組込み向けテスト実行フレームワークのご紹介

JaSST東海2023LT

2023/2/10　見澤 広志（ASTER 組込みCI研究WG）

自己紹介

- 見澤 広志
- 元半導体部品メーカのソフト開発者、現在、自動車関連会社でソフト開発に従事。
- 社外：JaSST Tokai実行委員／ASTER 組込みCI研究WG
- 最近の興味：AWS、Docker
- 趣味：子どもに動くおもちゃ作り、体を動かすこと

本日のLT資料はテスト自動化カンファレンス2021で発表した資料を抜粋したものです。

テスト自動化カンファレンス2021の発表資料と
発表動画は以下のリンクからご覧ください。

- 発表資料：https://speakerdeck.com/misawasan/zu-ip-mixiang-ketesutoshi-xing-huremuwaku-falsegoshao-jie
- 発表動画：https://www.youtube.com/watch?v=VggVeLqcGd4

今日お伝えしたいこと

１．仮想環境でのテストをCIパイプラインに
組込むと、組込みソフトのテストが
高度化する。

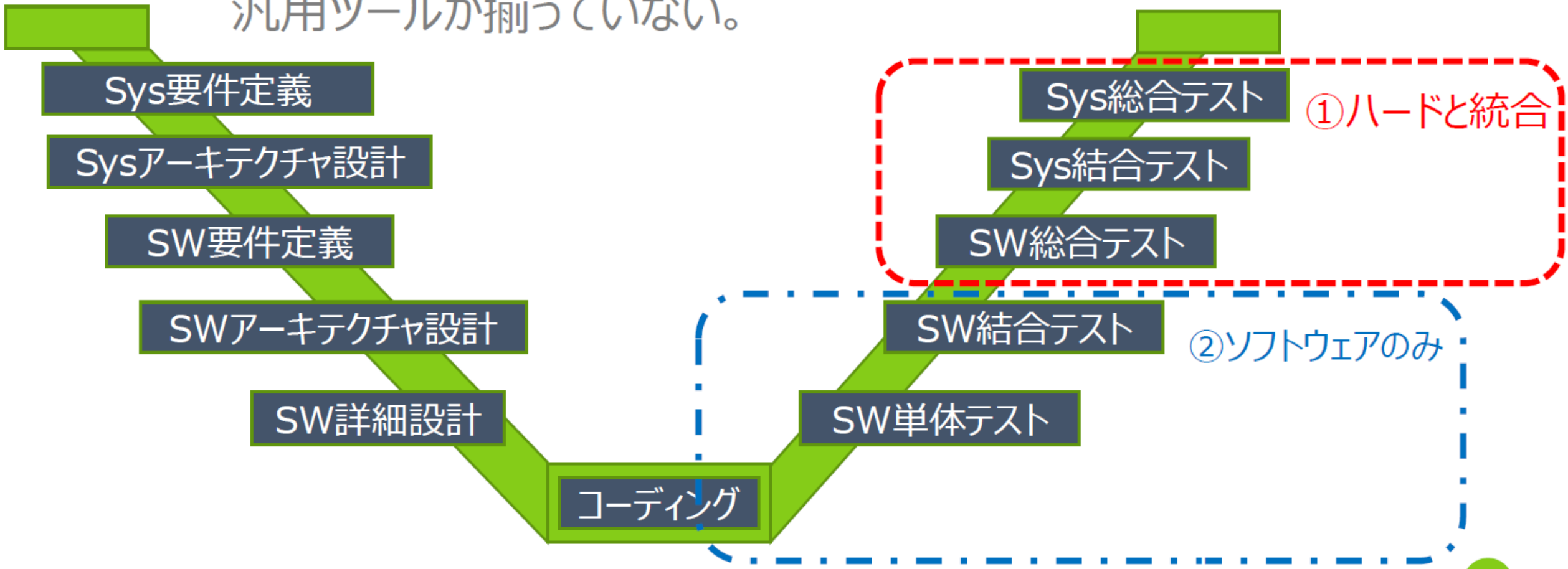
アジェンダ

- 組込み製品のテストの課題
- サンプルのテスト対象の説明
- テスト環境の説明(HILSとSILS)
- デモ(HILSとSILS)動画のURLの紹介
- 本ツールのメリット、デメリット

組込み製品のテストの課題

組込み製品のテストの課題

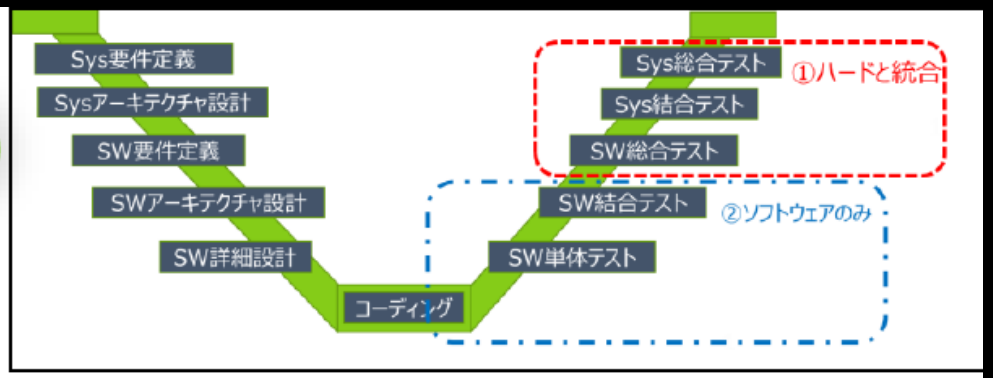
課題１　ハードと統合した状態でテストが必要。
課題２　ソフトウェアの機能確認を先行して不具合を潰しこみたいが、
　　汎用ツールが揃っていない。



© 2021 組込みCI研究WG All rights reserved

組込みのテスト前倒しの課題

課題３　ソフトウェアだけのテスト環境を作ったとしても、入出力が異なる為、テストケースを再利用しずらい。



【凡例】 実物　模擬

テスト環境②

テスト対象

入出力＝ソフトの入出力
（変数１＝250など）

マイコン

センサ　アクチュエータ

テスト環境
(SILS)

テスト環境②

テスト対象

入出力＝ソフトの入出力
（変数１＝250など）

センサ　アクチュエータ

テスト環境
(PILS/S-PILS)

テスト環境①

テスト対象
（ハードと統合）

センサ　アクチュエータ

入出力＝ハードの入出力
（ピン１＝２．５Vなど）

(HILS)

