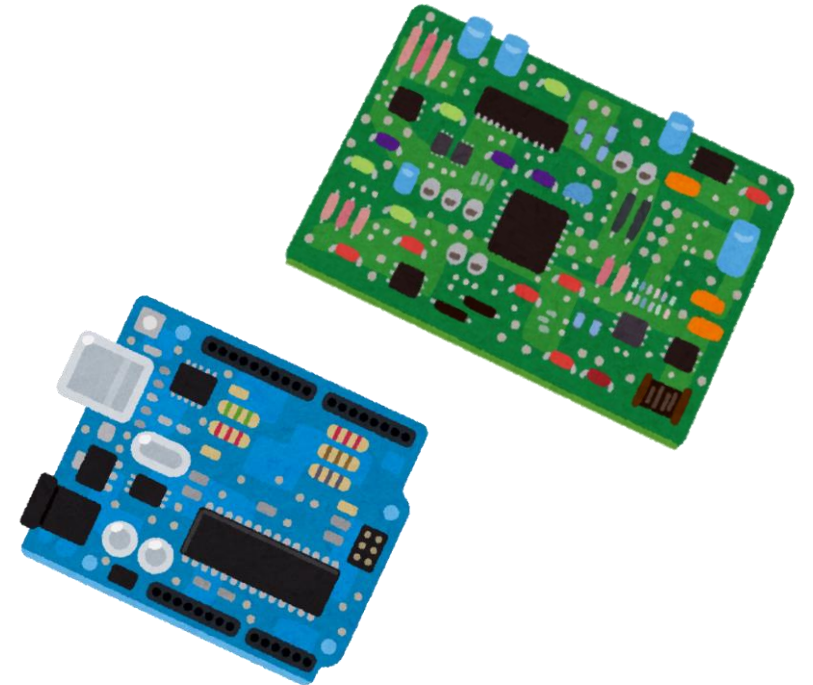
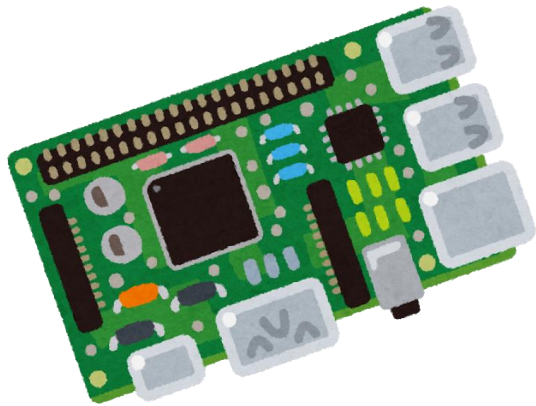


EmbLT

SWEST24



EmbLTって？

- 技術系イベントではお馴染みの**Lightning Talks**！
 - そんなLTのSWEST版こと **Embedded Lightning Talks** → **EmbLT**
- セッションやポスターほど頑張れなくてもOK！
- 気になるネタを見つけたら、是非LT終了後の徹夜部屋で続けて議論もOK！
- 楽しく！ワイガヤでやっていきましょー！

発表順

1. フィールドロボットの研究開発で試される忍耐力
2. 電子楽器の作品紹介2022
3. Elixir/Nerves衛星を打ち上げる日
4. FAをびよんどする
5. ギャラクシアンをIoT化する
6. EV3 Train
7. バイクツーリングログをGoogleEarthで可視化してみた。～KML瞬間解説～
8. ~~(予備スロット)~~

藤永 拓矢

石垣 良

山崎 進

菊池 豊

菊地 俊介

久保秋 真

曾根 卓朗

※. **赤字**はオンラインからの発表

参加者の皆さんへのお願い

- Slackへのコメント歓迎！ **自由に・積極的に書き込み下さい**
 - 発表後に拾いたいと思います
- **お酒 → ドリンク片手**にワイガヤで楽しくやりましょう！
 - オンライン側は自由に呑んで下さい😊
- オンライン発表者はカメラを常にオンで！
- **途中退出 & 途中入出OKです。** お気軽にどうぞ

Slackでリアクション & 質問歓迎！

- オンライン側の発表者は孤独…
- 皆さん，チャットで是非リアクション & 質問をくれると嬉しいです
- 例示
 - [拍手] 8888888888
 - [イイネ] bbb
 - [絵文字] 😊😊😊 🍷🍷 などなど

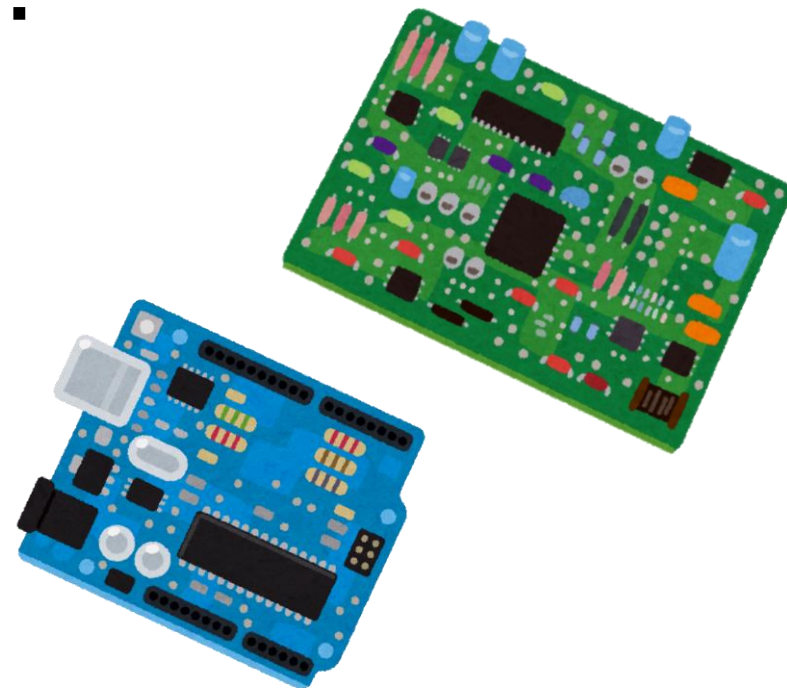
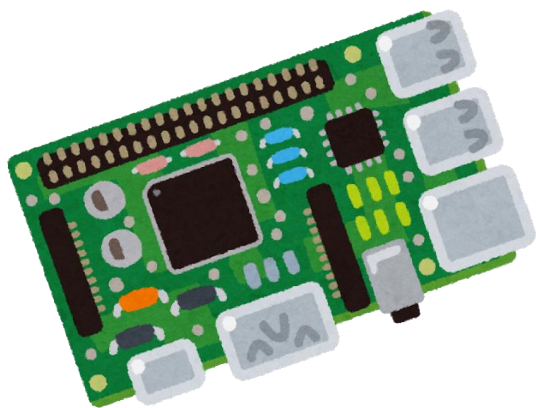
SNSへの発信は大歓迎！

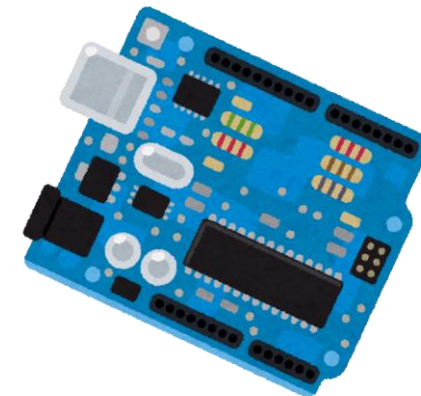
- 本セッションはSNSへの発信大歓迎です ☺

- ハッシュタグは **#SWEST24** **#EmbLT**
- スクショもOKのスタンスです

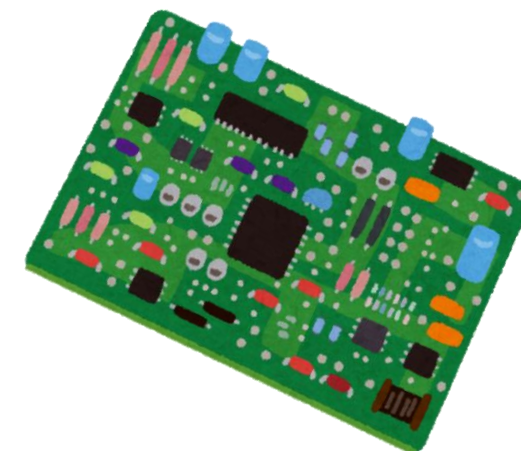
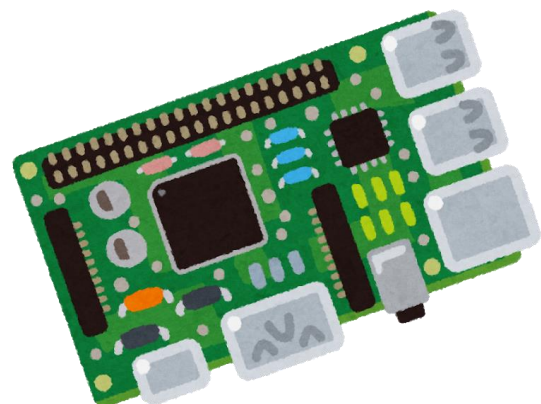
- LT発表者が「これはNGで…」と言ったモノはSNSへの発信を控えて頂けるようお願いします

い、お、LT!!





SWEST25 での
LT発表 &
インタラクティブセッションへの
ご参加お待ちしております！





フィールドロボットの研究開発で 試される忍耐力

福岡大学 藤永拓矢

目次

- 自己紹介
- フィールドロボットについて
- フィールドロボットとロボット競技会
- 研究開発事例
 - 農業ロボット
 - その他
- まとめ
- さいごに

自己紹介

■ 藤永 拓矢 (Takuya Fujinaga)

- 2012年4月—2021年3月
九州工業大学
(学士@飯塚キャンパス、修士・博士@若松キャンパス)
- 2021年4月—現在
福岡大学 工学部電子情報工学科 助教

■ 専門

- フィールドロボティクス
- 農業ロボットに関する研究開発

■ 趣味

- ランニング



7年前...

フィールドロボットについて

- 様々な要素技術を集約
- 対象とするフィールドへの理解
- **体力的、精神的な忍耐力**



Boston DynamicsのSpot^[1]



水中ロボットによる
福島第一原発の調査^[2]

[1] YouTube: Spot's Got an Arm!, <https://www.youtube.com/watch?v=6Zbhvaac68Y&t=59s>

[2] YouTube: 福島第一原発1号機でロボット使い調査, <https://www.youtube.com/watch?v=6ibrbAXYTjY>

フィールドロボットとロボット競技会

ロボット工学や知能工学など様々な分野の発展とその技術の普及、研究開発の成果の検証などを目的として、

- **ロボカップ**

サッカー、レスキュー、@ホーム、...

- **トマトロボット競技会**

生もの(トマト)を扱う珍しい競技会

- **水中ロボコン**

国内外／実海域でも実施

- **学研ヒルズ学際駅伝大会**

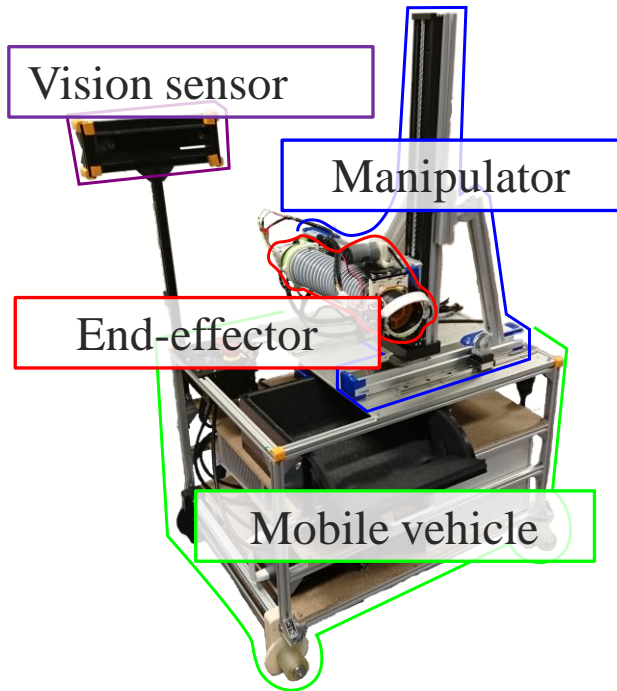
人間、動物、ロボットが共同して競技

などなど **様々なフィールドを対象としたロボット競技会**

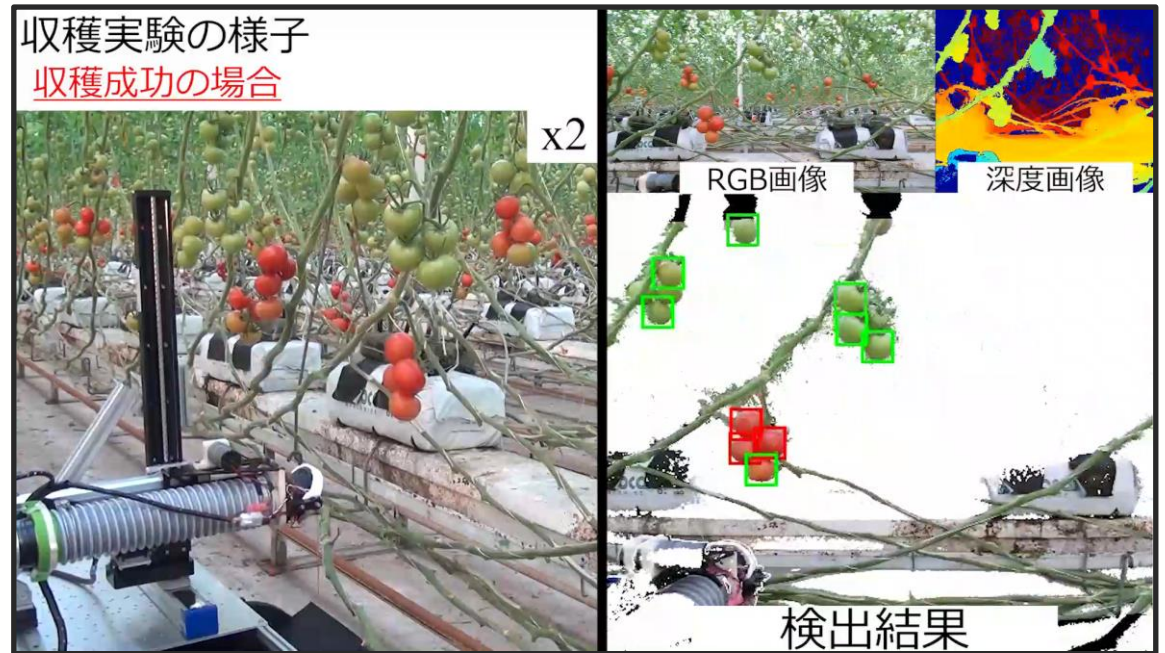
研究開発事例: 農業ロボット

■ 収穫ロボット

大規模施設園芸を対象として中玉トマトを自動収穫するロボット



収穫ロボット

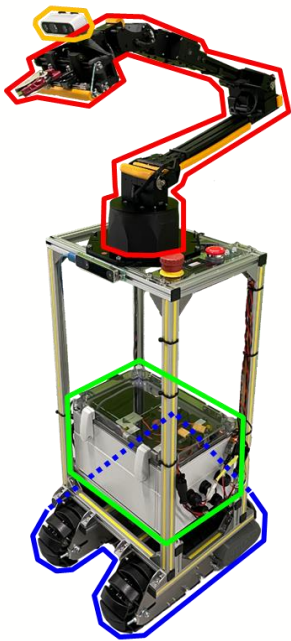


トマト菜園で実施した収穫実験のまとめ

研究開発事例: 農業ロボット

■ 小型農業ロボット

小規模施設園芸を対象とした汎用的なロボット



小型農業ロボット

SWEST24 インタラクティブセッション 自由工作部門

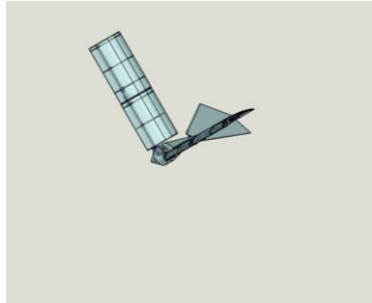
小規模施設園芸を対象とした
汎用的な小型農業ロボットの開発
藤永拓矢, 中西恒夫 (福岡大学)

農業ロボットのデモンストレーション
— トマトの半自動収穫 —
※検出～収穫動作は自動

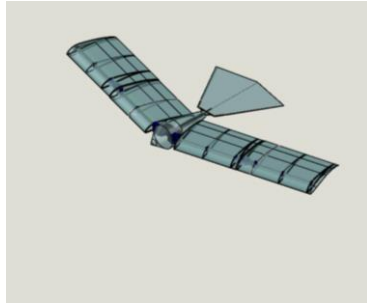
研究開発事例: その他

■ 羽ばたきロボット

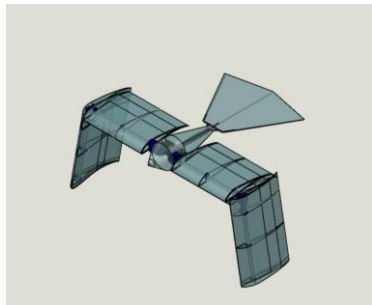
鳥の羽ばたき動作を模倣して飛行を目指したロボット



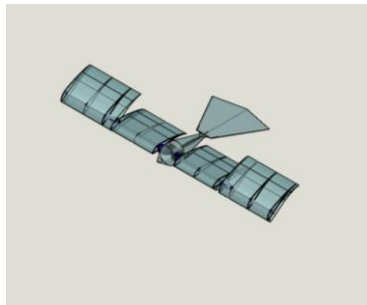
フラッピング
(上下動作)



