

# 双方向ハンドセンサー ユーザーマニュアル

<https://interactive-hand-sensor.com/root/>

目次(リンク)	1
1. 重要事項	2
2. 特徴	3
2-1. フォトリフレクタ型 3次元空間センサー	3
2-2. カラーLED (CLED)	3
2-3. 応用例	3
2-4. 電気仕様	3
原理図	4
3. ハードウェア	5
3-1. 本体基板	5
主なデジタルIC(ソフト開発の参考にして下さい)	5
3-2. 本体基板ピンアウト(JP1)	6
3-3. JP2ピンアウト	7
3-4. センサー基板	8
3-5. センサー基板取り付け1: 本体基板	8
3-6. センサー基板取り付け2: リボンケーブル	9
※極性反転用ピンヘッダは10ピン1.27mmピッチ、ピン長さ3mm	9
3-7. センサー感度調整方法	10
3-8-1. 本体基板回路図	11
3-8-2. センサー基板回路図	12
3-9. 用語	13
4. ソフトウェアとインターフェイス	14
4-1. SPIインターフェイス設定	14
4-2. センサー読み取り手順	14
4-3. SPI-read ADコンバータMCP3208の読み取り	14
4-4. CLED書き込み	16
5. 並列接続	18
5-1. 注意	18
5-2. 準備	18
5-3. 回路図	18
5-4. 応用例	19
改訂	20

# 1. 重要事項

## 1. 安全

- a. 動作中に異臭、煙などの異常を感じたとき、又は正しく動作しないときはすぐに電源をOFFして確認してください。
- b. 動作中は本体基板、センサー基板に触らないで下さい。  
静電気による破損、誤動作の恐れがあるため。

## 2. 付近のリモコン機器への影響

- a. 動作中は強い赤外線を出し続けるため付近の(エアコン、家電などの)赤外線リモコンに影響する恐れがあります。
- b. リモコンが効かないときは角度、場所を変えて下さい。

## 3. センサーの感度調節

- a. 8個のセンサーは出荷時に感度を調整していますが振動などでズれることがあります。必要に応じて[感度を調節して下さい。](#)

## 2. 特徴

### 2-1. フォトリフレクタ型 3次元空間センサー

#### 1. 原理

- a. 赤外線LED([ILED](#))に瞬間的に大電流を流しフォトトランジスタ([PhTr](#))で受光
  - b. 反射光を電圧に変換し12bitADコンバータで読む。(原理図)
  - c. センサーを[X,Y方向](#)に配置して手の動きをを3次元で検出する。
2. [Z方向](#)検出距離は20cm以上、強い光が当たると誤検出、感度低下する。
  3. Z方向の距離測定精度は低い。X,Y方向は高速、高精度で検出可能。
  4. 普通の赤外線LEDを使用、レーザーや紫外線など有害な光は使わない。
  5. ノイズやセンサーのばらつきをソフトウェアで補正し精度を上げている。

### 2-2. カラーLED ([CLED](#))

1. センサーとは独立しマイコンで設定する。(センサーと連動も可能)
2. PWMで光の強さを制御できる。

### 2-3. 応用例

[本体基板 並列接続](#) : 16枚まで動作確認済

[センサー基板 延長ケーブル](#): センサーを任意の場所(模型など)に配置できる

### 2-4. 電気仕様

項目	内容	備考
電源電圧	5V	USB電源を想定
コントローラ電圧	5V, 3.3V	信号出力3.3V
消費電流	180mA(スキャン時間1.2mS)	基板単体typical値
消費電力	1W	基板単体typical値
スキャン時間基板単体	1.2mS以上、1チャンネル150uS以上	
スキャン時間並列続	800uS以上、1チャンネル100uS以上※	

