

AJAN
IOコマンド
リファレンス
シリアル通信編

目 次

第 1 章	はじめに	3
1.1	シリアル通信コマンドについて.....	3
1.2	信号名説明	4
第 2 章	機能説明	5
2.1	初期化機能	5
2.2	データ送信機能	8
2.3	データ受信機能	9
2.4	制御信号入出力機能	10
2.5	使用するハードウェアによる制限について	10
2.6	一般ユーザーでシリアル制御を行うには.....	11
第 3 章	リファレンス	12
3.1	コマンド一覧	12
3.2	詳細個別説明（初期化/終了）	13
3.3	詳細個別説明（送受信）	15
3.4	詳細個別説明（制御信号）	17
3.5	詳細個別説明（ステータス）	19
3.6	内部エラーについて	20
第 4 章	サンプルプログラム	21
4.1	サンプルプログラム一覧	21
第 5 章	索引	22
第 6 章	重要な情報	23

第1章 はじめに

本ドキュメントでは、AJANで利用できるI/O制御コマンドの使い方について記載しています。

本ドキュメントでは、説明で表現している表記として下記のように定義します。

- ・ コマンドの書式の説明において、[]内の引数は省略できます。
- ・ 文字の大小について

コマンドは大文字/小文字のどちらでも動作します。

変数名は大文字/小文字も同じものとして扱われます。

ファイルパス/ファイル名は大文字/小文字で区別されます。



本ドキュメント記載の、AJANはIoT用プログラミング言語です。
Interface Linux System上でのみ動作可能です。

1.1 シリアル通信コマンドについて

シリアル通信コマンドは、RS-232C/RS-485に準拠したシリアルポートを制御するためのコマンド群です。

標準シリアル（PCに標準搭載）は、本コマンドを使ってすぐに制御することができますが、弊社インタフェースモジュール等でシリアルポートを増設した場合は、Linuxドライバをインストールする必要があります。

インストールを行い、ドライバの組み込みまで実施する必要があります。

インストール方法、ドライバの組み込み方法については、ドライバ製品のreadme/helpを参照してください。

ドライバは、弊社Web siteサイトからダウンロードしてインストールを行ってください。

カテゴリ	型式	リンク先
シリアル通信	GPG-4141	http://www.interface.co.jp/catalog/soft/prdc_soft_all.asp?name=gpg-4141



本コマンドはスーパーユーザでのみ使用できます。

スーパーユーザモードで起動して使用してください。

ユーザ権限で使いたい場合は、シリアル通信を制御するためのデバイスノードの権限を変更してください。

デバイスノードは、/dev/に格納されている「**ttyXX**」という名前がデバイスノードとなります。

1.2 信号名説明

シリアル通信の説明に出てくる信号名と意味について、簡単に紹介します。

RS-232Cの信号名(9ピン D-Subコネクタの場合)

番号	信号名	別名	入出力方向	機能
1	CD	DCD	入力	キャリア検出
2	RD	RxD	入力	受信データ
3	SD	TxD	出力	送信データ
4	ER	DTR	出力	データ・ターミナル・レディ
5	SG	GND	—	シグナルグランド
6	DR	DSR	入力	データ・セット・レディ
7	RS	RTS	出力	送信要求(リクエスト・トゥ・センド)
8	CS	CTS	入力	送信可(クリア・トゥ・センド)
9	CI	RI	入力	呼出表示

RS-485の信号名(15ピン D-Subコネクタの場合)

番号	信号名		入出力方向	機能
2	T(A)	T-	入出力	送信(RS-232CのSDに対応)
3	C(A)	C-	出力	制御(RS-232CのRSに対応)
4	R(A)	R-	入力	受信(RS-232CのRDに対応)
5	I(A)	I-	入力	表示(RS-232CのCSに対応)
8	SG			シグナルグランド
9	T(B)	T+	入出力	送信(RS-232CのSDに対応)
10	C(B)	C+	出力	制御(RS-232CのRSに対応)
11	R(B)	R+	入力	受信(RS-232CのRDに対応)
12	I(B)	I+	入力	表示(RS-232CのCSに対応)

第2章 機能説明

2.1 初期化機能

「COMOPEN」を使って、デバイスをオープンし、通信を行うために必要な初期化を行います。

デバイスの指定を行う場合には、デバイスノード名を使用し、任意の番号（ここではCOM番号）に割り当てて使用します。

デバイスノード名の確認方法

Classembly Devices製品の場合は、各マニュアルに記載されています。

記載されていない場合は、/procファイルシステムを参照することで、各通信ポートのデバイスノード名が確認できます。

・標準シリアル通信の場合

例) 車載Classembly Devices Atom E3845 車載モデル

```
# cat /proc/tty/driver/serial
0: uart:16550A mmio:0xB0A19000 irq:18 tx:0 rx:0
1: uart:16550A mmio:0xB0A1B000 irq:18 tx:0 rx:0
2: uart:unknown port:000003E8 irq:4
3: uart:unknown port:000002E8 irq:3
```