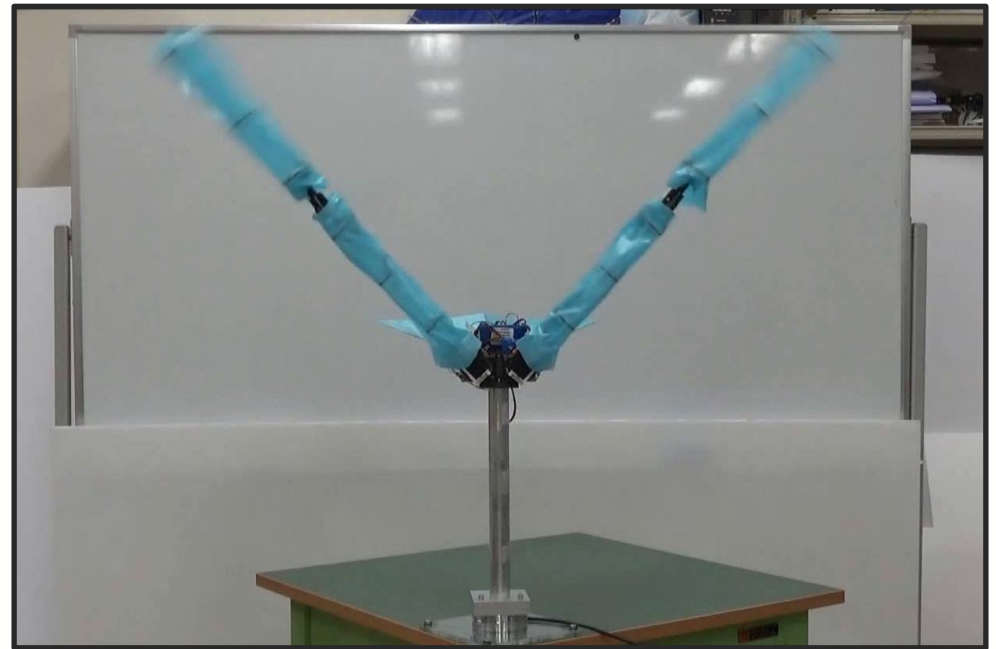
リードラグ
(前後動作)



フレクション
(曲げ動作)



フェザリング
(捻り動作)

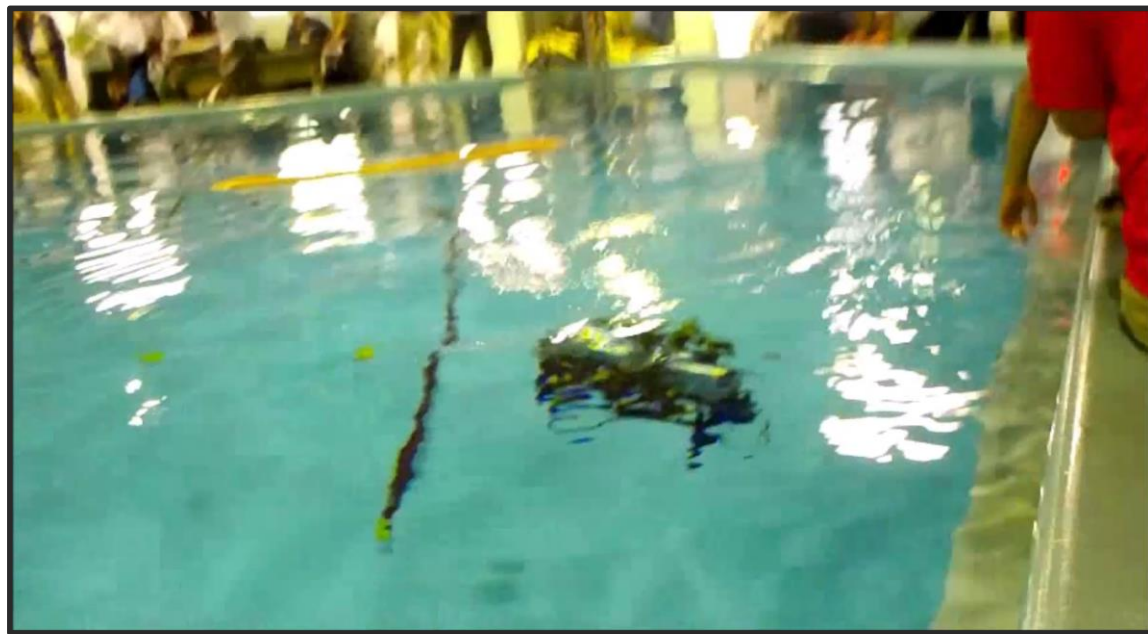
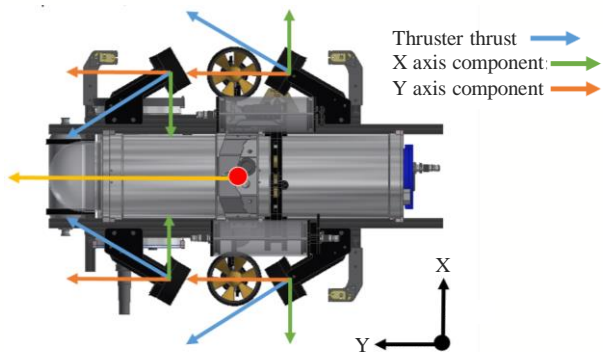


羽ばたき動作

研究開発事例: その他

■ 水中ロボット

水中環境下における自律動作を目的に開発したロボット



水中ロボット競技会の様子



ロボットの外観

まとめ

- フィールドロボットの研究開発において
 - 忍耐強く...
 - ハードウェア・ソフトウェアを開発すること
 - フィールドで検証すること
 - 小さな気づきを大切にすること
 - とは言え...
 - 先人(先行研究／開発)のリソースを活用すること
 - 時には諦めること
- も必要

さいごに... (宣伝です)

- トマトロボット競技会 (毎年12月上旬に開催)

第8回のHP:

<https://www.lsse.kyutech.ac.jp/~sociorobo/tomato-robot2021/>

- 学研ヒルズ学際駅伝大会 (毎年5月中旬に開催)

第7回のHP:

<https://www.lsse.kyutech.ac.jp/~sociorobo/gakkenekiden/ekiden2022/>

- 「50年後、人とロボットは？」はがき絵コンクール

第1回のHP:

<https://www.lsse.kyutech.ac.jp/~sociorobo/gakkenekiden/robot-hagakie-2022/>

電子楽器の作品紹介2022

石垣 良

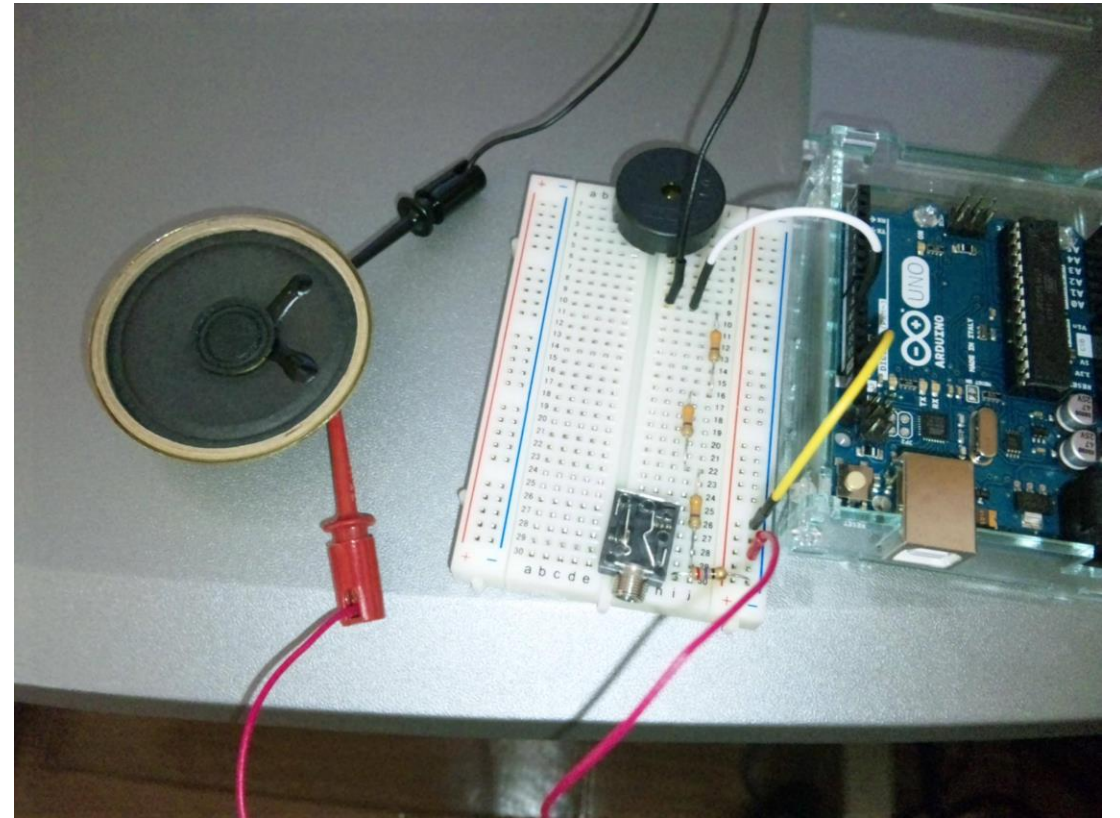
<https://risgk.github.io/>

2022年9月1日 SWEST24 EmbLT

第1世代

#1 Digital Synth VRA8 (2014年)

- Maker Faireを見学して電子工
作を始めたくなり、派遣先で
「ものづくりクラブ」に参加し
たことが開発のキッカケ
- ピコピコ音を出そうとしていた
が、**IIRフィルター**でシンセサイ
ザーを実現できることに気付く
- Arduinoだけで音を合成する、
Virtual Analog Synthesizer
- **VA-8**という名前を考えていた
が、名前衝突しそうなので**R**を
追加



#2 Digital Synth WRA32 (2014年)

- Web Audio APIを使った、ブラウザで動くシンセサイザー
- **VRA8**制御用のWebアプリ (MIDIコントローラー) を拡張したもの
- フィルター用のスライダーで音色を変えると、音が止まってしまう (VRA8も同様)
- もうメンテしていないが、今ならもっと良いものが作れる気がする

Digital Synth WRA32 5.1.1

Virtual Analog Synthesizer Web App

Synth Settings

MIDI IN
Basic Channel
Volume

We recommend Google Chrome, which implements Web MIDI API

Controllers

VCO 1 Waveform	<input type="range" value="0"/>	Mixer VCO 1 Level	<input type="range" value="64"/>
VCO 1 Coarse Tune	<input type="range" value="64"/>	Mixer VCO 2 Level	<input type="range" value="64"/>
VCO 2 Waveform	<input type="range" value="0"/>	Mixer VCO 3 Level	<input type="range" value="64"/>
VCO 2 Coarse Tune	<input type="range" value="64"/>	FEG Attack Time	<input type="range" value="21"/>
VCO 2 Fine Tune	<input type="range" value="70"/>	FEG Decay Time	<input type="range" value="85"/>
VCO 3 Waveform	<input type="range" value="2"/>	FEG Sustain Level	<input type="range" value="127"/>
VCO 3 Coarse Tune	<input type="range" value="52"/>	FEG Release Time	<input type="range" value="85"/>
VCO 3 Fine Tune	<input type="range" value="64"/>		
VCF Cutoff Frequency	<input type="range" value="91"/>		
VCF Resonance	<input type="range" value="51"/>		
VCF Envelope Amount	<input type="range" value="12"/>		
VCF Key Follow	<input type="range" value="0"/>		
AEG Attack Time	<input type="range" value="21"/>		
AEG Decay Time	<input type="range" value="85"/>		
AEG Sustain Level	<input type="range" value="127"/>		
AEG Release Time	<input type="range" value="85"/>		

Software Keyboard

C#3	D#3		F#3	G#3	A#3		C#4	D#4		F#4	G#4	A#4		
C3	D3	E3	F3	G3	A3	B3	C4	D4	E4	F4	G4	A4	B4	C5

Transpose

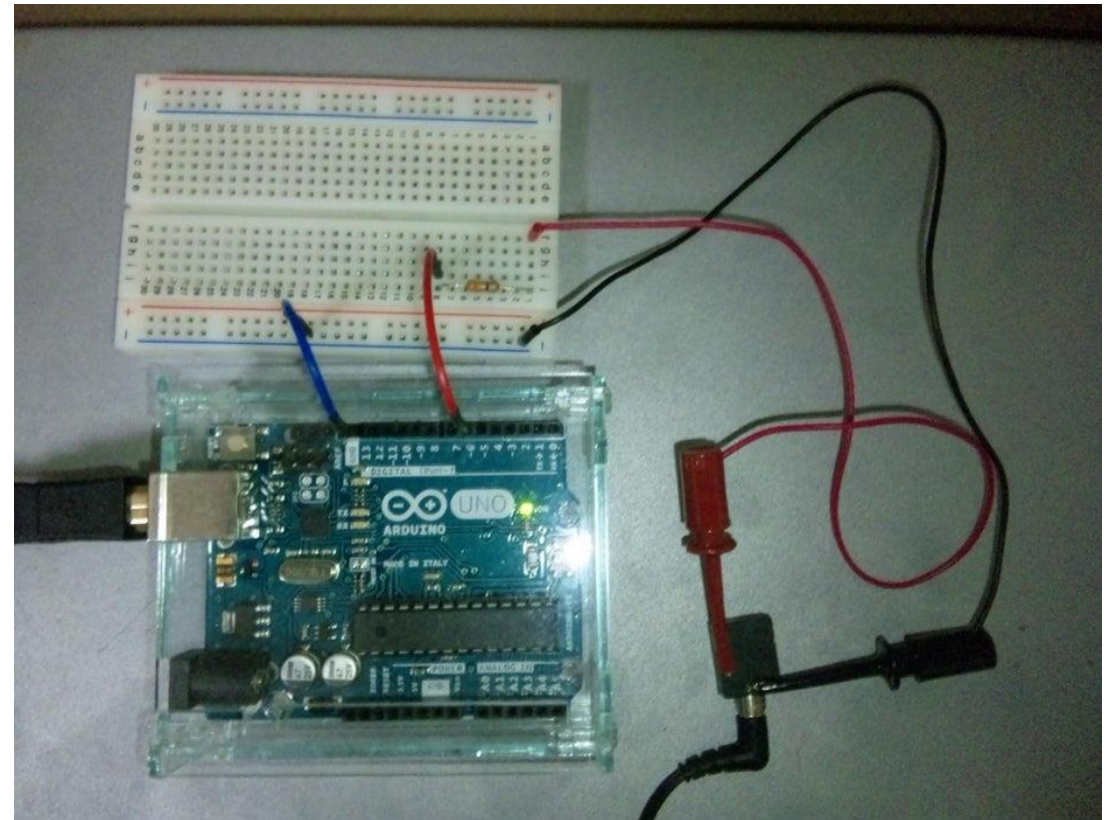
About

Made by ISGK Instruments ([View on GitHub](#))

第2世代

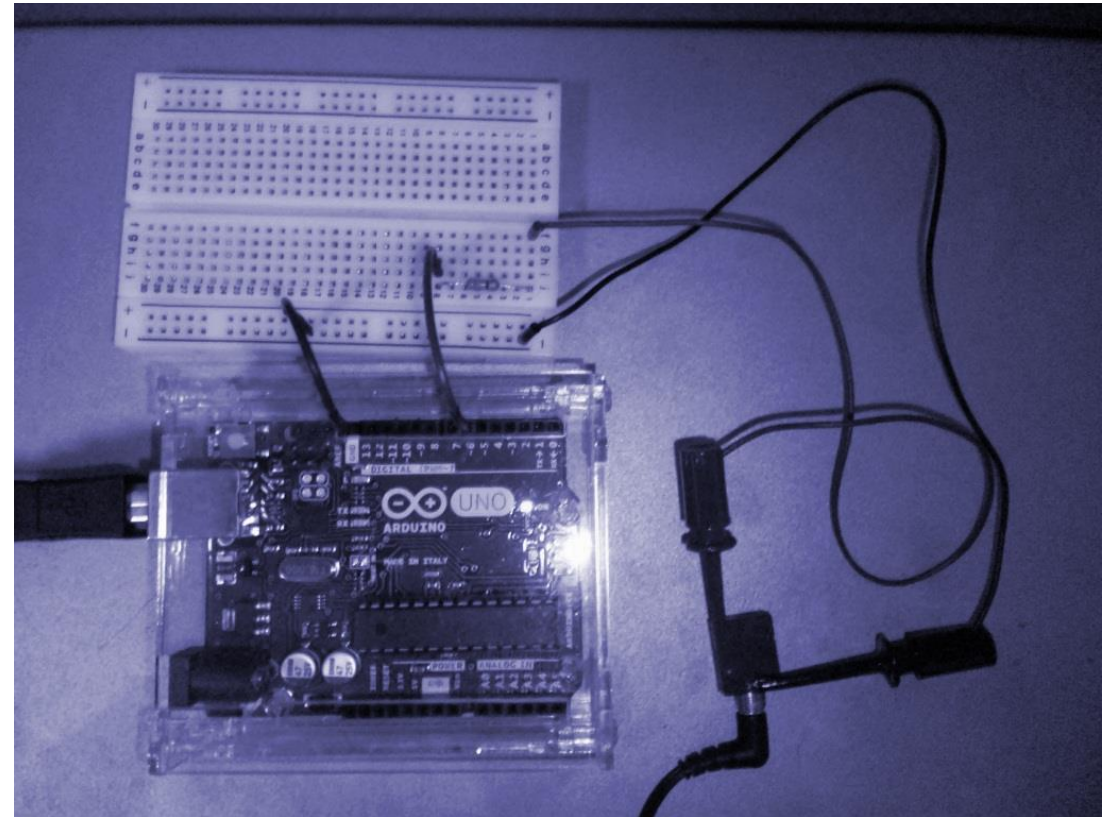
#3 Digital Synth VRA8-M (2015年)

