



## R 1. 使用物品

- RaspberryPi 2 or 3、USB 5V 電源+ケーブル
- キーボード
- 超音波センサ (Devantech SRF02)
- マウス
- I2C レベル変換器 (Devantech I2C-LVL01)
- HDMI ケーブル
- Web カメラ (WebCam C270)
- 数字ボード、メジャー
- タッチセンサ
- 演習室 PC ディスプレイ

## R 2. 参照ドキュメント

- (1) [MIRSMG4D-SYST-0004](#) Raspberry Pi のセットアップとデバイス接続
- (2) [MIRSMG4D-SYST-0006](#) 超音波センサの I2C アドレスの変更手順
- (3) [MIRSMG4D-SYST-0008](#) Raspberry Pi での動作試験

## R 3. 注意事項

- 機器の接続 (配線) は、ドキュメントのとおりになっているかを複数人でチェックすること。
- 電源 (5V USB モバイルバッテリー) の接続は、機器接続に問題がないことを確認した後にすること。

## R 4. 手順

R4-1. [MIRSMG4D-SYST-0004](#) からリンクされている「Raspberry Pi のインストールとセットアップ」の 4~7 を実施し、Raspbian 環境をセットアップする。

### R4-2. デバイス接続

[MIRSMG4D-SYST-0004](#) のドキュメントを参照して、Raspberry Pi に以下のデバイス接続を行う。

超音波センサ 2 個、タッチセンサ 1 個、Web カメラ 1 台

(キーボード、マウス、ディスプレイは演習室 PC を利用する)

### R4-3. Raspberry Pi での設定とテストの実行

- (1) [MIRSMG4D-SYST-0006](#) のドキュメントを参照して、超音波センサの I2C アドレスを確認する。
- (2) [MIRSMG4D-SYST-0008](#) のドキュメントからリンクしている Raspberry Pi 用 MG4 標準ソフトウェア [mg4\\_pi\\_ver3.0.1.tar.gz](#) をダウンロードし、展開する。

展開するコマンド : `tar xvzf mg4_pi_ver3.0.tar.gz`

- (3) テストプログラムを実行する

`./test_uss` を実行し、超音波センサによる距離計測を行う。

`sudo ./test_io` を実行し、ON/OFF IO (タッチセンサ) の信号を確認する。

`./test_camera`, `./test_number` を実行し、数字認識テストを行う。

(img 以下の処理画像をホームホルダーに保存しておく。)

## MIRS MG4 Arduino のセットアップと単体機能試験

### A1. 使用物品

- Arduino UNO
- USB ケーブル
- デジタルオシロスコープ (ラボから借用)
- モータ (Maxon motor or 朱雀技研モータ)
- モータドライバ (Cytron MD10C)
- 模型用バッテリー (3700mAh、定格 7.2V)
- ケーブル
- 演習室 PC

### A2. 参照ドキュメント

- (1) [MIRSMG4D-SYST-0003](#) Arduino のセットアップとデバイス接続
- (2) [MIRSMG4D-SYST-0007](#) Arduino 単体での動作試験

### A3. 注意事項

- 電源 (模型用バッテリー) の接続は、機器接続に問題がないことを確認した後にすること。
- 赤ケーブルは Vcc、黒 (または白) ケーブルはグランド (GND) として使うこと。
- 電源スイッチ基板は裏側で短絡しないよう絶縁物の上に乗せること。

### A4. 手順

#### A4-1. デバイス接続

[MIRSMG4D-SYST-0003](#) のドキュメントを参照して、Arduino に以下のデバイス接続を行う。

モータドライバ、電源スイッチ基板 (電圧モニタ端子)、モータ (エンコーダ信号)

#### A4-2. Arduino の設定とテスト実行

- (1) [MIRSMG4D-SYST-0007](#) から Arduino 用の MG4 標準ソフトウェア [mg4\\_arduino\\_ver311.zip](#) をダウンロードし、PC (Windows7) 上の日本語を含まないフォルダに展開する。
- (2) 使用するモータに合わせて define.h にエンコーダ分解能とギア比を設定する
  - 朱雀モータ (エンコーダ 13PPR、ギア比 27:1) の場合

```
#define ENC_RANGE (13*2) // エンコーダ分解能 13×2 通倍
#define T_E_RATIO 27.0 // ギア 27:1
```
  - MAXON モータの場合 (エンコーダ 100PPR、ギア比 16:1)

```
#define ENC_RANGE (100*2) // エンコーダ分解能 100×2 通倍
#define T_E_RATIO 16.0 // ギア比 16:1
```
- (3) [mg4\\_arduino\\_ver311.ino](#) を変更して次のテストを行う。
  - モータ回転、エンコーダ出力、速度制御
    - PWM 値 (duty 比)、速度指令値を変更して、速度、回転方向が異なることを確認する。
    - エンコーダ出力、速度制御のテスト時は、シリアルモニターを起動してその値を確認する。その際シリアルモニターの通信速度は 115200bps に設定する。
  - エンコーダ出力テスト時に PWM 波形とエンコーダ波形をオシロスコープで確認する。  
(波形をカメラで撮影しておくこと、またその際の duty 比を記録しておくこと)
- (4) 実験用のモータで確認できたら、自チームの機体に搭載したモータを使用して同様のテストを行う。