

2. アナログ信号の計測

77

2.1 アナログ出力センサ

78

アナログ出力センサ

制御対象の状態によって電気的特性が変化する素子を使い、様々な制御量を電気信号（電圧、抵抗、電流など）に変換

加速度センサ	加速度	→	電圧
差動トランス	変位	→	電圧
ロードセル	力・圧力	→	電圧
ひずみゲージ	力・圧力	→	抵抗値
CdSセル	光	→	抵抗値
フォトダイオード	光	→	電圧
熱電対	温度	→	電圧
サーミスタ	温度	→	抵抗値
ポテンシオメータ	回転角	→	抵抗値

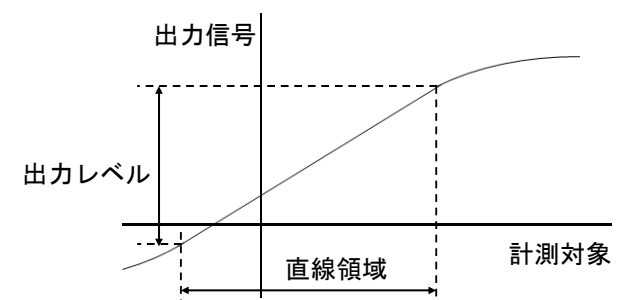
79

アナログ出力センサ

センサ素子から得られる信号は、必ずしも計測に適した信号とは限らない。

出力レベルの大きさ、線形性、ノイズ など

センサとして望ましいのは
出力レベルが広く、直線領域が大きいのが良い

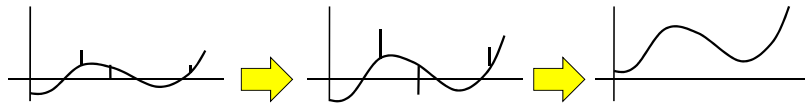


80

アナログ出力センサ

アナログ電気信号を別の電気信号に変換（電圧、電流）
信号処理用のアナログ信号として、電圧信号が使われる。

信号の大きさの調整：信号の定数倍
ノイズの除去：フィルタ
信号のレベルあわせ：信号の加減



これらの処理をアナログICなどを使ったアナログ回路で行う

81

2.2 アナログ信号の変換・増幅

アナログ信号の変換・増幅

アナログ電圧信号を別の信号に変換するのにアナログIC（オペアンプ）を使う。

オペアンプを使った回路

反転増幅回路：信号の正負を反転させて増幅
非反転増幅回路：信号の正負を反転させず増幅
差動増幅回路：二つの信号の差を増幅
加算回路：信号の加減（オフセット）

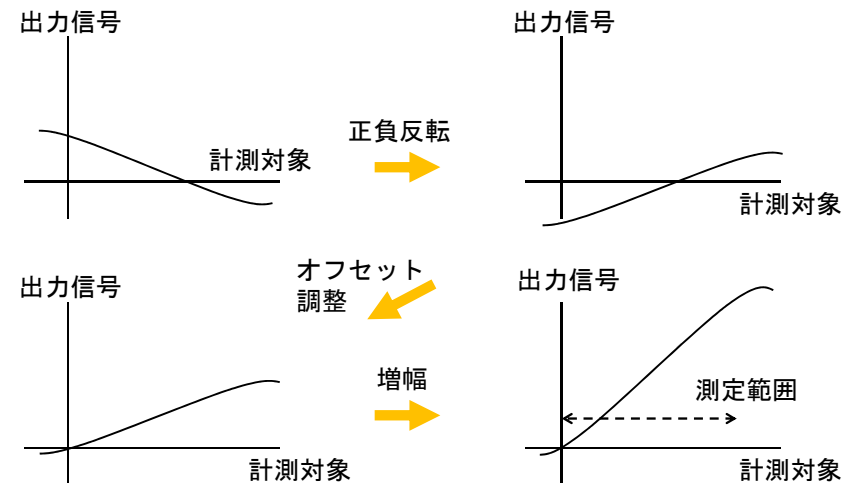
ローパスフィルタ：周波数の高いノイズの除去
ハイパスフィルタ：周波数の低いノイズの除去

