


通し番号 (e.g.
Gen.001)



**電装班新歓
-LEDをピカピカ光らせよ
う！-**

作成：服部開都 (24)
担当：服部開都 (24)

本日の流れ

CREATE



- ・ 新歓基板の紹介
- ・ パソコンで環境構築しよう！
- ・ LEDをピカピカ光らせよう！
- ・ 7セグを光らせて数字を表示させてみよう！
- ・ プルアップでスイッチを作ってみよう！
- ・ カウントダウンゲームを作ってみよう

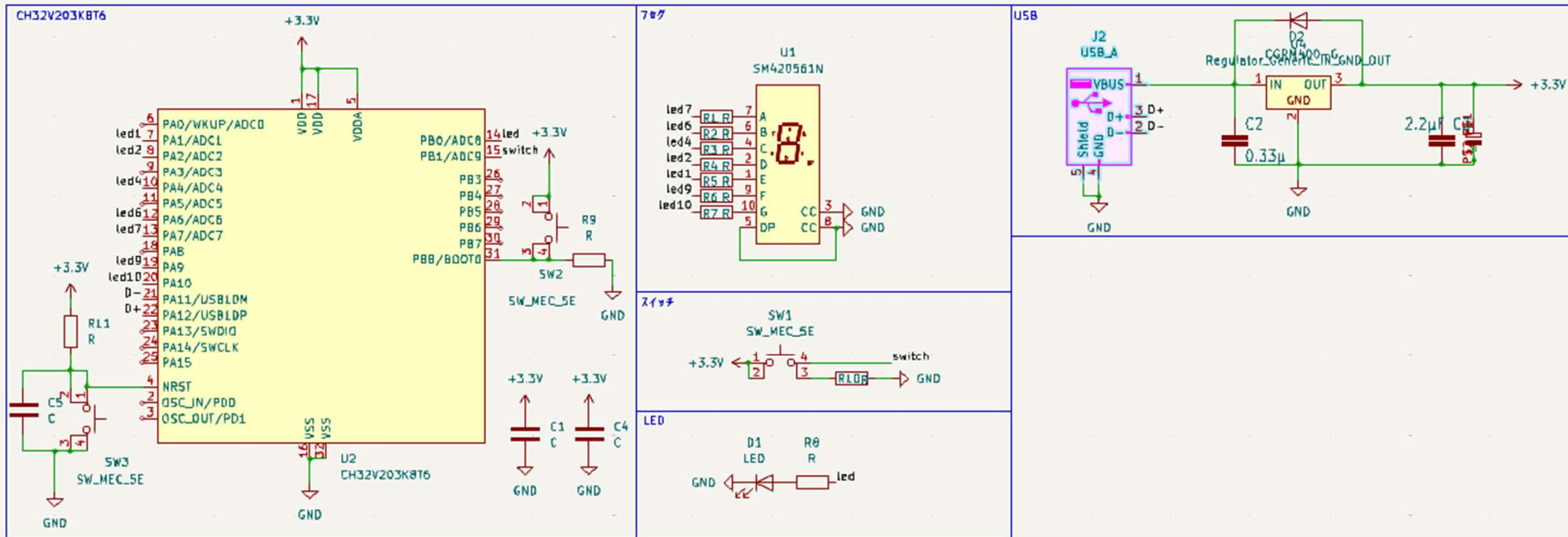


1. 新歓基板の紹介

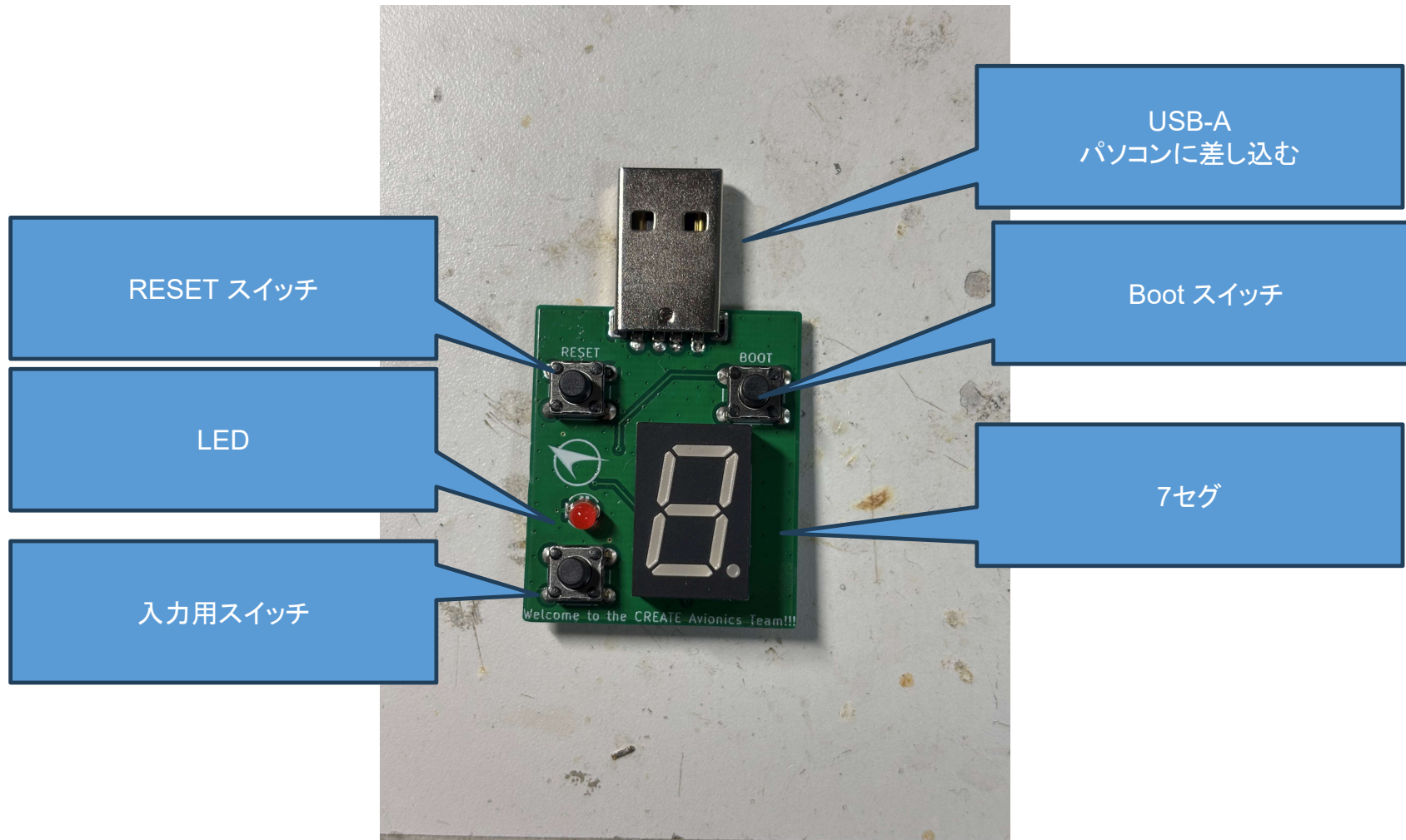


1. 新歓基板の説明

CREATE



1. 新歓基板の説明





1. 新歓基板の概要

- ・ スイッチの色は違うことがあります。



2. パソコンで環境構築しよう！



2. パソコンで環境構築しよう！



今日使うコード、このスライドのダウンロード

- 下記のgithubにアクセスしてフォルダをダウンロード(緑色のCode→Download Zip)
- <https://github.com/308029/shinkan>



2. パソコンで環境構築しよう！



Arduinoのインストール

- Arduinoのインストール
- 下のサイトに行き、arduino1.8.19をインストール
<https://www.arduino.cc/en/software>
- 寄付するか聞かれるので「JUST DOWNLOAD」を押す
- 2.3 zipファイルがダウンロードされるので展開して.exeファイルを開く

Legacy IDE (1.8.X)

A screenshot of the Arduino IDE 1.8.19 download page. The page has a light blue header with the Arduino logo and the title 'Arduino IDE 1.8.19'. Below the header, there is a paragraph of text explaining that the software is open-source and easy to use. To the right, there is a dark teal box titled 'DOWNLOAD OPTIONS' which lists download links for Windows, Linux, and Mac OS X. Below this, there is a section titled 'Previous Releases' which lists links for previous versions of the IDE. The page is clean and professional, with a clear focus on providing download options for different operating systems.

