

入門以前から始める PicoRuby のあそび方



@makicamel

PicoRuby Overflow会議

2025.07.19

自己紹介

- @makicamel / 川原万季
- Ruby 💎 とビール 🍺 とお酒が好き
- 好きな言葉は「エイってやってバーン」
- PicoPicoRuby始めました <https://picopicoruby.connpass.com/>
 - PicoRuby・電子工作歴 1ヶ月
 - ピカピカの初心者



RubyKaigi 2025 Lightning Talk



Road to RubyKaigi: Making Tinny Chiptunes with Ruby <https://rubykaigi.org/2025/presentations/lt/>

RubyKaigi 2025 Lightning Talk

● Road to RubyKaigi

- Rubyist による Rubyist のための Ruby 製のゲーム
- ターミナル上で遊ぶ横スクロールアクションゲーム
- バグを倒し〆切から逃れて RubyKaigi 参加を目指す

Road to RubyKaigi: Making Tinny Chiptunes with Ruby <https://rubykaigi.org/2025/presentations/lt/>
Road to RubyKaigi https://github.com/makicamel/road_to_rubykaigi

RubyKaigi 2025 Lightning Talk

● Ruby で音を作って鳴らすお話をした 🔊

- 音は波 🌊

- 波形データをつくる
オーディオ API に渡すと
音が鳴らせる

Making Music

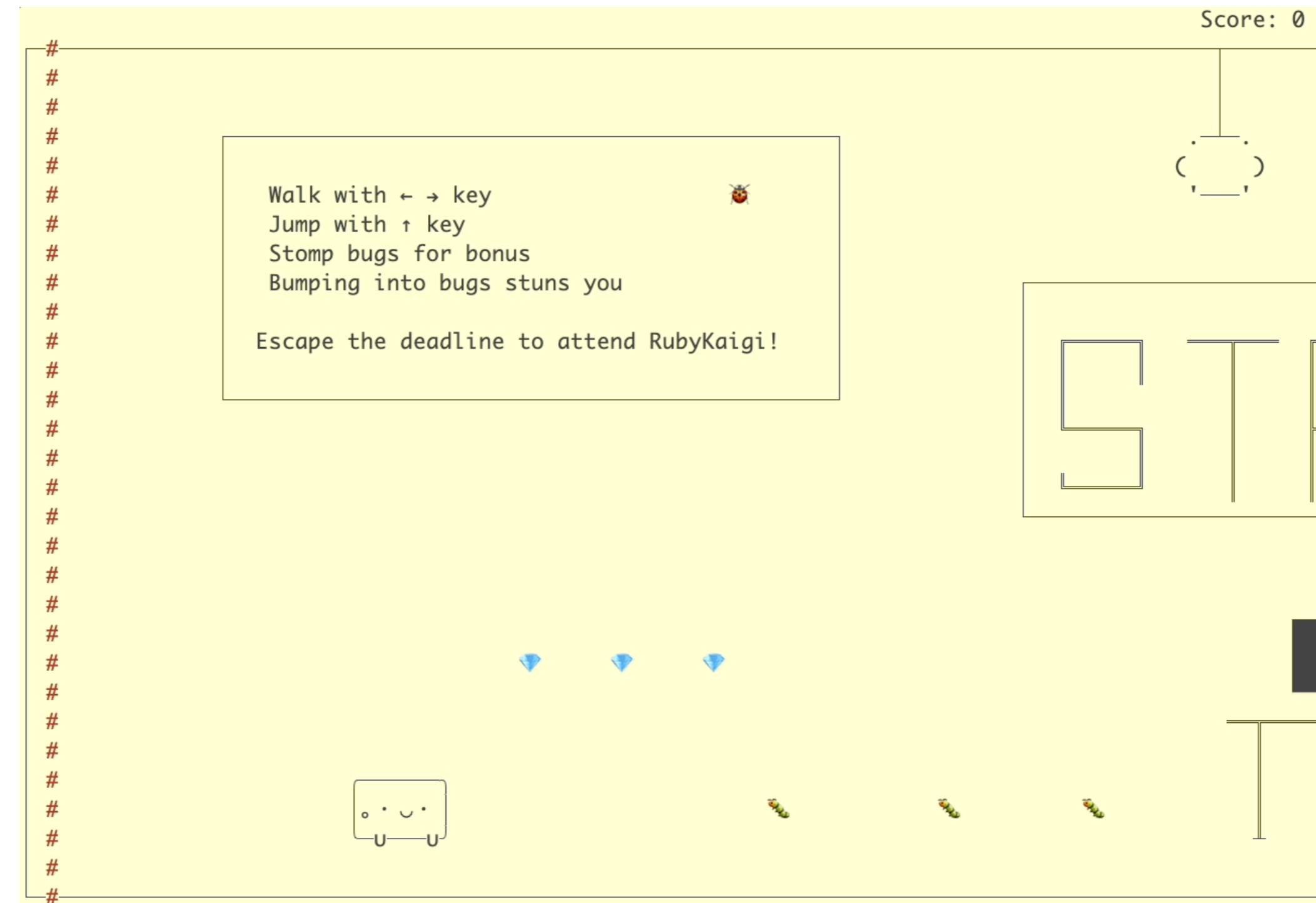
◆ Sound is a wave



Road to RubyKaigi: Making Tinny Chiptunes with Ruby <https://rubykaigi.org/2025/presentations/lt/>
Road to RubyKaigi https://github.com/makicamel/road_to_rubykaigi