オペアンプが使われる前はトランジスタや抵抗器、コンデンサ、コイルなどを使って信号変換を行っていた

83

アナログ信号の変換・増幅

アナログ信号の変換



84

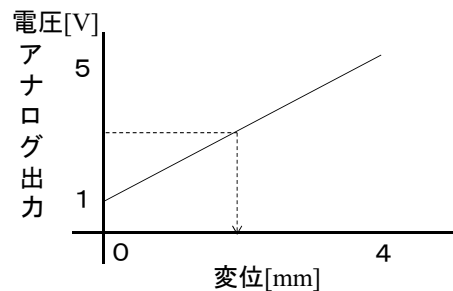
アナログ信号の変換・増幅

最終的に、様々なアナログ回路を使って標準化された信号に変換する。

電流信号 DC4~20mA
電圧信号 DC1~5V、DC0~5V、DC0~10V、
DC-5~5V、DC-10~10V など

測定対象と出力信号の関係をあらかじめ調べておく
(校正)

出力信号を測定することにより測定対象を知ることができる。



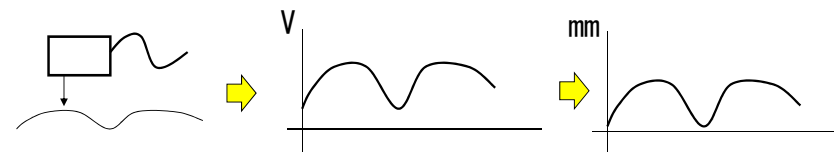
変位センサの例(電圧出力)

85

アナログ信号の変換・増幅

変位センサの例

電圧出力 DC1~5V → 測定範囲 0~4mm



変位センサ、液面センサ、圧力センサ、などプロセス制御などで使われる。

86

2.3 アナログ信号の計測

アナログ信号の計測

アナログ信号をそのまま測定する方法
アナログ電圧計、アナログ電流計、アナログテスタ

表示用であって、**自動化**ではほとんど使われない



87

88

アナログ信号の計測

アナログ信号は劣化が起きやすい
電気信号にはノイズがつきもの。
小さなノイズでも誤差の原因

アナログ信号のノイズ発生源

モータ……モータ等の出す火花ノイズ スパーク

出力機器……動作時の回路内の急激な電流低下

インバータ、電源装置
……スイッチング回路、高速のON/OFF

空中電磁波……空中から電線を通して

一度入ったノイズを取り除くのは困難

89

アナログ信号の計測

ノイズ対策として

シールドを施す : 金属ケース、導電樹脂ケース、

コンデンサの挿入 : 突入電流の除去

電源の変更 : スイッチング電源からリニアレギュ
レータや電池へ

電源フィルタの使用 : 電源装置からのノイズ侵入防止

ツイストペア線の使用 : 電源線からのノイズ侵入防止

信号媒体の変更 : 電源線からのノイズ侵入防止
光絶縁、光通信
(アナログ信号では困難)