※検出距離下げないためには1チャンネル150uS以上が望ましい。

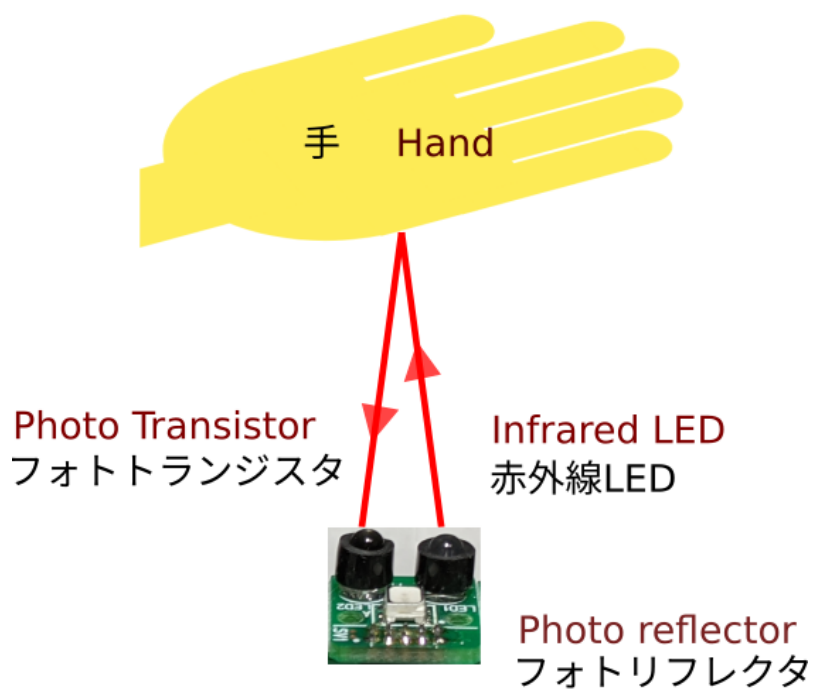


図2-1. フォトリフレクタ原理

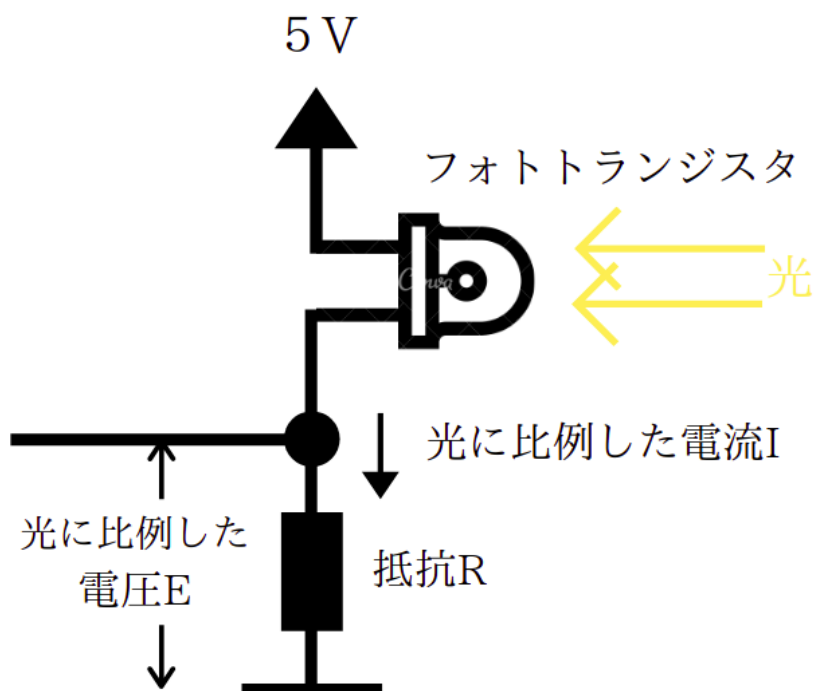


図2-2. 光—電圧変換のしくみ

1. フォトランジスタで光の強さを電流Iに変換
2. 抵抗Rに電流Iを流し電圧Eに変換
3. 電圧Eを12bit ADコンバータで読む

### 3. ハードウェア

#### 3-1. 本体基板

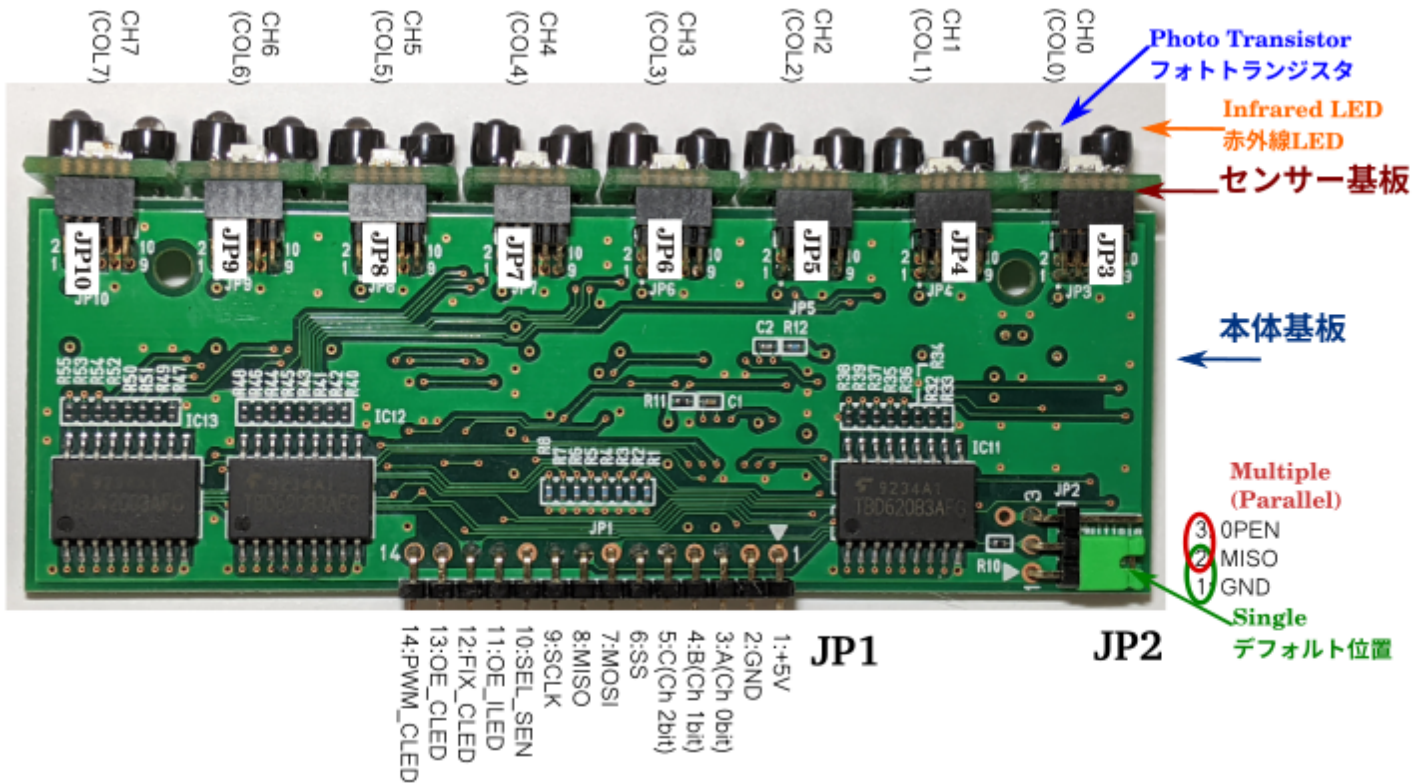


図3-1. 本体基板

※緑の基板は量産基板と同じパターンで色違いの試作3基板、色以外は量産品と同じです。  
今後の改訂で写真を黒の基板に変更予定。

主なデジタルIC(ソフト開発の参考にして下さい)

型式	リファレンス番号	PIN	説明
MCP3208T-BI/SL 12bitADコンバータ	IC1	6,7,8,9,10	フォトトランジスタの電圧を検出、インターフェイスはSPI
74HC595 シフトレジスタ	IC10,11,12	6,7,8,9,10, 12,13,14	シリアルパラレル変換でCLEDを制御する
74HC138 デコーダ	IC3	3,4,5	ILEDのチャンネル(0-7)を選択する

### 3-2.本体基板ピンアウト(JP1)

PIN番号	PIN名	説明
1	+5V	Vcc 5V
2	GND	GND
3	A	ILED channel(0-7) 0bit
4	B	ILED channel(0-7) 1bit
5	C	ILED channel(0-7) 2bit
6	SS	SPI Slave-Select (active-Low) 並列接続時の基板選択
7	MOSI	SPI Input
8	MISO	SPI Output (3.3V/5V切り替えJP2)
9	SCLK	SPI Clock
10	SEL_SEN	SPIの選択 High:センサー(ADコンバータ), Low:CLED
11	OE_ILED	ILED Output Enable ※ILED:赤外線LED ILED 有効 / 無効 ※赤外線放射はADコンバータに同期し自動で行う
12	FIX_CLED	CLED Shift Register Latch ※CLED:カラーLED シフトレジスタの値を出力に反映する
13	OE_CLED	CLED Output Enable CLED 出力 ON / OFF
14	PWM_CLED	PWM(Pulse Wide Modulation)制御でCLEDの明るさを調節するPWM を使わないときは13pin又は5Vにショート

#### まとめ

- センサー(ADコンバータ)とCLEDのSPI切り替えはSEL\_SENピン。
- ILEDのチャンネルはABCで設定する。
- CLEDはシフトレジスタにデータ転送後、FIX\_CLEDのパルスで出力に反映する。
- CLED ONは13pinと14pinのAND出力、14pinのPWMで明るさを調整できる。

### 3-3.JP2ピンアウト

PIN番号	PIN名	説明
1	GND	GND
2	MISO	単体／並列接続選択用
3	OPEN	OPEN

#### まとめ

- 並列接続のときSPI出力(MISO)が下がる対策用スイッチ
- 基板単体で使用するときはそのまま(single)  
PIN1-2をジャンパーピンでショート、出荷時の状態。
- 並列接続するときは1枚だけPIN1-2ショート(single)、残りは2-3ショート(multiple, OPEN)  
SPIの出力ピンMISOは5V信号を1K $\Omega$ と2.2k $\Omega$ で分割して3.3Vに変換している。基板を並列接続したとき(合成抵抗値低下で)信号レベルが下がる対策。

