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

C13/Rev 1.0

第１３回

# デジタルセンサを マイコンにつなぐ

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

## ○ デジタルセンサの概要と具体例

テーマ１：デジタルセンサ

- デジタルセンサとアナログセンサ
- デジタルセンサゆえの特徴

テーマ２：デジタルセンサのハードウエア

- マイコンとデジタルセンサの接続

テーマ３：デジタルセンサのソフトウエア

- 最低限のセンサ読み書き
- センサの情報読み取りと処理

# イントロダクション

## ○ センサによる測定 (C12より)

◇測定対象の情報を取得、活用

- センサで電気的変化にする。
- 電気的変化を電圧変化にする。
- 適切な電圧に増幅しフィルタをかける。
- アナログデジタル(AD)変換器でデジタル化。
- 適切な処理で情報に変換。

# イントロダクション

## ○ デジタルセンサとアナログセンサ

◇世の中：アナログ　／　処理：デジタル

・どこかで、デジタル化

◇デジタル化の場所

・マイコン{内蔵/接続}のAD変換器（C12）

・センサ内部のAD変換器

# イントロダクション

## ○ デジタルセンサとアナログセンサ

### ◇デジタルセンサの利点

- アナログ部の設計、製作が不要
  ＝ マイコンに数本のデジタル配線のみ。
- 計測条件(分解能、速度など)の切り替え可能。
- ある程度の信号処理済＝自前処理低減。
- ある程度のキャリブレーション済み。

# イントロダクション

## ○ デジタルセンサとアナログセンサ

### ◇デジタルセンサの欠点

- アナログ部の信号が見えない
  ＝ ハードの動作が全く見えない
- デジタル通信を理解しないと使えない
- 設定しないと動きすらしない

# イントロダクション

## ○ デジタルセンサ

### ◇使うべき理由

- 計測性能の向上（劣化の心配が少ない）
- 低コスト化（アナログ回路は高い）

### ◇避けるべき理由

- ちゃんと動かすまでの苦労
- ハード屋の比率＜＜ソフト屋の比率
- アナログセンサの開発からはギャップ大。
  ソフト屋にとっては朗報。

# デジタルセンサの概要

## ○ デジタルセンサの形態

### ◇最初から原理的にデジタル

- スイッチ類
- ロータリーエンコーダ

### ◇出力が2値（on/off）

- 各種産業用スイッチ
  ＝ 何らかの処理をして接点出力
- 近接スイッチ ～ 画像判断装置

# デジタルセンサの概要

## ○ デジタルセンサの形態

### ◇AD内蔵　単純型

- 調整箇所、初期化作業なし　（クロック程度）
- デジタルマイクICなど

※ADではない、パルス幅出力型あり

### ◇AD内蔵　デジタル処理あり

- 初期化作業必要、調整可能
- デジタルセンサで最近よく？見かける
- 加速度、ジャイロセンサなど

# デジタルセンサの概要

## ○ デジタルセンサの形態

### ◇AD内蔵　処理プロセッサ内蔵

- 内部にマイコン等を持ち、信号処理も行う。
- 光学マウスのセンサ、一部の姿勢センサ等

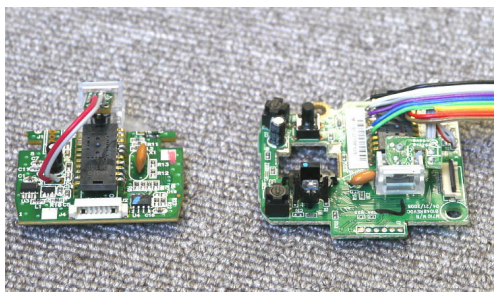
### ◇センシング処理装置

- 基板や箱としてのセンサ、計測装置
- GPS、レーザーレンジファインダ他

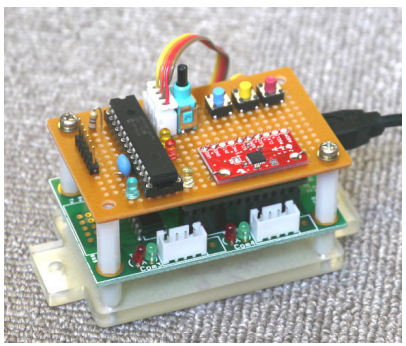
# デジタルセンサの概要

## ○ デジタルセンサの例

### ◇今日の話題に登場するセンサ



光学マウスセンサ
ADNS-6010
処理プロセッサ内蔵型
SPI接続

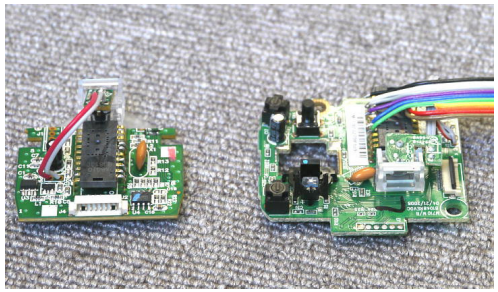


6軸姿勢センサ
MPU-6050
AD＋処理型
I2C接続


6軸力覚センサ
WDF-6A100-2
処理プロセッサ内蔵
USB(シリアル)接続

# デジタルセンサの概要

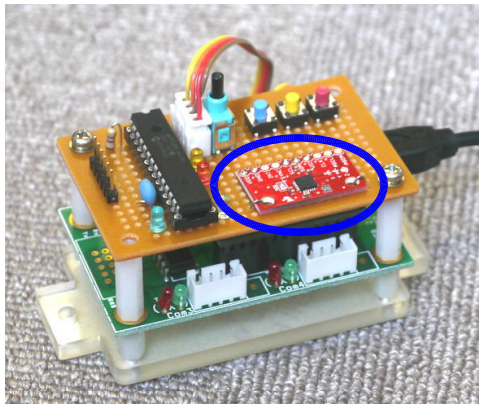
## ○ デジタルセンサの例



### ◇マウスセンサの概要

- レーザ光学式マウスのセンサ
- 内部に30x30=900画素の画像センサを持ち、内蔵プロセッサが画像の移動を計測して、マウスの移動を測定。
- 座標値を読み出すたびに、前回からの移動量を内部の単位で出力。
- 0.01mm程度，1m/s程度の測定可能。