90

3. デジタル信号の計測

アナログ信号の計測

アナログ信号での計測・制御は、
ノイズの影響を受けやすい
信号変換処理も容易ではない
あまり現実的ではない

検出部で物理量をアナログ電気信号に変換し、
変換されたアナログ信号をデジタル信号に変換し、(A/D変換)
デジタル信号で物理量を計測する。

調節部でデジタル信号化された物理量を入力し、制御信号をつく
り、出力する(デジタル制御)。

デジタル信号をアナログ信号に再度変換し、(D/A変換)
アナログ信号で操作部によって制御対象に働きかける。

* デジタル信号のまま制御対象に働きかける方法が一般的

91

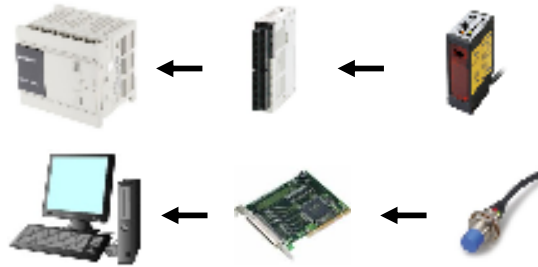
92

アナログ信号の計測

コントローラ（パソコン、PLCなど）にアナログ入力インターフェイスを接続し、コントローラで直接アナログ量を計測することが出来る。

*実際はA/D変換したデジタル信号を計測

この時のアナログ信号は、汎用的に使われる信号（電圧信号、電流信号）で、それに対応したインターフェイスを使用する。



93

3.1 デジタル信号への変換と計測

アナログ信号のデジタル信号への変換

アナログ信号をデジタル信号へ変換する方法として

1) nbitデータを使う方法

N個のON/OFFの組み合わせで連続量をあらわす

2) 1ビットデータを使う方法

周期（周波数）で連続量を表す

デューティ比で連続流を表す

パルス数で連続量を表す

3) 文字列を使う方法

汎用通信より送られる文字列で連続量を表す。

通信より送られる文字データから必要となる値を取り出し数値に変換する。

95

アナログ信号のデジタル信号への変換

1) nbitデータを使った計測

アナログ信号をnbitのデジタル信号に変換して計測する方法がある。

アナログ信号からデジタル信号への変換は、ほとんどが市販のA/D変換インターフェイスを用いる。

よって、ユーザは特に変換を行う必要はなく、変換後のデジタルデータの読み込みを行えばよい

市販のA/D変換インターフェイスが電氣的仕様が決められているので、センサの出力仕様や用途に合わせて選定する

94

96

デジタル信号の計測

nbitのON/OFFデータを入力し、各bitを 2^n 倍し、総和をとる。
そのデータを 2^n-1 で除して、入力レンジをかける

入力レンジ0~5、8bitデータの場合、

$$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ \times 128 & + 64 & + 32 & + 16 & + 8 & + 4 & + 2 & + 1 \\ \hline & & & & & & & = 173 \end{array}$$

$$\text{計測値} = 173 \times 5 / 255$$

入力されるデータは、使用するA/D変換インターフェイスやコントローラによって異なる。

パソコンの場合、nbitのデータを一つの変数に格納することも可能
PLCの場合、nbitのデータが一つのレジスタに格納される

97

デジタル信号の計測

入力レンジが符号付（正負）の場合

コンピュータの数値の扱いとして負の数は補数を使って表される。

パソコンの場合、変数を正負が使える符号付の型で宣言しておけば問題ないが、正の数しか使えない型やPLCで計測する場合、補数を意識していないと正しい測定ができなくなる。

98

デジタル信号の計測

インターフェイスの仕様によっては、正確な読み取りができない可能性があるため、仕様の概要については理解しておく必要がある

A/D変換の仕様として

入力信号、入力レンジ : DC1~5V、DC-10~10V、4~20mAなど
分解能 : 8(256)、10(1024)、12(4096)、16(65536)
サンプリング周期 : 4 μ s、10 μ s、20 μ s、
など

99

デジタル信号の計測

2) 1ビットデータを使った計測
アナログ信号を1bitのデジタル信号に変換して計測する方法がある。
その信号を使って、計測する量と信号が変化する時間の関係を使って計測する。
周期、デューティ比など

市販のデジタル入力インターフェイスを使い、1bitの信号の入力をおこなう。

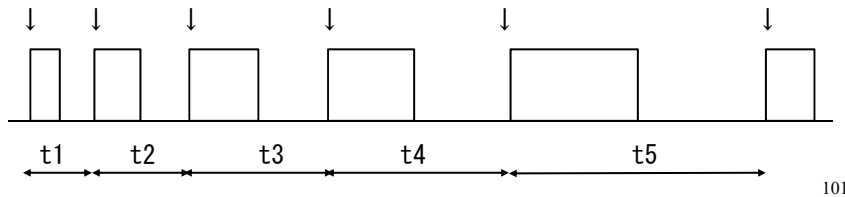
100

デジタル信号の計測

- a) 周期（周波数）で表されたアナログ量の計測
 あらかじめ、周期と測定対象の関係式を求めておく

周期 t と測定対象 s の関係 $s = kt$ (一般的に)

信号の立ち上がりを検出したらその時の時刻を記録し、次の立ち上がりを検出したらその時の時刻を記録する。その時刻の差（時間 = 周期）を算出し、周期と測定対象の関係式から測定対象の計測を行う。

