

仙台市/仙台市産業振興事業団

ロボット博士の基礎からのメカトロニクスセミナー

C11/Rev 1.0

第11回

マイコン・パソコン通信 の基礎

仙台市地域連携フェロー

熊谷 正 朗

kumagai@tjcc.tohoku-gakuin.ac.jp

東北学院大学工学部
ロボット開発工学研究室

RDE

今回の目的

○ コンピュータ間の通信技術

テーマ1：通信の基礎

- 通信の使いどころ
- 通信の基礎知識：ハードとソフト

テーマ2：おもな通信方式の概要

- SPI, I2C, 調歩同期(UART), CAN
- USB, ネット, XBee, TCP/IP

テーマ3：独自通信の実装例

- ロボット用通信 SerialLoop

コンピュータ間通信

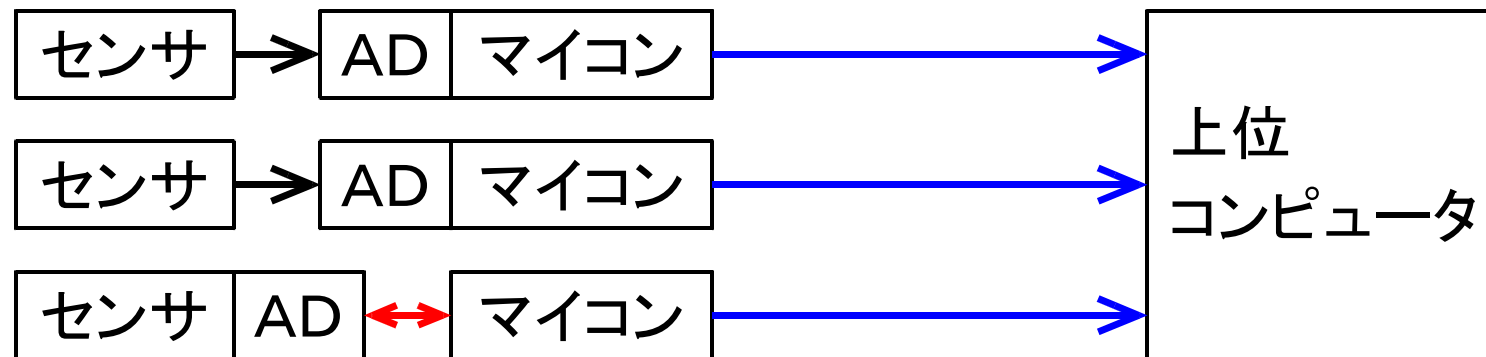
○ 通信の必要性

◇センサなどの信号伝達

- ・センサ信号を測定現場で処理

※AD変換、信号の下処理、判断等

- ・処理結果の伝達

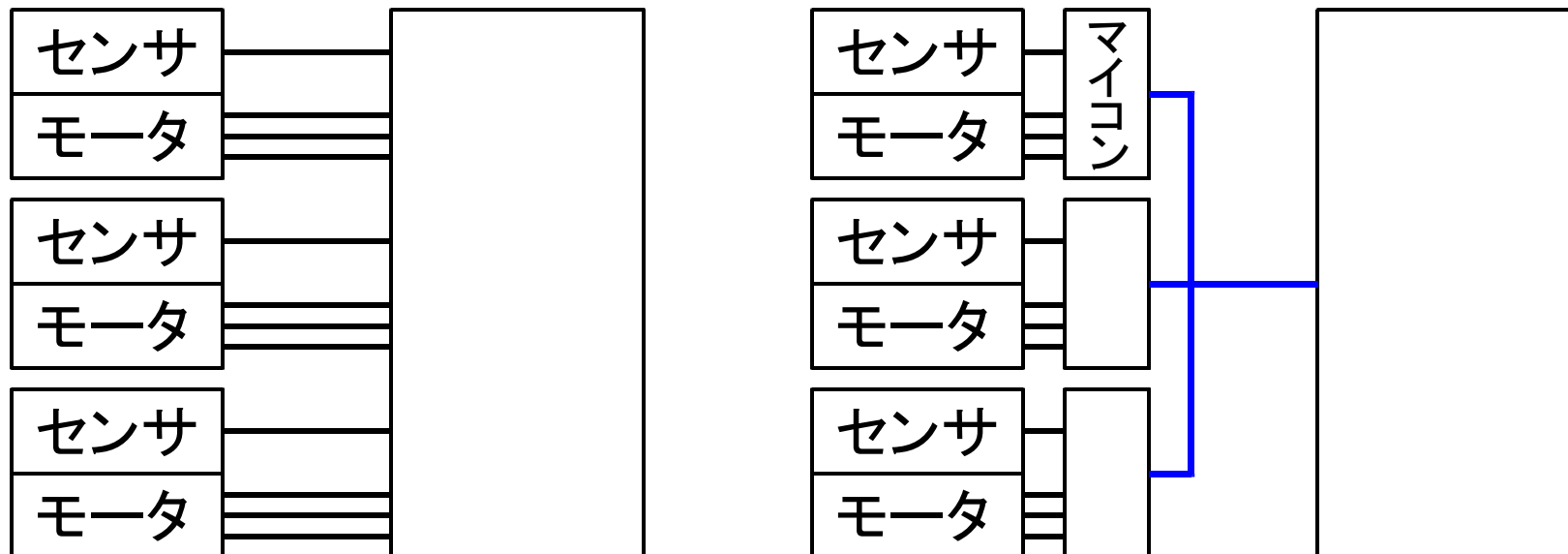


コンピュータ間通信

○ 通信の必要性

◇処理の分散 VS 中央集中制御

- ・大規模システムでは配線が問題に。
- ・局所ごとの制御系に通信と電源だけ配線。



コンピュータ間通信

○ 通信の基本要素

◇ハードウェア（物理層）

- ・電気信号(or光)として送受信する線、回路
- ・バイトと信号を相互変換する入出力回路

◇ソフトウェア

- ・バイト列をハードとやりとり
- ・データからバイト列を組み立てる／
バイト列からデータを復元する
- ・送受データの取り決め（プロトコル）

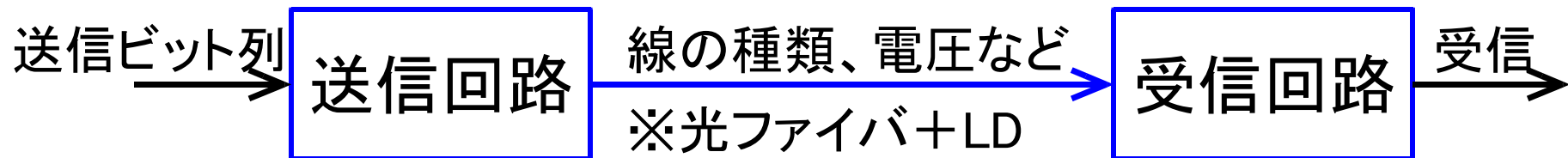
通信のハードウェア

○ 信号の伝送経路

◇ 線と送受信回路

- ・ ただの線 / インピーダンスの決まった線
- ・ 信号レベル (=0/1の表記)

例) ロジック信号(5V, 3.3V等)そのまま、
高めの電圧に(RS232C=±数V等)、
低めの差動電圧に(RS485, CAN, LVDS等)

