

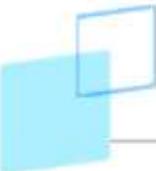


組込みシステム基本的な構造と活用方法の研究 ～Arduinoを活用したRCカーの作成～

平成25年 2月

平成24年度 OISA技術研究会

組込みシステム部会 研究成果発表



部会員およびアドバイザー技術委員

【部会員】

(順不同)

岡野 真大(部長)

(株)エイビス

西村 将利(副部長)

大分大学大学院

伊東 和浩

大分大学大学院

堤 和利

(株)オーガス

【アドバイザー・技術委員】

清水 太

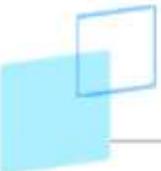
(株)ワイズ・システムズ

三宮 由裕

三井造船システム(株)

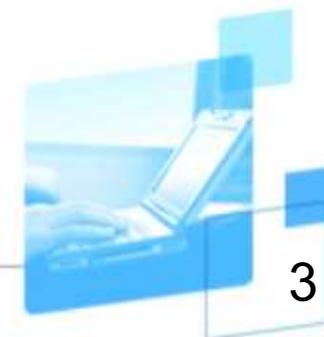
赤星 哲也

日本文理大学



目次

1. 組み込みシステムとは
2. Arduinoの概要
3. RC(Radio Control)カーの作成
4. 動作確認
5. 発展・まとめ
6. 組み込みシステムの今後の課題



組み込みシステムとは



～組み込みシステムって何～

組み込みシステムとは

組み込みシステム

産業機器や家電製品などに内蔵される特定の機能を実現するための
コンピュータシステム

パソコン・サーバ・メインフレーム以外でコンピュータ制御されている機器
(機器の中に組み込まれているプログラムを組み込みソフトウェアという)

身の回りの機械には何らかの組み込みシステムが搭載！



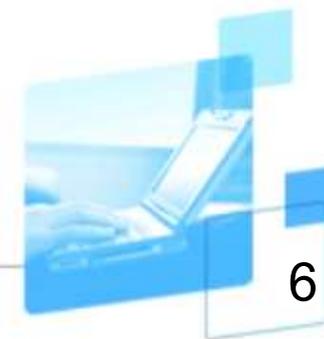
デジタル機器・・・携帯電話, ゲーム...
家電・・・テレビ, エアコン...
インフラ・・・信号機, 基地局...
乗り物・・・電車, 自動車...
etc...



～組み込みシステムの抱える問題～

組み込みシステムとは

- ・ 組み込みシステム技術の**ブラックボックス化**
 - 組み込みシステムはマイコンを中心としている
ソフトウェア側で柔軟に機能が変更できる反面、いくらでも複雑にできる
 - 機器を「**見て触る**」機会の減少
ケースを開けない、変更を加えられない、回路図が付属していないetc
- ・ ユーザの**ニーズの細分化**
 - 1つのプロダクトで多くの人々のニーズに応えるのは不可能
 - 最大公約数を目指せば中途半端に、最小公倍数を目指すと
とんでもなく複雑なものになり開発コストがかかる



パーソナルファブリケーション

「自分たちの欲しいものを自分たちで作る」という考え

大量生産・大量流通・大量消費を前提としたものづくり
個人の欲しい物に近い製品を様々な製品から選択

自分の欲しい物を自分で作る
自分の欲しいワンオフの製品を作る
同じようなものを欲しいと思ってる人たちに
欲しい数だけ販売

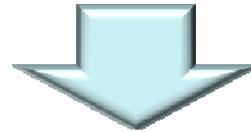
～どうやって作る？～

組み込みシステムとは

ソフトウェアと**ハードウェア**を組み合わせてのものづくり

プログラミングや電子工作の経験を持たない人にとっては難しい・・・

「電子工作＝はんだ付け」というイメージがあり大変そう・・・



「Arduino」や「Gainer」などのツールキットの登場

はんだ付けをすることなく電子回路を組み立てられる

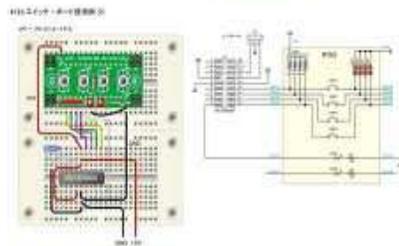
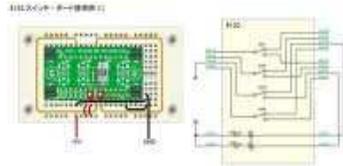
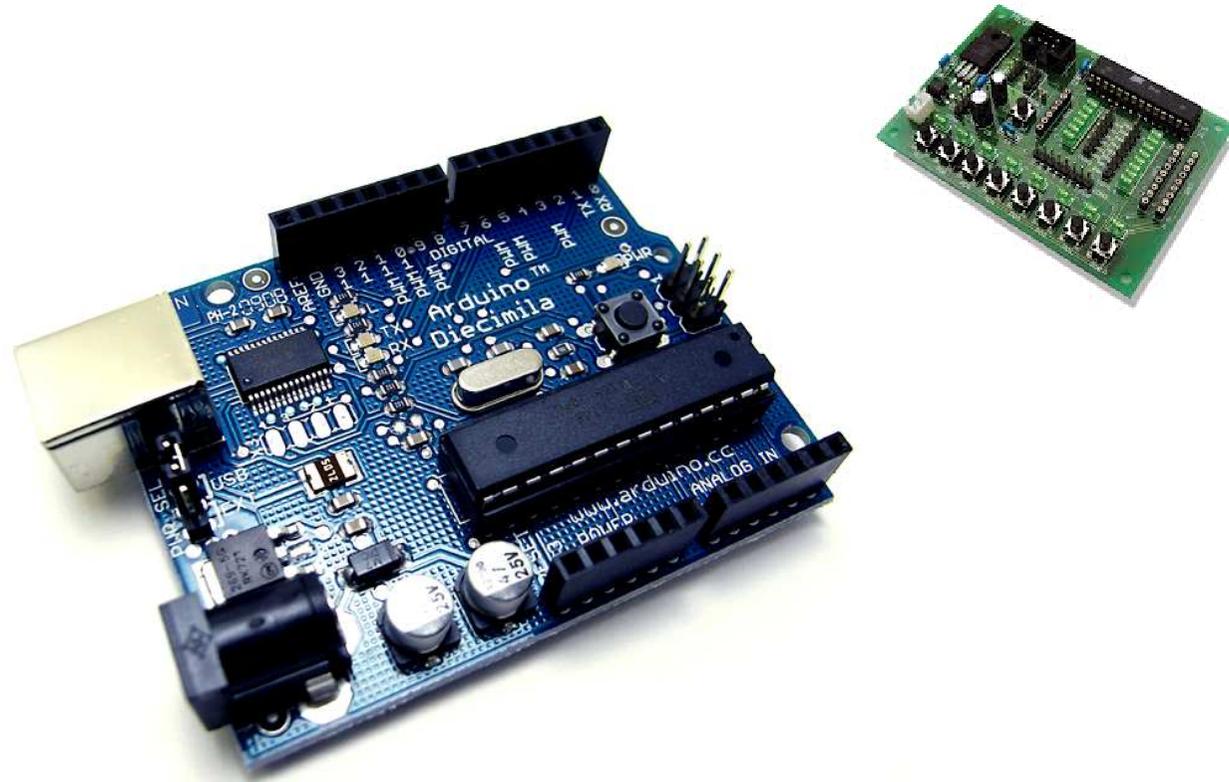
ブレッドボードを使うことで失敗しても簡単に組みなおし可能