上記の0,1が対応し、デバイスノード名は「ttyS0」と「ttyS1」となります。

・弊社 シリアル通信の場合

行頭のttyXX がデバイスノード名になります。

```
# cat /proc/tty/driver/cp4141
ttyS64: PCI-4141(bid=0h)CH1 [9600bps] tx:0 rx:0
ttyS65: PCI-4141(bid=0h)CH2 [9600bps] tx:0 rx:0
ttyS66: PCI-4142(bid=0h)CH1 [9600bps HD2W] tx:0 rx:0
ttyS67: PCI-4142(bid=0h)CH2 [9600bps HD2W] tx:0 rx:0
```

```
# cat /proc/tty/driver/cp4161
ttyG0: PCI-4161(bid=0h)CH1 [9600bps] tx:0 rx:0
ttyG1: PCI-4161(bid=0h)CH2 [9600bps] tx:0 rx:0
ttyG2: CBI-4641(bid=0h) (ID:0175421024) CH1 [9600bps] tx:0 rx:0
ttyG3: CBI-4641(bid=0h) (ID:0175421024) CH2 [9600bps] tx:0 rx:0
```

「/proc/tty/driver」の後に指定する名前については製品によって異なります。

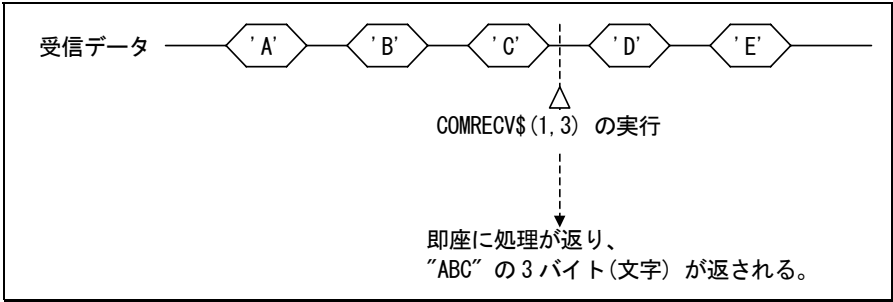
以下の対応表を確認してください。

インタフェースモジュール	ドライバ
PCI-4161, PCI-4661xx, LPC-400111, LPC-4661xx PEX-400111, PEX-4661xx, PEX-H4661xx	cp4161
上記以外	cp4141

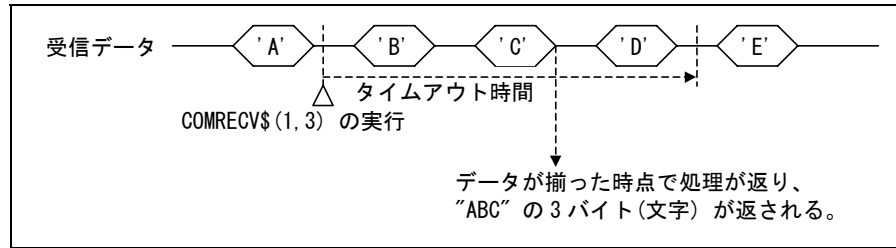
初期化に必要な項目は、以下の通りです。

なお、これらの各設定は使用するハードウェアによって制約がある場合があります。

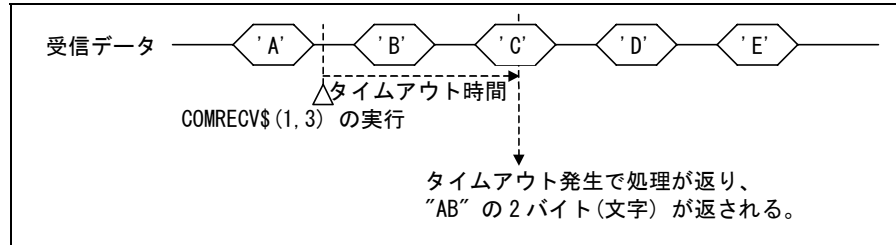
それぞれ、使用するハードウェアの仕様を確認してご使用ください。

項目	内容
転送速度	シリアル通信の転送速度(bps) を指定します。 この設定は、接続先と同じ設定とする必要があります。 転送速度は、50～2,000,000(bps) の範囲で指定することができます。 転送速度の指定を省略する事もできます。 転送速度の指定を省略した場合は、9600bpsになります。
パリティビット	シリアル通信のパリティビットを指定します。 この設定は、接続先と同じ設定とする必要があります。 パリティビットは、パリティなし、偶数パリティ、奇数パリティの3つのいずれか1つを指定することができます。 パリティビットの指定を省略することもできます。 パリティビットの指定を省略した場合は、パリティなしに設定されます。
データビット長	シリアル通信のデータビット長を指定します。 この設定は、接続先と同じ設定とする必要があります。 データビット長は、5～8(bit) の範囲で指定することができます。 データビット長の指定は省略する事もできます。 データビット長の指定を省略した場合は、自動で 8ビットに設定されます。
ストップビット長	シリアル通信のストップビット長を指定します。 ストップビット長は、1(bit), 1.5(bit), 2(bit) のいずれかを指定することができます。 この設定は、接続先と同じ設定とする必要があります。
通信方式	シリアル通信の通信方式を指定します。 通信方式は、全二重、2線式半二重、4線式半二重のいずれかを指定することができます。 この設定は、接続先と同じ設定とする必要があります。
フロー制御	シリアル通信のフロー制御方法を指定します。 フロー制御は、フロー制御なし、ハードウェアフロー(RS(C)/CS(I))、ソフトウェアフロー(XON/XOFF) のいずれかを指定することができます。 この設定は、接続先と同じ設定とする必要があります。
タイムアウト時間	<p>受信タイムアウト時間を 1ms単位で指定できます。</p> <p>「COMRECV\$」で受信バッファからデータを取り出す際、受信バッファには指定したサイズ未満の受信データしか存在しないことがあります。このような状態のとき、タイムアウト発生まで受信データが指定したデータサイズ以上になるのを待ちます。タイムアウトが発生した時、タイムアウト発生までに受信されたデータが返されます。</p> <p>「COMRECV\$」実行時の受信状態と、タイムアウト時間の関係は以下の通りです。 COMRECV\$, COMRECV関数実行時、既にデータが揃っている場合</p> 

