

Arduino(アルディーノ)互換ボード搭載 SkyBerrino(スカイベリーノ)ロボット セットアップ

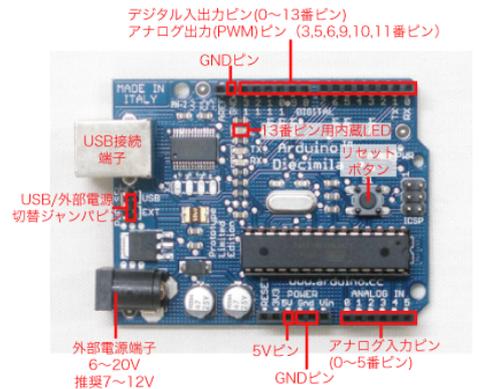


1. Arduino(アルディーノ)とは

2005年高価なロボット開発システムに不満を感じたイタリアの学生たちが、安価で簡単にオープン(自由)なシステムを目的に開発を行った。ハードウェア設計情報や開発環境は無料で公開されている。(オープンソースという)そのために Arduino は自作することも可能で、Arduino 互換機もたくさん販売されている。そのオープンでよりユーザーに身近なシステムを簡単に安価に開発できることから「フィジカルコンピューティング」のためのデバイス(装置)として、世界的に注目のシステムとなっている。

2. Arduinoハードウェア

Arduino は、米国 Atmel 社 AVR マイクロコントローラをベースとしている。マイコンには、ブートローダというソフトが事前にプログラムされているために、プログラムライター不要でシリアル通信で書き込みができる。USB ケーブル1本接続すれば、プログラム書き込み・シリアル通信・電源供給が可能となっている。現在もっとも標準的なものは、Unoという機種である。



3. Arduino統合開発環境(IDE*)

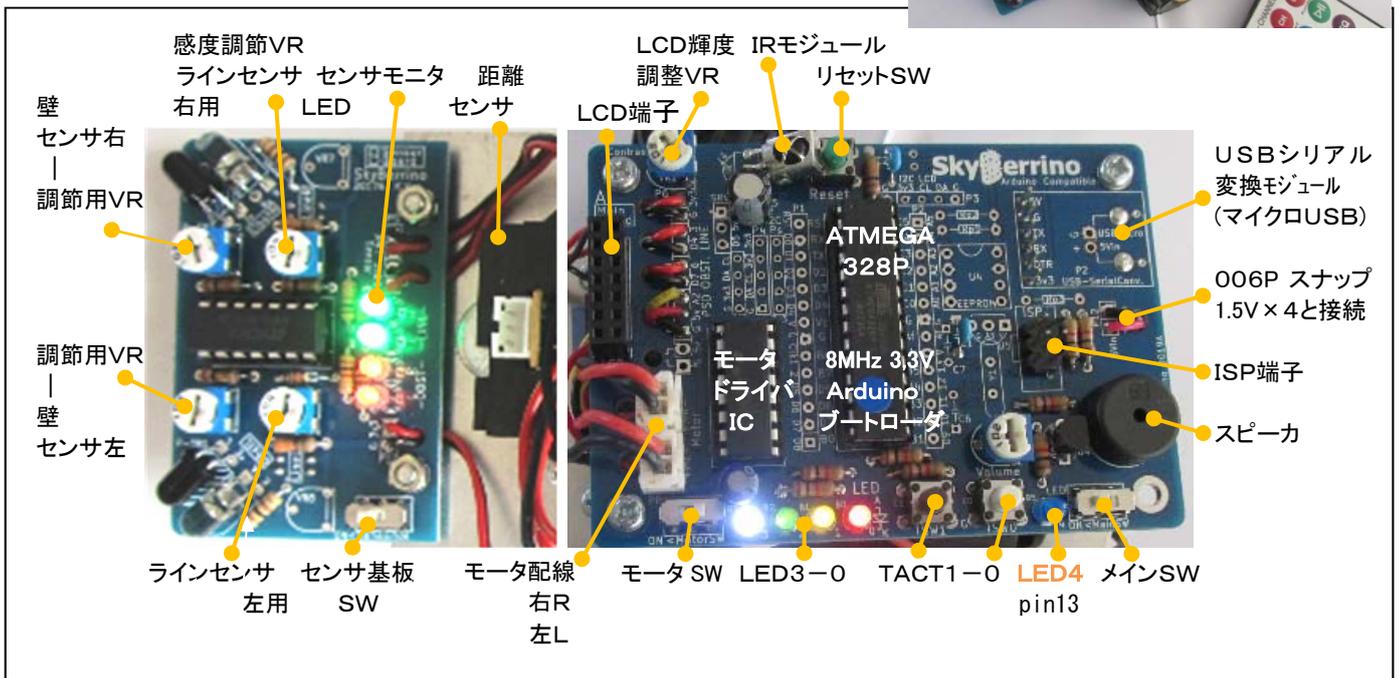
Arduino の統合開発環境 IDE は、オープンソースソフトウェアで開発されている。Processing というプログラミング言語開発環境がベースとなっていて、非常にシンプルな構成となっており分かりやすい。C 言語ライクな言語仕様を基に、数々の便利な関数やライブラリも内蔵追加も可能、ソフトウェア開発に不慣れな初心者でも容易にプログラミングできるよう設計されている。公式サイトから自由にダウンロードできる。
www.arduino.cc*Integrated Development Environment

