MG5 IRS

電子機械設計・製作Ⅰ

第４回

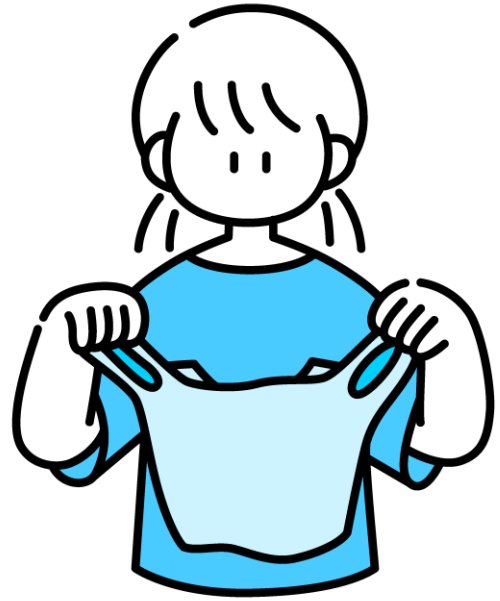
システム解説２

■ システム解説２回目

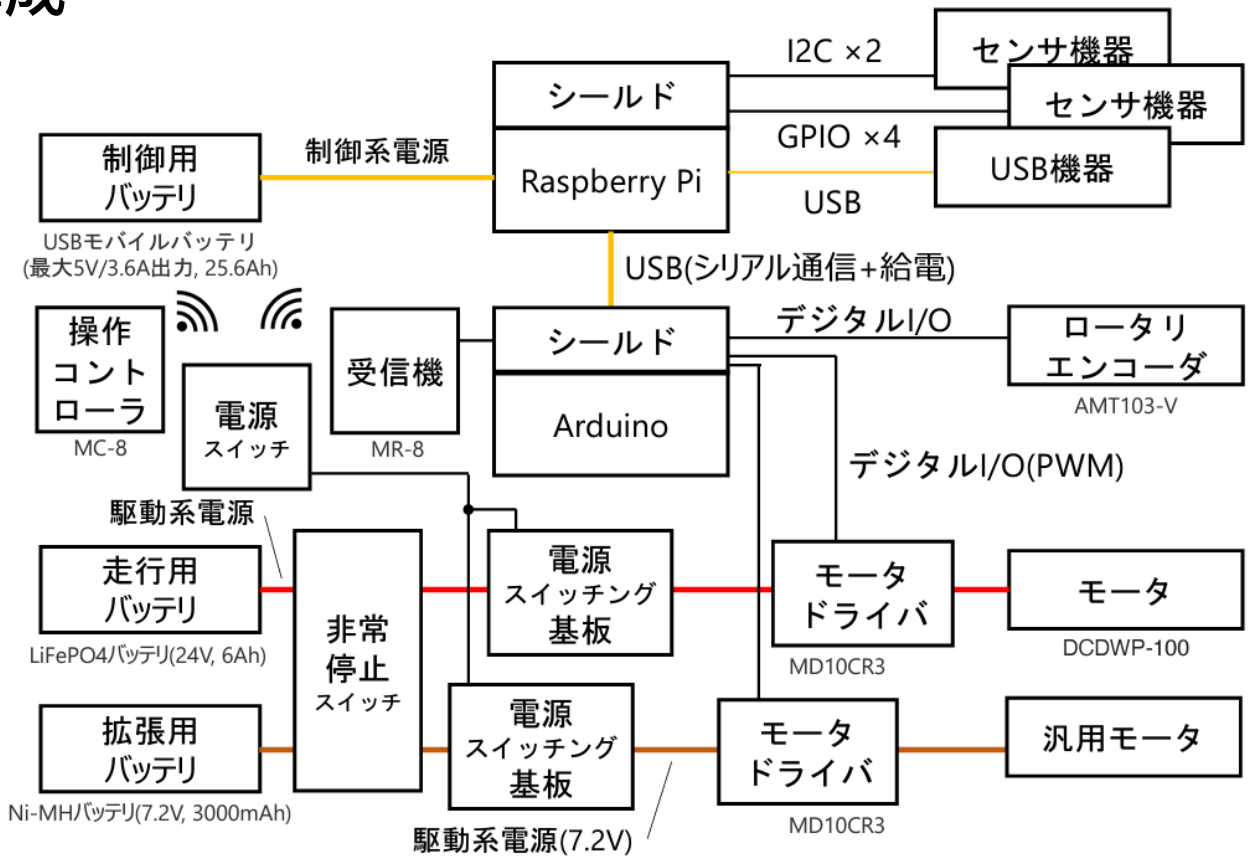
エレキ解説（大沼先生）

ソフト解説（香川）

残った時間はチームミーティングとラボの掃除



標準機の構成

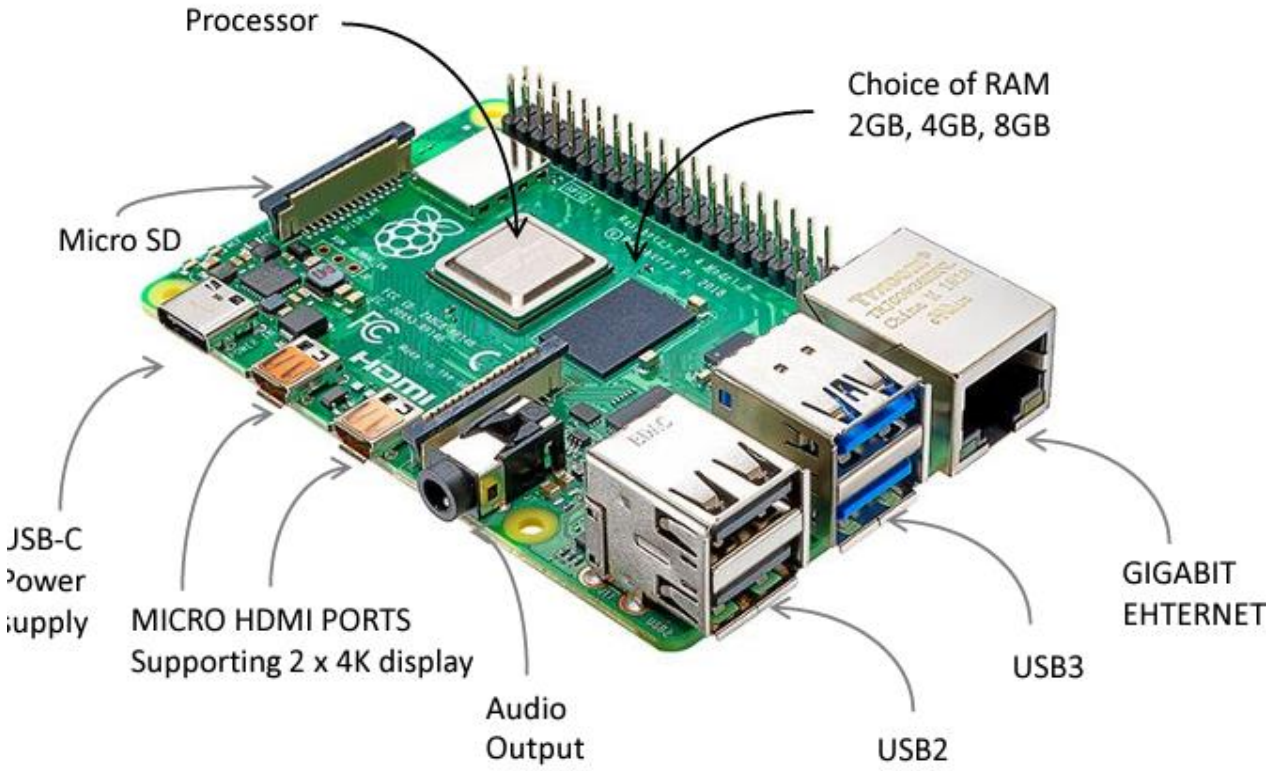


Arduino が走行制御を担い、それ以外の処理はRaspberry Pi が行っている

Raspberry Pi と Arduino はシリアル通信により命令・データを送受信する

03 ｜ Raspberry Pi 4（Model B）

- 読み方：らずべりーぱい
- シングルボードコンピュータ
- まあまあの機能・性能
  - CPU
    - 1.5 GHz
    - クアッドコア
    - ARM Cortex-A72
  - メモリ：4 GB
  - USB、HDMI、LAN
- 40 pin GPIO



- 読み方：あるでゅいーの　うの
- シングルボードコンピュータ
- Arduino 製品群の中での最もスタンダード
  - 5 V 駆動（ USB 供給可 ）
  - CPU：ATmega328P
  - I / O
    - 信号レベル：5 V
    - アナログピン 5
    - デジタルピン 14
      - 割り込み 2ピン
      - PWM出力 6 ピン