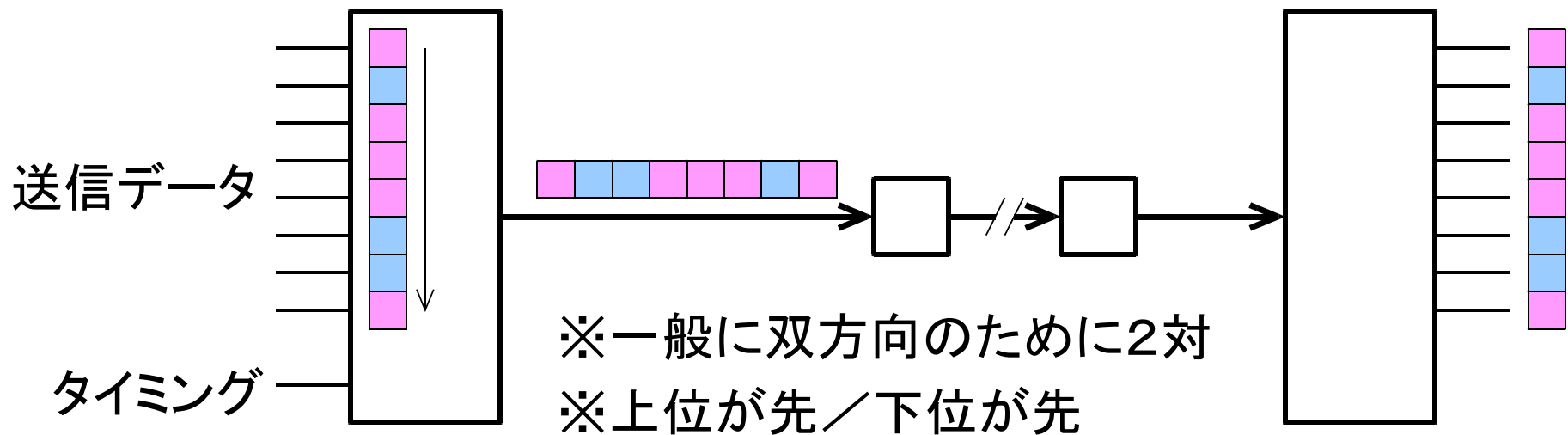


通信のハードウェア

○ バイトと信号の変換

◇UART, シリアライザ/デシリアライザ

- ・ 信号の直列化 / ビット列から復元
- ・ 一般に専用回路を使うがソフトで実装も。

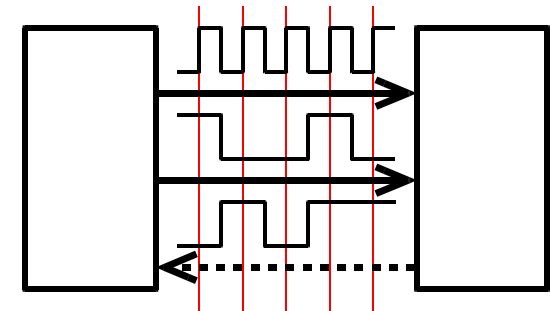


通信のハードウェア

○ 信号のタイミング

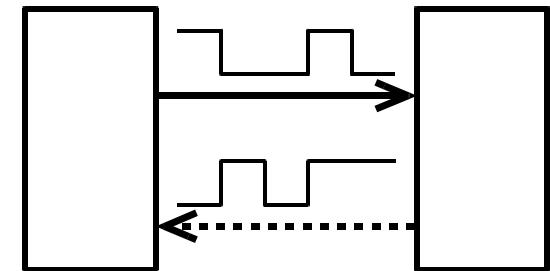
◇ データ線 + クロック線

- ・ 信号のタイミングを専用線(CLK)で規定。
- ・ 通信速度は自由。



◇ データの線のみ

- ・ 送受で速度・タイミングをあわせる。
- ・ データにクロックを埋める。

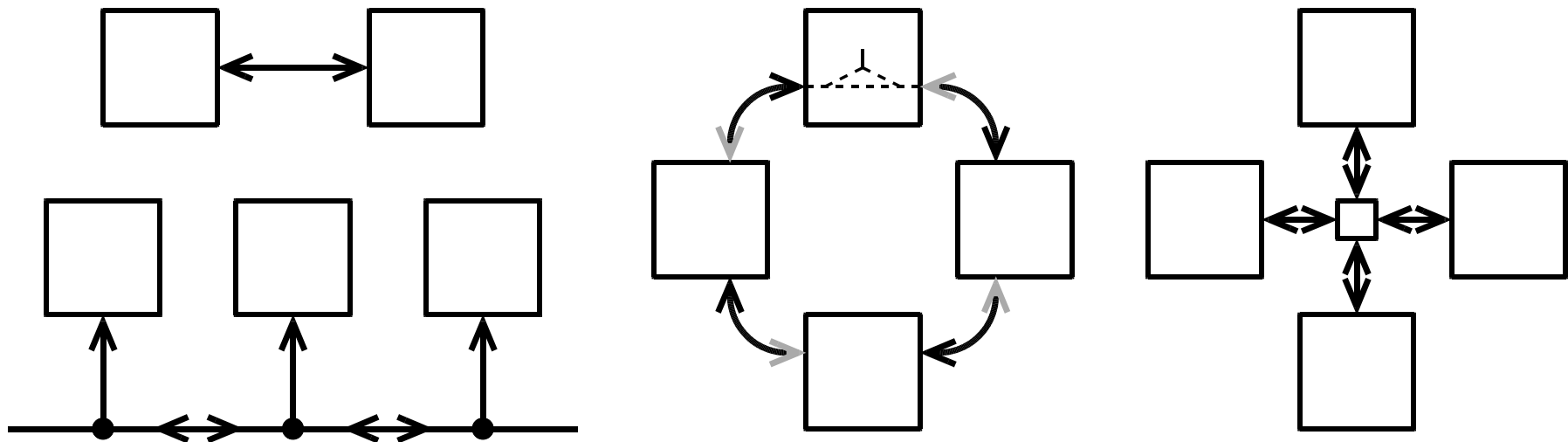


通信のハードウェア

○ トポロジー

◇ 接続の形態

- ・ 1対1 / バス / リング
(/ スター / メッシュ)

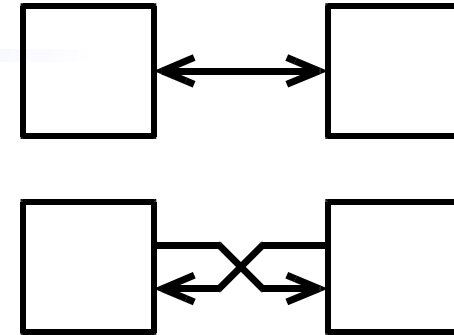


通信のハードウェア

○ トポロジー

◇ 1対1接続

- ・ 送受とも専用の入出力端子を一般に持つ。
- ・ 対等ならクロスで接続する。
マイコン～デバイスなどの接続では、
最初から役割明記 (MISO: Master In Slave Out)
- ・ 相手を特定する必要なし。シンプル。
- ・ RS232C, いまのネット、USB、など多数

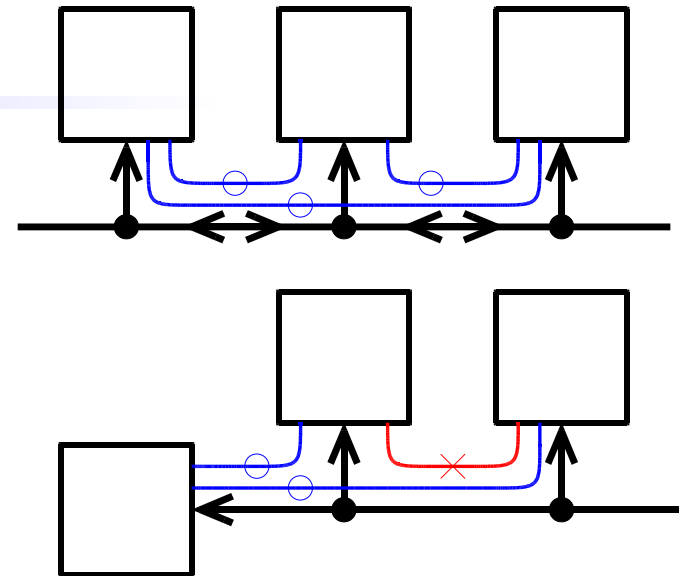


通信のハードウェア

○ トポロジー

◇バス接続

- ・ 通信用の信号を3個以上で**共用**する。
- ・ 多数を**省配線**で接続可能。
- ・ 全部対等 / 1個が全体を制御。
- ・ 通信の**衝突**に対する対策が必要(検出・再送)
- ・ I2C, CAN、昔のネット(10BASE2など)

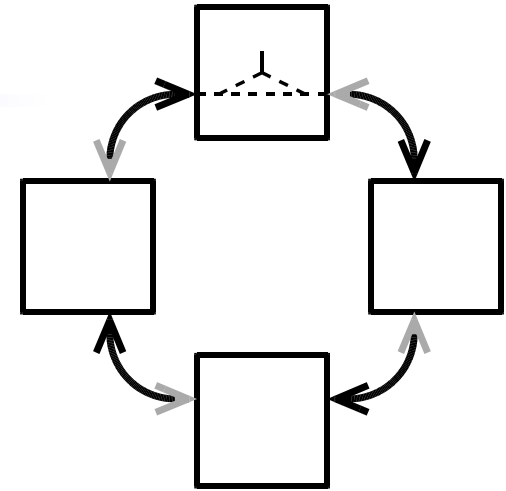


通信のハードウェア

○ トポロジー

◇リング(バス)接続

- ・環状に接続する。
- ・電氣的には1対1接続。
- ・回路的に送受/バイパスできるもの、一度受信してソフト的に転送するもの。
- ・1方向、双方向
- ・切断時に脆弱も、衝突無し、応答性など良。
- ・CELL(PS3のMPU)のプロセッサ間接続など。

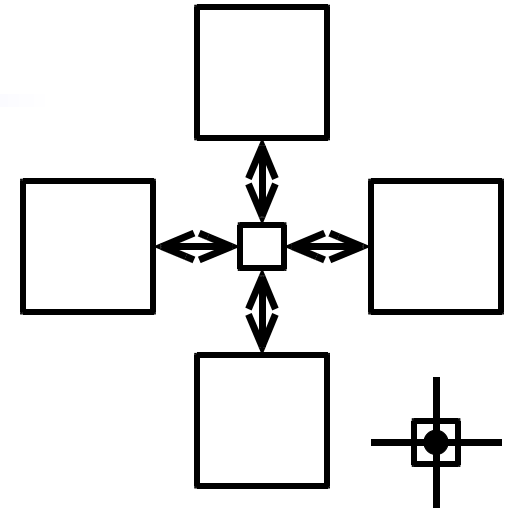


通信のハードウェア

○ トポロジー

◇ スター接続

- ・ 枝分かれ状に接続する。
- ・ 中央の接続点＝ハブが、
ただの電氣的並列接続 → バス型
一度情報に戻す場合 → 1対1型
の一種と見られる。
- ・ ネットの接続(100BASE-TX)など。
※対等ではないUSBはツリー型とされる。

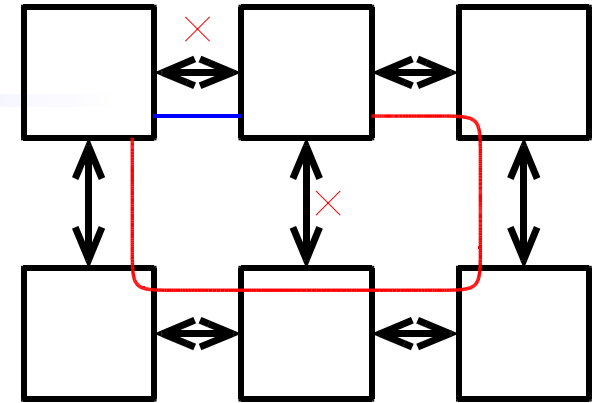


通信のハードウェア

○ トポロジー

◇ メッシュ

- ・ (主に) 1 対 1 の接続を相互に行って網目状に経路を構成する。
- ・ 一部が通信切断しても、別経路が存在。
- ・ 「どの経路を通すか」= ルーティング必要。
- ・ インターネット、Zigbee など



通信のソフトウェア

○ バイト列をハードとやりとり

◇OSありの場合: OSのデバイスドライバが担当

◇OSなしマイコンの場合:

