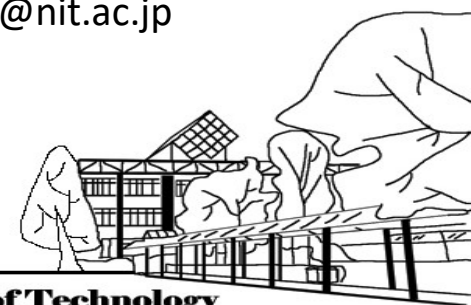


バーサライタを作ろう (ESP32バージョン)

本書はバーサライタを作るための手順を順に説明したものである。

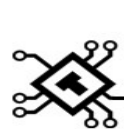
文責 滝田謙介 E-mail: takita@nit.ac.jp



目次

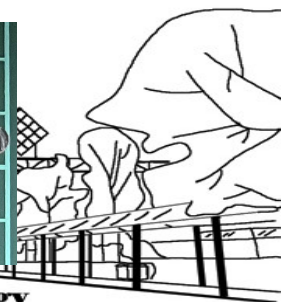
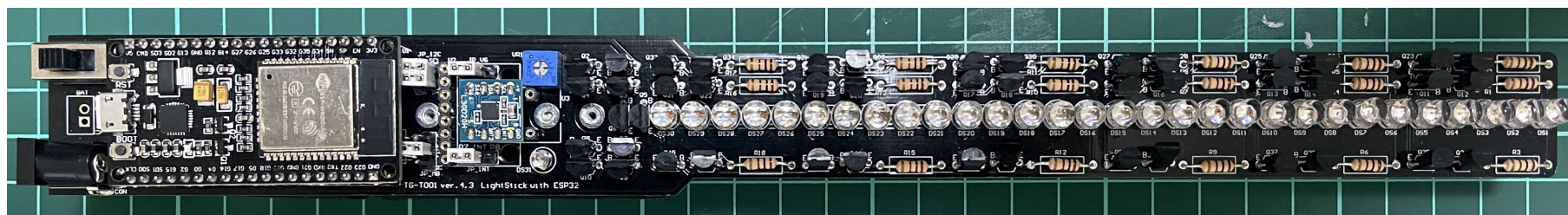
1. **概要**
2. 組み立て
3. Lチカに挑戦
4. タイマをつかう
5. センサを読み込む
6. センサとタイマを使った角度計算
7. LEDのスタティック点灯・ダイナミック点灯
8. 文字データの表示
9. 好きな文字を表示させてみよう