- Monophonic SynthesizerのM
(ノコギリ波と逆ノコギリ波を
重ねたイメージも)
- 「無印」で関数がインライン展
開されていなかったことに気付
いたことが開発のキッカケ？
(Arduinoでは**always_inline**指
定が必要)
- 「ものづくりクラブ」メンバー
として、Maker Faireに初出展
- なかなか評判が良かったので、
ちょっと調子に乗る



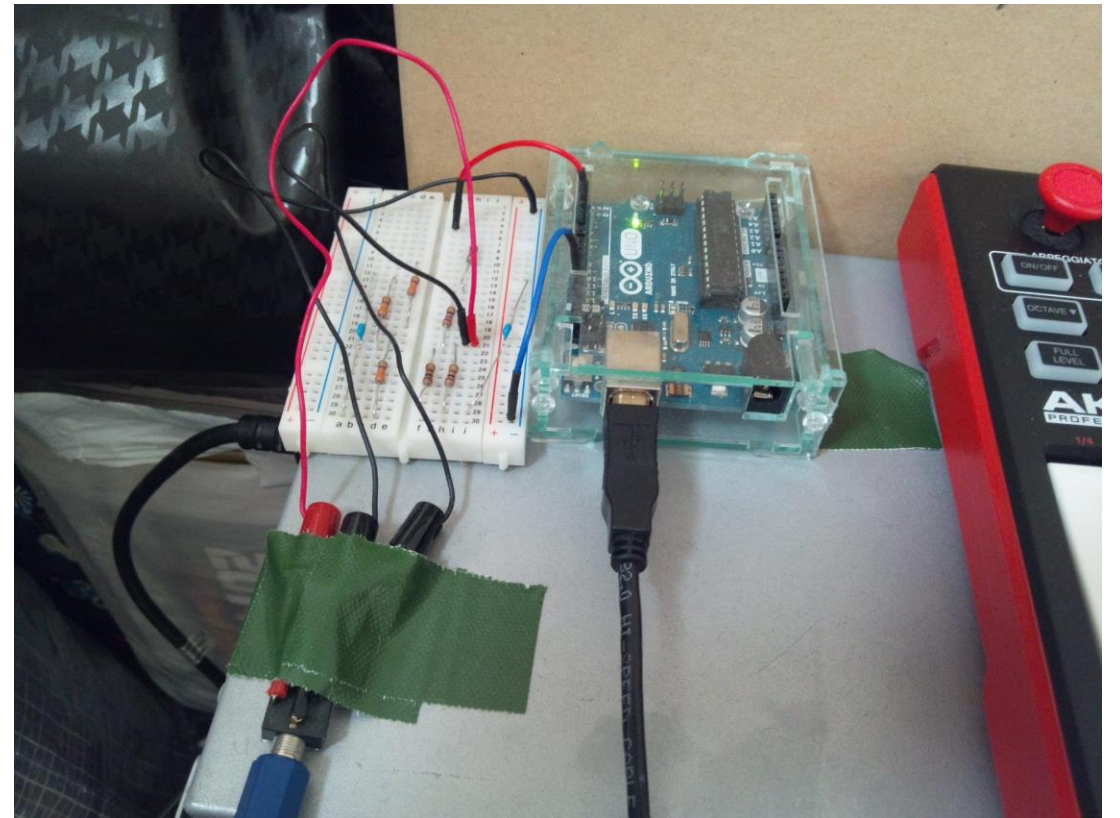
#4 Digital Synth VRA8-X (2015年)

- Experimental SynthesizerのX (シリーズの終わりを考えていたのかも?)
- 浜松市のイベントに間に合わせるように開発
- 色々な波形合成の「実験台」にしようと考えていたが、結局メンテせず
- ちょっと完成度が低いかも...
- シンボルカラーは「青」



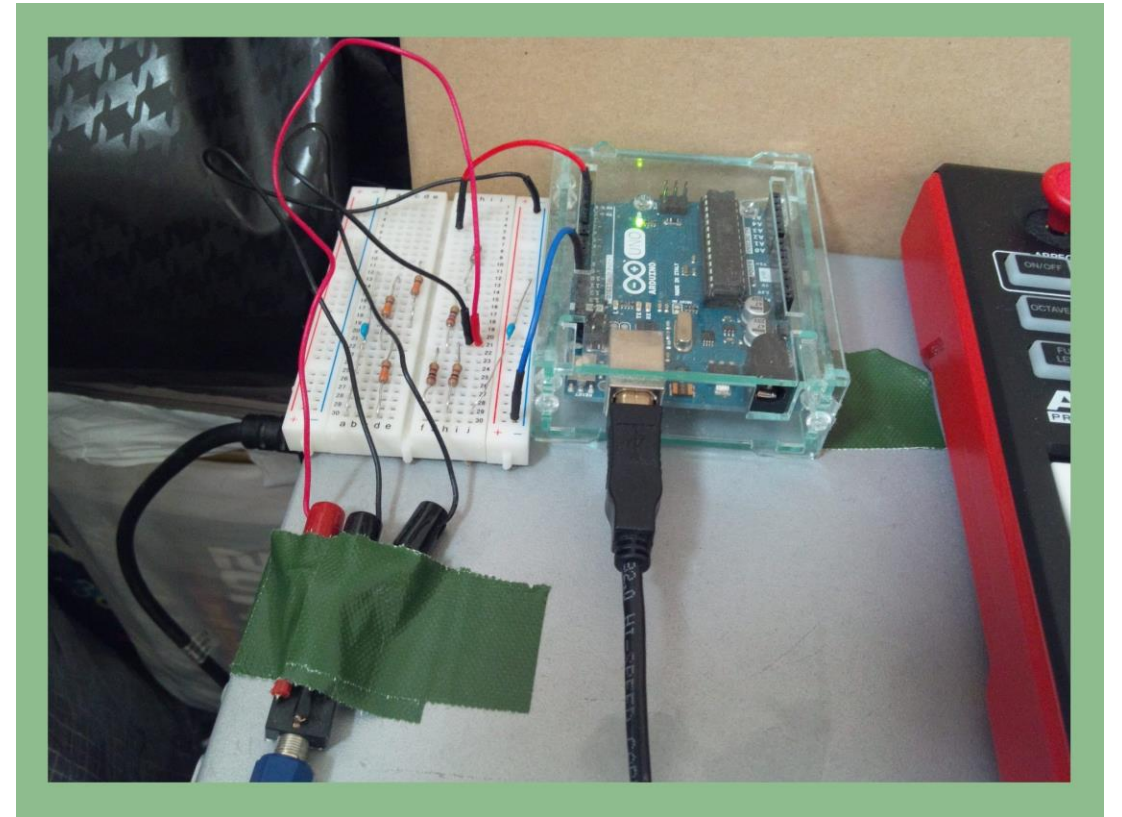
#5 Digital Synth VRA8-P (2016年)

- **P**seudo **P**olyphonic Synthesizerの**P**
- 3和音を出せるが、音質を変えるフィルターが1個なので「疑似」ポリフォニック（パラフォニックとも言う）
- 派遣先が変わったので「ものづくりクラブ」は卒業、この年からずっと個人サークルでMaker Faire Tokyoに出展
- SWEST自由工作発表（初回）



#6 Digital Synth VRA8-Px (2017年)

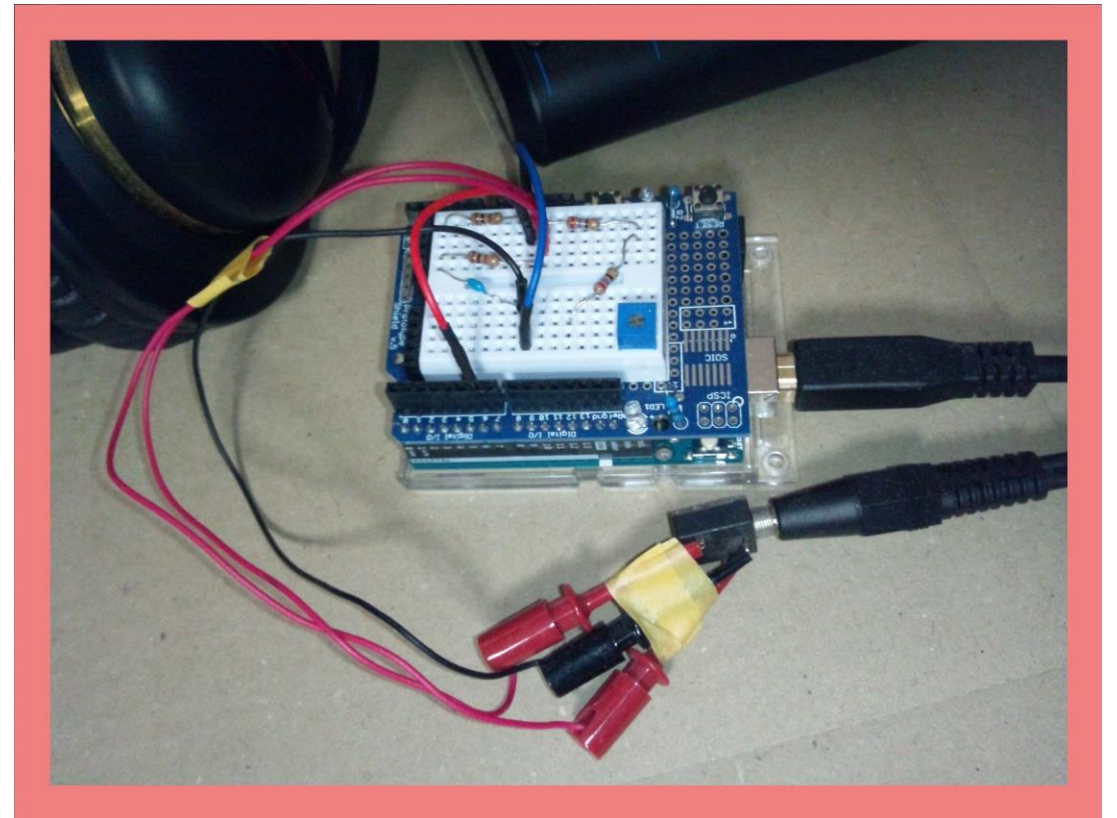
- Paraphonic SynthesizerのPにプラス (x)
- 会社のマイコン研修の素材にArduinoを使ってみたら、それまで**Timer0割り込みを無効にしていなかったことに気付く**
- **noInterrupts()** を呼んだら、CPU使用率が激減、かつ安定
- それまで**LED点灯でオーバーロード判定**していたが、この作品から**CPU使用率をMIDIデータでデバッグ出力** (計測重要!)



第3世代

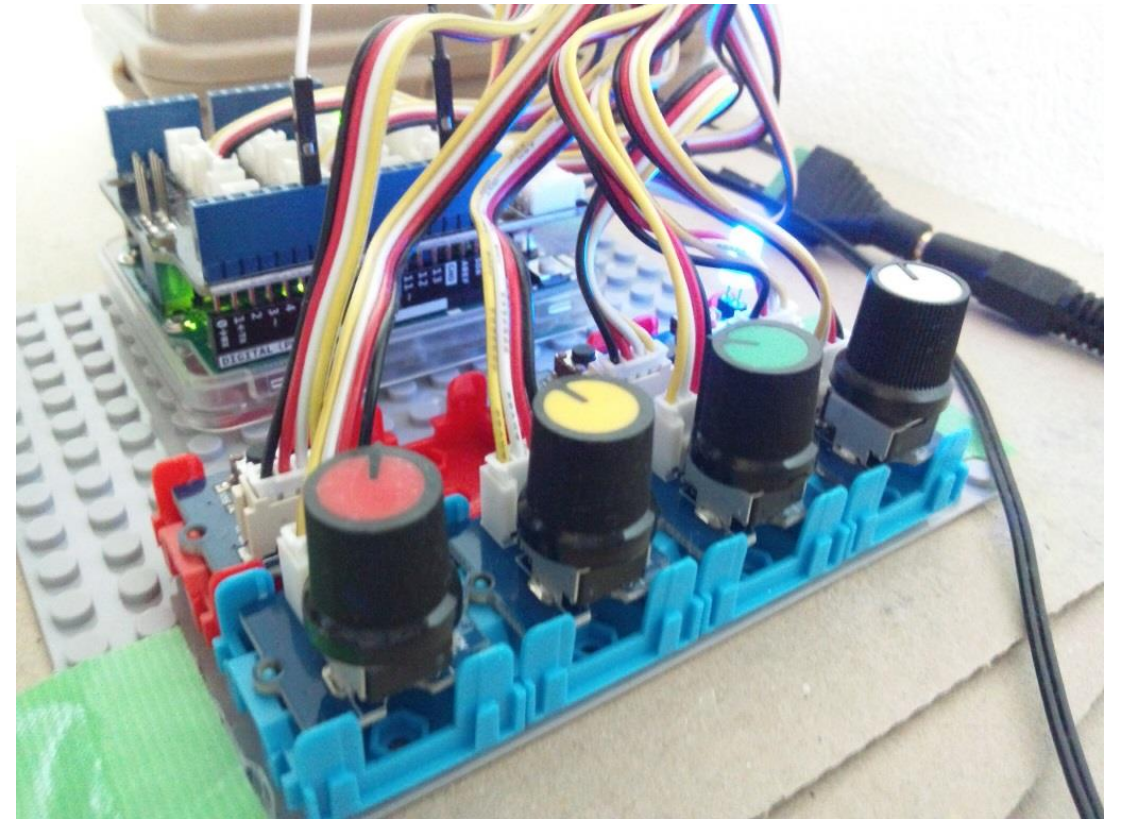
#7 Digital Synth VRA8-N (2018年)

- Monophonic SynthesizerのN
(Mの次とか、ノコギリ波のイメージもある)
- **サンプリング周波数**が15.625 kHzから**31.25 kHz**に倍増 (音質が大幅アップ!)
- もちろん、CPU使用率も倍増したので、頑張って「無駄」を省いて効率化
- SWEST20 インタラクティブセッション **ベストプロダクト賞**
ブロンズ



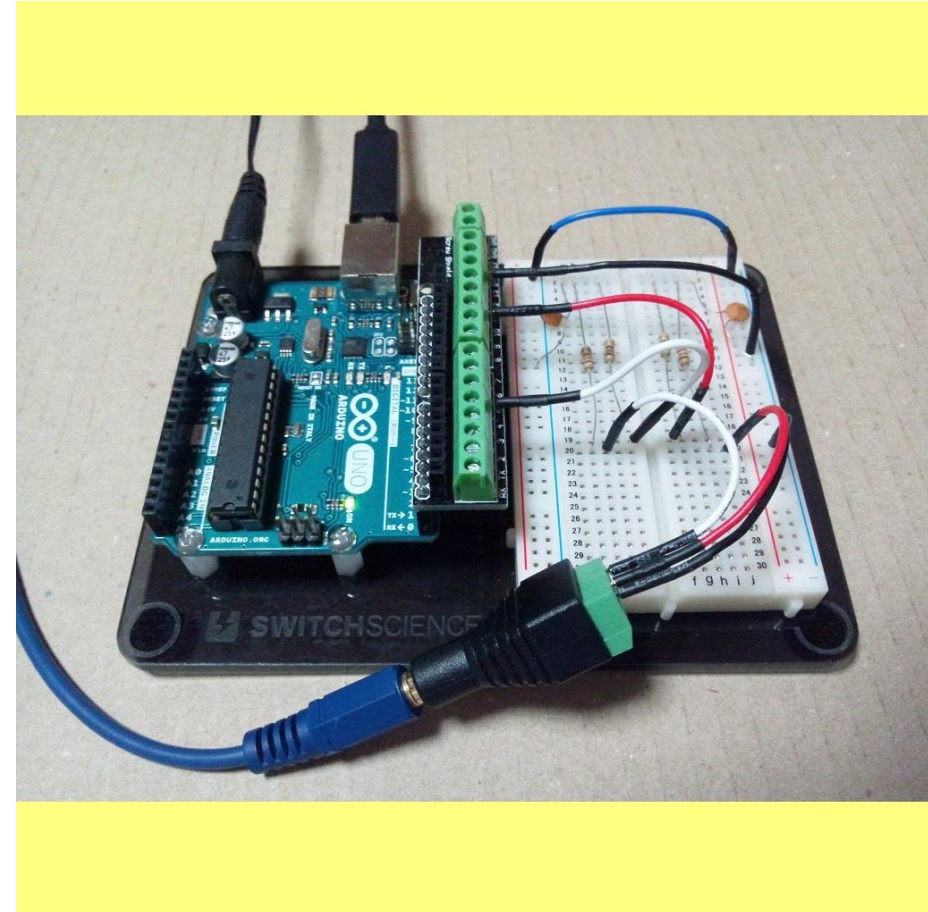
#7 (その2) VRA8-N mode-VC (2019年)

