

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

C02/Rev 1.0

第2回

マイコンの初歩

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

○ マイコンの初歩

テーマ1: コンピュータの動作の仕組み

- ・ 演算と記憶

テーマ2: コンピュータの構造

- ・ CPU/MPUとメモリと入出力
- ・ パソコンとマイコン

テーマ3: 組込マイコン

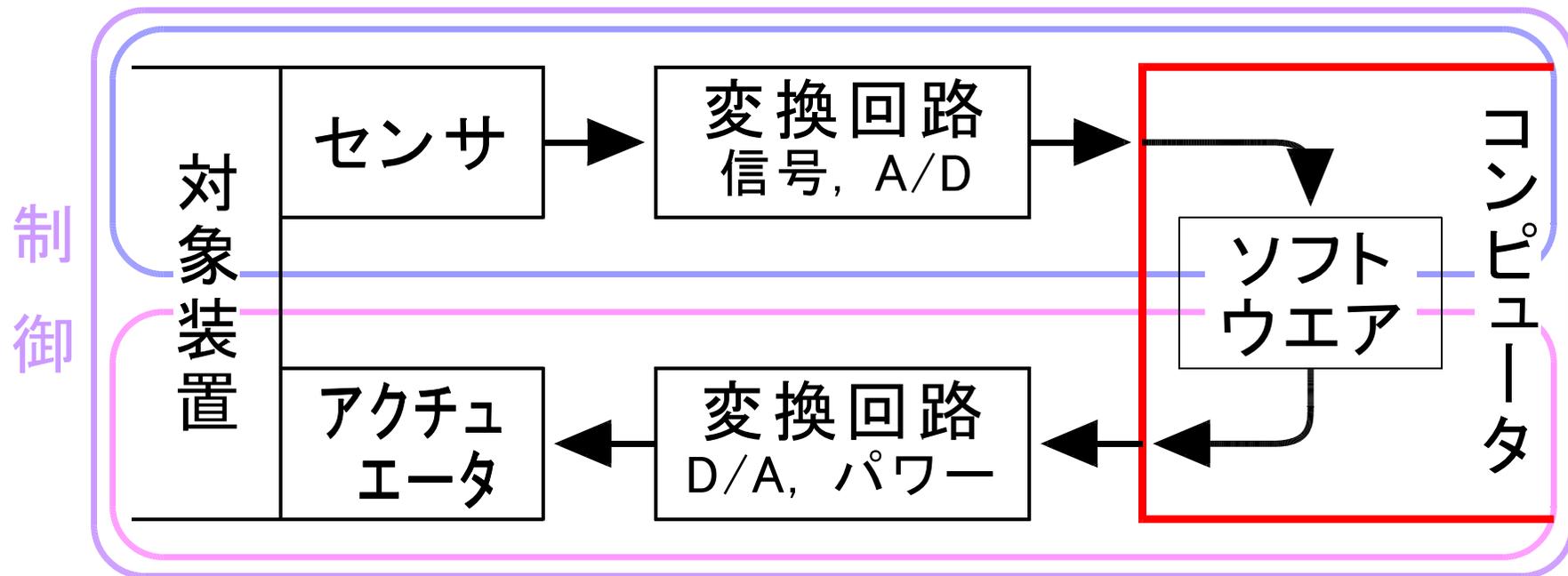
- ・ 用途と必要な機能
- ・ マイコンの選定, とっかかり

メカトロニクスなシステム (No 1より)

○ メカとコンピュータの情報ループ

制御 = 計測 → 演算 → 操作

計 測



操 作

メカトロとコンピュータ

○ コンピュータ制御のためのコンピュータ なぜコンピュータ制御か？

→ 簡単だから

1: 修正はソフトウェアのほうが簡単

2: 回路では面倒な処理が簡単

(足し算ですら回路だと部品が複数必要)

→ 安いから

1: 安いコンピュータは数十円

2: 部品削減

メカトロとコンピュータ

○ コンピュータ制御のためのコンピュータ

パソコンでOK？

→ たぶん、いいえ

1: オーバースペック過ぎ

2: 高コスト 大き過ぎ

マイコンでOK？

→ たぶん、はい

ただし、パソコンに比べて癖が多過ぎ

※敢えて癖を付けてある(後ほど)

メカトロとコンピュータ

- コンピュータ制御のためのコンピュータ
メカトロニクスだけ？

→ いいえ

メカトロニクス・ロボット +
世の中の電子機器、ほとんど全て

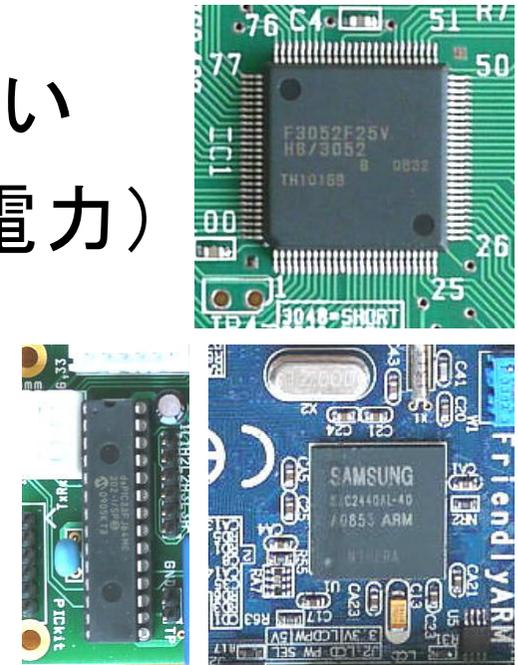
「制御」の対象はメカのみならず。
制御のみならず。

メカトロとコンピュータ

○ そもそもマイコンって？

とりあえず、

- ・ **小さい**コンピュータ(マイクロコンピュータ)
 - ・ **装置**というより**部品**
 - ・ パソコンに比べて性能は低い
 - ・ あまり電気を食わない(省電力)
- 程度と置いていてください。



予備知識

○ コンピュータの扱うデータ

★すべての基本は0と1、オフとオン

いわゆるデジタル → 第3回

回路の配線などの「状態」が2種類

★0と1を多数組み合わせる &

組み合わせを特定のルールで解釈する

→「いろいろな情報」を表現する

いろいろ: 数値(整数、小数)、命令、文字、など

予備知識

○ 0と1の組み合わせ

0 (オフ)

1 (オン)

1個の「0か1」
→ 2種類

0	0
---	---

0	1
---	---

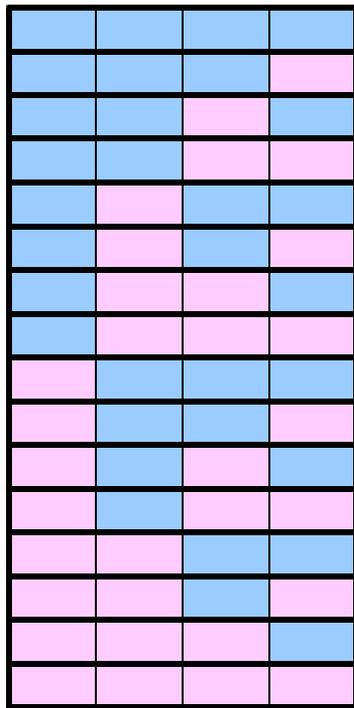
1	0
---	---

1	1
---	---

2個の「0か1」の組み合わせ
→ 4種類

予備知識

○ 0と1の組み合わせ



4個の「0か1」の組み合わせ

→ 16種類

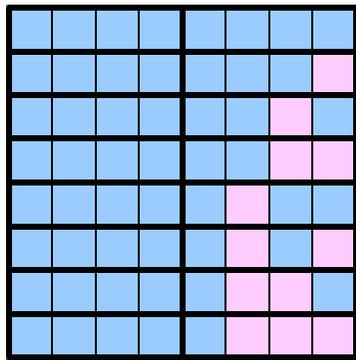
何種類か？

$$2 \times 2 \times 2 \times 2 = 2^4 = 16$$

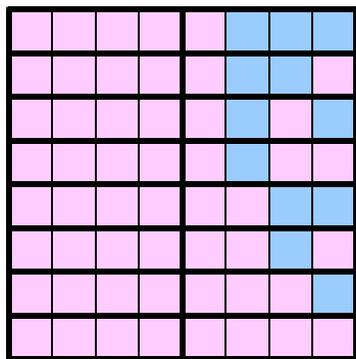
※それぞれ2通り選べる

予備知識

○ ビット(bit) と バイト(byte)



:



8個→256種類

(2を8回=2⁸)

コンピュータの基本単位:

この1個の「0か1」が「**ビット**」。

「0か1」8個の組みが「**バイト**」。

※1バイト=8ビット

予備知識

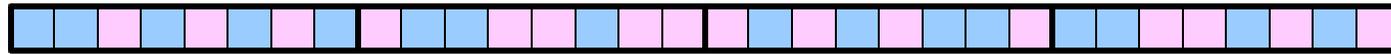
○ ビット数と表せる情報の種類数



8ビット = 256種類



16ビット = 65536種類



32ビット = 4294967296種類

= 約43億 種類



64ビット = 約1845京 種類

予備知識

○ コンピュータにおける 0/1 の解釈例

★数値

整数値（正のみ、正負、いわゆる2進数）

例 8bitで0～255, -128～127

小数値（浮動小数）

例 32bit（単精度）64bit（倍精度）

★命令/プログラム

例 0000 加算命令、0001 減算 …

★オンオフの組み合わせ（そのまま）

テーマ1:コンピュータの動作の仕組み

○ 概要

★コンピュータの動作の仕組み

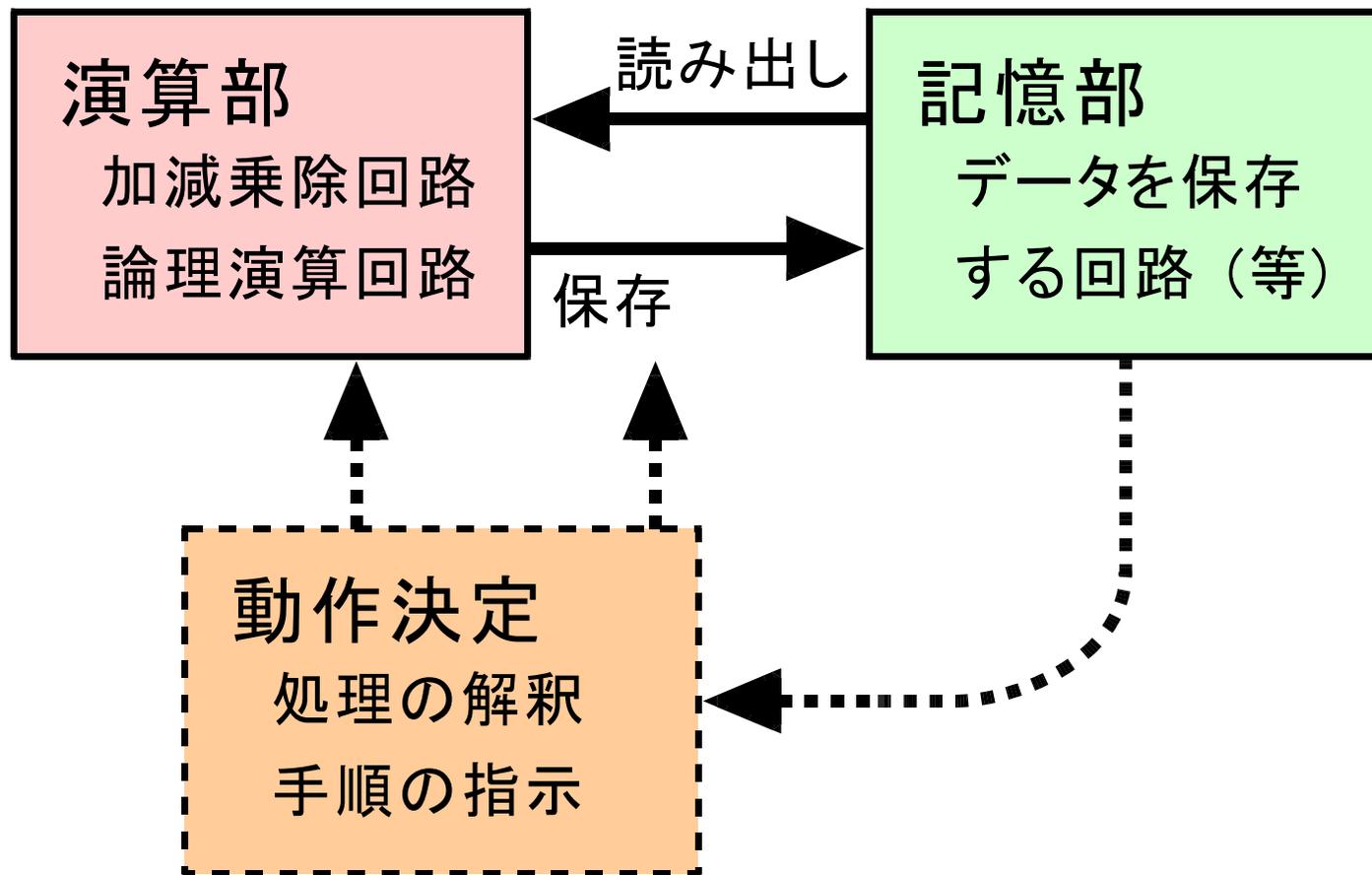
- ・演算部と記憶部

★処理の実行

- ・命令の取り込みと処理
- ・移動、演算、分岐、呼び出し
- ・条件判断

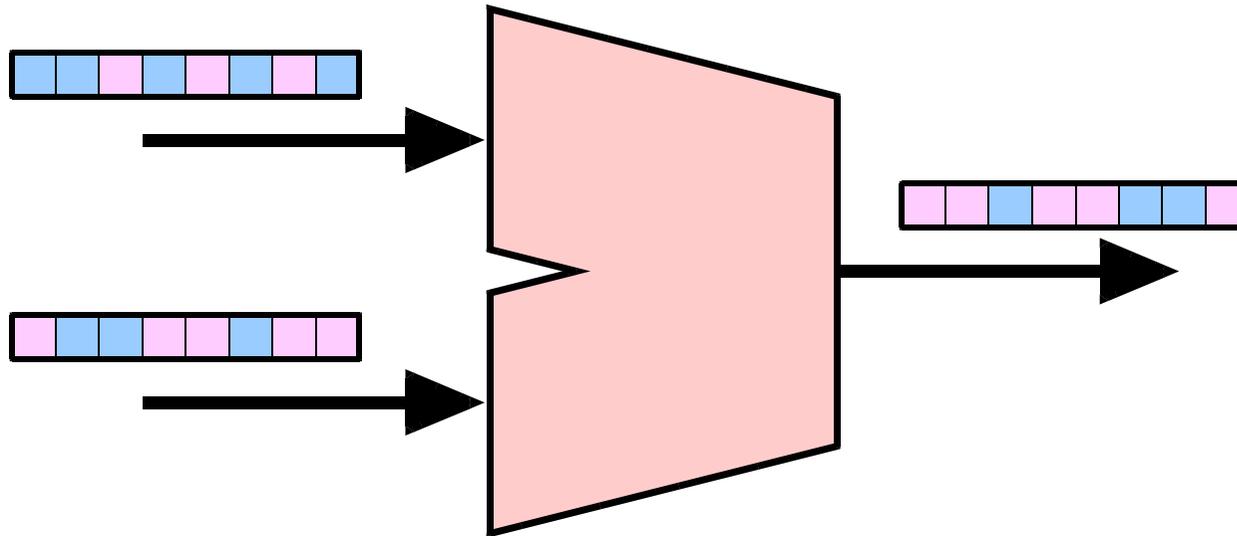
コンピュータのしくみ

○ 演算部と記憶部



コンピュータのしくみ

○ 演算部



何らかのルールで**入力を加工して出力**

ルール： 入力を特定種類の数値と解釈

→ 加減乗除など → 数値に該当する 0/1群

コンピュータのしくみ

○ 記憶部

★ 1バイト単位の保管箱群

★ 1カ所ごとに番地がある

番地 = アドレス

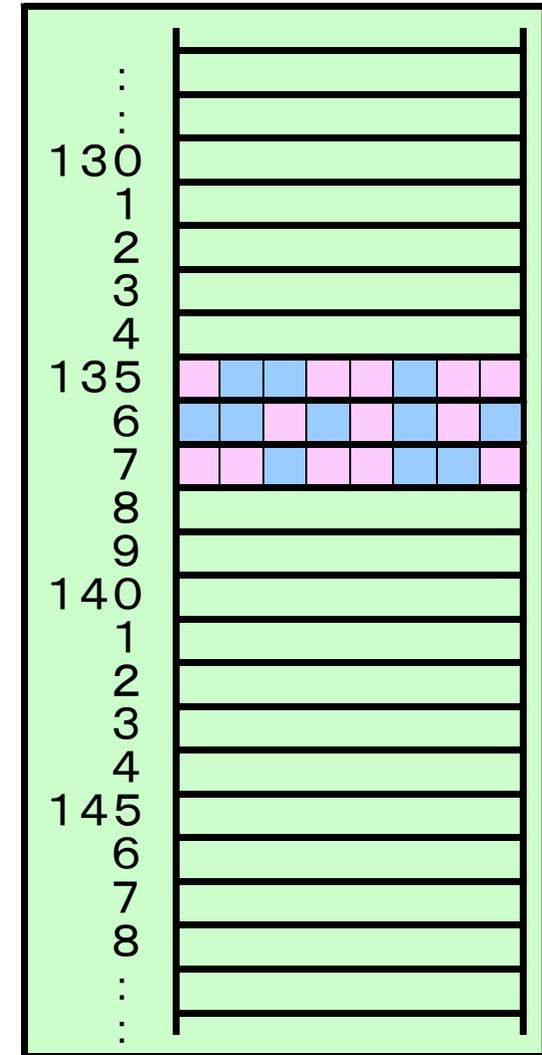
一般に通し番号

★ 番地を指定して

データを取得する場所

データを記憶させる場所

を特定する

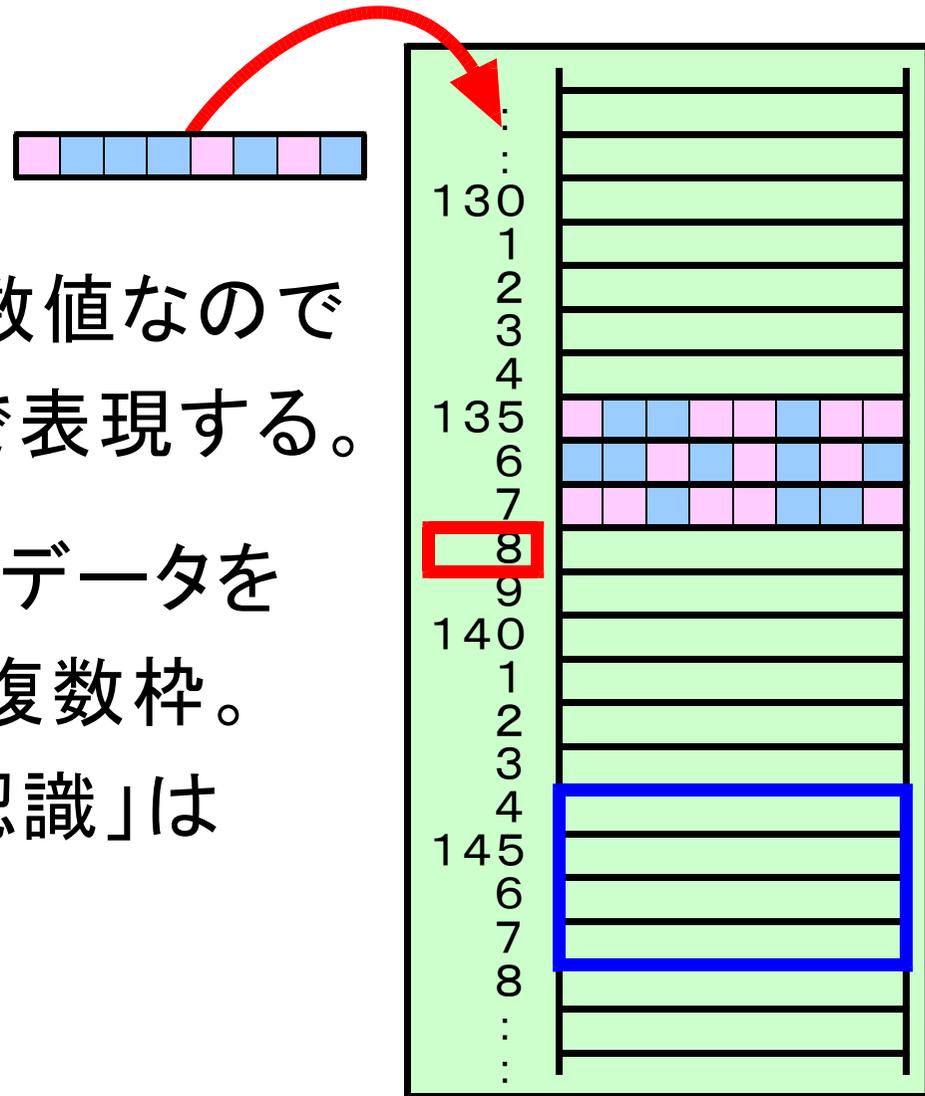


コンピュータのしくみ

○ 記憶部

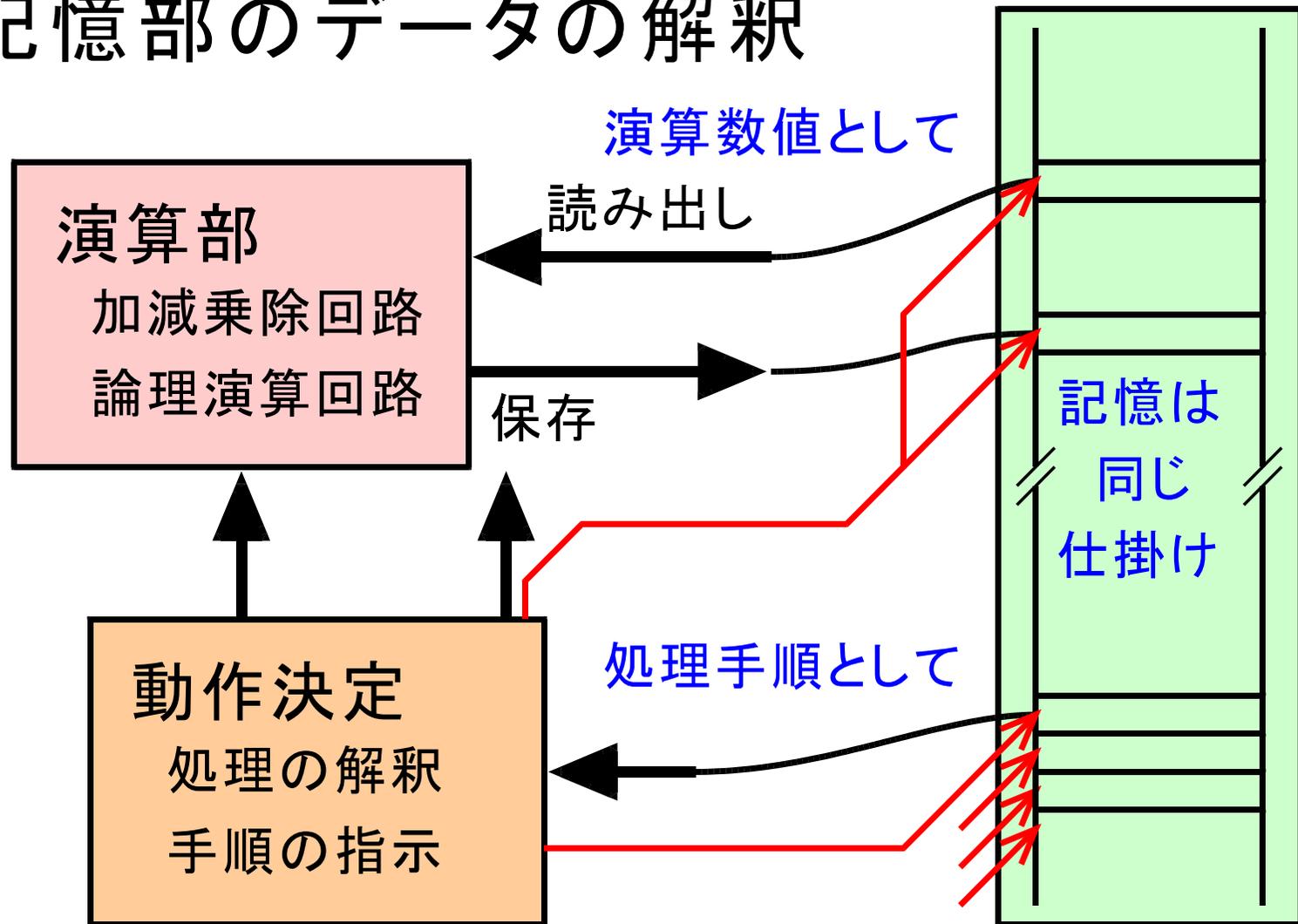
★補足

- ・ アドレス = 整数値なので必要なビットで表現する。
- ・ 16bit, 32bitのデータを入れるときは複数枠。ただし「枠の認識」は利用者側。



コンピュータのしくみ

○ 記憶部のデータの解釈



コンピュータの動作

○ 命令の実行

