

講習会

Arduino

9回目

変数を作る

# 目的

- 関数として処理をするだけでなく、変数として数値を出せるようにする。
- 変数を宣言するときに処理を書くことで簡単な数値として取り出せる。

# 関数紹介

- return

# return

- 関数を終了し、必要ならば、呼び元の関数に値を返す。
- return;
- もしくは
- return 式;

# 例

- センサーからの入力と閾値とを比較する関数。
- `int checkSensor(){`
- `if (analogRead(0) > 400) {`
- `return 1;`
- `else{`
- `return 0;`
- `}`
- `}`

- バグを含む多くのコードをコメントアウトせずにテストするときにも簡単に利用できる。

```
void loop(){
```

```
// コードのアイデアをここで試す。
```

```
return;
```

```
// 残りのうまく動かないスケッチ
```

```
// この部分のコードは実行されない。
```

```
}
```

# 例

- int a() {
- if(digitalRead(sw) == HIGH){
- return 1;
- }
- else{
- return 0;
- }
- }

- スイッチswが入力されているときに1を返して、それ以外では0を返す。

- 次の例はLCD\_keypad\_shieldのスイッチの入力に関する変数の設定である。

- `int read_LCD_buttons(){`
- `adc_key_in = analogRead(0);    // read the value from the sensor`
- `// my buttons when read are centered at these values: 0, 144, 329, 504, 741`
- `// we add approx 50 to those values and check to see if we are close`
- `if (adc_key_in > 1000) return btnNONE; // We make this the 1st option for speed reasons since it will be the most likely result`
- `// For V1.1 us this threshold`
- `if (adc_key_in < 50)   return btnRIGHT;`
- `if (adc_key_in < 250) return btnUP;`
- `if (adc_key_in < 450) return btnDOWN;`
- `if (adc_key_in < 650) return btnLEFT;`
- `if (adc_key_in < 850) return btnSELECT;`
- `return btnNONE; // when all others fail, return this...`
- `}`

# 変数の使い方

- `lcd_key = read_LCD_buttons();`
- 宣言した変数を使う
- 前のページで返した数字が出てくる。
- 上ならUP、RIGHT、DOWN、LEFT、SELECTのどれかになる。

- 作った変数を別の変数に代入するときは同じ型で宣言すること

- `int a(){`
- `//処理`
- `}`

- `int b;`
- `b = a();`

- `unsigned`も同じ

- 宣言する型が違うためaの数値を全部賄いきれない。

- `long a(){`

- `//処理`

- `}`

- `int b;`

- `b = a();`

- 使用できる数値の範囲は2回目の「変数と演算」を確認

- 関数同様引数を使うことができる。
- `int a(int b,int c){`
- `int d;`
- `d = b - c;`
- `return d;`
- `}`

# 関数との違い（関数の場合）

- void a(){
- //処理
- }
- void loop(){
- a();
- }

- 数値を返さないため別の変数に代入をすることはできない。
- モータの動作など返す必要がない場合に使う。

# 関数との違い (変数の場合)

- `int a(){`
- `//処理`
- `}`
- `void loop(){`
- `int b;`
- `b = a();`
- `}`

- 数値を返すため別の変数に代入することができる。
- センサー値など数値を返すときに使われる。
- aが処理次第では1でも2でも、returnで返した数値として扱われる。
- return 1で終わればaは1となる。

# 終わりに

- 関数同様宣言をして処理の中に入れるだけで、何回も同じ文章を書かなくても値を返してくれる。
- 変数を作ったら処理の中に入れてみよう。
- マイコンカーでもポテンシヨメータ等センサー値の部分で使われている。例0angle（電源を入れたときの角度を基準とする）
- 次回はシリアル（パソコンに表示）を取り扱う。

# 課題

- 数学的関数で紹介したminとmaxを自分で作ってみる。
- 同じ動作をするか確認