- **VRA8-N v2**を電圧制御 (Voltage Control) 可能にしたモードで、例えば可変抵抗で演奏可能
- モジュラー・シンセサイザーとも接続可能
- SWEST21 インタラクティブセッション **ベストプロダクト賞 シルバー**



#8 Digital Synth VRA8-Q (2020年)

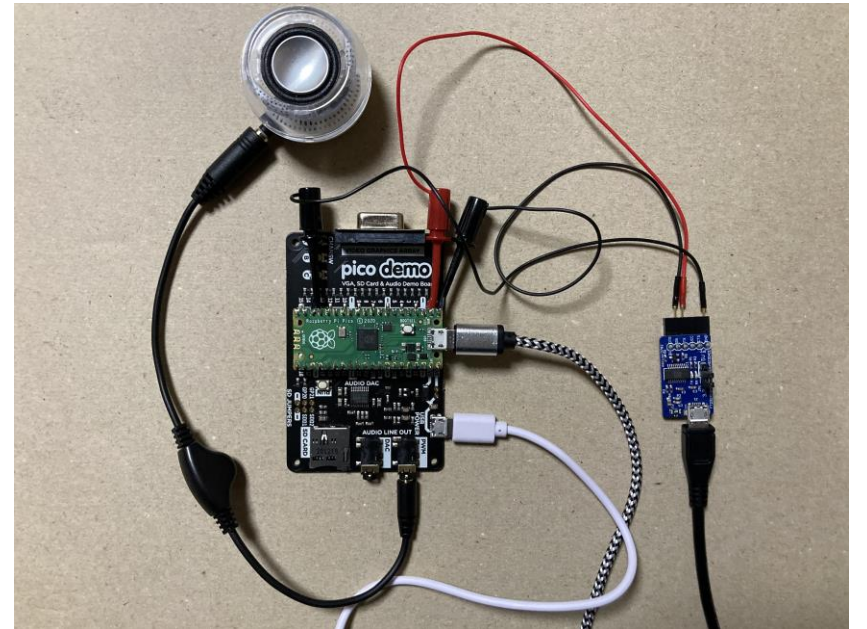
- **Quadraphonic SynthesizerのQ**
(Pの次のイメージもある)
- サンプルング周波数：31.25 kHz
- **4和音対応** (パラフォニック)
- **ステレオ・コーラス・エフェクター内蔵** (コーラス用のショート・ディレイ・バッファのサイズは512 Byte)
- SWEST22 インタラクティブセッションで発表



#9 pico_synth (2021年)

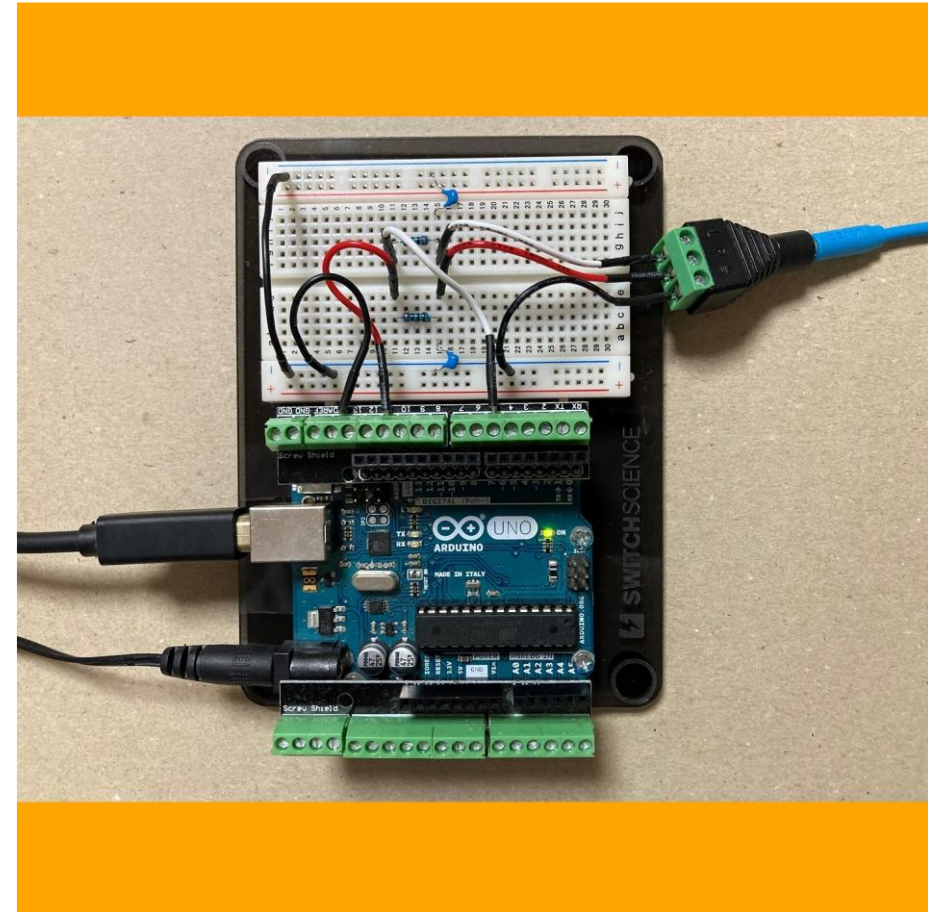
#9.1 pico_synth_ex (2021年)

- 『Interface 2021年8月号』のラズベリー・パイPico特集記事のために試作したシンセサイザと、それを拡張した4和音対応 (**ポリフォニック**) シンセサイザ
- 前者は、1オシレータ + 1フィルタ + 1アンプというシンプルな構成
- 後者は、SWEST23 インタラクティブセッション **ベストプロダクト賞 ゴールド** (しかし、v0.1止まり)
- ラズパイPicoは、今後本格的に使いたい



#10 Digital Synth VRA8-U (2022年)

- VRA8シリーズの集大成
- 単音シンセ「VRA8-N」と4和音シンセ「VRA8-Q」を合体させたような仕様
- Monophonic/Paraphonic **U**nited Synthesizer
- パルス波形では、パルス幅変調 (PWM) にも対応
- SWEST24 インタラクティブセッションで発表



To be continued...

Elixir/Nerves衛星を打ち上げる日

北九州市立大学 山崎 進

自己紹介

- 山崎 進 (ZACKY)
- 北九州市立大学 国際環境工学部 准教授
- 主な研究テーマ
 - Elixir, Phoenix, Nerves, Nxを使ったシステム実装・社会実装
 - Elixirベースの人工衛星画像処理システム
- 本日は福岡県北九州市内よりオンラインで発表しています



以下，山崎進の同タイトルのポジション
ペーパーを参照しながらご覧ください

話の前提

Elixir(エリクサー)とは？

並列プログラミング言語Elixir(エリクサー)

- 2012年にJosé Valim(ジョゼ・ヴァリム)によって開発されたプログラミング言語
 - 今年2022年は10周年！
- Joséは、以前はRuby on Railsのコミッターでした
- Joséの問題意識として、Rubyのかわりに、もっと高速に動作するプログラミング言語が必要であると考えていました
- いろいろ探究してErlang(アーラン)という関数型言語に出逢います
 - Erlangは故Joe Armstrong(ジョー・アームストロング)を中心にエリクソンで1986年から開発が進められていました
 - Erlangはエリクソンの事業の1つであった電話交換機向けに設計されており、電話交換機用途に耐えられるよう、堅牢かつ並行処理を追求していました
 - 電話交換機由来とは、TRON OSの1つのCTRONと近いモチベーションですね
 - ただし、Erlangの文法はとっつきにくいのが欠点でした
- そこで、JoséはRubyと同等以上のプログラムコードの書きやすさを追求し、Erlangをベースとして並行処理基盤を整備した新しいプログラミング言語 Elixir を開発します
 - Elixir は錬金術師が追い求めた不老不死の霊薬の名前です
 - Joséいわく命名理由は、Erlangと同じEで始まるカッコいい単語という以上のことは覚えていないそうです
 - 一時期言われていた、Joséがファイナルファンタジーに興じていて命名したという説は、José自身によって否定されました
- Elixirはパイプライン演算子|>が特徴的で、これを使って書いたプログラムコードは、左から右へ、あるいは上から下へ、データを変換していく様を流れるように読み下すことができます

```
1..1_000_000 # 1から1,000,000までのリストを  
|> Enum.map(& &1 * 2) # 各要素2倍して  
|> Enum.map(& &1 + 1) # さらに各要素に1足したリストを返す
```

- Elixirは、このようなコードを簡単に並列化することができます



© Elixir Team

Nerves(ナーヴス)とは？

使いやすいIoT基盤Nerves(ナーヴス)

- 2015年からNerves Project developersによって開発が進められてきた, ElixirベースのIoT基盤
- Tombo Worksの資料参照
<https://swest.toppers.jp/SWEST24/proceedings/interactive/SWEST24-IS-004.pdf>



Copyright (C) 2015- by the Nerves Project developers

とこるで

商用の人工衛星にmrubyが
載ったそうですね

商用の人工衛星にmrubyが載ったそうですね

02 15 2021 [EVENT](#)

2021年2月26日（金） 「Ruby Business Users Conference 2021 Online」にて弊社CEO大西俊輔が講演いたします

- Ruby Business Users Conference 2021 Online
 - 2021年2月26日(金) 15:35～16:15
 - 小型レーダー衛星開発におけるmrubyの適用
 - QPS研究所CEO 大西俊輔



<https://i-qps.net/news/438>

それに対抗するわけでも
ないですが

Elixir/Nervesを搭載した
人工衛星を打ち上げたいです

JAXAの

革新的衛星実証プログラム

KOSEN-1

Elixir/Nerves搭載人工衛星をJAXAの革新的
衛星技術実証プログラムで打ち上げる！

どんな衛星を
打ち上げる？

Elixirといえば分散並列処理
最新の機械学習基盤Nxも搭載して
データ処理衛星を実現

データ処理衛星

QPS研究所はJAXAと組んでSAR衛星からデータ処理衛星と衛星間通信を行なって宇宙空間上で必要なデータ処理をしてから地上に処理済みのデータを送るという構想を発表

SAR衛星

Nx

Elixirといえば分散並列処理
最新の機械学習基盤Nxも搭載して
データ処理衛星を実現

基礎研究をするための研究費を確保済
目下、研究を進めているところ

ここからEmbLTだけの内容です
ここだけの話ですので、配慮願います

人工衛星に載せる上で 重要な点

消費電力

コンピュータだけで10W以内に抑える必要

ボードは何を使うか？

Toradex

<https://developer.toradex.com/linux-bsp/how-to/hardware-related/verdin-imx8m-plus-power-consumption/>

計算のアクセラレーションを
どう確保するか？

DSP?

https://www.ti.com/lit/wp/spry277/spry277.pdf?ts=1662018371178&ref_url=https%253A%252F%252Fwww.google.com%252F

FPGA

他に人工衛星で気を付ける点 放射線耐性

ソフトウェアでの放射線対策

Elixir/Nervesの特性

基調講演を拝聴して
思ったこと



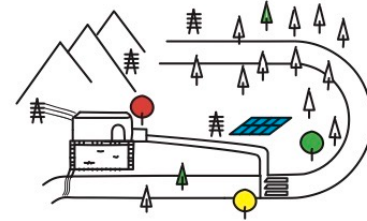
FA をびよんどする

～ Elixir/Nerves 小水力発電所を運転開始する日 ～



SWEST24

kikuyuta とは



高知工科大学
KOCHI UNIVERSITY OF TECHNOLOGY

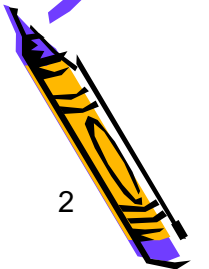


@kikuyuta



2022.09.01

菊池 豊 @高知工科大学



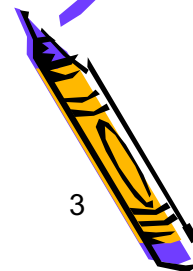
謝辞

- Special thanks to kochi.ex
 - @nishiuchikazuma, @32hiro, @myasu
- 国立研究開発法人情報通信研究機構
 - 【シーズ創出型プログラム】委託研究(04001)
 - 関数型パラダイムで実現する B5G 時代の資源透過型広域分散コンピューティング環境
- 高知県
 - 令和2年度高知県産学官連携産業創出研究推進事業
 - 高信頼性・高並行性を有するElixir言語を応用した工業制御用のIoTノードとクラウド連携技術の研究開発



2022.09.01

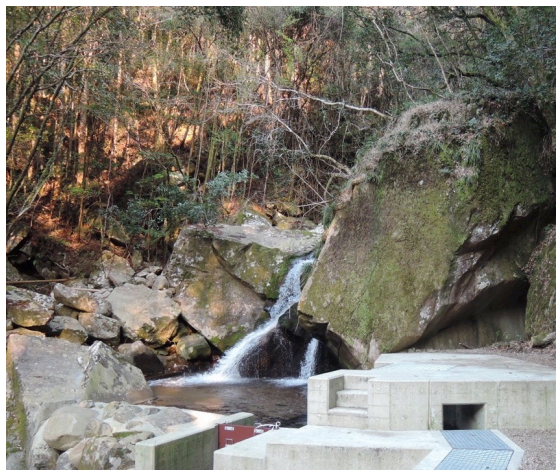
菊池 豊 @高知工科大学



馬路村小水力発電所 (2016.03.30運転開始)



- 発電所制御はPLC
- データ打上げと web 用にラズパイ2B x3 & Linux & Python



2022.09.01

菊池 豊 @高知工科大学

三菱電機のホームページより₄



小水力発電システム(高圧)



水車

発電機

盤

トランス

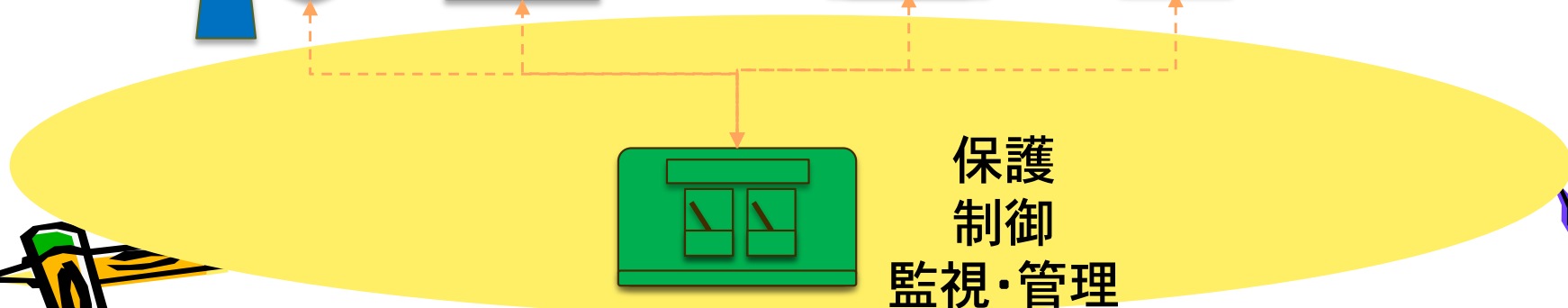
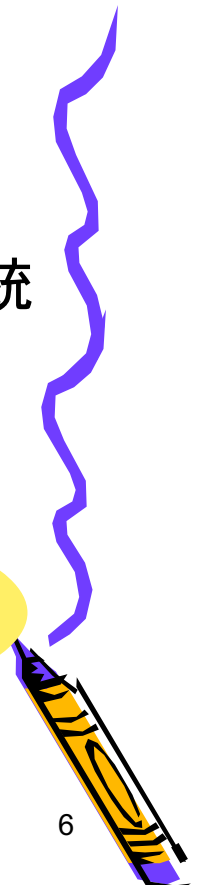
電力系統

保護
制御
監視・管理

2022.01.21

菊池豊 高知工科大学 & kochi.ex

6



小水力発電システムの機能

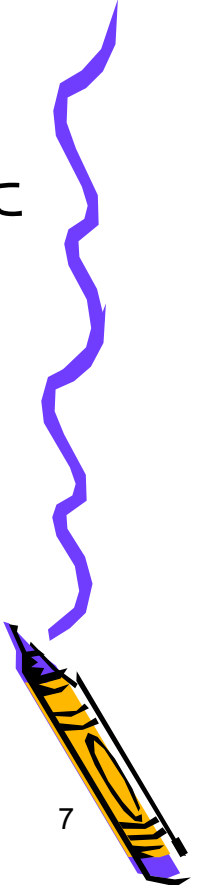


- 保護
 - 異常を検出したら安定した停止状態に遷移すること
 - 設備が壊れたり人が怪我をしたり、異常が電力系統に波及しないように
- 効率の最大化
 - 発生させるエネルギーが最も大きくなるように
 - いいかえると最もお金が儲かるように
- 管理・監視
 - 発電所の状態の把握、記録、遠隔での把握・操作