© 2021 組込みCI研究WG All rights reserved

課題と対策

課題と対策

《課題》

- ハードと統合した状態でテストが必要。
- ソフトウェアの機能確認を先行して不具合を潰しこみたいが、汎用ツールが揃っていない。
- テスト環境毎に入出力が異なる為、テストケースを再利用しずらい。

《対策》

- SILS環境をオープンソースを活用して作成する。
- シナリオレベルのテストケースをテスト環境で共通化する。
- テスト環境の違いは実装側（ライブラリ）で吸収する。

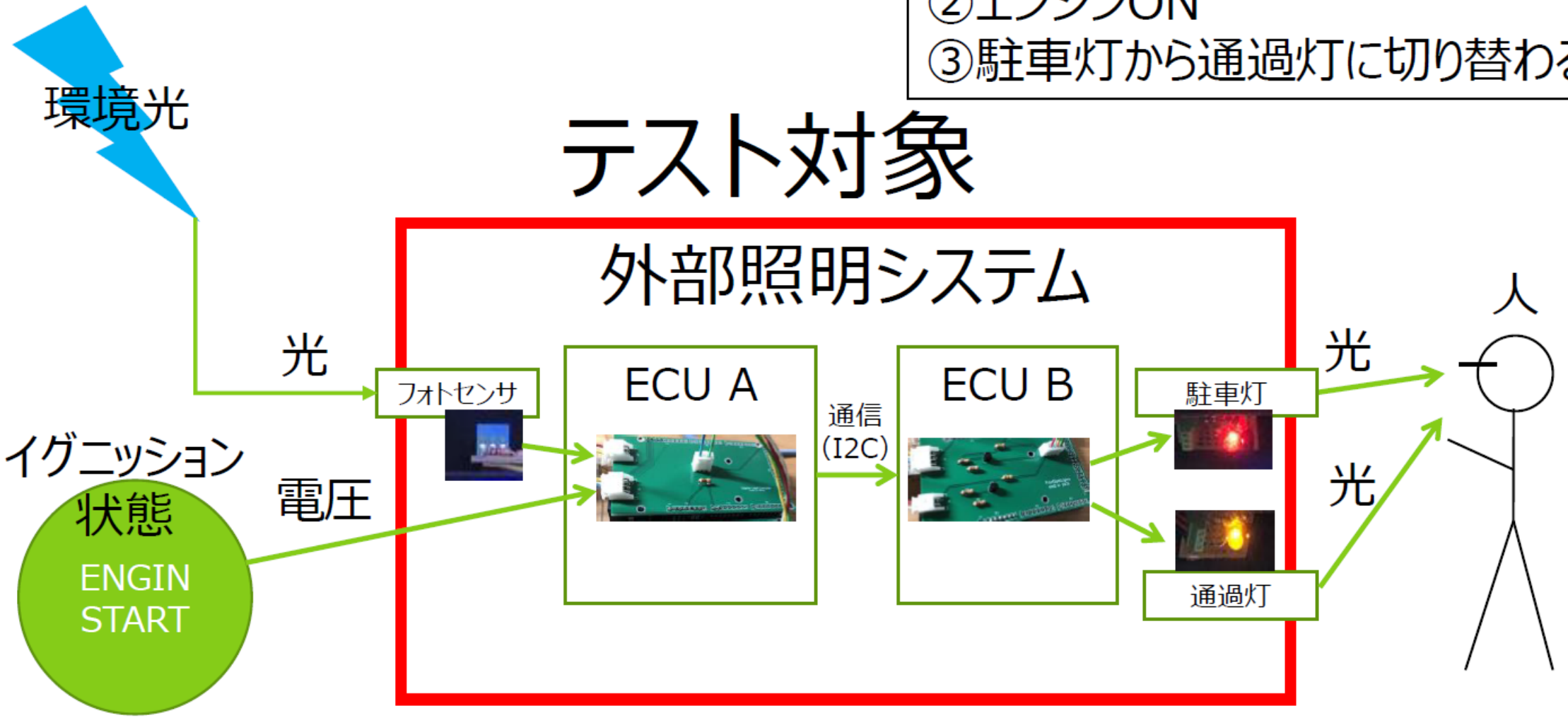
© 2021 組込みCI研究WG All rights reserved

サンプルのテスト対象の説明

外部照明システム

《テストのシナリオ》

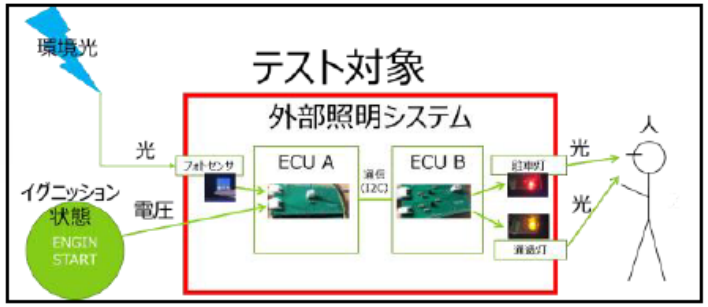
①夜間、駐車灯ON状態
②エンジンON
③駐車灯から通過灯に切り替わること

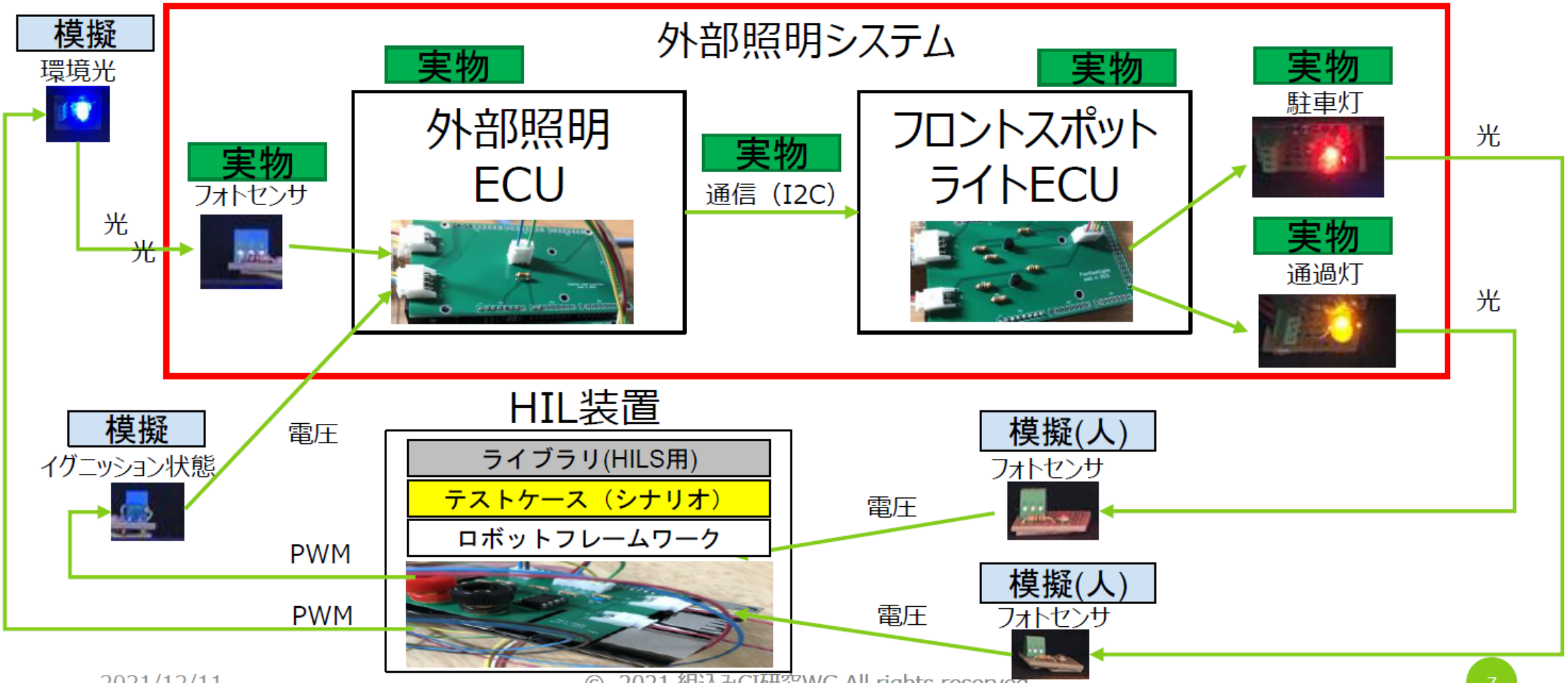


© 2021 組込みCI研究WG All rights reserved

テスト環境の説明(HILSとSILS)