- ・「COMRECV\$」実行時、データが揃っておらず、タイムアウト発生前にデータが揃った場合

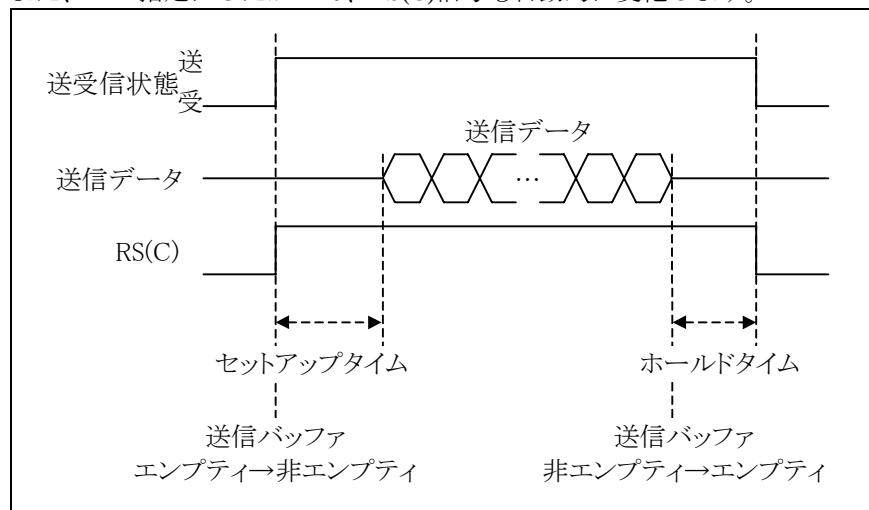


- ・「COMRECV\$」実行時、データが揃っておらず、タイムアウト発生した場合



セットアップタイム、
ホールドタイム

半二重通信を行う際の受信→送信切替時間(セットアップタイム) および、送信→受信切替時間(ホールドタイム) を $1\mu\text{s}$ 単位で指定できます。
また、この指定にしたがって、RS(C)信号も自動的に変化します。



●プログラム例

```
' ttyS0 のデバイスを COM 番号 1 として初期化する
' 115.2kbps パリティなし データ 8bit ストップ 1bit 全二重 フロー制御なし
' タイムアウト 100ms
COMOPEN 1, "ttyS0", "BAUDRATE=115200, TIMEOUT=100"
' 「TestData」 + 「CR+LF」を送信する
D$="TestData"+CHR$(&h0D)+CHR$(&h0A)
COMSEND 1, D$, LENB(D$)
COMCLOSE 1
END
```

2.2 データ送信機能

データを送信する機能です。

データ送信には「COMSEND」を使います。

「COMSEND」は、渡された送信データを送信バッファへセットすると即座に処理が返り、バックグラウンドで送信処理が行われます。

バックグラウンドで送信バッファのデータを送信中の状態でも、続けて次の送信データを「COMSEND」で送信バッファへセットすることができます。

●プログラム例

```
' ttyS0 のデバイスを COM 番号 1 として初期化する
' 115.2kbps パリティなし データ 8bit ストップ 1bit 全二重 フロー制御なし
' タイムアウト 100ms
COMOPEN 1, "ttyS0", "BAUDRATE=115200, TIMEOUT=100"
' 「TestData」 + 「CR+LF」を送信する
D$="TestData"+CHR$(&h0D)+CHR$(&h0A)
COMSEND 1, D$, LENB(D$)
COMCLOSE 1
END
```


2.3 データ受信機能

データを受信する機能です。

受信データは自動的にバックグラウンドで受信バッファへ蓄えられています。

「COMRCV\$」は、受信バッファへ蓄えられているデータを取り出します。

データ取り出し時に指定したサイズのデータがない場合、「COMOPEN」で指定するタイムアウト時間まで待ちます。

タイムアウト動作の詳細は、「2.1 初期化機能」の「タイムアウト時間」を参照してください。

●プログラム例

```
' ttyS0 のデバイスを COM 番号 1 として初期化する
' 115.2kbps パリティなし データ 8bit ストップ 1bit 全二重 フロー制御なし
' タイムアウト 100ms
COMOPEN 1, "ttyS0", "BAUDRATE=115200, TIMEOUT=100"
' 10 バイトの受信データを待つ
SIZE=COMRCVSIZE(1)
DO WHILE SIZE < 10
    SIZE=COMRCVSIZE(1)
LOOP
' 受信データを取り出して表示する
D$=COMRCV$(1, SIZE)
PRINT D$
' COM ポートをクローズして終了する
COMCLOSE 1
END
```

