公益財団法人
ひろしま産業振興機構

産×学 連携

3つのひかり 未来をつくる
広島市立大学
Hiroshima City University

Smart Factory推進Mgr養成 e-Learningコース

組込みシステム概論

広島市立大学　弘中哲夫

目次

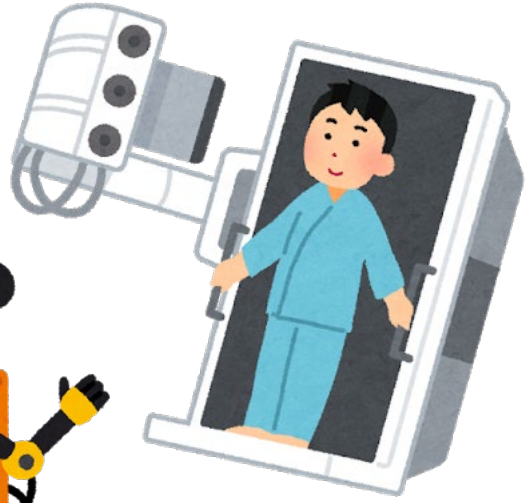
- 組込みシステムとは
- 組込みシステムで使われるプロセッサ
- 組込みシステムでよく使用されるI/O
- 組込みシステムのオペレーティングシステム
- 組込みシステム開発の考え方
- まとめ

1．組込みシステムとは

- **組込みシステム**とは，特定の機能を実現するために機械や機器に組み込まれる**コンピュータシステム**
- **組込みシステム**は，家電，産業用機器，機械設備，医療機器，輸送機器，観測機器，通信機器，娯楽機器，などなど日常のあらゆるところで使われている






組込みシステムの主な用途

- 多機能化
  - 多様な設定が可能な機器において，組込みシステムにより機能の切り替えを実現する
    ⇒**例：エアコン(おやすみ機能，ECO機能，冷暖房切り替え，除湿)**
- 高度化
  - 器機に入力されるデータや器機から出力されるデータに演算処理を行ったり，情報抽出を行うことで付加機能を追加する
    ⇒**例：コンピュータ断層撮影(CT)，音声認識による機器制御**
- 知能化
  - センサーや外部からの入力に応じて，自律的に機器が動作するようにする
    ⇒　例：産業用ロボット，自動航行システム
- ネットワーク化
  - 通信により器機同士が連携動作する
    ⇒**例：HEMS(Home Energy Management System)，スマートメータ**