# デジタルセンサの概要

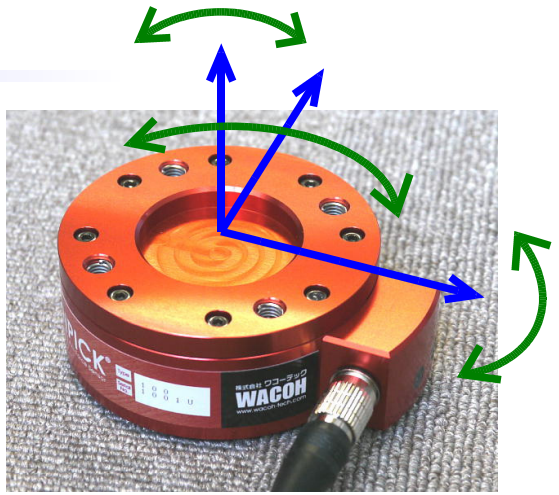


## ○ デジタルセンサの例

### ◇6軸姿勢センサの概要

- 加速度3軸＋角速度ジャイロ3軸
- スマホ、ゲームコントローラなどにも使われる、移動と回転の計測用。
- 傾きだけの計測には便利。(玉乗りロボ等)
- 絶対的な位置計測には向かない。
- 地磁気センサを外付けして、内部の処理機能で姿勢計算までできる(らしい)。

# デジタルセンサの概要




## ○ デジタルセンサの例

### ◇6軸力覚センサの概要

- 力3軸＋トルク3軸
- 様々な力の計測や、ロボットの手先、足のフィードバック制御などに用いられる。
- 力がかかると全体的に微小な変形→内部の複数の変形計測センサ→数学的処理で6軸の計測値。
- この製品は処理済みの測定値を出力。

# デジタルセンサの概要

## ○ デジタルセンサゆえの必須要素

### ◇デジタル通信が必要

- シリアル型の通信（→C11）
- SPI型、I2C、1-wire 他
- シリアルポート型（RS232，RS422等）
- USB接続（専用/シリアルポート変換型）
- Ethernet接続