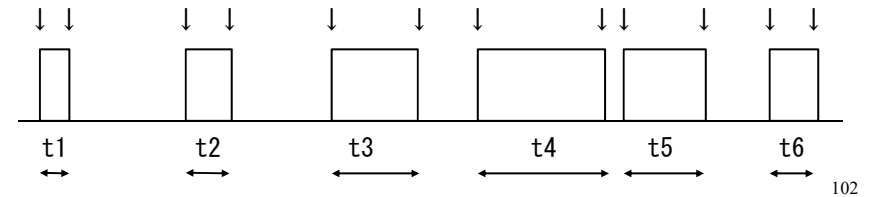


デジタル信号の計測

- b) デューティ比で表されたアナログ量の計測
 あらかじめ、出力パルスの周期を調べておく。また、ON時間（デューティ比）と測定対象の関係式を求めておく。

ON時間 t と測定対象 s の関係 $s = kt$ (一般的に)

信号の立ち上がりを検出したらその時の時刻を記録し、次に立ち下がりを検出したらその時の時刻を記録する。その時刻の差（時間 = ON時間）を算出し、ON時間と測定対象の関係式から測定対象の計測を行う。

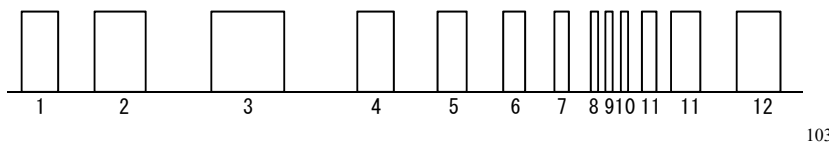


デジタル信号の計測

- c) パルス数で連続量を表す
 インクリメンタル式ロータリエンコーダでの回転角の計測で使われる。あらかじめエンコーダ1回転当たりのパルス数を調べ、パルス数と回転角の関係式を求めておく。

パルス数 n と回転角 θ の関係 $\theta = \delta n$ (δ : 分解能)

信号が立ち上がった回数を計数し、関係式から回転角を算出する。

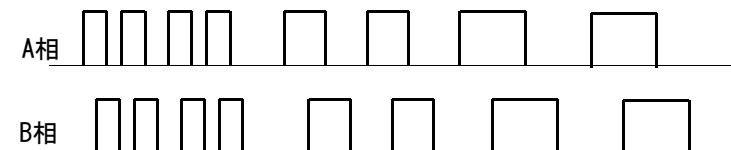


デジタル信号の計測

- パルス数により回転角度を計測
 360pls/revのロータリエンコーダにより速度を計測
 360度（1回転）で出力されるパルス数は360パルス
 1パルスあたり1度（1deg/PLS）
 パルス数が134plsであれば134deg回転