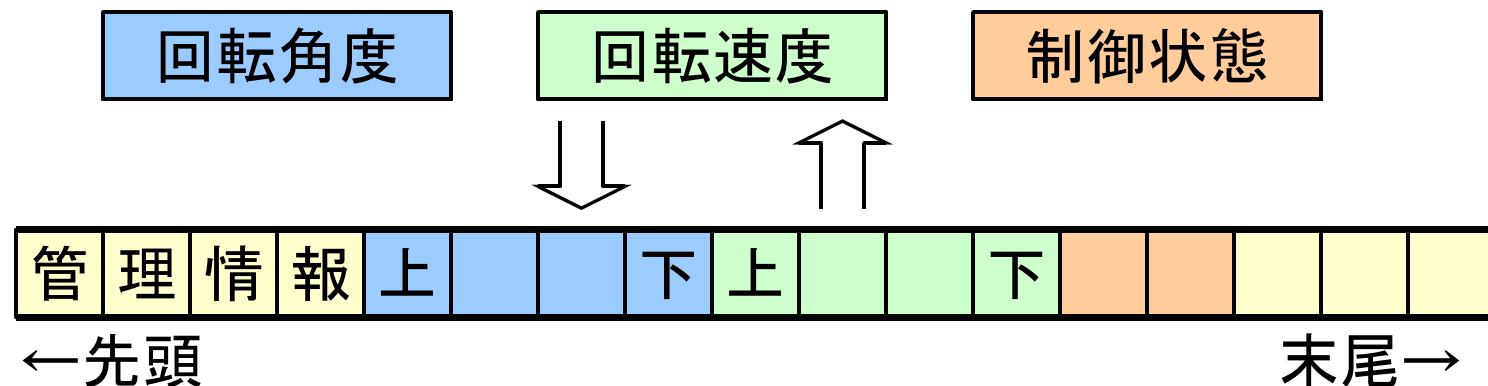
- ・ 特定のアドレスに1バイト書く→送信
- ・ 受信→特定のアドレスから読める
- ・ 送信完了、受信完了のフラグ
- ・ 送信完了、受信完了の割り込み
→ 継続するバイトの送信、受信処理

通信のソフトウェア

○ データとバイト列の変換

◇データのパックと送信順序

- ・多くの通信は本質的には1ビット単位、大抵はバイト単位にまとめて取り扱い。
- ・情報をバイト列に並べる、取り出す。
- ・上位と下位の位置に注意(エンディアン)



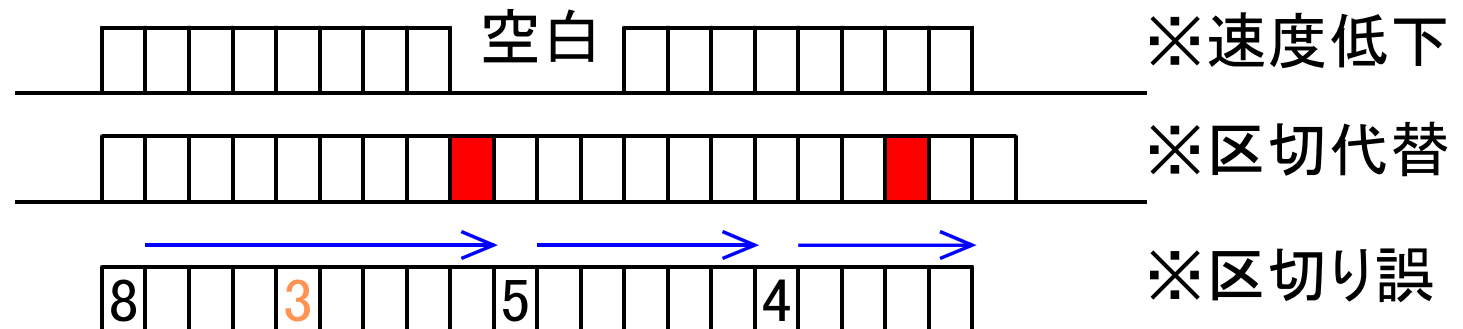
通信のソフトウェア

○ データとバイト列の変換

◇データの区切りの明確化

- ・主に受け側での復元処理のため。
- ・1: 明確な特定の信号状態、バイト値
例) 信号オフ、0は区切り、改行

2: バイト列に長さ情報を付加→終点

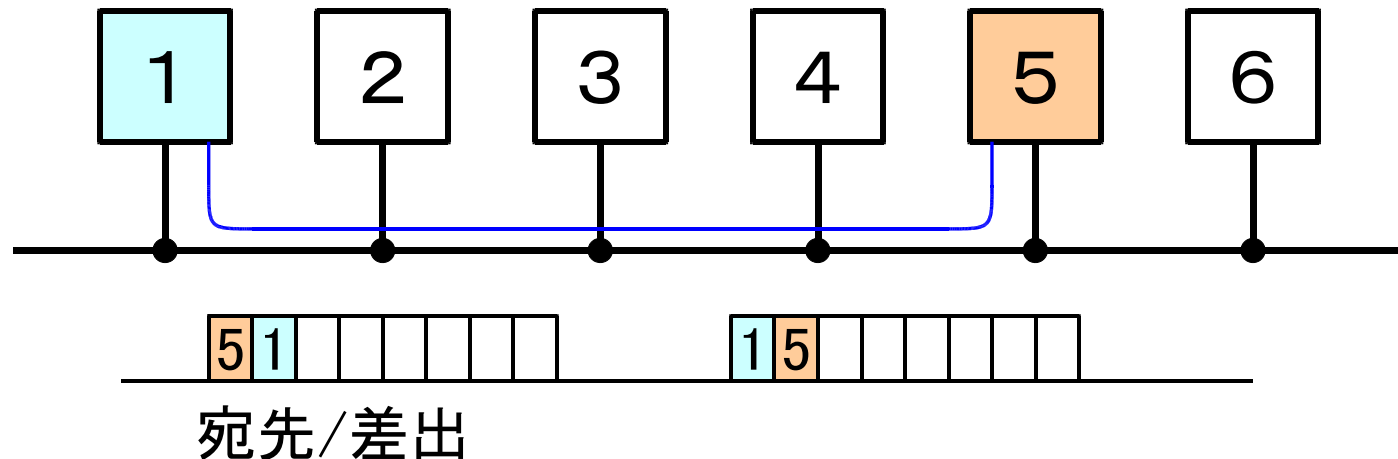


通信のソフトウェア

○ データとバイト列の変換

◇宛先(アドレス)の付加

- ・ 1対多、多対多の通信では宛先が必須。
- ・ 返答をもらうために自分のアドレスも含む。

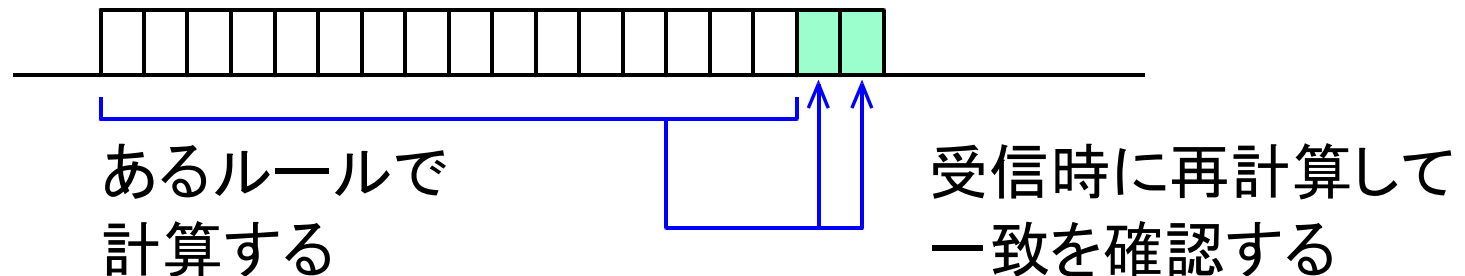


通信のソフトウェア

○ データとバイト列の変換

◇ バイト列の正しさの検出

- ・ ローレベルな通信では通信エラーは存在するとして、対策する必要がある。
- ・ エラー発生を検出→破棄、or→復元
- ・ パリティ / チェックサム / CRC



