

機能検査用自動化ツール「SmrAI」の開発

～製造業のプログラミング人材増加に向けたSmalrubyの新たな活用方法～

杉山 耕一郎¹, 奥田 彩月^{1,2}, 田部 翼³,
高尾 宏治⁴, 東 裕人⁵, 田中 和明⁶

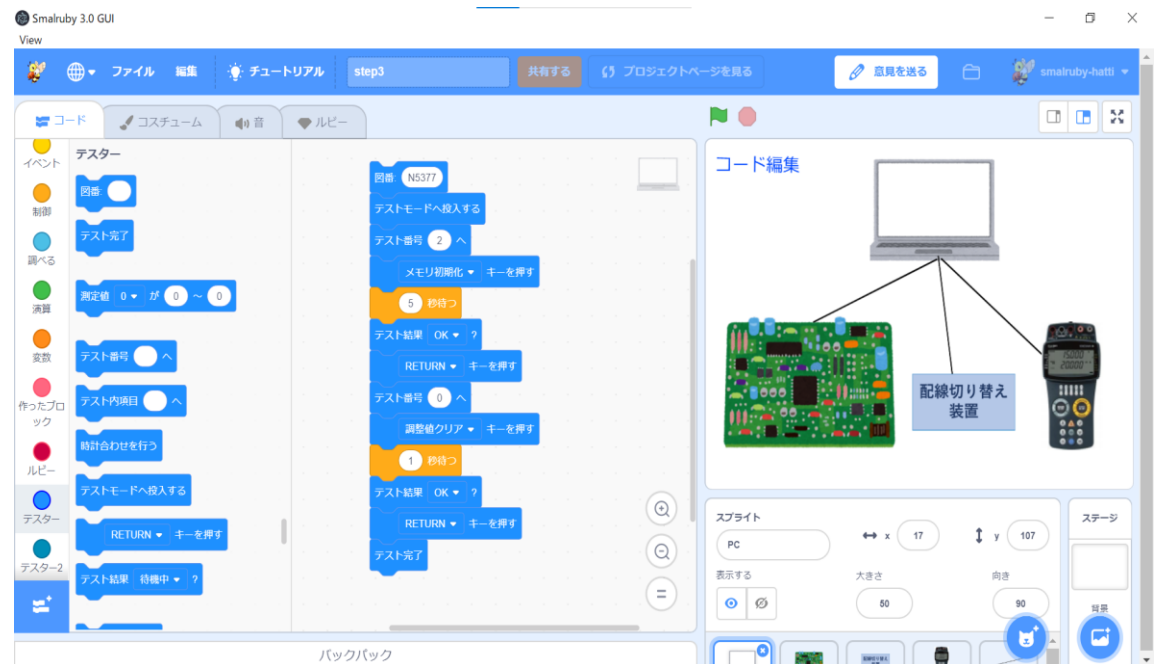
¹松江工業高等専門学校 情報工学科, ²サイバートラスト株式会社, ³株式会社ニッポー島根工場

⁴NPO 法人 Ruby プログラミング少年団, ⁵しまねソフト研究開発センター, ⁶九州工業大学 情報工学研究院

2021/12/16 RubyWorldConference 2021 Session-3

目次

- 着想に至った経緯：
松江高専と地域 Ruby コミュニティとの連携
- 機能検査用自動化ツール「SmrAI」の開発と成果について
 - 背景
 - 設計
 - ツール概要
 - 実証試験
 - まとめ