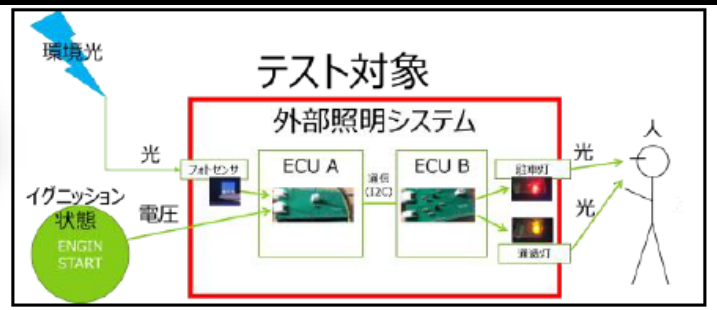
外部照明システムのテスト環境（HIL）

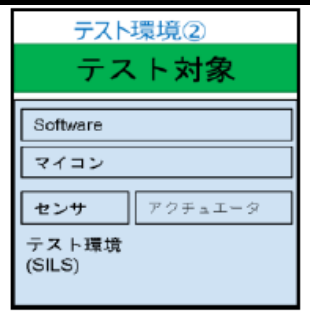


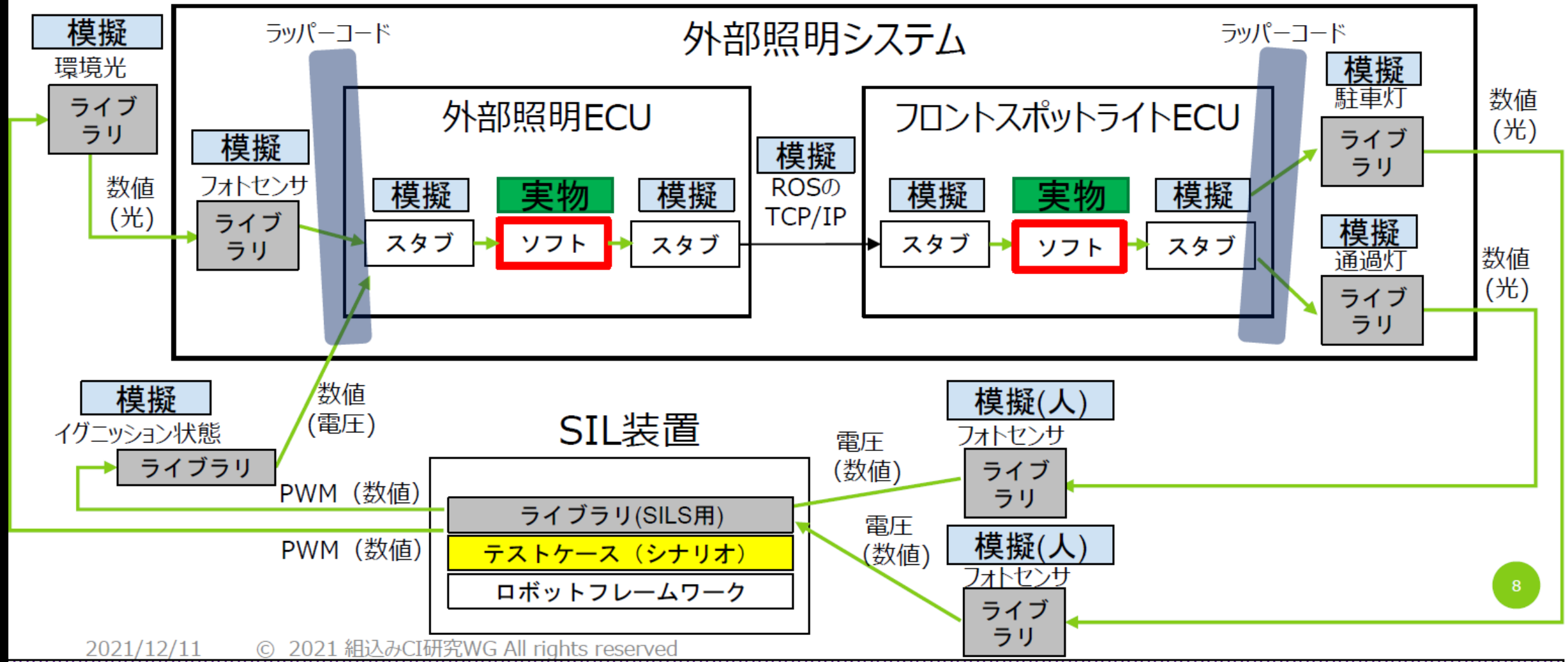


© 2021 組込みCI研究WG All rights reserved

外部照明システムのテスト環境（SIL）