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Arduino IDE 1.x documentation](#) for installation instructions.

SOURCE CODE

Active development of the Arduino software is hosted by GitHub. See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this gpg key](#).

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file [Get](#)

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes
Checksums (sha256)

Previous Releases

Download the previous version of the current release, the classic 1.0.x, or old beta releases.

DOWNLOAD OPTIONS

[Previous Release 1.8.18](#)
[Arduino 1.0.x](#)
[Arduino 1.5.x beta](#)

2. パソコンで環境構築しよう！



Arduinoの関数

- setup(): プログラム実行時に最初に 1 回だけ実装される
- loop(): プログラムを実行すると括弧の内側に書いたコードが上から順に実行され、一番下まで行くとまた上にもどってずっと繰り返される

A screenshot of the Arduino IDE interface. The title bar reads "sketch_apr07a | Arduino 1.8.10". The menu bar includes "ファイル", "編集", "スケッチ", "ツール", and "ヘルプ". Below the menu bar is a toolbar with icons for saving, running, and other functions. The main editor area shows the following code:

```
sketch_apr07a

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

2. パソコンで環境構築しよう！

CREATE



今回使うマイコンが動くようにする

- ・ 下のサイトを参考にarduinoの設定とWCH-Linkのインストールをする
- ・ <https://qiita.com/usashirou/items/a52cc331817c294a98dc>



補. 今回の開発の取り組み方

- ・ ミッションごとにゴールが与えられているのでそれを実装してみよう！
- ・ その際必要になる情報はヒントとして与えられている
- ・ ヒントを見てもわからなかったら電装班員に聞くor Githubからダウンロードしたコードを試してみる。
- ・ 電子工作をやるのが初めての人は徐々につかんでいこう！
- ・ ミッションが終わった人は最後に追加のミッション書いてあるのでやってみてください。
- ・ わからないことがあったら何でも電装班の人にきいてください！



3. LEDをピカピカ 光らせよう！





3. LEDをピカピカ光らせよう！

ゴール

- ・ 基板についているLEDを光らせる
- ・ LEDを1秒ごとに点滅を繰り返すようにする

ヒント

- ・ 今回のLEDのピン番号はPB0

3. LEDをピカピカ光らせよう！



使う関数

- `#define` 好きな名前 文字列：下のコードで設定した名前を打つと指定した文字列として認識する（数学の定数的な考え方）
→これでLEDのピン番号を指定しよう
- `pinMode(ピン番号、設定)`:指定したピン番号のピンをOUTPUTやINPUTに設定する
OUTPUT:電圧を出す INPUT:電圧を読み取る
- `digitalWrite(ピン番号、HIGH or LOW)`:指定したピン番号のピンからHIGH(3.3V)orLOW(0V)の電圧を出す
- `delay(ミリ秒)`:指定したミリ秒だけ待機する

これらをどこに書けばいいのか考えてみよう！

3. LEDをピカピカ光らせよう！



- ・ 正解(shinkan/2.Lチカ/led_tikatika.ino)

A screenshot of the Arduino IDE interface. The title bar reads "led_tikatika | Arduino 1.8.19". The code editor shows the following code:

```
#define led_pin PB0 //今回の基板でLEDはマイコンの"PB0"ピンに接続されている
void setup() {
  // put your setup code here, to run once:
  pinMode(led_pin, OUTPUT); //ピンを初期設定する
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(led_pin, HIGH); //ピンに3.3Vの電圧を送り、LEDを光らせる
  delay(500); //500ms待機
  digitalWrite(led_pin, LOW); //ピンに0Vの電圧を送り、LEDを光らせない
  delay(500);
}
```




書き込んでみよう！



3. LEDをピカピカ光らせよう！

CREATE



- Arduinoの上のツールバーの「スケッチ→コンパイルしたバイナリを出力」を選択
- スケッチが入っているフォルダにアクセスすると「ファイル名.bin」というファイルのができている
- WCHISPStudioを開く
- Chip optionを今回のマイコンであるCH32V203に,Dnld PortをUSBに設定
- Download Fileで作った.binファイルを選択

3. LEDをピカピカ光らせよう！



- ・ 6. Boot スイッチとRESETスイッチを同時に押してからRESETスイッチだけを離す。
- ・ 7. すると「CH32V203 detected」と出る
- ・ 8. 下の「Download」ボタンを押して書き込む
- ・ 9. LEDがチカチカ光れば成功！

3. LEDをピカピカ光らせよう！



The screenshot shows the WCHISPStudio_V370 interface. The 'Chip Option' section is highlighted with a red box, showing 'Chip Series' set to 'CH32V20x' and 'Chip Model' set to 'CH32V203K8T6'. The 'Download File' section is also highlighted with a red box, showing a list of files with 'Object File1' selected. The 'Download Config' section is visible, showing various settings like 'DL_Baud' and 'Disable stop-mode RST'. The 'Download' button is highlighted with a red box. The 'Device' status at the bottom left shows 'Device: Not Connected'. Blue callout boxes provide instructions: '今回使うのはCH32V203K8T6なのでそれに設定、ポートはUSB' (Set to CH32V203K8T6, port is USB), 'ここで作った.binファイルを指定する' (Specify the .bin file created here), 'Downloadで書き込み' (Write with Download), and 'Ch32V203 detected と表示されればOK' (OK if Ch32V203 detected is displayed).

Chip Option

Chip Series: CH32V20x Chip Model: CH32V203K8T6

Dnld Port: USB

Download File(Multiple .hex files can be selected for merging)

Name	FilePath
Object File1	arduino\sketch_mar25a\sketch_mar25a.ino.CH32V203G8.bin