1: 次に実行する命令を記憶から取得

- ※どのアドレスかを示す値があって、順に増加
- ※取得されるものは、あくまで01のかたまり

2: 命令を解釈して、実行する

- ※解釈法はものによる

←その方法に応じて01を並べてプログラムする

- ・演算（+入力と出力の指定）
- ・記憶を別の記憶場所へコピー
- ・「次の命令の場所」を変える（ジャンプ、呼出）

コンピュータの動作

○ 命令の実行

1: 次に実行する命令を記憶から取得

2: 命令を解釈して、実行する

・「次の命令の場所」を変える (ジャンプ、呼出)

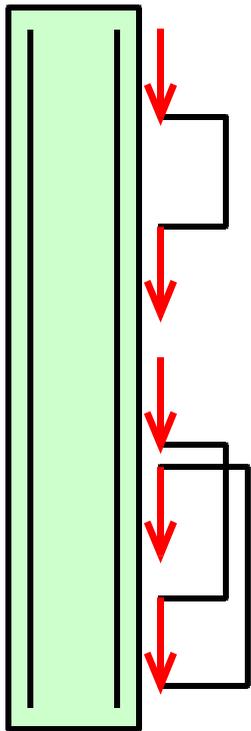
・ 通常は順番にアドレスが増える

・ ジャンプ(分岐、ブランチ)

命令の取得場所を変える

・ 呼び出し(コール)

一度変えた後、戻ってくる



コンピュータの動作

○ 条件判断

「もし～なら Aの処理」の仕組み

★直前の演算の結果の単純な判断

判断1: 直前の演算の結果がゼロ

判断2: 直前の演算で繰り下がり発生

およびこれらの組み合わせ

ならば分岐する、等

コンピュータの動作

○ 条件判断

判断1: 直前の演算の結果がゼロ

判断2: 直前の演算で繰り下がり発生

例: 二つの値で、 $A - B$ を計算する (結果は廃棄)

結果がゼロ → $A = B$ だった(等しい)

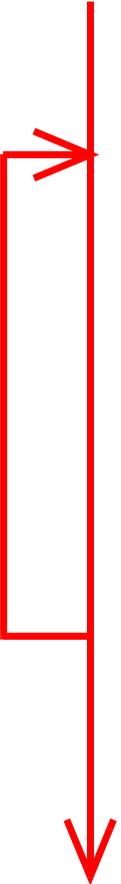
繰り下がり → $A < B$ だった

どちらでもない → $A > B$ だった

※ほぼこの形式、ただし実際には繰り上がりが使われる

コンピュータの動作

○ 動作の例：動作の繰り返し

- 1: 記憶場所Aに10を入れる
 - 2: 目的の処理をする
 - 3: Aの値を取り出して、1を引いて、結果をAの場所に戻す
 - 4: 直前の結果がゼロでなければ、2にジャンプ
 - 5: 次の処理 (4で2に行かなければ自然にここに)
- 

テーマ2:コンピュータの構造

○ 概要

★CPU、MPUとメモリ

演算部と記憶部の実際

★演算と記憶と入出力

計算以外のための仕掛け

★パソコンとマイコン

パソコン、マイコン、組込マイコン

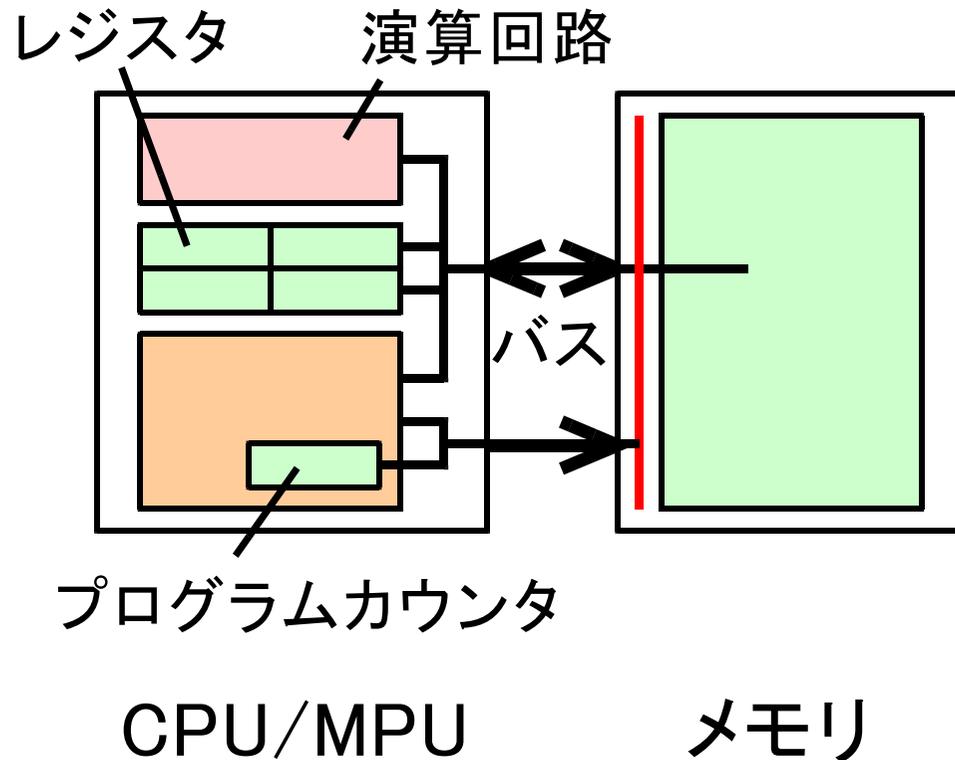
コンピュータの構造

演算部

記憶部

管理

○ CPU, MPU と メモリ



CPU, MPU:

少量の記憶を含む
コンピュータの中核

メモリ:

記憶そのもの
アドレスを指定して
データの出し入れを
する回路

コンピュータの構造

○ 用語

CPU: Central processing unit

MPU: Micro processing unit

レジスタ: 頻繁に使う値、途中経過などを置く小規模の
記憶回路 (類: 電卓の表示、そろばん)。

プログラムカウンタ: 「次の命令」のアドレス用。

メモリ: 記憶部の概念と同一の回路。

バス: 信号線の束。1対1ではなく多数で共有する。

アドレスバス: アドレスを指示する

データバス: CPU内外のデータのやりとりに使う

※レジスタ幅、バス幅が「CPUのビット数」の根拠

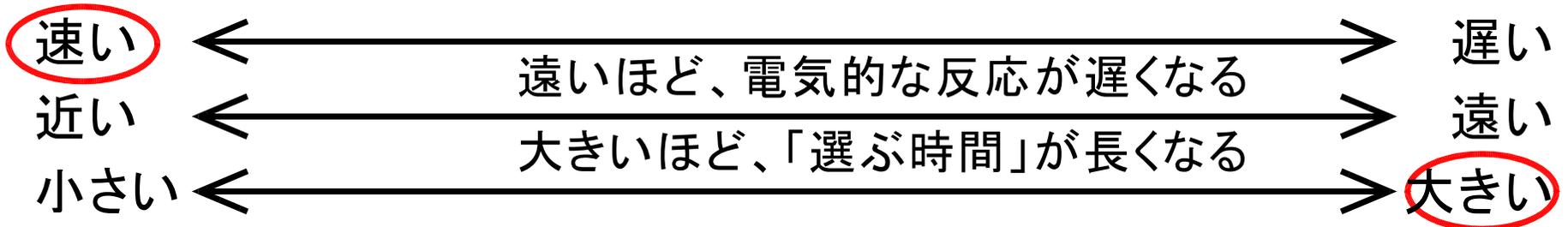
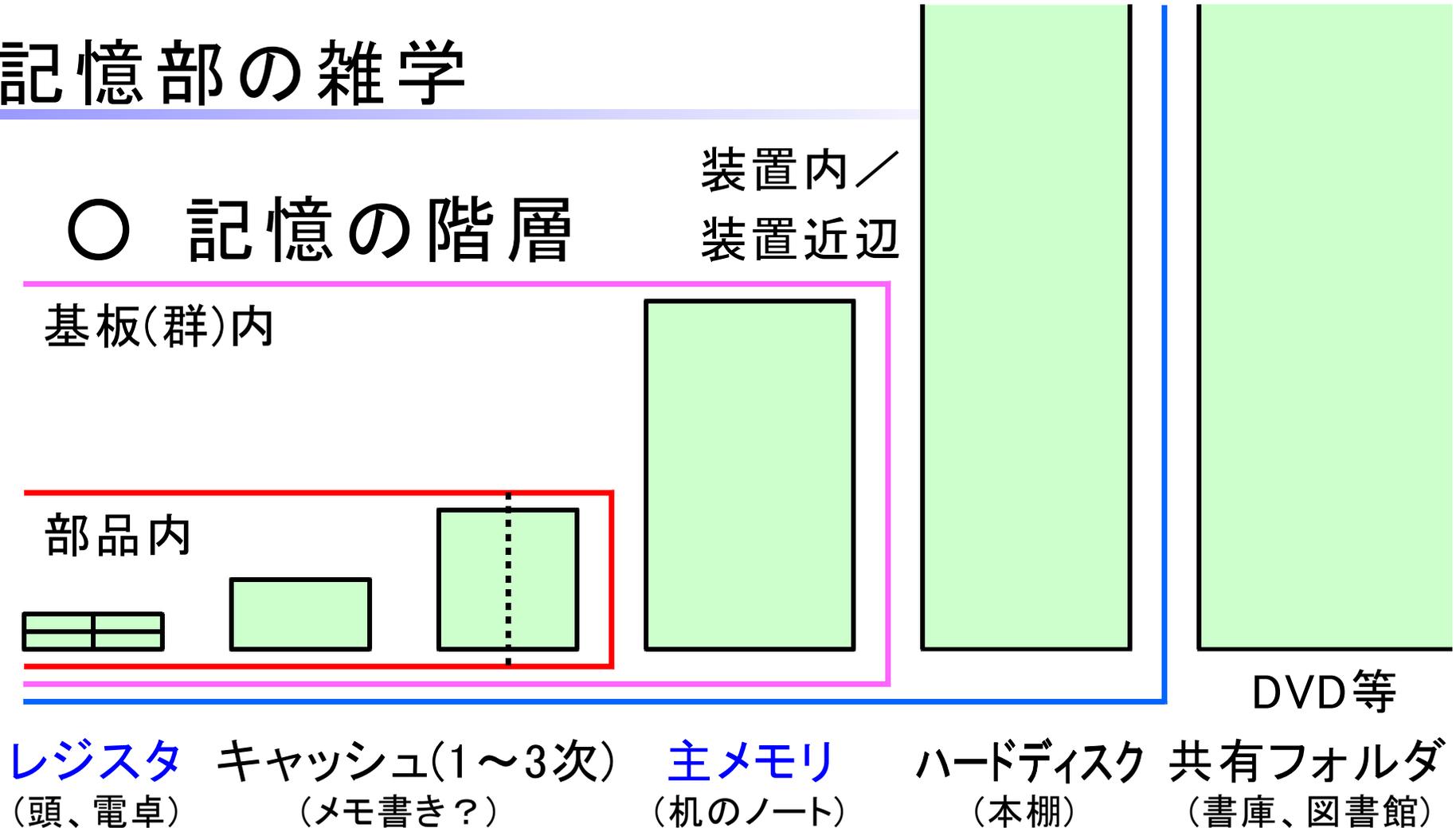
記憶部の雑学

○ メモリの種類

- **RAM** Random Access Memory
ほぼ瞬時に好きなところを読み書き可
- **ROM** Read Only Memory
(通常は)書けず、読むだけのメモリ
プログラムや固定データの記憶用
- PROM (Programmable) EPROM(Erasable PROM)
EEPROM(Electrically EPROM) 変更可能ROM
- フラッシュメモリ 消去法に特徴のあるEEPROM