Road to RubyKaigi

● デモ

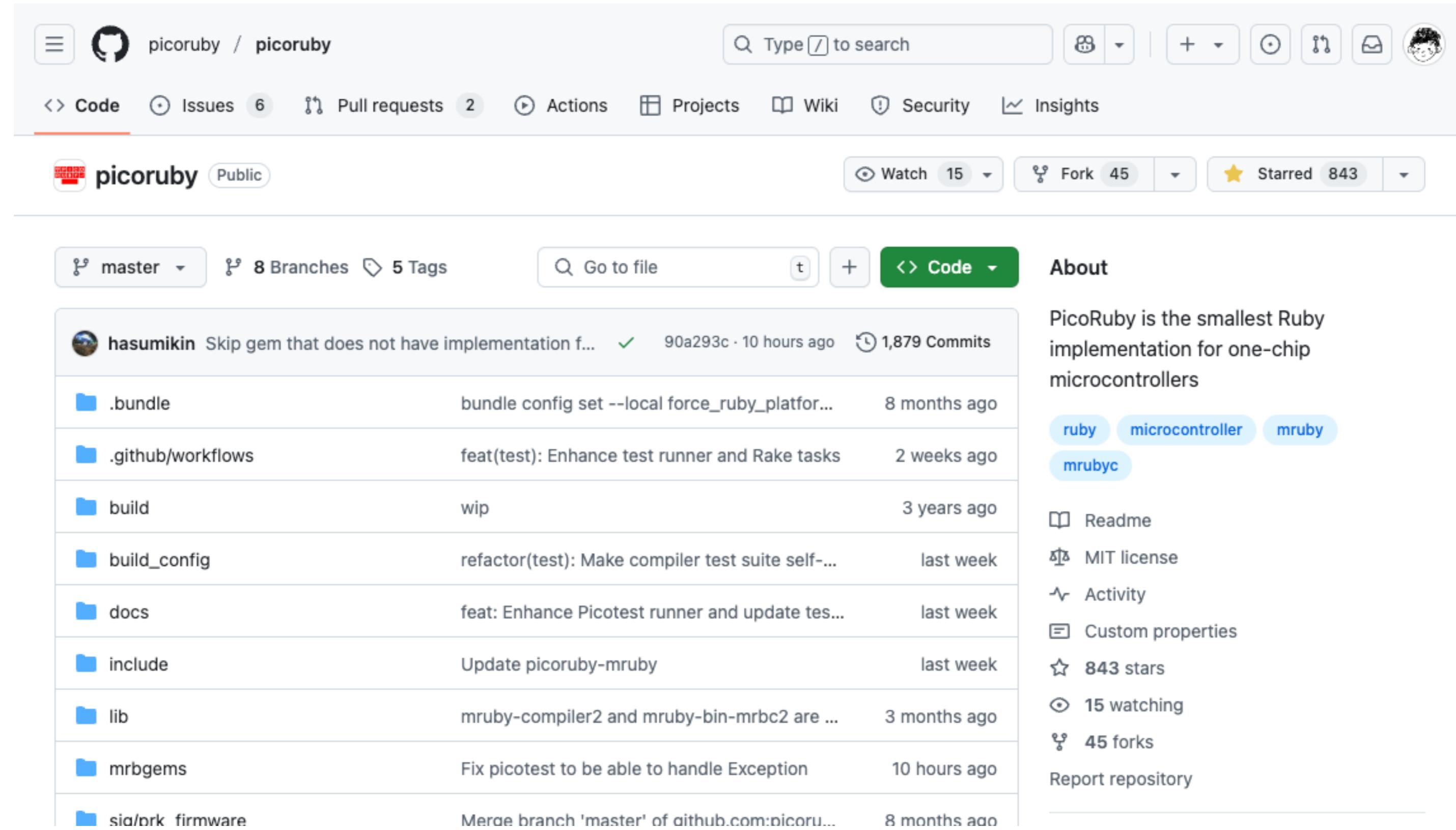


Road to RubyKaigi

ゲーム機もつくると
もっと面白いのでは



PicoRuby



The screenshot shows the GitHub repository page for "picoruby / picoruby". The repository is public and has 8 branches and 5 tags. The "Code" tab is selected. A recent commit by "hasumikin" is shown, along with a list of other commits across various files and folders.

Code Issues 6 Pull requests 2 Actions Projects Wiki Security Insights

picoruby Public Watch 15 Fork 45 Starred 843

master 8 Branches 5 Tags Go to file + <> Code About

Commit	Message	Time
 hasumikin	Skip gem that does not have implementation f...	90a293c · 10 hours ago
 .bundle	bundle config set --local force_ruby_platform...	8 months ago
 .github/workflows	feat(test): Enhance test runner and Rake tasks	2 weeks ago
 build	wip	3 years ago
 build_config	refactor(test): Make compiler test suite self...	last week
 docs	feat: Enhance Picotest runner and update tes...	last week
 include	Update picoruby-mruby	last week
 lib	mruby-compiler2 and mruby-bin-mrbc2 are ...	3 months ago
 mrbgems	Fix picotest to be able to handle Exception	10 hours ago
 sia/prk firmware	Merge branch 'master' of github.com:picoru...	8 months ago

PicoRuby is the smallest Ruby implementation for one-chip microcontrollers

ruby microcontroller mruby
mrubyc

Readme MIT license Activity Custom properties 843 stars 15 watching 45 forks Report repository

PicoRuby <https://github.com/picoruby/picoruby>

PicoRuby

- 軽量 Ruby 実装
 - マイコンで動作する
- @hasumikin さん作



Road to RubyKaigi ~First Mile~ のつくりかた

- @makicamel / 川原万季
- Ruby 💎 とビール 🍺 とお酒が好き
- 好きな言葉は「エイってやってバーン」
- PicoPicoRuby始めました <https://picopicoruby.connpass.com/>
 - PicoRuby・電子工作歴 1ヶ月
 - ピカピカの初心者



Road to RubyKaigi ~First Mile~

- @makicamel 川原万季
- Ruby 💎 とビール 🍺 とお酒が好き
- 好きな言葉は「エイってやってバーン」
- PicoPicoRuby始めました <https://picopicoruby.connpass.com/>
- **PicoRuby・電子工作歴 1ヶ月**
- ピカピカの初心者



電子工作未経験

データシートの読み方
わからん

マイコンボードって
どっちを上向きに
はんだ付けするの？

パーツすぐ壊れそうで
こわい

回路図読めない

Lチカってなに？

はんだづけ
こわい

何がわからないかもわ
からないし何もかもわ
からない

実はマイコンがなにか
よくわからない

モジュール…？

オームの法則？
学校で習ったような…



電子工作未経験

- ゴールまでの道筋が見えない
 - 何をどうしたら音が鳴るんです？
- 常識がない
 - とりあえず強い電圧かけたらいいんじゃないですか? (。∀°)
- 知らない単語・概念がとにかく多い
 - シリアル通信、ボーレート、GPIO、入出力ピン...

電子工作未経験

- ...というひとが音の鳴る小さなデバイスをつくった話をします
- トークのゴール
 - 初心者: ふんわりとした地図が描ける
 - ベテラン: 「初心者はこんなことがわからないんだ」という気づき