- パルス数により回転角度を計測
 パルスの周波数により速度を計測

位相が $\pi/2$ ずれたパルスを使うと回転方向が判別できる



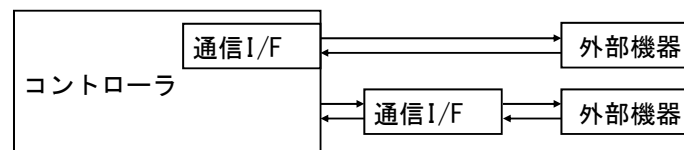
3.2 デジタル通信による計測

デジタル通信による計測

パソコンやPLCなどのコントローラと計測器などの外部機器との間でデータの受け渡しを行う方法として、規格化された汎用のデジタルデータ通信を用いる方法がある。

データ通信では、コントローラに通信インターフェイスを装着し、このインターフェイスを介して通信を行う。

コントローラによってはインターフェイスが内蔵されているものもある。インターフェイスがない場合もしくは足りない場合は外付けのインターフェイスを接続して通信を行う。



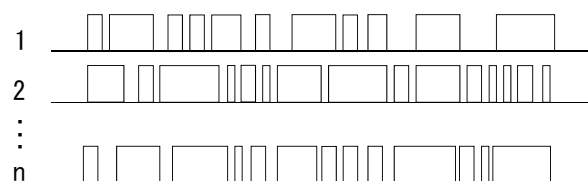
シリアル通信とパラレル通信

n bitデジタル信号の通信方法として、シリアル通信とデジタル通信がある。

シリアル通信：1本の線を使い、 n bitのデータを0ビット目から順番に送る。



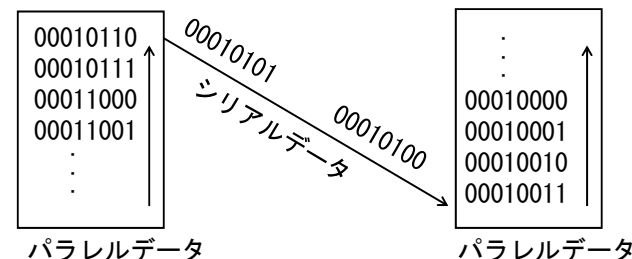
パラレル通信： n 本の線を使い、 n bitのデータを同時に送る。GP-IB、SCSI



シリアル通信

1本の線を使い、 n bitのデータを0ビット目から順番に送信もしくは受信する。

RS-232CやRS-422、RS-485、USB、イーサネットなど、現在の汎用通信のほとんどはシリアル通信である。

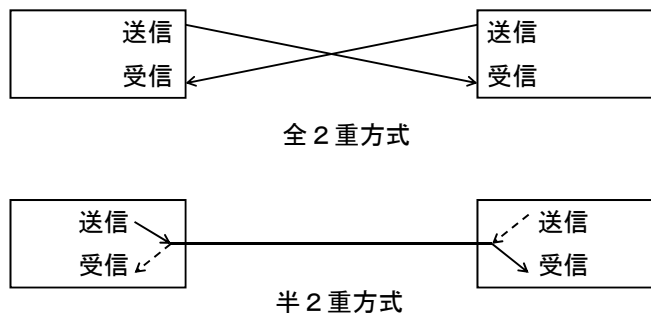


パラレルデータからシリアルデータへの変換、およびシリアルデータからパラレルデータへの変換が必要になる。

シリアル通信

通信方式として、2本の線を使って1本を送信用、もう1本を受信用とする全2重方式と、1本の線を使ってコントローラ内で送受信を切り替えながら通信する半2重方式がある。

現在は、ほとんどが全2重方式を使って通信するが、通信線に制限がある場合は半2重方式を使うこともある。



109

RS-232Cインタフェース

RS-232Cは汎用通信規格の一つ。

シリアル通信で、様々な通信方法（プロトコル）がある。基本的に、1本の送信線、1本の受信線で、データの送受信を行う（全二重方式）。小規模な通信（パソコンとテストなど）で使用

以前のパソコンでは標準で用意されていたが、最近のパソコンには装備されていないものがほとんど。この場合、USBなどからRS-232Cに変換するインターフェイスを使う。

1対1の接続で、1つのインターフェイスに対し1つの機器を接続。複数の機器を接続する場合、複数のインターフェイスが必要になる。

110

RS-232Cインタフェース

一般的に使われている通信プロトコルでは、送受信するデータは文字データ（ASCIIコード）が使われる。送受信データ作成時に、必要に応じて文字データの加工が行われる。