組込みシステムの開発

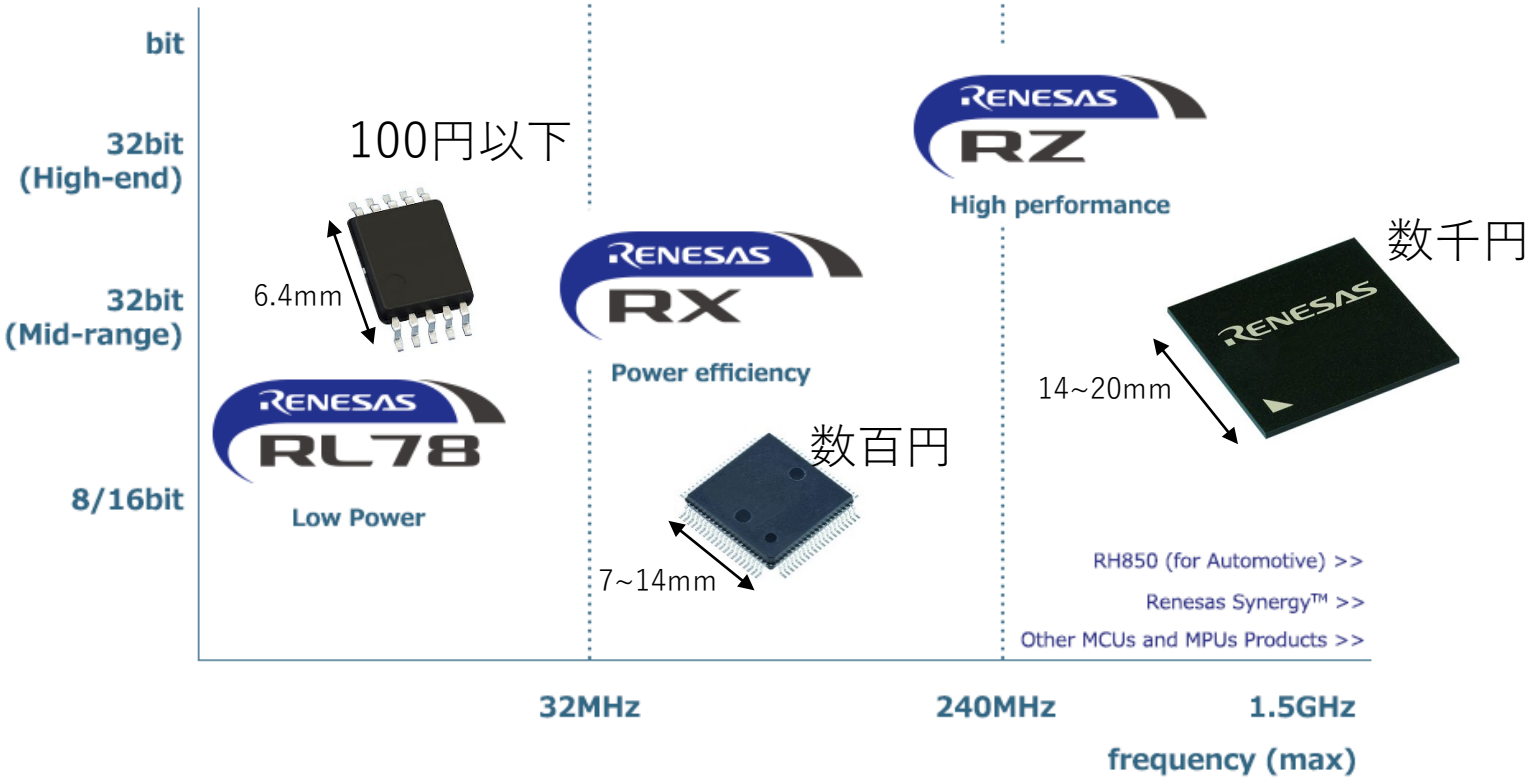
- 組込みシステムと一般のコンピュータシステムとの違い
  - 組込みシステムは**特定用途向けに特化したコンピュータシステム**
  - ハードウェアもソフトウェアも特定用途実現を必要最低限で実現
    - 汎用性を持たないだけ安価にできる
      ⇒ **安価な機器にも組込みシステムを導入可能**
      汎用コンピュータシステムは単体で数万円するので数百円の機器に使えない
- 組込みシステムのソフトウェア開発環境
  - 良く使用される言語は**C, C++, Java,アセンブラ**
  - 開発器機に使用する機器として**ICE, JTAG ICE**などを用いる
    - ICE(In-Circuit Emulator)：本来のCPUの代わりにエミュレータCPUを接続してデバッグ
    - JTAG ICE：CPU内に設けられたデバッグ回路に数本のピン通じて接続してデバッグ
    - ICEを用いればデバッガーソフトウェアを用いたようにデバッグ可能である
      ⇒ CPUレジスタの表示変更，メモリ内容の表示変更，ブレークポイント設定等々
  - オペレーティングシステム
    - **シングルタスクではOSなし** ⇒ OSを使用せずに実装**(開発難易度高い)**
    - **マルチタスクではリアルタイムOS** ⇒マルチタスク処理においてタスクの応答時間を保証するOS **(開発難易度中)**
    - **リアルタイム性を要求されないマルチタスクでは汎用OS** ⇒汎用OSの豊富なソフトウェア資産が活用可 **(開発難易度小)**

目次

- 組込みシステムとは
- **組込みシステムで使われるプロセッサ**
- 組込みシステムでよく使用されるI/O
- 組込みシステムのオペレーティングシステム
- 組込みシステムの開発の考え方
- まとめ