つくれみよう

- Road to RubyKaigi の音を鳴らすだけのデバイス
 - 波形データは前に作ったのでできそうな気がする
 - 音を鳴らすと楽しい

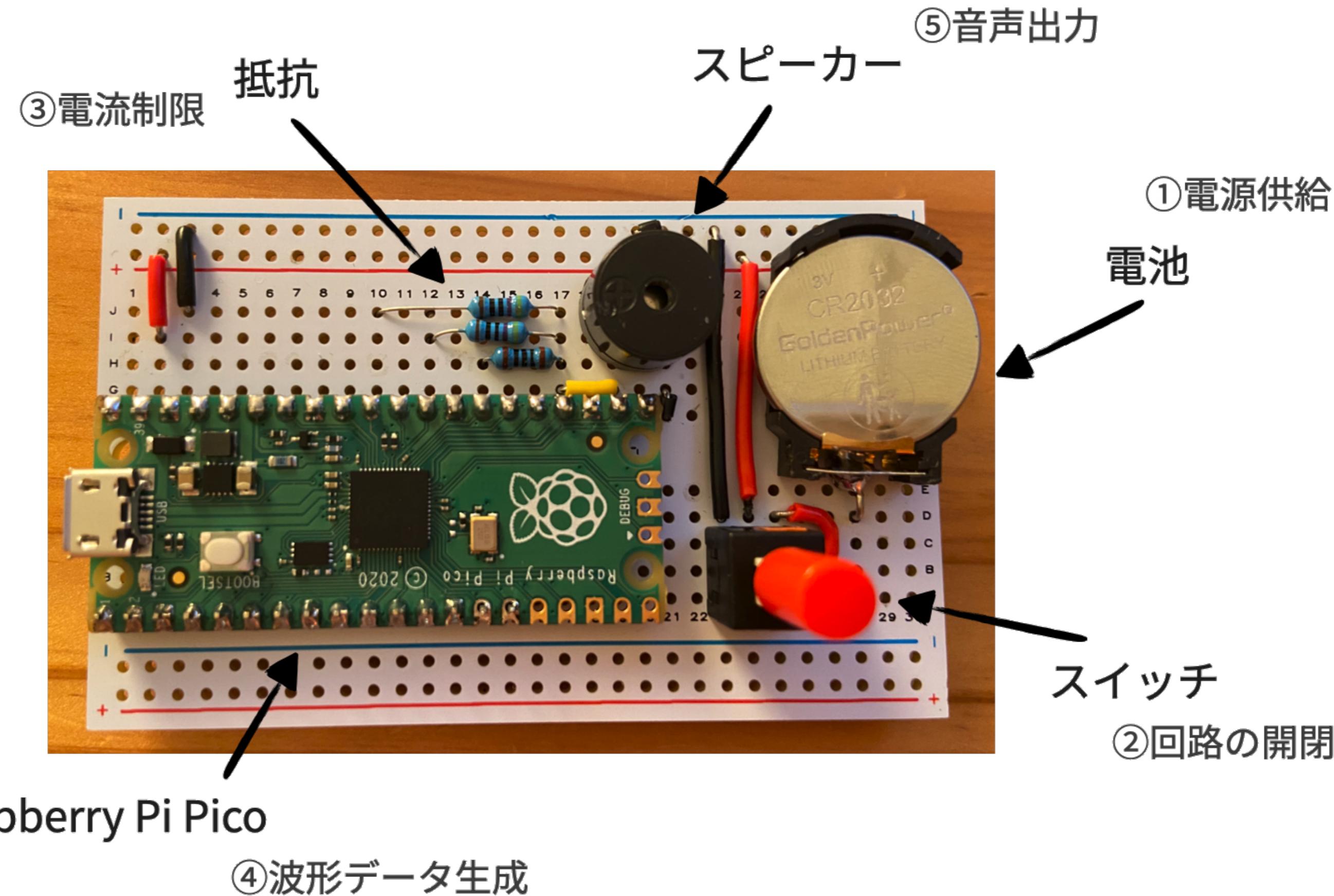
目次

1. 概観を眺める
2. PicoRuby のあそび方
3. 初心者のふりかえり

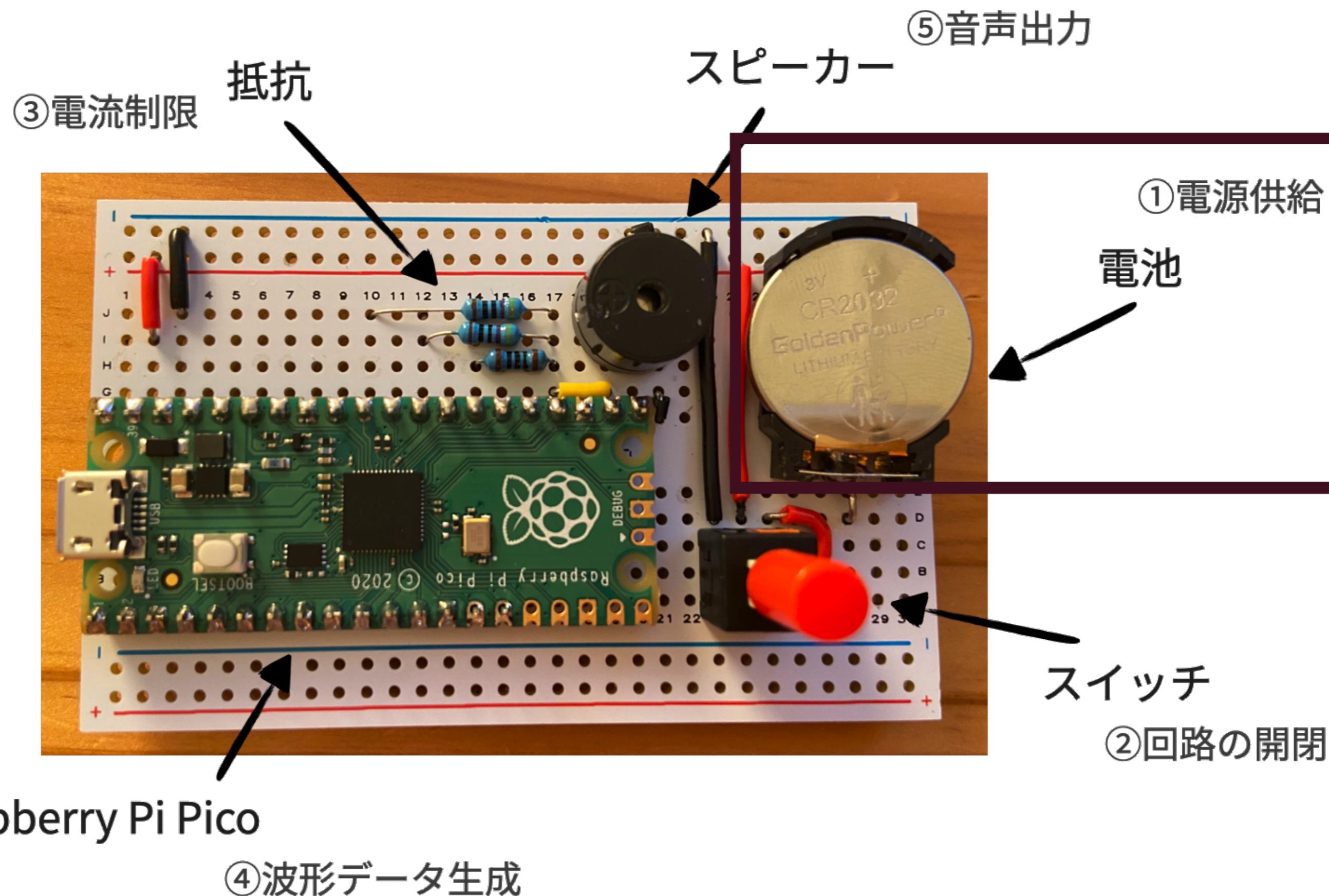
目次

1. 概観を眺める
2. PicoRuby のあそび方
3. 初心者のふりかえり

概観図



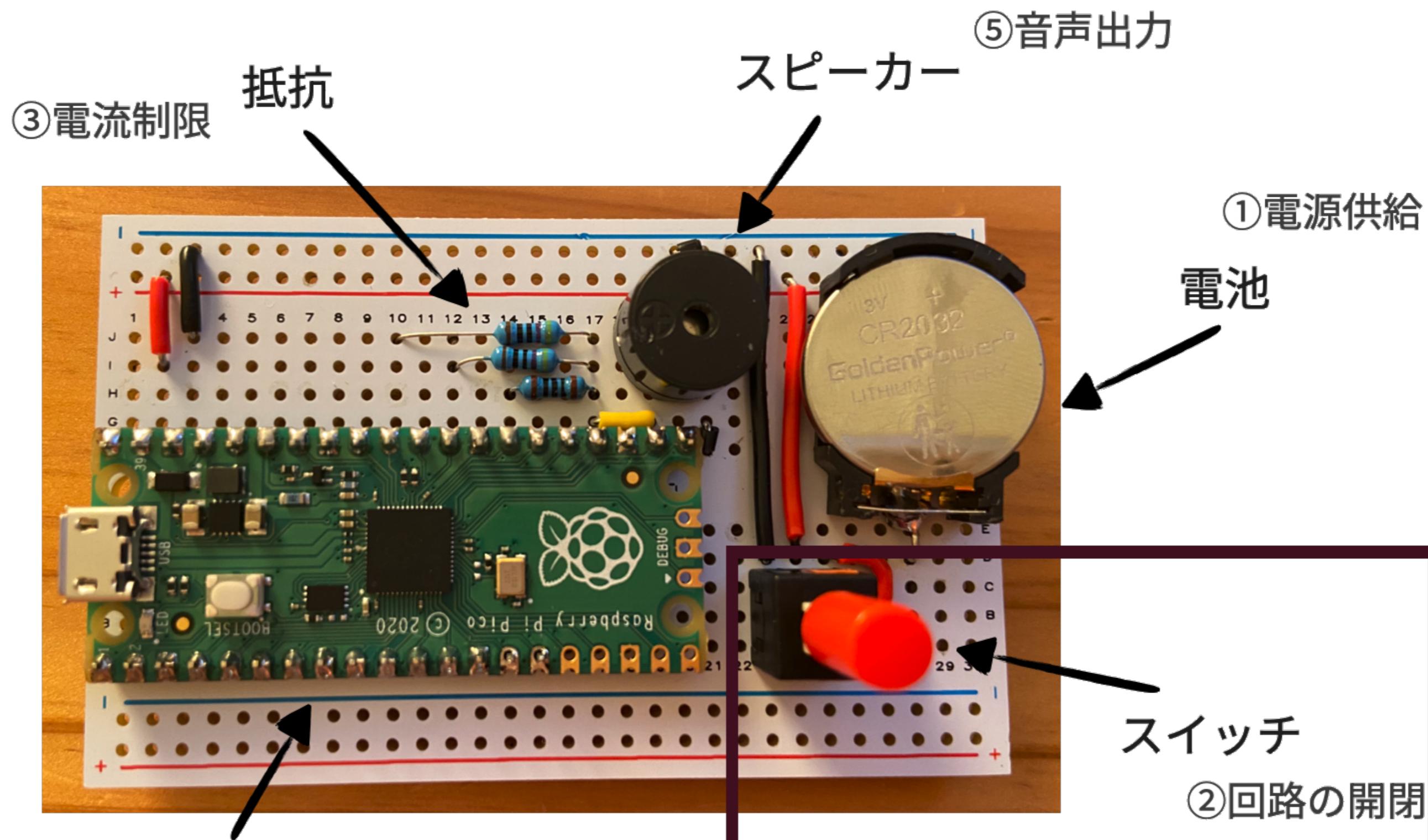
概観図 1. 電池



概観図 1. 電池

- 電圧 3V
 - Raspberry Pi Pico の起動には約 3V 必要
- 単三電池はだいたい 1 本 1.5V
 - 2 本直列でつないだら 3V
- 今回はリチウム電池を使用
 - 小さい基盤に乗せたかったので