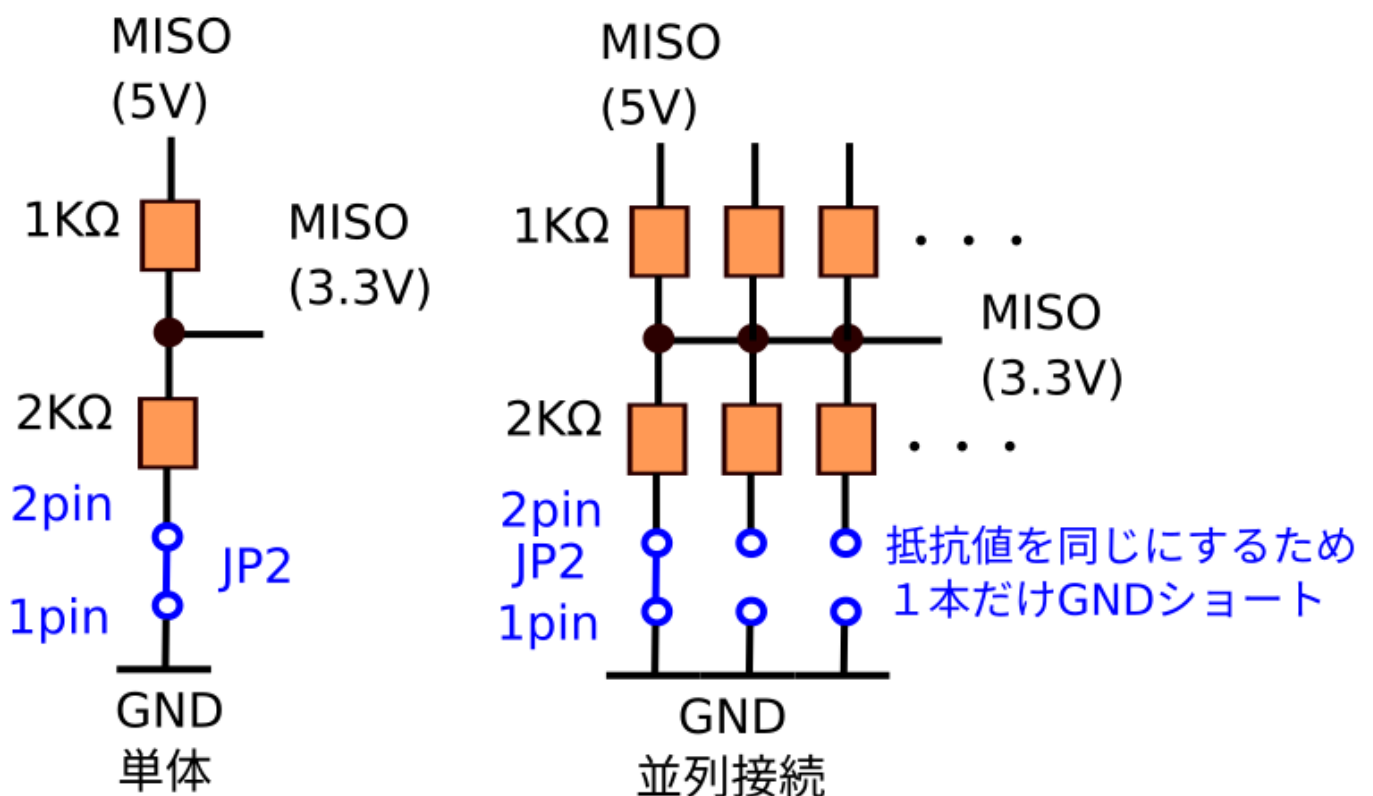


図3-2. JP2説明



### 3-4. センサー基板

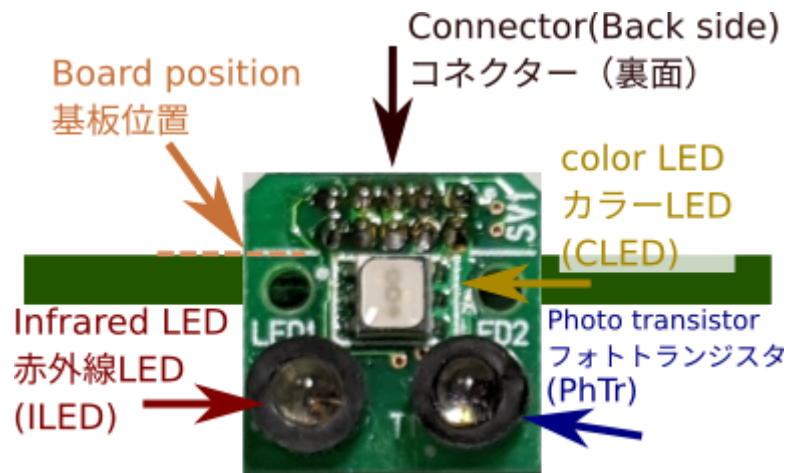


図3-3. センサー基板

- 動作前に向き、ピンずれを確認して下さい。
- コネクタに力をかけないでください。
- 動作中は発熱します。

### 3-5. センサー基板取り付け1: 本体基板

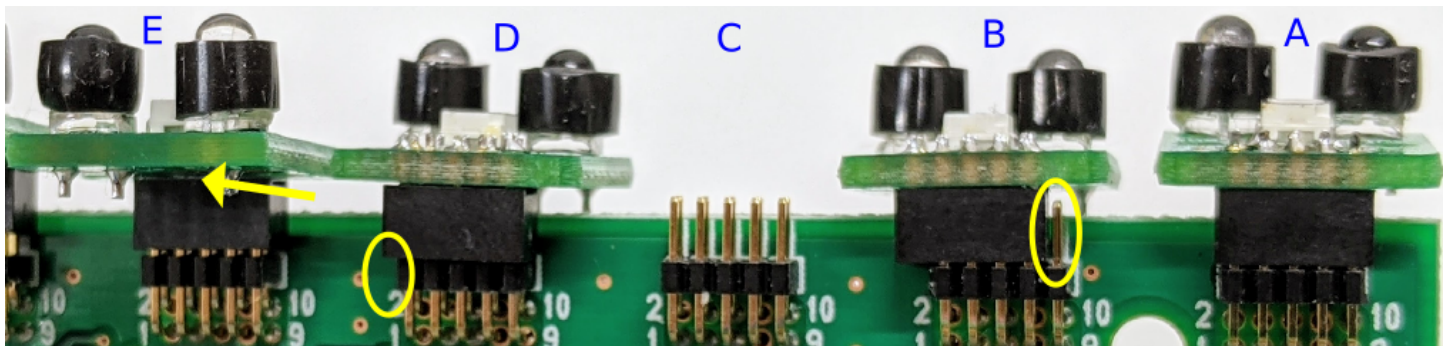


図3-4. センサー取り付け良い例、悪い例

- A: 良い
- B: 1列横ずれ、悪い
- C: 無し、悪い
- D: 1列手前ずれ、悪い
- E: 向きが逆、悪い

- 電源ONの状態ではセンサー抜き差しはしないでください。
- 1列ずれ逆向きなどで壊れ難いピン配置にしていますが、点灯しない場合は電源OFFにして直して下さい。