組込みシステムで使われるプロセッサ

- 組込みシステムで使われるプロセッサの能力と価格範囲



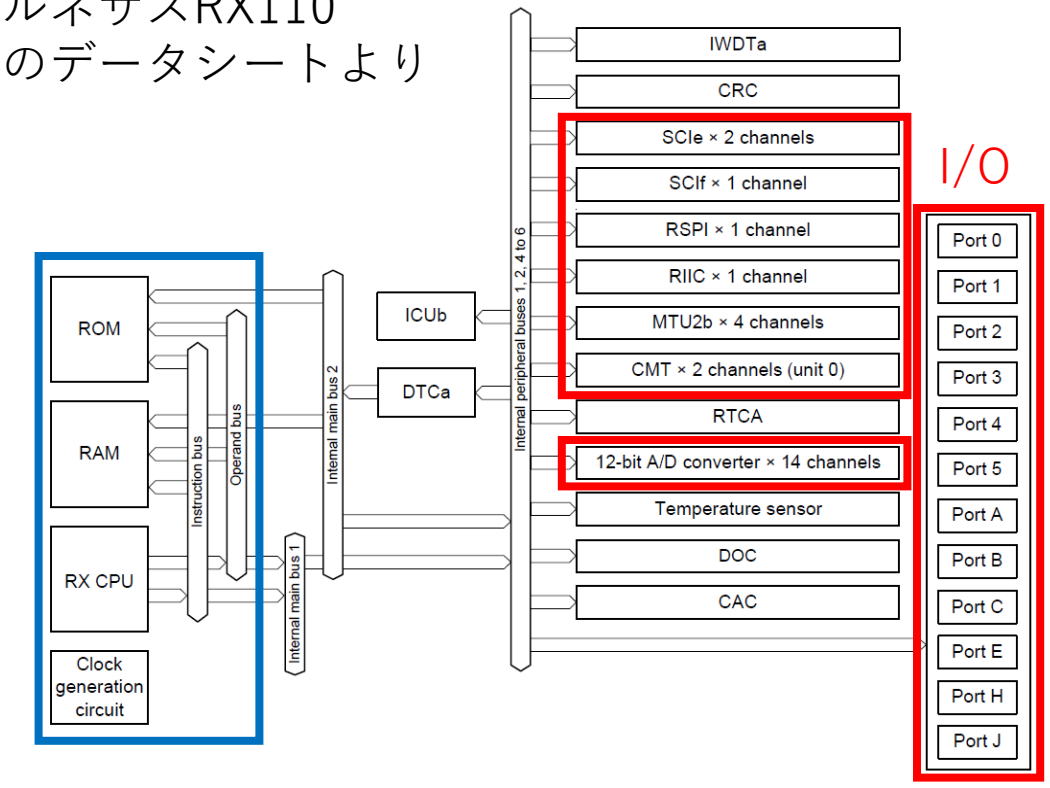
図は https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors.html より引用

パソコンなどで使用されるプロセッサはプロセッサ，メモリ，I/Oがそれぞれ別チップや別部品であることが多いのに対し，組込みシステムではプロセッサ，メモリ，I/Oが１つの単一チップにSoC(System on Chip)としてまとめられていることが多い

演算能力が求められる応用ではパソコン用のCPUや，演算処理能力に特化したDSPなどが使われる場合もある．

組込みシステム専用プロセッサの内部構造

ルネサスRX110
のデータシートより



ICUb: Interrupt controller
DTCa: Data transfer controller
IWDTa: Independent watchdog timer
CRC: CRC (cyclic redundancy check) calculator
SCIe/SCIf: Serial communications interface
RSPI: Serial peripheral interface
RIIC: I²C bus interface
MTU2b: Multi-function timer pulse unit 2
CMT: Compare match timer
RTCA: Realtime clock
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit

Figure 1.2 Block Diagram

- 32bit CPU 動作周波数32MHz
- ROM 8~128Kbyte
  （プログラム用）
- RAM 8~16Kbyte
  （データ用）
- Watchdog timer
- 温度センサー
- タイマ　6個