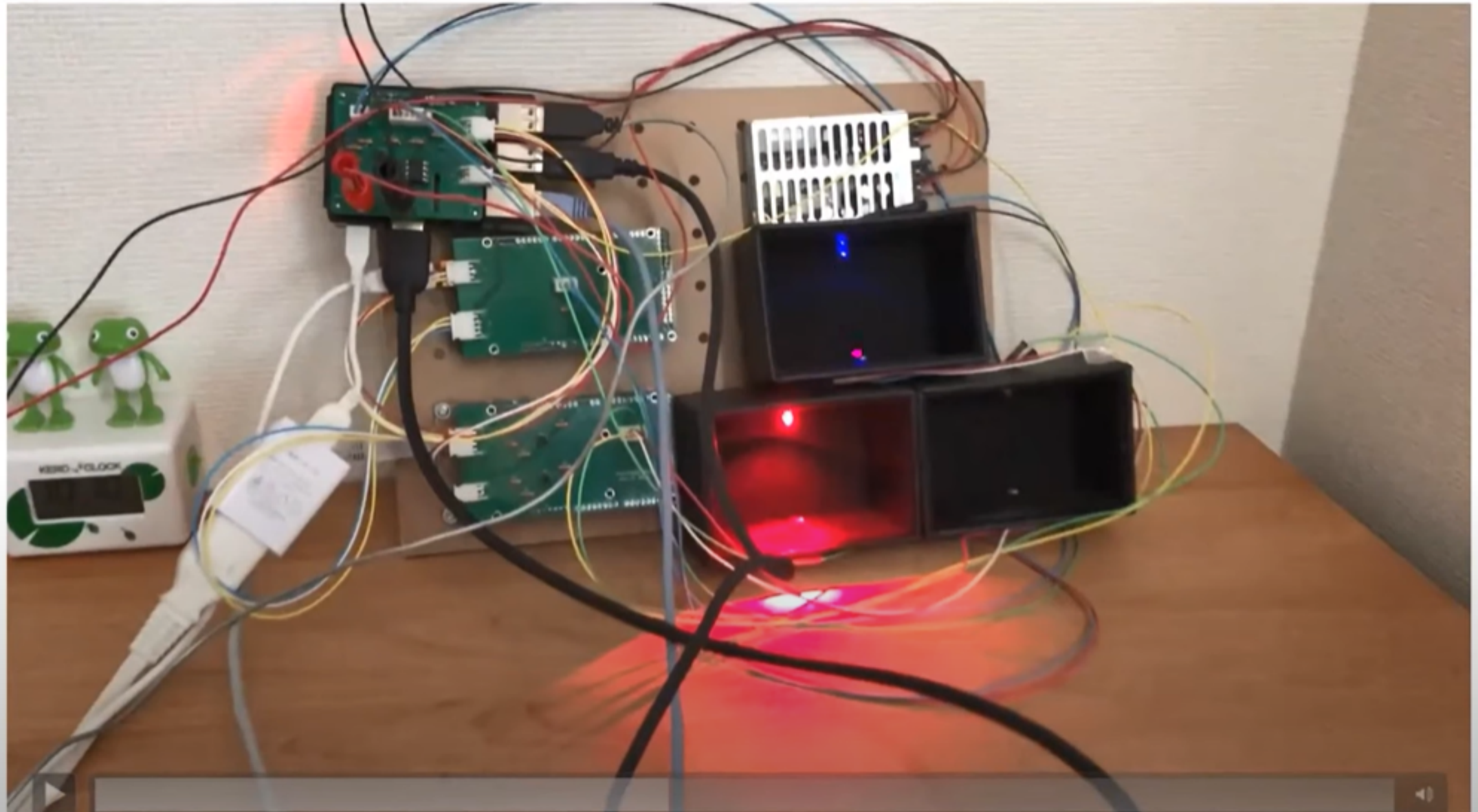


2021/12/11 © 2021 組込みCI研究WG All rights reserved

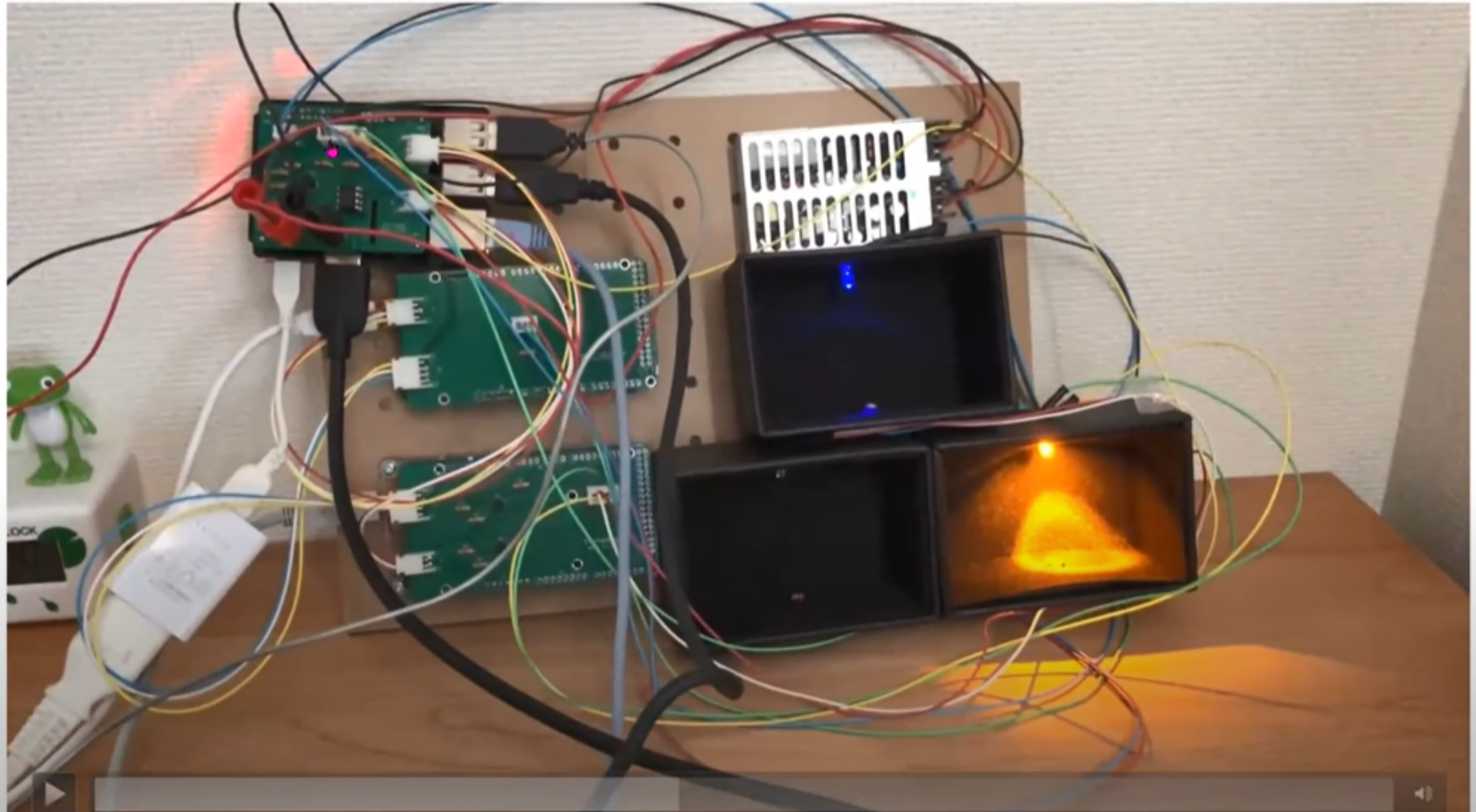
デモ(HILSとSILS)動画のURLの紹介

HILSデモ動画：https://www.youtube.com/watch?v=VggVeLqcGd4 (18分付近)