通信のソフトウェア

○ 送受データの取り決め（プロトコル）

◇送る内容の明確化

- ・複数種類のデータのやりとりをしたい。
→「なにを送るか」を送信に含める
※データ長などとあわせ、ヘッダと呼ばれる

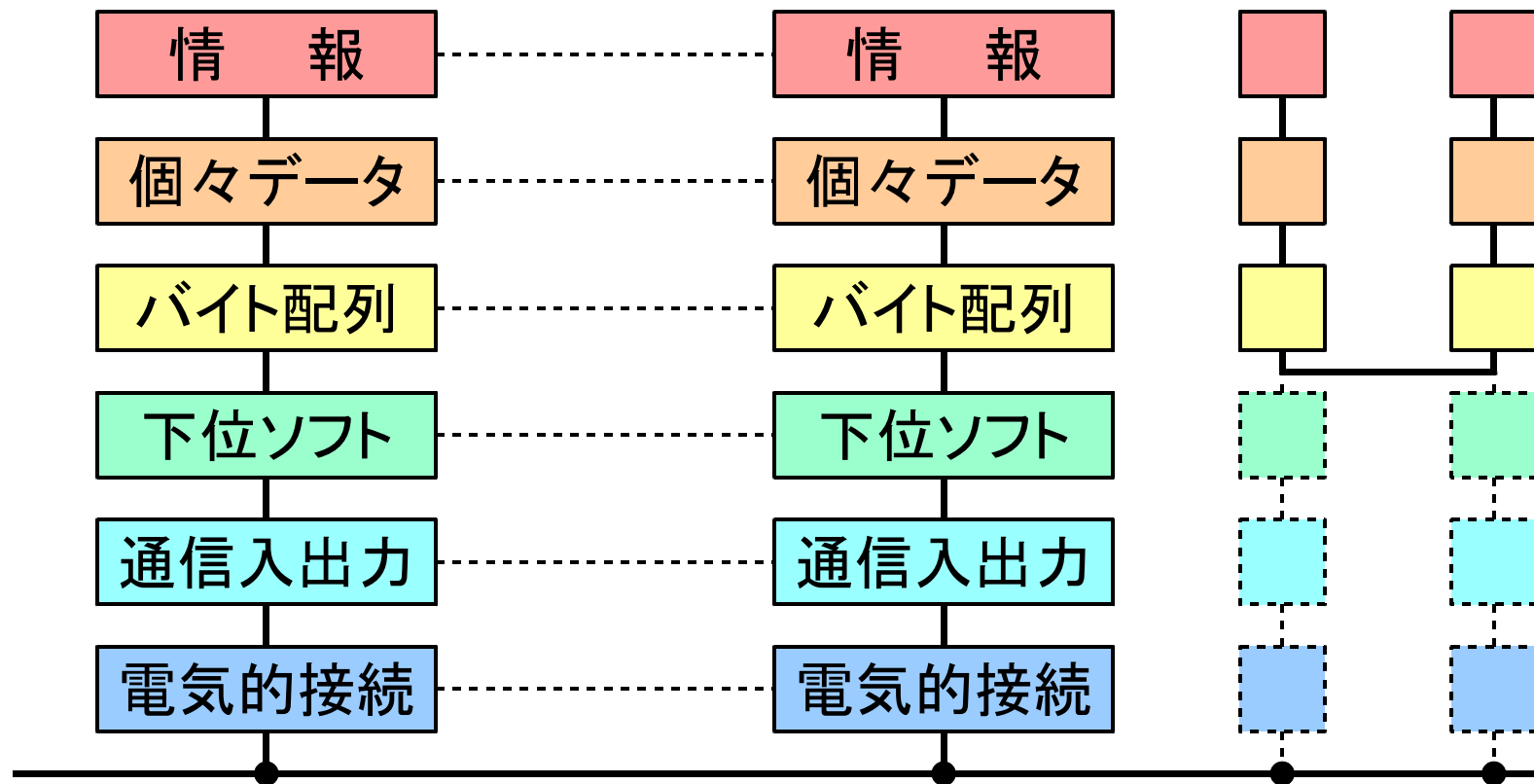
◇要求と応答

- ・情報の要求→応答のルール化
- ・情報受信の返答（ACK＝アクノリッジ）

コンピュータ間通信

○ 通信の実際

◇ハードとソフトの多層構造 / 下位の隠蔽



実際に用いられる通信

○ 概 要

◇デバイス(センサ、AD等)～マイコン間

- SPI型 (Microwire, 3線式、4線式、亜種多数)
- I2C

◇マイコン間

- 調歩同期 (シリアルポート、UART, RS232)
- I2C (ただし、対等ではない)
- CAN
- (Ethernet, USB)

実際に用いられる通信

○ 概 要

◇マイコン～パソコン(OS付き高性能マイコン)

- ・ 調歩同期 (シリアルポート、UART, RS232)
- ・ USB (ネイティブ、シリアルポート変換)
- ・ Ethernet (いわゆるネットワーク接続)

◇無線系

- ・ Bluetooth, Zigbee, Xbee, 特定小電力(2.4G)
- ・ 無線LAN
- ・ 携帯電話ネットワーク

実際に用いられる通信

○ 概 要

◇インターネットの通信

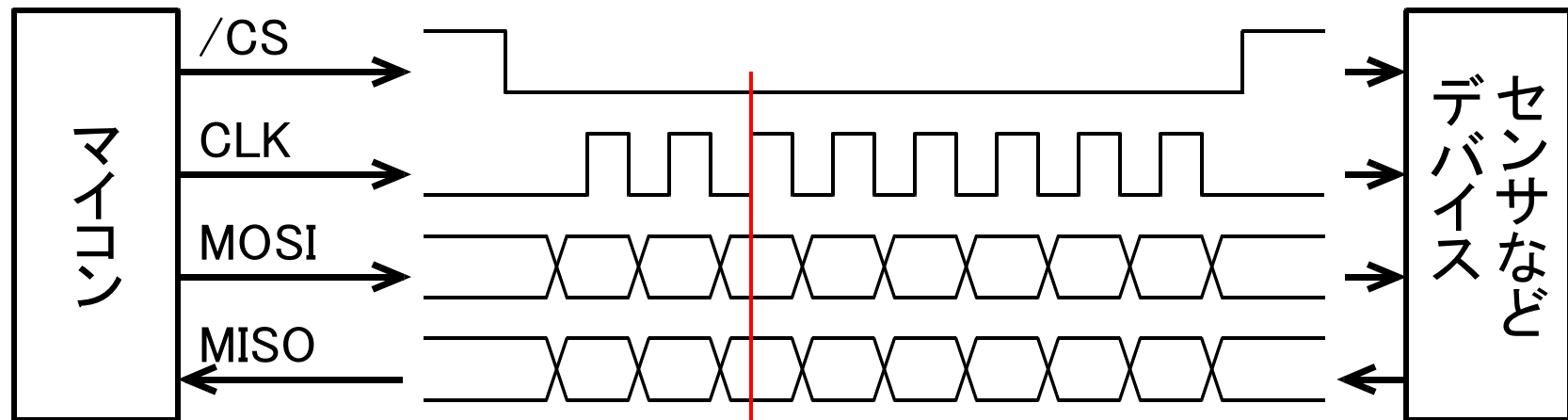
- TCP/IP、UDP/IP
- HTTP, SMTP, POP

デバイス～マイコン間の通信

○ SPI型

◇特徴

- ・ 1対1(1対多)
- ・ クロックあり、信号線の方法は専用。
- ・ センサやAD変換器の接続に多い。

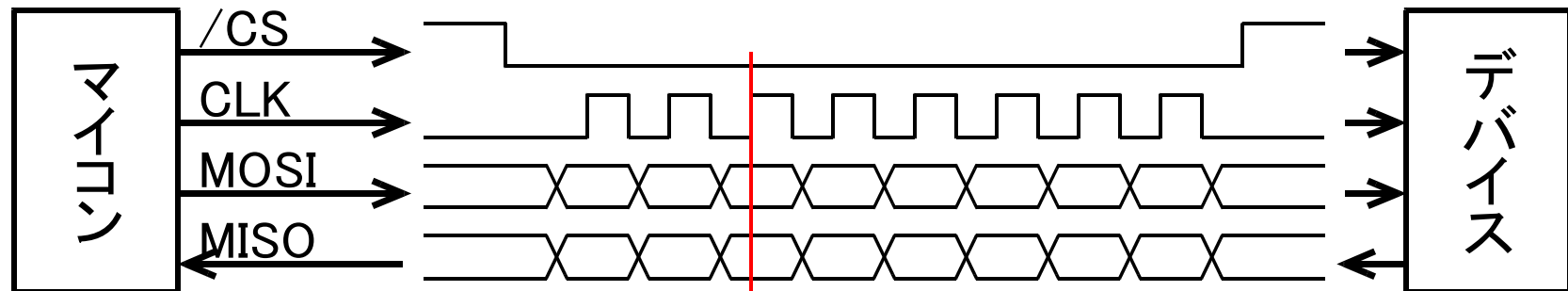


デバイス～マイコン間の通信

○ SPI型

◇使い方（専用IFは不要、ソフトでOK）

- ・ CSを下げる→MOSIに出力ビットを用意
→CLKを△▽→次のビット
- ・ CLK▽→MISOで読み→CLK△
- ・ CSを個別につなぐと複数デバイス可。



デバイス～マイコン間の通信

○ SPI型

◇補足メモ

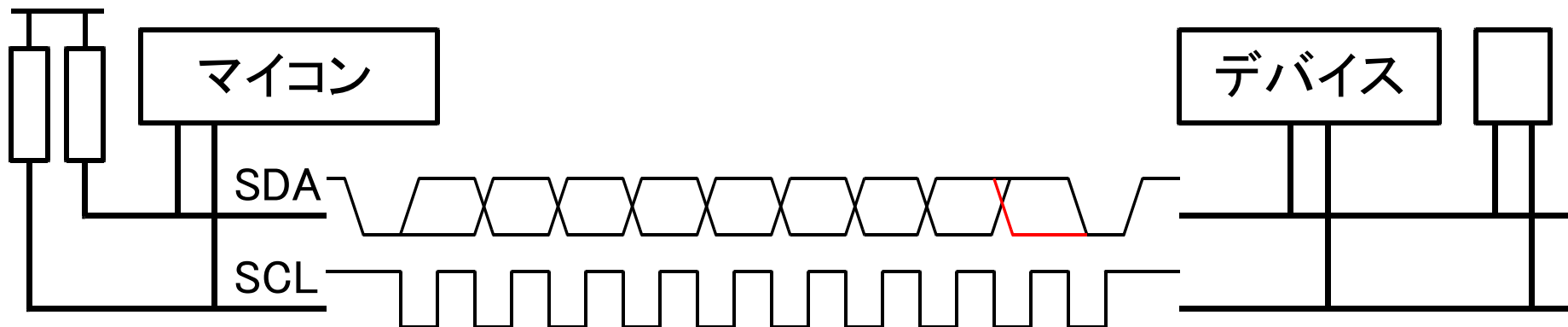
- ・専用IFを持つマイコンもあるが、その設定をあれこれするよりソフトが多分早い。
- ・似た形式のものが多いので、タイミングチャートを読み、扱えるようになると良い。
- ・最近のマイコンは速いので、CLKの上げ下げ速度に注意(デバイスの制限)。
- ・74**595, 597等シフトレジスタにも使える。