概観図 2. スイッチ



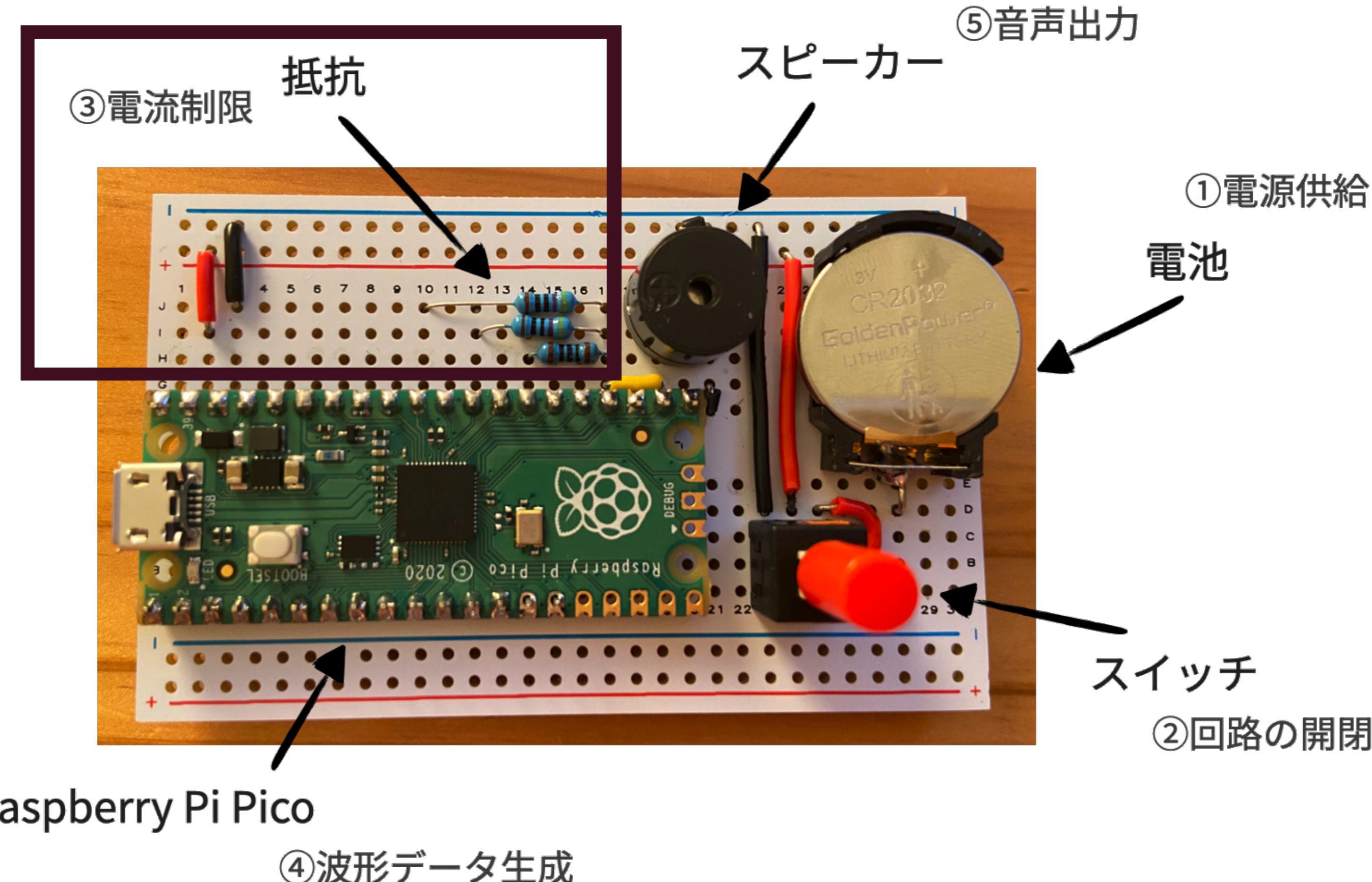
Raspberry Pi Pico

④ 波形データ生成

概観図 2. スイッチ

- 回路の開閉をする
 - 閉じていると回路がつながっている
 - Raspberry Pi Pico に電源が共有される 
 - 開いていると回路が切斷されている
 - Raspberry Pi Pico に電源が共有されない ■
- デバイス電源の ON / OFF 切り替えに使用

概観図 3. 抵抗



概観図 3. 抵抗

- 電源からpartsやマイコンに流す電流を制限する
 - 電流が小さすぎると動作しない 🤖
 - 電流が大きすぎるとpartsなどを破壊する 🤖
- partsには定格電流がある
 - 抵抗を利用して適切な電流を流す 🧑
- 定格電流はpartsのデータシートに記載がある

抵抗やpartsが熱くなったら
定格電流超過を疑う
(壊れる前に電源を外す)