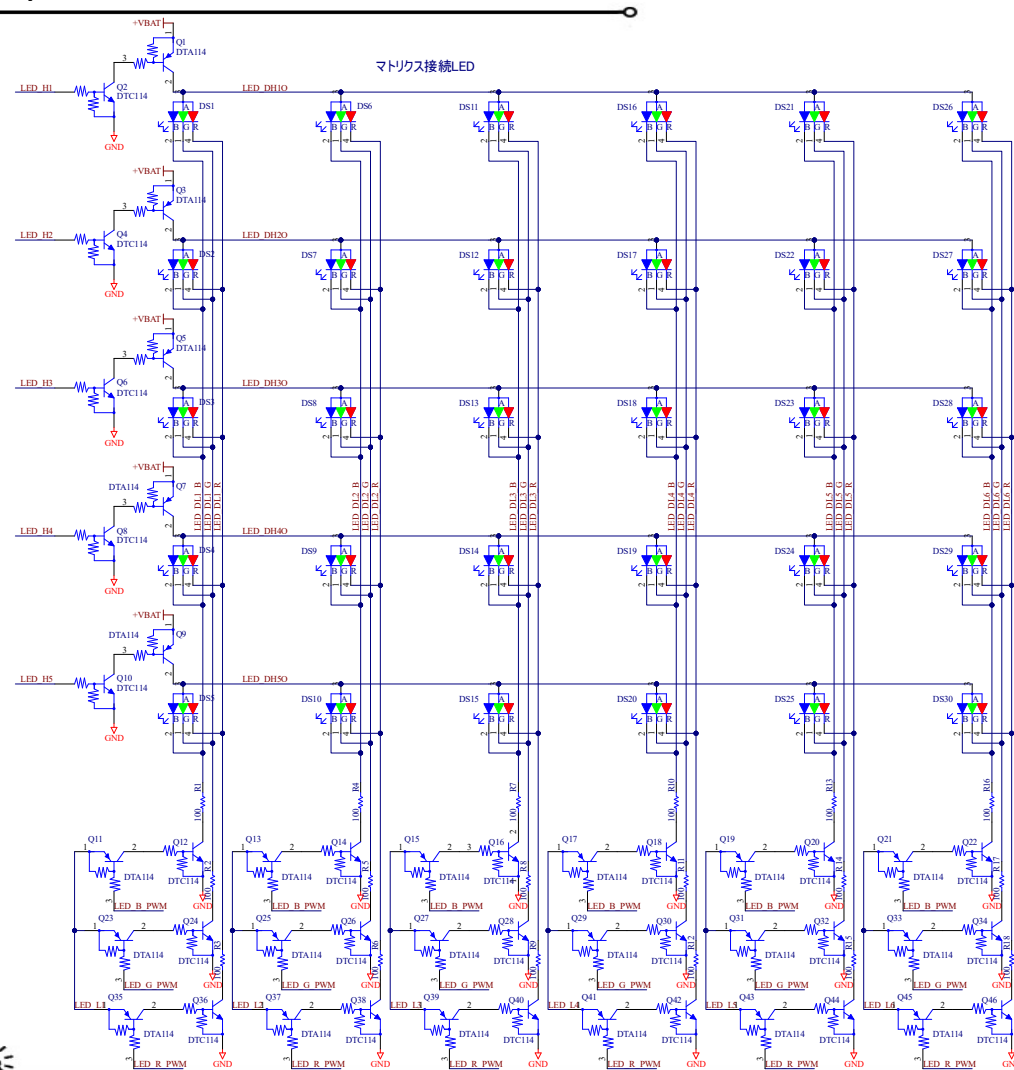
バーサイタ・概要

- バーサイタは、目の残像を利用して、空間に文字や絵を描くためのものである。
- ロボットの制御に必要な時間の管理を学ぶための教材でもある。
- 作成するバーサイタは、LEDを30個を搭載して高さ(または幅)30ドットの絵・文字を描くことが出来る。
- 姿勢センサ(ジャイロセンサ)と32ビットのIoTマイコンと言われるESP32を搭載している。
- ジャイロセンサから出力される角速度を取得し、時間係数をかけて足し算をすることで、角度のデータを作る。
- 32ビットマイコンでは、角度のデータから表示するデータを選択し、LEDを点灯させる。
- LEDの駆動方法には、スタティック点灯(静的点灯)ではなく、ダイナミック点灯(動的点灯)を採用し、省電力化を図っている。
- USBシリアル変換回路を搭載しているのでUSBがあれば、プログラムが可能。



回路図(秋)(単色版)

- マトリクス接続されたLEDを駆動するためのトランジスタアレイをCPUで制御する回路構成になっている。
- ソース側(電流供給側)を5組、ソース側(電流吸入側)を6組でアレイを構成している。
- ソース側は、RGBの三系統で構成している。
- NPN(プラスでON)+PNP(マイナスでON)の二つのトランジスタでスイッチ回路を構成している。
- LEDはダイナミック点灯とし、回路の簡素化・消費電力の低減を図っている。



- TAKITA[®] Labo.**

部品表

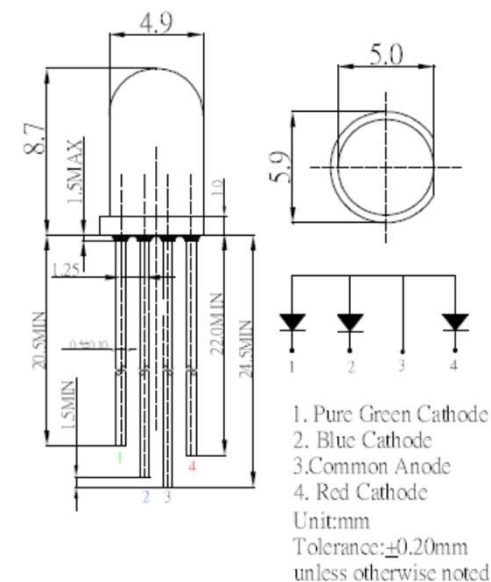
部品番号	製品型番	説明	量	値	秋月通販コード
BAT	BH-321-1A-150 2個	電池接続用パッド	1		P-17271
CON	MJ-179PH	DCジャック	1		C-06568
D1	SB240LES		12A Vf:0.41V		I-16419
DS1, DS2, DS3, DS4, DS5, DS6, DS7, DS8, DS9, DS10, DS11, DS12, DS13, DS14, DS15, DS16, DS17, DS18, DS19, DS20, DS21, DS22, DS23, DS24, DS25, DS26, DS27, DS28, DS29, DS30	OSTAMA5B31A	RGBフルカラーLED 大きさ 5mm アノードコモン	30	フルカラ	I-12168
DS31	指定なし	Φ3mmのLED (色は好みで)	1		例えばI-09851
JP_I2C	2x2 ピンヘッダ	ジャンパーピン	1		C-00080 または C-00167
JP_INT, JP_MB, JP_VG	1x3 ピンヘッダ	ジャンパーピン	3		C-00167
Q1, Q3, Q5, Q7, Q9, Q11, Q13, Q15, Q17, Q19, Q21, Q23, Q25, Q27, Q29, Q31, Q33, Q35, Q37, Q39, Q41, Q43, Q45	DTA114EL	抵抗入り PNPトランジスタ	23		I-12464
Q2, Q4, Q6, Q8, Q10, Q12, Q14, Q16, Q18, Q20, Q22, Q24, Q26, Q28, Q30, Q32, Q34, Q36, Q38, Q40, Q42, Q44, Q46	DTC114EL	抵抗入り NPNTトランジスタ	23		I-12467
R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17, R18, R22	1/2W カーボン抵抗 または 1/4W カーボン抵抗	抵抗	18	100Ω	R-07803 または R-25101
R19, R20, R21, R24, R25	1/4W カーボン抵抗	抵抗		64.7kΩ	R-25472
R23	1/4W カーボン抵抗	抵抗		11k	R-25102
S1	SS22F06G5-G	スイッチ	1		P-03252
U1			1		M-15673
U2	GY-521 または GY-BNO055	姿勢センサ	1		
U3	AE-BNO055-BO または AE-L3GD20	姿勢センサ	1		K-16996 または K-15096
VR1	3362P-1-503LF	半固定抵抗	1		P-14905 など

