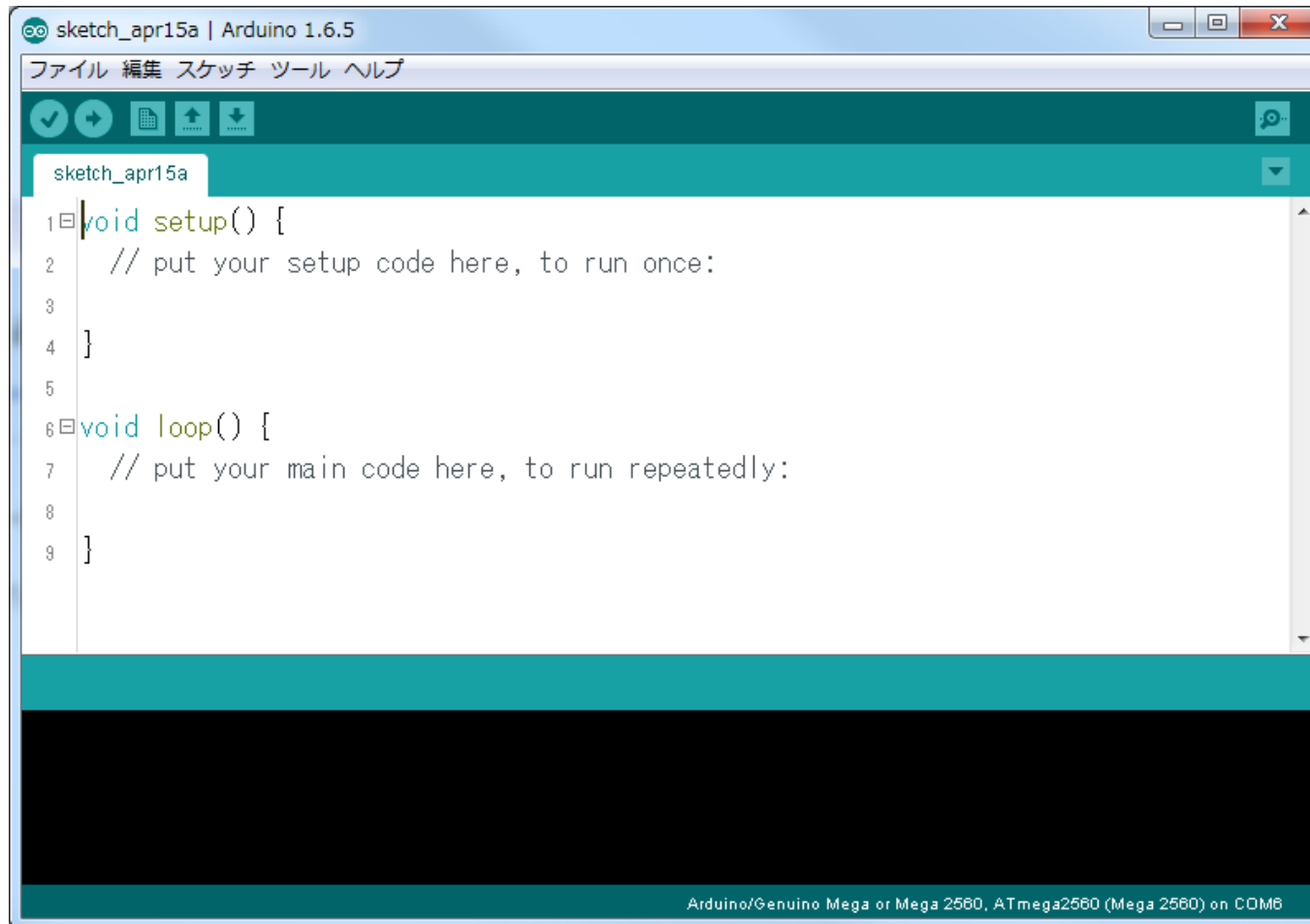