### 3-6. センサー基板取り付け2: リボンケーブル

- リボンケーブル2本、極性反転用のピンヘッダ、センサー取り付け用ピンヘッダが必要。
- リボンケーブルは1.27mmピッチ10pin

※リボンケーブル1本のみ端子が左右反転するため動作しない。

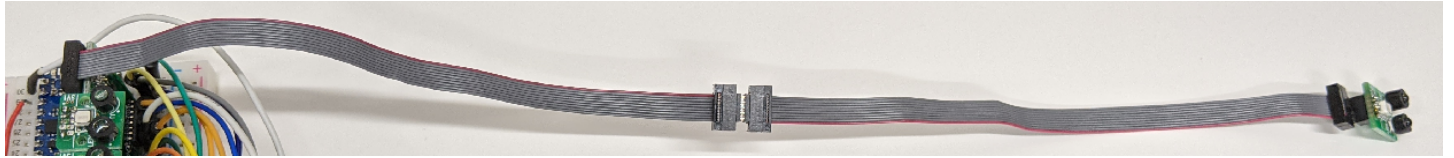


図3-5. リボンケーブル全体図

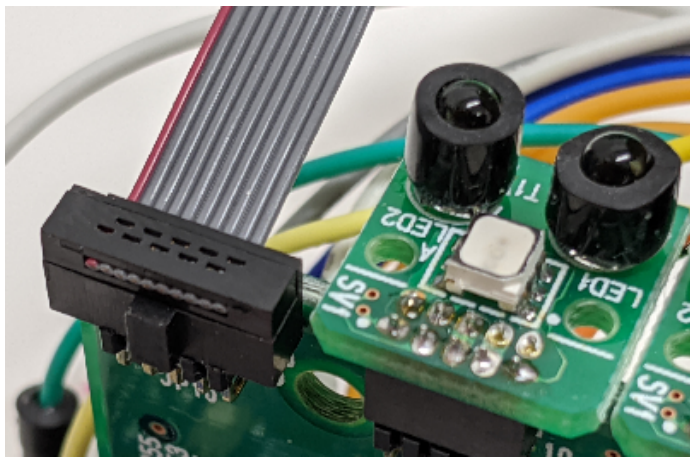


図3-7. リボンケーブル本体基板取り付け

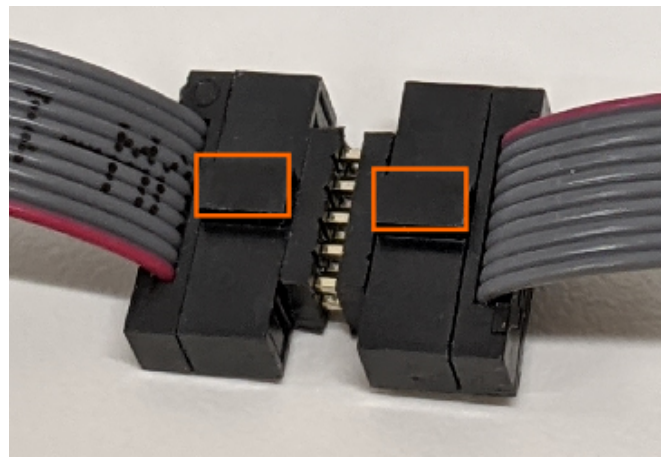


図3-6. リボンケーブル接合部分

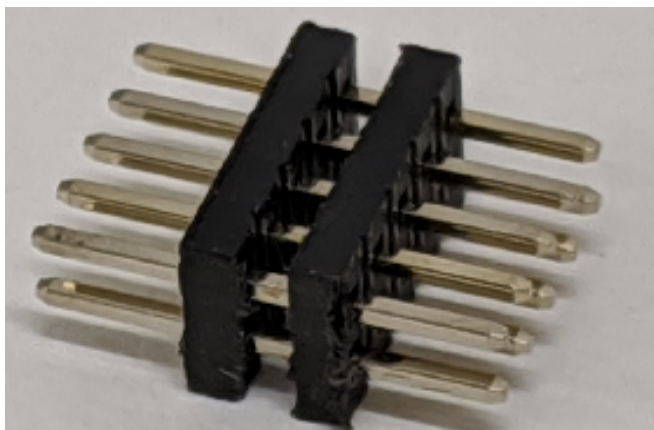


図3-8. 極性反転用ピンヘッダ1.27mmピッチ

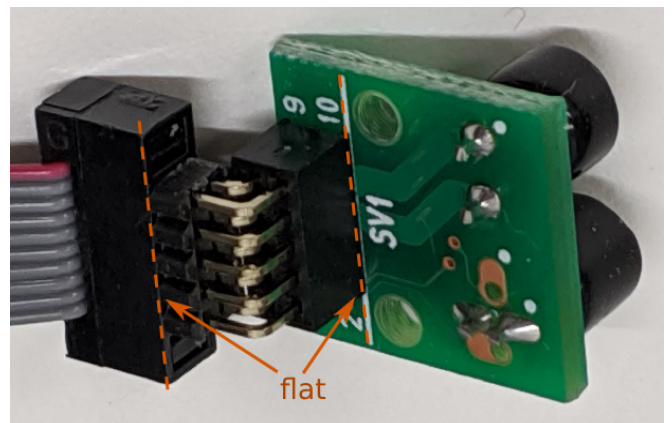


図3-9. リボンケーブルセンサー基板取り付け

※極性反転用ピンヘッダは10ピン1.27mmピッチ、ピン長さ3mm

### 3-7. センサー感度調整方法

ILEDとPhTrの黒カバーを上下に動かすとセンサーの感度調節ができますが最適な感度からズれる恐れがあるので注意して下さい。

※ILED: 赤外線LED PhTr: フォトトランジスタ

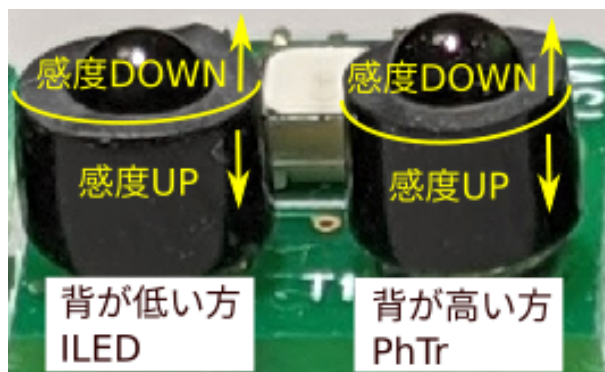


図3-10. 感度調整説明図

感度を上げる: (ILED, PhTrの露出を増やし、光を受け易く)

1. PhTrの黒カバーを下げる。
2. ILEDの黒カバーを下げる。
3. 黒カバーが高く、下げられないときは下図のように黒カバーをカットする。
  - a. 優先順位は下図A, Bの順

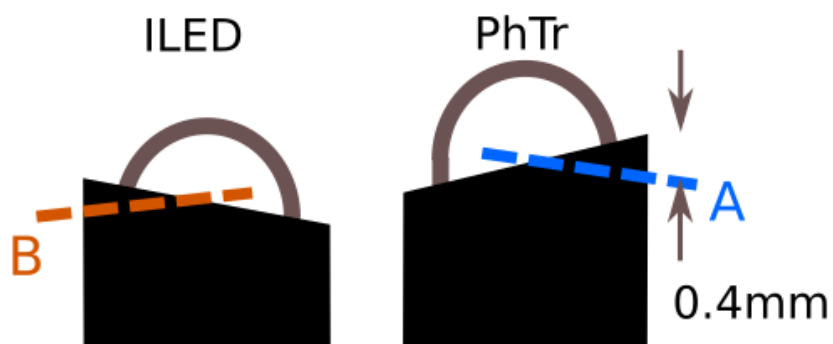


図3-11. 黒カバーカット

注意:

- デバイスのばらつきなどのため感度アップ、ダウンは限界がある。
- カットしすぎると壊れる。(常に反応するようになる)
- 黒カバーの高さが他と比べて低いときはカットしない。

感度を下げる: (ILED, PhTrの露出を減らし、光を受け難く)

1. PhTrの黒カバーを上げる。
2. 1がダメなときはILEDの黒カバーを上げる。

### 3-8-1.本体基板回路图

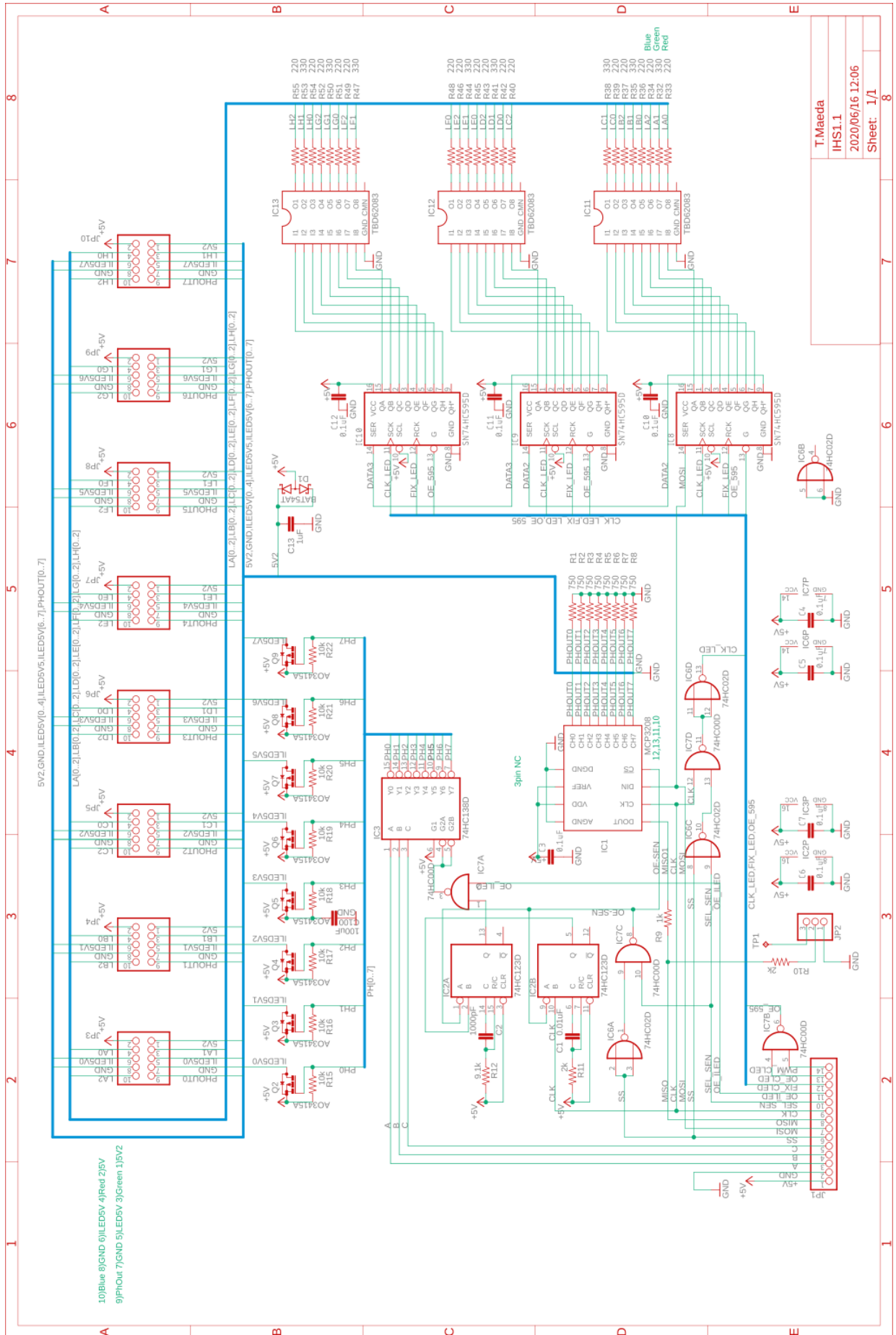


图3-11. 本体基板回路图

### 3-8-2. センサー基板回路図

#heading=h.5uszd6n5qua

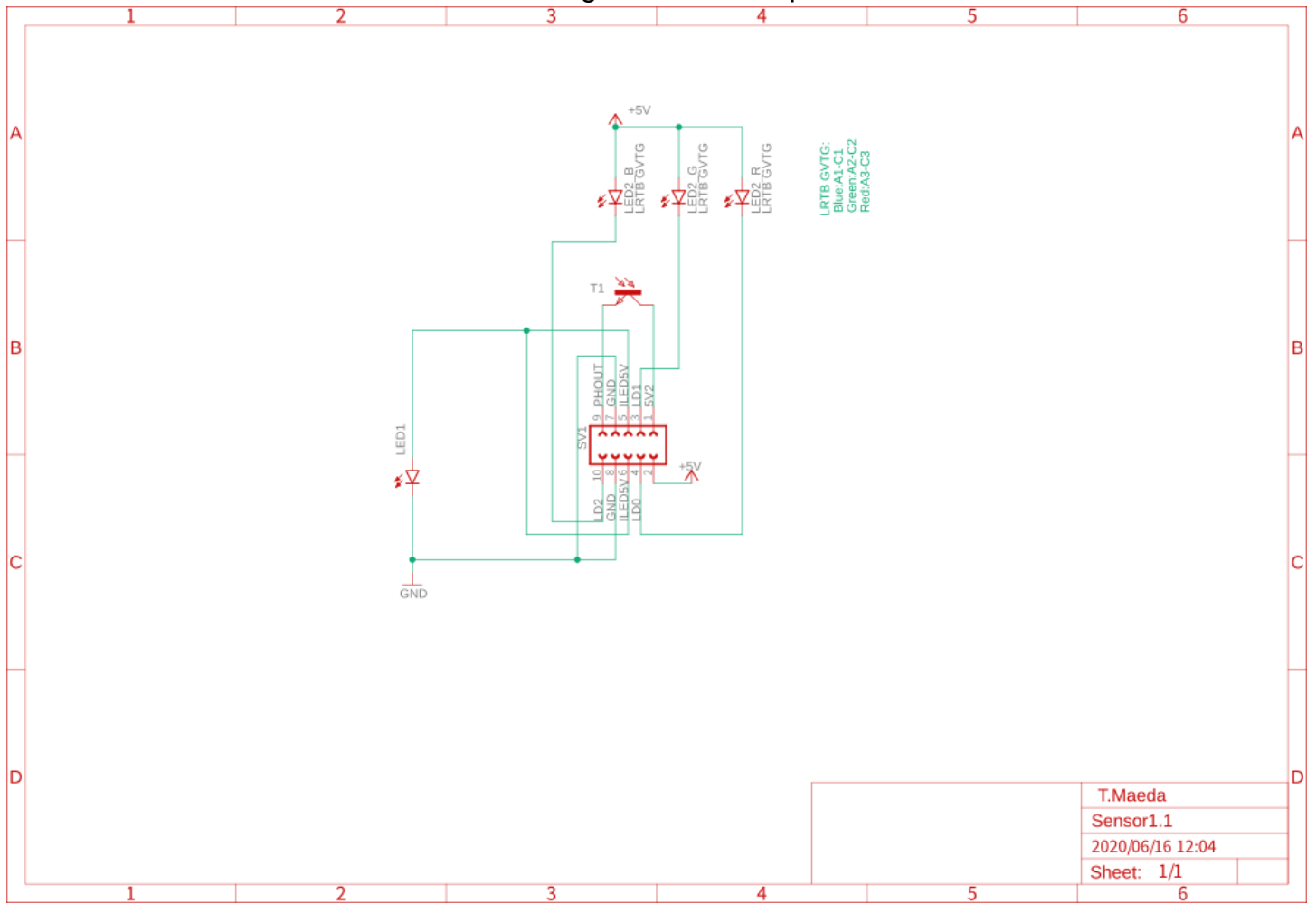


図3-12. センサー基板回路図

### 3-9.用語

単語	説明
CLED	Color LED RGBカラーLED( <a href="#">センサー基板</a> )