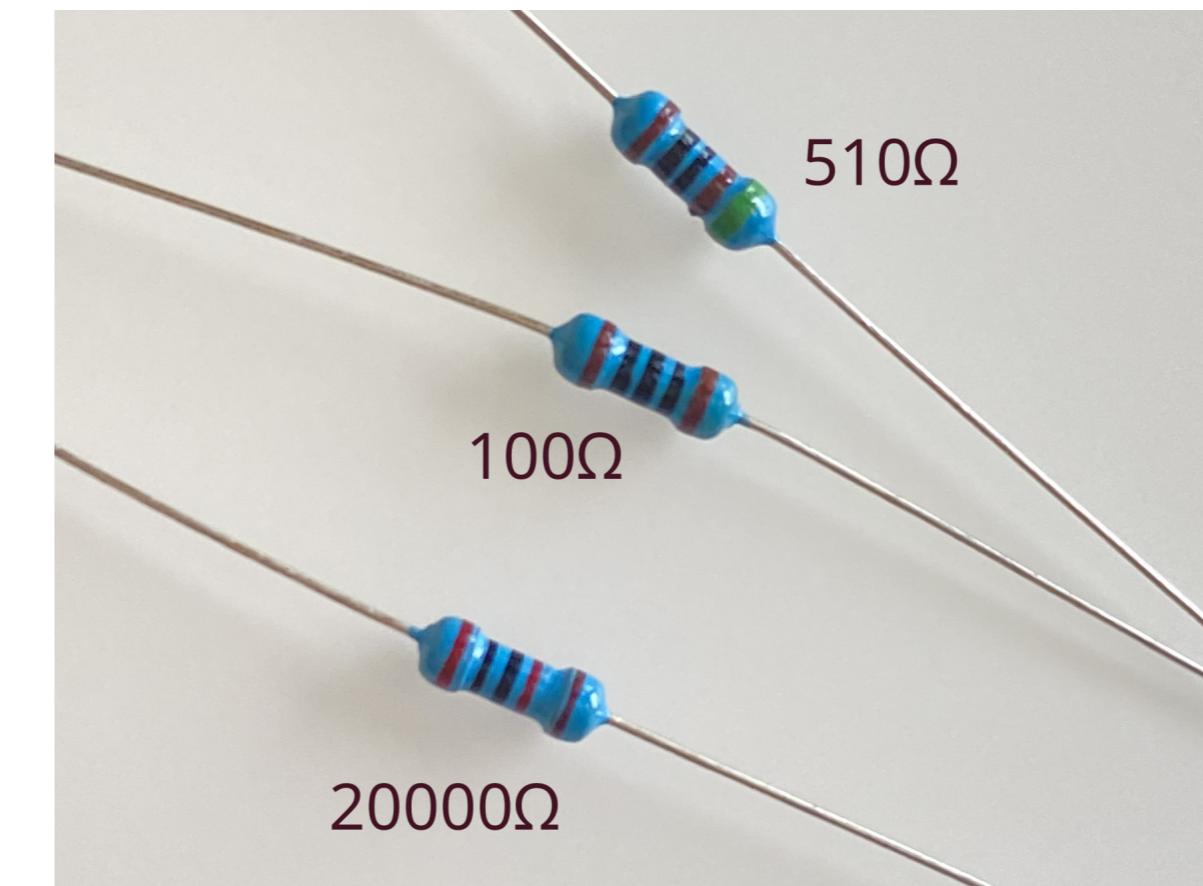


概観図 3. 抵抗

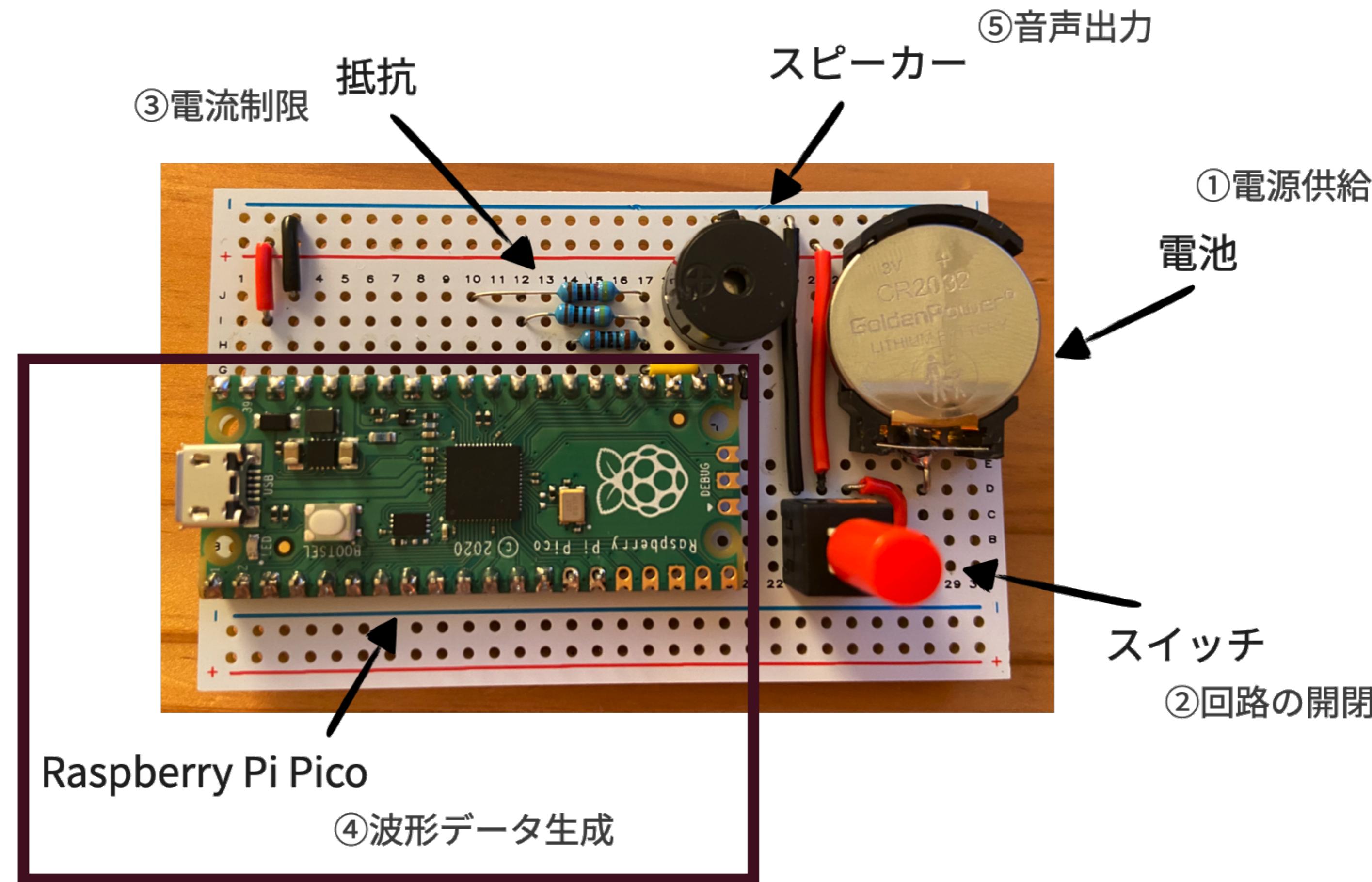
- 抵抗値はオームの法則で求める
 - $V = IR$
 - 電圧(V) = 電流(A) * 抵抗(Ω)
 - e.g. 電源電圧が3.3V、流したい電流が30mAの場合
 - $3.3V = 0.03A * R\Omega$
 - $R = 3.3 / 0.03$
 $\Rightarrow 110\Omega$
 - 110Ωの抵抗を使う

概観図 3. 抵抗

- 抵抗の抵抗値はカラーコードから読み取る
 - ref. 抵抗のカラーコード計算機 <https://akizukidensi.com/catalog/pages/reg-calc.aspx>
 - 判別むずすぎ問題
 - 赤と茶、青と黒...
- テスターでも確認可



概観図 4. Raspberry Pi Pico



概観図 4. Raspberry Pi Pico

- 波形データを作り電気信号として出力する
- PicoRuby でプログラミングできる

概観図 4. Raspberry Pi Pico

Raspberry Pi Pico

- マイコンボード
- RP2040 搭載
- USB で電源供給とプログラム書き込み可

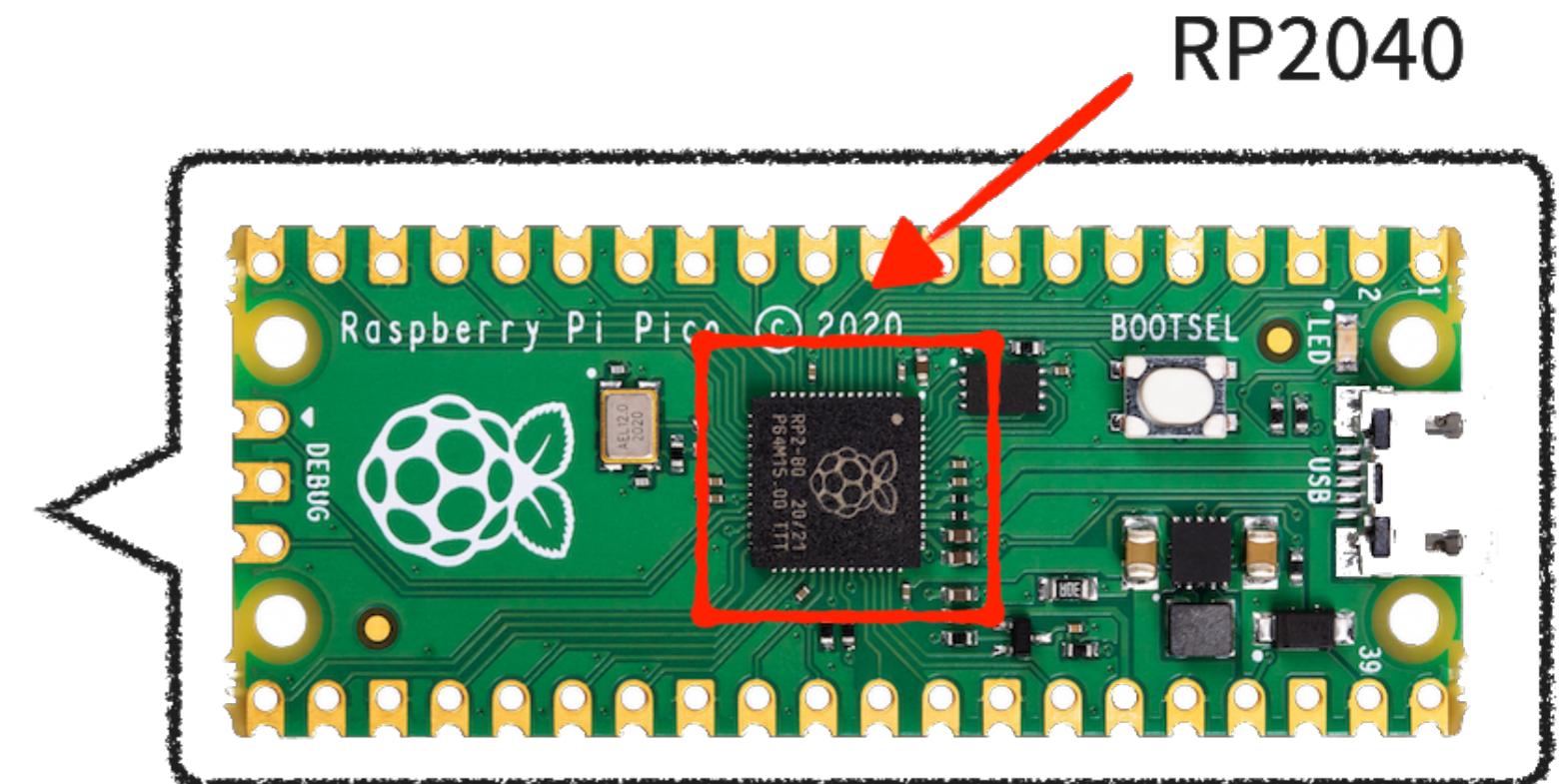
Raspberry Pi Pico <https://www.raspberrypi.com/products/raspberry-pi-pico/>