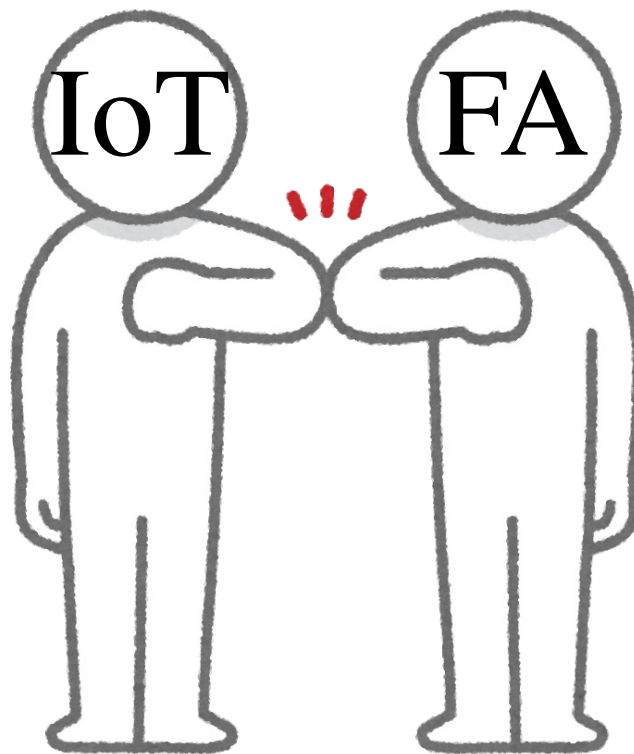


2022.01.21

菊池豊 高知工科大学 & kochi.ex

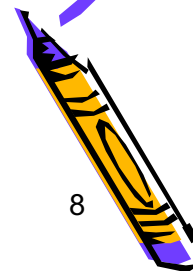


IoT 技術を FA に持ち込んだらいいんじゃないよね？



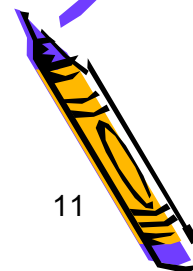
2022.09.01

菊池 豊 @高知工科大学



IoTをFAに应用する上での課題と解決

- IoT で使うようなマシンなんてFA じゃ使えねえ
 - FA でも使えるマシンを作ろう → 作った
- FA で使うプログラムを現代的な言語で作れるのか
 - 小規模な小水力発電所用ソースコード起こそう → 作った
- どこかで実際に FA ビジネスで使わないと説得力ない
 - 小水力発電所で使おう → 使った、金も動いた



2022.09.01

菊池 豊 @高知工科大学

11

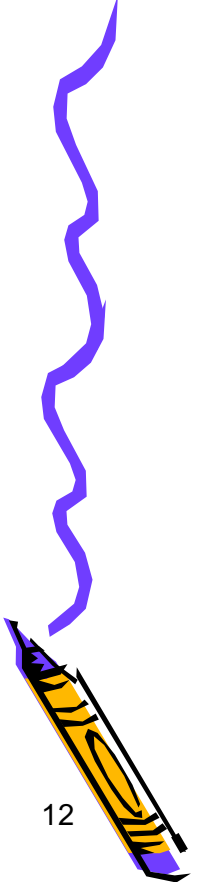
言語は Elixir を採用することに

- 並行・並列プログラミング言語
 - アクターモデルによる並行性(プロセスとメッセージ)
- プロセス(OSのプロセスとは異なる軽量スレッド)
 - 1CPUで数十万プロセスも可能
 - 1プロセスあたり 3kB程度～ のメモリ使用
- 耐障害性
 - アプリ停止なしの障害時プロセス再起動 (Let it Crash!)
 - プロセスのホットスワップ可能
- 元々は電話交換システム用(1986～)
 - にできた Erlang の上に乗ってる



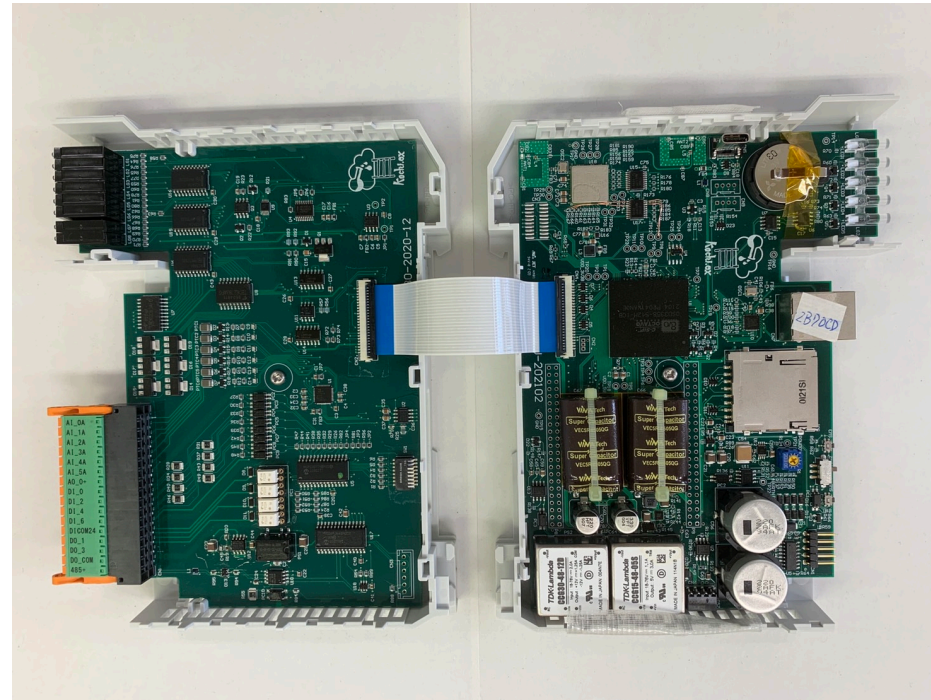
2022.09.01

菊池 豊 @高知工科大学



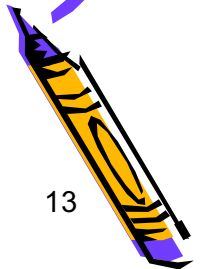
12

Elixir マシン ExiBee 爆誕!



2022.09.01

菊池 豊 @高知工科大学



13

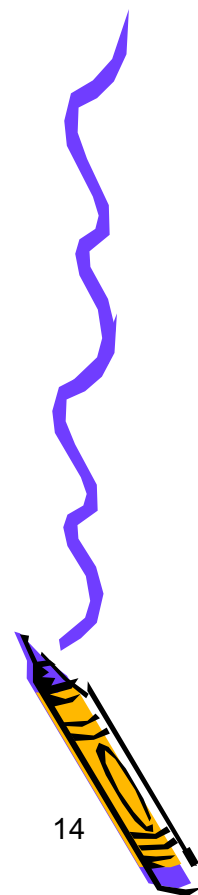
基本的な考え方

- Nerves (Elixir の IoT プラットフォーム) が簡単にいごくように
 - あまり特別なハードウェアだと buildroot が大変
- 商用のPLCに負けない I/O 点数
- 取り扱いの楽な電源周り
- オシャレな外見
- 仕様はオープンに
 - BeagleBone Green (Open Source Hardware SBC) ベース
 - <https://github.com/kikuyuta/exibee/blob/master/concept-jp.md>



2022.09.01

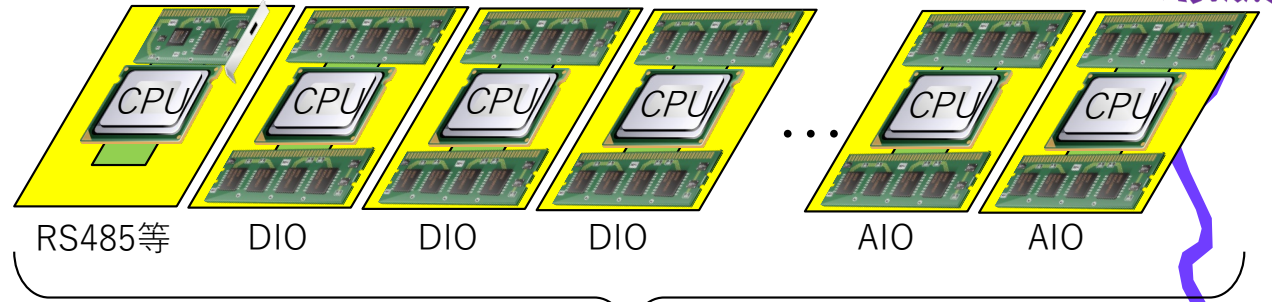
菊池 豊 @高知工科大学



14

全部のモジュールにCPU

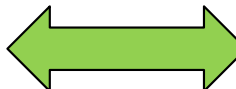
- ハードウェア
- ソフトウェア
- クラウド連携



全てのノードが Elixir で動作

 elixir




クラウドと連携



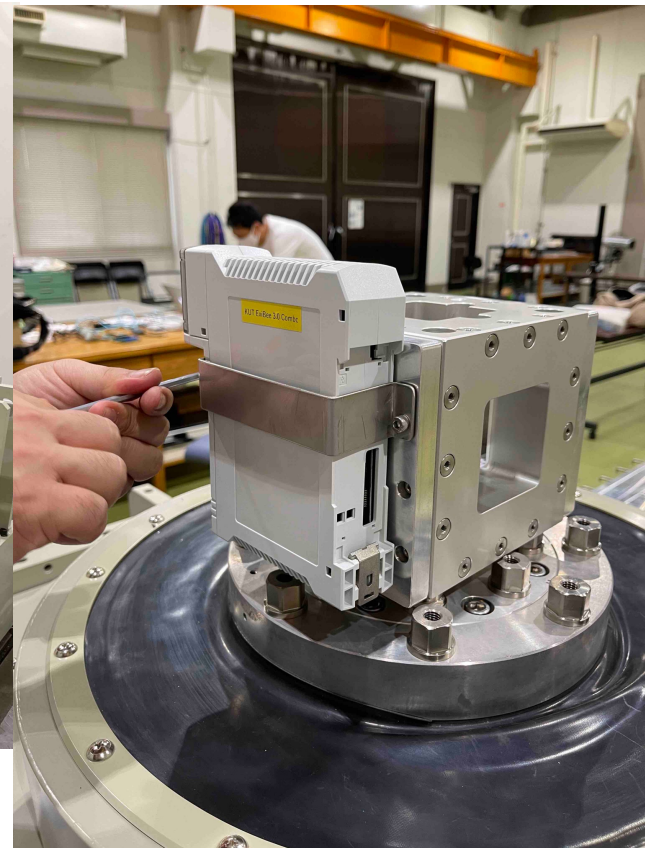
2022.09.01

菊池 豊 @高知工科大学

16



JIS B3502 振動・衝撃試験風景



2022.09.01

菊池 豊 @高知工科大学

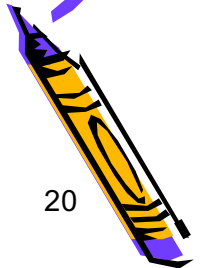


JIS B3502 温度湿度環境試験風景



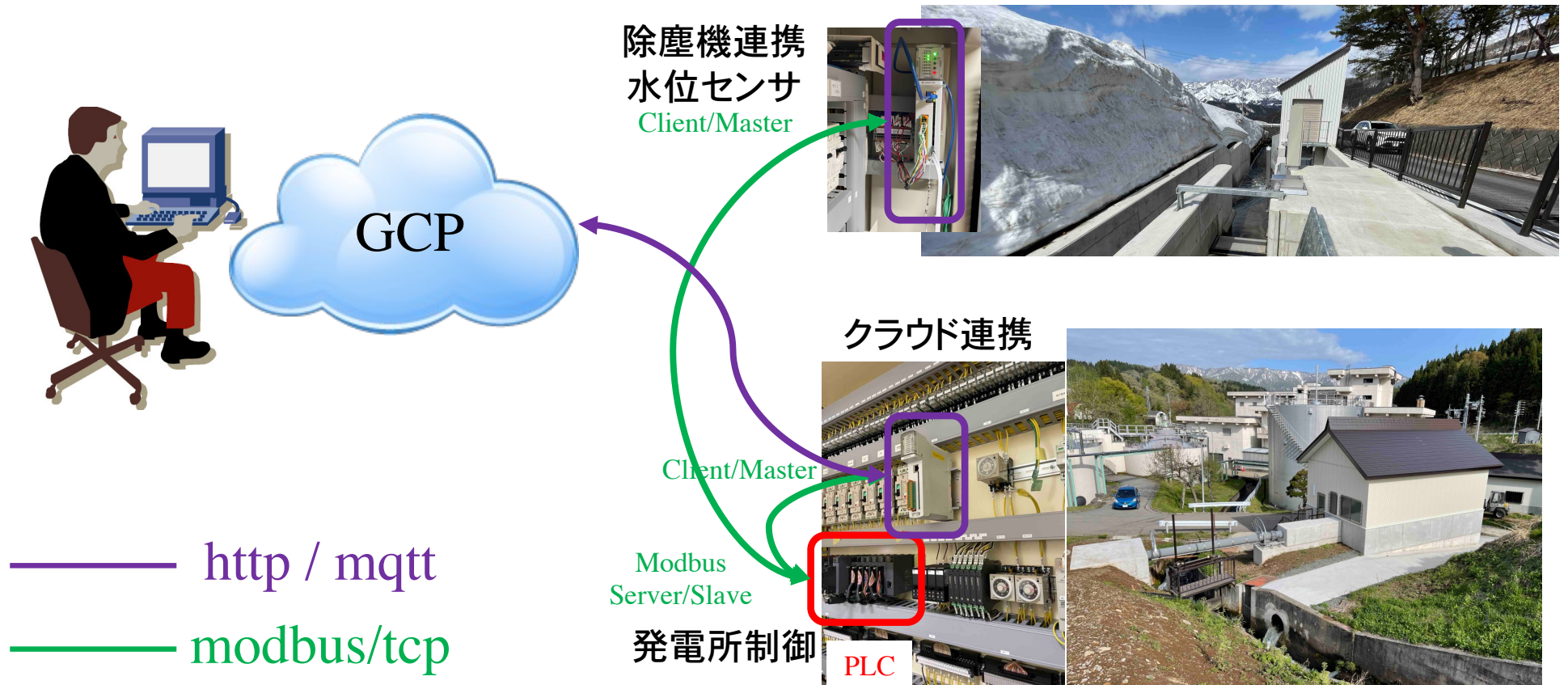
2022.09.01

菊池 豊 @高知工科大学



20

小水力発電所 システム構成



2022.09.01

菊池 豊 @高知工科大学

21

まくなえ川小水力発電所

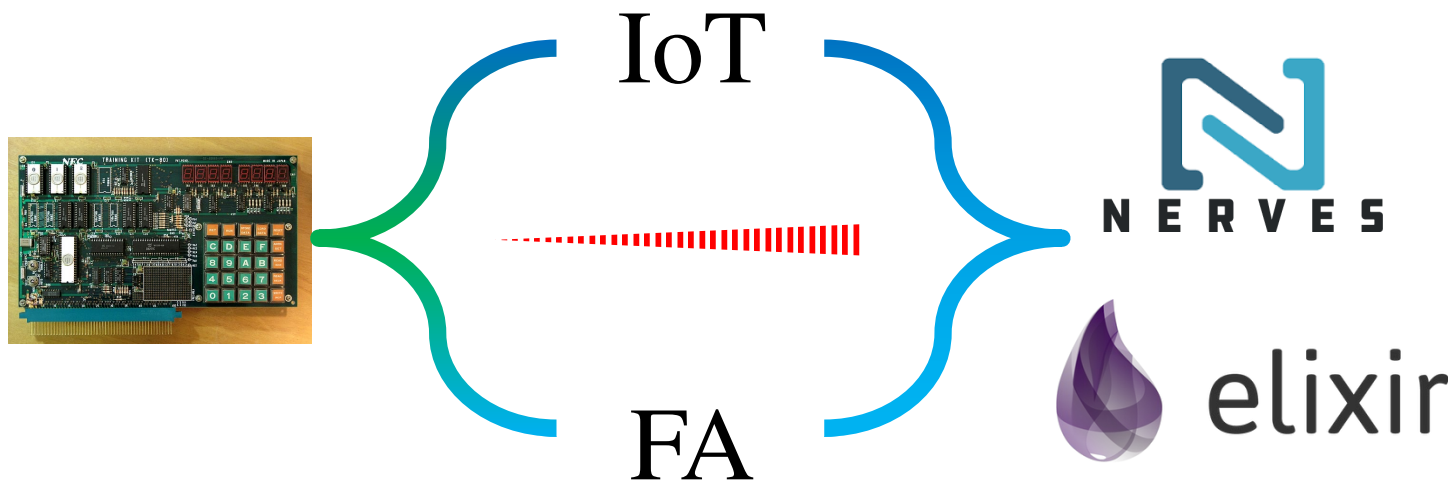


2022.09.01

菊池 豊 @高知工科大学

22

Nerves / Elixir が融合を可能にするのでは

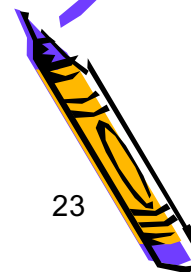


<https://ja.wikipedia.org/wiki/TK-80>



2022.09.01

菊池 豊 @高知工科大学



23

色々やってきたが次は何をやるか



RPi3B
Python
Linux

2016



BunkaPLC 1
Python
Linux
GMO

2018



BunkaPLC 2
(ExiArmadillo)

2020



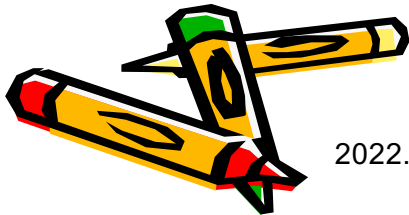
ExiBee
Elixir
Nerves (buildroot)
GCP/さくらinet

2022

?