自分の手で触れて**Try & Error**や**Tinkering**を試してみる！（いじくりまわす）

ツールキットを使いソフトウェアとハードウェアの世界をつなぎ

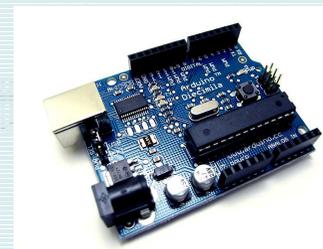
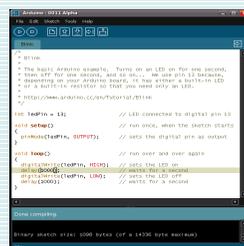
自分の欲しい物を作成する！

Arduinoの概要



【Arduino】

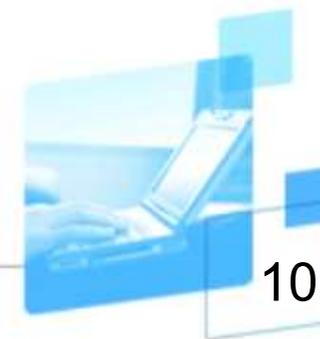
- ・Arduino言語
- ・開発環境 (ArduinoIDE)
- ・マイコンボード
- ・ウェブサイト
- ・ワークショップ
- ・コミュニティ



- ・柔軟で使いやすいハードウェアとソフトウェアを使ったオープンソースの電子プロトタイピング・プラットフォーム
- ・アーティスト・デザイナー・ホビイスト・そしてインタラクティブな物や環境を作りたい人に

例えばこんな利用例・・・

- ・地面に絵を描いたり雪の中を走り回るロボットを作りたい
- ・縫いぐるみの一部を光らせたい
- ・コーヒーが入ったらツイートするコーヒーポットを作りたい
- ・自転車に乗っている間の心拍数をモニターしたい



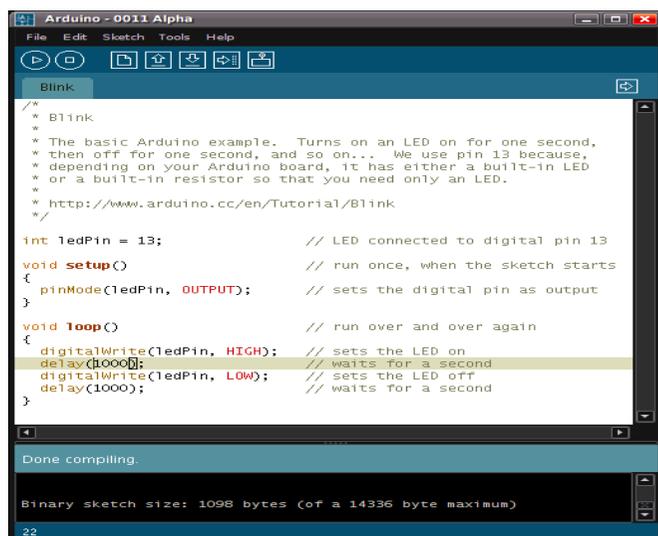
- OSに依存しないIDE・ドライバー
 - Mac, Win, Linuxで動作
 - Processingベースの言語で支援体制, 移植性に優れる, バグ発見が容易
- 豊富なライブラリ
 - SDカードへの書き込み, 液晶画面表示, データ解析など複雑な処理も
- センサーでの拡張
 - アナログ-デジタル入力を備え, 市販のセンサからデータの抽出等が可能
- 低価格
- オープンソース
 - 特殊用途の派生品をライセンスなしで作成・販売可能
- 大規模のコミュニティ
 - 無数の専門的なシールドやアクセサリ, コード例が存在

初心者は山ほどある使用例を見て構想を深めることができ慣れてくると自身のものづくりに必要なものを選定していく



- Arduinoの開発はArduinoボードとArduino IDEからなる
 - Arduinoボード … マイコンが載ったハードウェア
 - Arduino IDE … PC上で動作するソフトウェア

Arduino IDEでスケッチを書く



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/* Blink
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13
void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

