

IoTプログラミング教材シリーズ

チャレンジボード SB-07 ガイドブック

初心者から初級・中級者向け

カーネルキャリアスクール

2019年7月

はじめに（必ずお読みください）

本教材はIoTプログラミング初心者から初級レベルの方を想定した、プログラミング的思考トレーニングのための教材です。難しい機器の組立や電気配線を必要とせず。サンプルプログラムを使って、表示やセンサーなどの動きを見ながらプログラミングを体験できます。

本教材のサンプルプログラムを体験した後は、オリジナルなプログラムの作成にチャレンジしましょう。はじめは、プログラムを改造するところから始めます。一箇所ずつの改造からはじめてみましょう。エラーが出たり、うまくいかない時は、直前の状態に戻して考察しましょう。

プログラミング言語(C言語)の様々なコマンドや制御については、関連参考図書やインターネットの解説情報で自習することも大切です。自分で調べる、考える、失敗することで、スキルをアップができます。うまくいかない時、わからない時が、チャンスです。

グローバルなテーマですから関連情報が英語で提供されている場合があります。英語も併せて学びましょう。

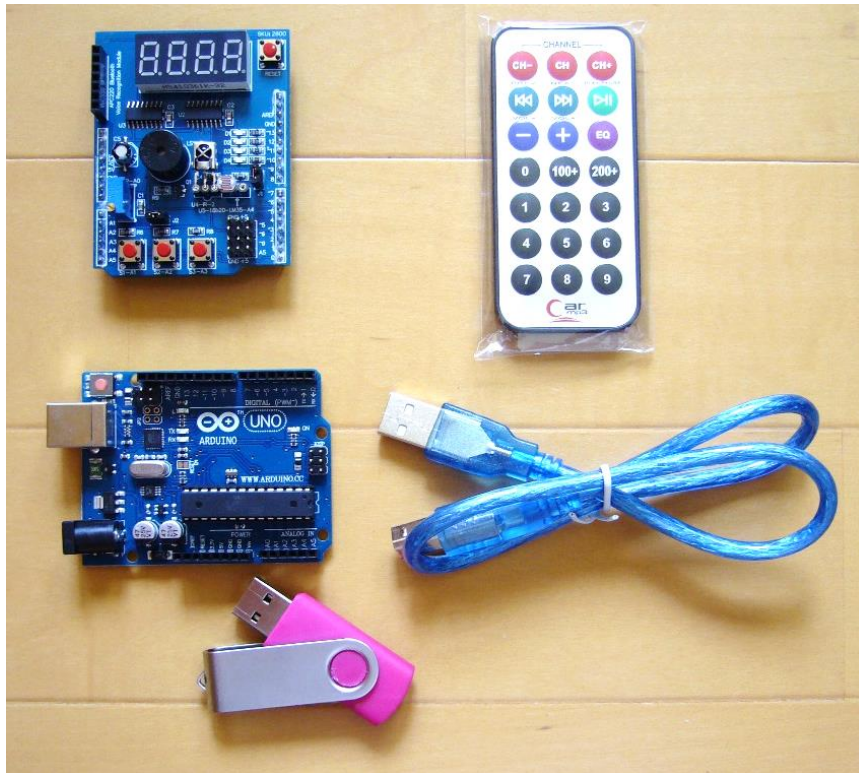
単独自習で本教材の初心者向けページの学習に要する時間は、[未経験初心者で10時間程度を想定](#)しています。さらに中級レベルまでの全内容の遂行に要する見込み時間は、経験上級者で1時間程度、[未経験初心者で100時間以上を要する想定](#)です。焦らずじっくり学習に取り組んでください。各ページに[初心者・初級者・中級者](#)向けの目安を表示してあります。

なお、本教材を利用するには、コンピュータ操作等に関して[以下のスキルがあらかじめ必要](#)となります。

Windows PCなどのコンピュータで以下の操作ができる

- 1) ワード、エクセルなどでの文書、表作成、ファイル操作など
- 2) アプリケーションソフトのインストールや設定
- 3) USBデバイスなど接続、設定
- 4) インターネットへのアクセス

セットの内容



* 製造時期等によって、各部品の色や形状が若干異なる場合がございます。ご了承ください。

自分で用意するもの

1. Windows PC
2. インターネット接続環境

内容物

1. 学習ボード, Arduinoボード
2. IR(赤外線)リモコン
3. USBケーブル
4. USBメモリー

* 学習ボードとArduinoボードは接続済み

ご使用上の注意

ボード裏面には電気配線があります。ショートしないよう、木や紙の上に置いてご利用ください。通電中にボードを金属など電気を通す物の上には絶対に置かないでください。また、金属などでボードに触れないでください。電気回路のショートは故障の原因です。

