Arduino/Suwano をはじめよう@KYOTO-U　No.1

2012 年 4 月 26 日
京都大学 学術情報メディアセンター
喜多　一

今回の学習目標

- ArduinoUno を PC に接続する方法を覚える.
- Arduino の統合開発環境(IDE) を起動する方法を覚える.
- Arduino IDE で簡単な例を用いて，ベリファイ，アップロード，実行する方法を覚える.
- 簡単なエラーを経験する.
- プログラムでの命令の逐次実行を知る.

０．　作業の心得

この授業ではプログラムを書いたり，電子回路を作成したりします．人が行う作業には必ず誤りが伴います．これを極力，無くすことで安全で効率的な作業を実現したいと思います．そこで，この授業では原則として２人１組で作業をしてもらいます．どちらかの任せるのではなく，**片方が作業すればその内容を他方が確認する**ようにします．どの作業でも，**どちらの人でも実行できるように交代して作業**してください．

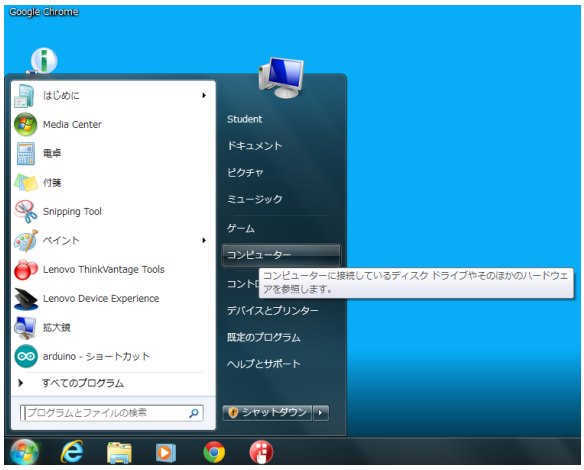
Arduino Uno の接続

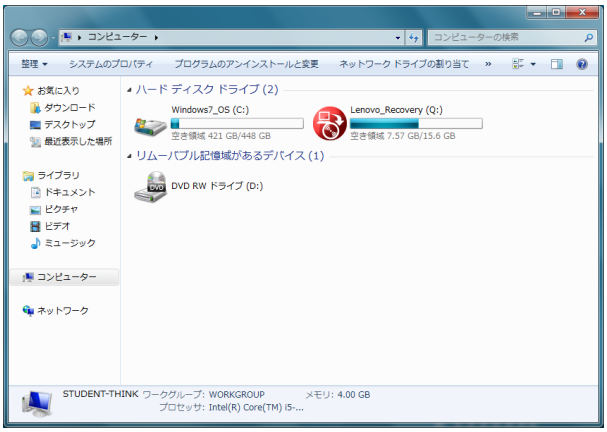
Suwano の接続

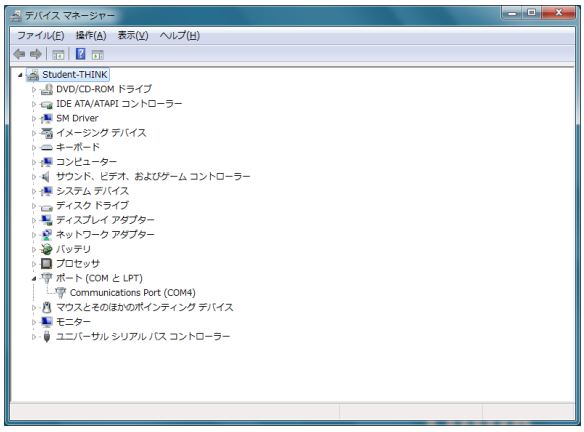
１．　準備

まず、以下の手順で Arduino/Suwano を使えるようにします。

a) 授業用 PC を起動し、Student でログインします。
b) Arduino Uno あるいは Suwano を USB ケーブルで PC のキーボード右側の USB ポートに接続します。写真参照
c) Arduino/Suwano の COM ポートの確認
   - スタートボタンを押し，メニューの「コンピュータ」を選びます.
   - 「コンピュータ」のウィンドウから「システムのプロパティ」を選びます.
   - 「システム」ウィンドウから「デバイスマネージャ」メニューを選びます．管理者権限を要求されますが閲覧するだけなので必要ありません.
   - ポートを選択し Communications Port (COM4) という表示を確認します。COM の後の数字は異なっているかもしれませんが、数字を控えておきます。







d) Arduino IDE (統合開発環境)の起動

デスクトップの adruino　ショートカットをダブルクリックします.

e) Arduino IDE の初期設定

- メニューバーから「Tools」→「Board」→「ArduinoUno」あるいは「Suwano」を選びます.
- また「Tools」→「Serial Port」→「COM4」を選びます. COM のあとの数字は 4 とは限りません. c) (エ)で確認したものであればかまいません.