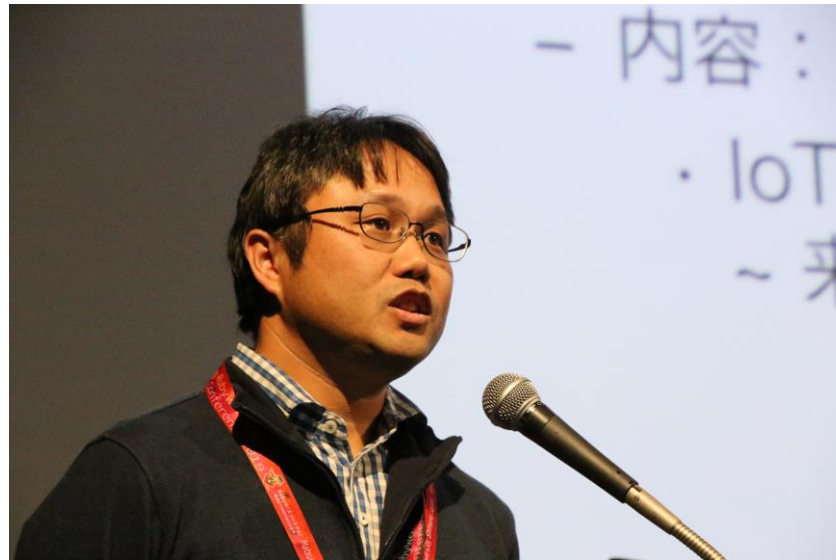
第1部

着想に至った経緯：松江高専と地域Rubyコミュニティとの連携

松江高専 情報工学科
杉山 耕一郎

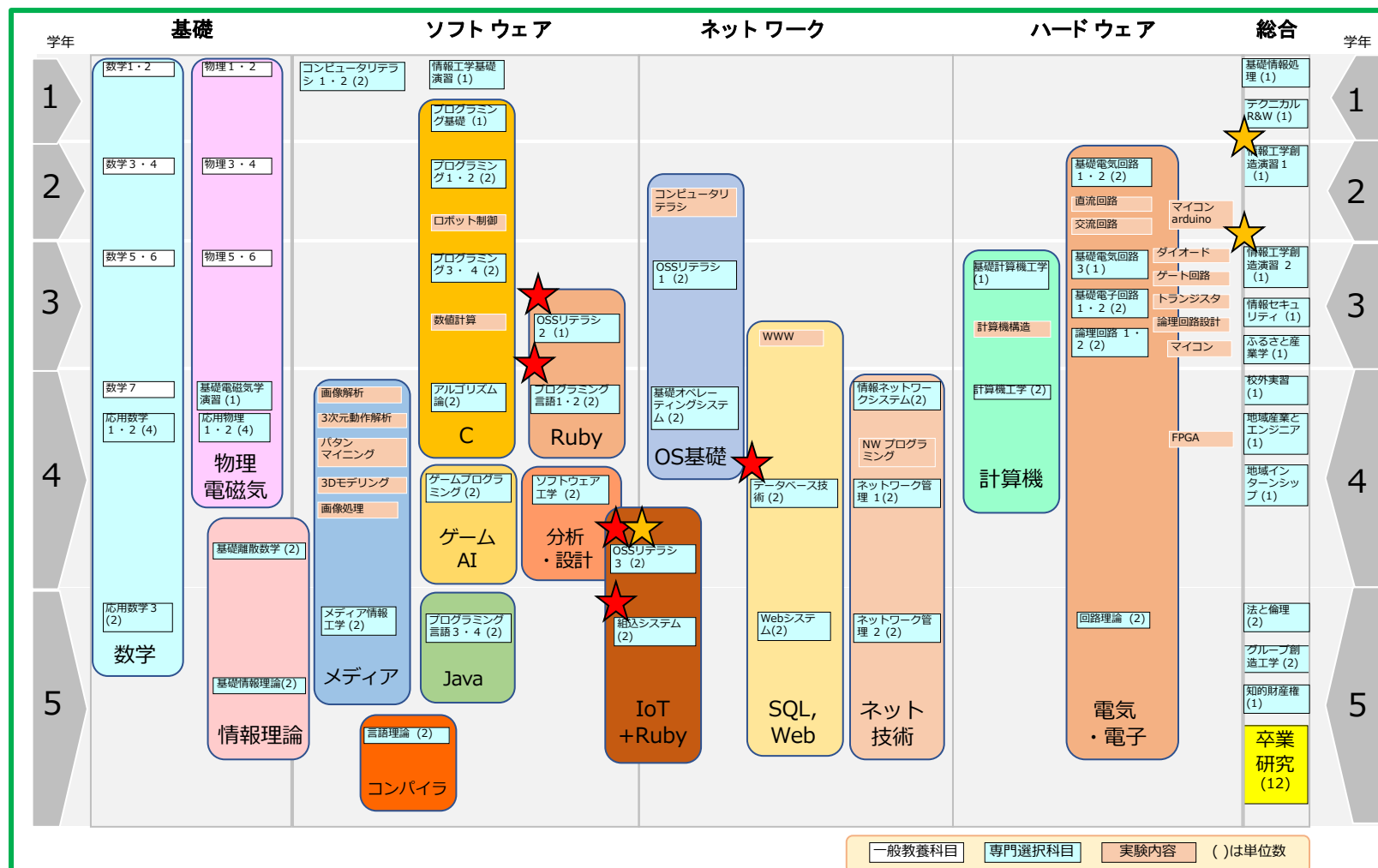
発表者略歴

- 名前: 杉山 耕一郎 (すぎやま こういちろう)
- 所属: 松江高専 情報工学科 (2016年度～)
- 専門: 惑星気象学 & 地球惑星情報学
 - 雲対流モデル(数値流体シミュレーションモデル)開発
 - 木星・金星・火星
 - IoT を活用した環境モニタリングシステム



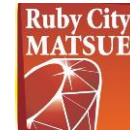
松江高専 情報工学科

- 特色：地域Rubyコミュニティとの連携
 - 授業でのRuby言語 ★ コミュニティイベント ☆ の積極活用



連携活動の例

• Ruby City Matsue プロジェクトとの連携



概要

- 目的：「Ruby人材育成」
- 技術向上のための支援 (2008年度～)
 - ✓ 地域IT技術者によるRubyプログラミング講座
 - ✓ 学生の技術スキル向上のための課外活動
- 地域 OSS イベントへの学生参加
 - ✓ OSSビジネスコンテスト
 - ✓ スモウルビー甲子園
 - ✓ Ruby World Conference など

取組実績

Rubyプログラミング講座

- プログラミング言語1,2 (夏季・冬季集中講義) として開講
- Ruby言語の修得 & オブジェクト指向の基礎的理解



課外活動：Ruby クラブ

- 学生に対する実践的プログラミング演習
 - ✓ スモウルビー開発を題材として
 - ✓ 開発者の高尾宏治氏 (高専OB)の指導
- 小中学生向けプログラミング講座の開催支援
 - ✓ 教材開発
 - ✓ 高専生がメンターとして活躍
- Ruby on Rails 講座



地域 OSS イベントへの学生参加

- 創造工学演習 (2,3年生)
 - ✓ グループ学習, アイデア出し+プレゼン
 - OSSビジネスコンテスト
 - スモウルビー甲子園
- OSSリテラシ3 (4年生)
 - ✓ OSSコミュニティに対する理解
 - Ruby World Conference への参加



共同研究の例(2019年度~)

• mruby/c を活用した授業実践

- 杉山ほか：高専の授業にmruby/cを取り入れたプロジェクトの始動と将来展望 (RubyWorldConference 2019)

共同研究の内容

- mruby/c を高専の新規開講授業(2020年度)への組み込み
 - ✓ ESP32 マイコンの利用を想定
 - ✓ 松山高専情報工学科 5 年生「組込システム演習」(2コマ 180 分 × 16 回)
- ESP32マイコン用mruby/c ライブラリ開発

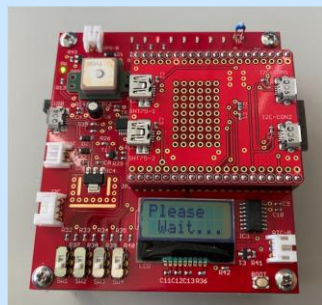
研究体制

- 松江市 まつえ産業支援センター
- 島根県 しまねソフト研究開発センター (ITOC)
- 株式会社 モンスターラボ (2019年度)
- Rubyプログラミング少年団 (2021年度~) ← 卒業生
- 株式会社 CMCソリューションズ (2021年度~) ← 卒業生