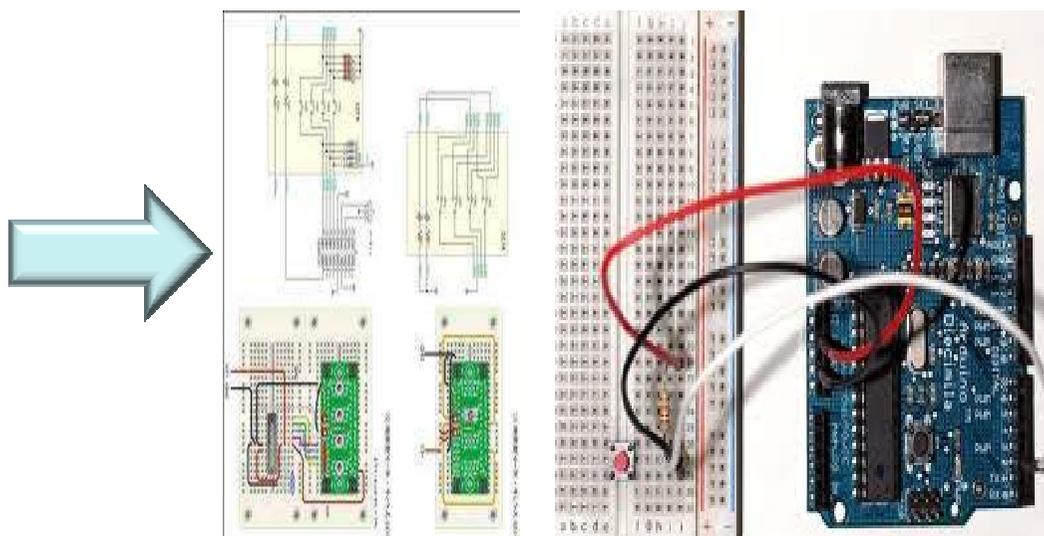
void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);               // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);               // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22
```

開発環境画面

回路を組みArduinoボードにアップロード！



回路作成

～入門 LED点灯～

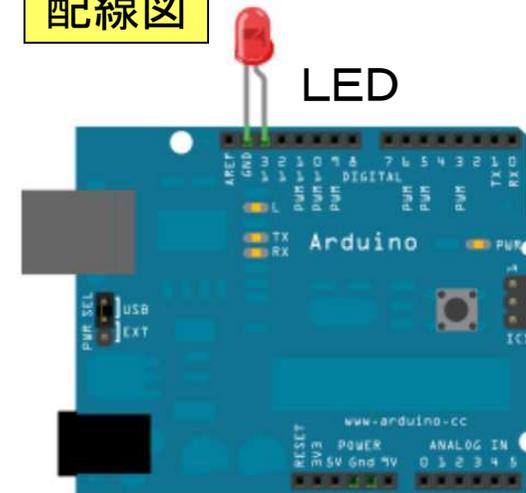
- LEDを1秒毎に点灯/消灯させてみる

スケッチ

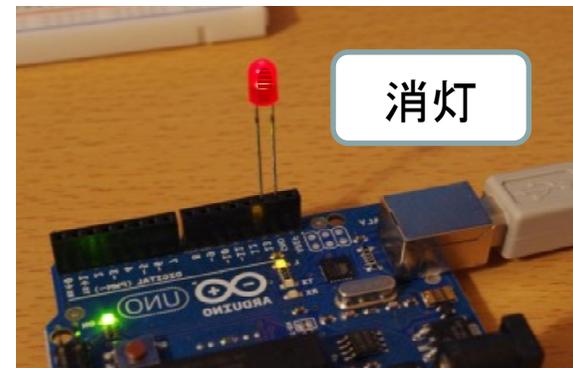
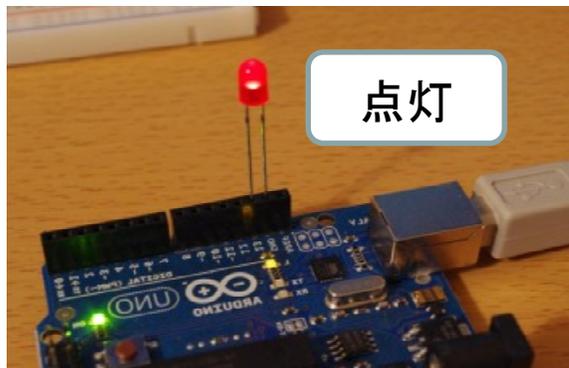
```
1 void setup() {  
2   pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(13, HIGH); // LED on  
7   delay(1000);  
8   digitalWrite(13, LOW); // LED off  