4. Arduino 互換スカイベリーノ(SkyBerrino)搭載ロボット

SkyBerrino(スカイベリーノ)は、栃木工業高校が作成した Arduino 互換ボードで、そのボードを搭載した組み込み学習用ロボットが右の写真のものである。

※スカイベリーノの名前は、スカイベリー(いちご)+アルディーノの造語からきている。

◆各部品の名称



5. PC と ArduinoIDE の設定(準備)

(Windows11 以降現在では、以下 USB シリアルドライバが標準インストールされている場合があります)

1)PC の管理者でログイン。

2)USB シリアルモジュールドライバ(Silcon Labs 社 <http://jp.silabs.com/>)を PC にインストール。
CP210xVCPInstaller_x86.exe(32bit用) または CP210xVCPInstaller_x64.exe (64bit用)

3) マイクロ USB ケーブルでロボット基板の USB モジュールと PC を接続する。(ロボット本体の電源 ON は不要、接続音になる)

4)デバイスマネージャで図1のようになるのを確認する **COM** の番号

5)Arduino-1.80 以降をインストール。(起動はまだしない)

c:¥Program Files(x86) 内にインストールされる

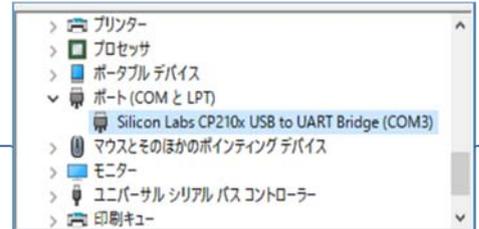


図1

6)セットアッププログラム `SkyBerrino_Setup_20xx.yy.zz.exe` を実行

(栃木工業高校ホームページ電子情報科ページ最下部のダウンロードコーナーより DL 可能)

実行すると、以下のファイルが Arduino-IDE に自動で追加される。

①SkyBerrino 用例題プログラム追加

<IDE-Menu><ファイル><スケッチ例><00SkyBerrino>で 各種の例題ファイルが出る(図2)

②SkyBerrino ボード情報が追加

<IDE-Menu><ツール><マイコンボード>で、SkyBerrino を設定(図3)