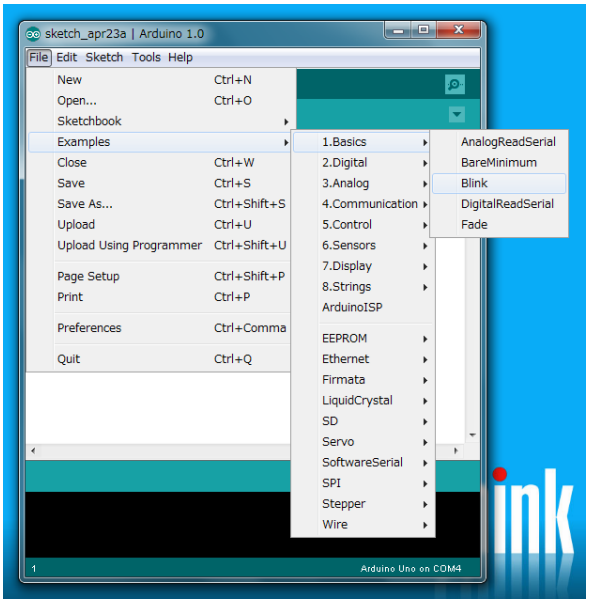
2.　例題の実行

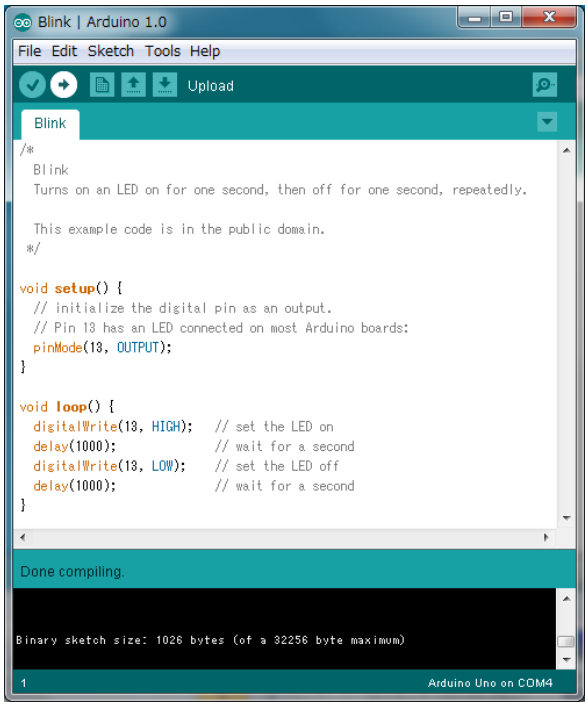
a) File メニューから「Examples」→「1.Basics」→「Blink」を選びます.

b) Blink というプログラムのウィンドウが現れます.

c) ベリファイ（コンパイルとも言います, 一番左のボタン）を行います. 終了すればウィンドウの下側に黒いパネルにメッセージが表れます.

d)





e) ベリファイ（コンパイル）が終了したら隣の矢印ボタンで実行可能プログラムを Arduino に送ります（アップロード）.

f) 転送が済めば Arduino でこのプログラムが自動的に動き始めます. このプログラムでは１秒おきにボード上の LED が点灯と消灯を繰り返します.

３． 例題の変更と再実行

例題の下から４行目の delay(1000); と下から２行目の delay(1000); をそれぞれ delay(1500); と delay(500); に書き換えて，ベリファイとアップロードを行ってみてください．点滅する時間が変化すれば成功です.
注意：使う文字はすべて半角文字です．行末のセミコロン（;）を忘れないでください.