2.4 制御信号入出力機能

COMポートの制御信号 (RS(C), CS(I), CD, ER, DR, CI等) の入出力を行います。

出力方向の制御信号 (RS(C), ER) は、「COMMODEMLINE」命令で制御することができます。

入力方向の制御信号 (CS(I), CD, DR, CI) は、「COMMODEMLINE」関数で状態を読み出すことができます。

●プログラム例

```
' ttyS0 のデバイスを COM 番号 1 として初期化する
' 115.2kbps パリティなし データ 8bit ストップ 1bit 全二重 フロー制御なし
' タイムアウト 100ms
COMOPEN 1, "ttyS0", "BAUDRATE=115200, TIMEOUT=100"
' ER=ON, RS(C)=OFF にする
COMMODEMLINE 1, 2
' ポートをクローズする
COMCLOSE 1
END
```

```
' ttyS0 のデバイスを COM 番号 1 として初期化する
' 115.2kbps パリティなし データ 8bit ストップ 1bit 全二重 フロー制御なし
' タイムアウト 100ms
COMOPEN 1, "ttyS0", "BAUDRATE=115200, TIMEOUT=100"
' 制御信号の状態を調べて表示する
MODEMLINE=COMMODEMLINE(1)
PRINT MODEMLINE
' ポートをクローズする
COMCLOSE 1
END
```

2.5 使用するハードウェアによる制限について

使用するハードウェアによって、使用できる機能に制約がある場合があります。

ハードウェア毎の制約事項については、弊社 [webページ](#)を参照してください。

2.6 一般ユーザーでシリアル制御を行うには

本コマンドは、管理者権限(スーパーユーザー)でのみシリアル制御が可能です。

これは、シリアル通信を制御するデバイスノードの権限によるためです。

一般ユーザーで、本コマンドを使用するには、以下の手順で、権限を付与してください。

1. 画面左上の「アプリケーション」メニューから、「システムツール」→「端末」を選択し、端末ウィンドウを開きます。
2. 以下のようにコマンドを入力し、シリアル通信で使用するデバイス名と、所属するグループを確認します。

下記事例は、デバイス名が「ttyS0」の場合のものです。

ここで、赤文字部分が、デバイスが所属するグループ名です。覚えておいてください。

```
$ ls -l /dev
...
crw-rw---- 1 root dialout 4, 64 5月 11 20:31 ttyS0
crw-rw---- 1 root dialout 4, 65 5月 11 20:31 ttyS1
...
```

3. 以下のようにコマンドを入力し、あなたのユーザーを、2. で確認したグループに所属させます。
下記事例は、グループ名が「dialout」、所属させたいユーザー名が「user」の場合のものです。
グループ名、ユーザー名が異なる場合、任意のものに変更してください。

```
$ sudo adduser user dialout
```

adduser コマンドについて

本コマンドはスーパーユーザで実行する必要があるため、冒頭に sudo を付加して実行します。

パスワードの入力を求められた場合は、ログイン時のパスワードを入力してください。

コマンドが実行できなかった場合は、スーパーユーザ（管理者）でログインして本コマンドを実行してください。スーパーユーザ（管理者）でログインする方法は、OS マニュアルを参照してください。

このように、制御したいデバイスのグループに、ユーザーを所属させる事で、スーパーユーザーにならずとも、シリアル通信が制御できるようになります。

第3章 リファレンス

3.1 コマンド一覧



本コマンドは、管理者権限(スーパーユーザー)で実行する必要があります。

コマンド名	機能
●初期化/終了	
COMOPEN	シリアル通信デバイスをオープンします。
COMCLOSE	シリアル通信デバイスをクローズします。
●送受信	
COMSEND	データを送信します。
COMRECV\$	受信データを取得します。
COMRECFSIZE	受信バッファに蓄積されているデータサイズ(byte)を取得します。
●制御信号	
COMMODEMLINE	制御信号の出力を制御します。
COMMODEMLINE	制御信号の入力状態を取得します。
●ステータス	
COMSTATUS	シリアル通信の情報を取得します。