Arduino Pro (328P 3.3v 8Mhz) でも OK

③Lcd2.h, IRremocon2.h などライブラリが追加・インストール

④ SkyBerrino 用ブートローダーHEX ファイルが追加

C:¥Program Files¥Arduino¥hardware¥arduino¥avr¥bootloaders¥metaboard 内に追加

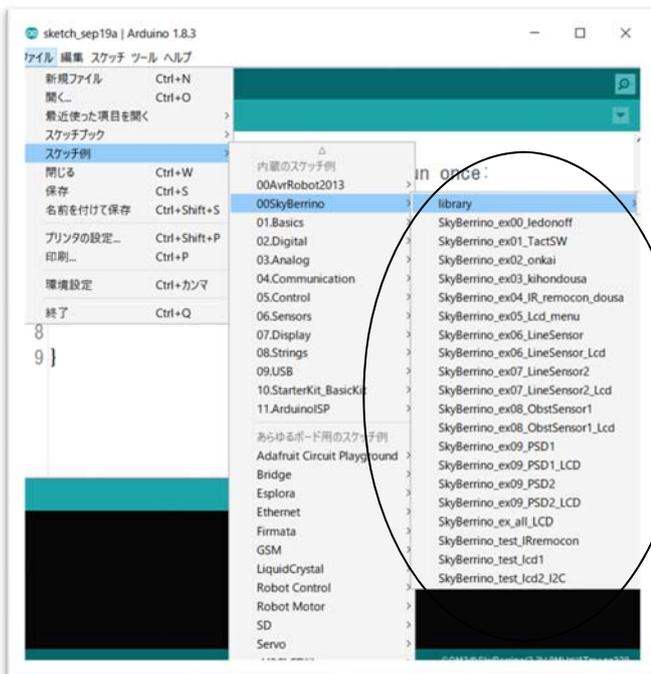


図2

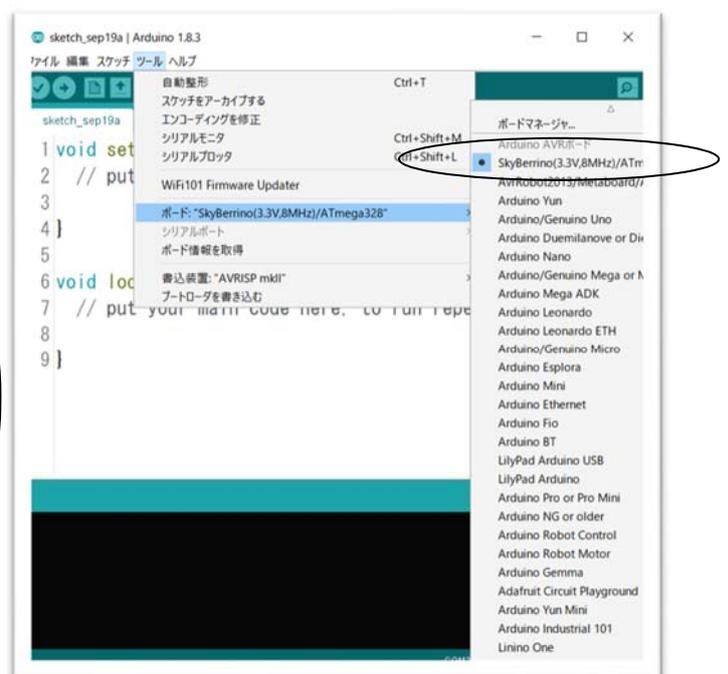


図3

6. サンプルプログラムの転送・確認

(1)ロボット本体と PC を USB ケーブルで接続し、メイン電源 SW)は OFF

(2)IDEを起動し、マイコンボードを **SkyBerrino**、シリアルポートを、前述の番号に合わせる、

(3)IDE の <ファイル><スケッチの例><01Basics>メニュー中の<Blink>などの プログラムを呼出す。

The screenshot shows the Arduino IDE with a Blink program. The code is as follows:

```

//-----
// LED (pin13) を点滅(on/off)させる
//-----

// 初期設定関数 setup()
void setup()
{
  pinMode(13, OUTPUT); // 13pin を出力に設定
}

// 繰り返し関数 loop()
void loop()
{
  digitalWrite(13, HIGH); // LED 点灯 13pin へ H=5v 出力 PORTB=b00100000 と同
  delay(100); // タイマディレイ ms ←(*A) この値を変化させてみる
  digitalWrite(13, LOW); // LED 消灯 13pin L=0v 出力
  delay(100); // タイマディレイ ms =1/1000 秒
}

```

Annotations in the image:

- A callout box points to the `setup()` function with the text: "一度だけ実行される関数" (Function executed only once).
- A callout box points to the `loop()` function with the text: "繰り返し実行 (無限ループ) される関数" (Function executed repeatedly in an infinite loop).

Flowchart titled "Arduino 処理の流れ" (Flow of Arduino processing):

```

graph TD
    Start(( )) --> Setup[setup() 関数]
    Setup --> Loop[loop() 関数]
    Loop --> Loop

```

(4)IDEの検証・書き込ボタンでマイコンに書き込む。(→Warningメッセージは無視できる)

(5)転送終了後、ロボットの<LED4> が点滅実行しているか確認する。

(<LED4>は、Arduino-13 番 pin 接続)

以下の関数や命令を理解する。 ↓説明を記入

```

void setup()関数    ....( )
void loop()        ....( )
pinMode(pin, OUTPUT) ....( )
digitalWrite(pin, HIGH) ....( )
delay( 100)        ....( )

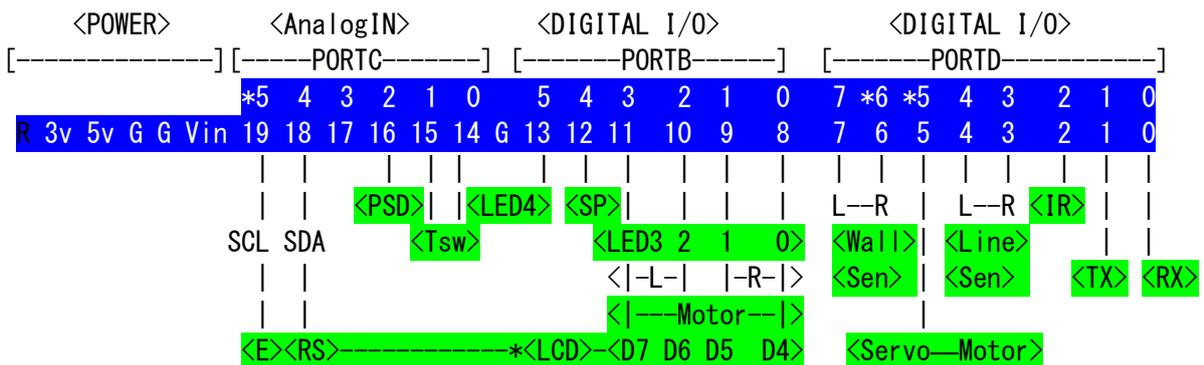
```