Lesson1 使用する道具について

- マイコン(マイクロコンピューター)

マイコンは、電気機器を制御するための電子部品、最近のほとんどの電気機器に組み込まれている。

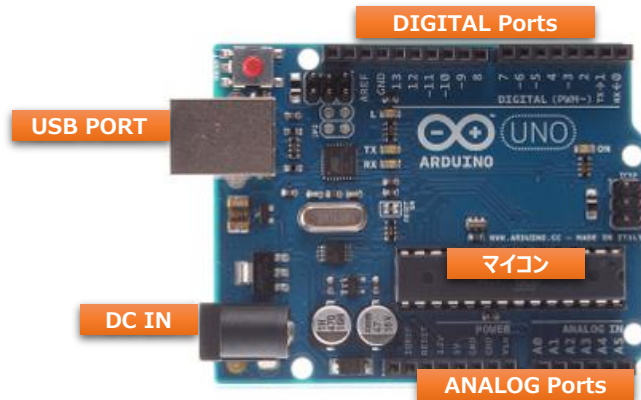
本テキストは、Ardunio(アルディーノ)またはその互換ボードというマイコンボードを使用。

マイコンボードは、マイコンチップを扱い易くするために最低限必要な整備を併せて基盤に載せたもの。

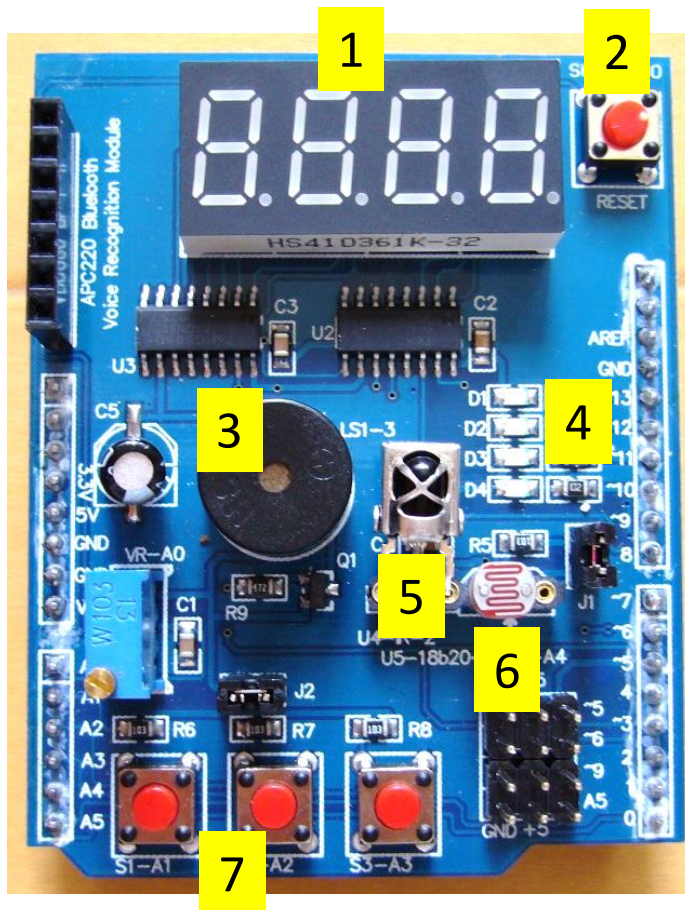
- Ardunio(アルディーノ)または互換機

- Arduinoは、本体の回路や開発環境(IDE)が無料公開されており、情報が豊富で取扱いが容易。

- マイコンを使いやすくまとめたArduinoまたはその互換機は、プログラミング教育に向いている。



Lesson1 使用する道具について



・学習ボード

1. 7セグメントLED(4桁)
2. リセットスイッチ
3. ブザー
4. LED(4個)
5. IR(赤外線)レシーバー
6. 光センサー
7. ボタンスイッチ(3個)