ILED	Infrared LED 赤外線LED(センサー基板)
PhTr	Photo Transistor フォトトランジスタ(センサー基板) 光を電流に変換するデバイス、 <a href="#">図2-2</a> のようにさらに電流を電圧に変換している。
X/Y/Z方向	下図の通り

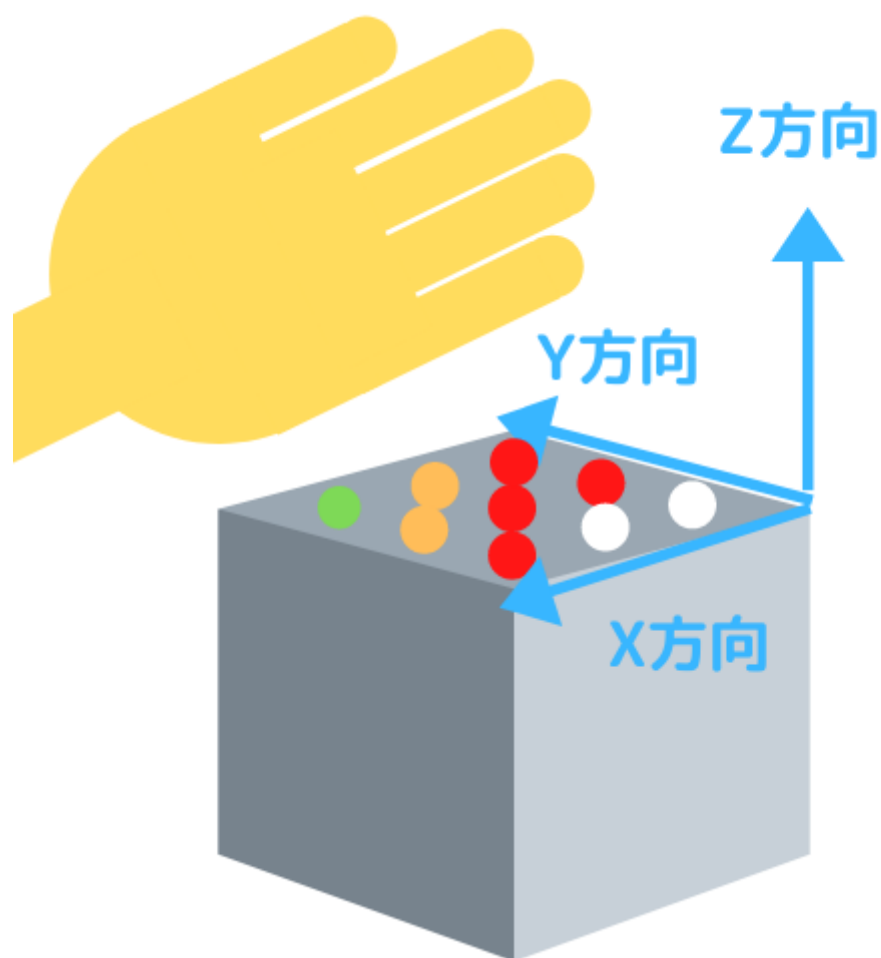


図3-13.XYZ方向

## 4. ソフトウェアとインターフェイス

### 4-1.SPIインターフェイス設定 ([ADコンバータ／シフトレジスタ](#))

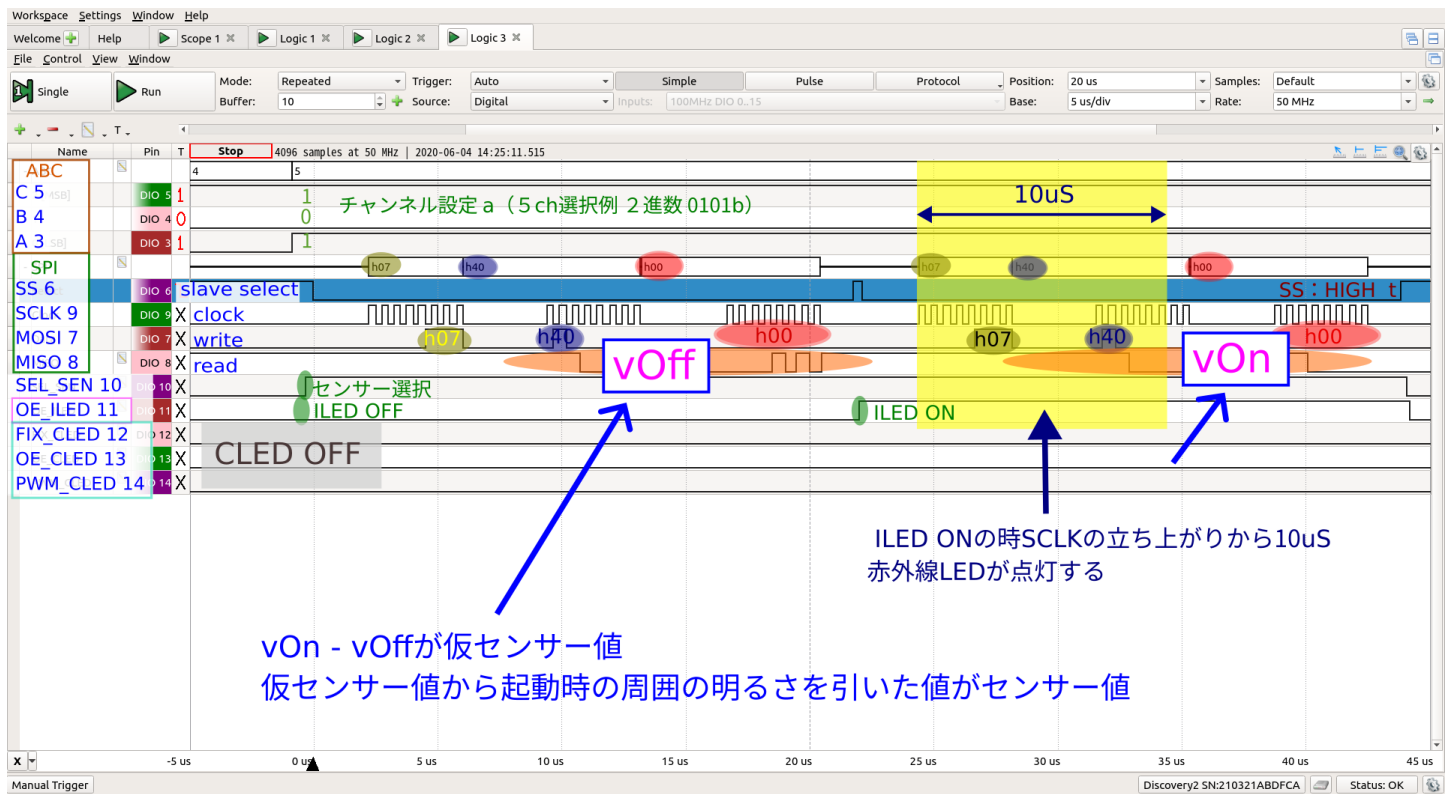
- SPI-MODE:0
- CLOCK:2MHz ※
- 1サイクル(同じILEDへのアクセス時間間隔)は1.2mS以上 ※