※そのほか特殊なパルス列など

# デジタルセンサの概要

## ○ デジタルセンサゆえの必須要素

◇初期設定、パラメータ設定が必要

・電源を入れただけでは期待通り動かず

・ほぼ何もしなくとも動作

・パラメータ設定→開始指示

・処理ソフトのダウンロード必要

など様々

# デジタルセンサの傾向と対策

## ○ マイコン直結型（SPI，I2C）

### ◇接続

・マイコンの専用/汎用入出力端子に
２～４本の信号線を接続。

### ◇データの送受

・マイコンの積極的な要求による。
（クロック、アドレス指定など）

# デジタルセンサの傾向と対策

## ○ シリアル通信型（RS232/422，USB，ネット）

### ◇接続

- 汎用の通信インタフェースに接続

  ※パソコンクラスが多いがマイコンも可

### ◇データの送受

- ソフト的バイト列の送受

  テキスト形式/バイナリ形式のデータ

- 「垂れ流し型」が多い（設定後は受け身）

# 今回の目的

## ○ デジタルセンサの概要と具体例

テーマ１：デジタルセンサ

- デジタルセンサとアナログセンサ
- デジタルセンサゆえの特徴

テーマ２：デジタルセンサのハードウエア

- マイコンとデジタルセンサの接続

テーマ３：デジタルセンサのソフトウエア

- 最低限のセンサ読み書き
- センサの情報読み取りと処理

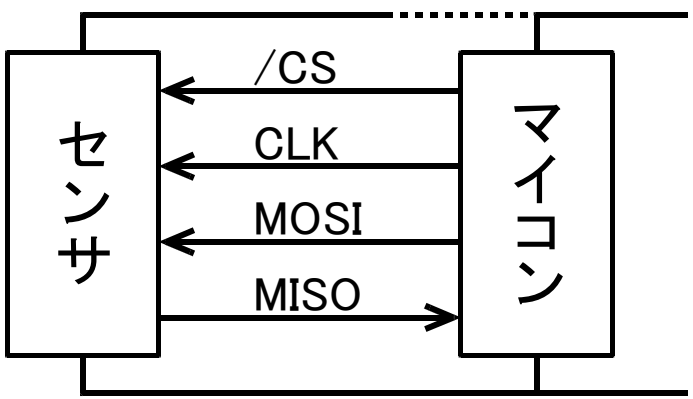
# デジタルセンサのハードウエア

## ○ マイコン直結型の場合

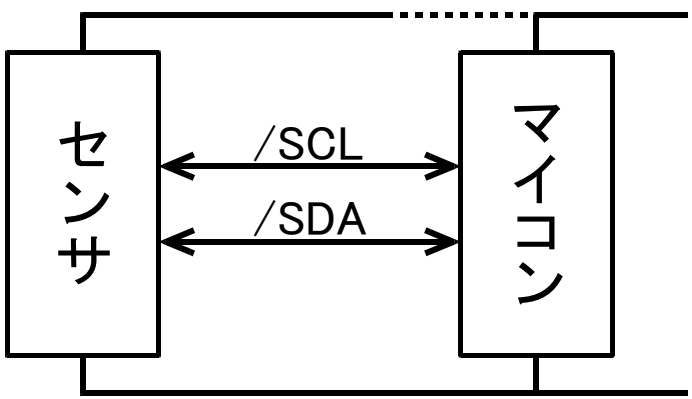
### ◇キーポイント

- 質の良い電源を供給(内部はアナログ)。
- 所定の信号線をマイコンと接続。

後述




SPI型の典型例




I2Cの典型例

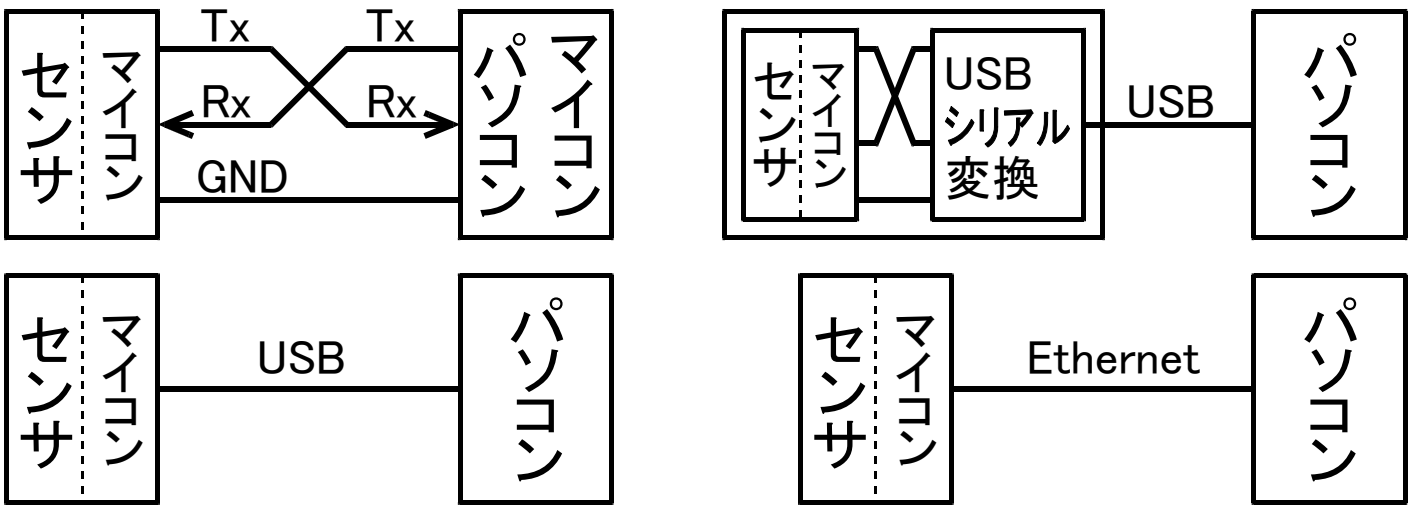
# デジタルセンサのハードウエア

## ○ シリアル通信型の場合

### ◇キーポイント

・コンピュータ間通信用の汎用手段を使用。

・信号の電圧振幅に注意。（3.3Vシリアルなど）



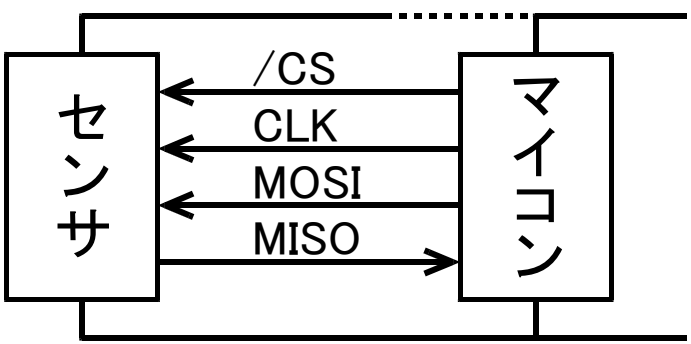
# デジタルセンサの接続

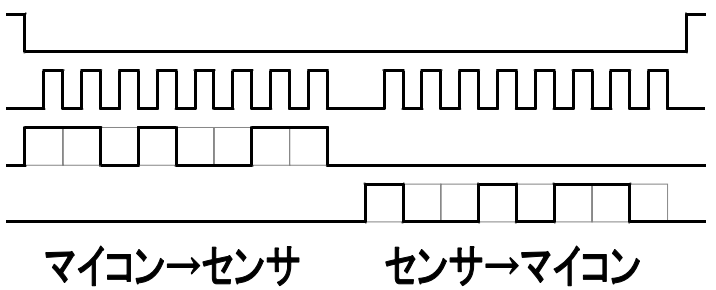