- MIRS標準機
  - Raspberry Pi とArduino の 2 つのマイコンで制御系を構成している
    - Arduino が走行制御を担い、それ以外の処理はRaspberry Pi が行っている
    - Raspberry Pi と Arduino はシリアル通信により命令・データを送受信する
  - それぞれに **標準プログラム** が用意されている
- 開発環境
  - Raspberry Pi
    - 開発言語：C言語
    - ソースコードはエディタがあれば可
    - GCC がインストールされていればコンパリルまで可
  - Arduino
    - 開発言語：Arduino言語（ ≒C言語 ）
    - Arduino IDE がインストールされているPC (Raspberry Pi含む ）

MG5 IRS

■ OS：Raspberry Pi OS

- 2020 年 5 月、Raspbian（ らずびあん ）から改名
- Debian 系 Linux を Raspberry Pi にカスタマイズしたもの
- 32 bit （ 8 GB メモリ搭載用には 64 bit ベータ版 ）

■ ライブラリ

- Wiring Pi：Raspberry Pi のGPIO、シリアル通信（ I2C、RS232C ）
- OpenCV ： 画像処理（APIはC++, Python に対応） バージョン3 or 4
- Pthread ： マルチスレッド（ マルチタスク ）

■ 開発言語

- 標準で C、C++、python （ 2 or 3 ）による開発が可能
- C、C++ のコンパイラ GCC（ gcc、g++ ）
- 標準プログラムは C 言語で記述（ 一部に C++ で記述 ）

- Arduino IDE
  - Arduino の統合開発環境
  - Windows、PC、Linux（ Raspberry Pi OS を含む ）版あり
  - Version 1.x と 2.x があるが、ソースレベルでは互換性あり（ 上位互換 ）
- Arduino 言語
  - C / C++ をベースにしており、C 言語のすべての構造といくつかの C++ の機能をサポート
  - Arduino 日本語リファレンス
  - 必須関数

    - loop 関数：main 関数の中の while(1) のようなもの

    - setup 関数：ピンモード（ 入出力、PWM、割込み ）の設定など

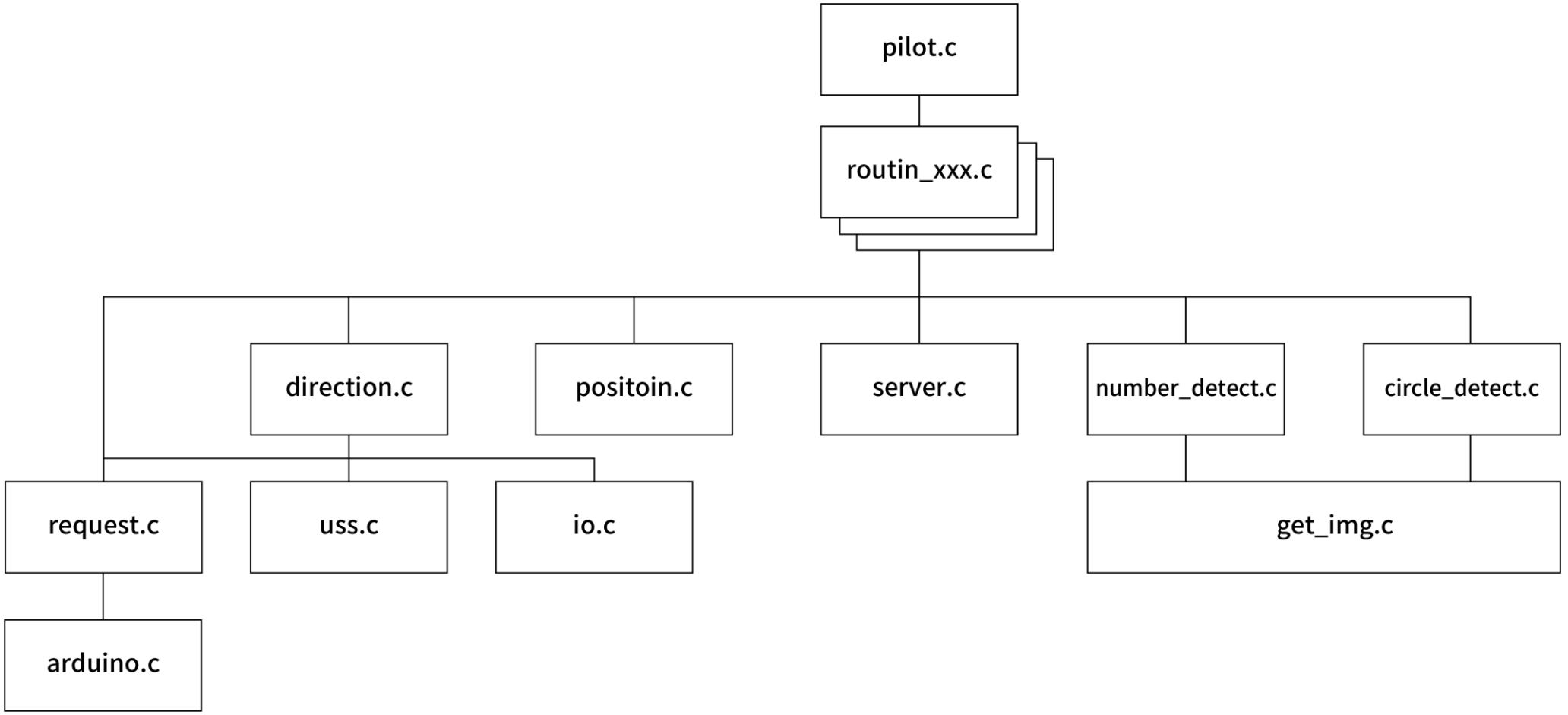
- Raspberry Pi には C 言語、Arduino には Arduino 言語でコーディングされた標準プログラムを用意している
- Raspberry Pi 用
  - MIRS2015 の巡回警備の競技会に必要な機能モジュール群とそのテストプログラム
- Arduino 用
  - 直進・回転の走行制御を行うのに必要なモジュール
  - Raspberry Pi との通信機能モジュールとそのテストプログラム