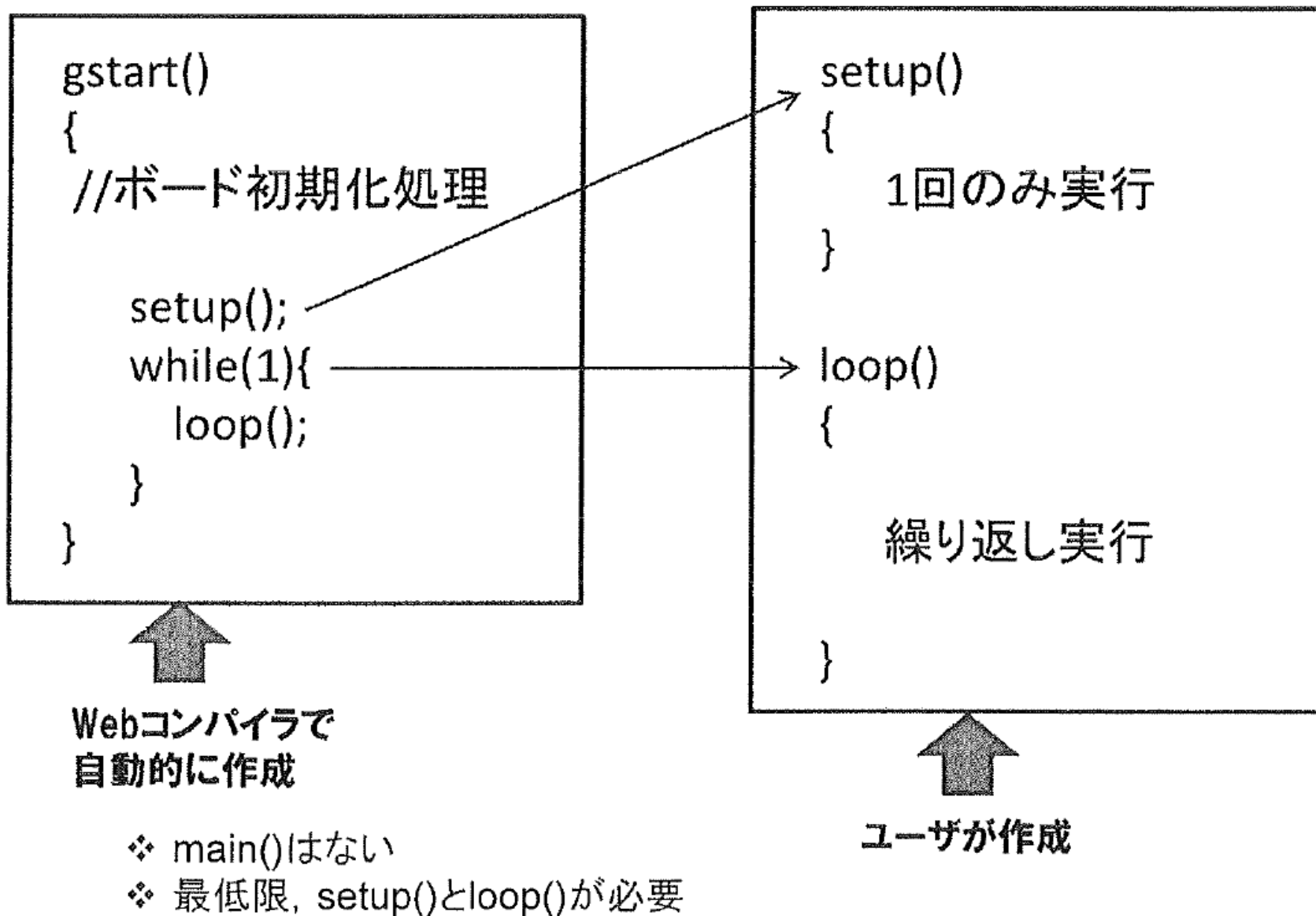