## ○ SPI型の接続

- 汎用のデジタル入出力端子4本で接続。
- 上りと下りが明確＝動作確認容易。
- 複数のセンサを使う場合CS以外は共用可。

※CS: chip select,　　MOSI: Master out Slave in

※CLK: clock,　　MISO: Master in Slave out



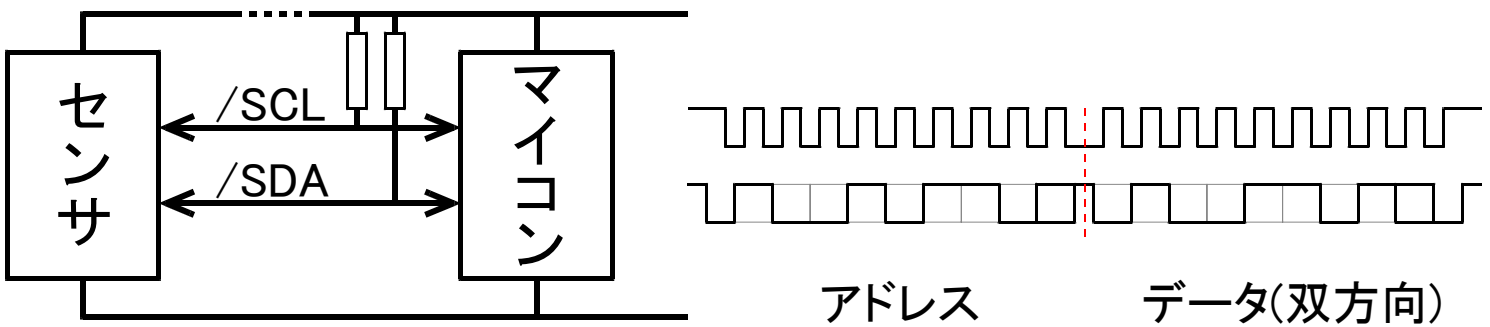



# デジタルセンサの接続

## ○ I2Cの接続

- 専用の入出力端子に接続
  ※汎用端子でもできないことはない
- データは上り下り共用＝確認時に留意。
- 複数のセンサを並列に接続可(アドレス有)。

※SCL: クロック,　SDA: データ



# 今回の目的

## ○ デジタルセンサの概要と具体例

テーマ１：デジタルセンサ

- デジタルセンサとアナログセンサ
- デジタルセンサゆえの特徴

テーマ２：デジタルセンサのハードウエア

- マイコンとデジタルセンサの接続

テーマ３：デジタルセンサのソフトウエア

- 最低限のセンサ読み書き
- センサの情報読み取りと処理

# デジタルセンサのソフトウエア

## ○ この先の流れ

### ◇一般的なセンサインターフェイスの解説

- I2CやSPI型のセンサをマイコンに繋ぐ方法
- これらのセンサの典型的？パターン

### ◇具体的すぎる例

- ADNS6010　　レーザーマウスセンサ
- MPU-6050　　６軸IMUセンサ
  ※加速度３軸＋ジャイロ３軸
- 具体例を通した「こういうモノあり」紹介。

# デジタルセンサのソフトウエア

## ○ ソフトウエアの構成

### ◇ハードと通信するためのコード

- SPI, I2Cの通信そのもの
- シリアルポートやネットワーク通信のコード

※コンピュータやOS固有

### ◇センサの機能を利用するためのコード

- レジスタの読み書き、データ列解釈
- 初期化、設定、計測値の読み出し

※センサ固有

# デジタルセンサのソフトウエア

## ○ SPI・I2Cとシリアル型

### ◇SPI型・I2Cの一般形

後述

・アドレス付きのレジスタを読み書きする。

・マイコン側が積極的に読みに行かないと、データは得られない。(マイコンがマスタ，pull型)