チップ単独で
プログラムを
実行可能

- **通信チャネル 5本**
- **A/Dコンバータ　14個**
- **汎用I/Oポート　24~50ピン**

豊富な
入出力

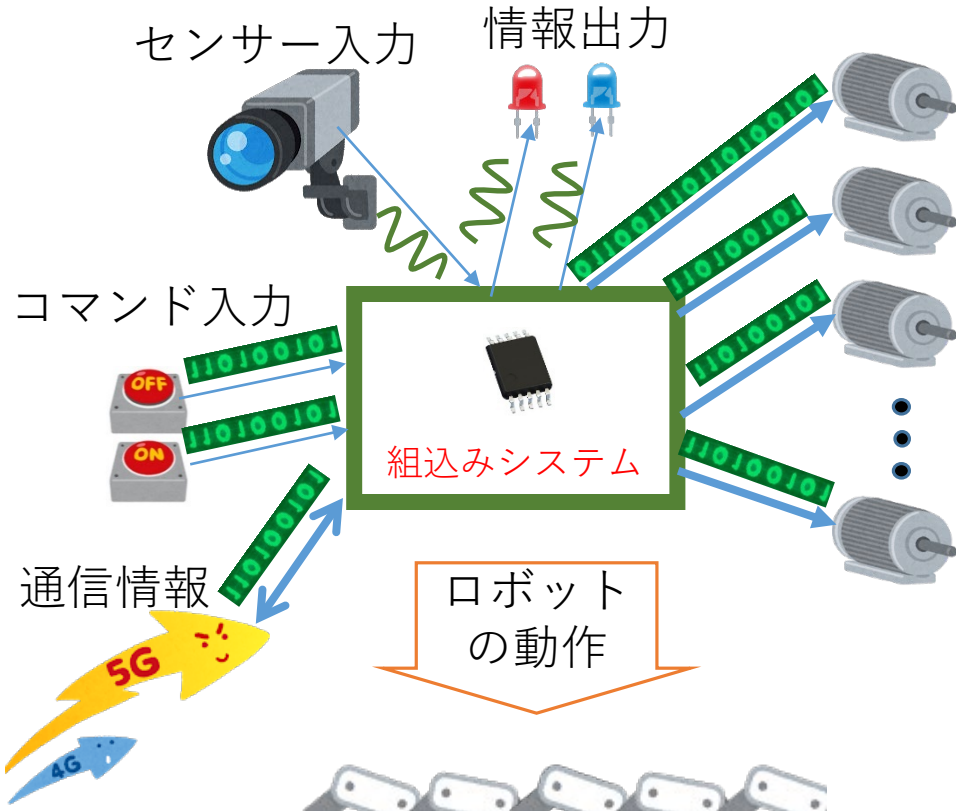
主記憶として大容量メモリが必要な場合は
別チップで用意する場合がある

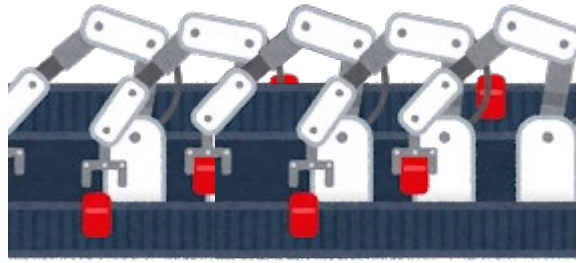
組込みシステムの動作

- 組込みシステム用プロセッサは
  - **入力として**
    - コマンド
    - 各種センサー情報
      アナログやデジタル値
    - 通信情報
    - 時刻情報(タイマーや時計)

    など用いて判断の結果,
  - **出力として**
    - 各種アクチュエータの制御値を出力
      アナログやデジタル値
    - 電子機器への制御情報
    - 通信情報

    などをプログラムされた手順で出力し，制御を行う





ロボットとしての機能を実現

目次

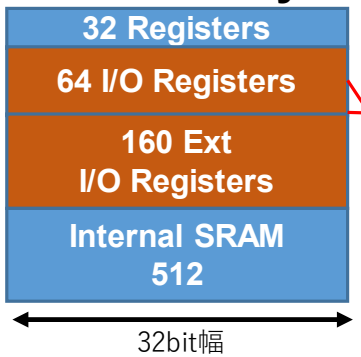
- 組込みシステムとは
- 組込みシステムで使われるプロセッサ
- **組込みシステムでよく使用されるI/O**
- 組込みシステムのオペレーティングシステム
- 組込みシステム開発の考え方
- まとめ

組込みシステムでよく使用されるI/O

- メモリマップドI/O（I/O機器の管理方式）
  - CPUのメモリに対するRead/Write命令を用いて入出力機器のレジスタにアクセス
  - プロセッサのアドレスバス，データバスを用いるので接続可能レジスタ数はメモリ空間の大きさに等価
  - 各I/O機器のレジスタにはメモリアドレスが振られ，アドレスに対しWriteするとI/O機器にデータ送信になり，ReadするとI/O機器からのデータ受信になる
- パラレルポート
  - パラレルバスとして多数の規格がある．一般にプロトコルが単純であるが長い配線長での高速通信に向かない．
  - GPIO(General-purpose input/output)ポートは汎用デジタル入出力ポートであり，ポート単位で入力または出力に使用する．各種センサーやスイッチ類からのデータの読み込み，各種制御対象機器の制御・データ信号出力に使われる．
- シリアルポート
  - I2C(Inter-Integrated Circuit)はフィリップス社で提唱されたシリアルバスである。バスは2本の双方向のオープンコレクタ信号線が2本だけから構成される低速通信バスである．最大112個の機器を接続でき，安価であることから広く普及．100Kbps，400Kbps，1Mbpsの通信が可能
  - SPI(Serial Peripheral Interface)はモトローラ(現在はNXPセミコンダクターズ)で提唱されたシリアルバスである．信号線は4本で構成される低速通信バスである． I2Cに比べ通信速度が速く数Mbpsの通信が可能(プロセッサの性能依存)
- アナログ入出力
  - アナログ入出力はプロセッサ内部ADC/DAC(Analog Digital Converter/Digital Analog Converter)に接続されたポートで行われる．ポートでは外部から入力されるアナログ値をデジタル値に変換してプロセッサで計算処理に使用できるようにする他，計算処理されたデジタル値をアナログ値に変換して外部へ出力する
  - PWM出力はデジタル出力であるが，パルスのデューティ比を一定周期で変化させることでパワー素子をオン/オフ駆動し，損失を抑えながら負荷をアナログ駆動するために使用する