概観図 4. Raspberry Pi Pico

Raspberry Pi Pico

- マイコンボード
 - RP2040 搭載
 - USB で電源供給とプログラム書き込み可
- RP2040
 - マイコンチップ
 - PicoRuby でプログラミングできる 💎

Raspberry Pi Pico

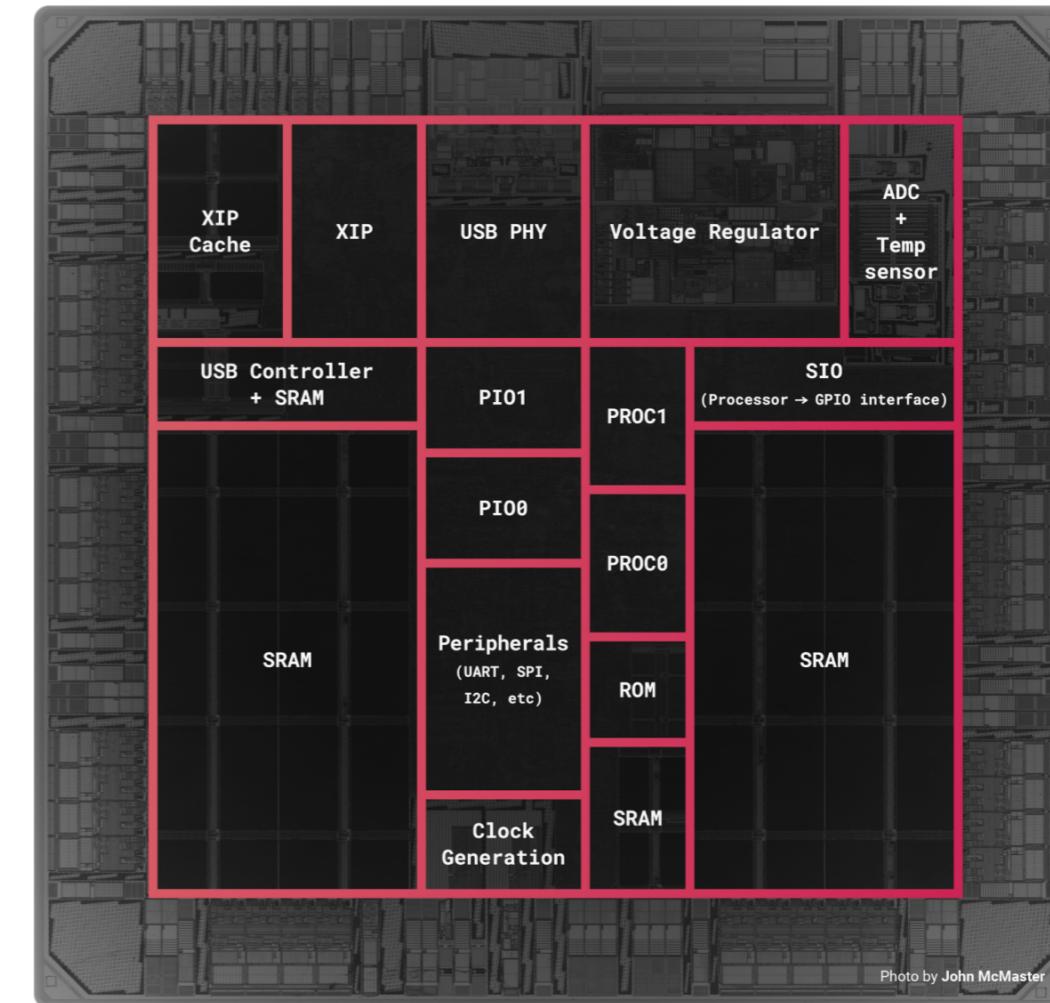


Raspberry Pi Pico <https://www.raspberrypi.com/products/raspberry-pi-pico/>

概観図 4. RP2040(Raspberry Pi Pico)

- 7mm²に CPU・メモリ・ペリフェラルなどがつまっている
- ペリフェラル: 機能専用ハードウェア
 - デジタル I/O
 - PWM
 - 通信インターフェース
 - etc

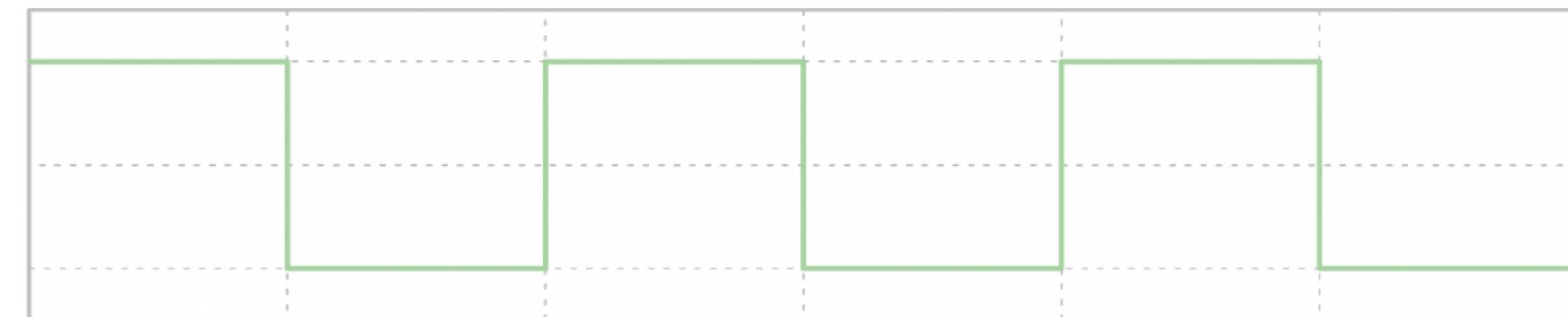
RP2040 <https://www.raspberrypi.com/products/rp2040/>



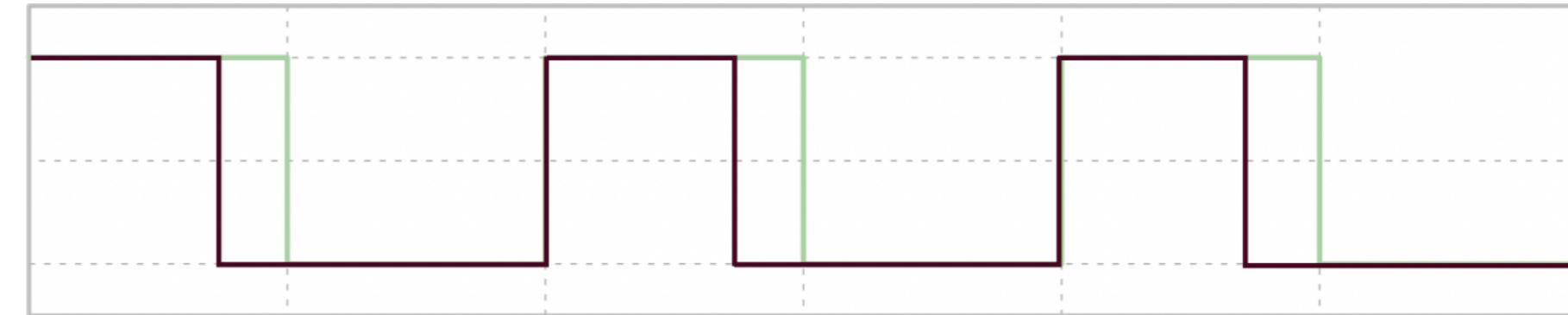
概観図 4. RP2040(Raspberry Pi Pico)

- PWM モジュール
- Pulse Width Modulation
- 一定の周波数・電圧 High の比率で信号を生成する
 - e.g. 赤外線リモコン
 - 赤外線信号パターンを指定
 - e.g. モーター制御
 - 平均電圧を変えて回転数を制御

