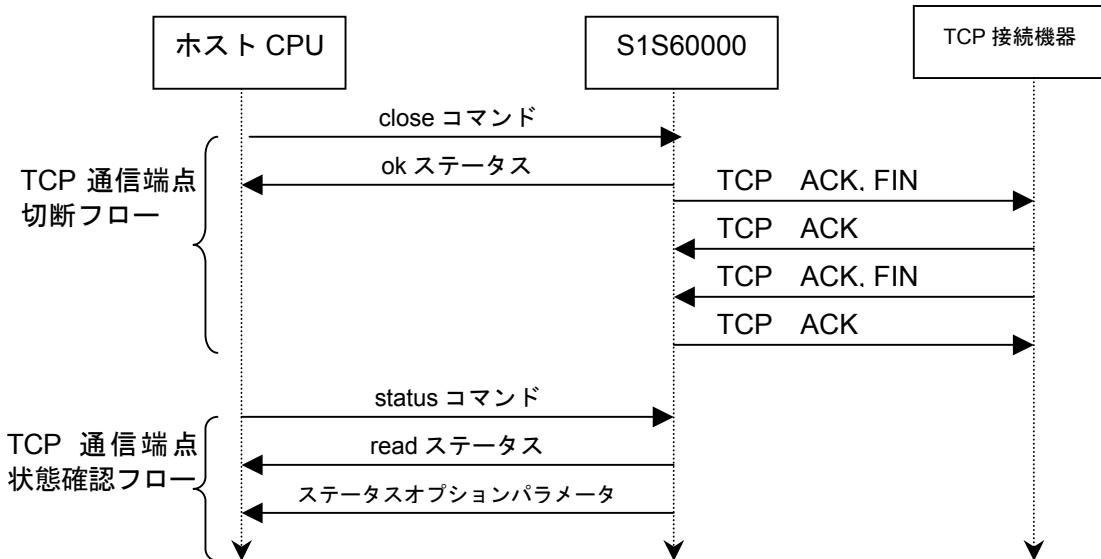


図 6-1 CP 通信端点切断フロー

ホスト CPU 側から切断を要求する場合は close コマンド (0x0624 : シーケンス番号 06、通信端点番号 2=TCP0、コマンド番号 4=close) を発行してください。S1S60000 は ok ステータス (0x0623 : シーケンス番号 06、通信端点番号 2=TCP0、ステータス番号 3=ok) を返します。その後、S1S60000 は非同期に切断処理を進めます。切断処理が終了するまでの時間は通信相手の応答時間に依存します。端点の状態が closed であれば再度 open できます。この通信端点に対して status コマンド (0x0727 : シーケンス番号 07、通信端点番号 2=TCP0、コマンド番号 7=status) を発行して確認が行えます。status コマンドに対して S1S60000 が read ステータス (0x0722 : シーケンス番号 07、通信端点番号 2=TCP0、ステータス番号 2=read) を返しますので、データポートから read ステータスオプションパラメータを読み出してください。通信端点が closed 状態であれば次のようなバイト列が読み出せます。

表 6-1 status-read ステータスオプションパラメータ読み出し値

BYTE	書き込み値	内容	コメント
0	0x00	固定値	
1	0x00		
2	0x00	フラグ	Closed 状態なので値は 0 になります
3	0x00	固定値	
4	0x00	相手の IP アドレス	Closed 状態なので値は 0 になります
5	0x00		
6	0x00		
7	0x00		
8	0x00		
9	0x00	相手のポート番号	
10	0x00		
11	0x00	自身のポート番号	
12	0x00		
13	0x00	タイムアウト	
14	0x01	通信端点の状態	TCP 通信端点の場合 0x01=close 状態
15	0x00	固定値	

オプションパラメータをメモリに格納すると表 6-1 の BYTE0 がメモリの下位アドレスになります。ホスト CPU タイプ別の読み出しデータ列は次のようにになります。

表 6-2 ホスト CPU タイプ別データ列

ホスト CPU タイプ	データ列
8bit 接続 LittleEndian	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00
8bit 接続 BigEndian	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00
16bit 接続 LittleEndian	0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0001
16bit 接続 BigEndian	0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0100

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

ホスト CPU 側から切断を要求する場合に abort コマンドを発行することもできます。この場合も S1S60000 は ok ステータスを返します。close コマンドと異なり、S1S60000 は通信相手に RST フラグを送信しただけで端点の状態を closed にします。

