

```
#!/bin/env -S marp -p
```

# ハードウェア講習 第3回

# index

- printデバッグ
- ボタンを読む
- センサを読む
- 7セグ

**printデバッグ**

# printデバッグ

- ArduinoのSerialクラス
- USB経由で文字列等のデータを送受信できる
- 115200はbaudレート
  - 送受信両方の環境で合わせる必要がある

```
void setup(){Serial.begin(115200);}
void loop(){Serial.printf("Hello %d\n",millis());}
```

- このサンプル書くと多分書き込みができなくなる
- platformio.ini に upload\_flags=--no-stub を追加
  - 書き込みにブートローダを使わないオプション

**ボタンを読む**

# ボタンを読む

- ボタンの両端をIO,GNDに結線してある
- IOにはVCCとの間に抵抗を挟んでおく機能がある(PULLUP)
  - ボタンが押されている
    - IO,GND間は0 ohm: IOはGNDと等電位
  - ボタンが押されていない
    - IO,GND間はInf ohm: IOはVCCと等電位
- PULLUP無しではIOが宙に浮いた状態 電位が定まらない

```
void setup(){Serial.begin(115200);pinMode(4, INPUT_PULLUP);}
void loop(){Serial.printf("pin 4 = %d\n", !digitalRead(4));delay(100);}
```

**センサを読む**



# センサを読む

- フォトトランジスタ
  - 明るいと電流を流す
  - 抵抗で電流を電圧に読み替える
- サーミスタ
  - 熱いと抵抗が大きくなる
  - 抵抗値をそのまま分圧に用いて電圧を得る

```
void setup(){Serial.begin(115200);pinMode(6,INPUT);}
void loop(){Serial.printf("pin 6 = %d\n",analogRead(6));delay(100);}
```

# ESP32C3でdigitalReadに対応しているピンはIO 0~5 だけ

Q. IO6,7にセンサ類がついてますが?

A. みなさんもデータシートを読む癖を付けましょう

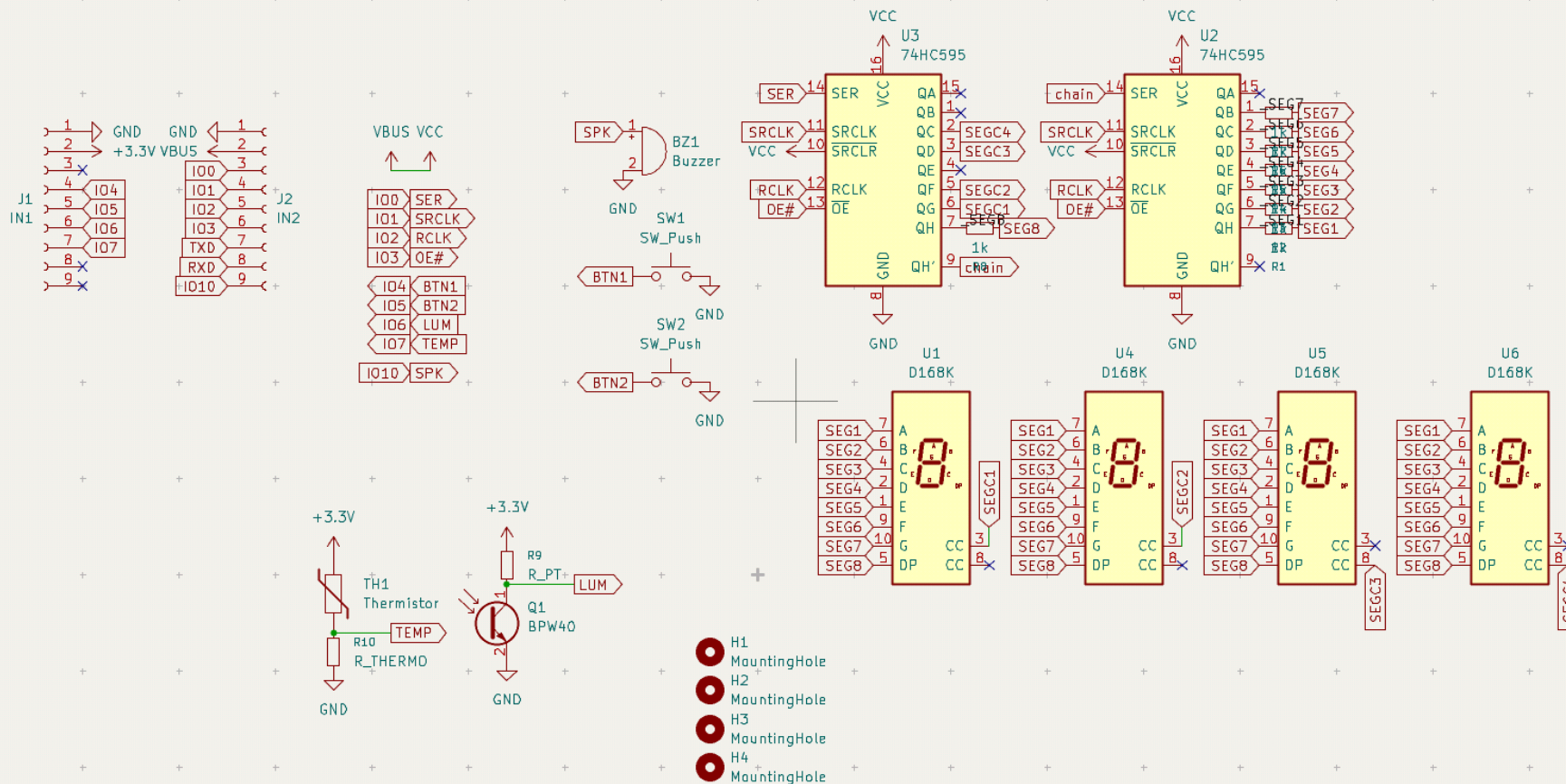
Q. 使えないってことですか?