文字列の結合、文字列の取り出し、文字から数字への変換など

通信プロトコル（プログラム）を自作すれば、bitデータの送受信なども行うことができる。

111

USBインタフェース

もともとは通信規格ではないが、現在は通信規格と同等とみなされている。

パソコンやマイコンに標準的に用意されている。

基本的に、RS-232Cと同じように、文字列の入出力を行う

112

パラレル通信

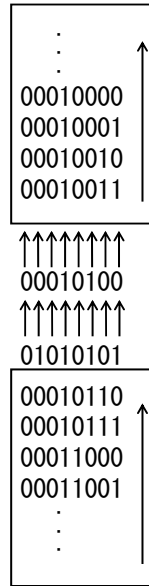
n本の線を使い、nbitのデータを同時に送る。
GP-IB、SCSIなど

メリット：通信速度が速い、
回路が簡単（シリアルデータに変換
する必要なし）

デメリット：線を長くすることが出来ない。

n本の線を使い
nbitの平行データを
そのまま送受信

シリアル通信の技術が向上したため、
パラレル通信の利点がなくなってきている。
現在はパソコン内部のバスで使われている程度。



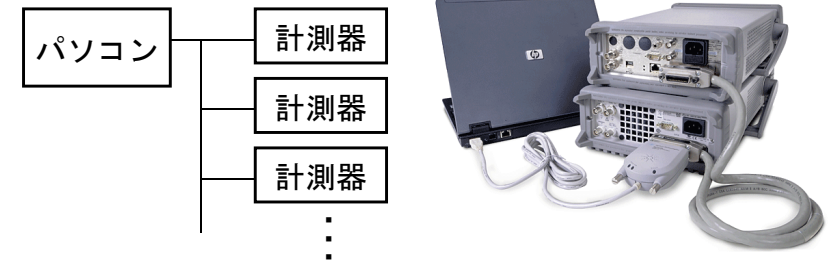
113

GP-IBインタフェース

GP-IB（HP-IBともいう）は汎用通信規格の一つ。
パソコンと高機能な計測器（デジタルオシロスコープなど）
の通信に用いる。

8bitの平行通信で、通信速度が速い（データ量が多い）。

1対nの通信が可能



114

GP-IBインタフェース

他のインターフェイスが進歩したため、
最近はあまり使われなくなってきている。

RS-232Cによるシリアル通信

シリアル通信の通信規格として、RS-232Cがある。
昔は主流であったが、最近ではUSBに変わりつつある
両方とも、基本的な考え方は同じ。
RS-232Cは自動計測、自動制御の分野で今でも使われている。

シリアル通信 ≡ USB通信 or RS-232C通信

RS-232C通信を行うためには、コントローラにRS-232Cイン
ターフェイスが付いていることが必要である。

最近のパソコンにはRS-232Cインターフェイスが
付いていないものが多い。その場合は外付け
RS-232Cインターフェイス（例えば、USB-RS232C
変換ケーブル）をパソコンに接続する。

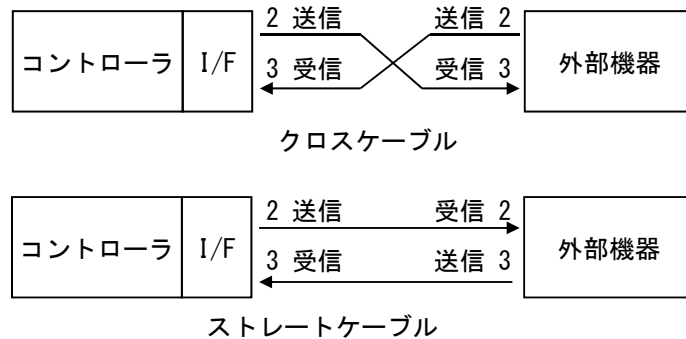


115

116

RS-232Cケーブル

標準規格として、接続ケーブルには、D-sub9ピンケーブルが使われる。（昔はD-sub25ピン）
コネクタのピン配置によって、クロスケーブルとストレートケーブルがある。通常は受信側、送信側でピン配置が共通にされているため、通信にはクロスケーブルを使う。



117

RS-232Cケーブル

ケーブルのコネクタにはオス（プラグ、ピン）、メス（ジャック、ソケット）がある。一般的にコントローラや外部機器にはオスコネクタが使われていることから、接続用ケーブルはメス-メスのクロスケーブルが、延長用ケーブルはオス-メスのストレートケーブルが使われる。
（機器によっては例外もある）



接続用クロスケーブル



延長用ストレートケーブル

118

通信ポート

コントローラと外部機器と通信する際のデータ送受信の接続口をポートという。

パソコンではCOMポート、PLCでは通信ポートと呼ばれ、一般的には複数のポートを持っている。

各ポートにはポート番号が割り振られ、コントローラが外部機器と通信するときは、ポート番号を指定してデータの送受信を行う。

パソコンの場合、RS-232CはCOMポートとして認識される。パソコンのデバイスマネージャーで認識されているかどうか、またそのポート番号を確認することが出来る。

119

通信パラメータ

データ送受信を行う時は、機器間で通信パラメータ（取り決め）を合わせる必要がある。通信パラメータがあていないと、正しいデータを送ることはできない。

通信パラメータとして以下のものがある。

通信速度 [bps]	1200、2400、4800、9600、19200・・・
パリティチェック	偶数、奇数、無し
データビット長	7ビット、8ビット
ストップビット長	1ビット、1.5ビット、2ビット
フロー制御	Xon/Xoff制御、ハードウェア制御（RS/CS制御）、なし