9   delay(1000);  
10 }
```

アップロード

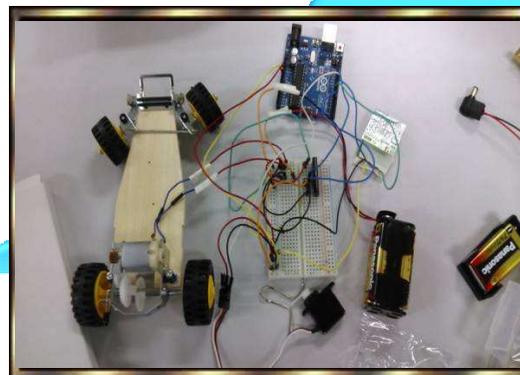
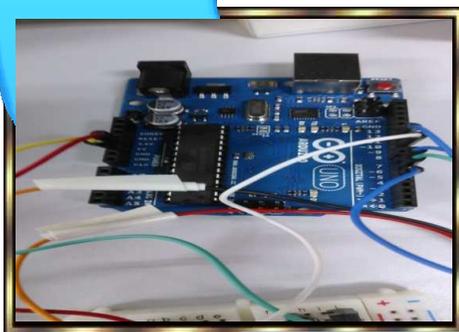
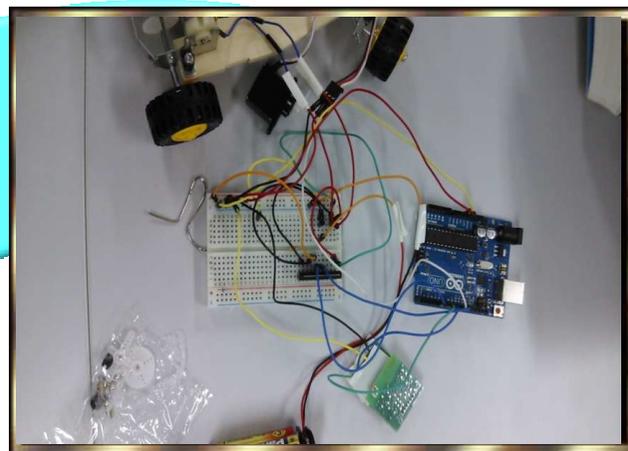
配線図



実行例



～RCカーの作成～

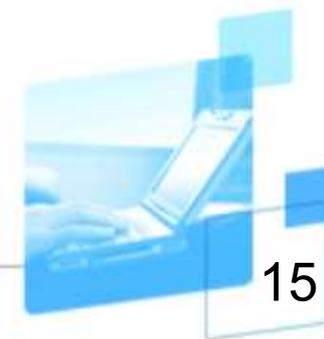


➤ 製作背景

- ・メンバー全員で何かArduinoを使用して形に残るものを製作したい…
- ・メンバー全員組み込みシステムや回路・配線等を触るのは初めて
→「動作」したことが目で見えるためわかりやすく、基礎的な配線技術や制御技術を学べるRCカーを作成！！

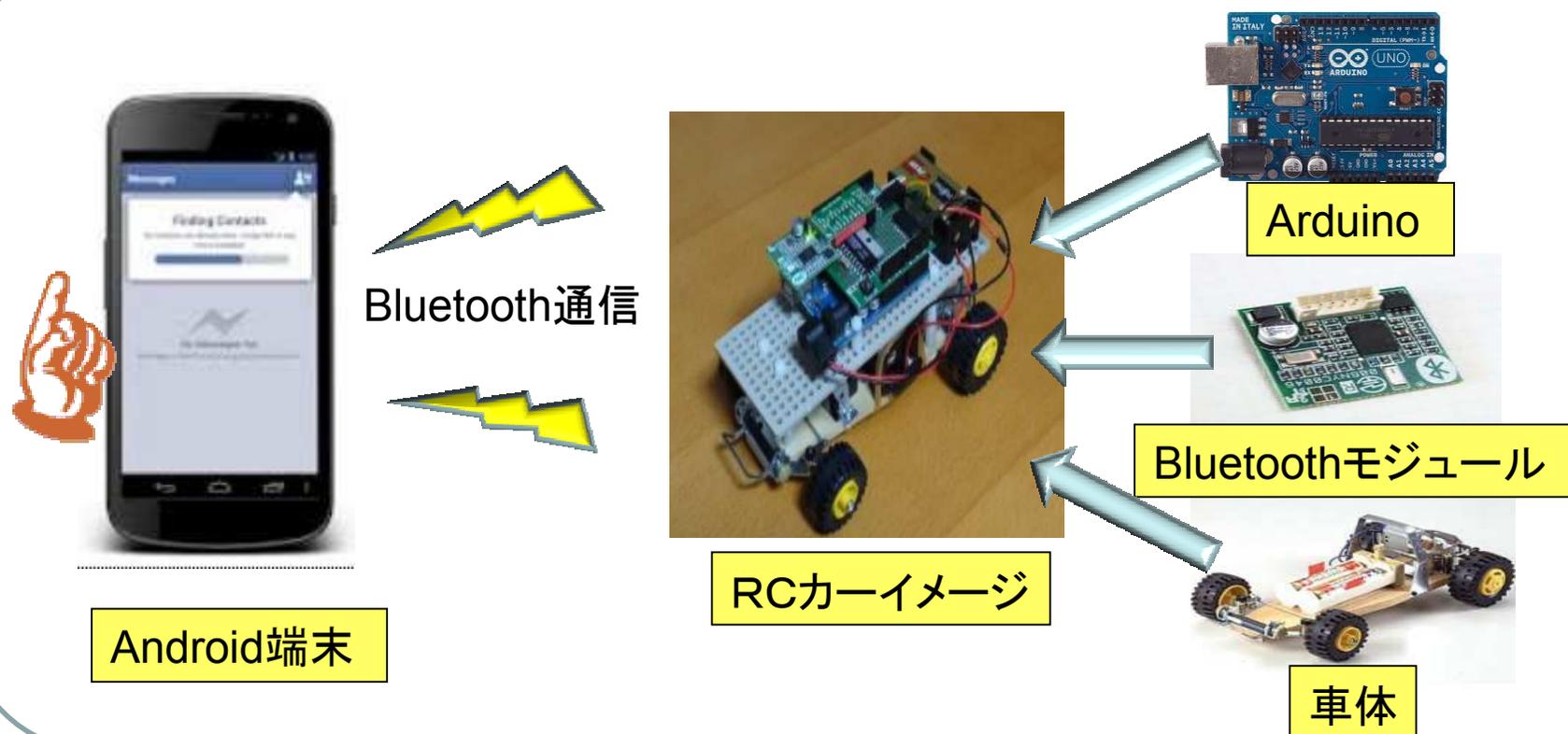
➤ 設計仕様

- ・RCカーの機能の中で最低限のものを実装
→前進・後進(スロットル)と左折・右折(ステアリング)操作
- ・遠隔操作のためにAndroidアプリでBluetooth通信を行う
- ・時間があればLEDや各種センサによる追加機能の検討
→ウインカーや障害物回避etc..



- 下記サイトを参考にArduino + Bluetoothを使ったRCカーを作成する