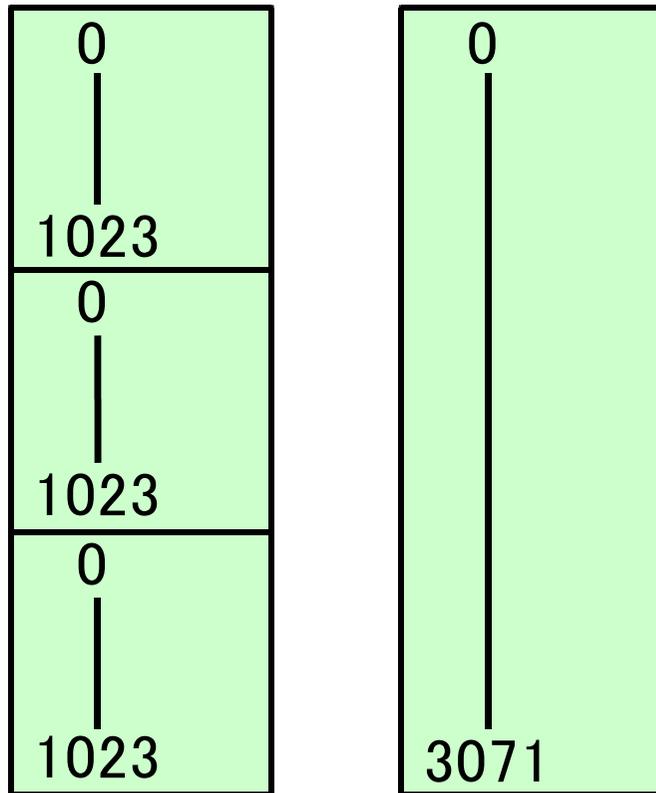
記憶部の雑学

○ 記憶の階層

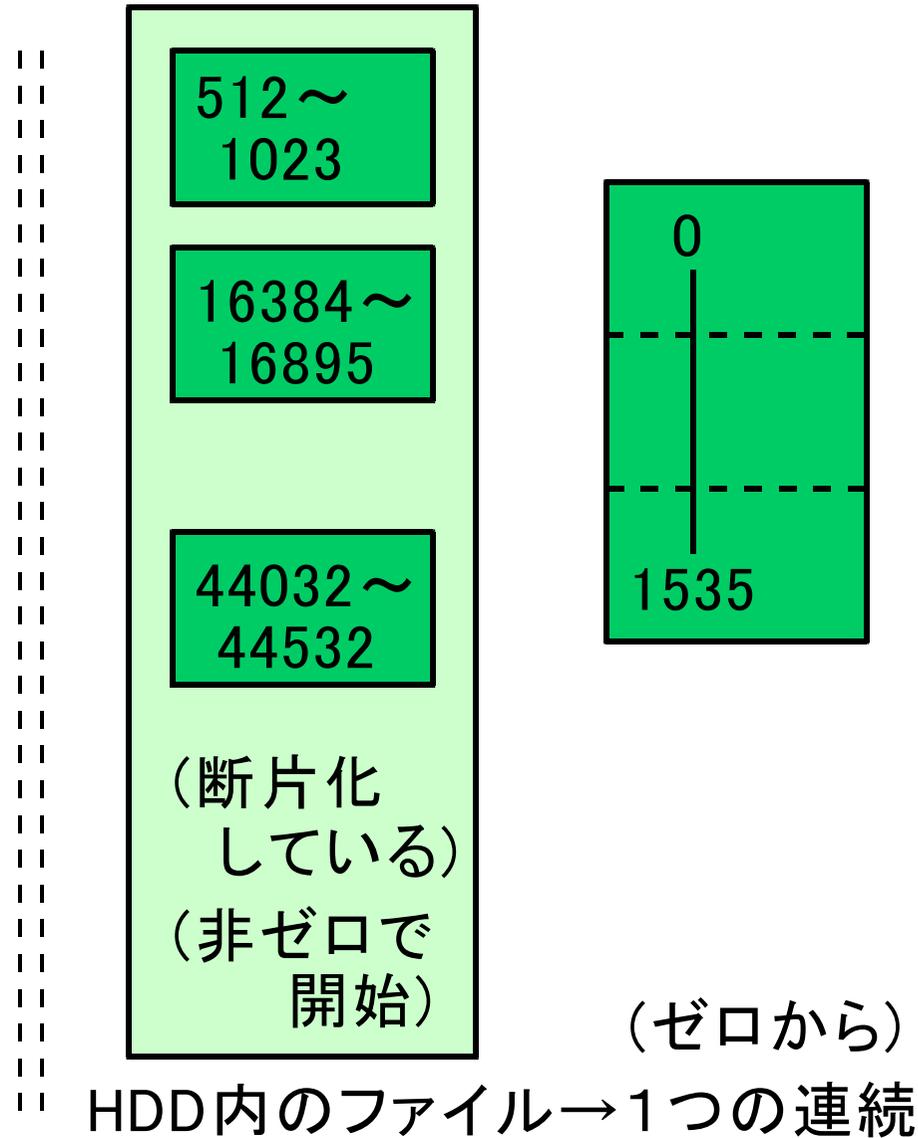


記憶部の雑学

○ 番地の別名



3個の部品 → 1つの連続



コンピュータの構造

○ 入出力部

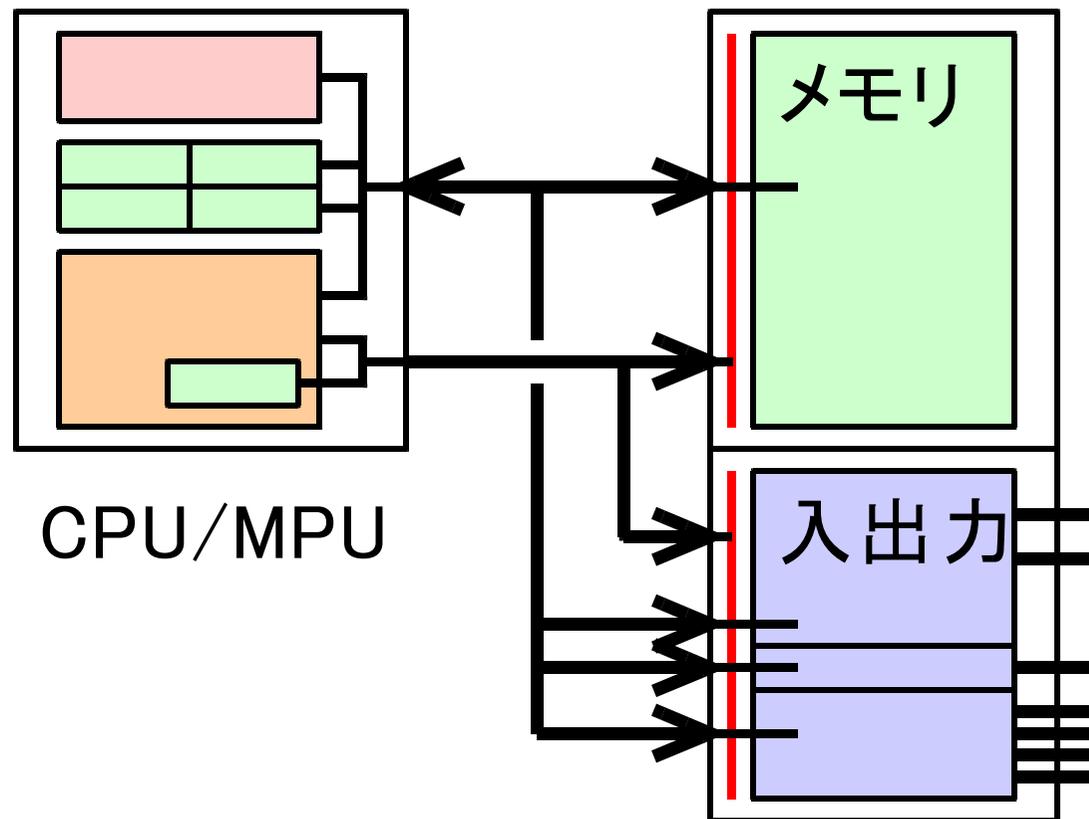
★CPUとメモリだけでは、役立たず
データの入力も、結果の出力もできない

★入出力用の機能が必要

- ・ キーボードやマウスなどの入力
※いまはまとめてUSBに
- ・ ディスプレイやプリンタなどの出力
- ・ ハードディスクなどの接続
- ・ ネットワーク通信

入出力機能

○ 入出力の方法 = メモリの置き換え



メモリ:

- ・書き→記憶
- ・読み←メモリ内データ

入出力:

- ・書き→出力
- ・読み←入力

入出力機能

○ 入出力機能の実現 = メモリの置き換え

★ なにか0/1を渡し、受け取れれば、あとは
入出力回路が好きなように加工する
= データの受け渡し方法

★ CPUはアドレスを指定して読み書き動作をしたときに、同じ電氣的動作をすれば、
なにが繋がってても気にしない
→ どのアドレスがなんの機能かは
開発者が管轄すること。

入出力機能

○ 入出力の例：

★ デジタル出力回路

- ・ あるアドレスに0/1のパターンを書き込むとLEDが光ったりブザーが鳴る。

★ 画面表示モニタ用回路

- ・ あるアドレスに数値を書き込むと、その数値に応じて画面に光点ができる。
- ・ 別のアドレスに数値を書き込むと、直線のデータを自動的に作ってくれる。

パソコンとマイコン

○ おおざっぱな区別

★パソコン Personal Computer, PC

- ・一般に汎用性重視で大型。
- ・CPUとメモリと入出力回路用の部品で構成。
- ・装置、箱（、基板）

★マイコン Micro Computer, Micro Controller, MCU

- ・一般に目的重視で小型、省電力指向。
- ・組込マイコン / 汎用マイコン



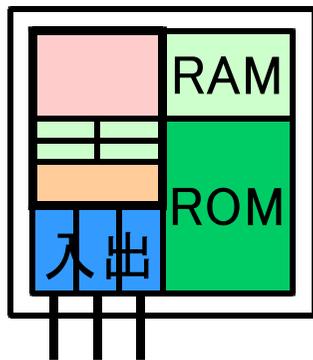
←熊研標準

※「自称」で区別という説もあり

パソコンとマイコン

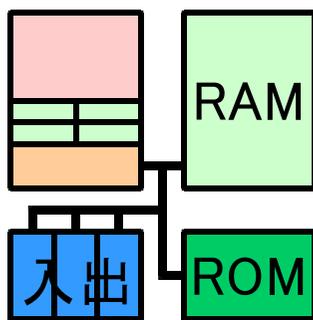
○ おおざっぱな区別 (区分はそこそこ曖昧)

★組込マイコン



- ・一つの集積回路(LSI)に、CPU、メモリ、特定の入出力回路を**全部入れた部品**。1個で目的を達成することが前提。

★汎用マイコン



- ・パソコンと似た**複数部品構成**のCPU部分。
- ・パソコンほどのパワーは不要ながら、ある程度汎用な設計にしたい場合など用。

テーマ3：組込マイコン

○ 概要

★組込マイコンの用途、実例

★組込マイコンに求められる機能
入出力の種類

★組込マイコンの選定

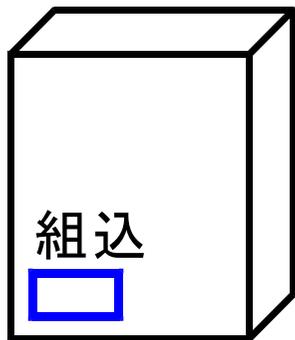
★組込マイコンとOS

★とりあえず触ってみたい、という場合は？

組込マイコン?

○ そもそも、組込(くみこみ)とは？

★何らかの装置に組み込まれて、その機能を担うような コンピュータ / ソフトウェア



- ・組込マイコン = 機能部品としてのマイコン
- ・組込パソコン = 制御装置化したパソコン
- ・組込ソフト = そういう環境でのソフト

※広い意味での組込ソフトには、携帯電話でのアプリも含まれるが、「ハードがらみに限る」という見解もあり、要注意。

組込マイコンの用途

○ ハードに密着した、なにか

★装置の中央制御部品として

家電

小型ロボット

(大型はPC内蔵のこともあり)

シーケンサ

(動作を担う部分)

プリンタ

(受信、描画、機械制御)

ネットワークカメラ

(画像処理、通信)

携帯電話

(汎用性重視＋特有機能)

組込マイコンの用途

○ ハードに密着した、なにか

★別のコンピュータの**下請け**として

例:

操作パネルだけを担当するマイコン

モータ制御を担当するマイコン

(モータ制御器としてみたら中央制御係)

★**高度な信号のやりとり**解釈用

例:

USBマウス、キーボード内蔵マイコン

組込マイコンの用途

○ 機能部品としての、なにか

★コンピュータを意識させない部品だけど
実は入っている場合

ありそうな例：テレビのリモコン用IC

※学習リモコンはマイコン製品

★デジタル回路の部品として

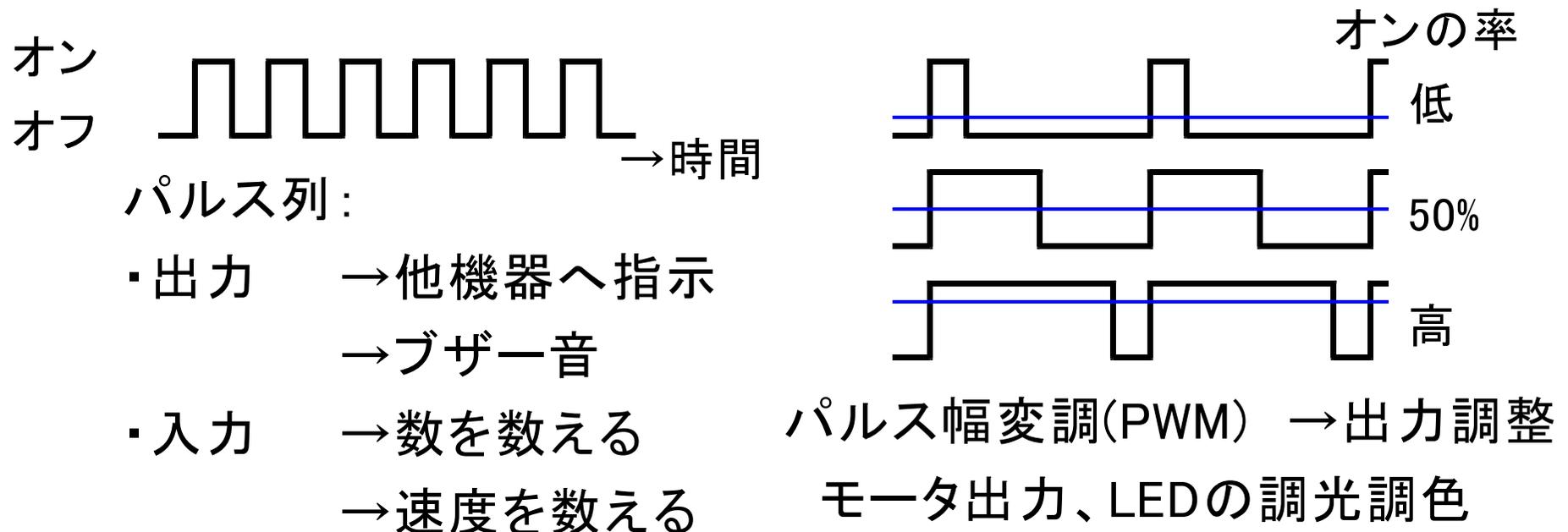
回路を設計するよりプログラムで
作った方が楽な場合

実務に必要な機能

○ ハードとの入出力

★ デジタルな入出力

- ・ ただの**オンオフ** (LEDの点灯、スイッチの検知)
- ・ タイミングの重要な**オンオフ変化**



実務に必要な機能

○ ハードとの入出力

★アナログ電圧の入力

- ・センサからの信号の入力

★アナログ電圧の出力

- ・スピーカーからの音再生、出力調整

→ 今は廃れつつある、デジタル(PWM)で代用

※回路構成上、アナログは効率が悪いいため

実務に必要な機能

○ ハード以外を指向した機能

★通信機能

- ・シリアルポート(調歩同期通信)、USB
- ・I²C(センサ等)、CAN(車内ネット等)、Ethernet

★マイコン自身の電源管理

休眠省電力と外部入力での復帰など

★特殊入出力

- ・メモリカード(SDカードなど)の読み書き
- ・液晶ディスプレイ(表示機能)

組込マイコンの種類

例: Renesas SHシリーズ

Selected Filters
 ファミリ: [SuperH] [解除]

表示件数: 20

534 件見つかりました。(1 - 20 件目を表示しています。)

1 2 3 4 5 6 7 8 9 10 .. [次へ >](#)

展開 | 折りたたむ | [すべて解除](#)

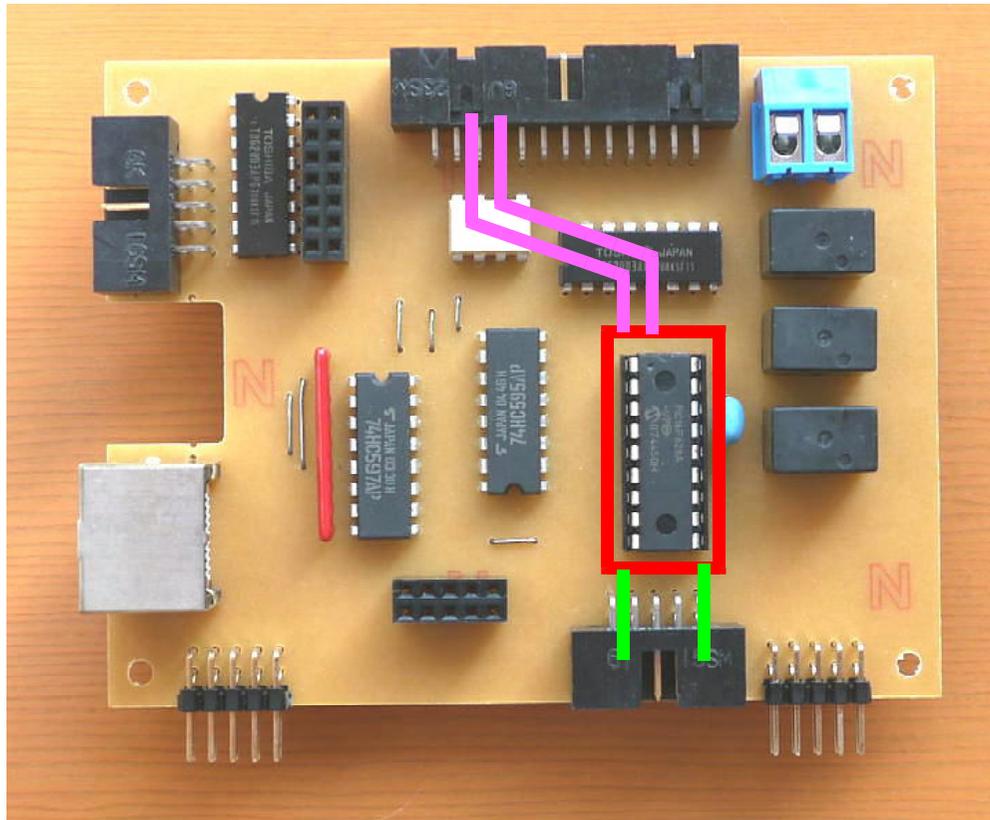
- Bit Size (534)
- ファミリ [解除]
- シリーズ (534)
- ROM (522)
- RAM (752)
- ROM展開 (534)
- 最小命令実行時間 (ns) (726)
- 動作周波数 / 電源電圧 (1213)
- 生産ステータス (534)
- データフラッシュ / E²データフラッシュ (80)
- キャッシュメモリ (323)
- DMAC

マイクロコンピュータ

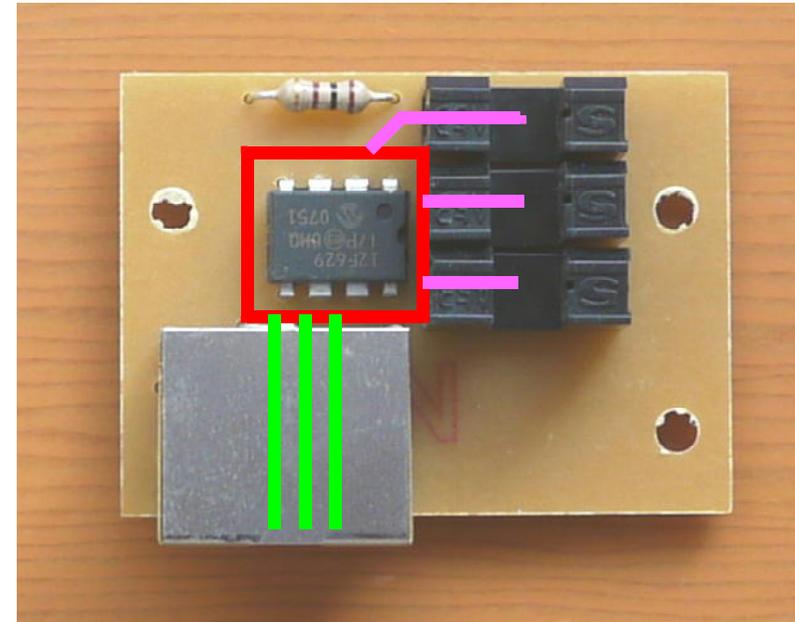
Part No.	ROM	RAM	A/D コンバータ 分解能 × チャンネル	32ビット タイマ	シリアル インタフェース クロック同期 /非同期兼用
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> R8A77240D500BG	-	16K	10ビット x 4	6(TMU) 1(CMT)	3(SCIF x 3) 3(SCIFA x 3)
<input type="checkbox"/> R8A77240B500BB	-	16K	10ビット x 4	6(TMU) 1(CMT)	3(SCIF x 3) 3(SCIFA x 3)
<input type="checkbox"/> R8A77860HBG	-	(8K + 16K) x 2	-	12(TMU)	6(max)
<input type="checkbox"/> R8A77860NBG	-	(8K + 16K) x 2	-	12(TMU)	6(max)
<input type="checkbox"/> R8A77850BADBG	-	8K(高速) + 16K(高速) + 128K(中速)	-	6(TMU)	6(Max.)
<input type="checkbox"/> R8A77850BANBG	-	8K(高速) + 16K(高速) + 128K(中速)	-	6(TMU)	6(Max.)