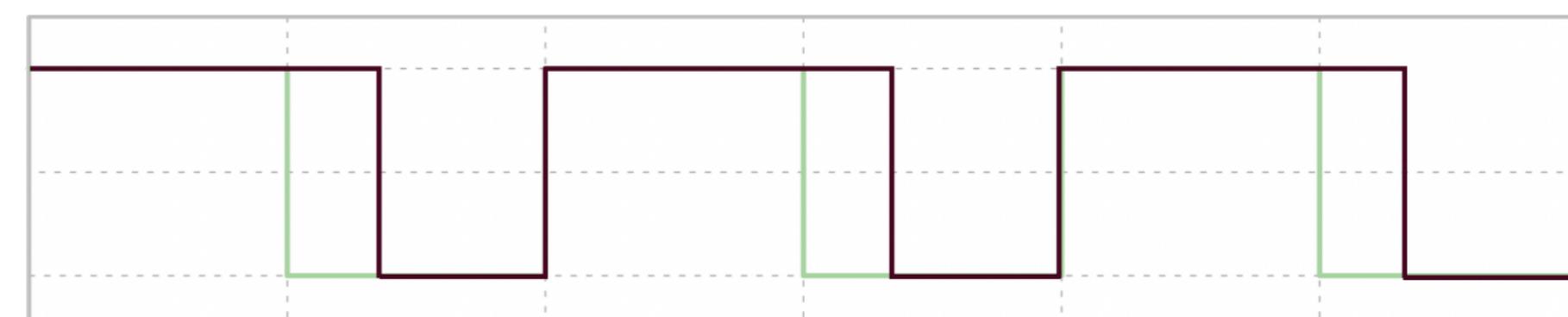
電圧HighとLowの比率が1：1



電圧Highの比率を下げる（平均電圧が下がる）



電圧Highの比率を上げる（平均電圧が上がる）

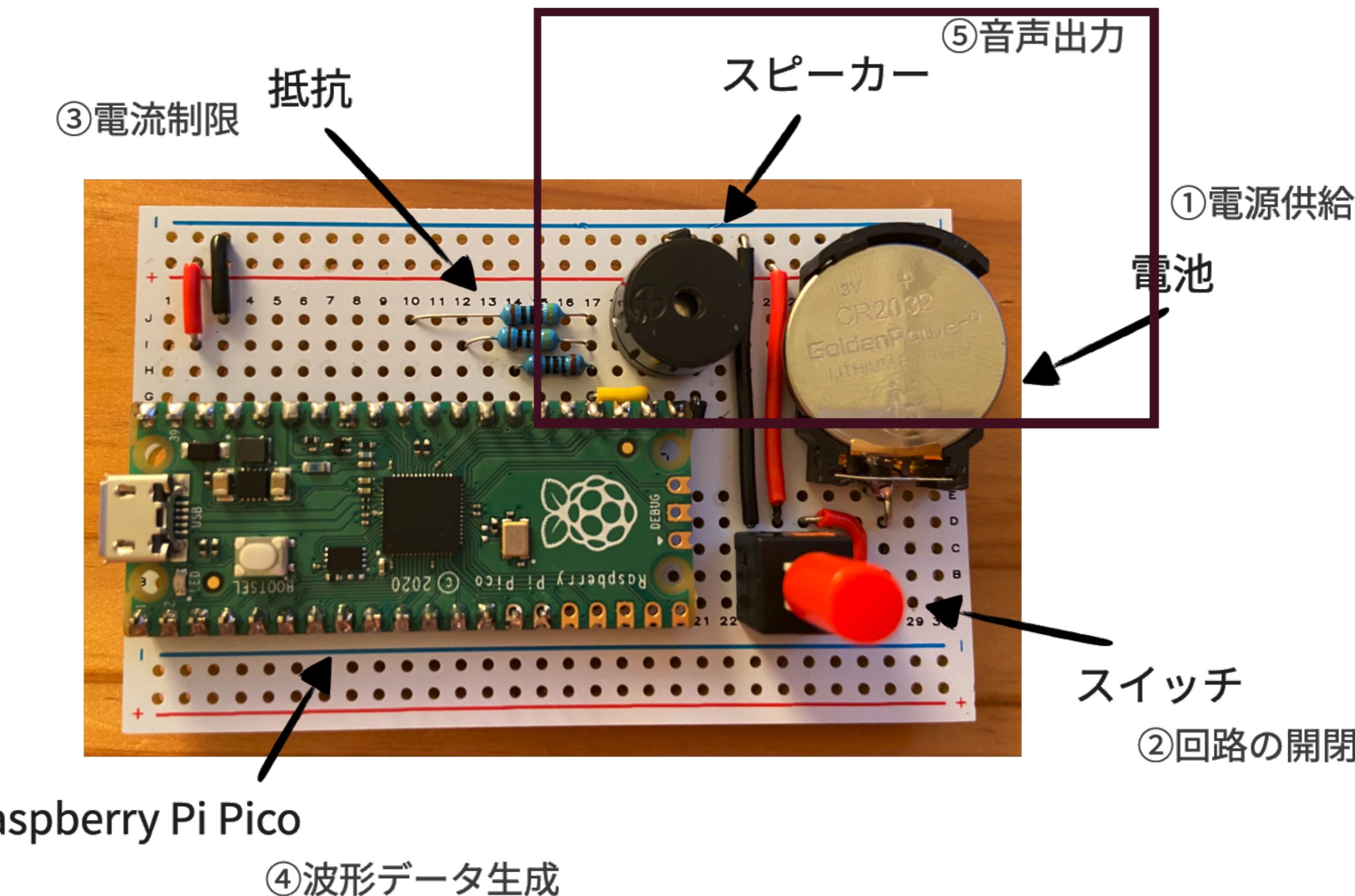


概観図 4. RP2040(Raspberry Pi Pico)

- PWM モジュール
- PicoRuby で波形データを生成 

```
# PWMモジュールで0.5秒ずつド,レ,ミの波形データを生成する
require 'pwm'
pwm = PWM.new(0, duty: 50)
[262, 294, 330].each do |frequency| # ド,レ,ミの周波数
    pwm.frequency(frequency)
    t = Time.now
    while (Time.now - t) < 0.5; end # 0.5秒ビギループさせる
end
```

概観図 5. スピーカー



概観図 5. スピーカー

- 音声出力装置
- PWM モジュールで生成した波形データを電気信号として入力する

概観図

完成



目次

1. 概観を眺める
2. PicoRuby のあそび方
3. 初心者のふりかえり

PicoRuby の遊び方

- 1. Raspberry Pi Pico を USB でパソコンとつなぐ
- 2. R2P2^[^1]を Raspberry Pi Pico にインストール
 - 最新リリースから uf2 ファイルをダウンロードしてドラッグ & ドロップで完了
- 3. シリアル通信で接続
 - mac だと `TERM=vt100 screen /dev/tty.usbmodem*`
- 4. シェル起動

[^1]: PicoRuby 用シェルシステム <https://github.com/picoruby/R2P2>



PicoRuby の遊び方

- irb
 - `puts 'Hello, PicoRuby!'`
- /home/app.rb
 - このパスにファイルを置くと Raspberry Pi Pico 起動時に実行される
- ドキュメント <https://picoruby.github.io/>

PicoRuby の遊び方

- irb で遊んでいると Out of Memory に遭遇しやすい
 - Raspberry Pi Pico のメモリは 256KB
 - ref MacBook Pro (M3 Pro) のメモリは 32GB
 - 1 / 125000
- 大きなオブジェクト生成は避けたほうがよい

PicoRuby の遊び方

- Road to RubyKaigi の BGM
- 元実装は配列を value としたハッシュの配列
- メロディー 1 音、ベース 2 音
- このままだとメモリ足りないかも

```
class MelodySequencer < SequencerBase
  GENERATOR = RoundedSquareOscillator
  STACCATO_RATIO = 0.35
  SCORE = [ # 6 Measures
    { frequency: %i[F5], duration: 0.5, envelope: { a: 0.05, d: 0.0, s: 0.5, sl: 0.4, rl: 1.0 } },
    { frequency: %i[C5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[C5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[C5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.25 },
    { frequency: %i[F5], duration: 0.25 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[G5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[C5], duration: 0.5 },
    { frequency: %i[F5], duration: 0.5 },
    { frequency: %i[REST], duration: 0.5 },
    { frequency: %i[A5], duration: 0.5 }]
```

PicoRuby の遊び方: データ圧縮

- 使用する音は 2 オクターブ 14 音 + 休符の 15 音
 - $2^4 \Rightarrow 16$ なので 1 音を 4bit で表せる
- 音程に番号を割り当て 16 進数で表す
 - e.g. E4 → 2(10 進数) → 0b0010(2 進数) → 0x2(16 進数)

```
NOTE_MAPPING = {  
    C4: 0, D4: 1, E4: 2, F4: 3, A4: 5, B4: 6, REST: 4, # ...  
}
```

PicoRuby の遊び方: データ圧縮

- 音階を 16 進数の配列に変換
 - e.g. [C4, D4, E4] → [0x0, 0x1, 0x2]
- 左シフトしてひとつの Integer にする
 - e.g. [0x0, 0x1, 0x2] → 0x210(16 進数) → 528(10 進数)
- 小さくなった!