人材の
循環

取組実績

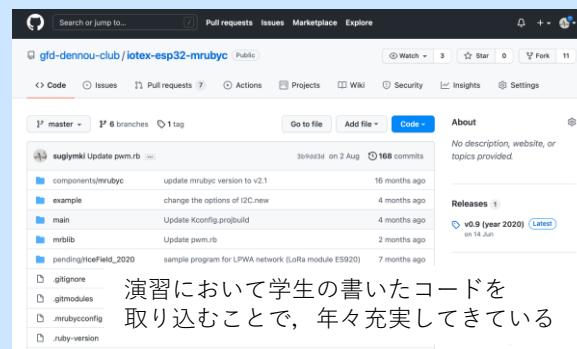
学習ボード



ピン	Pin 名
LED	GPIO13, 12, 14, 27, 26, 33, 32
スイッチ	GPIO34, 35, 18, 19
サーモスタ温度計	GPIO39 (ADC1_CH3)
圧電ブザー	GPIO15
超音波センサー	GPIO16 (TX), GPIO17 (RX) (電源: GPIO5)
LCD (AQM0802A-RN-GBW)	I2C (SDA: GPIO21, SCL: GPIO22, アドレス: 0x3E)
RTC (RC-8035SA)	I2C (SDA: GPIO21, SCL: GPIO22, アドレス: 0x32) (電源: GPIO4)
SD card	SPI (MOSI: GPIO23, MISO: GPIO19, SCK: GPIO18, CS: GPIO2)
SH75	COM1: GPIO25,26, COM2: GPIO32,33 (LED と共有)

教科書の整備

- サンプルコードも「教科書」
 - ✓ 教科書作りとしてのライブラリ開発



演習において学生の書いたコードを取り込むことで、年々充実してきている

高専の授業「組込システム演習」

- Arduino, C 言語(ESP-IDF), micropython, mruby/c の各種言語を用いたプログラミング技法の修得
- センサ利用のためのライブラリを mruby/c で書く
 - ✓ C 言語のラッパー作製, Ruby のクラス定義
 - ✓ センサのデータシートを読めるようになる
- Linux でテキストベースのプログラミング



特別講演
(地元企業さんによる事例紹介)

演習風景

共同研究の例(2019年度~)

• スモウルビー + mruby/c → 初等教育用教材

- 青笹ほか：IoTデバイス用ビジュアルプログラミングツール「SmT」の開発
～地域コミュニティと連携した人材育成の成果報告～ (RubyWorldConference 2020)

共同研究の内容

- プログラミング学習で利用可能なIoT教材の開発
 - ブロックから mruby/c のコード生成
 - クリック一つでマイコンボードにプログラムの書き込み
- 小中学生が IoT を学ぶためのツールとして活用できるかの有効性を検証

研究体制

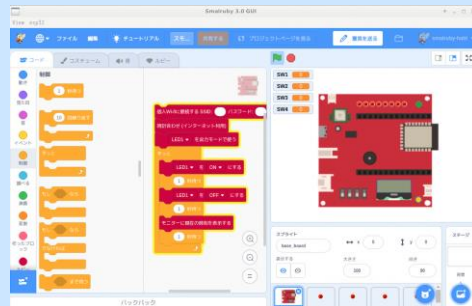
- 松江市まつえ産業支援センター
- 島根県 しまねソフト研究開発センター (ITOC)
- Rubyプログラミング少年団 ← 卒業生
- 株式会社 島根情報処理センター (2020年度~)
- 株式会社 CMCソリューションズ (2020年度~) ← 卒業生
- サイバートラスト 株式会社 (2021年度~) ← 卒業生