マイコン制御 ピン割当て

--

--

3

10,11,12,13

2

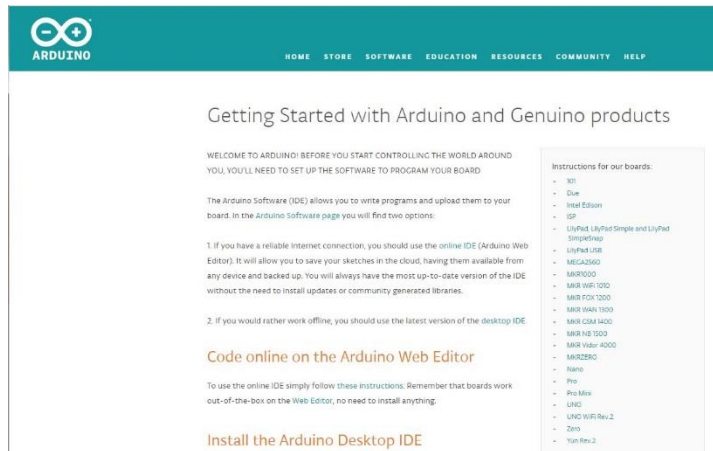
A4

A1,A2,A3

* プログラムに必要な各機能のピン割当てはボード上にも表示されています

Lesson1 ソフトウェアのインストール

・開発環境ソフトのPCへのインストール



<https://www.arduino.cc/en/Guide/Windows>から
Ardunioの開発環境(IDE)をPCにダウンロードし、イ
ンストール
(最新バージョン)

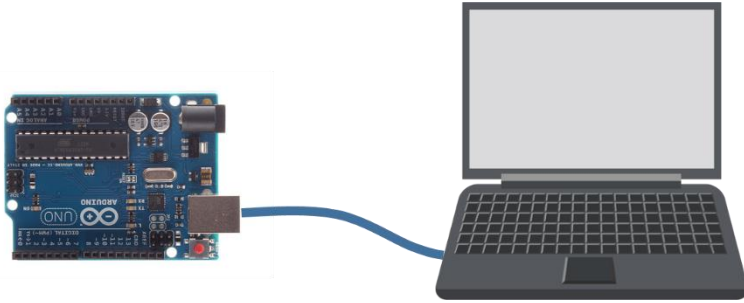
* または、付属USBメモリのIDE
(Arduino-1.8.8-windows)をインストール

インストール方法の詳細は
<https://www.arduino.cc/en/Guide/Windows>
にしたがって進めましょう



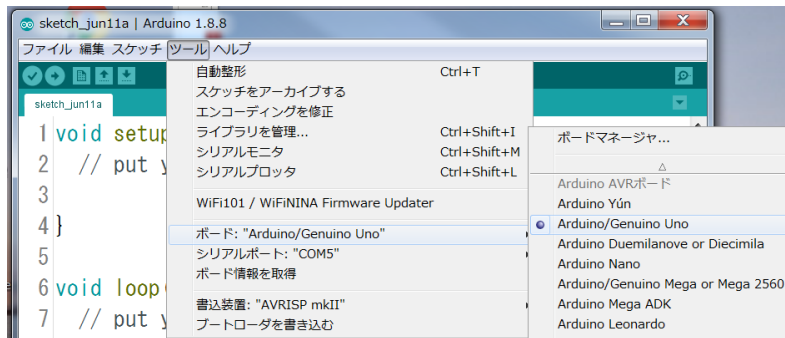
Windows OSの自動更新時などはソフトのインストールや
ソフトの動作が停止状態になることがあります。
PCのディスクへのアクセスが無くなってから試みてください。
また、うまく動作しないときはPCを再起動してください。

Lesson1 ソフトのインストール



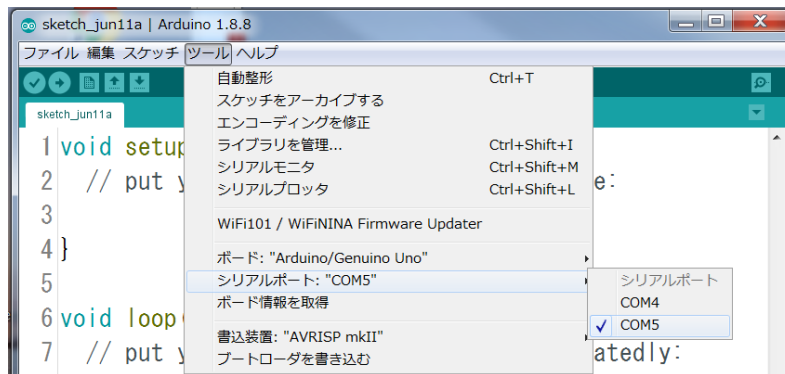
- Arduino ボードに USB ケーブルでつなぐ

PCとArduinoボード(学習ボードの下側)にUSBケーブルを差し込む



- ボード種類の設定

- メニュー> ツール> ボード: "Arduino/Genuino Uno" でボード種類を選択



- シリアルポート番号の設定

PCとボードで通信できるようにするシリアルポート番号の設定 (COM6など)