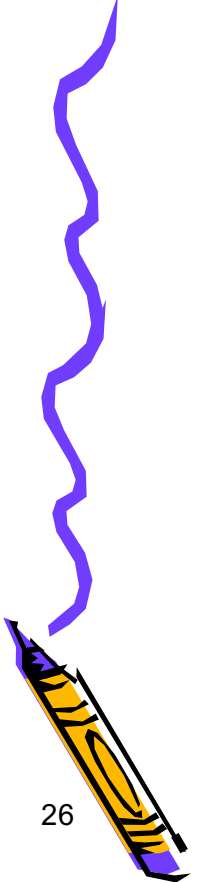
足りないものは何か

- パーツのモジュラリティ
 - 仕様の共通化、通信IFの共通化、無線化
 - 1パーツ 1ノード
 - BOM に API が入るような産業構造
- 分散環境での構築フレームワーク
 - 5Gが出てきて網に計算資源を置けるようになった
- 産業用途 (FA) におけるオープンなコミュニティ



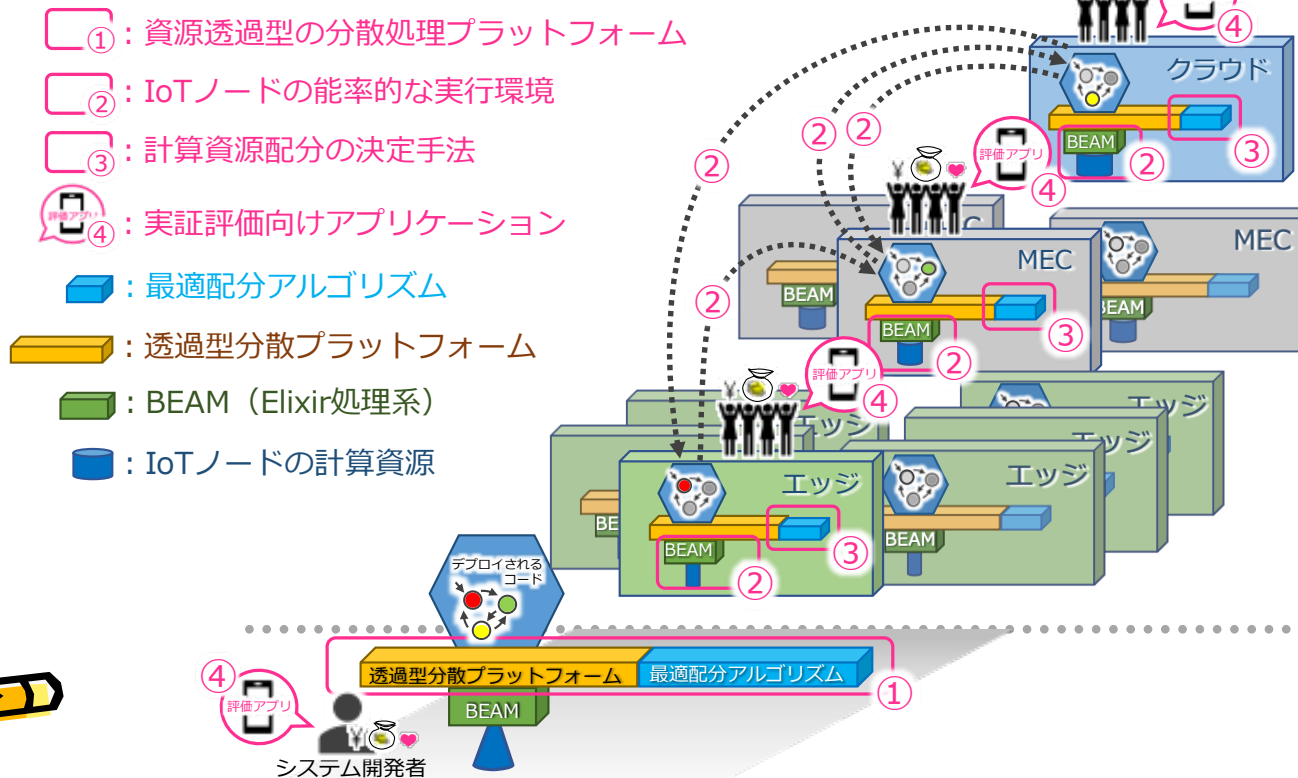
2022.09.01

菊池 豊 @高知工科大学



26

これからは... セッションs4a 5G & Beyond ~その先へ~



2022.09.01

菊池 豊 @高知工科大学

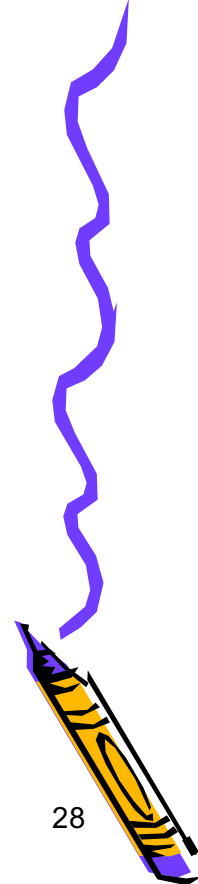


27

Nerves/Elixir で FA を

We believe in rough consensus
and blinking LEDs

論よりLチカ



2022.09.01

菊池 豊 @高知工科大学

28

本発表ならびに本セッションの内容は
国立研究開発法人情報通信研究機構の
委託研究(04001)により得られるものです

ギャラクシアンをIoT化する

SWEST24 embLT

<https://www.sakura.ad.jp/>

DAY

2022/09/01

COMPANY

さくらインターネット株式会社

DEPARTMENT

さくらインターネット研究所

NAME

上級研究員 菊地俊介

菊地 俊介 (東京都出身)



@kikuzokikuzo

<https://note.mu/kikuzokikuzo>

<https://www.facebook.com/kikuzokikuzo>

所属 さくらインターネット研究所

学歴 早稲田大学大学院 理工学研究科 電子・情報通信学専攻 修士課程修了
早稲田大学大学院 国際情報通信研究科 博士課程単位取得退学

職歴 富士通（株）富士通研究所に就職

ネットの研究やったり、SEやったり、NICTに出向したり、
トイレIoT作ったり

さくらインターネットに転職

データ流通（FIWARE, NGSI）、OpenFogコンソーシアム（標準化）、
量子（アニーリング）コンピュータ、AR/VR、RISC-V、Erlang/Elixir

専門 エッジ・Fogコンピューティング（分散系システムのあたり）
ビジョナリーとして技術・社会、会社の将来を思い描く
新規領域調査、PoC実施、社内適用コンサル、講師・講演

趣味 新技術調査、読書、最近はガンブラ作り

- Arcade1up!という、
1/2サイズのゲーム筐体
があります
- ギャラクシアン世代？
のうちの所長が
大フィーバー→購入



- <https://research.sakura.ad.jp/2019/02/12/galaxian/>

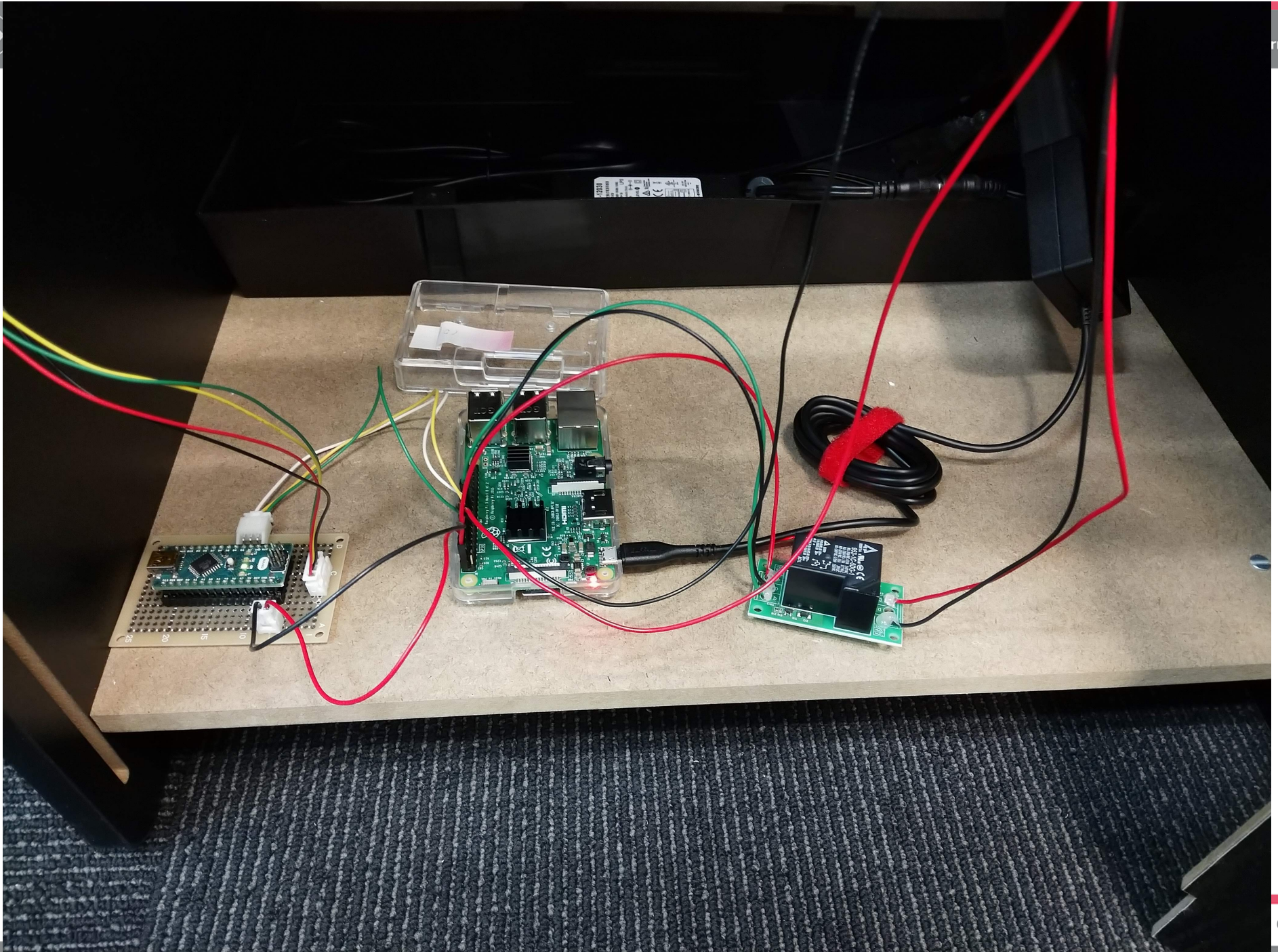


- 所長：「これ受付（会社の）に置こうか！」
- 所長：「受付に置くとなると、何かあったときにメンテナンスできるような体制を作らなくちゃならないんだよね」
- 私：「そうですね。サービスマンコールボタンみたいな要りますよね。それ押すとSlackに通知出るみたいな。」
- 私：「せっかく連携する仕組みを作るんなら、稼働状況も把握したいですね。」

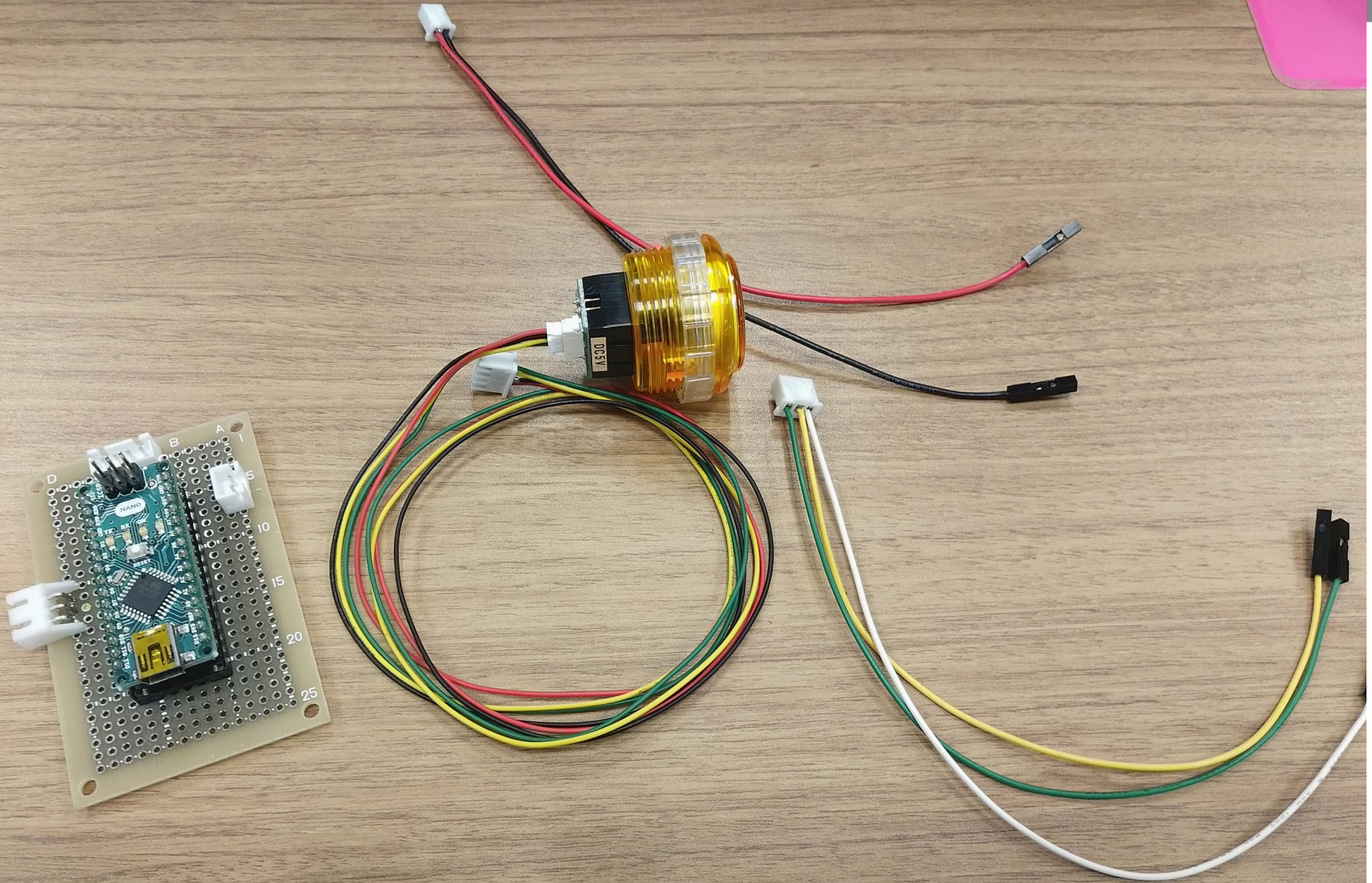
会話は現実に即してはいますがフィクションです

- LTE/5G接続すれば、どこにでも持っていける！
 - のですが...
- 実際の実装は
 - ラズパイ + Arduino
 - いろいろな事情でWiFiで。
- WebAPIを実装
- Slackへの通知、Slack経由での制御などを実現













arcade1up ▾  2

+ 関連ページを追加する

エマージェンシーボタンが押されました。
応答 or クリア 2019年6月27日 ▾

エマージェンシーボタンが押されました。
応答 or クリア

 s-kikuchi.sakura 🌸 11:19
エマージェンシーボタンにもローパスフィルタ入れて、誤検出対応してみた。

 arcade1up_watcherbot アプリ 11:33
エマージェンシーボタンが押されました。
応答 or クリア

2019年6月28日 ▾

 arcade1up_watcherbot アプリ 17:39
電源投入されました。
電源断されました。

2019年10月1日 ▾

 arcade1up_watcherbot アプリ 15:20
電源投入されました。
エマージェンシーボタンが押されました。
応答 or クリア
電源断されました。