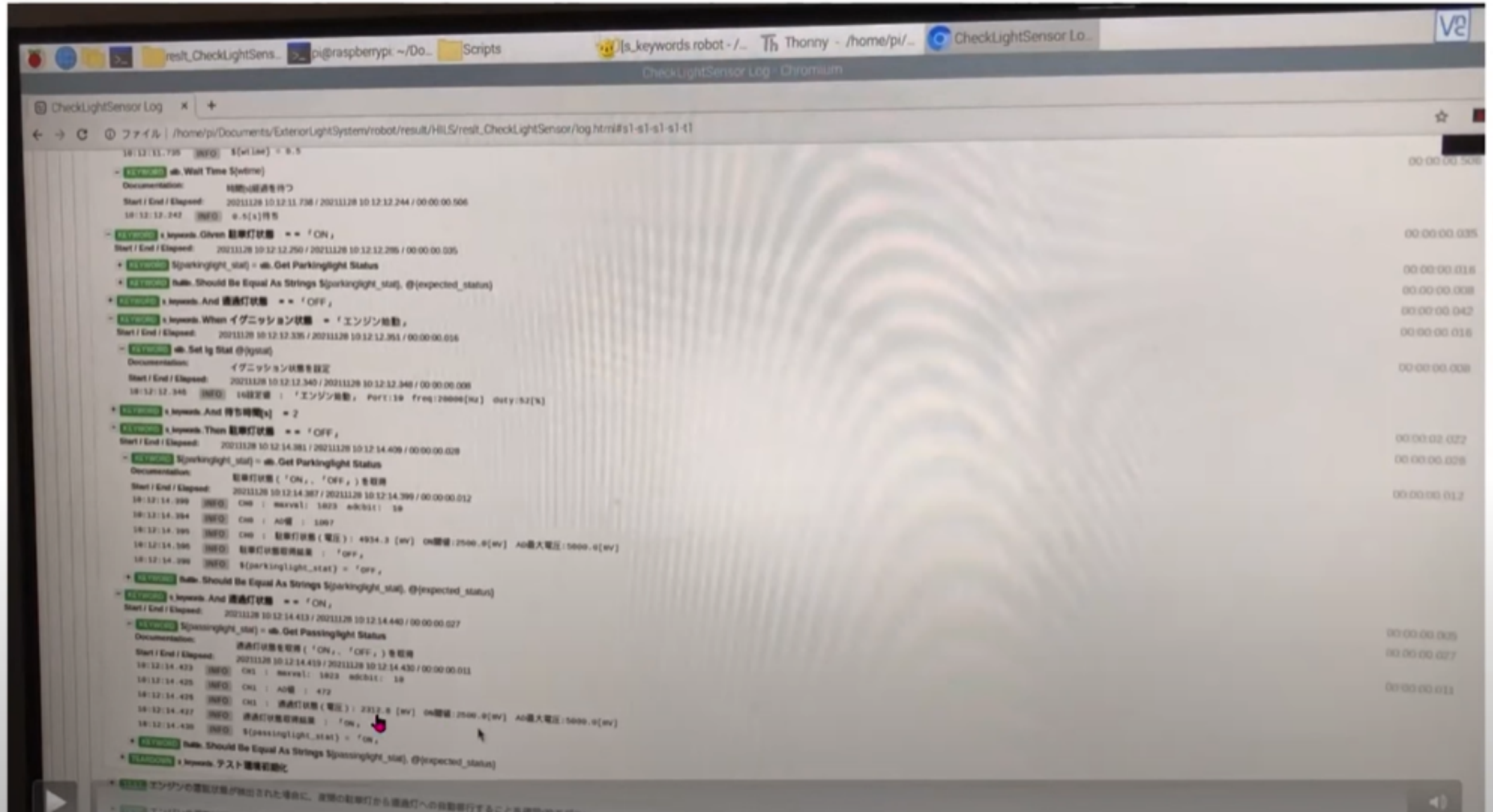
デモ（HILS環境）



デモ（HILS環境）

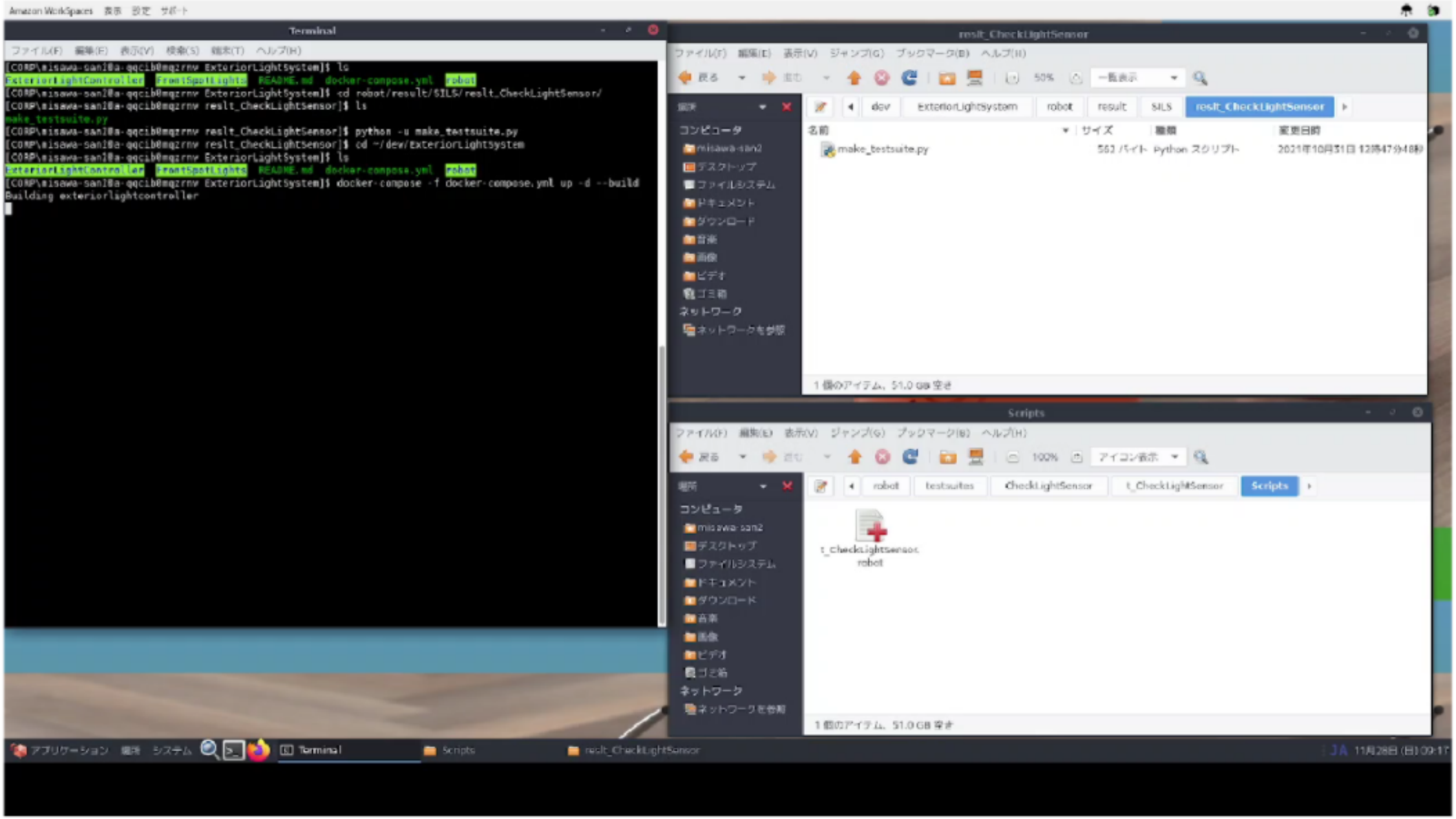


デモ（HILS環境）

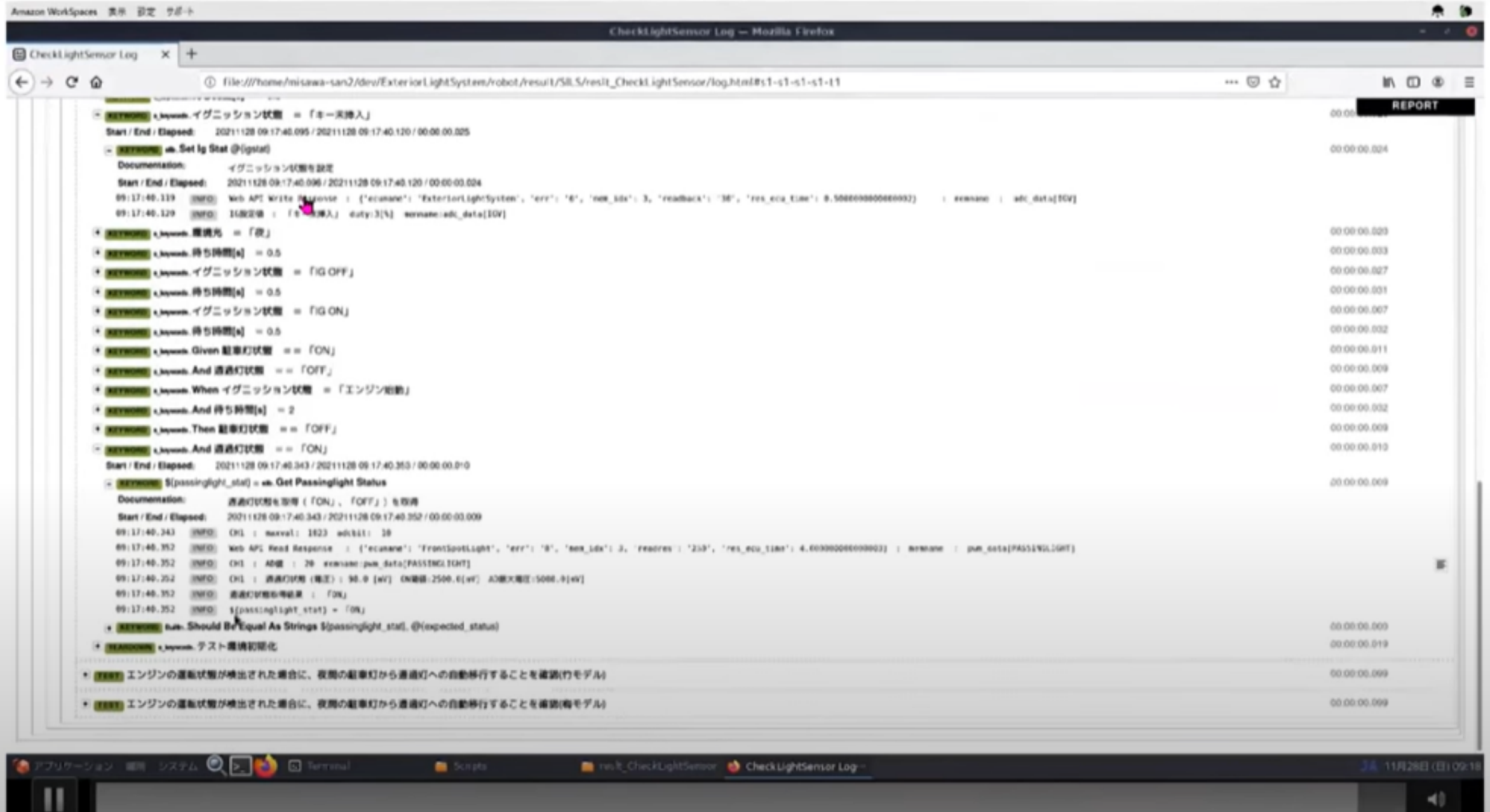

CheckLightSensor Log
/home/pi/Documents/ExtenorLightSystem/robot/result/HILS/resit_CheckLightSensor/log.html#s1-s1-s1-s1-t1