* 数字番号は使用するPCによって異なることがあります。

メニュー> ツール> シリアルポート> COM6で選択

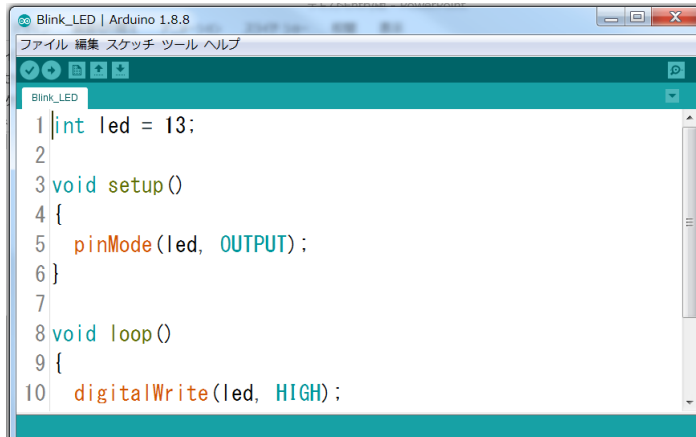
Lesson2 付属サンプルプログラム

注意！

付属USBの内容を、誤って消したり、書き換えてしまったりしないように、PCのデスクトップなどにコピーして使おう。

- Blink_LED 1つのLEDが点滅
- Blink_All_LEDs 4つのLEDが点滅
- 7segment_display 数字を表示
- Switches_LEDs ボタンスイッチ1, 2でLEDの点灯制御
- Switches_Buzzer ボタンスイッチ1, 2, 3でブザー制御
- Photosensor_Tone 明るさに応じて数値と音程が変化
- IR_remote_LED リモコンでLED制御
- Game_roulette ゲーム

Lesson2 LEDの点滅制御



```
Blink_LED | Arduino 1.8.8
ファイル 編集 スケッチ ツール ヘルプ

Blink_LED
1 int led = 13;
2
3 void setup()
4 {
5   pinMode(led, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(led, HIGH);
```

- サンプルプログラムBlink_LEDを起動

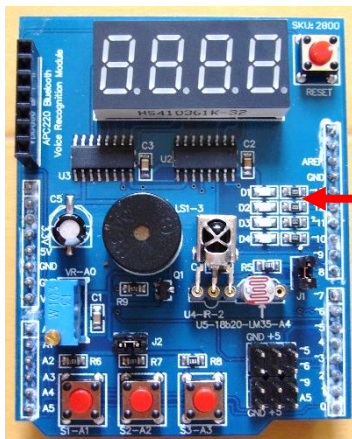
サンプルフォルダー内のBlink_LEDをダブルクリックすると、LEDアプリが起動しサンプルプログラムが画面に表示されます。

- プログラムをボードへ書き込む



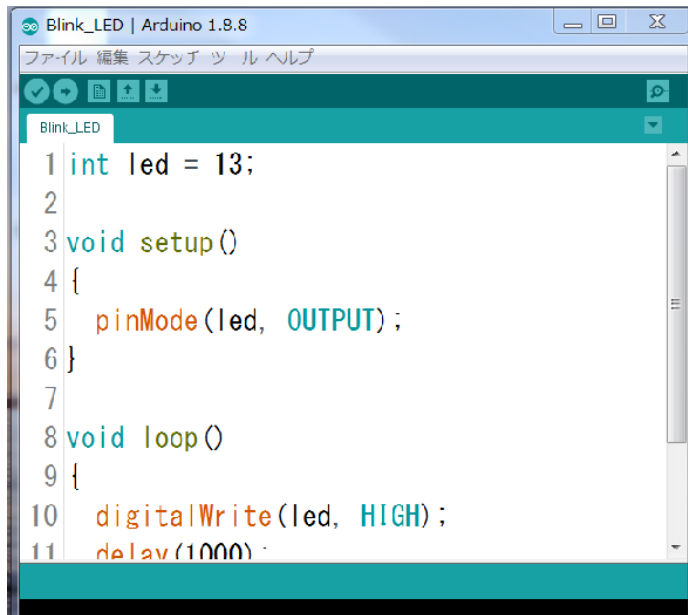
をクリックすると、サンプルプログラムをマイコンボードへ送る(書き込む)