A. ごめんなさい

7セグ

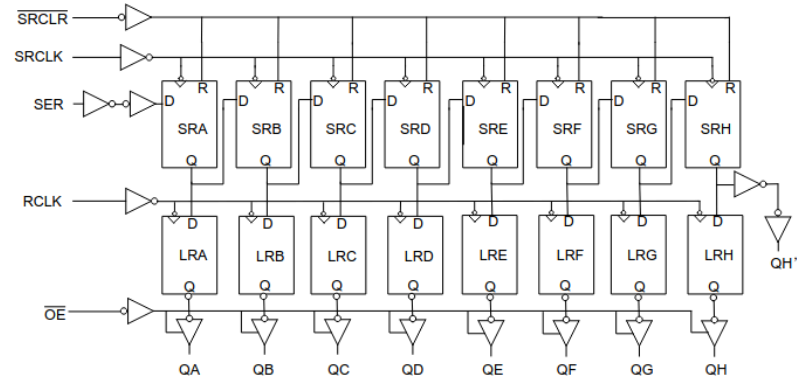
# 7セグ

- 74HC595
  - シフトレジスタ
- OSL10561-LW
  - 7セグ カソードコモン

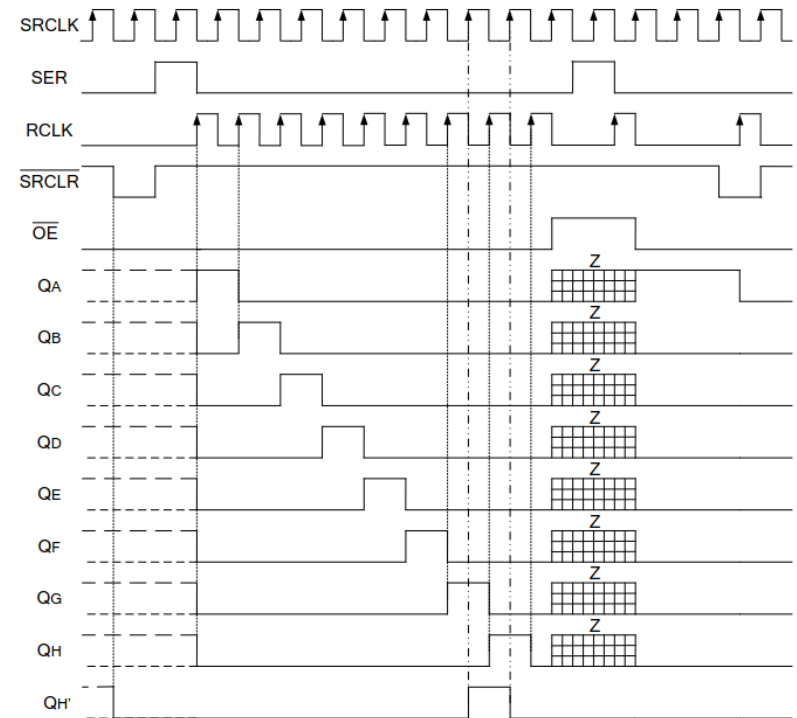


- H1 MountingHole