人材の
循環

取組実績

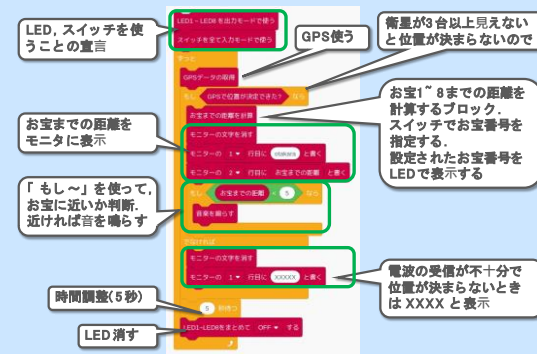
SmT : スモウルビー + 高専製の拡張機能

- ブロックとmruby/cのソースコードとの相互変換機能およびソースコードのコンパイルとRBoardへの書き込み機能を実装



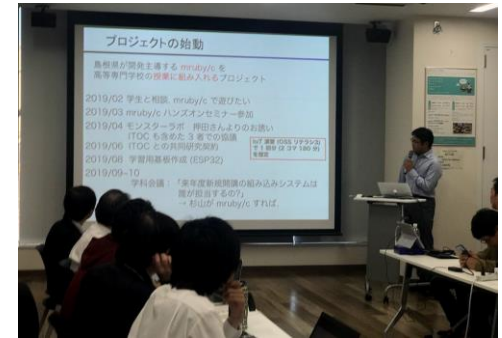
小中学生向け IoT 講座

- 換気センサ (CO₂センサ), 宝探しセンサ (GPS), などのテーマ作成
- IoT教材の有用性の検証



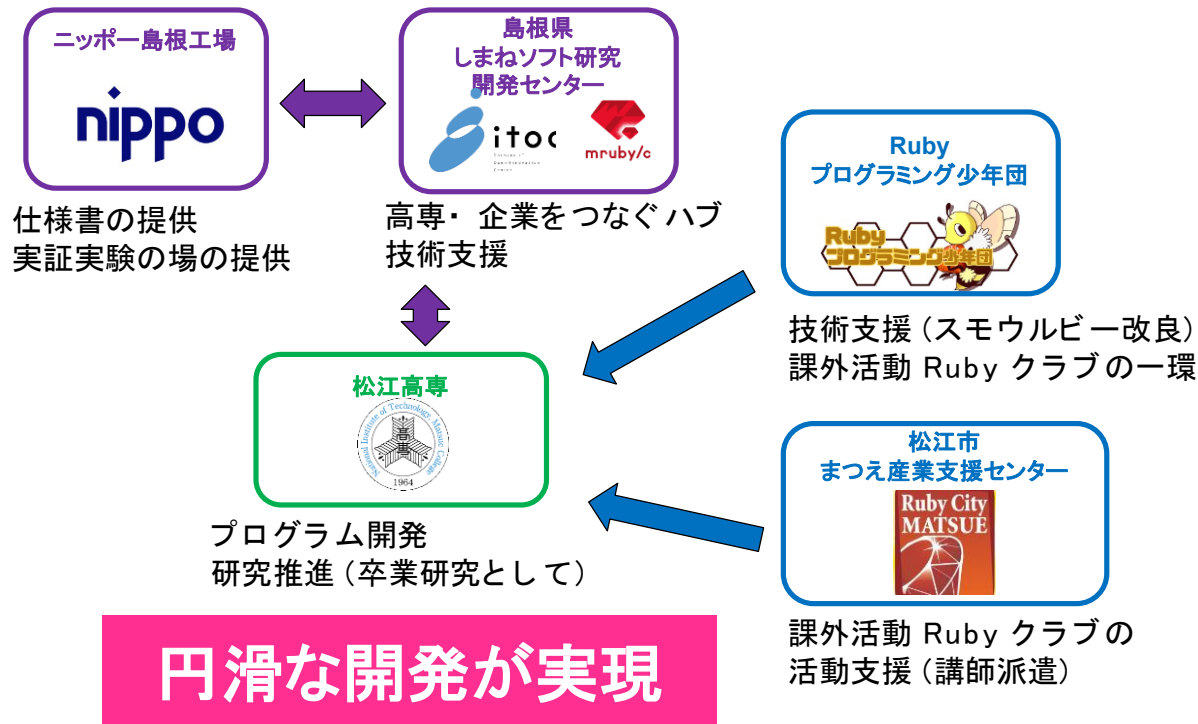
コミュニティでの新たな気づき

- ビジュアルプログラミングは初等教育だけのものか？企業現場でも使える？！
- うちのRubyコードをブロックでラップしたら…機能検査プログラムを現場でコーディングできるかも？！



mruby/c KAIGI #02 (2019)

研究体制の構築：



第2部

機能検査用自動化ツール「SmrAI」の開発と成果について

サイバートラスト株式会社
奥田彩月

自己紹介



名前: 奥田 彩月 (おくだ さつき)

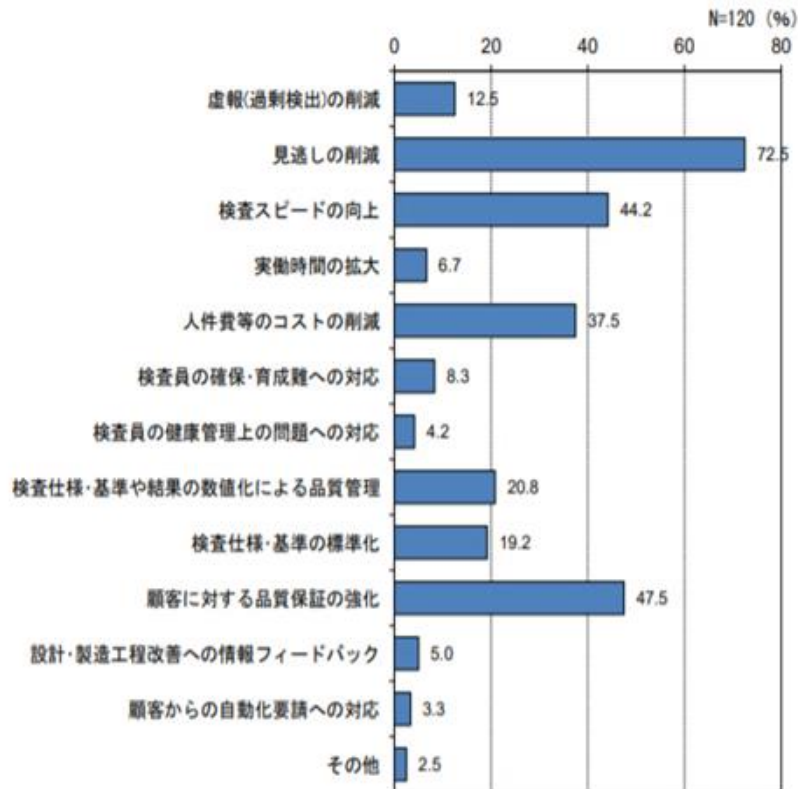
2020年4月～2021年3月 松江工業高等専門学校情報工学科 杉山研究室所属
2021年4月～ サイバートラスト株式会社で勤務

研究背景

製造業では品質検査の自動化が期待されている

図表 2.50 自動化の目的

問 21 【自動検査の実施企業が対象】自動化の目的は何ですか。(〇印はいくつでも)



主な目的

- 不良品の見落とし防止
- 顧客に対する品質保証の強化
- 検査スピードの向上
- 人件費等のコスト削減

ものづくり企業の生産現場における 検査の自動化促進可能性調査
平成28年3月 公益財団法人 ちゅうごく産業創造センター

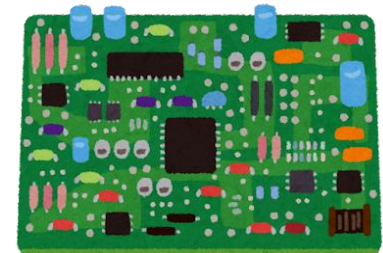
<https://bsys.hiroshima-u.ac.jp/wordpress/wp-content/uploads/2020/06/h27-2.pdf>

研究背景

多品種生産現場の課題 機能検査自動化のためのプログラミング人材育成

機能検査とは

- 検査製品の入出力端子に仕様に沿った電気信号等を印加あるいは負荷を接続し、期待される結果が得られるかを確認する。
- 製品の動作を保証しなければならない
- 自社で自動化プログラムを開発する
 - 製品の数だけ自動化プログラムが必要