Object File2	
Object File3	
Object File4	
Object File5	

Download Config

Chip Config

DL_Baud: 115200

Disable stop-mode RST: ☒

Disable standby-mode RST: ☒

Soft-Ctrl IWDG: ☒

Enable RRP: ☒

Clear CodeFlash: ☐

Perform a soft reset After Download: ☒

User Data

User data DATA0: 0xFF

User data DATA1: 0xFF

Progress:

Download Verify Deprotect

Device: Not Connected

MCU series selection

MCUs: CH32Vxxx

MCUs: CH32Fxxx

MCUs: CH57x-CH59x

MCUs: CH54x/CH55x

MCUs: CH56x/CH64x

【More information】

Chip information:

Categories:

IDE MRS download:

RISC-V core self-developed chip:

Version expert:

書き込む方法は以後も同じなので覚えておいてください

3. LEDをピカピカ光らせよう！

CREATE



早く終わった人用

- ・ LEDをもっと早くチカチカさせてみよう
- ・ LEDでモールス信号を表示させてみよう



4. 7セグを光らせて数字を表示させてみよう！





4. 数字を表示させてみよう！

ゴール

- ・ 基板についている7セグで数字を表示する
- ・ 数字が1秒ごとに0から9まで増えていき、9になったら次は0になってループしていく

ヒント

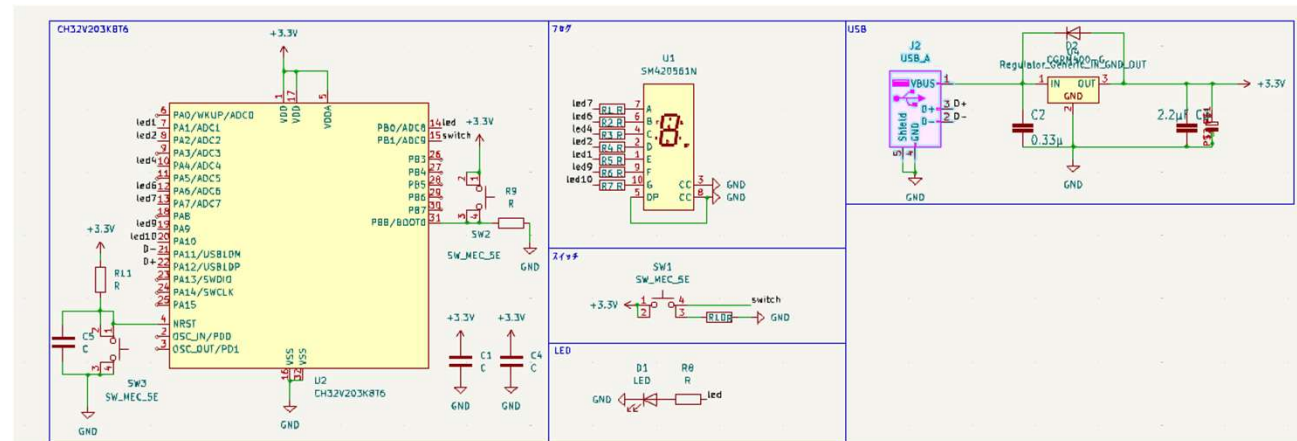
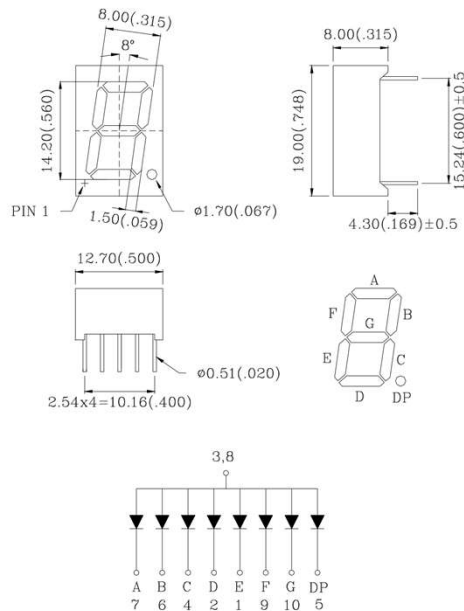
- ・ 7セグはLEDが7個（本来は右下のドットを入れて8個だが今回は配線していない）についているだけなので、2.のコードを応用すればできる
- ・ 各LEDのピン番号は次のスライドの表の通り
- ・ 今回はアノードコモンなのでピンにLOWを流すと光る

4. 数字を表示させてみよう！



- ・ 7セグのデータシート
- ・ <https://akizukidenshi.com/goodsaffix/a-551srd.pdf>
- ・ データシートと配線図から完成させられたあなたは電装班へ

PACKAGE DIMENSIONS



4. 数字を表示させてみよう！

CREATE



- ・ わかりやすく表にするとこういうこと
例えばAを光らせたければPA7をLOWにすればよい

光らせる場所	#defineするピン番号
A	PA7
B	PA5
C	PA4
D	PA2
E	PA1
F	PA9
G	PA10



4. 数字を表示させてみよう！

- ・ わからなくなったor打ち込むのが面倒だったらgithubからダウンロードしたファイルの「shinkan/src/4.7セグ/led_countup/led_countup.ino」を開いてコードをコピペしよう
- ・ 数字が表示でき、増えていったら完成！



4. 数字を表示させてみよう！

早く終わった人用

- ・ 数字が9から0に減っていくようにコードを変えてみよう
- ・ 今日の日付の数字を順に表示させてみよう
- ・ 各数字ごとにHIGH,LOWを指定するのは長いので省略してみよう（例2進法を使う）



4. プルアップでスイッチを作 ってみよう！





4. スイッチを作ってみよう！

プルアップとは？ Chat-GPTに聞いてみた

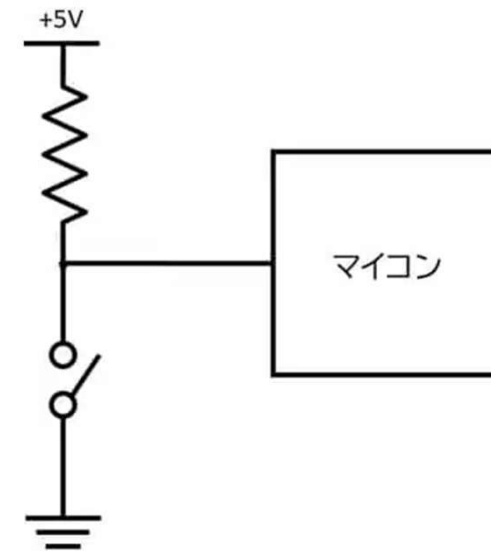