### ◇シリアル型の一般形

・文字列、バイト列の送信で設定など。

・設定した周期での「測定値垂れ流し」機能があることが多い。→受信を続ける

# デジタルセンサのソフトウエア


## ○ シリアル型の例：6軸力覚センサ

ワコーテック社製WDF-6A100-2（USB-IF版）

◇通信方式：USBシリアル変換内蔵型

・パソコン側からはシリアルポートCOM?に見える＝シリアルポートを開くプログラム

◇センサへのコマンド（送信）

・R:1回測定値要求　S:連続出力　E:停止

◇センサからの出力（受信）

・単純な16進数数値文字列

# デジタルセンサのソフトウエア

## ○ シリアル型の例：６軸力覚センサ

### ◇プログラムの構成例

・シリアルポートの通信設定

・Sコマンド送信→データが届きはじめる

・繰り返し：

データを１行読み込む

文字列を切り出して数値に変換

校正値をもとに[N]、[Nm]に換算

データの記録・利用した処理

# デジタルセンサのソフトウエア


## ○ シリアル型の例：レーザ距離計

SICK社製 LMS 200 レーザレンジファインダ

◇通信方式：RS422（差動平衡型シリアル）

・RS422通信できるハードをPCに接続

※USB変換 or 拡張スロットにボード

→シリアル通信としては、ソフトは同等

◇コマンド、出力

・パケット(データ塊)の送受

・可読性のないバイト列＝専用コード必要

# デジタルセンサのソフトウエア

## ○ シリアル型の特徴

◇データは「生」ではなく「料理済み」

・ある程度の信号処理済でそのまま利用可。

・校正なども不要な場合が多い。

◇対応ソフトウエアは、バイト列処理

・サーバソフト、何らかの通信プロトコルの実装経験があれば、そのまま対応できる。

・バイト欠損の対応処理は必要。

※不良データの破棄、データ境界の再認識、等

# デジタルセンサのソフトウエア

## ○ SPI型・I2Cの特徴

◇データは大抵は「生」

・センサのAD変換値そのまま、等。

・ゼロ点の補正や校正が必要。

◇対応ソフトウエアは、値の取得と信号処理

・値を取得するためのコード
＝ハードウエアを直接操作するプログラム

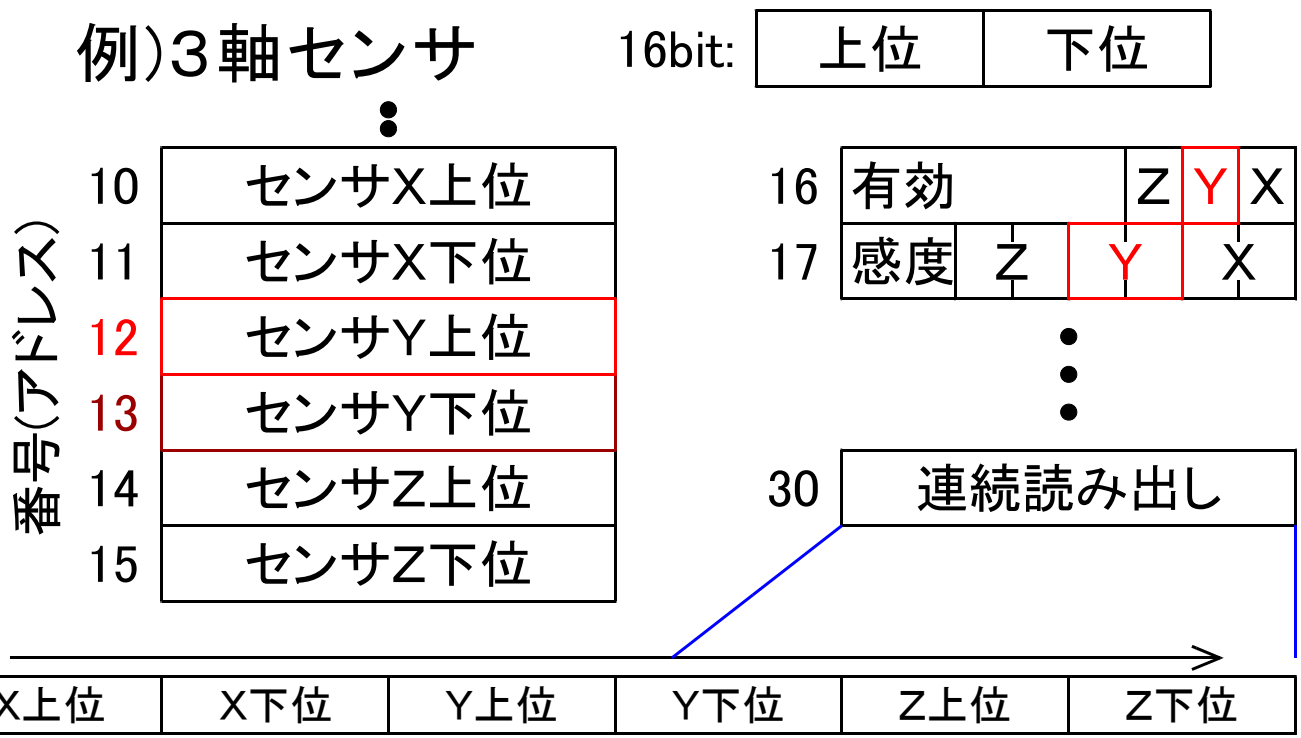
・直接的に値を触ることができ、プログラムはわかりやすい。

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ SPI型・I2Cのデータ構造（レジスタマップ）

### ◇レジスタ番号によるデータや設定の読み書き

例)3軸センサ



# SPI型・I2Cデジタルセンサのソフトウエア

## ○ SPI型・I2Cのデータ構造（レジスタマップ）

### ◇センサとやりとりする値の一覧

・センサの特定の番地(アドレス)を読み書きすることで、センサの値を得たり、設定の変更をしたりする。

・レジスタマップ（アドレスと値の対応）

### ◇一括連続読み出し （バーストリード）

・特定のアドレスを連続して読み込むとセンサ値がまとめて読み出せる。

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ0：おそらく必須なもの

### ◇用意すべきもの

- 正常に動くはずのハード
- オシロスコープ
- ソフトの動作確認のための仕掛け

例）シリアル通信でPCとやりとり(推奨)

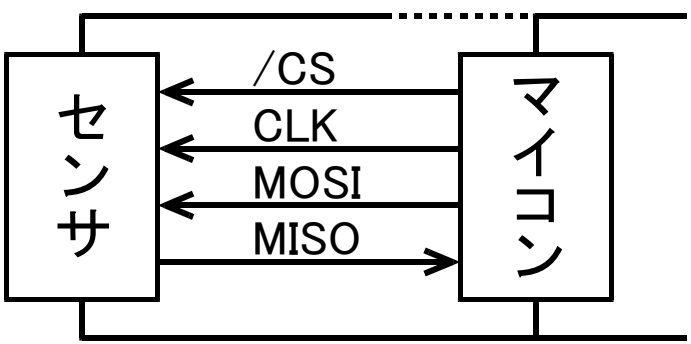
LEDで状態表示(上に加えて推奨)

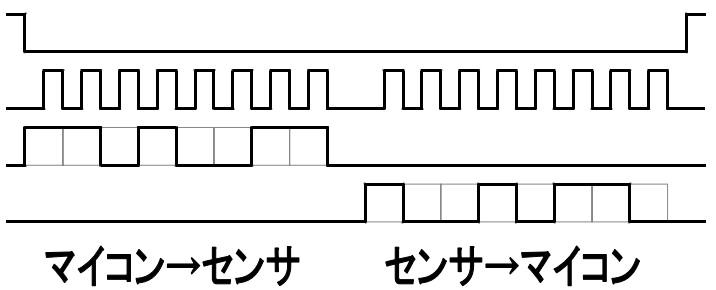
# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1−1：SPI型の通信コード

・汎用のデジタル入出力端子4本で接続。

・データシートのタイミングチャートに応じて、汎用ピンの0/1を書き換え、読み込みをする。



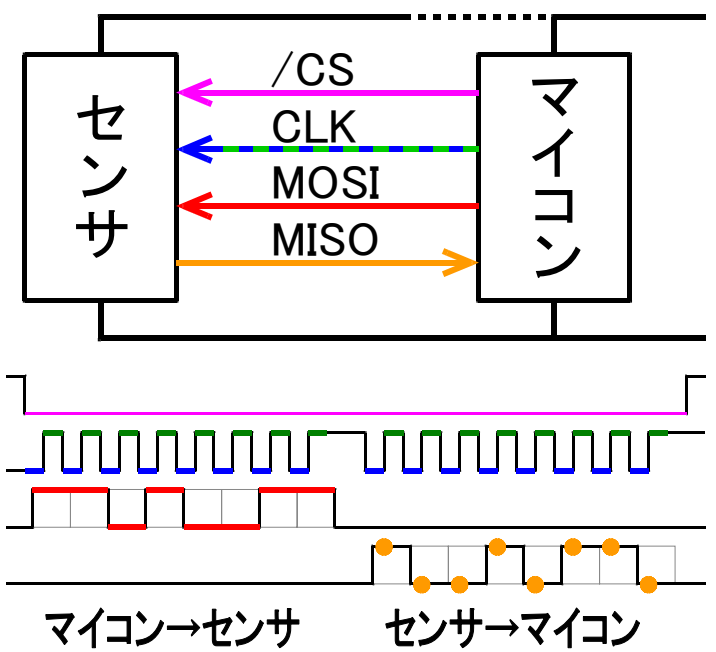
# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1−1：SPI型の通信コード（例）

```
byte ReadByte(byte reg) {
    CS=0;
    for(i=0;i<8;i++) { // 8回
        CLK=0; CLK=0;
        MOSI=(reg&0x80)?1:0;
        reg=reg<<1;
        CLK=0;
        CLK=1; CLK=1; CLK=1;
    }
    for(i=0;i<8;i++) {
        CLK=0; CLK=0;
        R=(R<<1)|MISO;
        CLK=1; CLK=1; CLK=1;
    }
```