４． エラーを経験する

プログラムを作成しているときにキー入力を誤ったりするエラーにしばしば出会います．ここでは意図的にエラーを体験してみましょう．Blink の例題について以下のことを試みて，ベリファイのときにどのようなメッセージが出るのか確認してみてください.

- 行末のセミコロンをなくす.
- 開いた中括弧を閉じ忘れる.
- 大文字と小文字を間違う.

５．　例題を最初から書いてみる

File メニューで新しいプログラムの作成を行います．内容は Blink の例題と同じものにしますが，コピー&ペースはせずにすべて自分で入力してみてください．

注意：以下の注意を守ってください．

- すべて半角文字で入力します．
- 行末のセミコロン（;）を忘れないでください．
- 中括弧({,})などの記号をまちがわないでください．
- setup, pinMode, OUTPUT, loop, digitalWrite, HIGH, LOW, delay などは大文字，小文字をこの通り入力してください．また綴り誤りがないか確認してください．
- setup(), loop() などの後の() は必要です．
- 文字の色は自動的にシステムはつけます．
- /* で始まり */ で終わる箇所，// から行末までは「注釈（コメント)」と言ってプログラムとしては意味をもちません．人がプログラムを理解するためにつけるものですので，省略することも可能です．ただし，プログラムに適切な注釈を入れることはプログラムの開発において重要なこととされています．

入力を終えたら必ず誤記がないか確認します．確認を終えたらベリファイ，アップロードを実行して動作するかどうか試してください．

ベリファイでエラーメッセージが出たらどこか誤っているはずです．エラーメッセージを参考に内容を確認してください．

６．　プログラムの内容

行	ソースコード	解説
1	// Example 01：Blinking LED	// で始まる行は注釈です．プログラムとしては機能
2		しません．
3	void setup()	setup() という関数（命令群）の定義です．
4	{	{ から } までが関数の中で実行する内容です．
5	pinMode(13, OUTPUT);	13 番目のピンを出力用に使う設定にします．
6	}	
7		
8	void loop()	loop() という関数の定義です．
9	{	{ から} までが関数の中で実行する内容です．
10	digitalWrite(13, HIGH);	LED 番目のピンに高い電圧を出力します．
11	delay(1000);	1000 ミリ秒待ちます．
12	digitalWrite(13, LOW);	LED 番目のピンに低い電圧を出力します．
13	delay(1000);	1000 ミリ秒待ちます．
14	}	

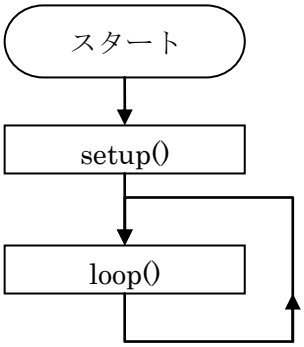
Arduino のプログラムは一連の命令群を並べたもの「関数」を作成することで行います．関数は次のような記述形式をとります．

```
返り値の型 関数名(引数の記述)
 {
   実行する内容
 }
```

上の例の関数 setup は

```
void setup()
 {
   pinMode(13, OUTPUT);
 }
```

返り値の型は void（なにも返さないことを意味します），関数名は setup，引数はなし，実行する内容は pinMode(13, OUTPUT)；です．



フローチャート

Arduino では関数 setup() と loop() は特別な役割を持ちます．Arduino が起動すると，準備のためまず一回だけ setup() と呼ばれる関数が呼び出されます．次に loop() という関数が繰り返し呼び出されます．フローチャートで表すと図のようになります．
上のプログラムではまず関数 setup() の中で関数呼び出し pinMode(13, OUTPUT) により，13 番目のピンを出力用(OUTPUT)に設定します. LED と OUTPUT は pinMode() という関数を呼び出す際に与える引数です．Arduino では 13 番目のピンには予めボード上の LED が接続されています．

つぎに関数 loop() が繰り返し呼び出されるのですが，loop() の中では，まず digitalWrite(13,HIGH) の呼び出しで 13 番目のピンの電圧が高く（5V）設定されます．つぎに delay(1000) の呼び出しで 1 秒（1000 ミリ秒）待ち, digitalWrite(13, LOW) の呼び出しで 13 番目のピンの電圧が低く（0V）設定されます．そして delay(1000) の呼び出しで再び 1 秒（1000 ミリ秒）待ちます．これが繰り返されることで LED が 1 秒おきに点滅を繰り返します．