MIRS MG5 管理台帳

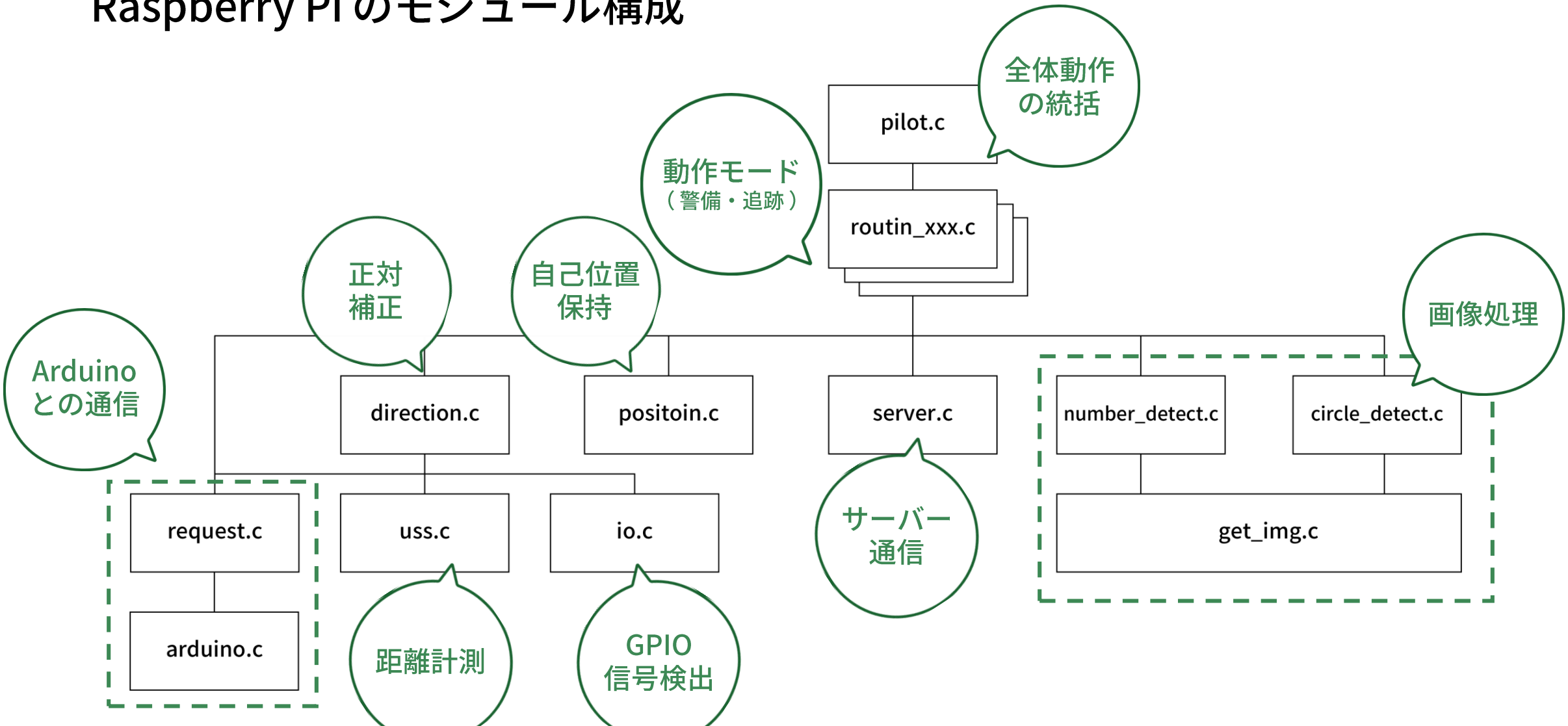
ソフトウェア

ドキュメント番号	ドキュメント名称	採番者	版数	作成者	登録日	備考
MIRSMG5D-SOFT-0001	Arduino ソフトウェア 解説	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0002	Arduino ソフトウェア 関数レファレンス	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0003	Arduino ソフトウェア ソースコード	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0004	Ardunio ソフトウェアのテストプログラム	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0005	RaspberryPi ソフトウェア 解説	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0006	RaspberryPi ソフトウェア 関数レファレンス	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0007	RaspberryPi ソフトウェア ソースコード	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0008	Raspberry Pi ソフトウェアのテストプログラム	小谷　進	A01	牛丸真司	2024/03/25	
MIRSMG5D-SOFT-0009	ソケット通信によるプロセス間通信	小谷　進	A01	牛丸真司	2024/03/25	