SILSデモ動画：https://www.youtube.com/watch?v=VggVeLqcGd4 (22分付近)

デモ（SILS環境）


Terminal
[CORP\misawa-san1@a-qqcib0mqzrnv ExteriorLightSystem]$ ls
ExteriorLightController FrontSpotlight README.md docker-compose.yml robot
[CORP\misawa-san1@a-qqcib0mqzrnv ExteriorLightSystem]$ cd robot/result/SILS/reslt_CheckLightSensor/
[CORP\misawa-san1@a-qqcib0mqzrnv reslt_CheckLightSensor]$ ls
make_testsuite.py
[CORP\misawa-san1@a-qqcib0mqzrnv reslt_CheckLightSensor]$ python -u make_testsuite.py
[CORP\misawa-san1@a-qqcib0mqzrnv reslt_CheckLightSensor]$ cd ~/dev/ExteriorLightSystem
[CORP\misawa-san1@a-qqcib0mqzrnv ExteriorLightSystem]$ ls
ExteriorLightController FrontSpotlight README.md docker-compose.yml robot
[CORP\misawa-san1@a-qqcib0mqzrnv ExteriorLightSystem]$ docker-compose -f docker-compose.yml up -d --build
Building exteriorlightcontroller
reslt_CheckLightSensor
dev
ExteriorLightSystem
robot
result
SILS
reslt_CheckLightSensor
make_testsuite.py
Scripts
robot
testsuites
CheckLightSensor
t_CheckLightSensor
Scripts
t_checklightsensor.robot

デモ（SILS環境）


Amazon WorkSpaces
CheckLightSensor Log — Mozilla Firefox
CheckLightSensor Log
REPORT
Start / End / Elapsed: 20211128 09:17:40.095 / 20211128 09:17:40.120 / 00:00:00.025
Start / End / Elapsed: 20211128 09:17:40.343 / 20211128 09:17:40.352 / 00:00:00.009

テスト実行結果レポート（SIL環境）

GCC Code Coverage Report

Directory: src/pubsub/src/app/
Date: 2021-10-31 03:50:16
Legend: low: >= 0% medium: >= 75.0% high: >= 90.0%

	Exec	Total	Coverage
Lines:	67	73	91.8%
Branches:	8	12	66.7%

File	Lines			Branches	
ExteriorLightController.c		100.0%	26 / 26	100.0%	4 / 4
adc.c		73.3%	11 / 15	-%	0 / 0
calc_v.c		93.8%	30 / 32	50.0%	4 / 8

Generated by: GCOVR (Version 5.0)

GCC Code Coverage Report

Directory: src/pubsub/src/app/
File: src/pubsub/src/app/calc_v.c
Date: 2021-10-31 03:50:16

	Exec	Total	Coverage
Lines:	30	32	93.8%
Branches:	4	8	50.0%

Line	Branch	Exec	Source
1			#include <adc.h>
2			#include <calc_v.h>
3			
4			uint16_t calc_v_env_light = 0U; /* ENV_LIGHT : unit mV */
5			uint16_t calc_v_igv = 0U; /* IGV : unit mV */
6			
7			void calc_v_step(void);
8			void calc_v_set_env_light(uint16_t env_light);
9			uint16_t calc_v_get_env_light(void);
10			void calc_v_set_igv(uint16_t igv);
11			uint16_t calc_v_get_igv(void);
12			
13		451	static void calc_v_ad2env_light(void)
14			{
15		451	uint32_t val = 0UL;
16			
17		451	val = adc_get_env_light();
18			
19	▶ 1/2	451	if(val < 1024UL)
20			{
21		451	val = CALC_V_ENV_LIGHT_LSB * val;
22	▶ 1/2	451	if(val > 65535UL)
23			{
24		✗	val = 65535UL;
25			}
26		451	calc_v_set_env_light((uint16_t)val);
27			}
28			else
29			{
30			/* 前回値保持 */
31			}
32		451	}

異常系処理の為、
実行されなかった。

© 2021 組込みCI研究WG All rights reserved

本ツールのメリット、デメリット

本ツールのメリット、デメリット

《メリット》

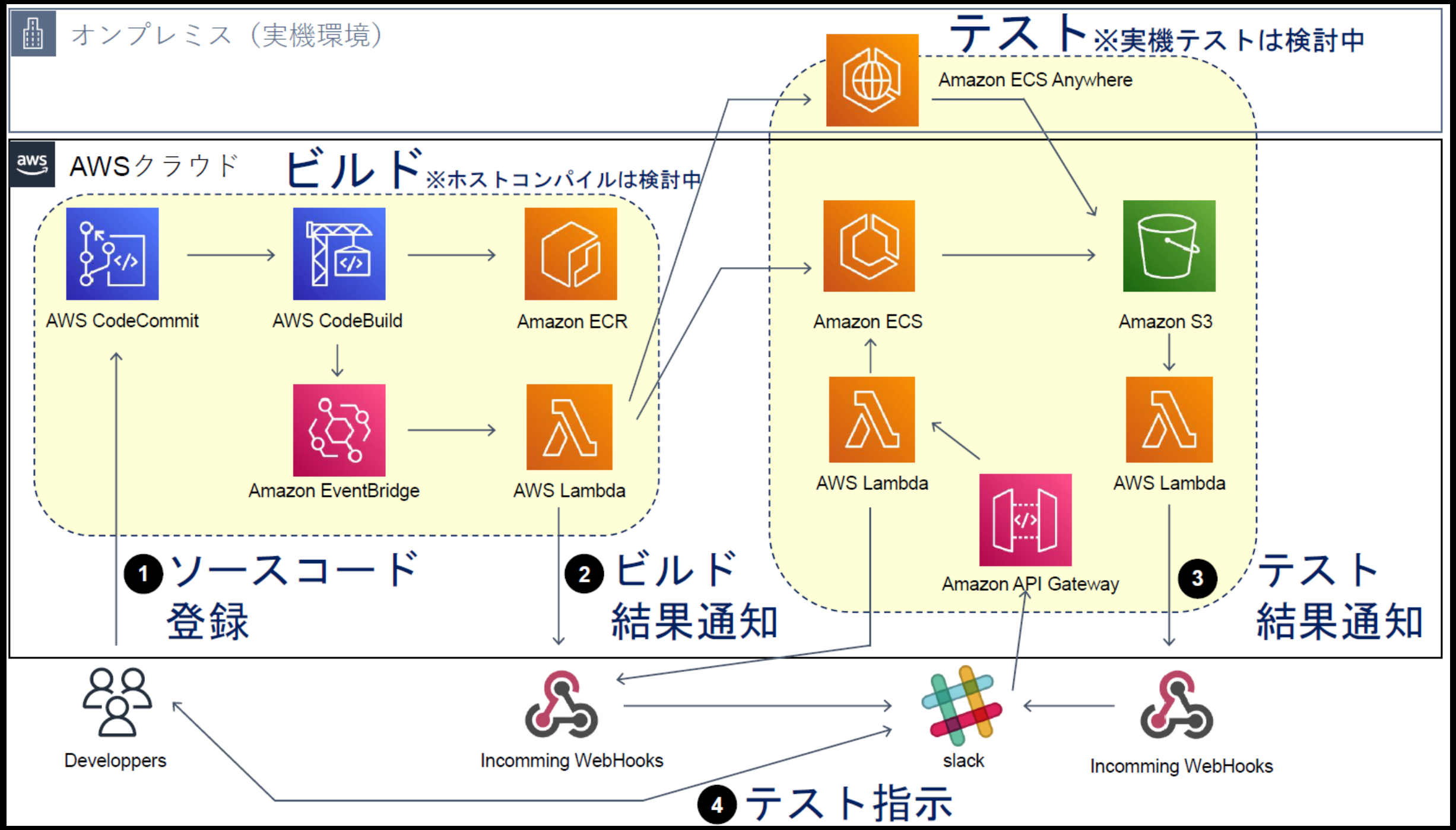
- ソースコードがあれば、**早期にテストを始めることができる。**（品質）
- 機能的な不具合を**実機が無くても**、潰すことができる。（品質）
- ソフトウェアの為、**テスト環境を増やすことができる**。（品質）
- シナリオレベルでテストケースを**再利用することができる。**（費用）
- オープンソースを使用している為、**安価に作成できる**。（費用）