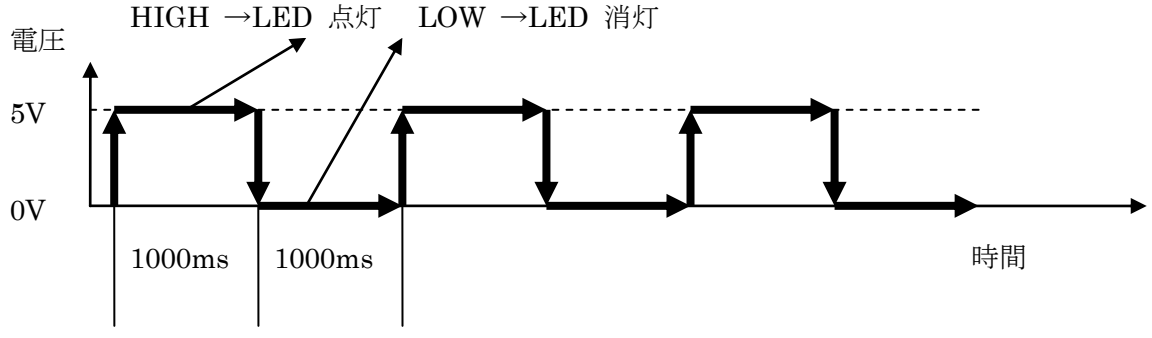


図 LED 点灯のタイムチャート

７．　動作を組み合わせる

プログラムの基本は逐次実行です．上から下へ命令を実行してゆきます．次の図に示すようなタイムチャートで LED が点灯するスケッチ（プログラム）を作成してみてください．

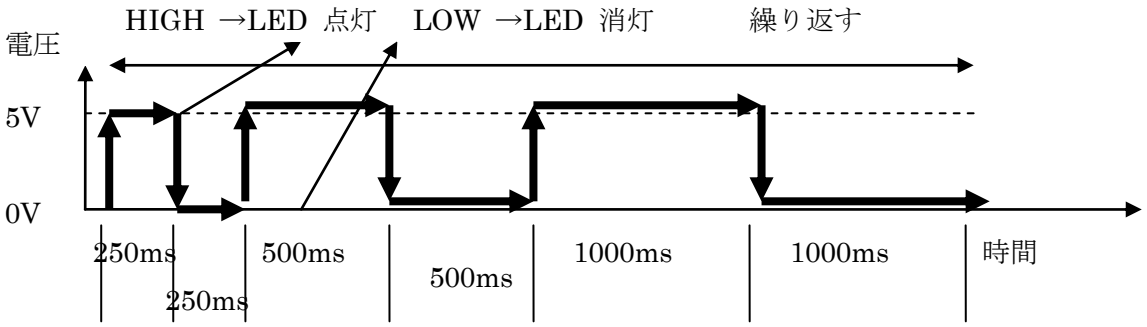


図 LED 点灯のタイムチャート

行	ソースコード	解説
1	// Example 01：Blinking LED	// で始まる行は注釈です．プログラムとしては機能
2		しません．
3	void setup()	setup() という関数（命令群）の定義です．
4	{	{ から } までが関数の中で実行する内容です．
5	pinMode(13, OUTPUT);	13 番目のピンを出力用に使う設定にします．
6	}	
7		
8	void loop()	loop() という関数の定義です．
9	{	{ から} までが関数の中で実行する内容です．
10	digitalWrite(13, HIGH);	LED 番目のピンに高い電圧を出力します．
11	delay(250);	250 ミリ秒待ちます．
12	digitalWrite(13, LOW);	LED 番目のピンに低い電圧を出力します．
13	delay(250);	250 ミリ秒待ちます．
14	digitalWrite(13, HIGH);	次の４行が間隔 500 ミリ秒での点滅です．
15	delay(500);	
16	digitalWrite(13, LOW);	
17	delay(500);	
18	digitalWrite(13, HIGH);	次の４行が間隔 1000 ミリ秒での点滅です．
19	delay(1000);	
20	digitalWrite(13, LOW);	
21	delay(1000);	
22	}	