Raspberry Pi のモジュール構成



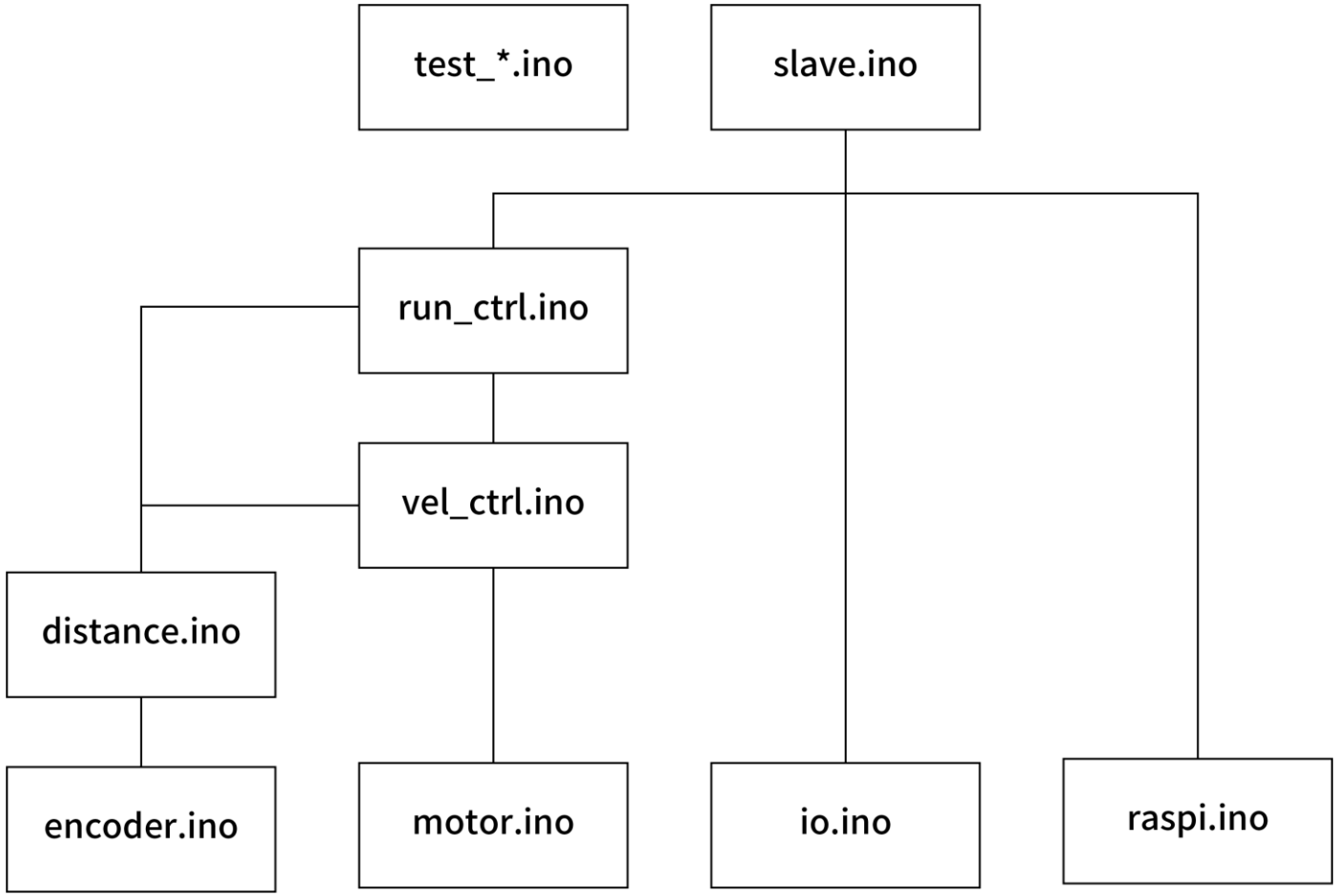
Raspberry Pi のモジュール構成



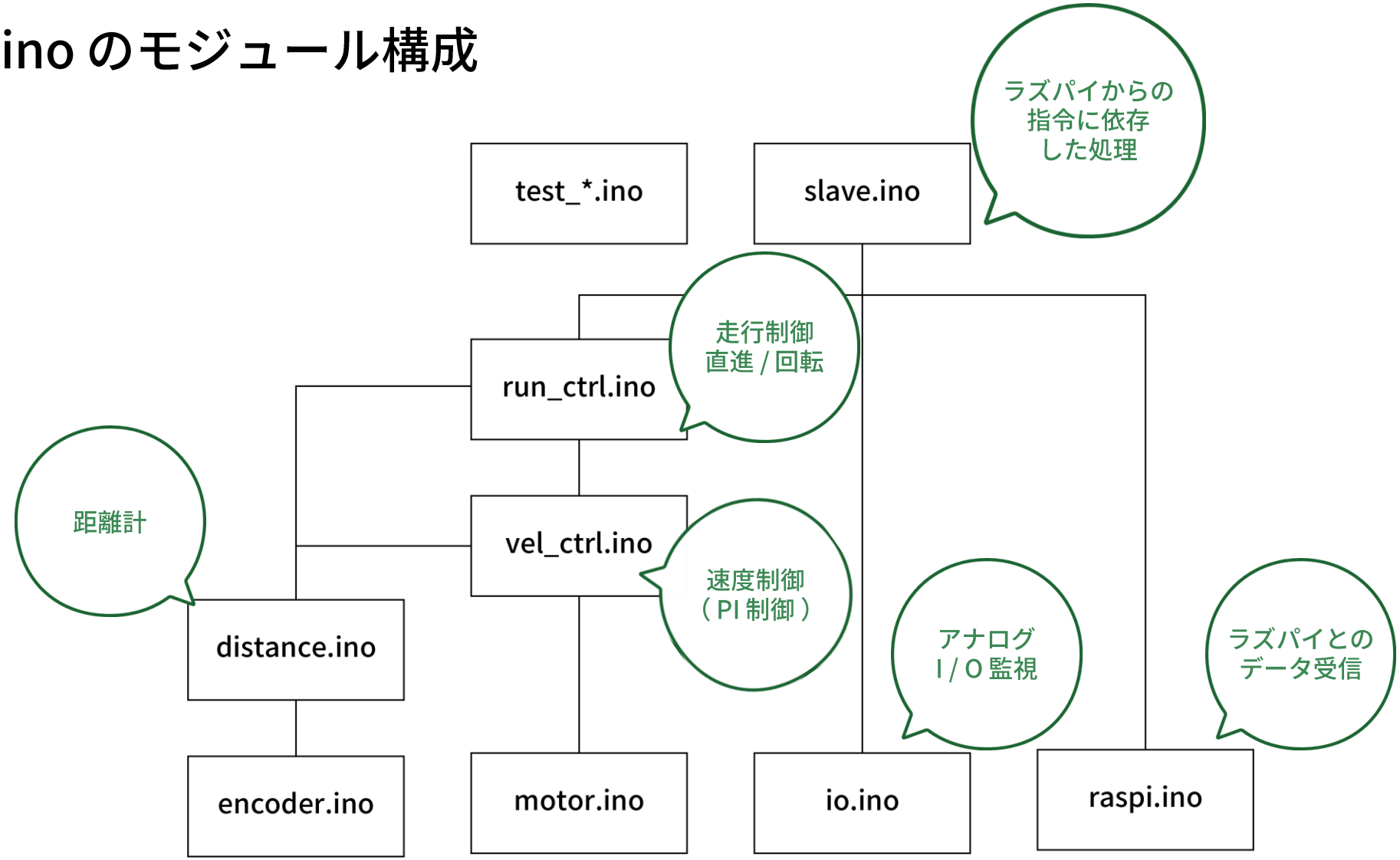
Raspberry Pi の標準プログラム

- モジュール毎に独立なファイルとしている（プログラム分割）
- ソースファイルに対応したヘッダファイルで、ソース内に定義している関数、共有変数の extern 宣言する
- モジュール単体テスト用のプログラムを test_*.c として用意している
- コンパイルはソースプログラムのトップで make する
  → トップディレクトリに Makefile を用意
- ソースプログラム（*.c）、ヘッダファイル（*.h）、オブジェクトモジュール（*.o）を、src、include、build ディレクトリに分けて置いている

Arduino のモジュール構成



Arduino のモジュール構成



Arduino の標準プログラム

- モジュール毎に独立なファイルとしている（IDE上ではタブで分離）
- テストプログラム
  - test のタブに記述
  - API は test_ で始まる関数
  - loop 関数中の一つの test_ 関数のみを使用する（他はコメントアウトする）