2020年1月27日 ▾

 arcade1up_watcherbot アプリ 10:44
エマージェンシーボタンが押されました。
応答 or クリア

2020年3月17日 ▾

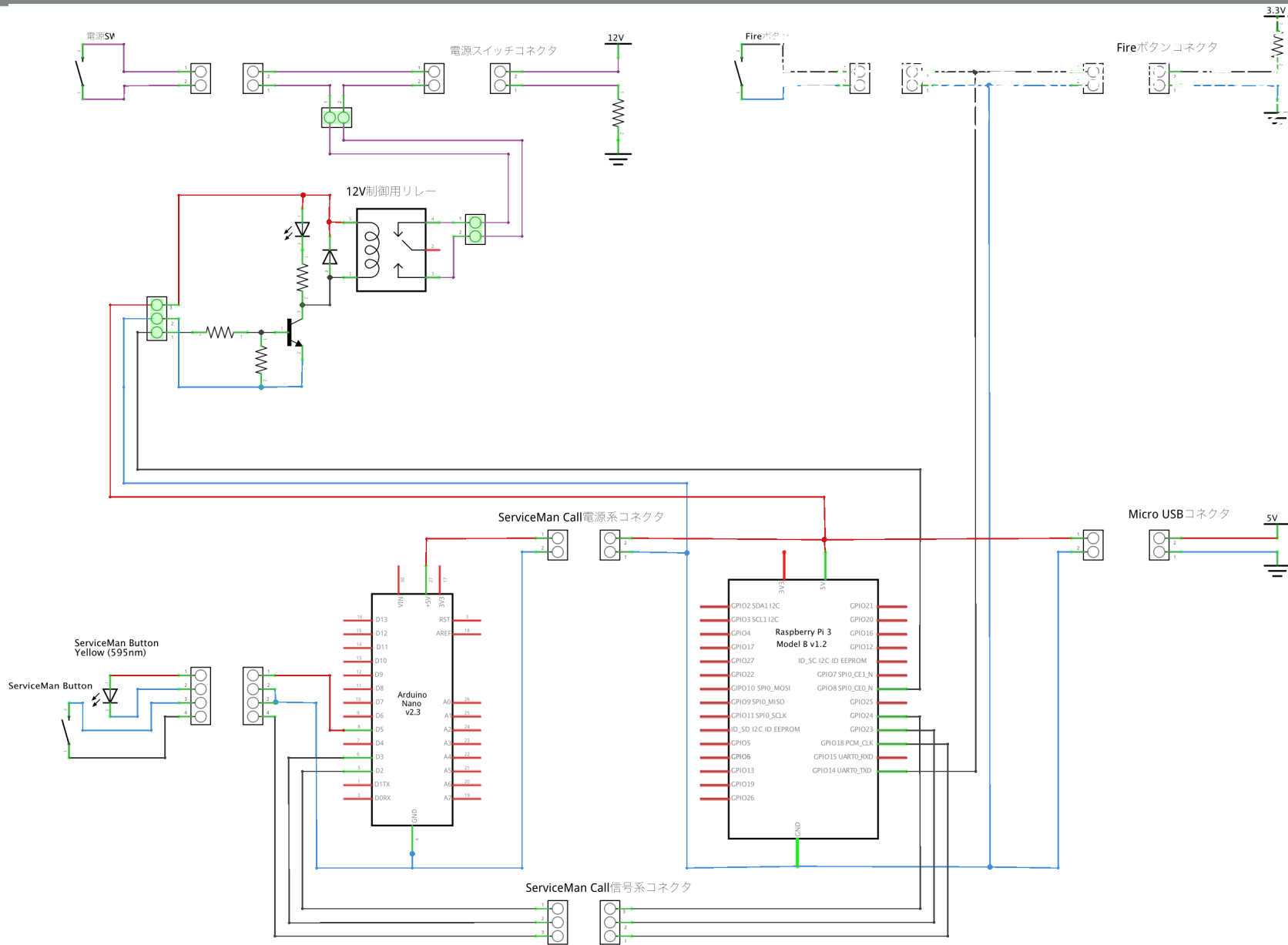
 arcade1up_watcherbot アプリ 08:57
エマージェンシーボタンが押されました。
応答 or クリア

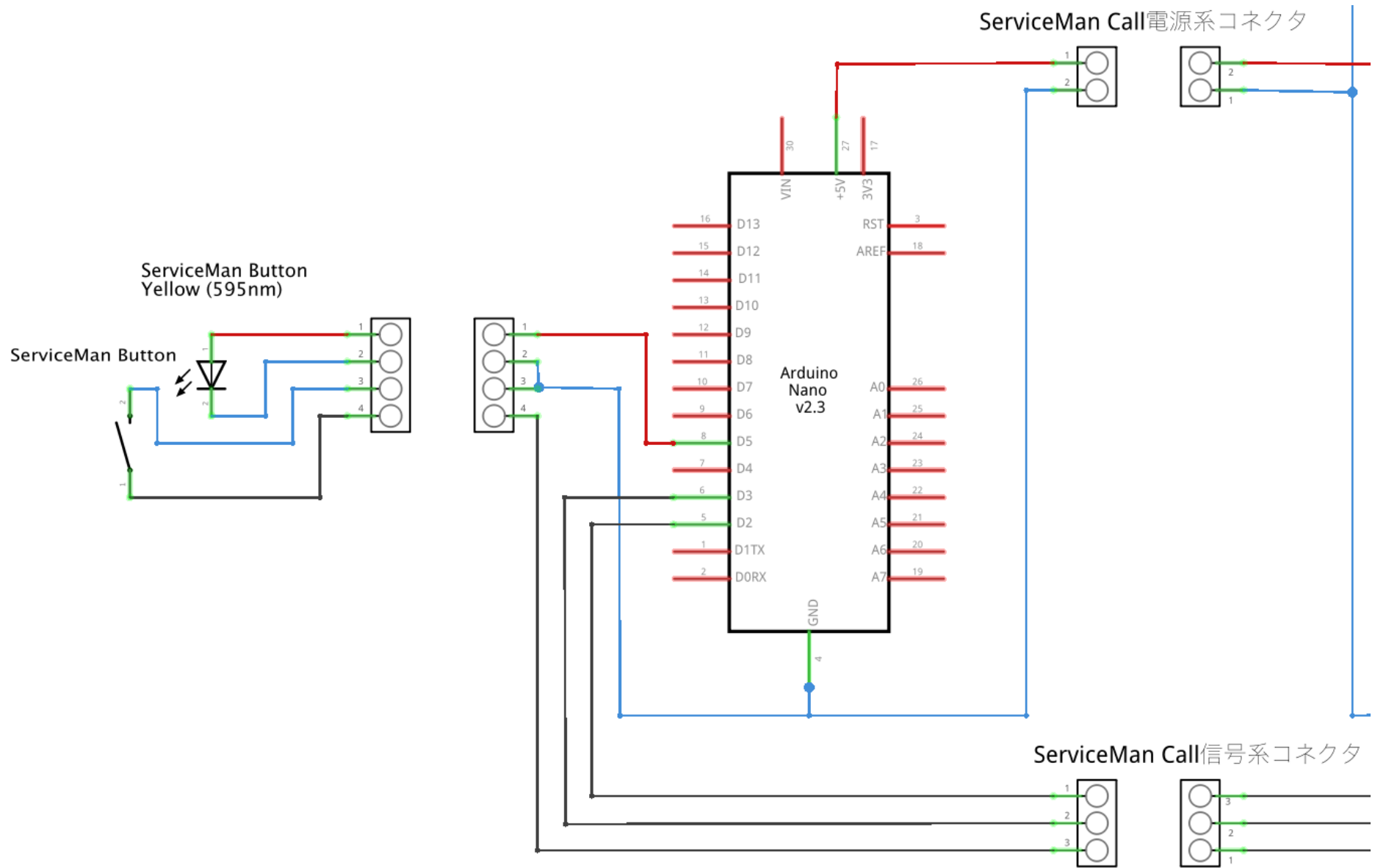
#arcade1up へのメッセージ

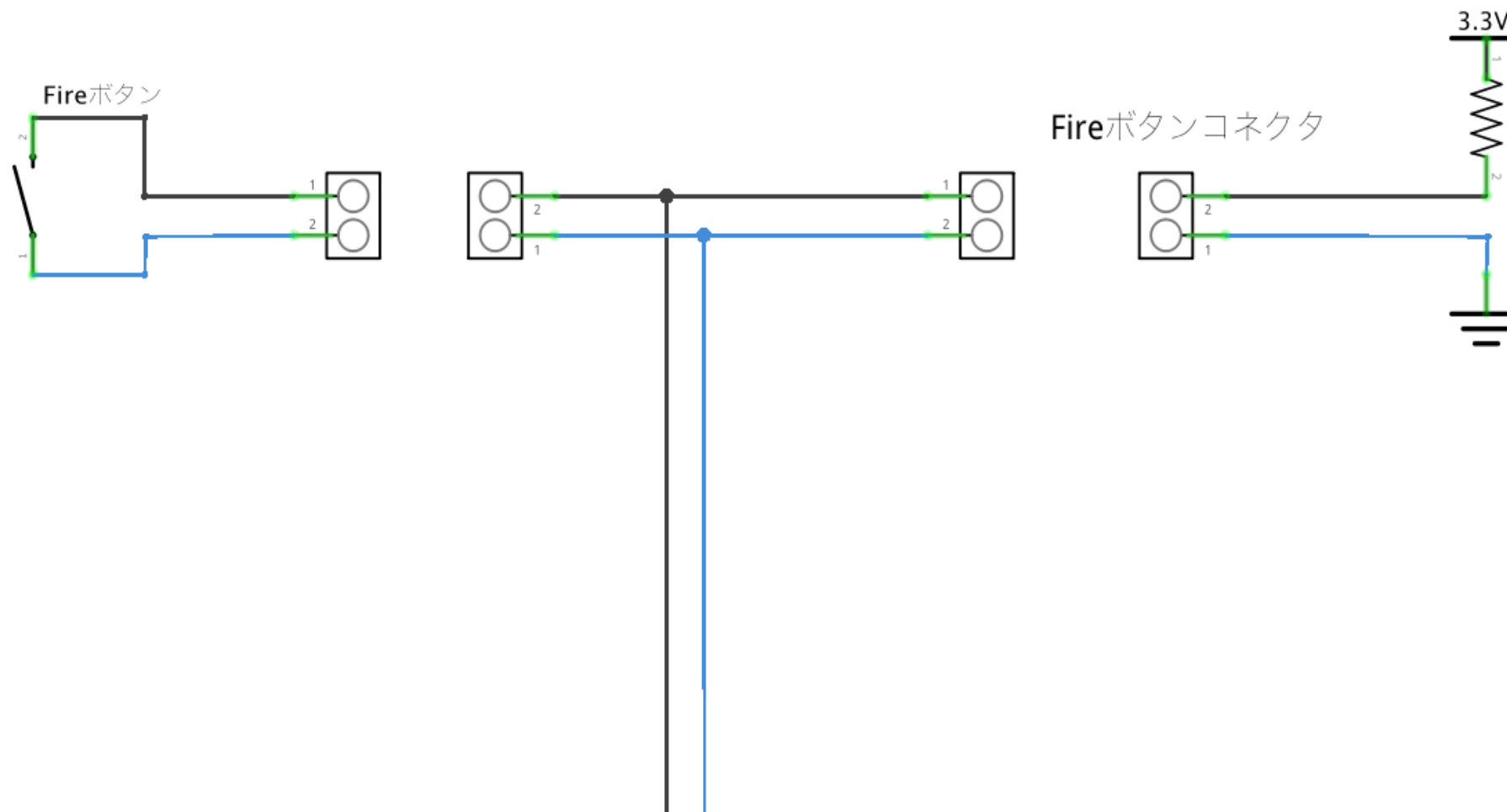
      ▶ ▾

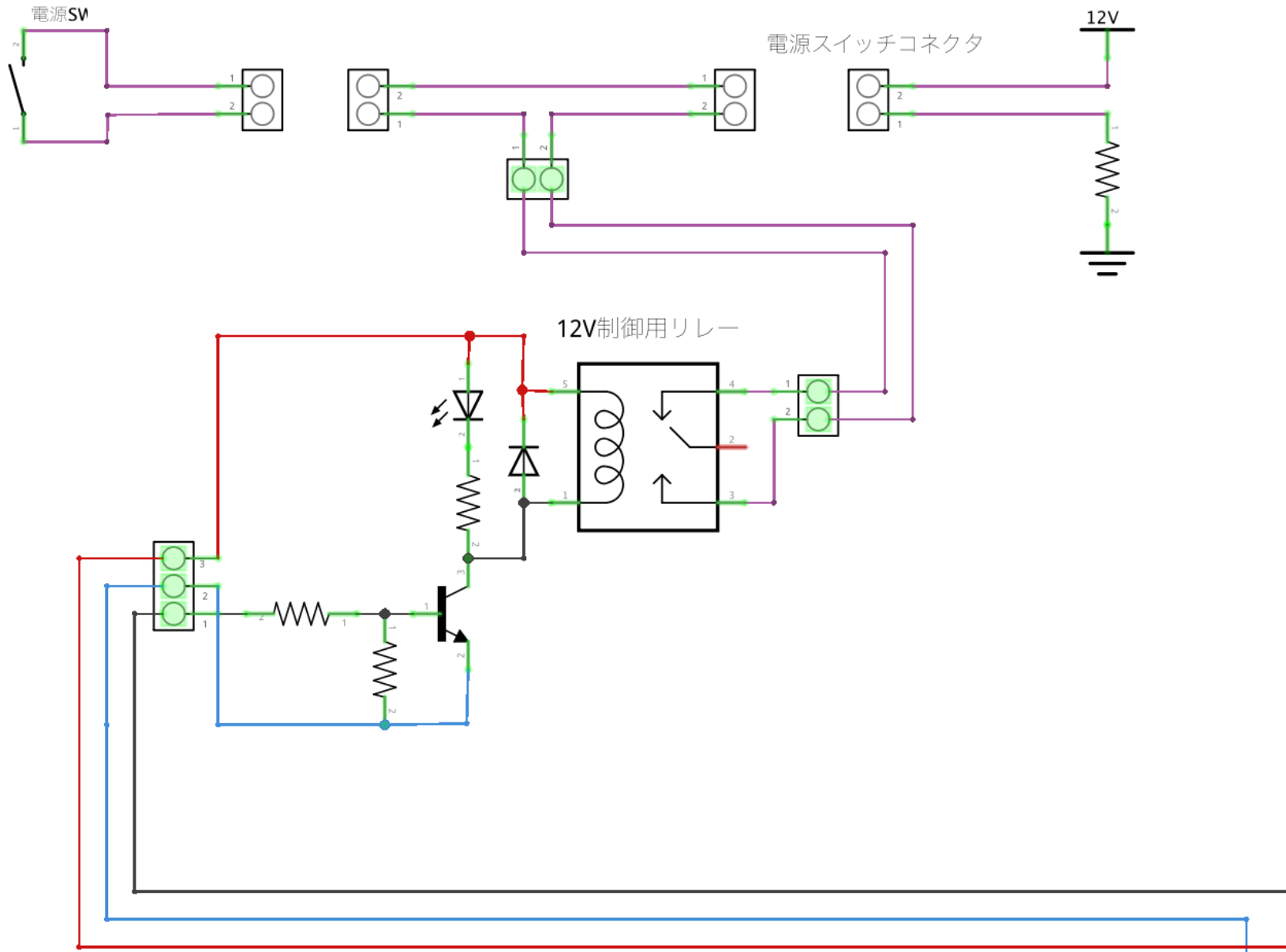


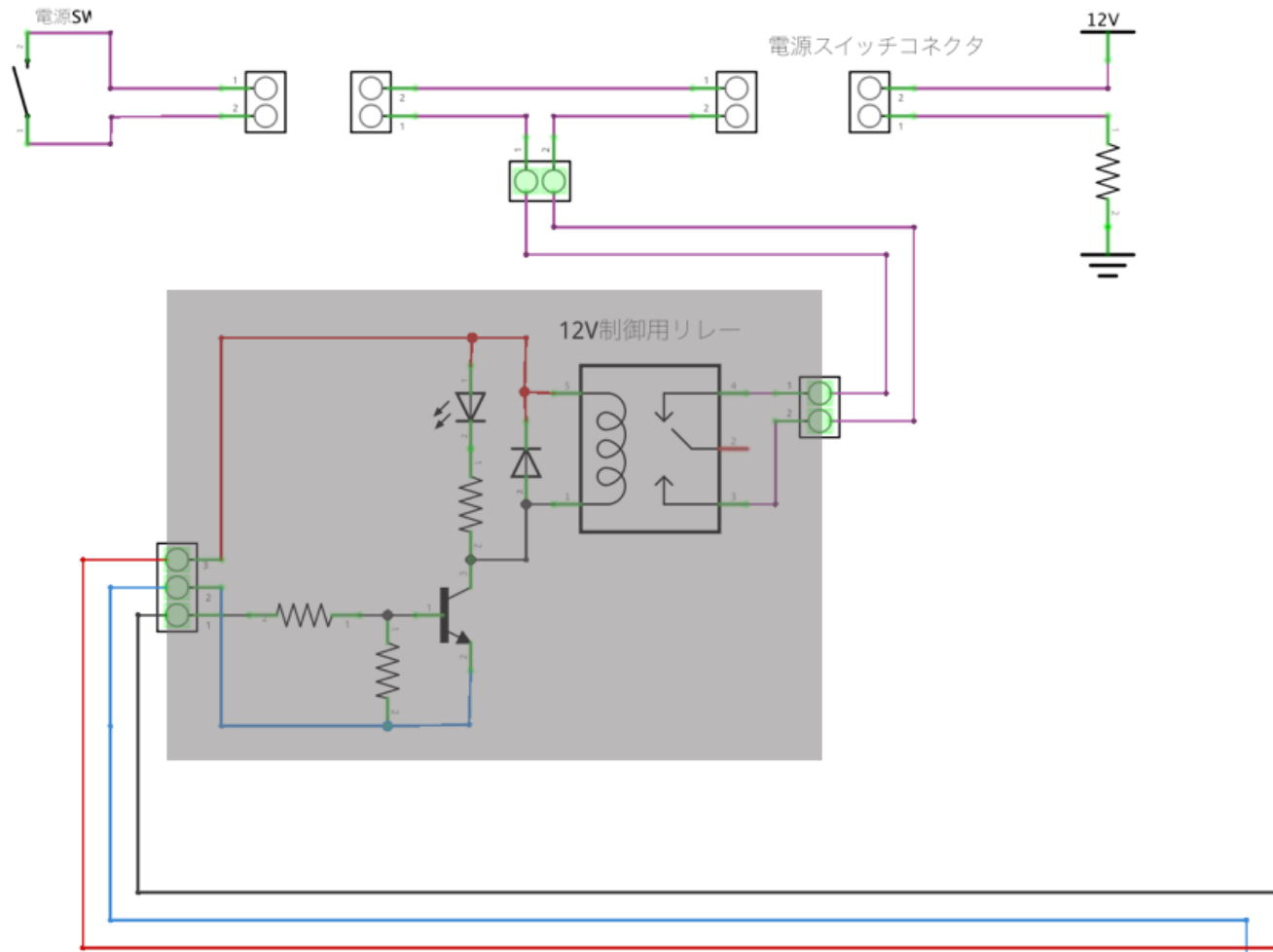
回路図 (全体)











実は最初はSSR(ソリッドステートリレー)で実装したが、
直流では使えないことがわかって作り直した...