(6)上記(*A)のタイマディレイ値を変化させて、再度ロボットの<RESET>ボタンを押して、書き込み・実行してみる。(7)IDEメニュー<ファイル><名前を付けて保存>で、自分の学年フォルダ中に ROBOT というフォルダを作り、

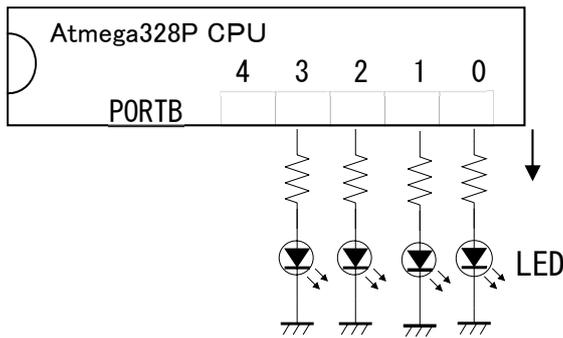
ex0_sample.ino というファイル名で保存する。

今後は、このファイルを基にして(ひな型)にして、プログラムを作成していく。

7. SkyBerrino ピンアサイン



例題. LEDx4つを チカチカさせる



LED4個は、左図のように接続されている。このようにマイコンと外部を接続(入力・出力)するところを**ポート**といい、このマイコンにはポートB、C、Dの3つがある。このポートの各ビットに1を出力すれば、5Vが出力されLEDに電流が流れ点灯することになる。たとえば、PORTB の bit3,2,1,0 に1111を出力指定すれば、4つすべて点灯する。0000を出力指定すれば、すべて消灯する。

しかし、これを連続に実行すると、マイコンの実行が速くて、点灯時間が短く人間の目には見えない。点灯時間・消灯時間を長くする(時間待ち)必要がある。この処理を**タイマ(遅延)処理**といい、ここでは delay(t) 等の関数を使用する。

タイマ関数 delay(t) delay_Microseconds(t)

<LED>	<流れ>	<2進数>	<16進数>	<Arduino 言語での書き方>
●●●●	↓ 点ける	PORTB←1111 ()	(PORTB=0x)
	↓ ちょっと待つ			()
○○○○	↓ 消す	PORTB←0000 ()	(PORTB=0x)
	↓ ちょっと待つ			()

■例題 LED点滅プログラム ex1_led_onoff.ino

```

void setup()
{
  // 各ポートの入出力の設定 (出力:1 入力:0)
  DDRB=0x3F; // PORTB 設定 11 1111 *pinMode()でもできるが長くなる
  DDRC=0x30; // PORTC 設定 11 0000
  DDRD=0x22; // PORTD 設定 0010 0010
  PORTB=0x00; // 初期代入
  PORTC=0x03; // 14,15pin 内部 PullUP にする (TACT-SW 用)
  PORTD=0x04; // 2pin 内部 PullUP にする (IR 用)
  ledonoff(); // ledonoff()関数を呼び出す (←動作チェック用にいつも入れておく)
  // ←(*B) 今後は、ここで新しく作成した関数を呼び出す
}

// ←(*A) 今後はここ setup()関数のすぐ後に、新しい関数を作成・追加していく。消さない

void loop() // 繰り返し関数 loop() ロボット停止処理 (以降変更しない)
{
  ledonoff(); // LED点滅を無限繰り返す (ロボット・モータ停止状態と同じ)
}

void ledonoff()
{
  PORTB=0x0F; // LED x 4 ON 0000 1111
  delay(500); // ちょっと待つ 0.5秒=500ms ←ココの数値を変えてみる
  PORTB=0x00; // LED x 4 OFF 0000 0000
  delay(500); // ちょっと待つ 0.5秒=500ms ←ココの数値を変えてみる
}

```