3.2 詳細個別説明（初期化/終了）

1.COMOPEN

命令																																																																										
機能	シリアル通信デバイスをオープンします。																																																																									
書式	COMOPEN <①COM番号>, <②デバイス名>, [<③初期化設定文字列>]																																																																									
パラメータ	①	<COM番号>	数値																																																																							
	割り当てるCOM番号を指定します。																																																																									
	②	<デバイス名>	文字列																																																																							
	デバイスのデバイスノード名を文字列で指定します。 ※デバイスノード名については、「2.1 初期化機能」を参照してください。																																																																									
	③	<初期化設定文字列>	文字列																																																																							
デバイスの初期設定を行います。 「識別子=設定値, 識別子=設定値, …」のような文字列で渡してください。 省略すると、以下の設定と同じになります。 "BAUDRATE=9600, PARITY=NONE, DATA=8, STOP=1, DUPLEX=FULL, FLOW=NONE, TIMEOUT=10, STIME=1000, HTIME=1000"																																																																										
<table border="1"> <thead> <tr> <th>識別子</th><th>設定値</th><th>内容</th><th>デフォルト</th></tr> </thead> <tbody> <tr> <td>BAUDRATE</td><td>50～2,000,000</td><td>転送速度 (bps)</td><td>9600</td></tr> <tr> <td rowspan="3">PARITY</td><td>NONE</td><td>パリティなし</td><td>○</td></tr> <tr> <td>EVEN</td><td>偶数パリティ</td><td></td></tr> <tr> <td>ODD</td><td>奇数パリティ</td><td></td></tr> <tr> <td rowspan="4">CHECK</td><td>NONE</td><td>入力パリティの追加設定なし</td><td>○</td></tr> <tr> <td>IGNORE</td><td>入力パリティエラーを無視</td><td></td></tr> <tr> <td>CLEAR</td><td>入力パリティを有効 (パリティエラーの受信データは、0になります)</td><td></td></tr> <tr> <td>MARK</td><td>入力パリティエラーの前に、&HFF, &H00 を挿入</td><td></td></tr> <tr> <td>DATA</td><td>5 / 6 / 7 / 8</td><td>データビット長</td><td>8</td></tr> <tr> <td>STOP</td><td>1 / 2</td><td>ストップビット長</td><td>1</td></tr> <tr> <td rowspan="3">DUPLEX</td><td>FULL</td><td>全二重</td><td>○</td></tr> <tr> <td>HALF</td><td>2線式半二重</td><td></td></tr> <tr> <td>HALF4W</td><td>4線式半二重</td><td></td></tr> <tr> <td rowspan="3">FLOW</td><td>NONE</td><td>なし</td><td>○</td></tr> <tr> <td>XONXOFF</td><td>XON/OFFによるソフトウェアフロー</td><td></td></tr> <tr> <td>RTSCTS</td><td>RS/CSによるハードウェアフロー</td><td></td></tr> <tr> <td>TIMEOUT</td><td>0～4,294,967,295</td><td>受信タイムアウト時間(ms)</td><td>10ms</td></tr> <tr> <td>STIME</td><td>0～150,000</td><td>セットアップタイム (μs)</td><td>1,000 μs</td></tr> <tr> <td>HTIME</td><td>0～150,000</td><td>ホールドタイム (μs)</td><td>1,000 μs</td></tr> </tbody> </table>				識別子	設定値	内容	デフォルト	BAUDRATE	50～2,000,000	転送速度 (bps)	9600	PARITY	NONE	パリティなし	○	EVEN	偶数パリティ		ODD	奇数パリティ		CHECK	NONE	入力パリティの追加設定なし	○	IGNORE	入力パリティエラーを無視		CLEAR	入力パリティを有効 (パリティエラーの受信データは、0になります)		MARK	入力パリティエラーの前に、&HFF, &H00 を挿入		DATA	5 / 6 / 7 / 8	データビット長	8	STOP	1 / 2	ストップビット長	1	DUPLEX	FULL	全二重	○	HALF	2線式半二重		HALF4W	4線式半二重		FLOW	NONE	なし	○	XONXOFF	XON/OFFによるソフトウェアフロー		RTSCTS	RS/CSによるハードウェアフロー		TIMEOUT	0～4,294,967,295	受信タイムアウト時間(ms)	10ms	STIME	0～150,000	セットアップタイム (μs)	1,000 μs	HTIME	0～150,000	ホールドタイム (μs)	1,000 μs
識別子	設定値	内容	デフォルト																																																																							
BAUDRATE	50～2,000,000	転送速度 (bps)	9600																																																																							
PARITY	NONE	パリティなし	○																																																																							
	EVEN	偶数パリティ																																																																								
	ODD	奇数パリティ																																																																								
CHECK	NONE	入力パリティの追加設定なし	○																																																																							
	IGNORE	入力パリティエラーを無視																																																																								
	CLEAR	入力パリティを有効 (パリティエラーの受信データは、0になります)																																																																								
	MARK	入力パリティエラーの前に、&HFF, &H00 を挿入																																																																								
DATA	5 / 6 / 7 / 8	データビット長	8																																																																							
STOP	1 / 2	ストップビット長	1																																																																							
DUPLEX	FULL	全二重	○																																																																							
	HALF	2線式半二重																																																																								
	HALF4W	4線式半二重																																																																								
FLOW	NONE	なし	○																																																																							
	XONXOFF	XON/OFFによるソフトウェアフロー																																																																								
	RTSCTS	RS/CSによるハードウェアフロー																																																																								
TIMEOUT	0～4,294,967,295	受信タイムアウト時間(ms)	10ms																																																																							
STIME	0～150,000	セットアップタイム (μs)	1,000 μs																																																																							
HTIME	0～150,000	ホールドタイム (μs)	1,000 μs																																																																							
備考	<ul style="list-style-type: none"> 転送速度の上限は、使用するハードウェアにより異なります。 ハードウェアの仕様をご確認ください。 セットアップタイム、ホールドタイムの設定は、半二重通信の場合のみ有効です。 セットアップタイム、ホールドタイムの設定は、以下の計算式で求められる値のみ設定可能です。 $\text{RATE} \times \text{CLK} \quad (\text{RATE} = 0 \sim 15 \quad \text{CLK} = 1, 10, 100, 1000, 10000 \quad \text{単位は } \mu s)$ データビット長が5ビットの時、ストップビット長を2ビットにすることはできません。 																																																																									
注意	本コマンドは、管理者権限(スーパーユーザー)で実行する必要があります。 これは、オープンするデバイスノードの権限設定によります。																																																																									
使用例	「2.1 初期化機能」のプログラム例を参照してください。 ' シリアルポートを、デバイスノード名 ttyS0、転送速度 115200bps でオープンします。 COMOPEN 1, "ttyS0", "BAUDRATE=115200"																																																																									