http://www.hiramine.com/physicalcomputing/rccar_arduino_bluetooth_android/index.html

概要イメージ図

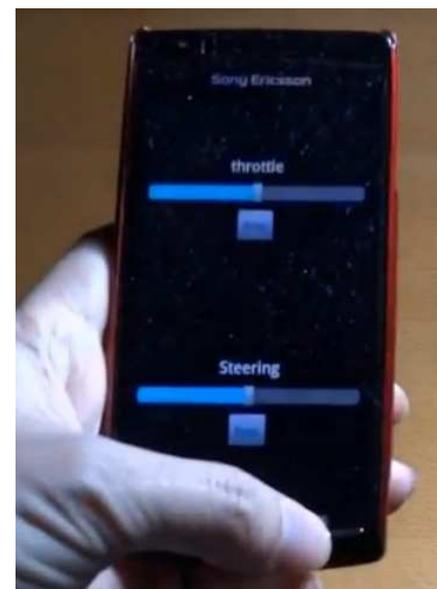


➤ ハード仕様

- モータドライバとモータで駆動・制御を行う
- 操舵用サーボモータと前輪車軸を同期させることで操舵を行う
- BluetoothモジュールとArduinoを組み合わせてAndroid側で操作する

➤ ソフト仕様

- Android画面に映されたバーで操作
 - ・スロットル操作(前進・後進)
 - ・ステアリング操作(右折・左折)
- 送信側からのBluetooth通信を受信側で読み取り解析し動作させる



操作イメージ

～準備物と役割～

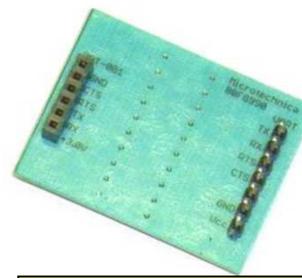
- ✓ Arduino ボード
 - ✓ 車体
 - ✓ サーボモータ
 - ✓ モータードライバ
 - ✓ Bluetoothモジュール
 - ✓ RBT-001用拡張アダプタ
 - ✓ ブレッドボード
 - ✓ ジャンパーワイヤやプラグなど
 - ✓ Android端末
- … ハードウェア側の制御
 - … RCカーの基礎
 - … ステアリング調整