Arduinoによる基礎演習



ソースプログラムの構造を理解する



Arduino日本語リファレンス C++

https://garretlab.web.fc2.com/arduino_reference/index.html

<http://www.musashinodenpa.com/arduino/ref/>

Arduino 日本語リファレンス

Arduino 言語

Arduino言語はC/C++をベースとしており、C言語のすべての構造と、いくつかのC++の機能をサポートしています。また、AVR Libcにリンクされていて、その関数を利用できます。

[setup\(\)](#)
[loop\(\)](#)

制御文

[if](#)
[if else](#)
[switch case](#)
[for](#)
[while](#)
[do while](#)
[break](#)
[continue](#)
[return](#)
[goto](#)

基本的な文法

[.\(セミコロン\)](#)
[{}\(波カッコ\)](#)
[コメント](#)
[#define](#)
[#include](#)

算術演算子

[+-*/](#)
[%\(剰余\)](#)
[=\(代入\)](#)

比較演算子

[== < > <= >=](#)

← → ← → ← → ← →

標準ライブラリ

Arduinoのライブラリは、あらかじめIDEに付属する標準のライブラリと、コミュニティのメンバーから寄稿されたライブラリの2種類があります。寄稿されたものはユーザーが個々にダウンロードしてインストールする必要があります。

ライブラリの使い方

EEPROM

Arduinoボードが搭載するマイクロコントローラはEEPROMと呼ばれるメモリを持っています。EEPROMは(まるで小さなハードディスクのように)電源を切っても内容が消えません。その容量は機種によって異なり、ATmega168は512バイト、ATmega328は1KB、ATmega1280とATmega2560は4KBです。このライブラリはEEPROMに対する書き込みと読み込みを可能にします。

[EEPROM.read\(address\)](#)
[EEPROM.write\(address, value\)](#)

SoftwareSerial

ソフトウェアシリアルライブラリはArduinoボードの0~1番以外のピンを使ってシリアル通信を行うために開発されました。本来ハードウェアで実現されている機能をソフトウェアによって複製したので、SoftwareSerialと名付けられました。

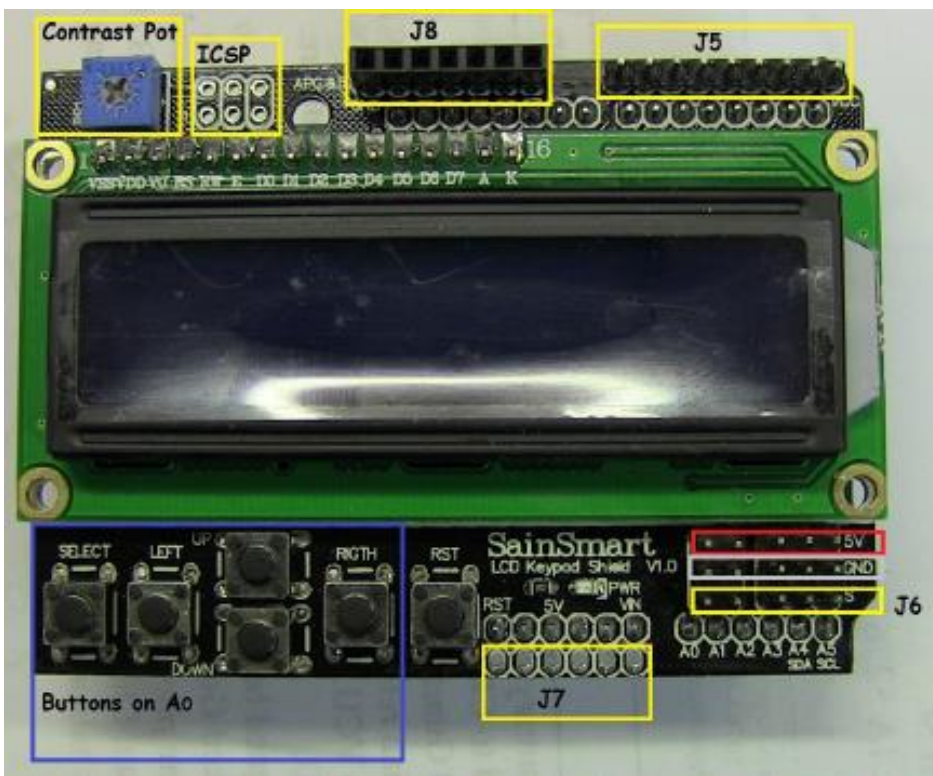
ソフトウェアシリアルのサンプルコード

[SoftwareSerial\(rxPin, txPin\)](#)
[SoftwareSerial.begin\(speed\)](#)
[SoftwareSerial.available\(\)](#)
[SoftwareSerial.isListening\(\)](#)
[SoftwareSerial.overflow\(\)](#)
[SoftwareSerial.read\(\)](#)
[SoftwareSerial.print\(data\)](#)
[SoftwareSerial.println\(data\)](#)
[SoftwareSerial.listen\(\)](#)
[SoftwareSerial.write\(data\)](#)

Stepper

ユニポーラおよびバイポーラのステッピングモータをコントロールするためのライブラリです。このライブラリを利用するには、ステッピングモータと制御のための適切なハードウェアが必要です。