プログラムの追加・改変

- Raspberry Pi
  - Makefile に変更を加える
    - 具体的な変更方法はプログラミング応用の授業で説明するとのこと
    - 標準プログラム内の Makefile から類推することも可能
- Arduino
  - Arduino IDE のタブを追加
- 標準プログラムのコーディングルールに従う場合は、ソースプログラム以外にヘッダファイル、テストプログラムを用意する

- 組込みシステムの開発（MIRS 開発を含む）では、ソフトとハードの並行開発が一般的
- コーディングは、ハードウェアの有無に関係なく可能
  - コーディングだけなら、エディタさえあれば可能
  - コンパイルチェックは必要なライブラリがある環境で可能
- 単体・機能試験
  - 試験を行いたいハードウェアがあれば、単体・機能試験は可能
  - ハードウェアは実機にマウントされている必要はない
- ハードウェアが完成していなくとも（手元になくても）、**ハードウェアの仮想化**による動作試験が可能

仮想化による試験

- ハードウェアの仮想化
  - 実際のハードウェアの別のハードウェアまたは事前に用意したデータで代用する
    - タッチセンサ押下　→　キーボード入力
    - カメラ画像　→　事前に用意した画像の読み込み
    - 通信　→　相手から受け取るデータを事前に用意
- ソフトウェアの仮想化
    - 画像認識処理結果　→　事前に用意した処理結果情報を渡す

他言語での開発

- 最近の MIRS オリジナル開発では、画像処理系や通信系など一部（または全て）を Python や C++ で記述するチームが増えている
  - C で開発したプログラムと他言語で開発したプログラムを共存させることは可能
    - データ・情報はプロセス間通信によって受け渡しできる
  - 画像処理系は Python で書かれたソースコードのサンプルが多い（特に DL 系）
    - OpenCV 3.x のAPI は C 言語に非対応（C++、Python、Java に対応）
    - TensorFlow、PiTorch などの DL フレームワークは自前で環境を作る必要がある
- Raspberry Pi の標準プログラムを全て Python で書き換えたチームもあり

走行制御

- モータは PWM 信号で実行電圧（= 平均電圧）を変化させて制御する
- ロータリーエンコーダの A 層信号を用いた割り込み処理により、エンコーダの矩形信号エッジのカウントを行う
- ロータリーエンコーダのカウント値を利用して、左右のモータの速度制御を PI 制御で行う
- ロータリーエンコーダのカウント値を利用して、機体の直進・回転制御を PID（P / PD / PID）制御行う