多品種生産の場合

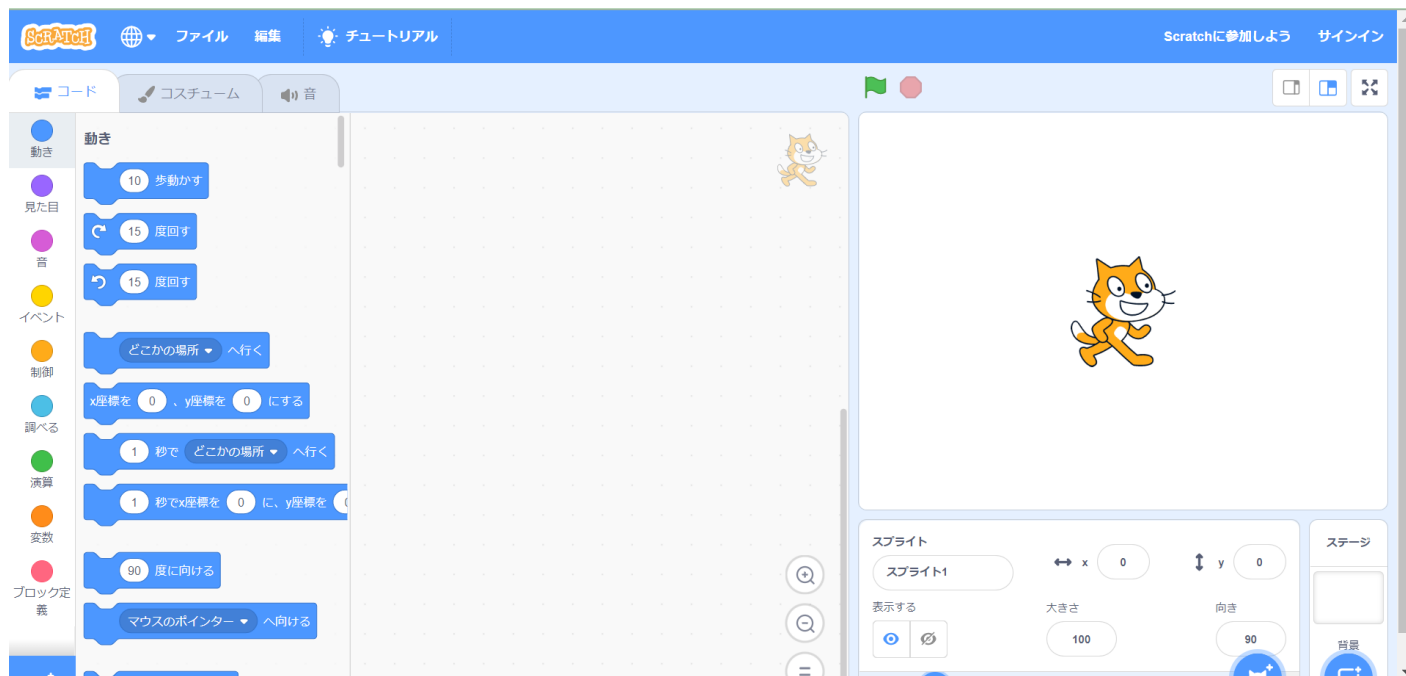
- 扱う製品数が多い
- 開発する自動化プログラムも多くなる



研究背景

ビジュアルプログラミングで機能検査自動化用プログラムを作るのはどうか？

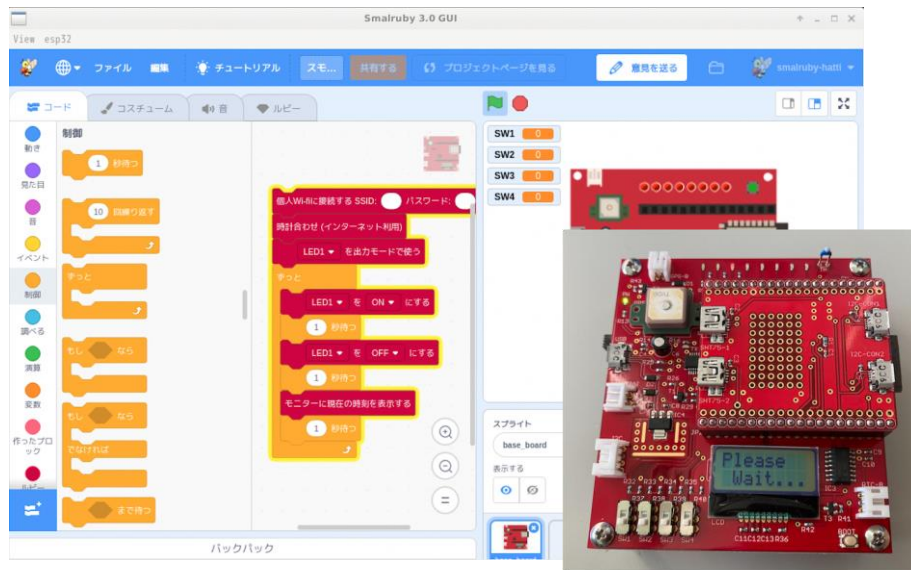
- ビジュアルプログラミングツール⇒Scratch、Smalrubyなど
- 初等教育でのプログラミング教育で採用
- 視覚的に理解しやすい
- プログラミングの基礎をすぐに身に付けられるアプローチ



研究背景

ビジュアルプログラミングツールで教育用キットの制御は可能になっている

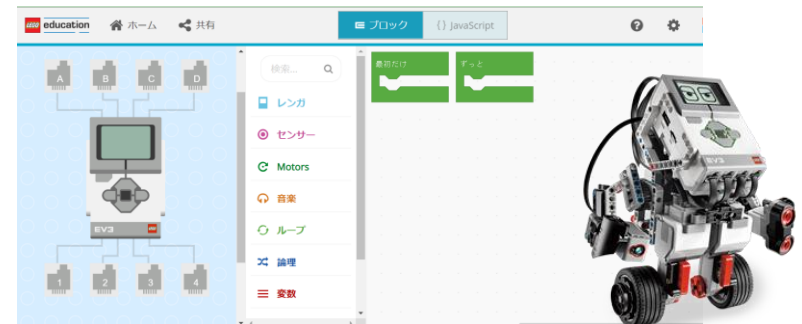
SmT(ブロック⇔mruby/c)
ESP32マイコンを制御



RubyWorldConference2020
Session-5:IoTデバイス用ビジュアルプログラミングツール「SmT」の開発
～地域コミュニティと連携した人材育成の成果報告～

<https://2020.rubyworld-conf.org/ja/program/session-5/>

LEGO MINDSTORMS
(ブロック⇔JavaScript)



micro:bit
(ブロック⇔JavaScript)