メモリマップドI/O（ATmega48Aの場合）

Data Memory Map



Program Memory Map

ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	101
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	101
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	101
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	100
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	100
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	101
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	100
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	100
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	100
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–	

I/Oメモリマップ（※より引用）

- 例えば，正しくPORTを初期設定すれば
  プログラムでアドレス0x0028に0x03を書き込むと
  GPIOであるPORTCからPORT0=1，PORT1=1，
  PORT2~6=0を出力できる．

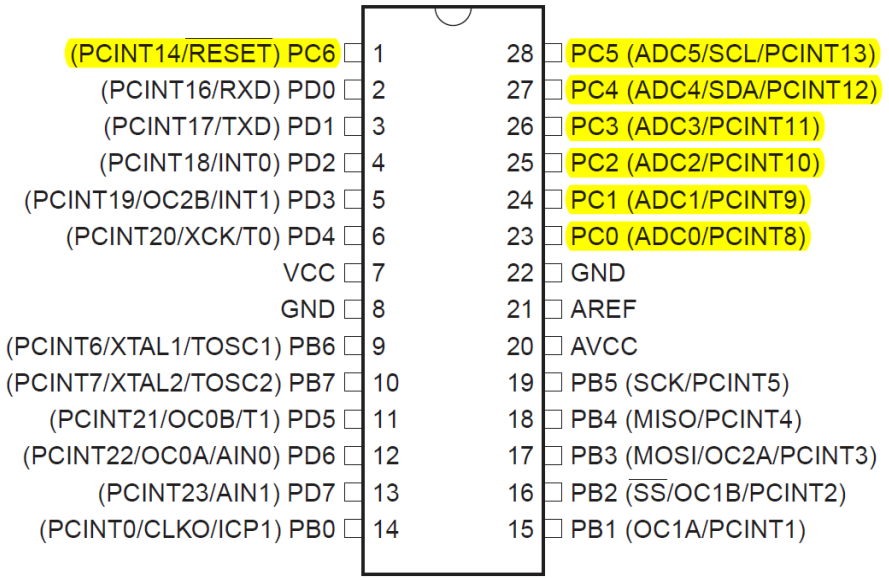
※Microchip Technology Inc. ATmega48A DataSheet

メモリマップドI/Oを使ったプログラム

- C言語使ったメモリマップドI/Oのアクセス方法の例

```
#define PORTC    (volatile unsigned char *)(0x0028)
          .
          .
unsigned char rdata;
rdata = PORTC;   //PORTCから変数rdataにデータを取り込む
PORTC =0x31;    //PORTCから0x31を出力する
```

- 上記例は，プロセッサメーカが用意するライブラリなどを使用しないアクセス例である．多くの場合はもっと簡易にアクセスできるライブラリを用意されている

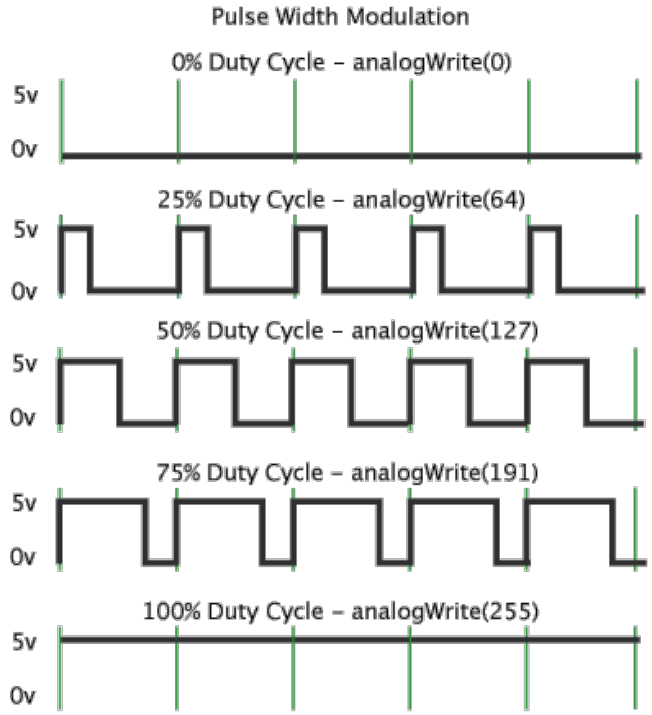


ATmega48Aの場合，左のプログラムで上記図のピンPC0~PC6でデジタル信号の入出力ができる．

※Microchip Technology Inc. ATmega48A DataSheet

PWMを用いたアナログ出力

- アナログ出力としてPWM(Pulse Width Modulation)出力をハードウェア的にサポートしたプロセッサも多い
- PWMは右図のようにデジタル出力において信号の振幅を変えずにデューティ比を変化させることで信号の平均電圧や電流を変化させることでアナログ値の出力を行う
- PWMを使うことでLEDの調光，DCモータの速度調整などに使用可能




```
for (i=0; i<256; i++){
    // iの値でLEDを点灯させる
    analogWrite(LED_PIN, i);
    delay(20);
}
```