 - … モーターの動作を調整
 - … Androidとの通信を行う
 - … 電圧調整
 - … 回路基盤
 - … 回路組み立て
 - … RCカーを操作



モータードライバ



サーボモータ



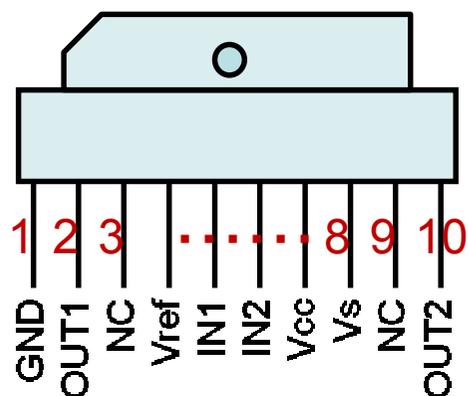
拡張アダプタ



～ハードウェア側実装・配線(1)～

- モータ制御回路の配線

モータドライバ(TA7291P)



対応表

Vcc	論理回路電源(4.5~20[V])
Vs	モータ電源0~20[V]
Vref	モータ電圧制御電源
GND	基準電圧0[V]
IN1	入力1
IN2	入力2
OUT1	モータ出力1
OUT2	モータ出力2

外部電源端子

Arduino

- ・GND
- ・Digital 11
- ・Digital 5
- ・5V

Arduino

モータ

- ・モータ出力1
- ・モータ出力2

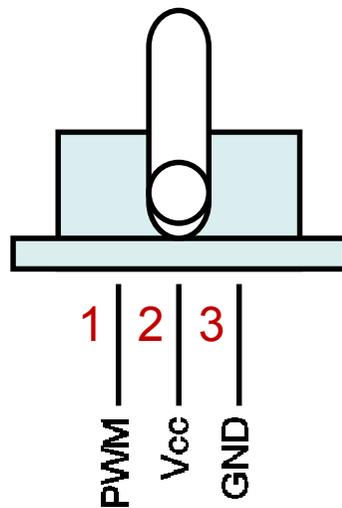
モータ



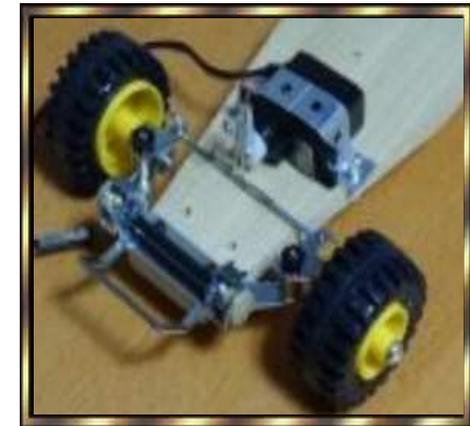
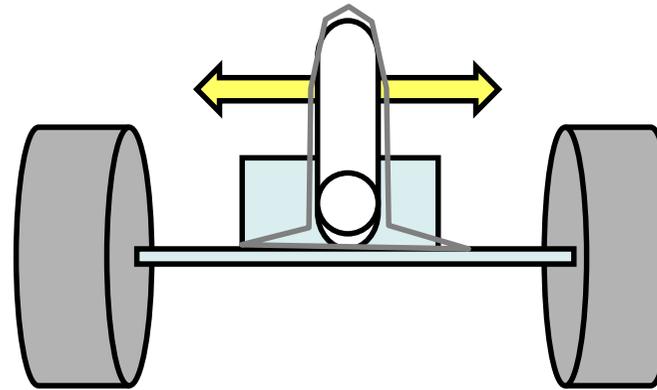
～ハードウェア側実装・配線(2)～

- サーボモータの配線

サーボモータ