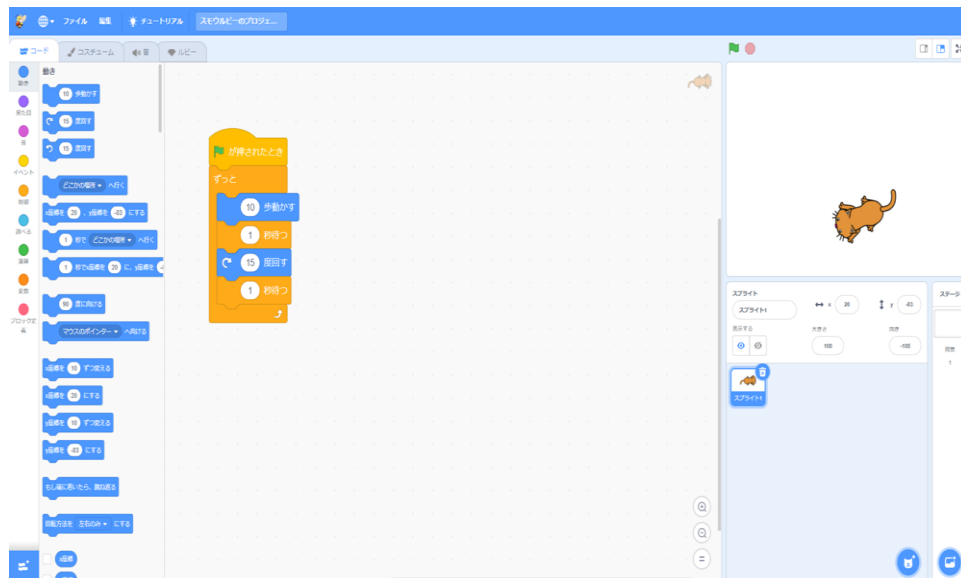


➡ 製造業の製品との連携は想定されていない

研究目的

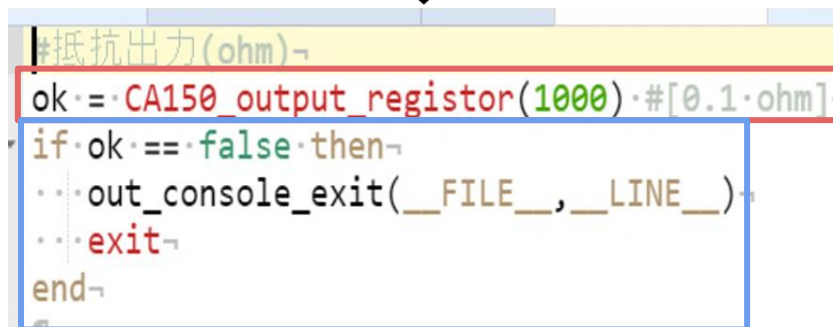
製品の機能検査に特化したビジュアルプログラミングツールが製造業におけるプログラミング人材不足の解消につながるか検証する

- Smalrubyをベースにした機能検査用ビジュアルプログラミングツールを開発する
- 社内にある既存のプログラム資産をSmalrubyのブロックでラップする
- 機能検査に精通している&プログラミング経験がない方をターゲットに実証試験



～プログラミングレベルに合わせた2種類のブロックカテゴリ～

従来の機能検査の操作順にブロックを組み立てられる



ブロックの設計

～プログラミングレベルに合わせた2種類のブロックカテゴリ～

テスター2カテゴリ

利用者自身でブロックプログラムの試行錯誤が可能



- 1ブロックにラップしているプログラム
= 操作実施
- Smalrubyのデフォルトブロックと組み合わせて
使うことを想定



```
| = CA150_output_register(1000) -
```

SmrAI (シンライ: Smalruby × Automatic Inspection)

The image displays the Smalruby 3.0 GUI and a test log window. The GUI window, titled "Smalruby 3.0 result", shows the test results for "テスト2(メモリ初期化)" and "テスト0(調整値クリア)", both marked as "OK". The test log window, titled "2021115_testlog.txt - メモ帳", shows the test results for "テストモードへ投入" and "テスト番号2を選択", both marked as "成功". The test log also includes a "finish test" message and a "test.rb:2:in `<main>': uninitialized constant TEST_NO (NameError)" message.

結果表示

テスト2(メモリ初期化) : OK
テスト0(調整値クリア) : OK
finish test
<<実行終了>>

結果の記録

2021115_testlog.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
Fri Jan 15 2021 17:21:09 GMT+0900 (日本標準時)
テストモードへ投入 : 成功
テスト番号2を選択 : 成功
成功
メモリ初期化キーを押す : 成功
テスト2の結果判定 : 失敗
RERURNキーを押す : 成功
finish test
Fri Jan 15 2021 17:23:20 GMT+0900 (日本標準時)
テストモードへ投入 : 成功
テスト番号2を選択 : 成功
成功
メモリ初期化キーを押す : 成功
2
RERURNキーを押す : 成功
finish test
Fri Jan 15 2021 17:23:33 GMT+0900 (日本標準時)
テストモードへ投入 : 成功
テスト番号2を選択 : 成功
成功
メモリ初期化キーを押す : 成功
2
RERURNキーを押す : 成功
finish test
Fri Jan 15 2021 17:23:43 GMT+0900 (日本標準時)
テストモードへ投入 : 成功
テスト番号2を選択 : 成功
成功
メモリ初期化キーを押す : 成功
2
RERURNキーを押す : 成功
finish test
Fri Jan 15 2021 17:32:32 GMT+0900 (日本標準時)
test.rb:2:in `<main>': uninitialized constant TEST_NO (NameError)

実証試験

機能検査に精通しているがプログラミング経験のない社員がSmrAIを利用して機能検査の自動化が実施できるかを確認する



【被験者】

株式会社ニッポ一島根工場の社員4名
プログラミング未経験:3名
C言語に若干触れたことがある:1名

前半:2名、後半:2名の2グループ制

【講師】

奥田

- SmrAIのチュートリアル説明
- 演習時の質問対応

株式会社ニッポ一島根工場 田部氏

- 演習時の質問対応

実証試験

【事前準備】

株式会社ニッポ一島根工場製の製品開発・検査用Rubyライブラリをブロックにラップ



```
#抵抗出力(ohm)-  
ok = CA150_output_registor(1000).#[0.1·ohm]-  
if ok == false then-  
  out_console_exit(__FILE__, __LINE__)-  
  exit-  
end-
```



```
= CA150 output registor(1000)-
```