※ ILEDの品質、寿命を確保するため。

### 4-2.ADコンバータ読み取り手順

1. OE\_CLED:LOW
2. A,B,C:                    センサーチャンネル設定
3. SEL\_SEN:HIGH    センサー選択、CLEDを選択しない
  
4. OE\_ILED:LOW    ILED OFF        ※[ILED](#):赤外線LED
5. SS                :LOW
6. **SPI-read**    AD読み取り値:**vOff**
7. SS                :HIGH
  
8. OE\_ILED:HIGH    ILED ON
9. SS                :LOW
10. **SPI-read**    AD読み取り値:**vOn**
11. SS                :HIGH
12. 仮センサー値:**vOn - vOff**    配列に保存

※ 仮センサー値から起動時に測定した周囲の明るさを引いた値がセンサー値です



4-1.ロジックアナライザ センサー読み取り図



### 4-3.ADコンバータの読み取りコード

#### 8個のセンサー読み取り

```
void Sensor::setAd(bool blnit) { // blnit:add val to adAryInit[]
    for (int col = 0; col < COL_LEN; col++) { // COL_LEN: 8
        setCol(col, blnit); // *** sensing ***
    }
    if (!blnit) // *** set CLED ***
        cled.set(val); // sensor-data, indicator
}
```

#### センサー単体の読み取り

```
void Sensor::setCol(int col, bool blnit) {
    int ledSta = digitalRead(OE_CLED);
    digitalWrite(OE_CLED, LOW);
    out3bit(abc, col);
    digitalWrite(SEL_SEN, HIGH);
    digitalWrite(OE_ILED, LOW); // ILED-OFF
    int vOff = getAdc(col); // *** read ADC ***
    digitalWrite(OE_ILED, HIGH); // ILED-ON
    int vOn = getAdc(col); // *** read ADC ***
    digitalWrite(SEL_SEN, LOW);
    digitalWrite(OE_ILED, LOW); // ILED-OFF
    digitalWrite(OE_CLED, ledSta);

    if (blnit)
        adAryInit[col] += vOn - vOff - (vOff >> 2) + (vOn >> 4);
    else
        adAry[col] = noMinus(vOn - vOff - (vOff >> 2) + (vOn >> 4) - adAryInit[col]);
    val[col] = ad2val(adAry[col]); // adAryInit[col]: 起動時に測定した周囲の明るさ
}
```

```
unsigned int Sensor::getAdc(int col) {
    digitalWrite(SS, LOW);
    SPI.transfer(6 | (col >> 2)); //
    unsigned char r1 = SPI.transfer(col << 6);
    unsigned char r2 = SPI.transfer(0);
    digitalWrite(SS, HIGH);
    return ((r1 & 0x0f) << 8) + r2;
}
```

#### 4-4. CLEDデータ転送方法

1. SEL\_SENをLOW : CLEDを選択する。
2. OE\_CLEDとPWM\_CLEDをLOW : CLED出力をOFFする。
3. Sensor値1-6の色をRGB値3bitに変換してシフトレジスタに転送する。(下表)
4. FIX\_CLEDをHIGH、LOWして1パルス出力するとシフトレジスタに値が反映される。
5. OE\_CLEDとPWM\_CLEDをHIGH: CLEDが光る。

Sensor値1-6:色 変換テーブル

Sensor値1-6	1	2	3	4	5	6
色	赤	黄	緑	水色	青	紫

色:RGB値3bit 変換テーブル

色	赤(R)	黄(GR)	緑(G)	水色(BG)	青(B)	紫(BR)
BGR値3bit	1 (001)	3 (011)	2 (010)	6 (110)	4 (100)	5 (101)

Sensorサンプル値1-6をColor値に変換

channel	0	1	2	3	4	5	6	7
Sensor値1-6	1	2	3	4	5	6	1	2
BGR値3bit	1 (001)	3 (011)	2 (010)	6 (110)	4 (100)	5 (101)	1 (001)	3 (011)

Color値を左右反転

channel	0	1	2	3	4	5	6	7
BGR値3bit	1 (001)	3 (011)	2 (010)	6 (110)	4 (100)	5 (101)	1 (001)	3 (011)
Shift bit (左右反転)	<<0	<<3	<<6	<<9	<<12	<<15	<<18	<<21
Shift後 decimal	3 (011) 1 (001) 5 (101) 4 (100) 6 (110) 2 (010) 3 (011) 1 (001)							
Shift後 24bit binary	011 001 101 100 110 010 011 001							
Serial out 24bit	0110 0110 1100 1100 1001 1001							