2.COMCLOSE

命令			
機 能	シリアル通信デバイスをクローズします。		
書 式	COMCLOSE <①COM番号>		
パラ メータ	①	<COM番号>	数値
使用例	<p>「2.1 初期化機能」のプログラム例を参照してください。</p> <p>' シリアルポートをオープンした後、クローズします。 COMOPEN 1, "ttyS0", "BAUDRATE=115200" COMCLOSE 1</p>		

3.3 詳細個別説明（送受信）

1. COMSEND

命令			
機能	データを送信します。		
書式	COMSEND <①COM番号>, <②送信データ> [, <③送信データサイズ>]		
パラメータ	①	<COM番号>	数値
	「COMOPEN」で指定したCOM番号を指定します。		
	②	<送信データ>	文字列
	送信データを文字列で指定します。		
パラメータ	③	<送信データサイズ>	数値
	<送信データ>のバイト数を1以上の整数で指定します。 省略すると、送信データ分のサイズが採用されます。		
備考	<ul style="list-style-type: none"> ・<送信データ>へ任意の1バイトを与えるには、「CHRB\$」関数を使用します。 ・<送信データサイズ>は、文字数ではなくバイト数です。 文字列のバイト数は、「LENB」関数で求めることができます。 ・「COMSEND」では、送信データを送信バッファへセットし終わると処理が返ります。 このため、「COMSEND」の実行が終わっても、送信データの送信が完了しているとは限りません。 		
使用例	<p>「2.2 データ送信機能」のプログラム例を参照してください。</p> <p>' 「Hello」という文字列をデータ送信します。 \$\$ = "Hello" COMSEND 1, \$\$, LENB(\$\$)</p>		

2.COMRCV\$

関数		
機 能	受信データを取得します。	
書 式	<(戻り値)受信データ>=COMRCV\$(<①COM番号>,<②受信データサイズ>)	
戻り値	戻り値	文字列
	<受信データ> 受信データが文字列として返されます。	
パラ メータ	①	数値
	<COM番号> 「COMOPEN」で指定したCOM番号を指定します。	
	②	数値
	<受信データサイズ> 受信バッファから取り出すデータのバイト数を1以上の整数で指定します。	
備 考	「COMRCV\$」実行時の受信バッファのデータ蓄積状態と、「COMOPEN」で指定した受信タイムアウト時間により挙動が変化します。	
使用例	「2.3 データ受信機能」のプログラム例を参照してください。 ' 受信バッファのサイズを取得します SZ = COMRCVSIZE(1) ' 受信バッファにデータがあれば、そのサイズ分、データを受信します IF SZ > 0 THEN S\$ = COMRCV\$(1, SZ) END IF	

3.COMRCVSIZE

関数		
機 能	受信バッファに蓄積されているデータサイズ(byte)を取得します。	
書 式	<(戻り値)受信バッファ内データサイズ>=COMRCVSIZE(<①COM番号>)	
戻り値	戻り値	数値
	<受信バッファ内データサイズ> 受信バッファに蓄積されているデータのバイト数を返します。	
パラ メータ	①	数値
	<COM番号> 「COMOPEN」で指定したCOM番号を指定します。	
使用例	「2.3 データ受信機能」のプログラム例を参照してください。 ' 受信バッファのサイズを取得します SZ = COMRCVSIZE(1) ' 受信バッファにデータがあれば、そのサイズ分、データを受信します IF SZ > 0 THEN S\$ = COMRCV\$(1, SZ) END IF	

3.4 詳細個別説明（制御信号）

1. COMMODEMLINE

命令																									
機能書式	制御信号の出力を制御します。																								
	COMMODEMLINE <①COM番号>, <②制御信号>																								
パラメータ	①	<COM番号>						数値																	
	「COMOPEN」で指定したCOM番号を指定します。																								
	②	<制御信号>						数値																	
	制御信号(ER, RS(C)) の出力を指定します。																								
	<table><tr><td>bit8</td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr><tr><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>RS</td><td>ER</td><td>—</td></tr></table>								bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	—	—	—	—	—	—	RS	ER
bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0																	
—	—	—	—	—	—	RS	ER	—																	
	<table><tr><th>信号名</th><th>内容</th></tr><tr><td>RS</td><td>RS(C) 信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>ER</td><td>ER信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)</td></tr></table>								信号名	内容	RS	RS(C) 信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)	ER	ER信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)											
信号名	内容																								
RS	RS(C) 信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)																								
ER	ER信号の出力状態を指定します。 1: ON(スペース) 0: OFF(マーク)																								
備考	RS-485では、C信号はRS、I信号はCSにそれぞれ対応しています。 RS-485では、DR, CD, CI, ERは使用しません。																								
使用例	「2.4 制御信号入出力機能」のプログラム例を参照してください。 'ER制御信号をONにします COMMODEMLINE 1, 2																								