ボードのLEDが点滅
1秒点灯、1秒消灯の繰り返し



* 制御されていない数字表示のLEDがランダムに点灯することがありますが、このプログラムとは関係がありません。

Lesson3 LED点滅制御プログラミング



```
Blink_LED | Arduino 1.8.8
ファイル 編集 スケッチ ツール ヘルプ
Blink_LED
1 int led = 13;
2
3 void setup()
4 {
5   pinMode(led, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(led, HIGH);
11  delay(1000);
```

いろんなコマンドについての詳細は、
メニュー>ヘルプ>レファレンス
でLanguage Referenceを参照しましょう。

• サンプルプログラムに手を加えてみよう

変更事項: 0.5秒点灯、0.5秒消灯の繰り返し

プログラムコマンド(命令)の説明

digitalWrite(出力先, 状態)

出力先を指定した状態(HIGH/LOW)にする

delay(時間ms)

指定しただけ時間待つ

プログラムの変更

```
void loop()
```

```
{
  digitalWrite(led, HIGH);
  delay(500);
  digitalWrite(led, LOW);
  delay(500);
}
```

この数値を
色々変えて
実験しよう!

* 変更したプログラム
は再度ボードに書き込
む必要があります

* プログラムを変更したら、別のファイル名で保存することをお勧めします。
そのまま、ボードへ書き込むと、元のサンプルプログラムファイルが書き換えられてしまいます。

Lesson4 サンプルプログラム体験

サンプルプログラムは、各フォルダーを開き、プログラムをクリックすれば画面が現れます。後は、先ほどのLEDの点滅の場合と同じように、ボードへ書き込むだけです。

実際の動作とプログラムの記述を比較検討し、各プログラムの構成、コマンドの意味や使い方を研究しよう。

ひとつとおり出来たら、サンプルプログラムの数値変更や組み合わせなどによって、オリジナルな動作にチャレンジしてみよう。

- Blink_LED 1つのLEDが点滅
- Blink_All_LEDs 4つのLEDが点滅
- 7segment_display 数字のを表示
- Switches_LEDs ボタンスイッチ1, 2でLEDの点灯制御
- Switches_Buzzer ボタンスイッチ1, 2, 3でブザー制御
- Photosensor_Tone 明るさに応じて数値と音程が変化(ボタンを押す)

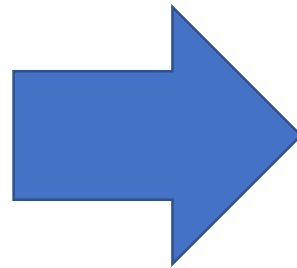
Lesson4 サンプルプログラムの変更例

Blink_All_LEDs

* 制御されていない数字表示のLEDがランダムに点灯することがありますが、このプログラムとは関係がありません。

digitalWrite(出力端子, 状態);
出力端子の状態を設定する命令

```
void loop()  
{  
  digitalWrite(led1, HIGH);  
  digitalWrite(led2, HIGH);  
  digitalWrite(led3, HIGH);  
  digitalWrite(led4, HIGH);  
  delay(1000);  
  digitalWrite(led1, LOW);  
  digitalWrite(led2, LOW);  
  digitalWrite(led3, LOW);  
  digitalWrite(led4, LOW);  
  delay(1000);  
}
```



```
void loop()  
{  
  digitalWrite(led1, HIGH);  
  digitalWrite(led2, LOW);  
  digitalWrite(led3, HIGH);  
  digitalWrite(led4, LOW);  
  delay(1000);  
  digitalWrite(led1, LOW);  
  digitalWrite(led2, HIGH);  
  digitalWrite(led3, LOW);  
  digitalWrite(led4, HIGH);  
  delay(1000);  
}
```