組込マイコンの種類

熊谷研の実例



モータ信号のカウント用

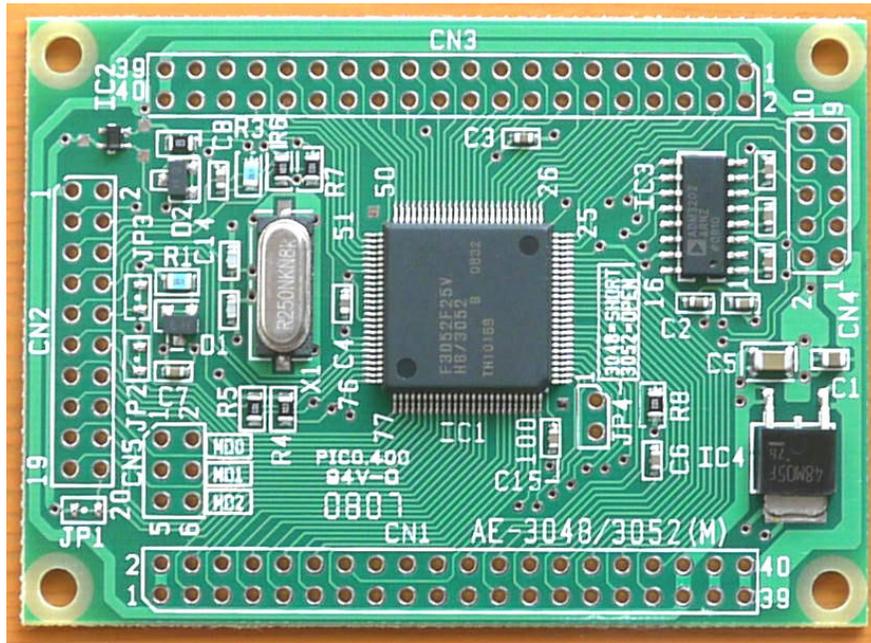


3個の光センサの
判断と状況出力

いずれもMicrochip社製PICマイコン: デジタル部品として

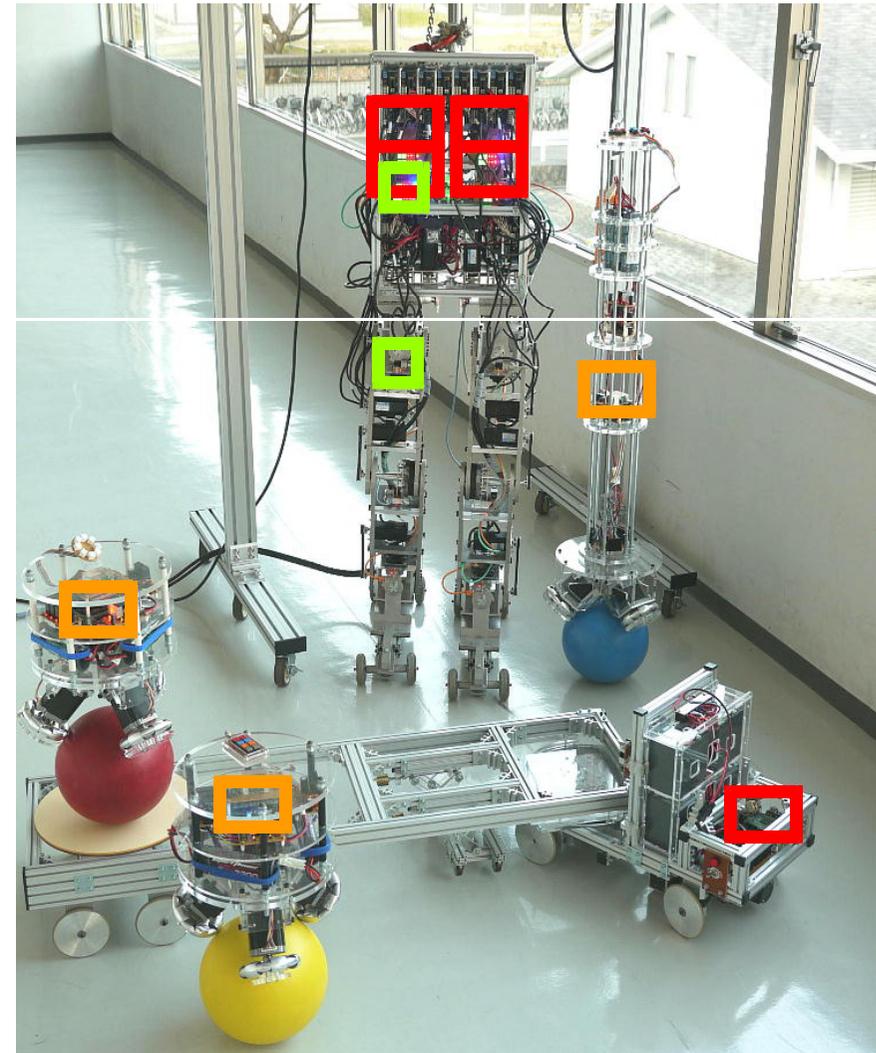
組込マイコンの種類

熊谷研の実例



基板搭載済みマイコン

- ロボットの中央制御 □
- ロボットの現場制御 □
- ※さっきの回路 □



Renesas H8/3052, 秋月電子 AKI H8/3052ボード

組込マイコンの種類

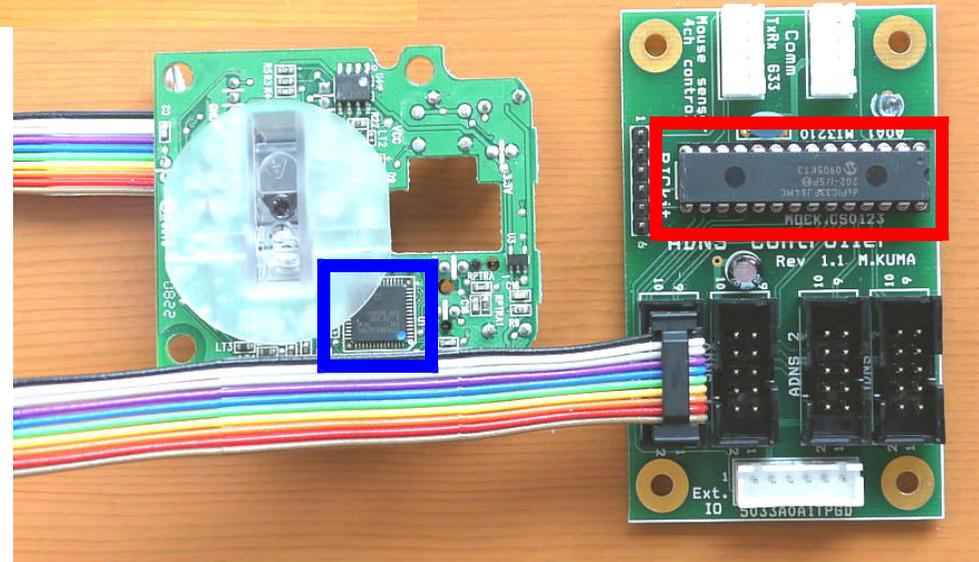
熊谷研の実例



マウスのセンサを
流用する回路

3相ベクトルインバータ

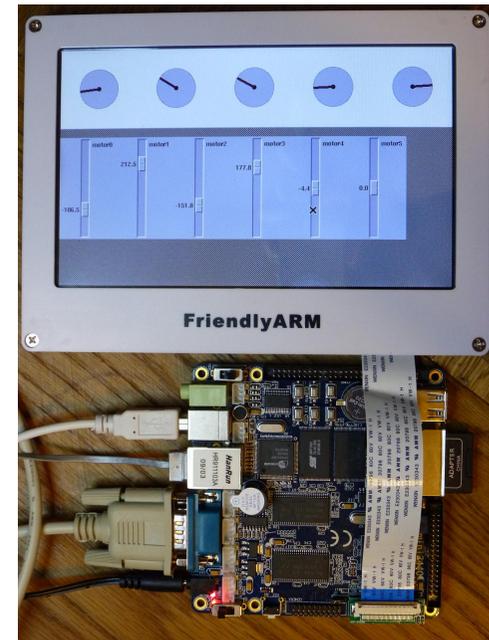
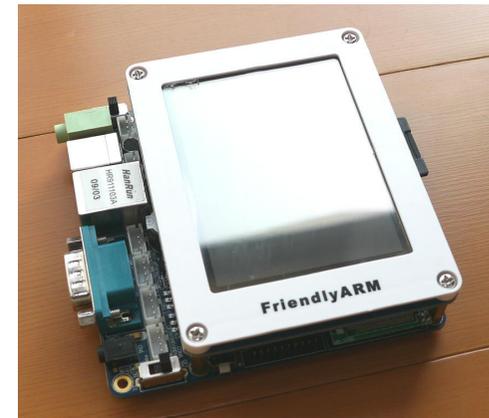
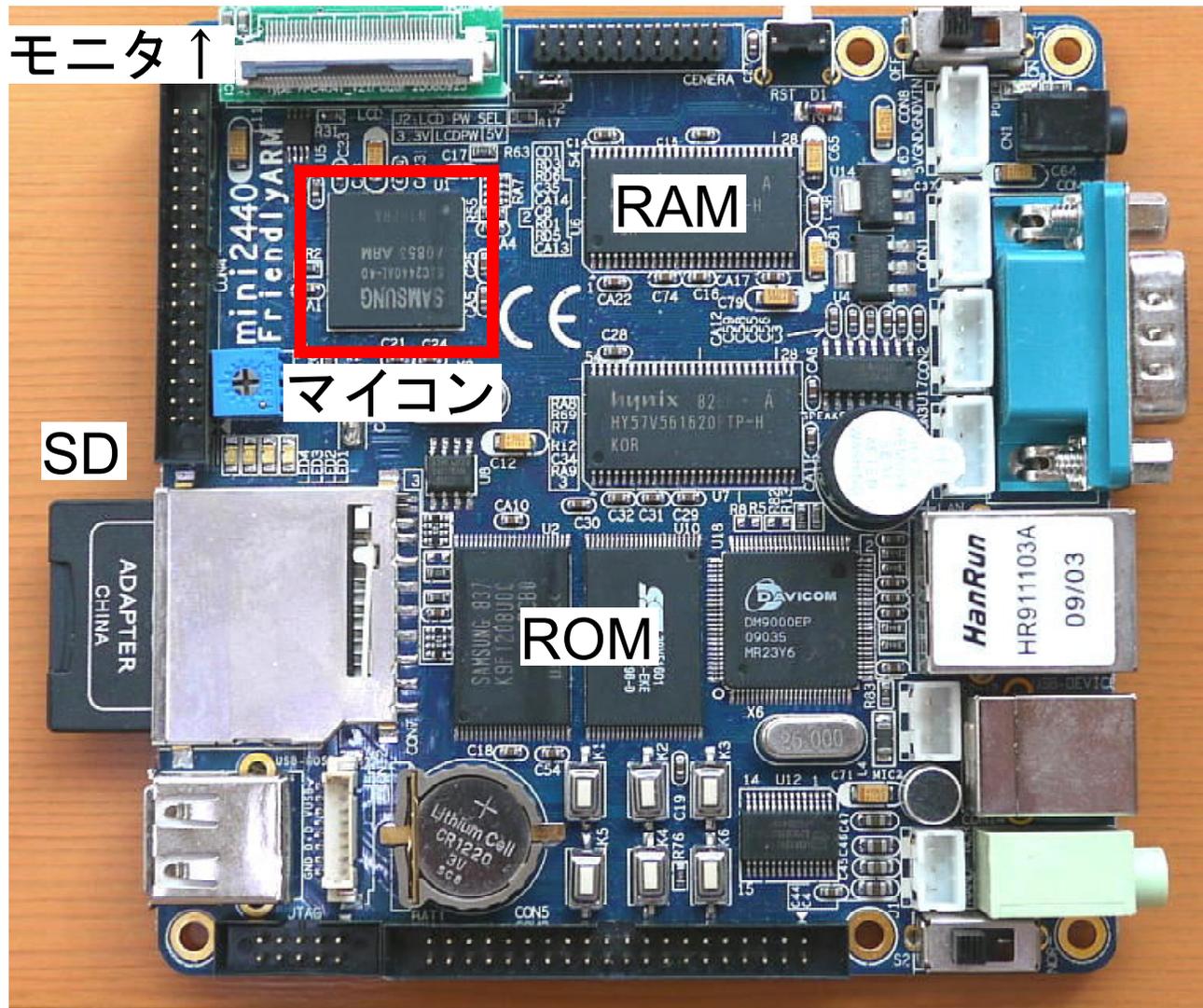
PWM x6
アナログ入力 x6