Arduinoフレームワークで PWM制御する例

※https://www.arduino.cc/en/Tutorial/PWM

目次

- 組込みシステムとは
- 組込みシステムで使われるプロセッサ
- 組込みシステムで使われるI/O
- **組込みシステムのオペレーティングシステム**
- 組込みシステム開発の考え方
- まとめ

組込みシステムのオペレーティングシステム

- オペレーティングシステムが必要な理由
  - 開発の手間削減（デバイスドライバ，プログラミング用API）
  - プログラムの可搬性(既存の開発リソースの活用，再利用)
- PC等の汎用システム用OSとの違い
  - 組込みOSはアプリケーションに**必要のない機能が削除可能**
    （組込みシステムは特定の目的のために存在する）
  - **リアルタイム性**がある.
    （用途によってはリアルタイム性が不要，その場合はリアルタイム性のないOSも使われる）
    - 1秒以内に必ず応答する等の応答のデッドラインが設けられるOS
    - 平均的に0.1秒で応答するOSでも稀に1秒のデッドラインを守れない場合は，リアルタイム処理が可能とは言わない
  - **リアルタイム性保証に向いたスケジューラ**
    - 外部割込み等で発生したリアルタイム性の重要なタスクはできるだけ高速に処理できるようにする
      ⇒　優先的にCPUを割り当てる等
    - タスクの応答時間を見積もれるスケジューリング
    - 汎用OSは特定タスクのリアルタイム性より全体のスループットを重視
      ⇒ 全タスクに公平にCPUを割り当てる

リアルタイム処理の種類(レベル)

- ハードリアルタイム処理
  - 時間制約(デッドライン)厳守が必須のリアルタイム処理，厳守できない場合は**システムは故障とみなされる**処理
  - ロボットのアーム制御等（入力信号の取りこぼしや出力信号が間に合わない場合，アームが異常動作をする）
- ファームリアルタイム処理
  - 時間制約厳守できなかった場合，**その処理は無価値**になるが頻発しなければ，**システムとしては異常が無い**処理
  - 動画や音声のエンコーディング処理等(1フレームやサンプルの圧縮に失敗しても動画や音声が使用に耐える状況等)
- ソフトリアルタイム処理
  - 時間制約厳守できなかった場合，**その処理の価値は低下する**が，無価値とまでにはならない処理 ⇒　場合によっては汎用OSでも対応可能
  - ヒューマンインタフェース処理等(キー操作に対するディスプレイの反応などもたつく事で正しく動作するが性能を悪く感じるなどの状況)

リアルタイムOS中に同時に3種のリアルタイム処理が同時に存在することも可能である．各タスクが必要とするリアルタイム処理の種類を適切に設定することが組込みシステム開発において重要である

主要なリアルタイムOS

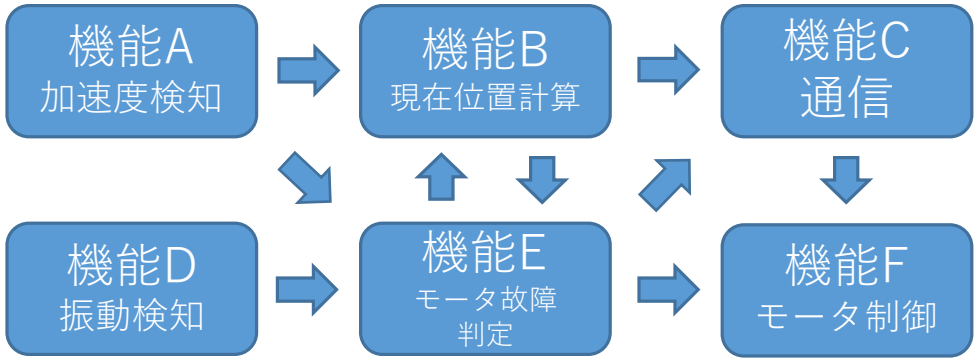
- ITRON
  - TRONプロジェクトの一環として，トロンフォーラムで開発．日本ではメジャーな組込みOSである．ITRONの他により限定した仕様を持つuITRONも良く利用されている．ライセンスは無料で利用可能．
- T-Kernel
  - TRONプロジェクトの一環として，トロンフォーラムで開発．標準開発環境上でソフトウェアの開発を進めて完成度を高め，ターゲットのハードウェア上へ移植するべきであるという方法論の元開発されている．そして，そのための環境としてT-Engineを提供している．T-KernelはT-Licenseというライセンスの下で製品開発すればライセンス費用は不要．
- VxWorks
  - Wind River Systemsが開発したリアルタイムOS．大型産業機器など比較的大きな組込み機器に使用されることが多い．使用にはライセンス契約が必要
- RT-Linux
  - Linuxから派生したリアルタイムOS．Linux Foundationで開発．ネットワークを活用する商用製品の開発にも使用されている．ワンボードマイコンであるRaspberry Piにも搭載されている．Linuxで開発したアプリを流用できるメリットがある
- INtime
  - RadiSys社が開発したリアルタイムOS．組込みシステムにWindows向けプログラムを活用できるメリットがある．産業機器やロボット向け．使用にはTenAsys社とのライセンス契約が必要