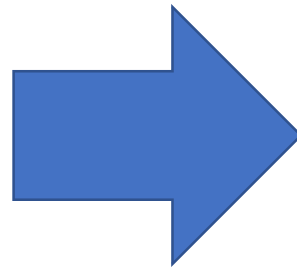
Lesson4 サンプルプログラムの変更例

7segment_display

WriteNumberToSegment(表示位置, 数字);
表示位置に数字を表示する命令

delay(時間);
設定時間だけ待つ命令(時間はミリ秒)

```
void loop()
{
  WriteNumberToSegment(0 , 5);
  delay(5);
  WriteNumberToSegment(1 , 6);
  delay(5);
  WriteNumberToSegment(2 , 7);
  delay(5);
  WriteNumberToSegment(3 , 8);
  delay(5);
}
```



```
void loop()
{
  WriteNumberToSegment(0 , 5);
  delay(100);
  WriteNumberToSegment(1 , 6);
  delay(100);
  WriteNumberToSegment(2 , 7);
  delay(100);
  WriteNumberToSegment(3 , 8);
  delay(100);
}
```

Lesson4 サンプルプログラムの変更例

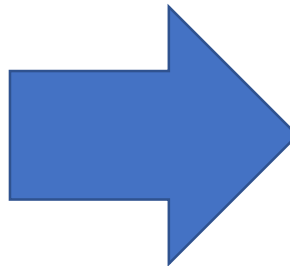
Switches_LEDs

#define 名前 入出力端子
入出力端子に名前をつける定義

```
#define BUTTON1 A1
#define BUTTON2 A2

void loop()
{
  if(!digitalRead(BUTTON1))
  {
    digitalWrite(LED[0], HIGH);
    digitalWrite(LED[1], HIGH);
    digitalWrite(LED[2], HIGH);
    digitalWrite(LED[3], HIGH);
  }

  if(!digitalRead(BUTTON2))
  {
    digitalWrite(LED[0], LOW);
    digitalWrite(LED[1], LOW);
    digitalWrite(LED[2], LOW);
    digitalWrite(LED[3], LOW);
  }
}
```



```
#define BUTTON1 A1
#define BUTTON2 A2
#define BUTTON3 A3

void loop()
{
  if(!digitalRead(BUTTON1))
  {
    digitalWrite(LED[0], HIGH);
    digitalWrite(LED[1], HIGH);
    digitalWrite(LED[2], HIGH);
    digitalWrite(LED[3], HIGH);
  }

  if(!digitalRead(BUTTON2))
  {
    digitalWrite(LED[0], LOW);
    digitalWrite(LED[1], LOW);
    digitalWrite(LED[2], LOW);
    digitalWrite(LED[3], LOW);
  }

  if(!digitalRead(BUTTON3))
  {
    digitalWrite(LED[0], HIGH);
    digitalWrite(LED[1], LOW);
    digitalWrite(LED[2], HIGH);
    digitalWrite(LED[3], LOW);
  }
}
```


Lesson4 サンプルプログラムの変更例

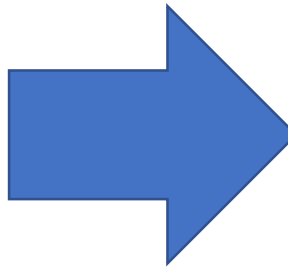
Switches_Buzzer

tone(出力先, 周波数, 時間);
出力先に指定周波数の信号を指定時間出す命令

```
void loop()
{
  if(!digitalRead(BUTTON1))
  {
    tone(Buzz,523,10);
  }

  if(!digitalRead(BUTTON2))
  {
    tone(Buzz,587,10);
  }

  if(!digitalRead(BUTTON3))
  {
    tone(Buzz,659,10);
  }
}
```



```
void loop()
{
  if(!digitalRead(BUTTON1))
  {
    tone(Buzz,523,500);
    delay(500);
    tone(Buzz,659,500);
    delay(500);
    tone(Buzz,784,500);
    delay(500);
  }

  if(!digitalRead(BUTTON2))
  {
    tone(Buzz,494,500);
    delay(500);
    tone(Buzz,587,500);
    delay(500);
    tone(Buzz,784,500);
    delay(500);
  }

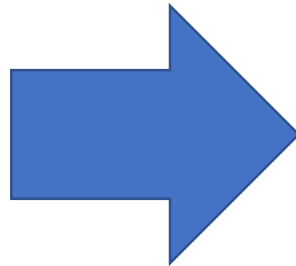
  if(!digitalRead(BUTTON3))
  {
    tone(Buzz,440,500);
    delay(500);
    tone(Buzz,523,500);
    delay(500);
    tone(Buzz,698,500);
    delay(500);
  }
}
```