2.COMMODEMLINE

関数																				
機 能	制御信号の入力状態を取得します。																			
書 式	<(戻り値)制御信号> = COMMODEMLINE(<①COM番号>)																			
戻り値	戻り値	<制御信号> 数値																		
	制御信号(DR, CI, CD, CS(I)) の現在の状態を返します。																			
	<table><tr><td>bit8</td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr><tr><td>DR</td><td>CI</td><td>CD</td><td>CS</td><td>—</td><td>—</td><td>RS</td><td>ER</td><td>—</td></tr></table>		bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	DR	CI	CD	CS	—	—	RS	ER	—
	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0											
	DR	CI	CD	CS	—	—	RS	ER	—											
<table><tr><th>信号名</th><th>内容</th></tr><tr><td>DR</td><td>DR信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>CI</td><td>CI信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>CD</td><td>CD信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>CS</td><td>CS信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>RS</td><td>RS信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr><tr><td>ER</td><td>ER信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)</td></tr></table>		信号名	内容	DR	DR信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)	CI	CI信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)	CD	CD信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)	CS	CS信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)	RS	RS信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)	ER	ER信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)					
信号名	内容																			
DR	DR信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
CI	CI信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
CD	CD信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
CS	CS信号の入力状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
RS	RS信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
ER	ER信号の設定状態を取得します。 1: ON(スペース) 0: OFF(マーク)																			
パラメータ	①	<COM番号> 数値																		
	「COMOPEN」で指定したCOM番号を指定します。																			
備 考	・ RS-485では、C信号はRS、I信号はCSにそれぞれ対応しています。 ・ RS-485では、DR, CD, CI, ERは使用しません。																			
使用例	「2.4 制御信号入出力機能」のプログラム例を参照してください。 ' 制御信号の状態を取得します STS = COMMODEMLINE(1)																			

3.5 詳細個別説明（ステータス）

1. COMSTATUS

関数																		
機能書式	シリアル通信の情報を取得します。																	
	<(戻り値)ステータス>= COMSTATUS(<①COM番号>, <②取得ID>)																	
戻り値	戻り値	<ステータス>																
	取得IDで指定した、シリアル通信の情報が得られます。																	
	<table><tr><th>取得ID</th><th>得られる内容</th></tr><tr><td>"SD_COUNT"</td><td>送信数</td></tr><tr><td>"RD_COUNT"</td><td>受信数</td></tr><tr><td>"FRAME_COUNT"</td><td>frameエラー数</td></tr><tr><td>"OVERRUN_COUNT"</td><td>overrunエラー数</td></tr><tr><td>"PARITY_COUNT"</td><td>parityエラー数</td></tr><tr><td>"BREAK_COUNT"</td><td>break数</td></tr><tr><td>"FIFO_SIZE"</td><td>FIFOサイズ</td></tr></table>		取得ID	得られる内容	"SD_COUNT"	送信数	"RD_COUNT"	受信数	"FRAME_COUNT"	frameエラー数	"OVERRUN_COUNT"	overrunエラー数	"PARITY_COUNT"	parityエラー数	"BREAK_COUNT"	break数	"FIFO_SIZE"	FIFOサイズ
	取得ID	得られる内容																
"SD_COUNT"	送信数																	
"RD_COUNT"	受信数																	
"FRAME_COUNT"	frameエラー数																	
"OVERRUN_COUNT"	overrunエラー数																	
"PARITY_COUNT"	parityエラー数																	
"BREAK_COUNT"	break数																	
"FIFO_SIZE"	FIFOサイズ																	
パラメータ	①	<COM番号>																
	「COMOPEN」で指定したCOM番号を指定します。																	
	②	<取得ID>																
	情報を取得したい種別を、文字列で指定します。																	
備考	<ul style="list-style-type: none">シリアル通信の情報取得には、ioctl の TIOCGICOUNT システムコール や TIOCGSERIAL システムコール を呼び出す事で、情報を得ています。"PARITY_COUNT"などの数値は、OS稼動中に クリアできません。 受信中に発生したか否かは、受信前後にステータスを読み取って、その差分で判断してください。																	
使用例	<pre>' 受信時に、パリティエラーが発生したか確認する事例です。 COMOPEN 1, "ttyS0", "BAUDRATE=9600, PARITY=ODD" PRE_CNT = COMSTATUS(1, "PARITY_COUNT") \$\$ = COMRECV\$(1, 10) ' 10バイト受信しようとしています ' シリアル通信のパリティエラー数を取得します POST_CNT = COMSTATUS(1, "PARITY_COUNT") IF PRE_CNT <> POST_CNT THEN PRINT "受信時にパリティエラーが発生しました"; PRE_CNT, POST_CNT END IF</pre>																	