Microchip製dsPICマイコン: モータ制御用

組込マイコンの種類

熊谷研の実例



携帯電話用ARMマイコン+Linux, Friendly ARM製

組込マイコンの選定

○ 量産目的で無ければ

★性能

十分な**処理能力**、速度、メモリの大きさ、
必要な**入出力**

★入手性

1個単位で変えるかどうか（ネット通販など）

※ 組込マイコンは品種が多い一方で、
「実は1000個単位発注」というケースが多い。
あくまで、量産品への組込が前提。

組込マイコンの選定

○ 量産目的で無ければ

★ 開発のしやすさ

- ・ 基板で売られているかどうか？

マイコン基板、マイコンユニット、評価基板

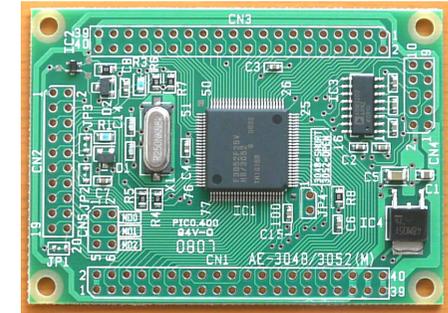
- ・ 開発に特殊なものは必要か？

開発ソフト（有償、無償、対応OS）

マイコンへのプログラム書き込み手段

（専用回路、専用装置、専用ソフト）

※最近は特殊なものは不要 or 安価



組込マイコンの選定

○ 量産目的で無ければ

★開発のしやすさ

- ・基板で売られているかどうか？
- ・開発に特殊なものは必要か？
- ・世の中/周囲でよく使われているか？

＝解説の多さ、聞ける場所

※マニュアルを見るだけで、短時間で理解することは容易ではない。

組込マイコンの選定

○ 量産目的なら （数千個～）

★コスト ただし装置全体を見たトータルコスト

- ・ぎりぎりの処理力
- ・ぎりぎりのメモリ
- ・ぎりぎりの入出力 （注：削りすぎると逆効果）

★開発の容易さ

- ・同じシリーズの上位に開発しやすい品種があること(試作は前述と同じ)。

組込マイコンとOS（オペレーティングシステム）

○ マイコンにOSを搭載する意義

★能力の分配、複数処理の同時実行

- ・時分割でCPUを複数プログラムに分配。
- ・メモリの管理（割り当て、回収）

★デバイスドライバを介した入出力の簡素化

- ・違うハードをプログラムに同じように見せる

例：Linuxで開発したネット通信プログラムは、PCとLinux搭載マイコンで、同じ書き方で動作する。

組込マイコンとOS

○ マイコンとOS選定

- 1: 無理にOSを搭載する必要はない
 - ・ OSを使うと、OS分だけ余分にマイコンが働く＝メモリや演算力を取られる。
 - ・ OS無しで限界性能を引き出せる。

- 2: 高機能なマイコンで、OSとドライバがあるなら、OSを使った方がずっと楽。
 - ・ 自分で全てを用意することは困難。

組込マイコンとOS

○ マイコンとOS選定

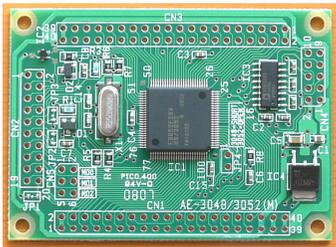
- 3: 小さいマイコンでも、簡易的なOSを導入すると、機能のライブラリとして開発が簡単になることがある。
(特に通信、表示機能など)

※Linuxが使えるから、Androidが使えるから、とOS目当てでのマイコン選定もあり。

マイコンをとりあえず触ってみたい

○ 回路設計経験はあるけどソフト未経験

- ・ マイコン開発を始めるという点では、回路設計経験はあまり役立たない。
(量産設計などの時には非常に心強い)



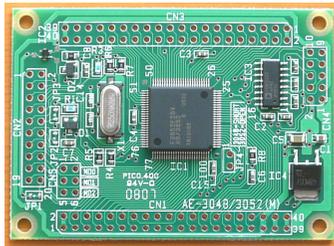
- ・ 初心者向け: [Arduino系\(*\)](#)
- ・ 誰でもできる系: [H8/3052\(※\)](#), [PIC系](#)
- ・ 解説本、WEBサイトなどでお試し

* いま巷でかなり話題 (熊谷は未経験)

※秋月電子 AKI-H8/3052基板

マイコンをとりあえず触ってみたい

○ ソフト開発経験あるけど回路不慣れ



- ・動くマイコンさえあれば、大きく変わらず。
- ・入門向けマイコン基板(Arduino, 秋月H8)などを買ってきて、とりあえずソフトを書いてみる。



- ・Linux, Android搭載の32bitマイコン (ARM系など)を買ってきて、「Linux PC / Linuxサーバの小さいの」と思って触ってみる。

まとめ

- ★ コンピュータの動作の仕組み
 - ・ 演算と記憶
- ★ コンピュータの構造
 - ・ CPU と メモリ と 入出力回路
- ★ 組込マイコン
 - ・ マイコン=小さいコンピュータ
 - ・ 組込=装置の一部品として機能する
 - ・ パソコンと異なり、様々な入出力機能やメモリなどを1個の部品に内蔵している。
 - ・ 規模も機能も多種多様。

参考情報

ロボット博士の基礎からのメカトロニクスセミナー

今後の予定：

3月：デジタルの基礎

4月：アナログ信号の基礎

5月：アナログ信号のコンピュータへの取り込み

参考情報

ロボット開発工学研究室

<http://www.mech.tohoku-gakuin.ac.jp/rde/>

→ 講義情報

- ・「メカトロニクスI, II」

主にメカトロに必要なた電子回路系基礎

- ・「ロボット基礎」

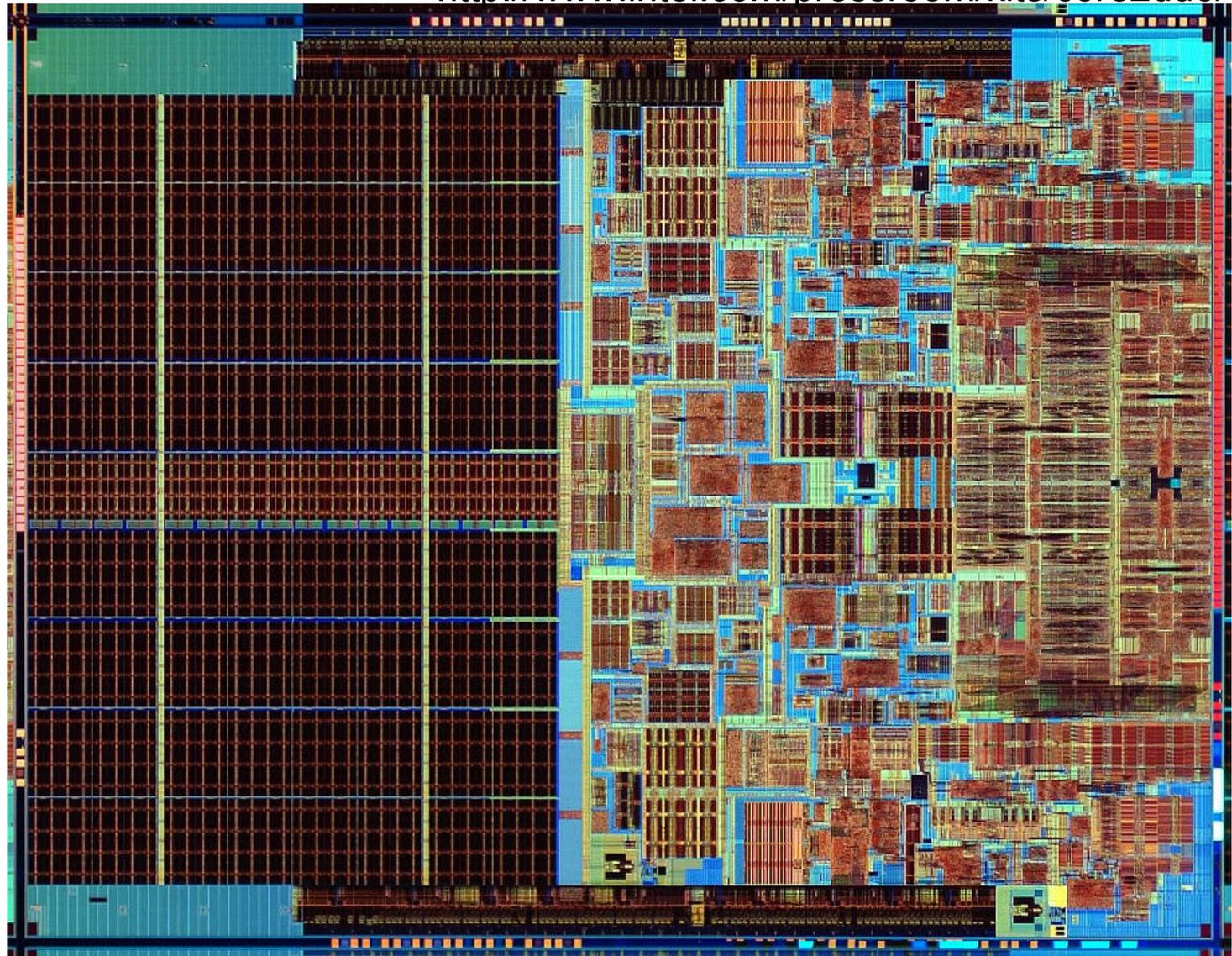
ロボットとされるものに関する基礎理論

- ・「ロボット開発工学」(まだ工事中)

メカトロニクス総合

CPU の構造の例 (Intel Core2Duo, Intelサイトより)

<http://www.intel.com/pressroom/kits/core2duo/>



一般に碁盤の目状に整然と並んでい
るところは記憶。ごちゃごちゃの所は
演算や動作制御部。
左半分はキャッシュ。