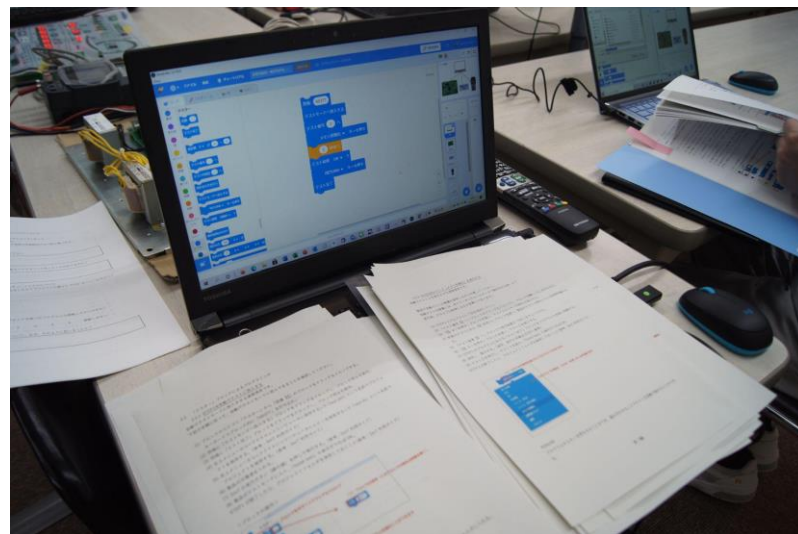
実証試験

【ゴール】

2時間でSmrAIを利用して全6項目の検査を自動化する

【実施手順】

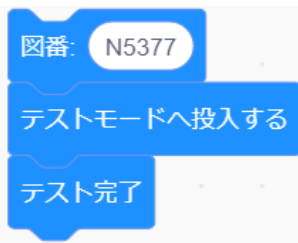
1. SmrAIのチュートリアル実施
 - ・ 講師の操作画面を映し、一緒にサンプルプログラムを作る
2. 演習用ドキュメントを参照しながら、自身でプログラム作成、検査の実施に取り組む
 - ・ 2時間という時間制限があるため「テスター」カテゴリを優先的に使う
 - ・ 演習が早く終わった場合は「テスター2」カテゴリを利用したプログラムも書く
 - ・ エラー対応や質問がある際は講師に尋ねる
3. アンケートに回答



実証試験 ～プログラム例～

各STEPでは、前STEPで作成したブロックにつなげる形でブロックを組み立てる
全STEPを終了することで、6項目すべての検査を実施する1つのプログラムが完成する