abort コマンドによる切断は正規の手順ではありません。相手からの応答が無い等のやむを得ない状況で使用して下さい。

## 6. TCP 接続の切断

### 6.2 リモートマシンからの接続の切断

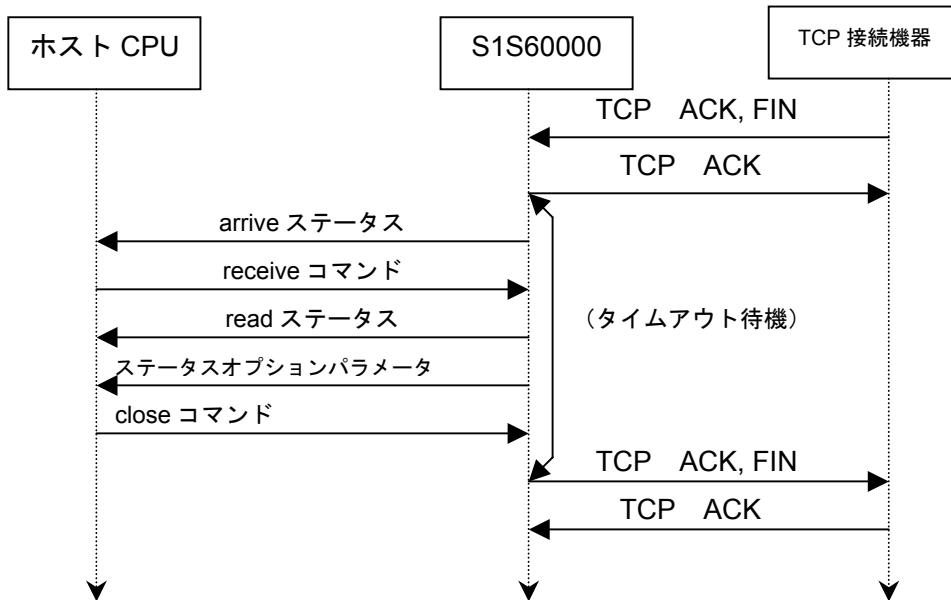


図 6-2 TCP 通信端点切断フロー

リモートマシンから切断を要求された場合、S1S60000 は接続の確立の確認やデータの受信と同じく `arrive` ステータス (0x002d : シーケンス番号 00、通信端点番号 2=TCP0、ステータス番号 d=`arrive`) を出力します。ホスト CPU は `receive` コマンド (0x0822 : シーケンス番号 08、通信端点番号 2=TCP0、コマンド番号 2=`receive`) を発行してください。`receive` コマンドに対して S1S60000 が `read` ステータス (0x0822 : シーケンス番号 08、通信端点番号 2=TCP 通信端点 0、ステータス番号 2=`read`) を返しますので、データポートから `read` ステータスオプションパラメータを読み出してください。リモートマシンが切断を要求し、S1S60000 が FIN フラグ (BYTE10 Bit0=1) を受信した場合、表 6-3 のようなバイト列が読み出せます。

また、ホスト CPU は `read` ステータスオプションパラメータを読み出した後に `close` コマンド (0x0924: シーケンス番号 09、通信端点番号 2=TCP0、コマンド番号 4=`close`) を発行してください。S1S60000 は `close` コマンドを受けるとリモートマシンへ切断終了通知をおこないます。ただし `close` コマンドを発行しない場合、S1S60000 はリモートマシンからの切断要求を受けた直後から、設定されたタイムアウト時間で待機した後にリモートマシンへ切断終了通知を自動的に発行します。

表 6-3 receive-read ステータスオプションパラメータ読み出し値

BYTE	書き込み値	内容	コメント
0	0x00	受信データ長	TCP 制御フラグの受信なので通常はデータがありません
1	0x00		
2	0x6a	フラグ	Bit7(データ終了)=0、 Bit6(相手の IP アドレス有効)=1、 Bit5(相手のポート番号有効)=1、 Bit4(unicast)=0、 Bit3(制御フラグ有効)=1、 Bit2(予約)=0、 Bit1(データオフセット有効)=1、 Bit0(予約)=0
3	0x00	固定値	
4	0xc0	相手のIPアドレス	192.168.0.2 (0xc0.0xa8.0x00.0x02)
5	0xa8		
6	0x00		
7	0x02		
8	0x04		
9	0x0f	相手のポート番号	1039 (0x040f)
10	0x11		
11	0x00		
12	0x00	データオフセット	
13	0x00		
14	0x00	データ全長	
15	0x00		

オプションパラメータをメモリに格納すると表 6-3 の BYTE0 がメモリの下位アドレスになります。

## 6. TCP 接続の切断

---

ホスト CPU タイプ別の読み出しデータ列は次のようにになります。

表 6-4 ホスト CPU タイプ別データ列

ホスト CPU タイプ	データ列
8bit 接続 LittleEndian	0x00,0x00,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x00,0x11,0x00,0x00,0x00,0x00
8bit 接続 BigEndian	0x00,0x00,0x6a,0x00,0xc0,0xa8,0x00,0x02,0x04,0x0f,0x00,0x11,0x00,0x00,0x00,0x00
16bit 接続 LittleEndian	0x0000,0x006a,0xa8c0,0x0200,0x0f04,0x1100,0x0000,0x0000
16bit 接続 BigEndian	0x0000,0x6a00,0xc0a8,0x0002,0x040f,0x0011,0x0000,0x0000

\* 8bit 接続時のアクセス順序：下位ポート→上位ポート

## **セイコーエプソン株式会社**

**半導体事業部 IC 営業部**

---

**<IC 東日本営業グループ>**

東京 〒191-8501 東京都日野市日野 421-8

TEL (042) 587-5313 (直通) FAX (042) 587-5116

**<IC 西日本営業グループ>**

大阪 〒541-0059 大阪市中央区博労町 3-5-1 エプソン大阪ビル 15F

TEL (06) 6120-6000 (代表) FAX (06) 6120-6100

名古屋 〒460-0008 名古屋市中区栄 1-10-21 名古屋御園ビル 6F

TEL (052) 205-8421 (代表) FAX (052) 231-2538