Lesson4 サンプルプログラムの変更例

Switches_Buzzer

```
if ( 条件 ) 命令;  
または  
if ( 条件 )  
{  
    命令;  
}  
条件がマッチしたときに命令を実行する
```

```
if ( sound == HIGH )  
{  
    tone(BUZZ, Phot);  
}  
else  
{  
    noTone(BUZZ);  
}
```



```
if ( sound == HIGH )  
{  
    if ( Phot >= 500 ) tone(BUZZ, 523);  
    if ( Phot < 500 ) tone(BUZZ, 1046);  
}  
else  
{  
    noTone(BUZZ);  
}
```

初心者コース おわり

お疲れ様でした

初心者コースは以上でおわりです
自分のものになるまで、繰り返しトライしましょう

ここから先の内容は初級者、中級者向けです
自分自身で課題解決することが求められる内容です
答えは一つではありません。
それぞれ考えた結果は動作で確かめられます。

初心者コースで十分自信をつけてからチャレンジしてみましょう

Lesson5 課題プログラミング

• Blink_All_LEDs

4つのLEDが同時に点滅するこのプログラムの機能を改造してみよう！

要求される機能は、
4つのLEDが流れるように点滅する。

ヒント

各LEDの点灯、消灯を制御しているコマンドdigitalWriteと待ち時間制御コマンドのdelayの組み合わせ。

* 改造したプログラムはファイル名を新たにつけて別に保存しよう。

プログラムへの表現はオリジナリティあふれ、人それぞれです。見た目の機能は同じでも、プログラムの正解は一つではありません。自由にプログラミングしましょう。

Lesson5 課題プログラミング

- 7segment_display

あらかじめ決まった数字を表示するこのプログラムの機能を改造してみよう！

要求される機能は、
ランダムな数字を表示する。

ヒント
乱数を作るコマンド関数randomがあります。

* 改造したプログラムはファイル名を新たに付けて別に保存しよう。

Lesson5 課題プログラミング

- Switches_LEDs

スイッチボタン1、2で、4つのLED同時点滅を制御するこのプログラムの機能を改造てみよう！

要求される機能は、
前出の流れる点滅に改造したBlink_All_LEDs(改造版)を参考に、LED点滅の流れる方向をスイッチ1では上から下、スイッチ2では下から上へとなるようにする。

* 改造したプログラムはファイル名を新たにつけて別に保存しよう。

Lesson5 課題プログラミング

• Photosensor_Tone

明るさに応じた数値を表示し、その周波数の音を鳴らすこのプログラムの機能を改造してみよう！

要求される機能は、
明るさの程度に応じて、C調の音階(ド、レ、ミ.....)をBuzzerで鳴らす楽器にする。

ヒント

明るさの数値範囲は0～1023。まずは、1オクターブの8音程で分割すれば明るさ0～127をド、128～255をレ.....

音階を出すにはtone関数が便利。

次は、3オクターブ程度に音域を広げ、演奏してみよう。

* 改造したプログラムはファイル名を新たに付けて別に保存しよう。

Lesson5 課題プログラミング

- 7segment_display
- Switches_Buzzer

この2つのプログラムの機能を1つに合体させてみよう！

要求される機能は、
スイッチボタン1, 2, 3を押した時に、
それぞれミ、ソ、ドの音階（周波数）でブザーが鳴る。
同時にその周波数がLEDに数値表示される。

一方のプログラムに、もう一方のプログラムの必要な部分をコピーペーストしながら、新しいプログラムを作っていこう。コピーなら得意だよね。
新しいファイル名で、保存することを忘れないように。

ヒントは次のページ

Lesson5 課題プログラミング

ヒント

- 1) まずは、Switches_Buzzerに書かれている記述をPot_7segの適切な対応する位置にコピーで追加。
- 2) 音階と周波数の関係は.....
- 3) LEDの各桁に対応する数字を当てはめる。変数が整数ならば、456を100で割ると4。456を10で割った後にさらに10で割った余りは5.....