LEDの電流制限抵抗(R1～R18)

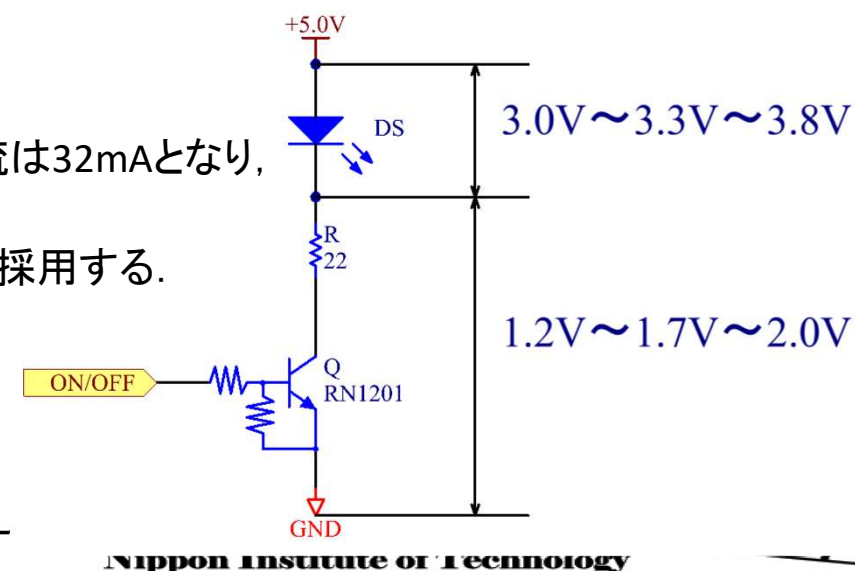
- LEDは様々な特性を持っているが、回路を設計する上で重要なパラメータが、順方向電圧と順方向電流である。
- 特に順方向電流は最大値を知ることが壊さないで使用するために重要である。
- 今回使用するフルカラーLED(OSTAMA5B31A)の最大順方向電流および順方向電圧は、30mA(赤)・50mA(青・緑)と1.8～2.6V(赤), 2.8～3.6V(青・緑)である。
- 電源電圧を5Vとすると、抵抗とトランジスタに印加される電圧が2.4V～3.2V(赤) / 1.4V～2.2V(青・緑)となる。トランジスタのVce(コレクタ・エミッタ間電圧)は、0.1Vぐらいなので、ここでは無視する。

赤色LEDについて考える。

- 2.4Vの時に、30mAとすると、 $2.4V/30mA \div 80\Omega$, $2.4V \times 30mA = 72mW$
- 3.2Vの時に、30mAとすると、 $3.2V/30mA \div 107\Omega$, $3.2V \times 30mA = 96mW$
- 仮に100Ωを選ぶとすると、LEDの順方向電圧が低い場合、LEDに流れる電流は32mAとなり、定格の最大絶対30mAを超えてしまう。
- しかし、トランジスタなどでの電圧降下もあるので、抵抗値としては、100Ωを採用する。
- 青・緑の抵抗についても同様に計算して決める。
- 最悪値を考えるならば、150Ωの抵抗にしておく。



データシートより



動作確認

- Arduino IDEをインストールして、以下のサイトを参考にESP32をボードマネージに登録する.

<https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>

※ ファイル自体は、下記のサイトを参照.

<https://github.com/espressif/arduino-esp32>

- パソコンなどのUSB端子にMicroUSB Type Bのコネクタを持ったケーブルのTypeA側を差し、MicroUSBをESP32 DevkitCのUSB端子に接続する.
- 必要ならば、DevkitC搭載のICのドライバをSilicon Labs社のWEBからダウンロードしてインストールする. (<https://jp.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>)
- テスト用プログラムをArduino IDEからESP32に書き込みLEDの駆動が正常に行なわれることを確認する.

※ テスト用プログラムは、下記のURLを参照する.

https://github.com/blues27/ESP32_POV_LightStickV4/tree/main/VERSAWRITER_L3GD20H_TEST