3.6 内部エラーについて

エラーコードが01000012(=ドライバまたはライブラリの呼び出しに失敗した) の時、ERRSUB関数で ドライバ内部のエラーコード値を取得できます。

この値は、Linuxの標準シリアル通信で用いられる `errno` の値です。`errno`の詳細内容については、Linux関連の専門書籍、Web siteをご覧ください。

ERRSUB関数で値を受け取るには、ON ERROR CALL命令で、エラー発生時の割り込みルーチン内で、以下のよう記述します。

```
' エラーが発生したら、CB_ERR サブルーチンを呼び出すように定義します
ON ERROR CALL CB_ERR
ERROR ON

SUB CB_ERR(I_ERR, I_ERM$, I_ERL)
' ドライバまたはライブラリの呼び出しに失敗した
IF I_ERR = &H01000012 THEN
  NUM& = ERRSUB      ' ドライバ内部のエラーコード値を取得します
  PRINT "UART 内部のエラーコード値="; HEX$(NUM&)
END IF
END SUB
```

第4章 サンプルプログラム

サンプルプログラムは「/usr/share/interface/AJANPro/samples/IML/UART/」に格納されています。

AJAN統合開発環境を起動すると、左ペインのエクスプローラウィンドウ内の「Samples/IML/UART/」に、ファイルが取り込まれて配置されます。

4.1 サンプルプログラム一覧

No.	ファイル名	内 容
1	COMSEND.AJN	シリアル通信の送信を行うサンプルプログラムです。 COMOPEN命令で、デバイス名 ttyS0 に対して、オープンします。 その後、文字列を入力する都度、COMSEND命令で入力した文字列を送信します。 「Q」を入力すると、COMCLOSE命令でクローズして終了します。
2	COMRECV.AJN	シリアル通信の受信を行うサンプルプログラムです。 COMOPEN命令で、デバイス名 ttyS0 に対して、オープンします。 その後、COMRECFSIZE関数で、シリアルデータの受信サイズを確認し、受信データがあれば、COMRECV\$関数で受信します。 受信データの中で、「Q」を受信すると、COMCLOSE命令でクローズして終了します。
3	COMSENDRECV.AJN	2つのデバイスを開いて、片方から送信しつつ、もう片方から受信するサンプルプログラムです。 COMOPEN命令で、ttyS0 と ttyS1 の2つのデバイスに対してオープンします。 次に、ATTACH THREAD 命令で、新規にスレッドを生成して、COMRECFSIZE関数とCOMRECV\$関数で、シリアルデータの受信と受信したデータを表示し続けます。 メインスレッドでは、"SAMPLE" という文字列を COMSEND命令で送信します。これを10回繰り返して、プログラムを終了します。

第5章 索引

C	
COMCLOSE	14
COMMODEMLINE	17, 18
COMOPEN	13
COMRCV\$	16
COMRCVSIZE	16
COMSEND	15
COMSTATUS	19

第6章 重要な情報

保証の内容と制限

弊社は本ドキュメントに含まれるソースプログラムの実行が中断しないこと、またはその実行に誤りが無いことを保証していません。

本製品の品質や使用に起因する、性能に起因するいかなるリスクも使用者が負うものとします。

弊社はドキュメント内の情報の正確さに万全を期しています。万一、誤記または誤植などがあった場合、弊社は予告無く改訂する場合があります。ドキュメントまたはドキュメント内の情報に起因するいかなる損害に対しても弊社は責任を負いません。

ドキュメント内の図や表は説明のためであり、ユーザ個別の応用事例により変化する場合があります。

著作権、知的所有権

弊社は本製品に含まれるおよび本製品に対する権利や知的所有権を保持しています。

本製品はコンピュータ ソフトウェア、映像/音声(例えば図、文章、写真など)を含んでいます。

医療機器/器具への適用における注意

弊社の製品は人命に関わるような状況下で使用される機器に用いられる事を目的として設計、製造された物では有りません。

弊社の製品は人体の検査などに使用するに適する信頼性を確保する事を意図された部品や検査機器と共に設計された物では有りません。

医療機器、治療器具などの本製品の適用により、製品の故障、ユーザ、設計者の過失などにより、損傷/損害を引き起こす場合が有ります。

複製の禁止

弊社の許可なく、本ドキュメントの全て、または一部に関わらず、複製、改変などを行うことはできません。

責任の制限

弊社は、弊社または再販売者の予見の有無にかかわらず発生したいかなる特別損害、偶発的損害、間接的な損害、重大な損害について、責任を負いません。

本製品(ハードウェア、ソフトウェア)のシステム組み込み、使用、ならびに本製品から得られる結果に関する一切のリスクについては、本製品の使用者に帰属するものとします。

本製品に含まれる不都合、あるいは本製品の供給(納期遅延)、性能もしくは使用に起因する付随的損害もしくは間接的損害に対して、弊社に全面的に責がある場合でも、弊社はその製品に対する改良(有償サービスの利用)、代品交換までとし、製品の予防交換並びに、代金減額等、金銭面での賠償の責任は負わないものとします。

本製品は、日本国内仕様です。

商標/登録商標

本書に掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

改訂履歴

Ver.	年 月	改 訂 内 容
0.90	2019年10月	新規作成
1.00	2022年1月	最新情報に更新

このマニュアルは、製品の改良その他により将来予告なく改訂しますので、予めご了承ください。