目次

- 組込みシステムとは
- 組込みシステムで使われるプロセッサ
- 組込みシステムで使われるI/O
- 組込みシステムのオペレーティングシステム
- **組込みシステム開発の考え方**
- まとめ

組込みシステム開発の考え方

- 汎用システムを器機に組み込んで組込みシステムにするという考え方では，効果的に組込みシステムを構築できない．
  - 組込みシステム ⇒　**特定の機能を実現するためにだけ存在（部品）**
  - 汎用システム　⇒　どうしても開発器機全体の頭脳を作るイメージになる
    （このタイプの組込みシステムも存在するが一部）



実現したい装置機能ブロック図

各機能のインプリメントを考える上で
組込みシステムを1つの選択肢として考え，
コストやパフォーマンスを踏まえて
組込みシステムで実現するのか，その他の方法で実現するのかを決定する

例：機能A,D,Eは計算速度を必要としないので50円の組込みプロセッサ１つとセンサで構成，機能B,Cは通信モジュールを内蔵した500円の組込みプロセッサ１つで実現，機能Fは200円で実績がありモータを直接ドライブできる専用LSIで実現などなど．

機能分割の手順

- 目的のシステムを実現するのに必要な機能(処理)を列挙し，機能間(処理間)で受け渡される情報を明確化する
- 各機能の実現方法（HW? SW?）はここでは決めない
- 曖昧さのない明確な機能分割を行い，各機能をきっちり定義する
- 必要に応じて機能を階層的にさらに分割し，機能の粒度を適切なものにする

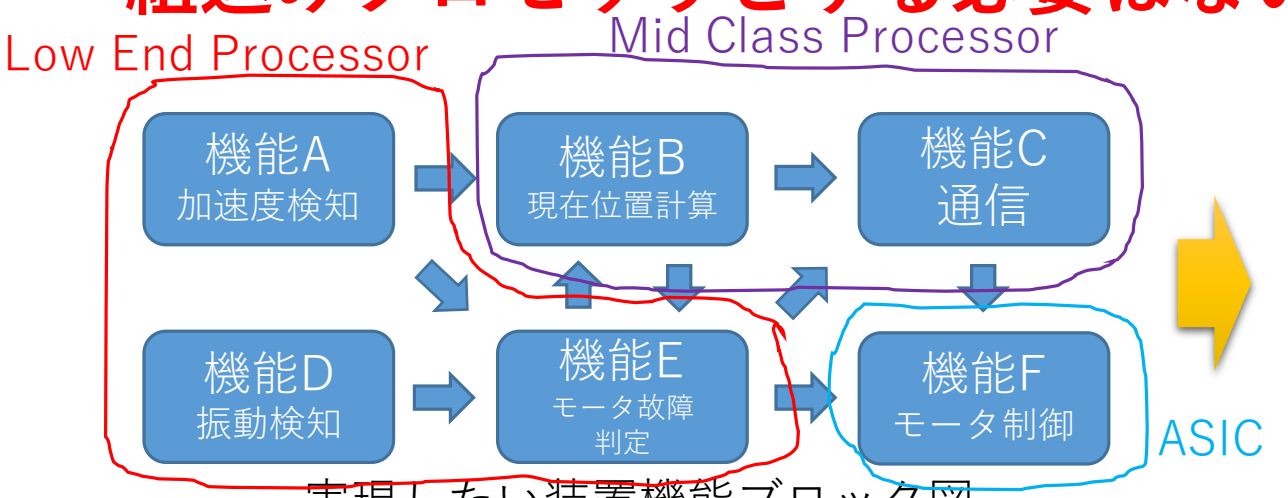
機能リスト
機能A：加速度を検知し，方向ベクトルと継続時間を出力する
機能B：方向ベクトルと継続時間から新しい現在位置を求め出力する
機能C：現在位置と異常種別とその有無を送信し，次の移動方向と時間を受信し，それを出力する
機能D：振動を検知し，ノイズを除いた上で振動情報を出力する
機能E：振動情報と方向ベクトルから異常振動のみ選別し，異常種別とその有無を出力する
機能F：移動方向と時間に応じてモータを制御する．また，異常が有る場合は直ちに停止する