前輪車軸とサーボモータの動きを同期させて進行方向を変える

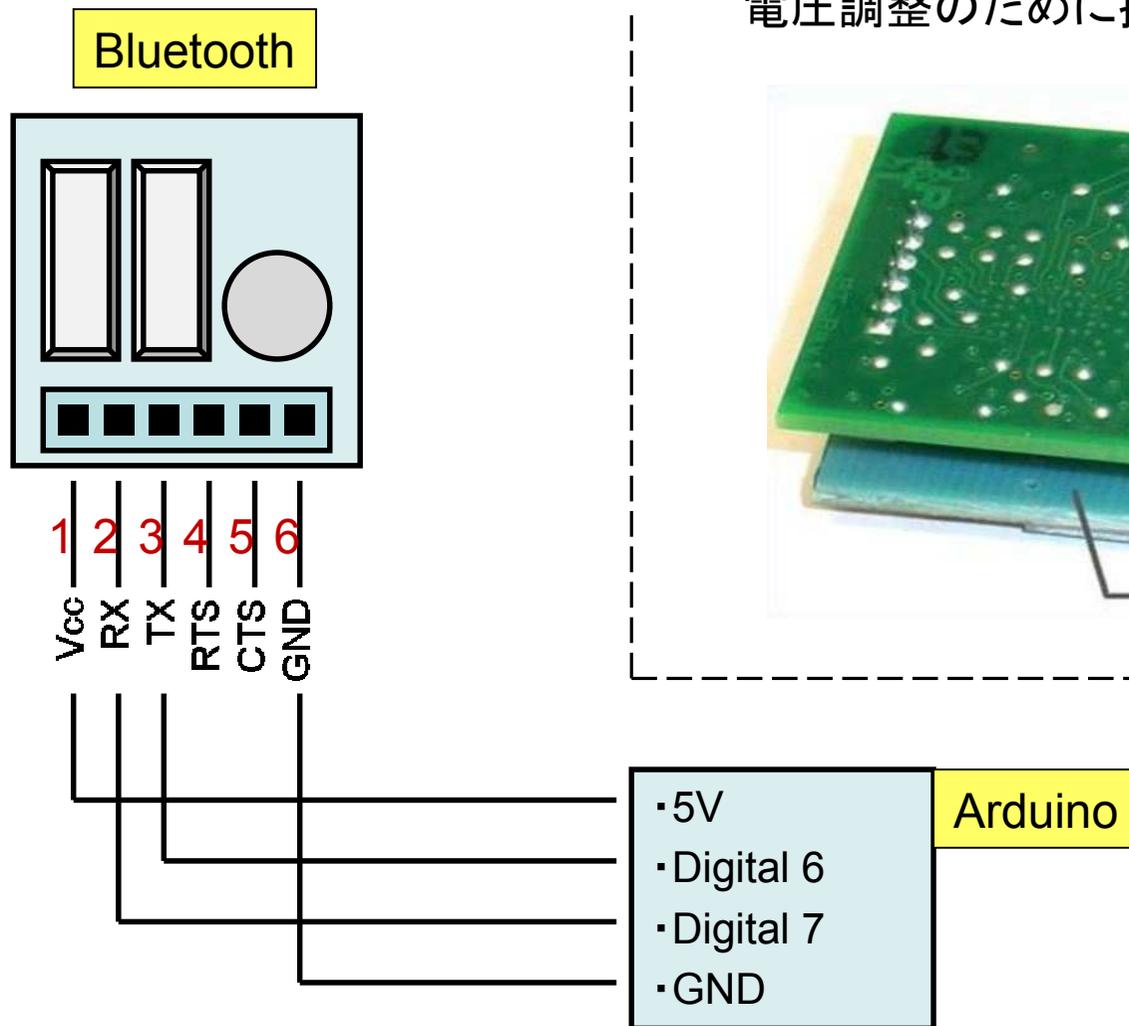


・Digital 9
・5V
・GND

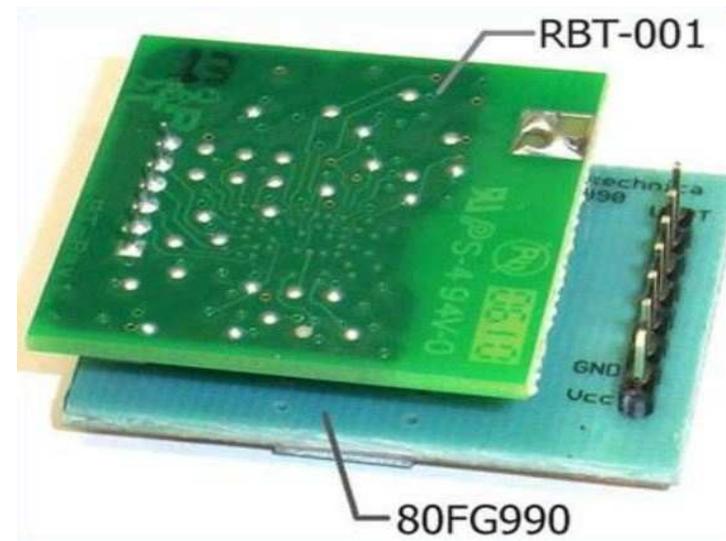
Arduino

～ハードウェア側実装・配線(3)～

- Bluetooth受信回路の配線



電圧調整のために拡張アダプタを使用

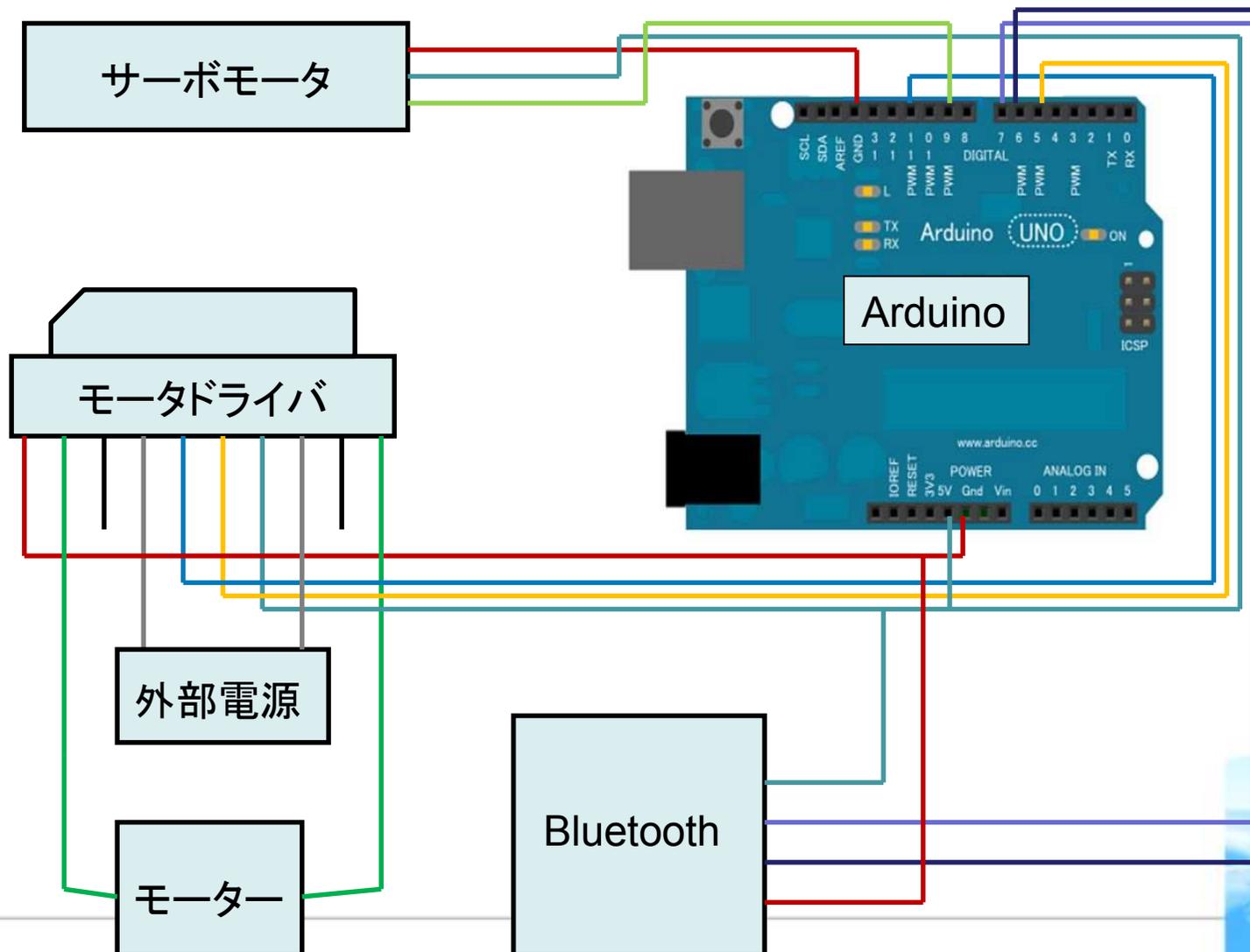


～ハードウェア側実装・配線(4)～

RCカーの作成

- (1)～(3)をまとめる

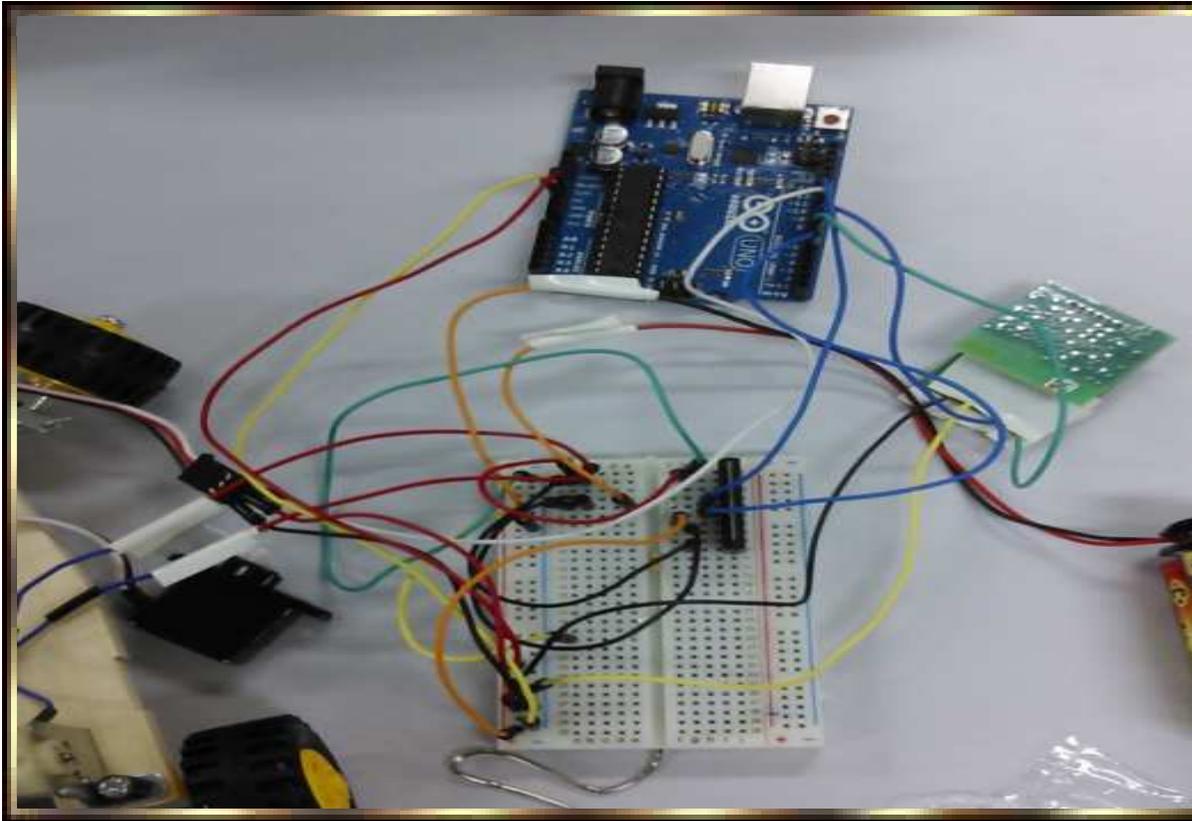
配線図



～ハードウェア側実装・配線(5)～

RCカーの作成

- ・ 実際のArduinoとブレッドボードを使って配線！！

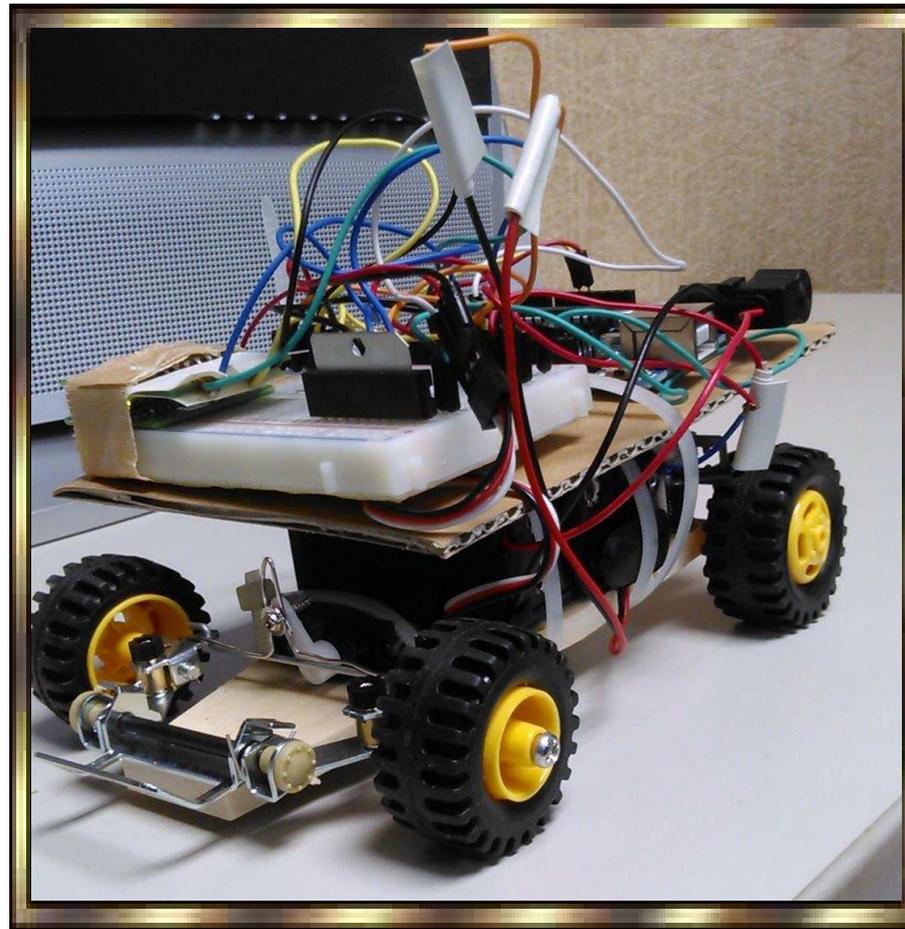


ブレッドボード上にジャンプワイヤを大量に差し込んだため少し煩雑・・・
→本格的な配線をするためには「はんだ付け」があったほうが良い？

～ハードウェア側実装・車体～

RCカーの作成

- ・ 配線が完成したので車体を組み立てる



～ソフトウェア側実装・受信側～

- ・ Loop関数(メイン関数)から処理の流れを追う