- H2 MountingHole
- H3 MountingHole
- H4 MountingHole

### ■ LOGIC DIAGRAM



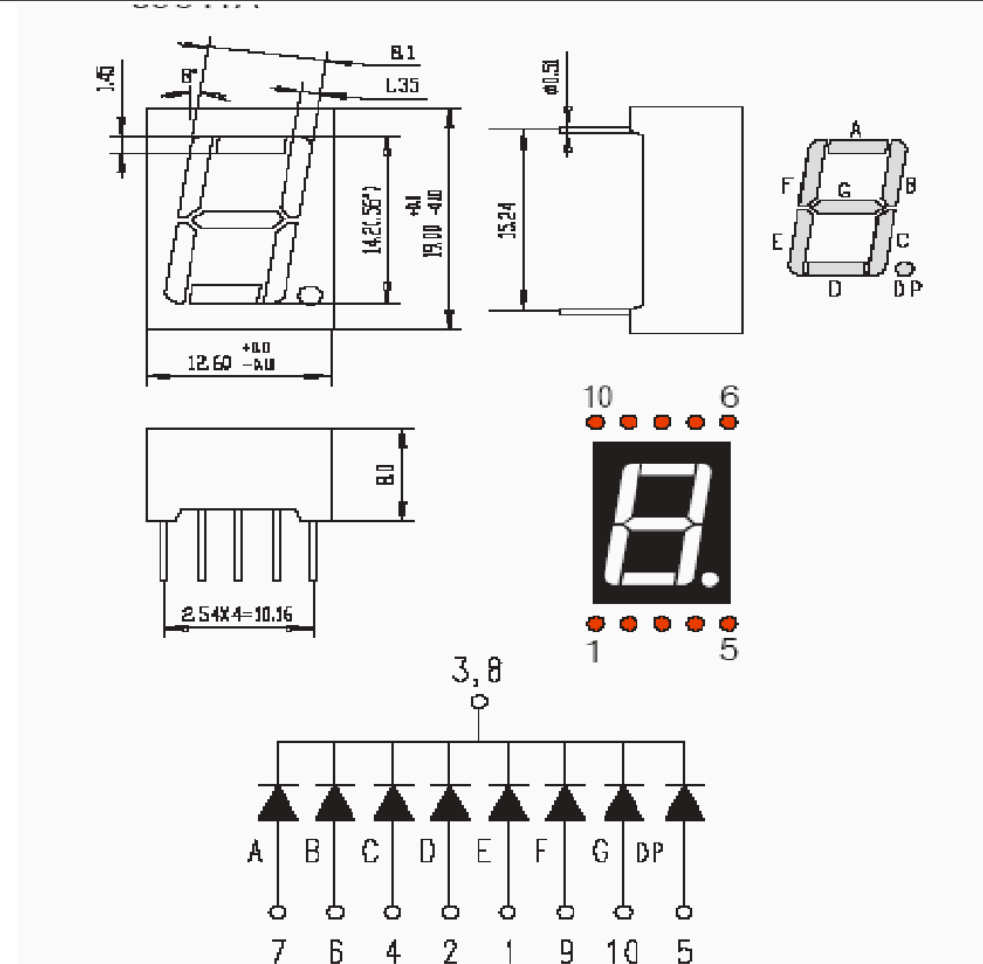
### ■ TIMING DIAGRAM



## OSL10561-LX (Common Cathode type)

Note:

- 1, Unit : mm (Tolerance:  $\pm 0.25$  mm unless otherwise noted)
- 2, The slope angle of any PIN may be  $\pm 5.0^\circ$  Max



- データの流れる方向 >>> \_43\_21d\_FEDCBA

```
#define PIN_SER 0
#define PIN_SERCLK 1
#define PIN_RCLK 2
#define PIN_OE_ 3
void setup(){
    pinMode(PIN_SER, OUTPUT);digitalWrite(PIN_SER, HIGH);
    pinMode(PIN_SERCLK, OUTPUT);digitalWrite(PIN_SERCLK, HIGH);
    pinMode(PIN_RCLK, OUTPUT);digitalWrite(PIN_RCLK, HIGH);
    pinMode(PIN_OE_, OUTPUT);digitalWrite(PIN_OE_, LOW);
    analogWrite(PIN_OE_, 200);

    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00111111);
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00110101);
    digitalWrite(PIN_RCLK, LOW);digitalWrite(PIN_RCLK, HIGH);

}
void loop(){delay(100);}
```



- 同時に1種類の文字しか点灯できない
  - 人間の知覚できない速度で各桁毎に表示する => ダイナミック点灯

```
void loop(){
    // 0.
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00111111);
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00110101);
    digitalWrite(PIN_RCLK, LOW); digitalWrite(PIN_RCLK, HIGH);
    // 1
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00000110);
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00110010);
    digitalWrite(PIN_RCLK, LOW); digitalWrite(PIN_RCLK, HIGH);
    // 2
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b01011011);
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00100110);
    digitalWrite(PIN_RCLK, LOW); digitalWrite(PIN_RCLK, HIGH);
    // 3
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b01001111);
    shiftOut(PIN_SER, PIN_SERCLK, LSBFIRST, 0b00010110);
    digitalWrite(PIN_RCLK, LOW); digitalWrite(PIN_RCLK, HIGH);
}
```

- 3がEになった
- 7セグを逆さにつける設計にしてある
- ソフトで逆さにする
- GFEDCBA をGCBAFED にする

```
uint8_t flip(uint8_t x){return (x&0xc0)|((x&0x38)>>3)|((x&0x03)<<3);}
```

- 数を数字で表示するために文字データを定義しておく

```
uint8_t num[16]={// 0bABCDEFGd  
                0b11111100, // 0  
                0b01100000, // 1  
                0b11011010, // 2
```

...

```
shiftOut(PIN_SER, PIN_SERCLK, MSBFIRST, num[x]&0xfe);  
shiftOut(PIN_SER, PIN_SERCLK, MSBFIRST, 0b00101100);
```

- for文で配列から4桁表示する

```
uint8_t disp[4]={0,1,2,3};
void flush(){for(uint8_t i=0;i<4;++i){
    shiftOut(PIN_SER,PIN_SERCLK,MSBFIRST,(i&2?flip(num[disp[i]]):num[disp[i]])&0xfe);
    shiftOut(PIN_SER,PIN_SERCLK,MSBFIRST,((uint8_t[]){0x2a,0x4a,0x64,0x68})[i]);
    digitalWrite(PIN_RCLK,LOW);digitalWrite(PIN_RCLK,HIGH);
    delay(1);
}}
void loop(){flush();}
```

- このままではloopに他の処理が書けない
  - 書くと描画が止まる
  - ダイナミック点灯は常に描画を回す必要がある
- マルチタスクする
  - ESP32のArduinoではFreeRTOSを使える

```
TaskHandle_t *h_flush;  
  
void setup(){  
    ...  
    xTaskCreateUniversal(flush, "flush", 1024, NULL, 1, h_flush, CONFIG_ARDUINO_RUNNING_CORE);  
}
```

おわり