デバイス～マイコン間の通信

○ I²C

◇特徴

- ・ 1対多（多対多） **バス配線型**（＝双方向）
- ・ デバイスにアドレスがある。
- ・ **クロックあり**、データは上り下り共通。
- ・ 低～中速のセンサを複数つかうケース。

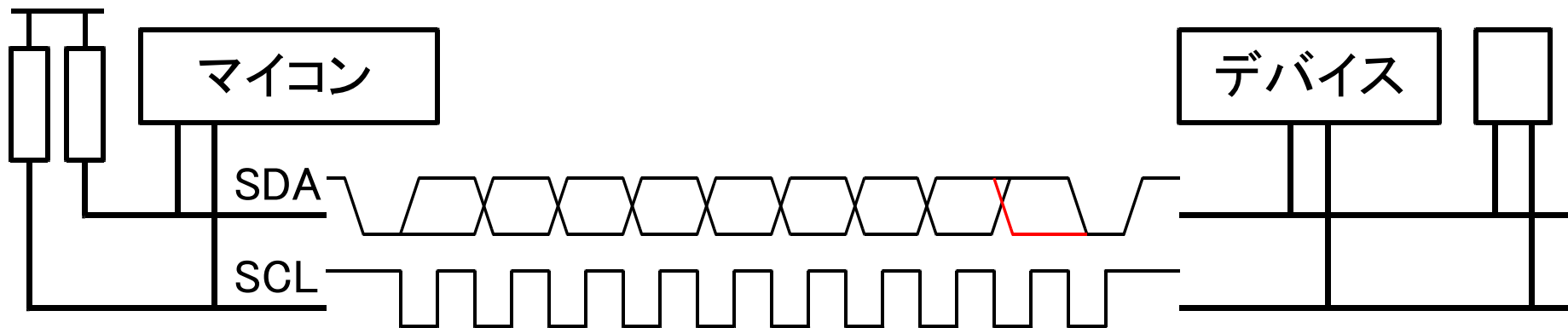


デバイス～マイコン間の通信

○ I²C

◇使い方

- ・ 専用IFを持つマイコンを採用する。
- ・ マイコンのIFを適切に設定する。
- ・ デバイスの説明書に従い、アドレスを指定、データを送受(さらに内部アドレスの指定が一般的)



デバイス～マイコン間の通信

○ I²C

◇補足メモ

- ・ SPI型に比べて、マイコン設定のため複雑。
- ・ デバイスの内部アドレスの指定→
デバイスからの応答受信
などの手続きが面倒（信号線共有のため）
- ・ トラブル発生時に、IFの設定が悪いのか
デバイスの使い方が悪いかの切分け面倒。
→確実に動く機能をデバイスで見つける

マイコン間の通信

○ 調歩同期式通信

◇特徴

- ・ 1本の線で、送信側から受信側にする。
- ・ **クロックなし**(速度は送受側の双方で一致させる)

◇バリエーション

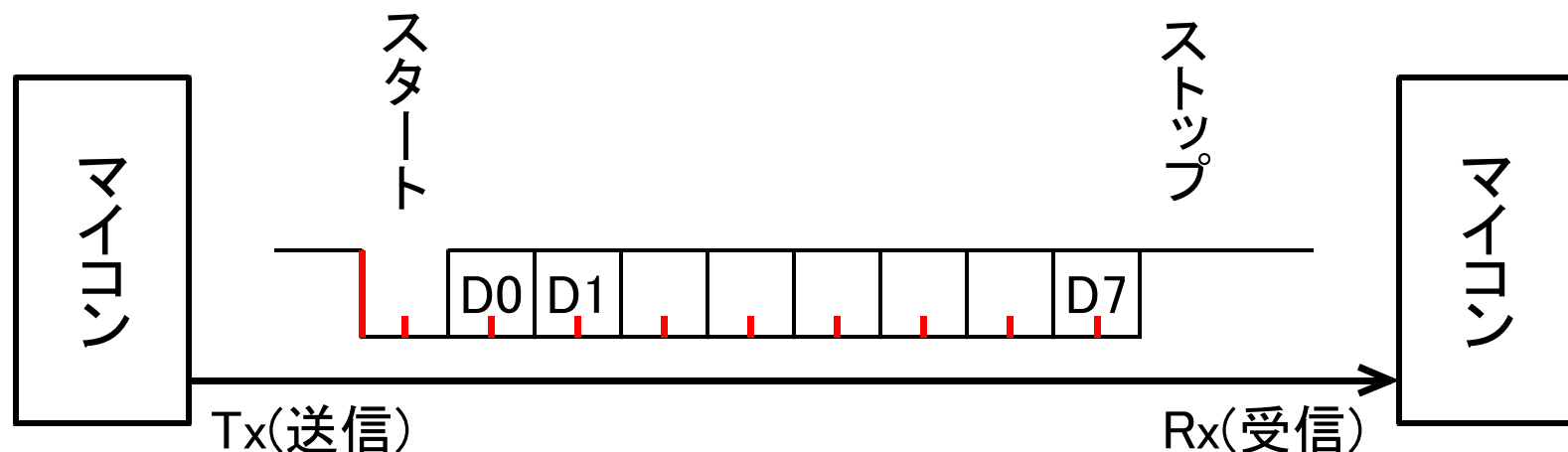
- ・ いわゆるシリアルポート、RS-232C(D,E)
1対1、信号電圧±数V、もしくは0-5, 0-3.3
- ・ RS422 1対1、高速、差動信号
- ・ RS485 1対多、多対多に対応した422

マイコン間の通信

○ 調歩同期式通信

◇通信の原理

- ・ 1ビットあたりの速度を送受で合わせる。
- ・ **スタートビット**を検出して、データの取り込みタイミングを決定する。

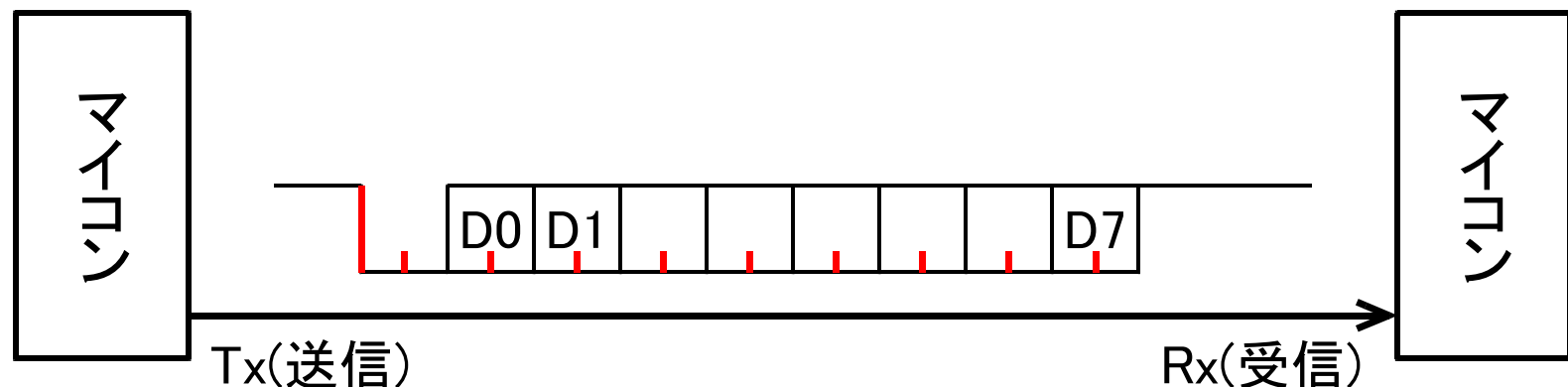


マイコン間の通信

○ 調歩同期式通信

◇特徴

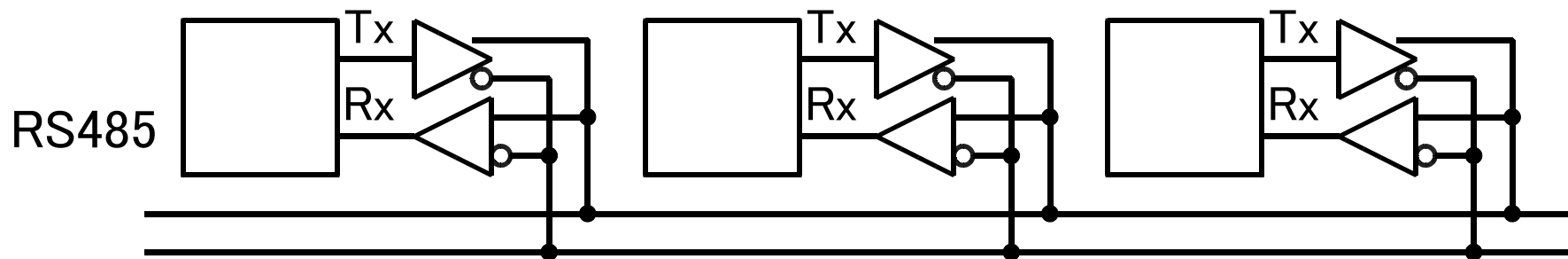
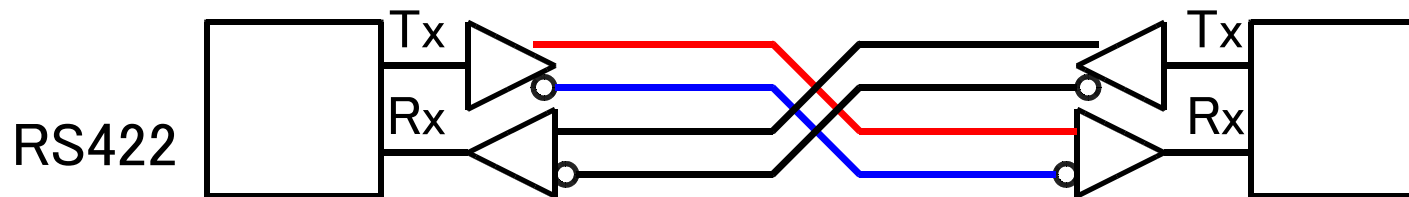
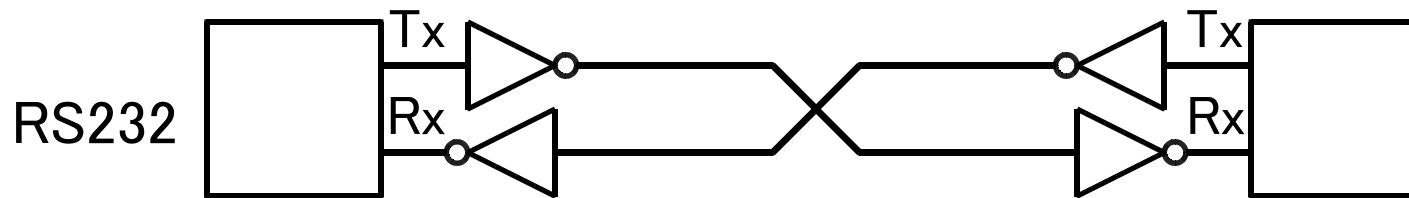
- ・ほとんど全てのコンピュータが持つ機能
(マイコン、パソコン等)
- ・USBから変換する機器多数。
- ・手頃な速度（無難には $\sim 115.2\text{kbps}$ ）