```
void loop(){
  static int s_iMotor = 0;
  static int s_iAngle = 90;
  char szLineString[READBUFFERSIZE];
  if( !ReadLineString( g_bluetooth, g_szReadBuffer, READBUFFERSIZE,
                      g_iIndexChar, szLineString, READBUFFERSIZE ) ){
    return;
  }
  boolean bMotor = false;
  boolean bAngle = false;
  int iMotor = 0;
  int iAngle = 0;
  if( !AnalyzeLineString( szLineString, bMotor, iMotor, bAngle, iAngle ) ){
    return;
  }
  if( bMotor ) {
    s_iMotor = iMotor;
  }
  MotorDrive( MOTOR_IN1_PIN, MOTOR_IN2_PIN, s_iMotor );
  if( bAngle ) {
    s_iAngle = iAngle;
  }
  g_servo.write( s_iAngle );
}
```

Bluetoothからの
通信を読み取り

通信内容の解析

スロットル

ステアリング

～ソフトウェア側実装・送信側～

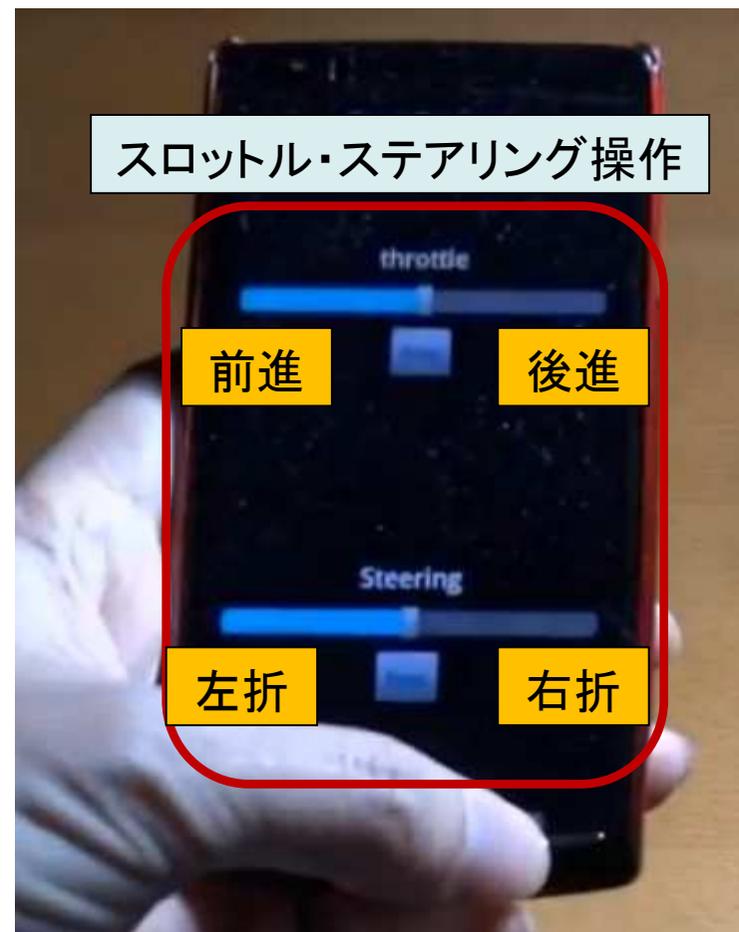
RCカーの作成

- Android端末からスロットルとステアリング操作を行う



Bluetooth端末の選択

起動時



スロットル・ステアリング操作

前進

後進

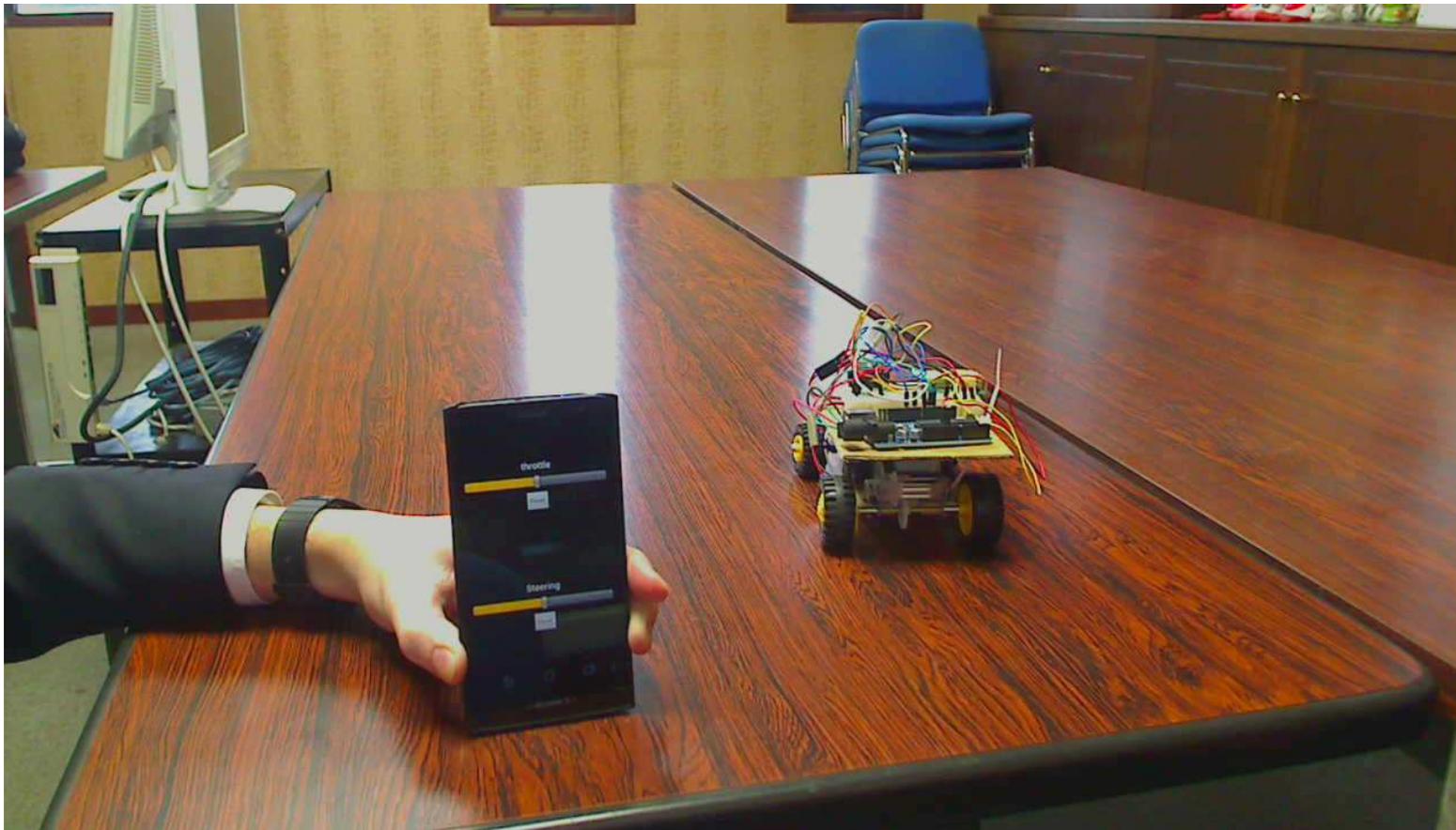
左折

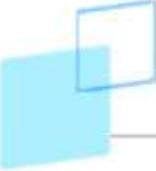
右折

操作時

～動作確認～

実際の走行動画





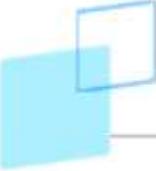
～発展～

時間や予算面での都合上実装できなかった機能

- 超音波センサや赤外線センサを使った距離計測
→自律走行や障害物回避が可能なRCカーがつかれる？
- LEDを使った応用
→Android側にLEDボタンを追加、またはステアリング操作と同期して点灯することで指示器機能の代わりに使えないか？

各種センサに関する知識や、もう少し応用の効いた
回路や配線技術を学ぶ必要がある！！





～まとめ～

➤ 組込みシステム

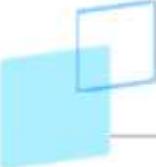
- 直接触れたことのなかった組込みシステムを調べることで、組込みシステムの現状や実装方法を知ることができた
- 回路作りや配線作業といった敷居が高く感じる作業をツールキット「Arduino」とブレッドボードの使用で作業負担を減らすことができた

➤ Arduino

- 開発環境が整っており従来のツールキットに比べ安価で互換性に優れる
また、コミュニティが活発なので自分の作成したいものに似た実装をすぐに見つけることができ、そこから改良を加え新しいものを作ることが可能

➤ RCカー

- 配線や回路などの基礎技術や制御技術を養うことができ良い経験となった
- Bluetooth通信やAndroidアプリといった要素技術の習得にもつながった



～組み込みシステムの今後の課題～

➤ 品質の向上

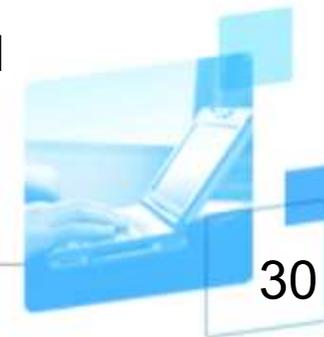
重機のブレーキ制御のように組み込みシステムは人命に直接関わるものがある
激しい運動や炎天下、極寒の中で正常に動作する必要がある
→あらゆる状況を想定し高い信頼性と安定性を築かなければならない

➤ 開発コストの低減

家電製品のライフサイクルが短くなる一方、開発期間も短くなりつつある
→より無駄を無くし、コストを削減しなければならない

➤ 人材不足の解消

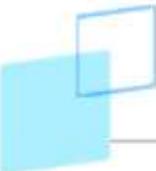
冒頭で述べた組み込みシステムのブラックボックス化・ニーズの細分化
また、近年の情報処理系の進歩により必要なプログラム量が増加
→若いうちから組み込み技術に触る機会を大切にした上で、
よりハイレベルな技術者の育成が望まれている





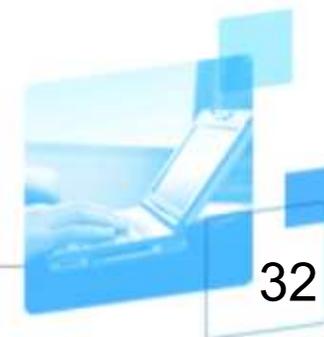
ご清聴有難う御座いました。

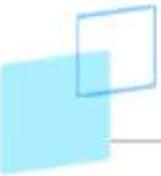




～謝辞～

本部会を実施するにあたり、多くの方にご協力をいただきました。
心より感謝の意を表します。
また、開発キットの購入に際し、大分県情報サービス
産業協会殿より資金のご援助を頂きましたことに、
お礼申し上げます。





～参考文献～

- RCカーを作る（バギー工作基本セット + Arduino + Bluetooth + Android スマートフォン）
http://www.hiramine.com/physicalcomputing/rccar_arduino_bluetooth_android/index.html
- Android + Arduino でラジコンカー作った
<http://my.opera.com/crckyl/blog/2011/08/07/android-arduino-rc-car>
- 小林茂, Prototyping Lab「作りながら考える」ためのArduino実践レシピ
- 中村文隆, Arduino工作アイデア集