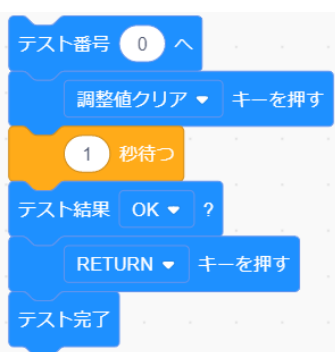
STEP1 テストモードへの投入



STEP2 メモリ初期化



STEP3 調整値クリア



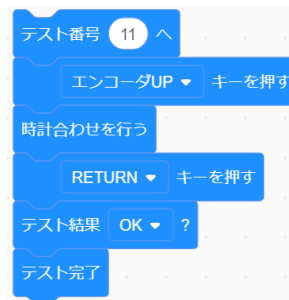
STEP4 センサ調整



STEP5 計時チェック



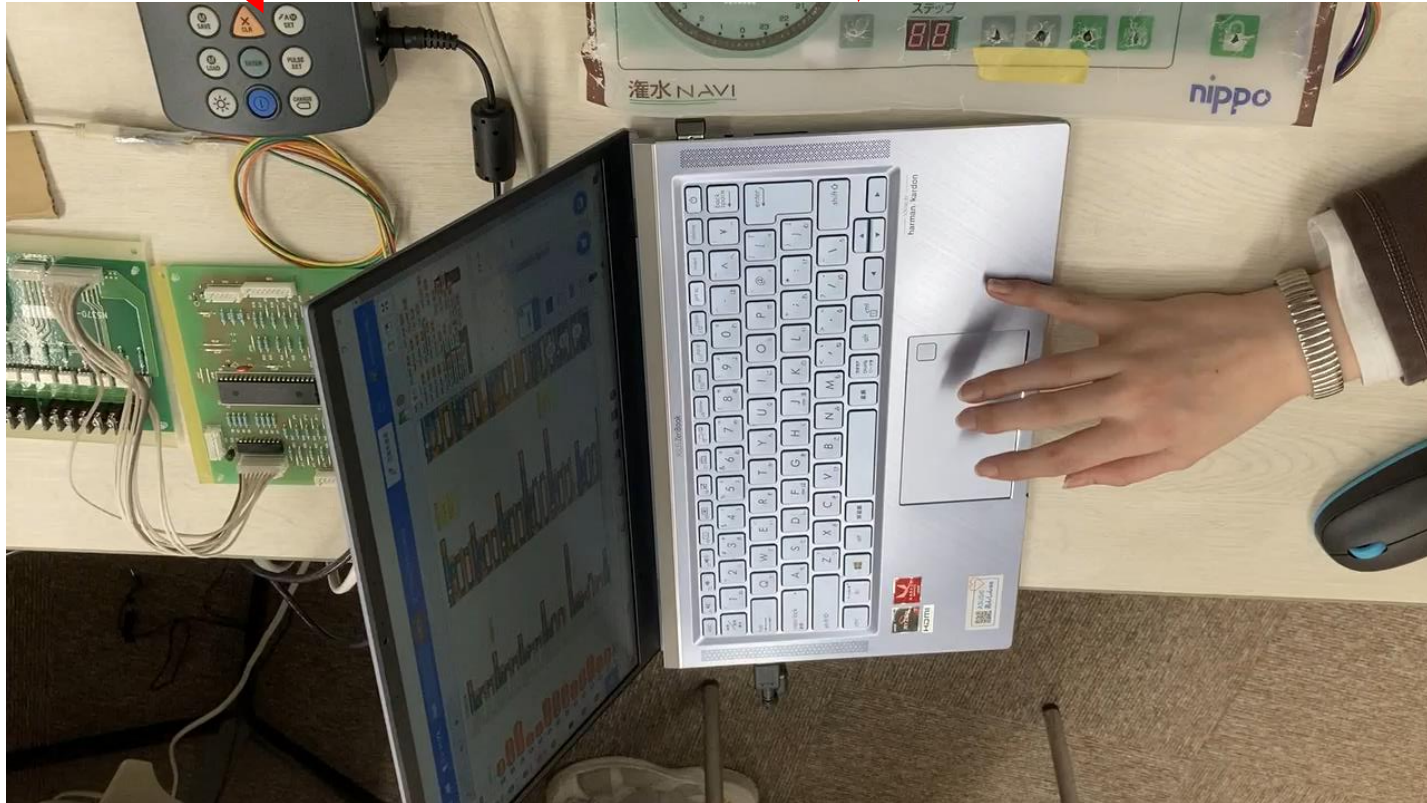
STEP6 時計合わせ



SmrAIで自動化した機能検査の様子

疑似負荷出力装置

検査対象



SmrAIを利用しているPC

試験結果

被験者4名とも機能検査自動化プログラムを完成させることが出来た

		STEP1 テストモードへの投入	STEP2 メモリ初期化	STEP3 調整値クリア	STEP4 センサ調整	STEP5 計時チェック	STEP6 時計合わせ
被験者A	「テスター」 カテゴリ	○	○	○	○	○	○
被験者B	「テスター」 カテゴリ	○	○	○	○	○	○
被験者C	「テスター」 カテゴリ	○	○	○	○	○	○
	「テスター2」 カテゴリ	○	○				
被験者D	「テスター」 カテゴリ	○	○	○	○	○	○
	「テスター2」 カテゴリ	○	○				

演習内容は実際の現場で行われる機能検査のテスト項目を実施するもの
その演習内容を完遂することが出来た



現場で求められるレベルのプログラムを書くことが出来た

アンケート結果

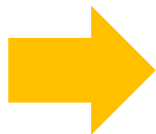
Q. SmrAIは自動化ツールとして業務利用に適していると思いますか？

	5(良)	4	3	2	1(悪)
業務利用に適している	☆	★★★			

☆:C言語に少し触れたことがある方 ★:プログラミング未経験の方

被験者の方からの感想

- テストモードを自動化できるだけで大幅に工数が減らせるから
- 別製品で導入している自動化ソフトよりも動作が早かった



プログラミング未経験の方が, SmrAIを使うことで
業務でプログラミングができたと思えた

アンケート結果

Q.「テスター」カテゴリは使いやすいですか？



	5(良)	4	3	2	1(悪)
理解しやすい	☆☆	★	★		
簡単に書ける	☆	★★	★		
自分でもプログラムが作れそう	☆	★★	★		

☆:C言語に少し触れたことがある方 ★:プログラミング未経験の方

被験者の方からの感想

- ブロックごとにプログラムが出来ており、並べるだけで完成するから
- 視覚的に見やすかったから
- 普段の業務の流れの通りにブロックを並べるだけで完成するから



従来の検査時と同様の操作順にブロックを並べるだけでプログラムが作れる
「テスター」カテゴリのメリットが生かせている

アンケート結果

Q.「テスト2」カテゴリは使いやすいですか？

抵抗出力

100

0

Ωを加える 処理結果保存用変数：



	5(良)	4	3	2	1(悪)
理解しやすい	☆			★	
簡単に書ける		☆		★	
自分でもプログラムが作れそう		☆		★	

☆:C言語に少し触れたことがある方 ★:プログラミング未経験の方

被験者の方からの感想

- 検査結果やエラー処理の表示を自分で分かりやすくできる
- ブロックが自由に組めてデバッグがしやすい
- 使用ブロック数が多くて煩雑だった

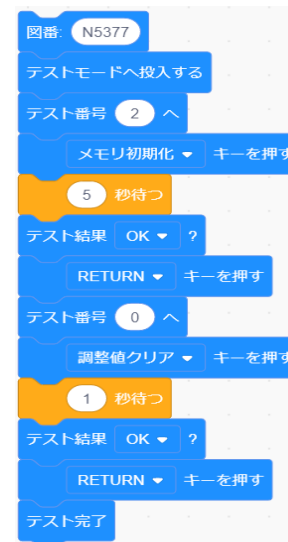


利用者自身でブロックプログラムの試行錯誤が可能
「テスト2」カテゴリのメリットが生かしている

考察

なぜ、プログラミング未経験でも現場で求められるレベルの機能検査自動化プログラムが書けたのか？

1. 既存の製品開発・検査用Rubyライブラリをブロックでラップしたから
2. 機能検査に精通している社員の経験が活かせるようにブロックを設計したから
 - 1ブロック=1操作
 - 操作順のとおりブロックを並べるだけでプログラムが完成する



まとめ

【目的】

製品の機能検査に特化したビジュアルプログラミングツールが製造業におけるプログラミング人材不足の解消につながるのか？

【手法】

- SmrAI を開発
- 社内にある既存のプログラム資産をSmalrubyのブロックでラップする
- 機能検査に精通している＆プログラミング経験がない方をターゲットに実証試験



【結果】

SmrAIはプログラミング経験がない社員がプログラミング人材として評価され、プログラミング人材の掘り起こしにつながる可能性を示した。

将来展望

将来的なプログラミング人材の増加、多様性

小中学校でビジュアルプログラミングツールを用いたプログラミング教育を受ける



製造業に従事する時にもビジュアルプログラミングツールによるプログラミング技術が生かせる



- 製造業のプログラミング人材が増える
- ビジュアルプログラミングでも業務が出来る

株式会社ニッポ一島根工場の実運用に向けた取り組み

- 運用時の課題確認
- ブロックの種類の増加
- 動作検証