マイコン間の通信

○ 調歩同期式通信

◇ バリエーション

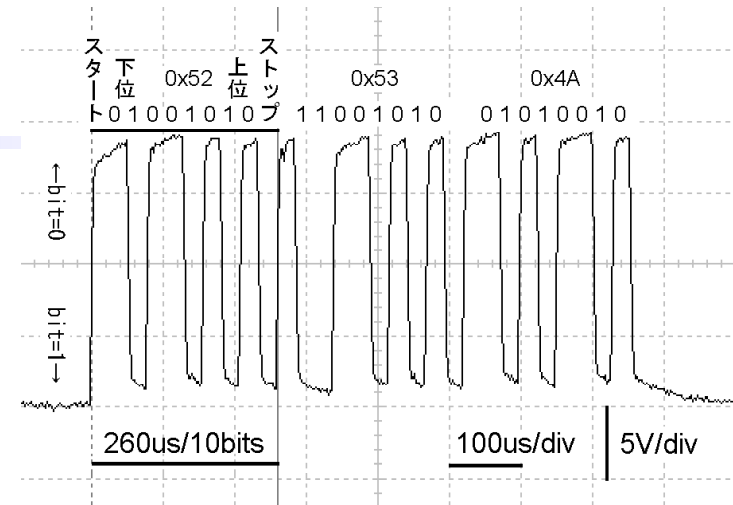


マイコン間の通信

○ 調歩同期式通信

◇補足メモ

- ・ 送受信のクロック(タイミング設定)に誤差があると通信エラーになる、5%以内にする
→ものによって高速時に誤差が出やすい。
※メインのクロックを $1/n$ してつくるため
- ・ RS422,485は終端抵抗が必要。
- ・ 速度設定はオシロスコープでわかる。

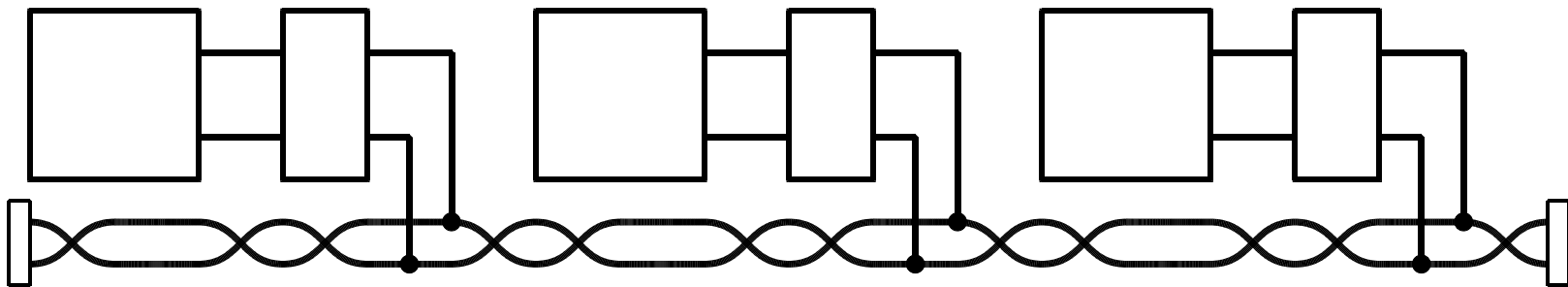


マイコン間の通信

○ CAN (Controller Area Network)

◇特徴

- ・ 多対多のリアルタイム通信を前提。
衝突回避、送信の優先度
- ・ クロック無し、バス接続。
- ・ 差動信号(平衡信号)による耐ノイズ性。



マイコン間の通信

○ CAN (Controller Area Network)

◇補足メモ ※ほぼ未経験

- ・ 多対多で伝統的なRS485に比較して、

通信の方式まで定めている

リアルタイム制御を志向している

※優先度、データ長の短縮

速度控えめ(485:10Mbps越え/CAN:1Mbps程度)

という違いが見られる。

実際に用いられる通信

○ 概 要

◇マイコン～パソコン(OS付き高性能マイコン)

- ・ 調歩同期（シリアルポート、UART, RS232）
- ・ USB（ネイティブ、シリアルポート変換）
- ・ Ethernet（いわゆるネットワーク接続）

◇無線系

- ・ Bluetooth, Zigbee, Xbee, 特定小電力(2.4G)
- ・ 無線LAN
- ・ 携帯電話ネットワーク

マイコン～パソコン間の通信

○ USB

◇特徴

- ・ (比較的) **高速** に、多数のデバイスを接続。
- ・ ホスト(制御する親)とデバイス(接続する機器)
- ・ デバイス機能を内蔵したマイコンは多数。
- ・ **ホスト** には **複雑** な処理が求められるため、小型のマイコンでは無理。

最近はホスト機能を持つ小型マイコンも。

例) PIC24の一部、RX621など

マイコン～パソコン間の通信

○ USB

◇使い方

- ・ハードは対応マイコンに線をつなぐ程度。
 - ・動作が複雑なため、**ソフトの難易度が高い**。
 - ・USB対応の機器をつくるには
 - ・フルに書く＋デバドラをつくる
 - ・何かを偽装する＋汎用ドライバ
 - ・**USB-シリアル変換IC**をつかう
- ※FTDI社 FT2232,4232 など

マイコン～パソコン間の通信

○ USB

◇補足メモ

- ・ 信号速度の速いものなので基板設計なども注意が必要(12Mまでは楽？)。
- ・ 通信が最短でも1ms周期(480Mbps時は125us)となるため、リアルタイム性に注意。
- ・ USBホストが可能なマイコンもあるが、「なんとなく動く」から先が難しい。
※なんとなく、でよければ便利。

マイコン～パソコン間の通信

○ イーサネット Ethernet

◇特徴・使い方

- ・いわゆる「ネットワーク」(の一つ)で汎用性高。
- ・高性能マイコンでは機能を内蔵しており、多少の外付けで、ハードはできる。
- ・「ネットワーク通信」(後述)をさせるには、その上の**プロトコルのコード群**が必要で、“すべて自作”は困難。
- ・基本的にOS上で使用。

無線系の通信

○ Bluetooth

◇特徴・使い方

- ・省電力で短距離の周辺機器通信でよく使われている。
- ・ただし、制御系での使用例などはあまり見かけない。
- ・ホビー系の使用例：
Wii用のコントローラ、ボードをPCで使用
マイコンでUSBホスト＋BT dongleで通信

無線系の通信

○ ZigBee

◇特徴・使い方

- ・センサネットワーク用に、低速ながら
低消費電力を想定した規格。
- ・メッシュネットワークが可能で、直接通信ができなくとも、ネット経由で情報送受が可
→ 広範囲の通信
- ・コーディネータ(ネットに1個必須)、
ルータ(転送)、エンドデバイス、の3種

無線系の通信

○ XBee

◇特徴

- ・ Digi社の無線モジュール基板
シリーズ1: 独自通信規格
シリーズ2: ZigBee仕様
- ・ 少しの設定で
モジュールでシリアル通信を無線化
モジュールの汎用入出力を遠隔操作
などが可能。



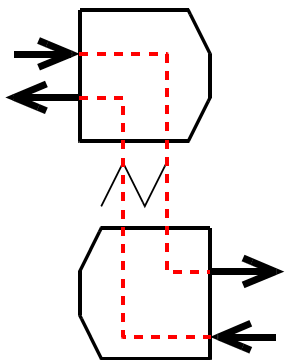
Digi社サイトより

無線系の通信

○ XBee

◇使い方

- ・ 3.3V動作、シリアルポートも。
- ・ 最低限、電源とシリアルをつなぐと動作。
- ・ パソコン上のX-CTUソフトで設定。
- ・ 送受アドレスを設定すると単に通信経路を無線化できる。



- ・ シリーズ2はコーディネータが必須。
(ファームの書き換えで設定)