PicoRuby の遊び方: データ圧縮

- 音の長さも同様に圧縮
 - 十六部音符: 0, 八分音符: 1, 四分音符: 2
- エンコードした Integer をコードに埋め込み

```
SCORES = [
    { notes: [45589323163098, 120693318330058, ...]}, # メロディー
    { notes: [72588760057059, 72588760057059, ...]}, # ベース 1
    { notes: [85991239857509, 85991239857509, ...]}, # ベース 2
]
```

PicoRuby の遊び方: データ圧縮

● 再生時にデコード

```
DECODER = Proc.new do |packed, shift|
    # 音程インデックスと音の長さをセットで取り出す
    index_duration = (packed >> (shift * NOTE_BITS)) & 0b111111 # デコード時は右シフト
    [
        NOTE_MAPPING[index_duration & 0b1111], # 音程の番号
        0.25 * 2 ** ((index_duration >> FREQUENCY_BITS) & 0b11) # 音の長さ
    ]
end
```

● 無事音が鳴った

<https://github.com/makicamel/picoruby-phrase/blob/main/player.rb>

目次

1. 概観を眺める
2. PicoRuby のあそび方
3. 初心者のふりかえり

初心者のふりかえり

- 勢いよく進むタイプの話者が勢いよくハマってきたこととそのふりかえり

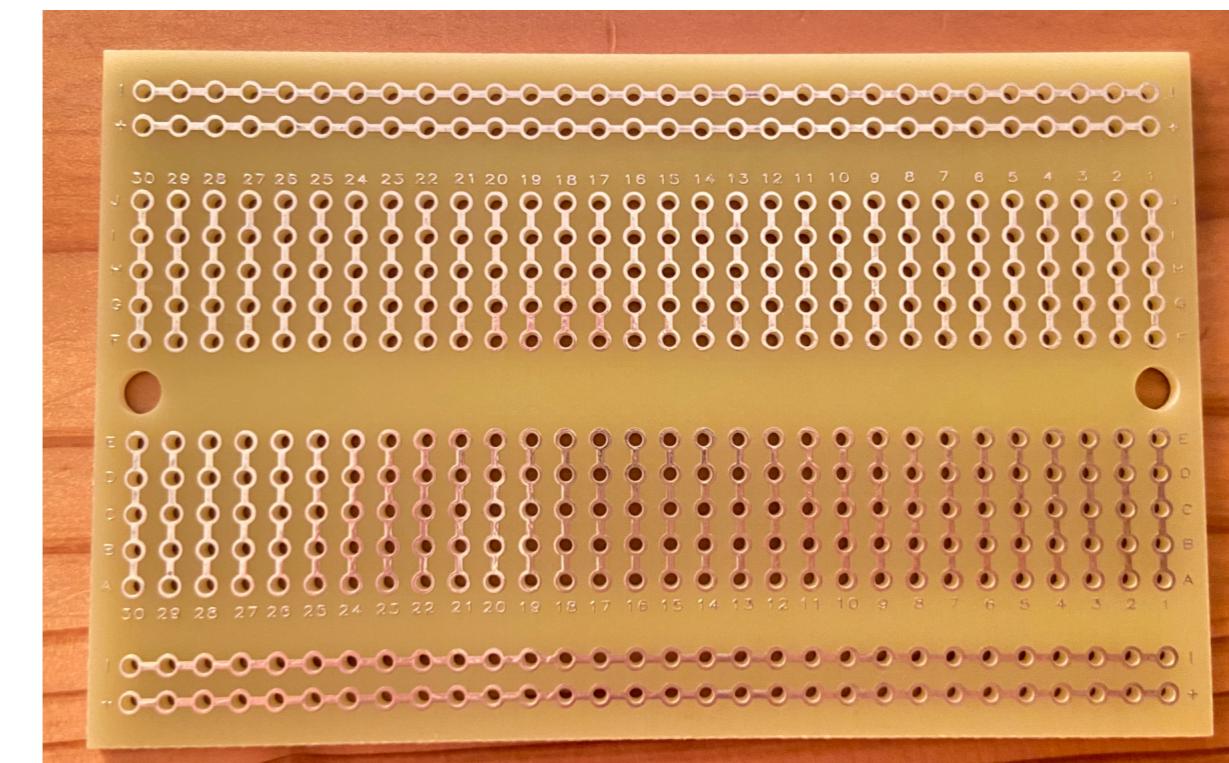
初心者のふりかえり

- はんだづけが下手 😞
 - 正しい回路のはずなのに電圧が低い
 - はんだが溶けない・つかない
 - やり直しすぎてラウンドが溶けて電導体が無
- YouTube べんり 😊
 - How To Solder Electronic Component. Tutorial^[^2] はとくにお勧め
 - 自分のはんだづけがうまくいかない理由・対処法がわかる

[^2]: <https://www.youtube.com/watch?v=dQ7AUjb1tkA&t=1323s>

初心者のふりかえり

- はんだづけが下手 😞
- 両面スルーホール基盤を使う 😊
- 片面スルーホール基盤は非初心者向け
- ブレッドボード基盤を使う 😊
 - ブレッドボードのように縦横にラウンドホールが繋がっている
 - はんだづけ負担が減る



初心者のふりかえり

- 配線が下手 😐

- よくわからないけどごちゃついている

- 対策 😊

- 電源用配線は専用にして混ぜない

- 電源、グラウンド、それ以外の色を決める

- 配線は短いほうがいい

- パーツの下をくぐらせるとメンテしづらいので避ける

初心者のふりかえり

- 回路のデバッグができない 😐
- 意図通り動かない時にどう調査したらよいかわからない
- 何はともあれテスター 😊
- 意図通りの電圧がきているか?
 - 配線が間違っているかも
 - はんだづけがうまくできていないかも
 - 電池がへたっているかも
- 意図通りに導通しているか?
 - PWM は電圧が変化するので電圧ではなく導通チェックする

初心者のふりかえり

- 回路のデバッグができない 😐
- 大きな故障箇所から範囲を狭めて問題箇所を探る 😊
- プリントデバッグと一緒に

初心者のふりかえり

- とりあえず全部配線したけどうまく動かない 😞
 - どこに原因があるかさっぱりわからない
- とりあえず全部配線しない 😊
 - 1ステップずつ確かに動作することを確認する
 - パーツが壊れている可能性もある
 - プログラミングで単体テストするのと同じ

初心者のふりかえり

- データシートが読めない 😕
 - 常用単語がわからない
 - 読み方がわからない
 - 何が書いてあるかわからない
- 欲しい情報を明確にしてから臨むとよさそう 😊
 - 定格電流
 - ピン番号
 - etc

初心者のふりかえり

- パーツショップに何も調べず行く 😕
 - えっ、パーツの種類、多すぎ...?
- 欲しい部品を事前に調べる 😊
 - 商品コードをメモしておくとスムーズ
 - 見ているだけでも楽しいのは楽しい
 - 詳しい人と行くと発見がたくさんあって楽しい
 - 7月20日にアフターイベントやります

初心者がハマらなかつた穴

- 聞く人がいない
- 毎日 @youchan に相談していた 😊
 - マイコンボードの裏表からデバッグの仕方、通信プロトコルまで
 - 物理デバイスは特に相談できる人がいたほうがいい
- コミュニティ
 - naniwa.rb(大阪) <https://naniwarb.doorkeeper.jp/>
 - PicoPicoRuby(東京) <https://picoricoruby.connpass.com/>
- Twitter でつぶやくと拾ってくれるひともいるかも

Special Thanks

● @youchan

Happy Hacking!