（続き）

```
    CS=1;
    return R;
}
```



# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1−1：SPI型の通信コード

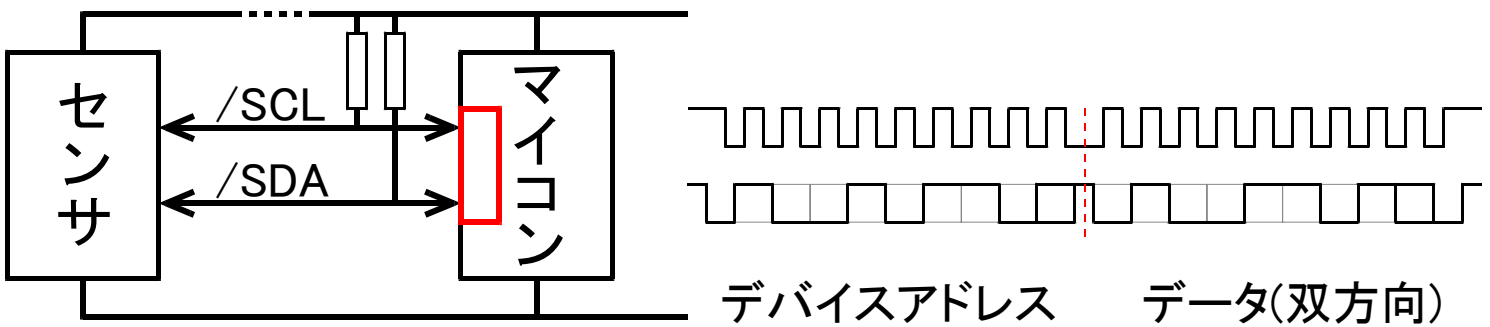
### ◇ソフトウエアでSPI型を実現する注意点

- ピンの入出力設定、タイミングチャート
- パルス幅、タイミングとマイコンの速度
  - 遅いマイコンは手順のみ重要。
  - 数十MHzで動作するマイコンは、
    速すぎるパルスを出す可能性がある
    → 同じ出力命令を重ねる
      ※最適化に注意、オシロで確認

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1-2：I2Cの通信コード

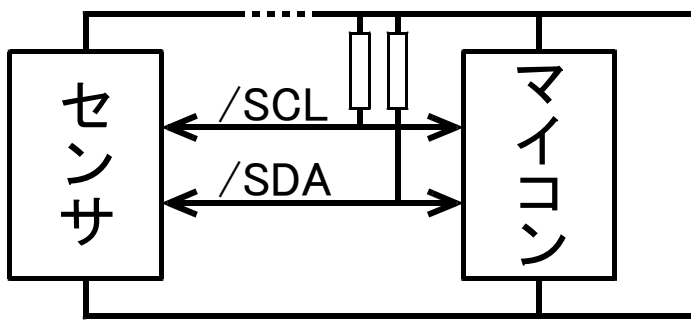
- 専用の入出力端子に接続
  ※汎用端子でもできないことはない
- マイコンの機能設定(初期化、通信速度)
- マイコンの機能を使った送受信

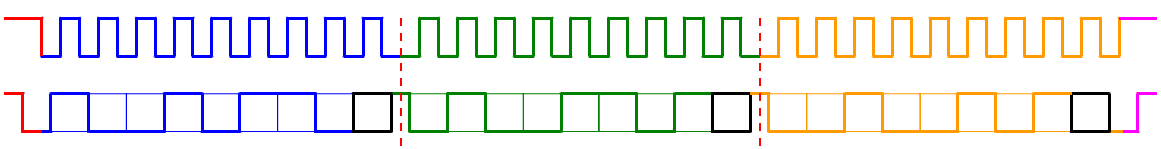


# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1-2：I2Cの通信コード（例）

```
void WriteByte(byte reg,data)
{
    StartI2C1();
    IdleI2C1();
    MasterWriteI2C1(IMUWADDR);
    IdleI2C1();
    MasterWriteI2C1(reg);
    IdleI2C1();
    MasterWriteI2C1(data);
    IdleI2C1();
    StopI2C1();
}
```

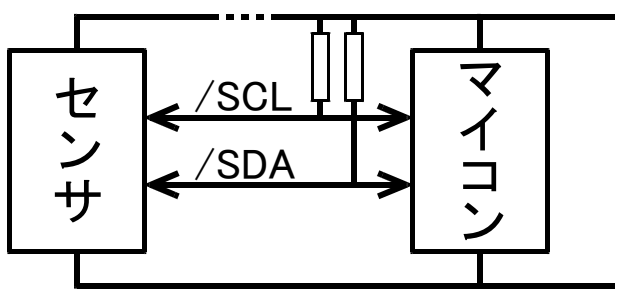


# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1-2：I2Cの通信コード（例）

```
byte ReadByte(byte reg)
{
    StartI2C1();    IdleI2C1();
    MasterWriteI2C1(IMUWADDR);
    IdleI2C1();
    MasterWriteI2C1(reg);
    IdleI2C1();
    RestartI2C1(); IdleI2C1();
    MasterWriteI2C1(IMURADDR);
    IdleI2C1();
    r=MasterReadI2C1();
    NotAckI2C1(); StopI2C1();
    return r;
}
```



# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ1-2：I2Cの通信コード

### ◇マイコンの機能サポート確認

・ソフトで実装は難易度高め（双方向性等）。

・機能の有無、機能の使い方を確認。

・メーカからライブラリ、例があると楽(まし)。

### ◇実装後のオシロ確認

・SCL端子のクロック周期が意図した通りか。

・データが出ていくか ／
読み込み時にデバイスから反応があるか。

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ２：センサの応答試験

### ◇テスト対象を探す

・初期化しなくとも値が返るものを探す

例）センサのバージョン情報
センサのアドレス(I2C)
初期値ゼロではない、読める制御レジスタ

### ◇読んでみる

・最初からフル機能のプログラムを書かず
上記対象にのみ周期的にreadする。
※オシロで確認しやすいように

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ３：初期化と値の読み込み

### ◇センサ固有の作業

・初期化の必要性

例）リセット、計測対象の有効化、
スリープ解除、コードのダウンロード
マニュアル、サンプルコード、ネットの公開プログラム

・なにか反応する値を読む

注）計測値がベストとは限らない

注）読む順番が重要な場合有り：

例）上位→下位の順で読むこと

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ3-2：初期化　例：姿勢センサ(I2C)

```
void SetupMPU6050(void)
{
    // device reset, sleep release
    Delay( 10); I2CWriteByte(0x6b,0x80);
    Delay(100); I2CWriteByte(0x6b,0x00);

    // Data sampling setting, 1kHz/(3+1)=250Hz
    Delay(1);   I2CWriteByte(0x19,0x03);
    // digital LPF setting, 1kHz-184Hz band
    Delay(1);   I2CWriteByte(0x1a,0x01);
    // gyro scale setting, 250deg/s
    Delay(1);   I2CWriteByte(0x1b,0x00);
    // acceleration scale setting, 2g
    Delay(1);   I2CWriteByte(0x1c,0x00);
```

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ3-2：初期化　例：姿勢センサ(I2C)

```
// Data sampling setting, 1kHz/(3+1)=250Hz
I2CWriteByte(0x19,0x03);
```

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|
| SMPLRT_DIV[7:0] | | | | | | | |

＝[3]

```
// digital LPF setting, 1kHz-184Hz band
I2CWriteByte(0x1a,0x01);
```

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|
| | | ExtSyncSet[2:0] | | | DLPF_CFG[2:0] | | |

＝[1]

```
// gyro scale setting, 250deg/s
I2CWriteByte(0x1b,0x00);
```

| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|
| XGST | YGST | ZGST | FSSEL[1:0] | | ＝[0] | | |

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ3-2：取得　例：姿勢センサ(I2C)

```
void ReadMPU6050Register(void)
{
    Delay(1); imu.GX =  I2CReadWord(0x43);
    Delay(1); imu.GY =  I2CReadWord(0x45);
    Delay(1); imu.GZ =  I2CReadWord(0x47);
    Delay(1); imu.AX =  I2CReadWord(0x3b);
    Delay(1); imu.AY =  I2CReadWord(0x3d);
    Delay(1); imu.AZ =  I2CReadWord(0x3f);
    Delay(1); imu.TMP = I2CReadWord(0x41);
}
```

※G?:　ジャイロ測定値　A?:加速度測定値　TMP:温度センサ

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ４：センサの性能を引き出す

### ◇パラメータの調整

- センサ固有の感度などの設定
- 信号処理パラメータの設定

### ◇バースト転送モードの活用（センサによる）

- いちいちアドレス指定せずに、主要な値を連続的に取得できる。
- ソフトの負担低減、データ測定周期向上。

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ4-1：バースト　例)マウスセンサ

◇レジスタ 0x50 Motion Burst　説明書抄訳：

*[運動取得]はMotionBurstの読込による。*
*ADNS6010は[動作状況][X移動][Y移動]*
*[表面状態][レーザ出力]..を順番に出力*
*する。*

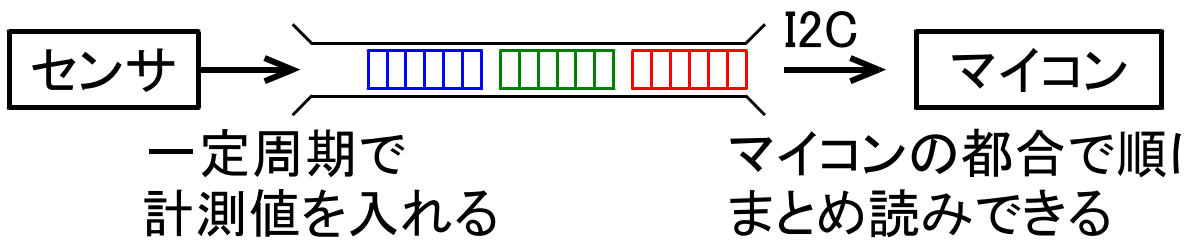
・それぞれ、個別のレジスタにあるものを
一括して読める　＆　途中打ち切りも可。
・データを同一タイミングで取得できる。

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ4-2：バースト　例)姿勢センサ

### ◇FIFOを活用する

- FIFO: First In First Out




- センサの測定タイミングの決定をセンサの内部回路が担当

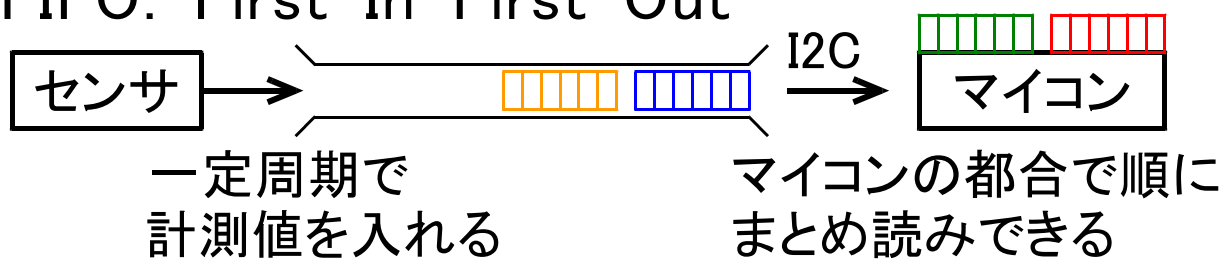
  → マイコンはデータの有無を確認して、あれば読み出し。（バーストにて）

# SPI型・I2Cデジタルセンサのソフトウエア

## ○ ステップ4-2：バースト　例)姿勢センサ

### ◇FIFOを活用する

・FIFO:  First  In  First  Out




・マイコン側のタイミング設計が楽になる。
時間等間隔の正確さを任せられる。
制御とセンシングの周期を変えやすい。

・あるだけ読み出し→等周期を仮定して処理

# デジタルセンサのソフトウエア

## ○ ステップ５　信号処理

### ◇データの加工、情報化

・信号処理の必要性はアナログと変わらず。
ただし、一般に信号の質が高く、小細工的処理が不要で、目的の処理のみ。

・高級なセンサは処理不要で使用できる。

# デジタルセンサのソフトウエア

## ○ ステップ６　校正(キャリブレーション)

◇デジタルセンサの校正は楽(たぶん)