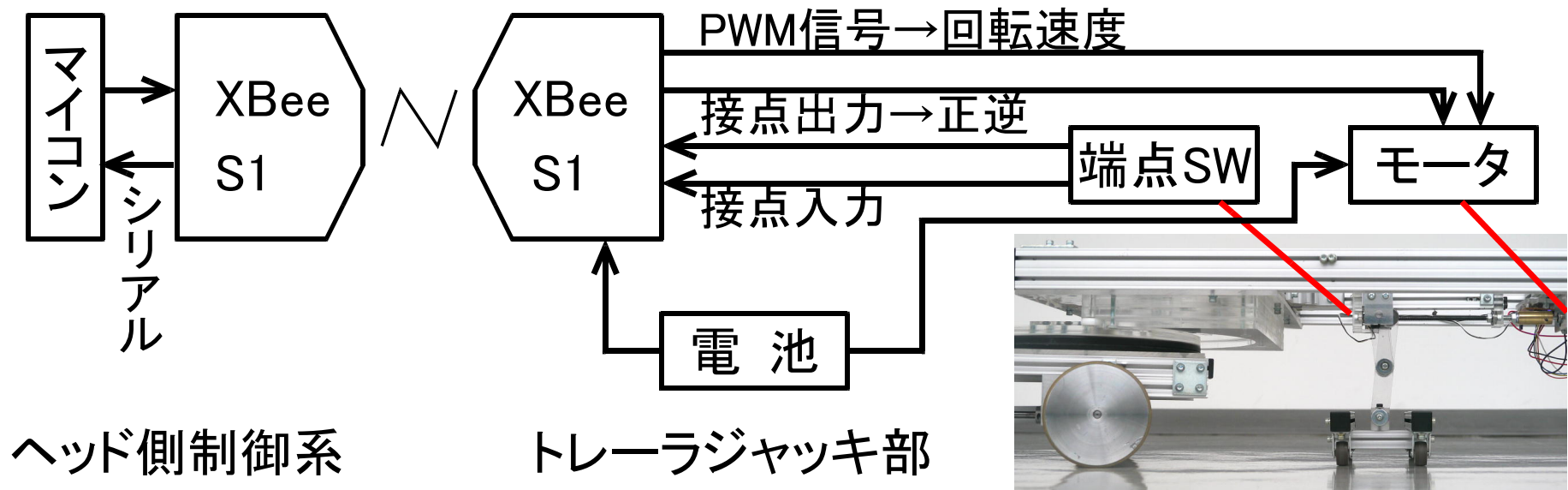
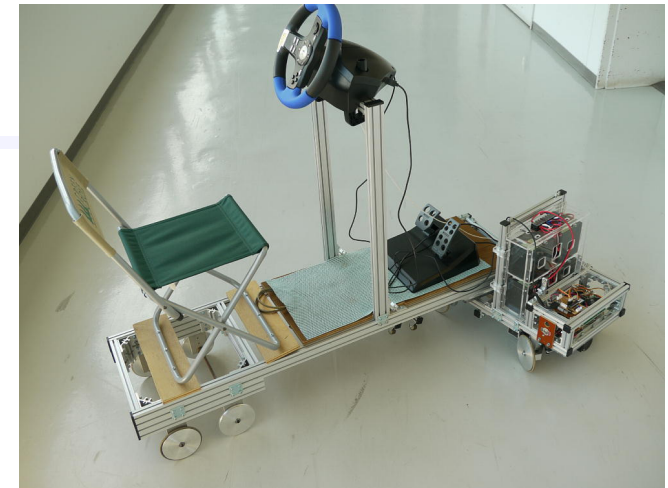


無線系の通信

○ XBee

◇使用例: トレーラロボット

- ・ 遠隔で自動連結用ジャッキの上げ下げ
- ・ “APIモード”で使用→追加マイコン不要。



無線系の通信

○ XBee

◇補足メモ

- ・ 国内販売のモジュールは技適対応。
海外通販は注意が必要。
- ・ 単なる速度、レスポンスはS1のほうが、
S2(ZigBee対応)より良いらしい。
- ・ 同じピン配置で他にも多数のラインナップ
＝差し替え可能。



無線系の通信

○ 携帯電話ネットワーク

◇参考

- ・ 携帯電話ネットワークを使うことで、広域の通信可能な機器をつくることができる。
- ・ 対応マイコン基板例：
アットマークテクノ社製 Armadillo FX500
ウィルコム W-SIM 搭載で通信可能。
- ・ 開発情報は少ないものの、機器自体はかなり世の中にあるもよう。

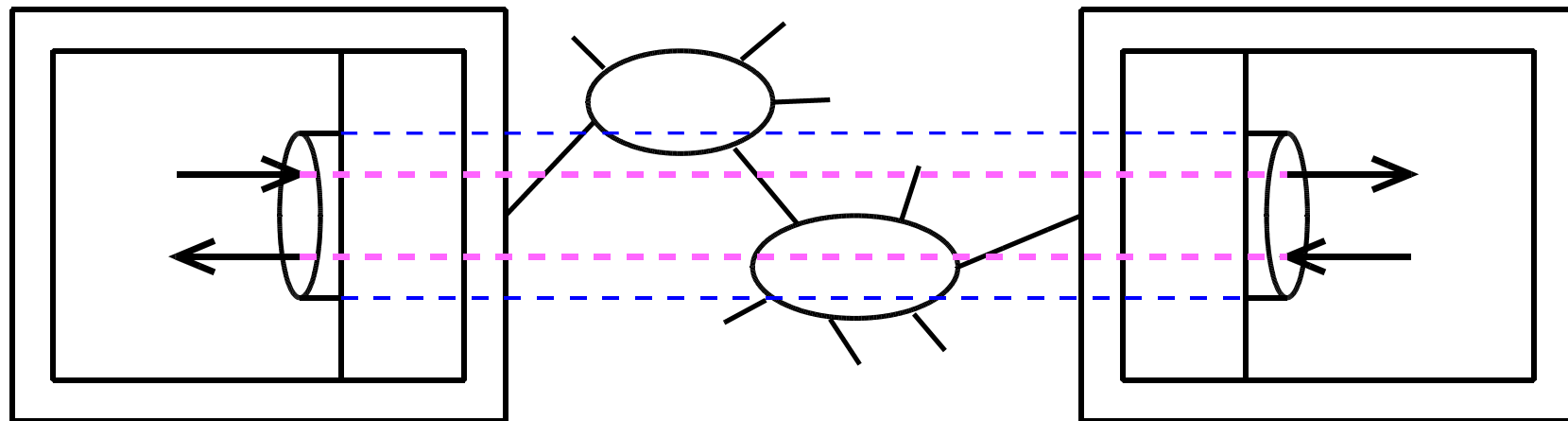
インターネットの通信

○ TCP/IP通信

◇特徴

※入れた順に取り出せる

- ・ ネットを介した、仮想的な双方向FIFO。
- ・ バイト列を流し込み、取り出す。
- ・ エラー処理などは全て済んでいる＝確実。

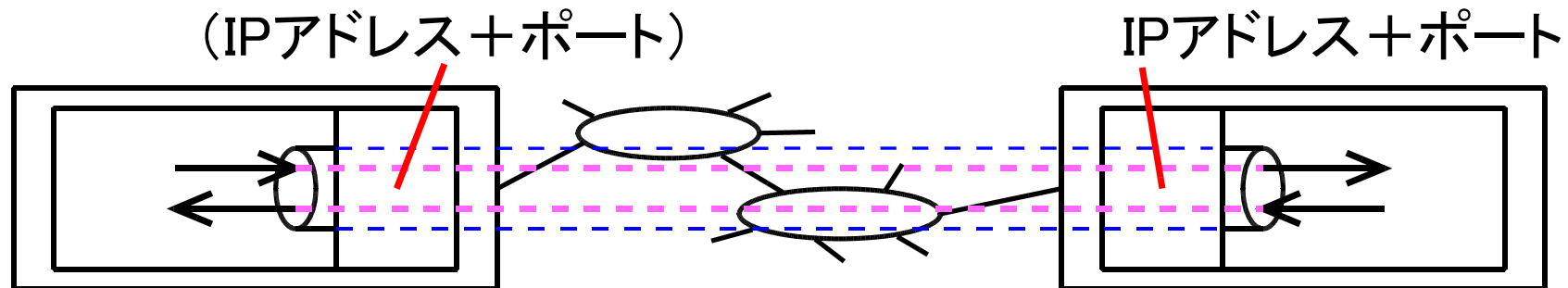


インターネットの通信

○ TCP/IP通信

◇使い方

- ・ 接続作業→使用→終了。
- ・ 端1： ポート(ソケット)を開いて待機
端2： 上記ポートに対して接続要求
- ・ ポートの指定：**IPアドレス**＋**ポート**番号



インターネットの通信

○ UDP/IP通信

◇特徴

※小分けにしたデータの塊

- ・ IP通信の「パケット」をほぼそのまま
- ・ IPアドレス＋ポートで、データ塊を送る。
- ・ TCPと異なり、途中で行方不明になっても再送などはされない。
壊れてエラーが出たら破棄。
→環境によって低信頼、だが高速。

インターネットの通信

○ TCP/IPとUDP/IP

◇補足メモ

- ・動くものをつくることは簡単(特にUNIX系)。
- ・いずれも「到達時間」の保障無し。
＝近距離でも高信頼リアルタイムには不適
→改良品もある
- ・特にTCPはエラー時に再送作業があり、
信頼性はあるが、その間は待たされる。

インターネットの通信

○ HTTP, SMTP, POP

◇各種サービスのプロトコル

- HyperText Transfer Protocol (ポート80)
- Simple Mail Transfer Protocol (ポート25)
- Post Office Protocol (ポート110)
- TCP/IPで特定のポートをつかい、目的に応じて「どのような要求／応答をするか」を規定した手続き集。
- 一般にRFC文書で規定。

インターネットの通信

○ HTTP, SMTP, POP

◇例) HTTP

```
[kumagai@kumagaiSV1 ~]$ telnet www.siip.city.sendai.jp 80
```

```
Connected to www.siip.city.sendai.jp
```

```
GET /index.html HTTP/1.0 入力
```

```
空行入力
```

```
HTTP/1.1 200 OK 以下、応答
```

```
Date: Mon, 20 Aug 2012 11:09:13 GMT
```

```
Last-Modified: Thu, 02 Aug 2012 02:07:17 GMT
```

```
Content-Length: 17370
```

```
Content-Type: text/html ここまでヘッダ
```

```
<!DOCTYPE HTML> ここから内容
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>公益財団法人仙台市産業振興事業団</title> 以下続く
```