《デメリット》

- IT系のツールを駆使する為、組込み系の人材には敷居が高い。
- ライブラリのメンテナンスするのが難しい。
- リアルタイム性能の検証に使用できない。

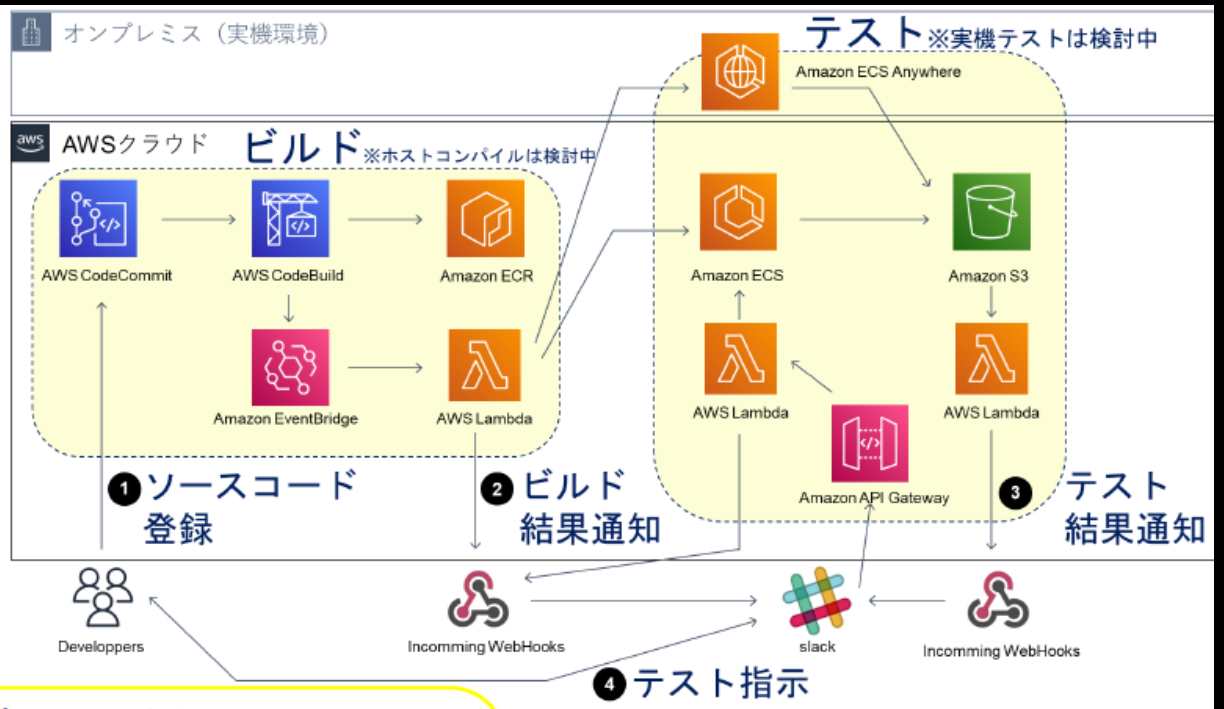
© 2021 組込みCI研究WG All rights reserved

Now..


オンプレミス（実機環境）
テスト※実機テストは検討中
Amazon ECS Anywhere
AWSクラウド
ビルド※ホストコンパイルは検討中
AWS CodeCommit
AWS CodeBuild
Amazon ECR
Amazon EventBridge
AWS Lambda
Amazon ECS
Amazon S3
AWS Lambda
AWS Lambda
Amazon API Gateway
1 ソースコード登録
2 ビルド結果通知
3 テスト結果通知
Developpers
Incomming WebHooks
slack
Incomming WebHooks
4 テスト指示

```
ec2-user:~/environment $
ec2-user:~/environment $ git push
```

❶ ソースコード登録



incoming-webhook アプリ 15:38

ビルド状況をお知らせします
CodeBuild: test-pip3-bld-prj - SUCCEEDED

ビルド成功したので、テストスイートを実行します
CheckLightSensor

ビルド成功したので、テストスイートを実行し
CheckLightSensor2

テスト実行（並列）

❷ ビルド結果通知

ビルド状況をお知らせします
CodeBuild: test-pip3-bld-prj - IN_PROGRESS

テストが完了しました
https:/ amazonaws.com/event-trigger/dest/robotf/

テストが完了しました
https:/ amazonaws.com/event-trigger/dest/robotf/

❸ テスト結果通知

組込みCI研究WG参加者募集

組込みCI研究WGでは、継続的インテグレーション、継続的テスト、継続的デプロイなどの研究を実施しております。
もし、興味がありましたら、ご参加ください。

月1度、主に日曜日の13:30～17:00で実施。
(オンライン)

ご興味ある方は、以下のメールアドレスまで連絡をお願いします。

aster-embci@qualab.jp

Photo by Ronak Valobobhai on Unsplash