・センサの仕様はアナログ部などの校正も含めて精度規定→校正の必要性低下

c.f.アナログ：センサ特性、増幅回路特性等

・校正が必要な事例：

ゼロ点のずれ（含む　温度依存）

取り付けに依存するセンサ外の校正

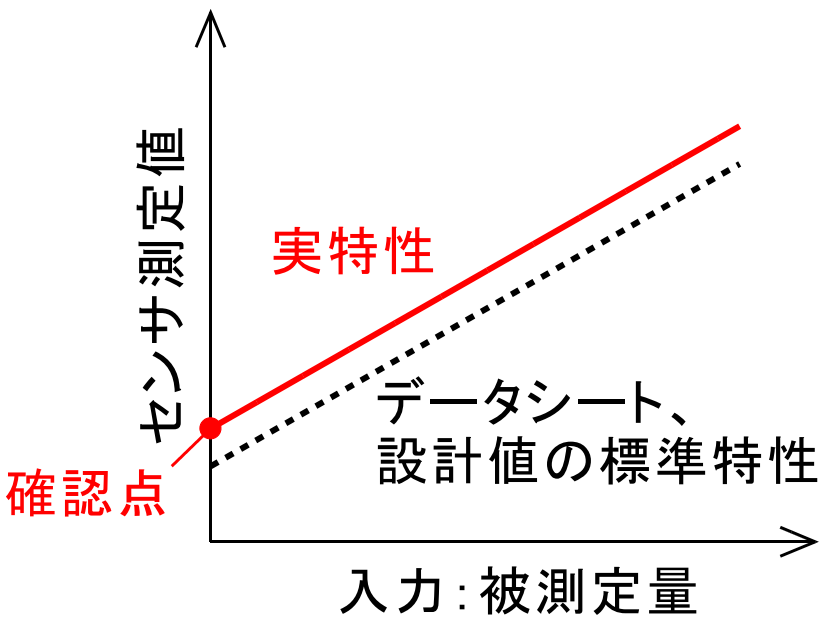
本格的校正（精度を保証する場合）

# キャリブレーション

## ○ キャリブレーションの一例 （ゼロ点のみ）

「ゼロ」に対する出力を測定 ＋

被測定値に対する感度(傾き固定)

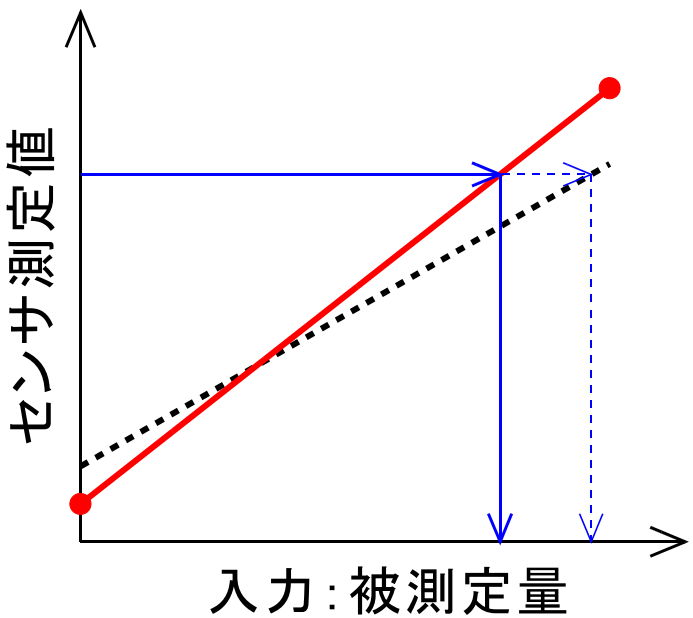



- 体重計
- 機械の原点検出

# キャリブレーション

## ○ キャリブレーションの一例 （直線的）

測定対象区間で2カ所の測定を行い、
その間を一次式(y=ax+b, x=(y−b)/a)で求める。




- 体重計
  （出荷前、傾きを記憶）
- 姿勢センサ
  （導入時のみ）
- ３点で2次式

※一度は確認すべし

# まとめ：これまでのメカトロセミナー関連

○ 検索：[ロボット開発工学]→メカトロセミナー

- なにを測定するか →C06,C07
- センサとマイコンの選択 →C06,C11
- 通信方法の検討、確認 →C11
- 信号処理の実装 →C07
- キャリブレーション/校正 →C07

- 「デジタル」 →C03
- 「マイコン」 →C02
- アナログセンサをつなぐ →C12

# まとめ

## ○ デジタルセンサをマイコンにつなぐ

- アナログセンサに比較して便利で強力。

  アナログ部の隠蔽

  十分なAD変換分解能

  各種信号処理込み

  誤差発生要因の低減　　など

- アナログ回路経験なくとも、ソフトウエアに強ければ、センサを活用できる。
  逆に、ソフト経験ないと使いづらい。

# まとめ

## ○ 開発手順と注意点

- 開発は、大半がソフトで、回路は少ない。
- センサの信号仕様に合わせた、通信部の実装 → センサ固有の設定と活用
- 電源を入れてすぐに動作が見えるわけではないため、トラブル時に問題の切り分けに苦労する場合がある → 調査手段の充実
- 設定項目は多いが、理由があるはずで、使いこなしには、意図を読むことが重要。