機能分割のポイント

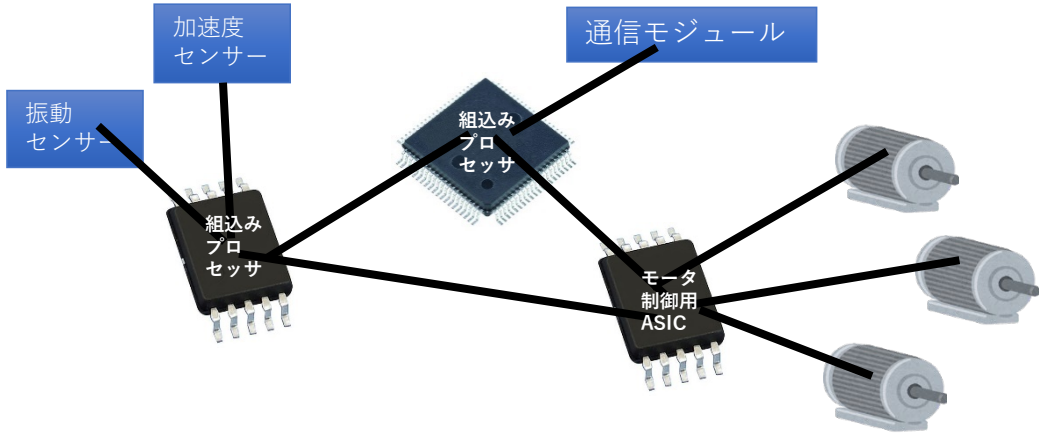
- 各機能を**きっちり曖昧性が無いように定義**する
  - 複数人での協調作業の効率にもつながる
  - 組込みシステム開発を外部に委託することが可能になる
- 機能間で**受け渡す信号/データを詳細化**する
  （信号/データの意味，タイミング，ビット幅）
  - これを曖昧のまま設計を始めると，動作異常，再設計につながる
  - 上記を防止するため，詳細設計前に機能分割段階でグループによる設計レビューが非常に大事です

実装方法のマッピング

- 機能分割結果の各機能を組込みプロセッサで組込みシステムとして実現するのか，その他の実現方法（専用LSI，機械部品，等々）の**いずれで実装するのかコストやパフォーマンスを考慮して決める**
- 組込みプロセッサを実装方法として選ぶ場合，**１つ機能１つの組込みプロセッサとする必要はない**



実現したい装置機能ブロック図



組込み専用プロセッサを使用すると
各機能のワンチップ実装も可能になる

まとめ

- 組込みシステムとは
  - 特定の機能を実現するためにだけに使われるコンピュータシステムである
- 組込みシステムに使われるプロセッサ
  - 多様な価格帯，性能の物が存在する
  - 組込みシステム用プロセッサはCPUのみでなく，プログラム用フラッシュメモリ，データメモリ，I/Oデバイスなども含めて周辺回路含めてワンチップ化されているものも多い．
- 組込みシステムのオペレーティングシステム
  - 特定機能実現に不要なOS機能を削除できる
  - リアルタイム性の保証ができる
- 組込みシステムの開発の考え方
  - 機能の実装方法はコストやパフォーマンスを考慮して決める
  - 組込みプロセッサはあくまでも部品であり，機能を実現する手段である

確認問題

問題

問1 組込みシステムとパソコン等の汎用コンピュータシステムの違いは何か

問2 リアルタイム処理にはどのような種類があるか

問3 機能分割を行ったあと，組込みシステム用プロセッサで実現する部分を決定するが，どのような点を考慮するべきか

問題解答

問1 組込みシステムとパソコン等の汎用コンピュータシステムの違いは何か

特定の機能実現に特化したコンピュータシステムであるか否か

問2 リアルタイム処理にはどのような種類があるか

ハードリアルタイム処理，ファームリアルタイム処理，ソフトリアルタイム処理の3種

問3 機能分割を行ったあと，組込みシステム用プロセッサで実現する部分を決定するが，どのような点を考慮するべきか

コストやパフォーマンスを考慮して組込みシステム用プロセッサで実装するか否か決める