プログラム言語の細かな意味は、余裕が出てから調べよう。
まずは、要求を噛み砕き、プログラムとしてどのように表現、構成するか、といったプログラムとして具現化する創造や思考が大切。

ここからは中級者コース

ここからは中級レベルです

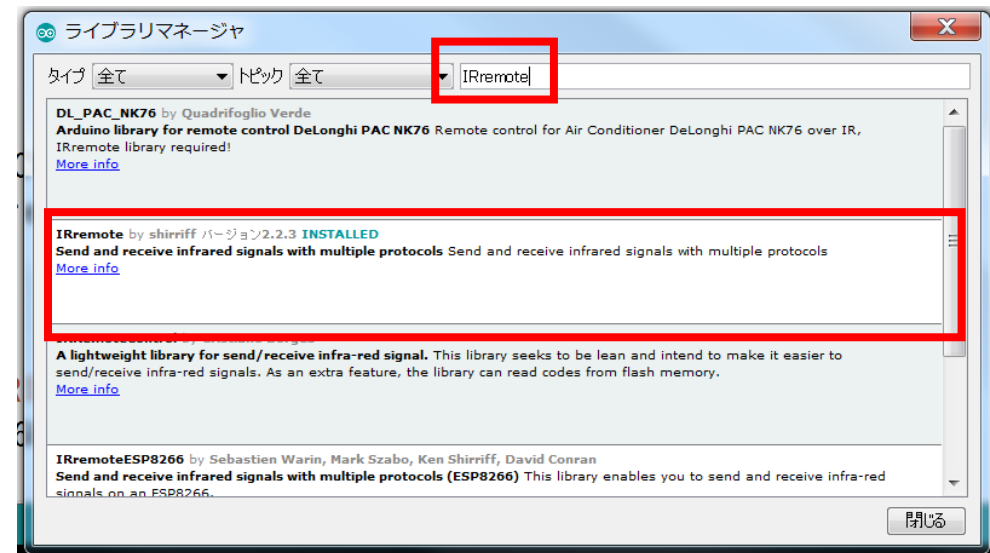
ある程度経験を積んでからチャレンジしてみましょう

使い方や機能について、プログラムを調べ考えてみよう

- IR_remote_LED

IR(赤外線)リモコン(透明フィルムを外し使用)でLEDの点灯制御

メニュー>ツール>ライブラリを管理
でライブラリマネージャ画面を開き、
検索窓にIRremote
と入力し、表示されたライブラリリスト
からIRremoteライブラリを選択してイ
ンストールしておく必要があります。



リモコンに対する反応が悪いときは、学習ボードとリモコンの距離や角度をいろいろ変えて試みてください(例えば2m以上)。

- Game_roulette

ルーレットゲーム

Lesson6 ちょっと複雑プログラミング

• IR_remote_LED

リモコンボタンに応じて4つのLEDが点灯、消灯するこのプログラムの機能を改造してみよう！

要求される機能は、
押されたリモコンの数字ボタンに対応した数字を、4桁の数字としてLEDに数値表示する。
出来たら次に、数字ボタン以外のボタンをLED表示クリア機能とする。
最後は、電卓にしてみよう(ボタン割り当てはお好みで)。

ヒント

メニュー>ツール>シリアルモニタで新たな画面を開くと、リモコンから送出される赤外線信号のコードが16進で表示されます。この数字をプログラム中のボタン判定に使います。

* 改造したプログラムはファイル名を新たに付けて別に保存しよう。

Lesson6 ちょっと複雑プログラミング

• Game_roulette

ルーレットゲームの不完全さを改良して完成度を向上してみよう！

不具合点

スイッチ(1, 2)で2桁の数字をセットし、スイッチ3でルーレットが回ります。
数字の一致のしかたに応じた得点が加算され、LEDに表示されます。

コンピューターが発生する右2桁の数字には、規則性が残っています。
(リセット毎に同じ数字が再現される)

要求される機能は、
リセット毎にルーレット数字の規則性が無いものにする。
ある得点以上で、音楽を流す。
リモコンで操作

ヒント

乱数発生計算に用いる種に、リセット毎に不確定なものを利用する。
身の回りの明るさは安定？

Lesson7 オリジナルプログラム

オリジナルなプログラムを作ろう！

オリジナルな機能の仕様を決定(創造)し、プログラムで実現(プログラミング的思考)してみよう。

はじめは、サンプルプログラムの組み合わせや、改造から始めてみる。

メニュー＞ヘルプ＞リファレンス

Language Referenceを眺めてみると、色々なことができそうです
サンプルプログラムで使っていなかったコマンドや制御をどんどん使ってみよう。

自分で作ったプログラムを勉強会、ブログや
SNSで公表、発信して仲間を作ろう！

おわり

プログラムへの表現はオリジナリティあふれ、人それぞれです。見た目の機能は同じでも、プログラムの正解は一つではありません。自由にプログラミングしましょう。