Sain Smart LCD-Keypad Shield v1.0 Front - Connector Pinout



Connector Pins Listed From Left to Right

Connector J5

Shield	Arduino
1	D13
2	D12
3	D11
4	D3
5	D2
6	D1
7	D0
8	Gnd
9	5V

Connector J6

Shield	Arduino
1	A1
2	A2
3	A3
4	A4
5	A5

Connector J7

Shield	Arduino
1	Rst
2	3.3V
3	5V
4	Gnd
5	Gnd
6	Vin

Connector J8

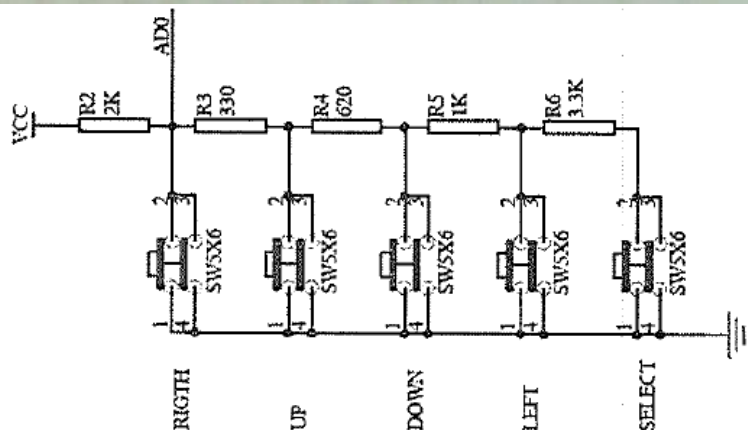
Shield	Arduino
1	Gnd
2	5V
3	N/C
4	D1
5	D0
6	N/C
7	N/C

ICSP (Left to Right – Top to Bottom)

Pin 2: Vcc/5V
Pin 1: D12/MISO

Pin 4: D11/MOSI
Pin 3: D13/SCK

Pin 6: Gnd
Pin 5: Rst

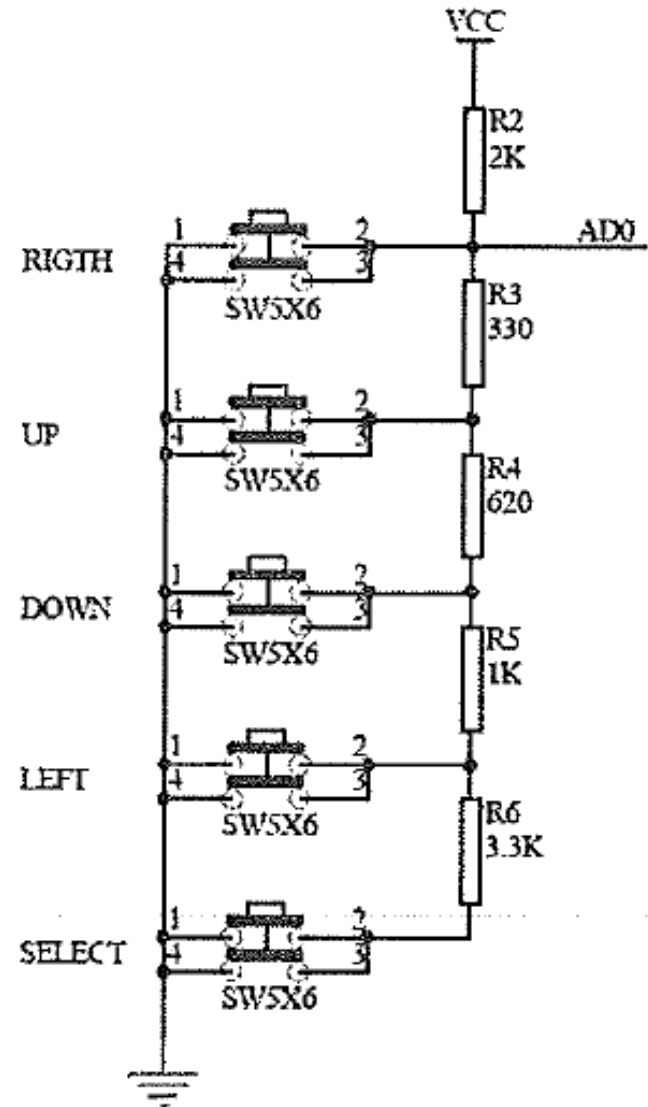


プログラム例

5つのスイッチ (L,R,U,D,S)の動作を表示

```
//*****  
// example for Keypad operation  
//*****
```

```
#include <LiquidCrystal.h>  
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);  
int key_action;  
int k, input;  
char keypad[6] = {'R', 'U', 'D', 'L', 'S', 'N'};  
#define RIGHT 0  
#define UP 1  
#define DOWN 2  
#define LEFT 3  
#define SELECT 4  
#define EVENTNON 5  
  
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
    lcd.begin(2, 16);  
    lcd.clear();  
}
```