- ソフトウェア側から入った人間としては、電子工作にいくらか苦勞したが、いい経験になった。
 - →SSRが直流では使えないとか。
- 作った当初はうまく動いていたが、冬になったら**幽霊が操作**している現象が発生
 - →コンデンサを追加
 - 最初ソフトウェアで対処しようとしたがうまくいかず、ハードウェア対処であっさり解消
- ソフトウェア部をラズパイ+Pythonで実装したが、**リアルタイム性やマルチタスク制御などに苦勞**した
 - →Elixirに注目するきっかけに
- 装置を外して(素のArcade1upにして)移設したのだが、噂話に尾ひれがついて、**気味悪がって避けられる**ようになってしまった。
 - →「勝手にデータ取られるのは嫌」という意識の発生
- いい経験になりました！

バイクツーリングログを
GoogleEarth で可視化して
みた。～KML瞬間解説～

静岡大学 曾根卓朗

2022/09/01 SWEST24 embLT

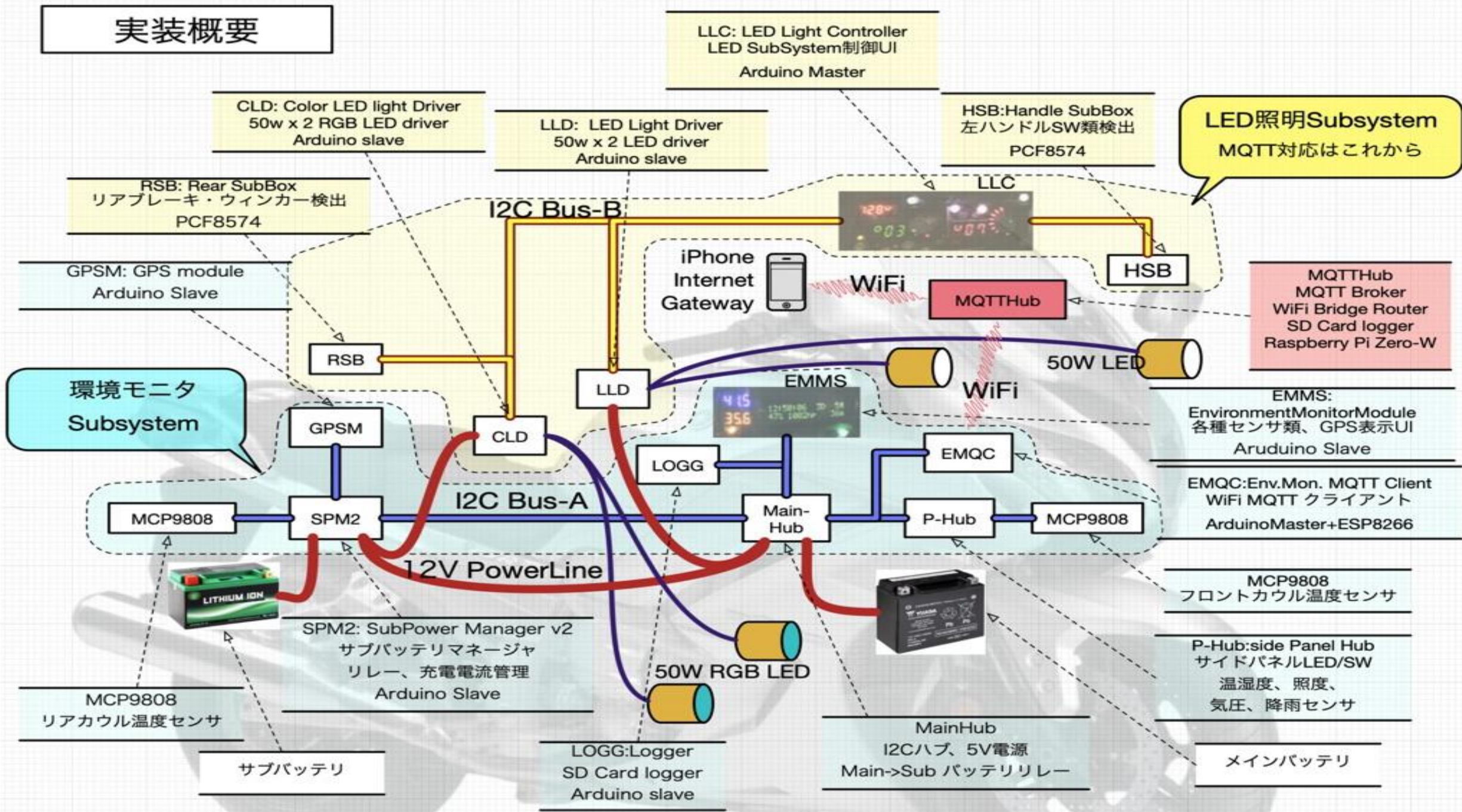


東京 横浜 郡山 福島 仙台 山形県 奥州 岩手県 盛岡 八戸 長岡 新潟 秋田 秋田県 弘前 青森 青森県 五所川原 函館 室蘭 小樽 千歳 留萌 旭川 富良野 上川町 士別 名寄 美深町 天塩町 幌延町 利尻富士町 礼文 大韓民国

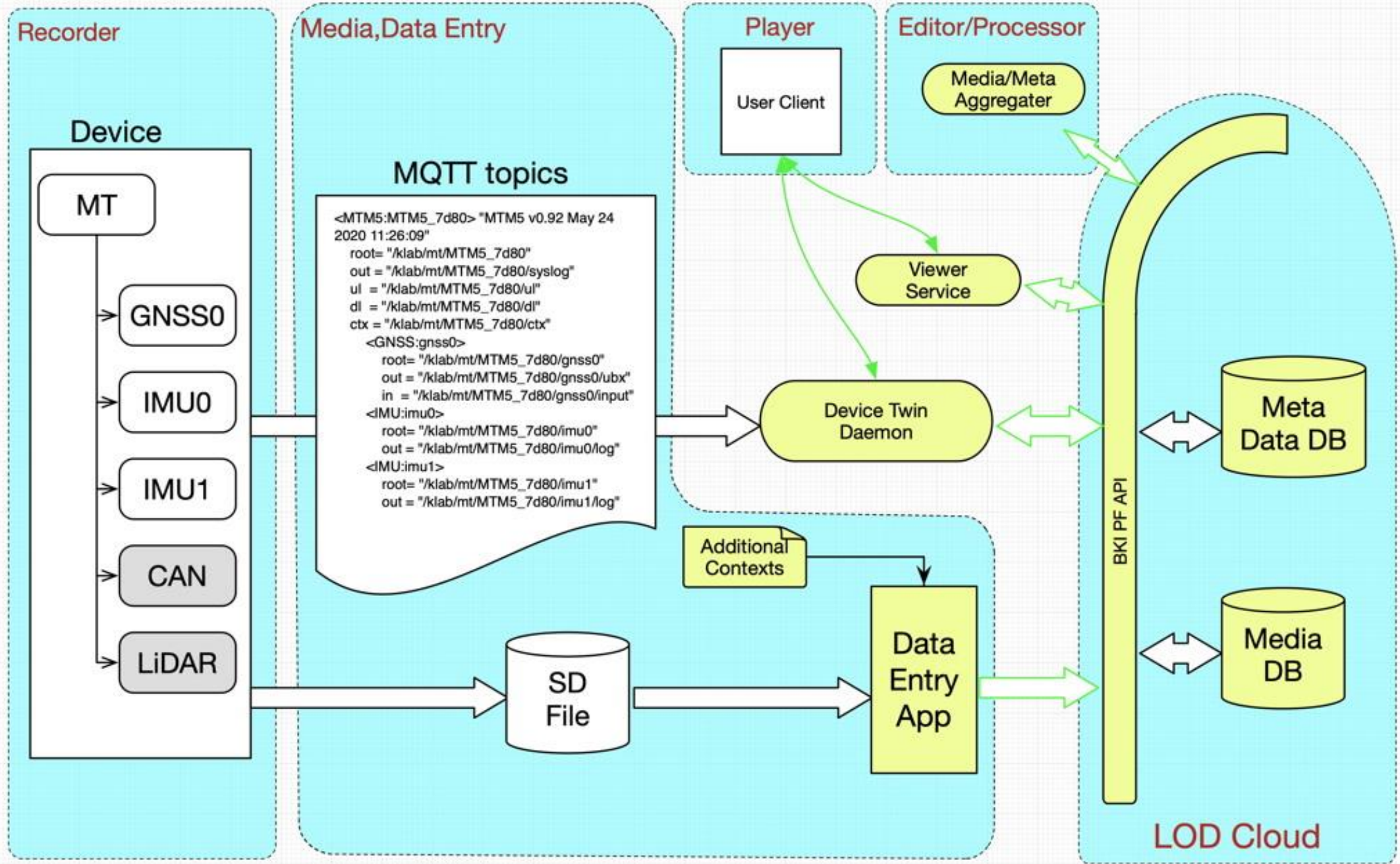
B0717_2008 E0717_2204 四国 広島 福岡 プサン 부산 クアンジュ 光州 大韓民国
E0731_2244 名古屋 B0731_2049 B0717_2305 金沢
E0720_0959 B0720_1038 E0719_1507 B0718_2035 E0719_1326 B0720_1141 E0719_1738 B0719_1632 E0730_2146 B0730_1223 B0719_1213
B0720_1151 E0726_1340 E0730_1259 E0730_1059
E0720_1133 B0720_1133 E0726_1027 B0726_0953 留萌 B0730_1050
E0726_1454 E0720_1514 B0726_1027 B0726_0953 留萌 B0730_1050
E0721_1102 E0720_1617 B0729_1101 B0729_1338 B0730_0937
E0721_1231 B0721_1201 E0721_1002 B0720_1600 上川町 E0729_1200 B0725_1426 E0729_1533
E0721_1231 B0721_1201 E0721_1002 B0720_1600 上川町 E0729_1200 B0725_1426 E0729_1533
B0727_1156 津別町 北見 B0721_1455 士別 名寄 美深町 B0725_1322 天塩町 B0729_1630
B0728_1339 E0728_1436 B0729_0831 速軽町 紋別 E0721_1653 E0722_0846 E0725_1229 B0725_1203 幌延町
B0728_1412 E0728_11501 B0728_1637 紋別 E0721_1653 E0722_0846 E0725_1229 B0725_1203 幌延町
B0728_1239 斜里町 網走 B0728_1541 B0722_0806 B0722_0856 E0722_1009 E0725_1111 利尻富士町
B0727_1803 羅臼町 E0727_1544 B0727_1639 B0722_1007 江頓別町 E0722_1002 E0725_1134 E0725_1111 利尻富士町
E0722_1149 E0725_1032

日本海

実装概要







GoogleEarthにはKML

The image shows a screenshot of the Google Developers website for KML (Keyhole Markup Language) documentation in Japanese. The browser address bar shows the URL: `developers.google.com/kml/documentation/kml_tut?hl=ja`. The page title is "KML (Keyhole Markup Language)". The navigation menu includes "ホーム", "ガイド", "リファレンス", "サンプル", and "サポート". The left sidebar contains a "フィルタ" (Filter) button and a list of sections: "概要" (Overview), "KML チュートリアル" (KML Tutorial), "デベロッパー ガイド" (Developer Guide), "Changing the Camera ツアー" (Changing the Camera Tour), "Data Types" (Data Types), and "Developer Concepts" (Developer Concepts). The main content area is titled "KML" and includes a breadcrumb trail: "ホーム > プロダクト > KML (Keyhole Markup Language) > ガイド". The text explains that KML is a file format used for displaying geographic data in Google Earth, Google Maps, and mobile Google Maps. It is based on XML and uses nested tags and attributes. The text also mentions that the documentation includes a reference for tags and that the order of elements in a KML file matters. A "フィードバックを送信" (Send Feedback) button is visible. An inset image shows a 3D map view with various colored overlays.

developers.google.com/kml/documentation/kml_tut?hl=ja

KML (Keyhole Markup Language)

ホーム ガイド リファレンス サンプル サポート

フィルタ

概要

KML チュートリアル

デベロッパー ガイド

Changing the Camera ツアー

標高モード

時間とアニメーション

カメラ

Data Types

KML の Sky データ

写真オーバーレイ

カスタム データの追加

モデル

Developer Concepts

KMZ ファイル

リージョン

更新

有効期限

ホーム > プロダクト > KML (Keyhole Markup Language) > ガイド

KML

KML は、Google Earth、Google マップ、モバイル Google マップなどの Earth ブラウザで、地理データの表示に使用するファイル形式です。KML は、XML 標準をベースにしており、ネストされた要素や属性を含むタグ構造を使用します。すべてのタグで大文字と小文字が区別されません。タグは [KML リファレンス](#) の定義どおりに記述する必要があります。このリファレンスには、省略できるタグが記載されています。タグは、リファレンスで記述されている順番で要素内に指定する必要があります。

KML を初めて使う場合は、このドキュメントと付属のサンプル ファイル ([SamplesInEarth](#) と [SamplesInMaps](#)) を参照して、KML ファイルの基本構造とよく使われるタグについて確認してください。最初のセクションでは、Google Earth のユーザー インターフェースを使用して作成できるアイテムについて説明します。これには、目印、説明、地面オーバーレイ、パス、ポリゴンなどが含まれます。2 番目のセクションでは、テキスト エディタを使用して KML を作成する必要があるアイテムについて説明します。`.kml` または `.kmz` 拡張子を付けて保存したテキスト ファイルは、Earth ブラウザで表示できます。

この情報は役に立ちましたか?

フィードバックを送信

可視化したログのKML表現

- <?xml ..>
- <kml xmlns=<http://earth.google.com/kml/2.1>>
- <Document>
- <Style id="BEGIN"> ... </Style> ... **マーカー定義**
- <Folder>
- <name>Day1:210717</name> **1日目のログ**
- <Folder> ... **1つ目の走行軌跡**
- <Placemark> 始点のstyleidを指定 </Placemark>
- <Placemark> ... <LineString> 走行軌跡の3次元点列 </LineString></Placemark>
- <Placemark> 終点のstyleidを指定 </Placemark>
- </Folder>
- ... 2つ目の走行軌跡...
- ...
- </Folder>
- ...Folder 列...
- </Document>
- </kml>
- </xml>

2021_Hokkaido

Converted by rnxp2kml v1.3 at 211012_132428

KML 自動保存日: 数秒前

アイテムを追
加

プレゼンテーションを
開始

- ▼ Day1:210717
 - ▼ 2008-2204
 - 📍 B0717_2008
 - 📍 MOVE:210717_2008-2204
 - 📍 E0717_2204
 - ▼ 2305-2308
 - 📍 B0717_2305
 - 📍 MOVE:210717_2305-2308
 - 📍 E0717_2308
 - ▼ Operations
 - ▶ BRAKE(91)
 - ▶ RTURN(102)
 - ▶ LTURN(103)
 - ▶ HIBEAM(2)
 - ▶ PASSING(1)
 - ▶ Day2:210718
 - ▶ Day3:210719

Google EarthのKML実装

- 使った機能はほんの一部
- カメラアングル指定、写真オーバーレイ等々、多くの機能がある。
- KMLを読み込んだ状態を Project として公開する方法は見つからず。
- まだ作りかけな感じではある。
- とはいえ、データ作成は難しくない。

GoogleEarthのトピックス

- 平野部以外の地形の標高はあまり正確でない
 - 山間部は地形が丸め込まれている
 - 海岸線沿いは実際より標高が高く表現されている
- 平野部の標高はほぼ正確と思われる
 - GNSS測位は 回転楕円体高が求まるので、国土地理院のジオイド高を加えて標高を求める=>この値とGoogleEarth の表示はほぼあっていた。
 - (GNSS受信機が吐き出す「標高(MSL)」は近似値なのでメートル単位ですれる。)
- KMLのサイズが大きくなると動きが遅くなる
 - 今回のデータは約19Mb