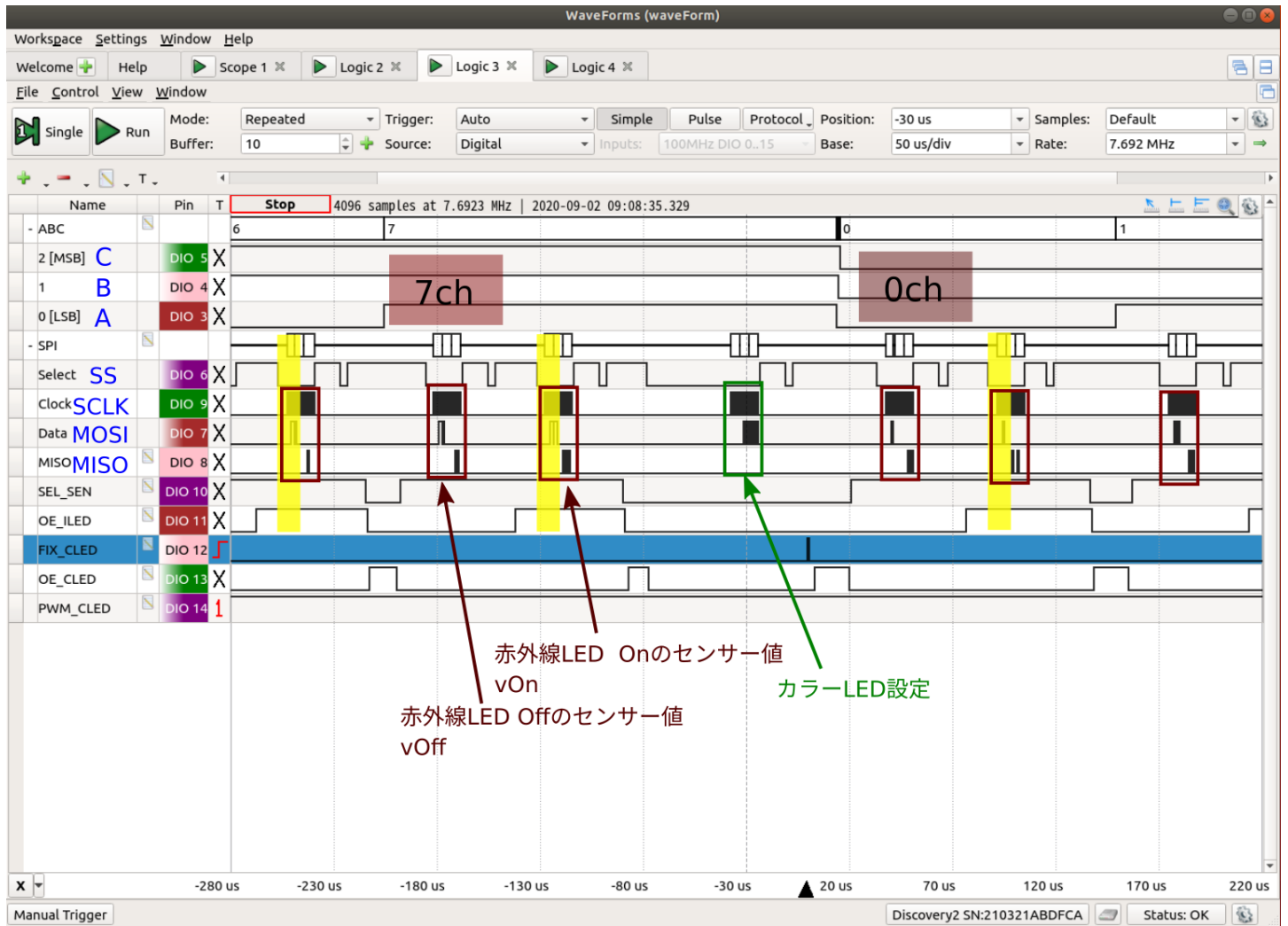


図4-2.ロジックアナライザ CLED書き込み

## 4-5.CLEDコード

```
void CLED::set(byte* pAry, bool isInd) {
    if (isInd) {
        int indiCnt = INDI_CNT << ((inOn) ? 0 : 2);
        if (++inCnt == indiCnt) {
            inCnt = 0;
            inOn = !inOn;
        }
    }
    // *****
    unsigned long val24 = 0;
    bool isCledOn = false;
    for (int col = 0; col < COL_LEN; col++) {
        isCledOn |= (pAry[col] > 0);
        bool indiOn = isInd && inOn && (col == COL_LEN - 1) && (!isCledOn); // indicator
        byte val1_6 = indiOn ? 6 : pAry[col];
        val24 += (unsigned long)color[val1_6] << col * 3;

        val1_6: センサー値: 1赤 ~ 6赤紫
        color[val1_6]: センサー値→BGR変換値
        val24: カラー値、左右反転、24ビット値
    }
    digitalWrite(SEL_SEN, LOW);
    digitalWrite(OE_CLED, LOW);
    digitalWrite(SS, LOW);
    for (int b = 2; b >= 0; b--) {
        SPI.transfer((val24 >> b * COL_LEN) & 0xff); // 8ビットずつSPIで
    } // シフトレジスタに書き込む
    digitalWrite(FIX_CLED, HIGH);
    digitalWrite(FIX_CLED, LOW); // 書き込んだ値を保存
    digitalWrite(SS, HIGH);
    digitalWrite(OE_CLED, HIGH); // (PWMで)カラーLEDオン
}
```



デコーダーで選択された行のみ有効にしている

#### 5-4.応用例

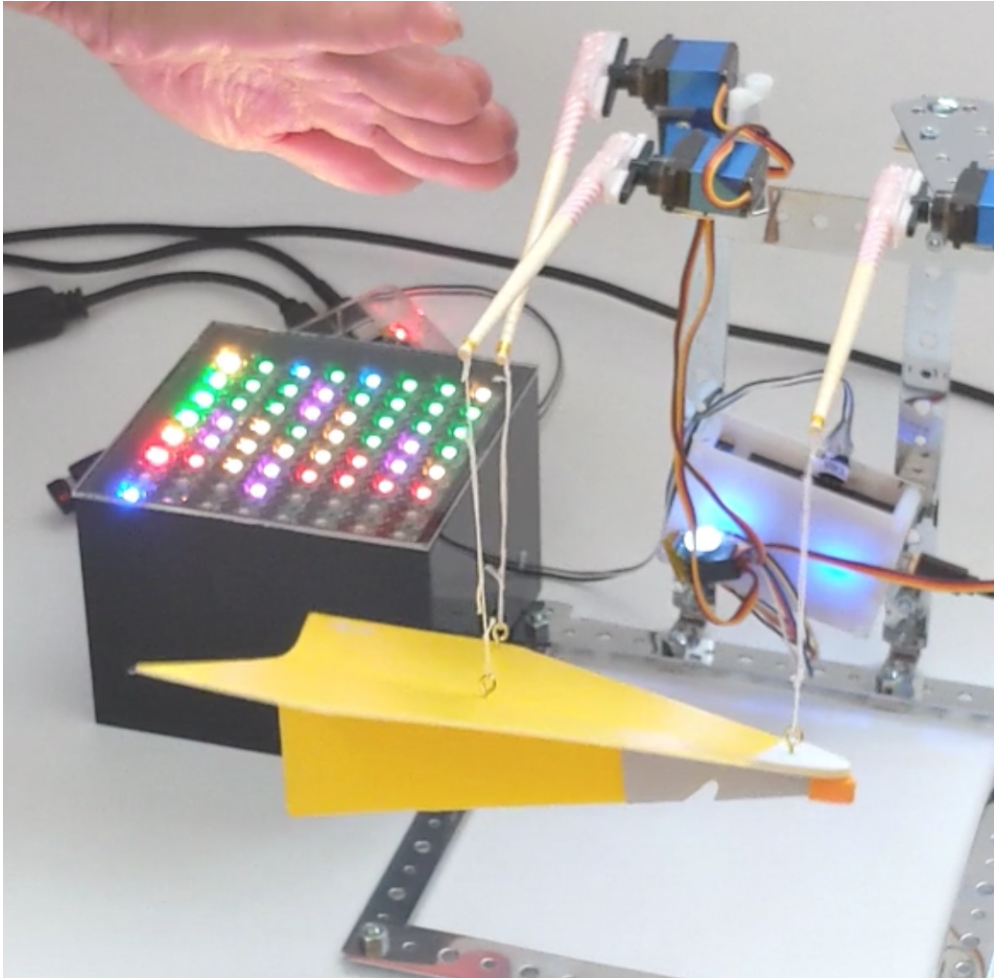


図5-2. キューブ(並列8個)による紙飛行機制御例

## 改訂

ver	日付	変更箇所
1.00	2020/07/06	初版
1.01	2021/02/01	1.重要事項 安全、センサー感度調節
1.02	2021/02/05	4.ソフトウェアとインターフェイス
1.03	2021/07/01	4-1.SPIインターフェイス設定
1.04	2021/09/17	1.3センサーの感度調節
		3-1.本体基板
		3-2.本体基板ピンアウト(JP1)
		3-7.センサー感度調整方法
		4.ソフトウェアとインターフェイス
1.20	2023/08/12	4-2.ADコンバータ読み取り手順
		4-3.SPI-read ADコンバータの読み取り
		4-4.CLEDデータ転送方法
		4-5.CLEDコード