インターネットの通信

○ HTTP, SMTP, POP

◇例) SMTP

```
[kumagai@kumagaiSV1 ~]$ telnet localhost 25
Connected to localhost.localdomain (127.0.0.1).
220 kumagai.*****.net ESMTP Postfix
HELO kumagai.*****.net
250 kumagai.*****.net
MAIL FROM:<kumagai@kumagai.*****.net>
250 2.1.0 Ok
RCPT TO:<kumagai@tjcc.tohoku-gakuin.**.**>
250 2.1.5 Ok
```

続く

```
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: test mail
To: kumagai@tjcc.tohoku-gakuin.**.**
From: kumagai@kumagai.*****.net

This is a manual test of SMTP.
.    ←行頭にピリオドのみで終了
250 2.0.0 Ok: queued as 34A29340078
QUIT
221 2.0.0 Bye
```

→メールがとどく

インターネットの通信

○ HTTP, SMTP, POP

◇補足メモ

- 案外、この段階は単純。(下は地獄、上も地獄)
- telnetはTCP接続をしたうえで、文字の送受信を行える→こういう実験ができる。
※本来は遠隔サーバにログインするため
- 多くの通信は、見ての通り、「そのまま」文字が飛ぶ。→セキュリティに不安？
※HTTPSは暗号化したHTTP SMTPも暗号化技術あり

通信系の独自設計例 : Serial Loop

○ 複数のマイコンで相互に通信させたい

◇開発の背景、要請

- ・ロボット制御のため、マイコンを3個以上、接続したい。
- ・マイコン(H8/3052)に外付けせず、標準的機能のみで決着を付けたい。

◇NG案

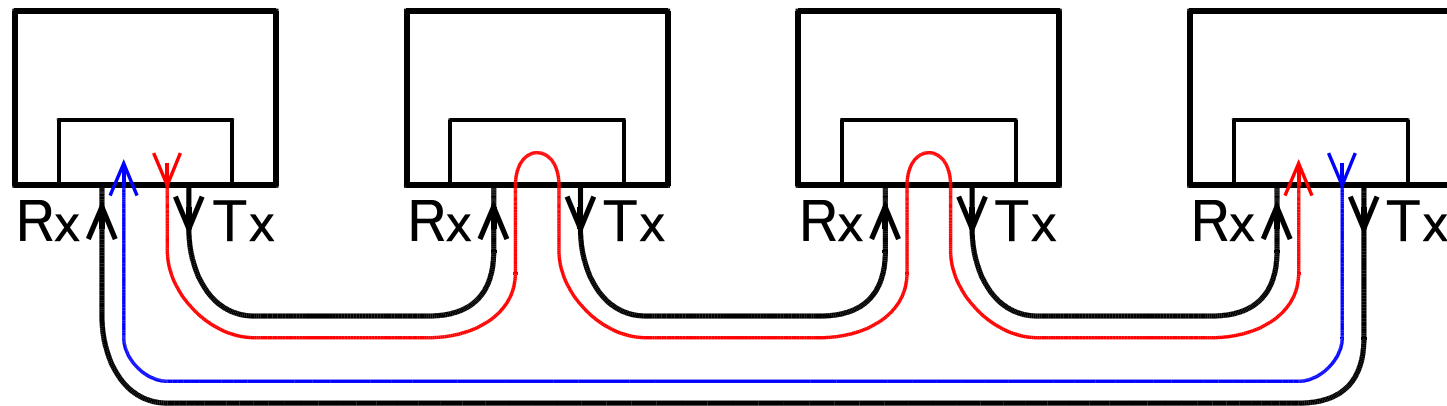
- ・シリアルポートをスイッチ → 無信号期間
- ・RS485のようなバス型 → 専用回路

通信系の独自設計例 : Serial Loop

○ シリアルポートでリングを形成

◇採用した案

- ・ シリアルポートのTxを隣のRxにつなぐ。
- ・ 通信のバケツリレーをする。
→標準機能のみで複数接続

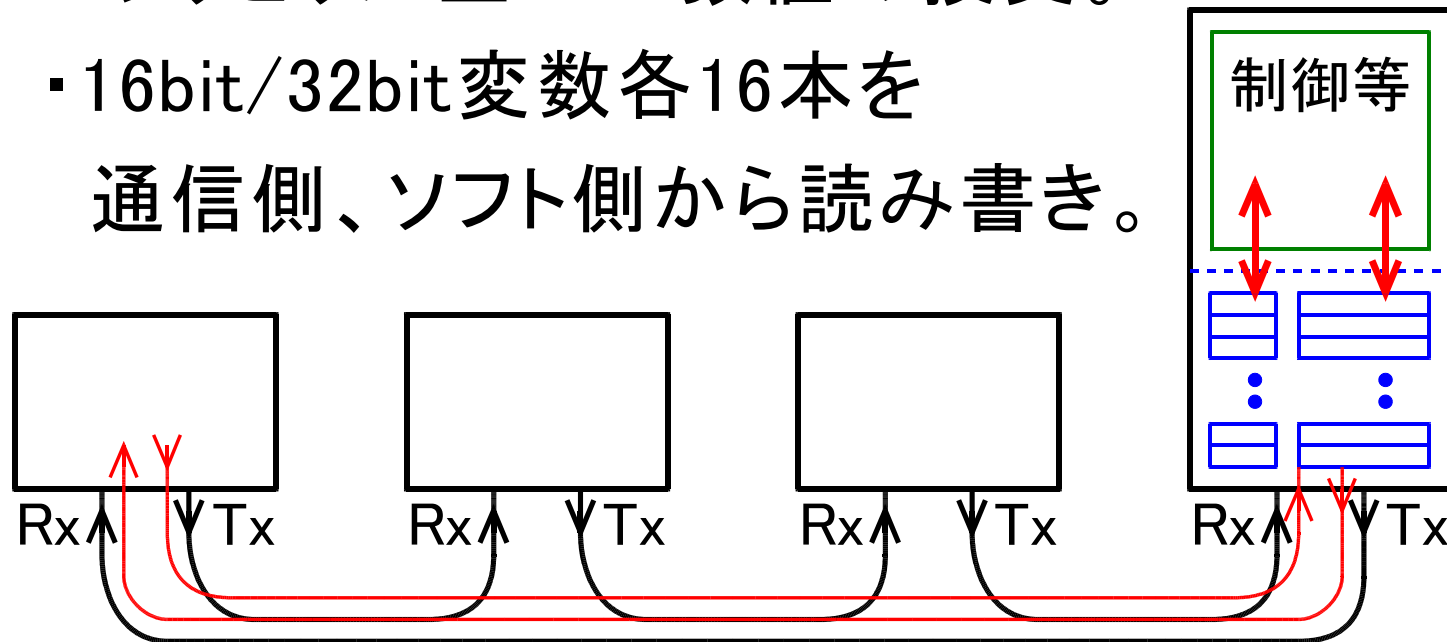


通信系の独自設計例: Serial Loop

○ 疑似共有レジスタでソフトを簡略化

◇通信部分を**共有レジスタ**とあわせて隠蔽化

- ・制御用途はパラメータや目標値、現在値のやりとりが主 → 数値の授受。
- ・16bit/32bit変数各16本を
通信側、ソフト側から読み書き。



通信系の独自設計例: Serial Loop

○ 通信のパケット仕様の概略

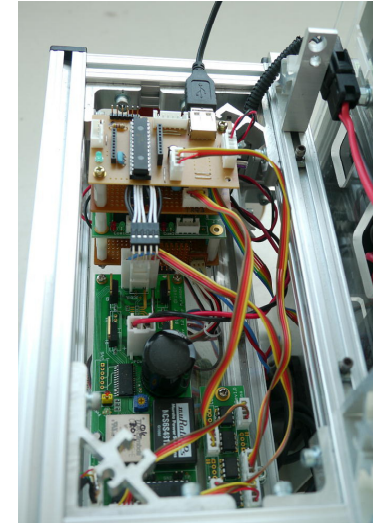
◇パケット構成

- ・先頭に宛先 → 1バイト目見て転送開始
- ・長さ、内容、チェックサム

◇パケットの種類(主なもの)

- ・ set16/32: 指定ノードのレジスタを変更
- ・ get16/32: 指定ノードのレジスタ値を取得
- ・ reply16/32: getの返答
+ 自発的replyで状態報告

通信系の独自設計例 : Serial Loop



○ 評価と課題

◇現状＝ほぼ問題なく活用(約10年)

- ・ 研究室内のほぼ全てのロボットに使用。
- ・ 数十～数百Hzの制御系にも使用。
- ・ 新しいマイコンの使い始めに移植。

◇課題＝信頼性

- ・ 断線 → 全体的な機能不全。
- ・ 通信エラー→偶然頼みの回復に時間要す。

まとめ

○ 通信の基礎

- ・ 通信にはハード部分とソフト部分がある。
- ・ 接続方法には、大きく分けて1対1とバス型接続がある。
- ・ バス型は少ない配線で多数の部品を接続できるが、衝突回避の工夫が必要。
- ・ ソフト部は、最低限のバイト列の送受部分、データとバイト列の変換をする部分、データのやりとり手順の部分がある。

まとめ

※予定していたセンサの接続は
2回後のデジタルセンサの回へ

○ 通信の基礎

- ・ マイコンとセンサを接続する手法に、
SPI型やI2Cの接続がある。
- ・ マイコンとマイコンの接続手法に、
調歩同期シリアル、I2C、CANがある。
- ・ マイコンとパソコンの接続手法に
調歩同期シリアル、USBがある。
- ・ ネットワーク通信は主にTCP/IP, UDP/IP
からなり、比較的簡単に使える。