外部機器の通信パラメータは固定されていることが多いので、コントローラ側で外部機器に合わせて通信パラメータを設定する。

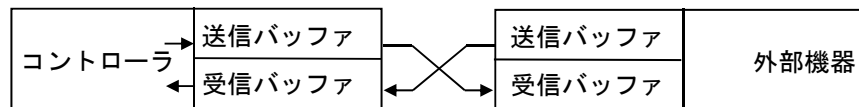
120

バッファメモリー

インターフェイスにはデータを一時的に格納するバッファメモリー（バッファ）が用意されている。送信側、受信側それぞれに送信バッファ、受信バッファがある。

コントローラから送信するときは、ソフトウェアで送信バッファにデータを格納する。すると、自動的に相手側の受信バッファへデータが送信される。

逆に相手側から送信されたデータは、自動的にコントローラの受信バッファに格納されるので、ソフトウェアでそのデータを読み出す。



送受信バッファ

121

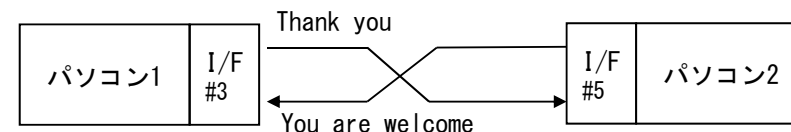
RS-232Cによるシリアル通信(パソコン-パソコン)

RS-232Cによるデータ通信の方法として、一般的には送信命令（「WT」など）を使いデータを送信し、受信命令（「RD」など）を使ってデータを受信する

例 WT (PortNO, DATA) 指定したポート番号から文字列を送信
a=RD (PortNO) 指定したポート番号から送信した文字列を変数aに格納

パソコン1のプログラム
WT (3, "Thank you")
s = RD (3)

パソコン2のプログラム
a = RD (5)
WT (5, "You are welcome")



122

RS-232Cによる自動計測(パソコン-計測器)

RS-232Cによる自動計測では、一般的にはパソコン（コントローラ）が通信権を持つ。計測器はパソコンからのデータを受信し、命令に対し応答（測定データ出力など）を行う。

パソコンのプログラム

WT (3, "Give me") (データ要求コマンド)
a = RD (3) (文字データ "DCV235_R3" を受信)
Dat = Val (Mid (a\$, 4, 3)) (必要なデータを取り出す)



123

区切り文字(デリミタ)

受信バッファで連続で複数のデータを受信したのちにデータを取り込むと、全てのデータを一つの変数に取り込んでしまう。

例えば、"abc"、"xyz"、"232C"、"COM"・・・を連続で受信（送信）し、それを変数aに一度に取り込むと a="abcxyz232CCOM・・・" となる。

このようなトラブルを防ぐために、送信データの最後に区切り文字を入れて送信し、受信時には区切り文字までを取り込むようにする。

例えば、区切り文字を#とし、"abc#"、"xyz#"、"232C#"、"COM#"・・・を連続で受信（送信）し、それを変数aに区切り文字までを取り込むと a="abc"、a="xyz"、a="232C"、a="COM"・・・ となる。

124

区切り文字(デリミタ)

このような区切り文字をデリミタまたはターミネータという。
送信側、受信側でデリミタの有無および使用する文字を決めておくと、送信されたデータを正確に受け取ることができる。

デリミタとして、一般的に制御文字が使われる。
良く使われる制御文字としてCR (Chr (13))、LF (Chr (10))、HT (Chr (9))などがある。(アスキーコード参照)

区切り文字の他にも、データ文字列の先頭にヘッダを、末尾にフッタをつけることもある。

ヘッダ	データ	フッタ	デリミタ
-----	-----	-----	------

データフォーマット例