Keypad回路図

```

void loop() {
  lcd.clear();
  key_action = Keypad();
  lcd.setCursor(0, 0);
  lcd.print(" Selected key: " );
  lcd.setCursor(0, 1);
  lcd.print(" Lev=");
  lcd.print(input);
  lcd.print(",K=");
  lcd.print (key_action);
  lcd.print(", ");
  lcd.print(keypad[key_action]);
  delay (1000);
}

```

```

int Keypad(){
  // KEYPAD_RIGHT 0
  // KEYPAD_UP 1
  // KEYPAD_DOWN 2
  // KEYPAD_LEFT 3
  // KEYPAD_SELECT 4
  // KEYPAD_NONE 5

  int NUM_KEYS = 5;
  int adc_key_val[5] = {30, 150, 360, 535, 760 };//閾値電圧設定
  input = analogRead(0); // AD出力1023 = 5V
  Serial.print ("analog value=" );
  Serial.print(input);
  for (k = 0; k < NUM_KEYS; k++){
    if (input < adc_key_val[k]) {
      Serial.print(", K= ");
      Serial.println(k);
      return k;
    }
  }
  if (k >= NUM_KEYS)
    k = 5; // No valid key pressed
  return k;
}

```

デバッグの方法

■ プログラムのデバッグ

- パソコンに接続したUSBを使います。

■ ユーザプログラムにprint文を追加

- 初期化はポートを指定しない

■ パソコンはTeraTermなどの通信ソフトを実行

```
void setup()
{
    Serial.begin(9600); //初期化
}
```

```
void loop()
{
    //不具合を見つけない箇所に表示を挿入
    Serial.println("1");
    //
}
```

Serialのクラス

write: シリアル通信ポートに文字列やデータを出力する。

print: シリアル通信ポートに文字列を出力する。

println: シリアル通信ポートに文字列を改行付きで出力する。

演習

- 1) 1～9 9までの足し算を行いLCDに表示させる。

1+2	=	3
2+3	=	5

- 2) ボタン動作で計算のスタート，ストップを行う。

RIGHT : スタート (継続)

LEFT : ストップ

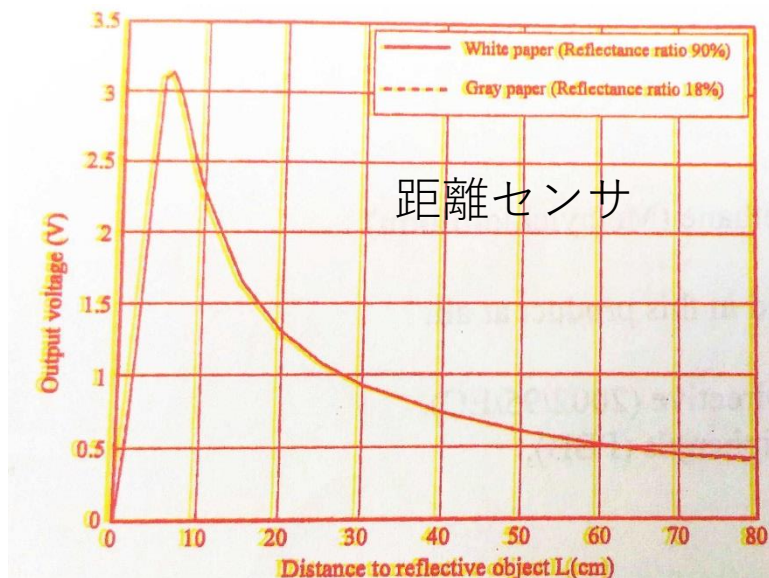
(UP : リセットしてからスタート)

- 3) 距離計

シャープ製距離センサーを用いて距離を測定し表示させるプログラムを作る
AD変換された値を電圧値に直し下左図から距離に変換する。

- 4) 温度計

サーミスターを用いて抵抗値から温度を算出



$$\text{距離(cm)} = (6787 / (\text{AD値} - 3)) - 4$$

温度センサ

$$R = R_0 \exp \left\{ \beta \left(\frac{1}{T} - \frac{1}{T_0} \right) \right\}$$

$$R_0 = 10\text{k}\Omega, T_0 = 297\text{K}, \beta = 3380\text{K} \quad (25\text{--}50^\circ\text{C})$$