- ・ 電子工作におけるプルアップとは、デジタル回路の入力端子が不安定な状態（浮いた状態）にならないように、入力端子を電源（通常は正の電圧）に接続するための抵抗（プルアップ抵抗）を使うことです。この抵抗により、入力端子が確実に「HIGH」レベルに保たれるようになります。
- ・ スイッチなどが接続されている場合、スイッチが開いているときでも電圧が安定し、誤動作を防ぐことができます。逆に、GNDに接続するプルダウン抵抗もあり、これは入力を「LOW」レベルに保つ役割を果たします。



4. スイッチを作ってみよう！

つまり、、、

- ・ 右の回路で
スイッチが開いているとき
マイコンは5Vにつながっている
→HIGHが入力される
スイッチが閉じているとき
マイコンは0Vにつながっている
→LOWが入力される
なにもしないときにHIGHが入力されるから「プルアップ」



参考：

<https://voltechno.com/blog/pullup-pulldown/>



4. スイッチを作ってみよう！

ゴール

- ・ 基板についているスイッチを押すとLEDが光り、離すと消える

ヒント

- ・ digitalRead(ピン番号)関数を使うと、HIGHの電圧(3.3V)がピンにくると1,LOWの電圧(0V)がくると0を返す
- ・ 今回の配線ではスイッチとつながっているピンは「PB1」
- ・ 配線を見るとスイッチはプルアップで接続されているのでスイッチを押すとLOWがピンに入る

4. スイッチを作ってみよう！

CREATE



- 「shinkan/src/5.switch/switch/switch.ino」 をみてみよう



5. カウントダウンゲームを作ってみよう



5. カウントダウンゲームを作ってみよう

CREATE



ゴール

- ・ スイッチを押すと5から0へカウントダウンスタート
- ・ 7セグの数字が0になった時にスイッチをおして0.1秒以内に押せたらLEDが光る

ヒント

- ・ 0.1秒以内にスイッチが押されたかは`millis()`関数を使おう
`millis()`:実行してからの今の時間のミリ秒が分かる

5. カウントダウンゲームを作ってみよう

CREATE



- 「shinkan/src/5.ラストゲーム/last_game/last_game.ino」 を
見てみよう



CREATE電装班紹介



CREATE・電装班紹介



- CREATEとは？
- Challengers of Rocket Engineering and Avionics at TokyoTechを略してCREATEという団体名になっています。
-
- CREATEはハイブリッドロケットの開発・製作・打上げを行う，東京科学大学の技術系公認サークルです。
-
- 部員は自分達で設定した目標に向けて日々技術開発・製作に勤しんでいます。
- 到達高度の高高度化といった長期に渡る開発に取り組む傍ら，ユニークなミッションを設定した機体を作ることもあります。
-
- CREATEにはロケット・宇宙好きはもちろん，航空機・エンジンに興味がある人や単にモノづくりがしたいという人も在籍しています。
- 工学院のみならず理学院から生命理工学院まで幅広い専攻の学生が集まっていることも特徴です。

CREATE・電装班紹介

CREATE



- ・ CREATEのいいところ
- ・ 先輩方が優しく教えてくれる
- ・ 部室にいつも誰かいる
- ・ みんなで一つのものを作る(共同開発を体験できる)
- ・ 構造班、電装班、推進班、シミュレーション班に分かれているので同じロケットを作るにしても自分がやりたい分野を担当できる

- ・ 電装班の仕事
- ・ 飛行データを記録する
 - ・ 飛んでいる時の気圧や加速度を記録する
- ・ パラシュートを開かせる
 - ・ 一定の条件を満たしたときにサーボモータを回してパラシュートを開く
- ・ 地上と通信する
- ・ 離床検知したり開傘した時にそれを地上にいる班員のパソコンに送る
- ・ ミッションを成功させる
 - ・ 前回の新入生期待ではピトー管と呼ばれるもので風速を測定する予定だった



ぜひCREATEへ！

編集履歴

CREATE



- ・ 編集者氏名 (YYYY/MM/DD)