

Studuino

ブロックプログラミング環境

入門ガイド 拡張編

【カラーセンサー/ジャイロセンサー/温度センサー/赤外線リモコン受信センサー
超音波センサー/ブルートゥースモジュール】

2017/01/16 作成

2018/01/22 改訂

ArTeC[®]

改訂履歴

改訂日付	改訂内容
2017/01/16	初版発行
2018/01/22	ソフトウェア、OS のバージョンアップに伴う改訂

目次

1. はじめに	1
2. ご利用の前に.....	2
3. カラーセンサー	3
3.1. 概要.....	3
3.2. 仕様.....	3
3.3. 接続方法	3
3.4. 設定.....	4
3.5. カラーセンサーの値について	5
3.6. カラーセンサーを使用したプログラミング例	5
4. ジャイロ・加速度センサー	6
4.1. 概要.....	6
4.2. 仕様.....	6
4.3. 接続方法	6
4.4. 設定.....	7
4.5. ジャイロセンサーの値について	8
4.6. プログラミング例.....	8
5. 温度センサー	10
5.1. 概要.....	10
5.2. 仕様.....	10
5.3. 接続方法	10
5.4. 設定.....	11
5.5. 温度センサーの値について	12
5.6. プログラミング例.....	12
6. 赤外線リモコン受信センサー	13
6.1. 概要.....	13
6.2. 仕様.....	13
6.3. 接続方法	13
6.4. 設定.....	14

6.5.	赤外線リモコン受信ブロック値について.....	15
6.6.	プログラミング例.....	15
7.	超音波距離センサー.....	19
7.1.	概要.....	19
7.2.	仕様.....	19
7.3.	接続方法.....	19
7.4.	設定.....	20
7.5.	超音波センサーブロック値について.....	21
7.6.	プログラミング例.....	21
8.	Bluetoothモジュール.....	22
8.1.	概要.....	22
8.2.	仕様.....	22
8.3.	接続方法.....	23
8.3.1.	Bluetoothモジュール RBT-001 (品番 : 86873)	23
8.3.2.	ロボット用Bluetoothモジュール (86876)	24
8.4.	設定.....	25
8.5.	PC、タブレット端末 (スマートフォン) 等との接続方法 (ペアリング)	26
8.6.	ロボット用Bluetoothモジュールを使用したプログラミング例	26
8.6.1.	Bluetooth基本ブロックを使用したプログラミング例.....	26
8.6.2.	コントローラアプリを使用したプログラミング例.....	29
8.7.	テストモード中の動作について.....	30
9.	お問い合わせ先.....	31

1. はじめに

本書は、ブロックプログラミング環境で以下の電子パーツを利用する方法について記載しています。

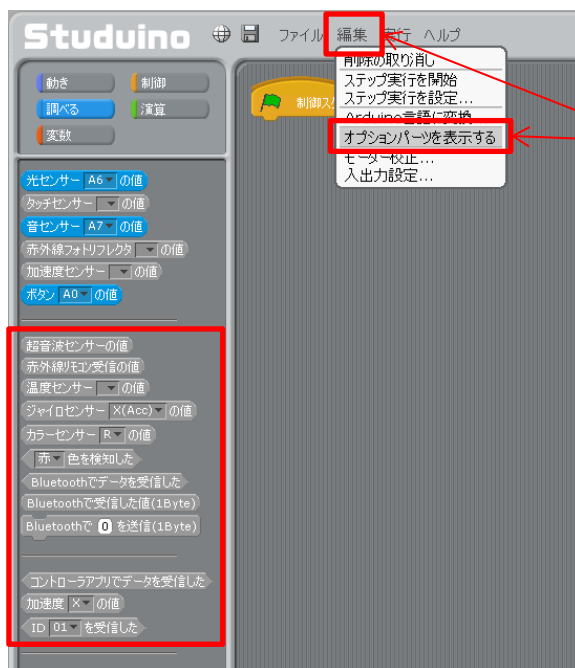
- ・カラーセンサー
- ・ジャイロセンサー
- ・温度センサー
- ・赤外線リモコン受信センサー
- ・超音波センサー
- ・ブルートゥースモジュール

本書の内容は製品の仕様変更などにより、加筆・修正が加えられることがあります。

2. ご利用の前に

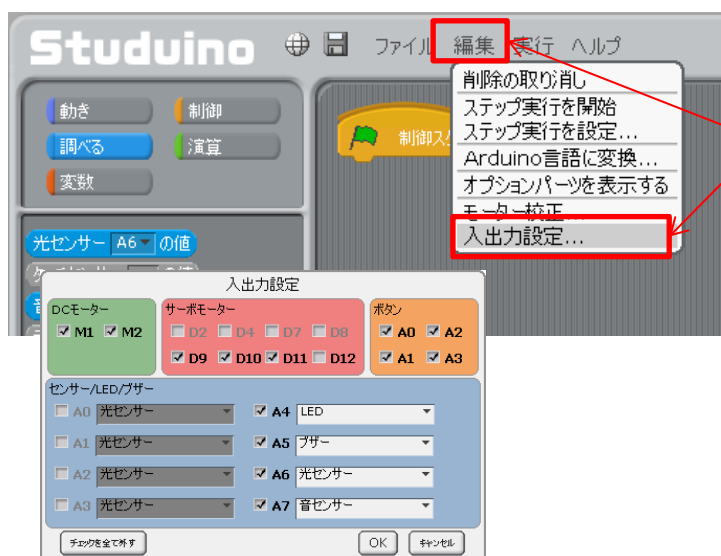
本書に記載の各電子パーツを使用する場合、最初に下記の設定を行ってください。

- ① 「編集」メニューから「オプションパーツを表示する」を選択し、新規センサブロックを表示します。



「編集」メニューをクリックして「オプションパーツを表示する」を選択します。

- ② 「編集」メニューから「入出力設定…」を選択し入出力設定ダイアログを表示し、使用する電子パーツの設定を行います。設定方法は各電子パーツの解説ページをご覧ください。



「編集」メニューをクリックして「入出力設定…」を選択します。

3. カラーセンサー

3.1. 概要

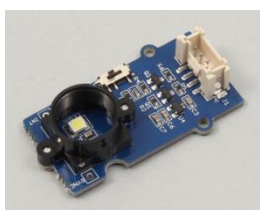
カラーセンサーは、カラーセンサーIC TCS3414CS を搭載したセンサーです。
環境光の色度や、物体の色を測定します。

3.2. 仕様

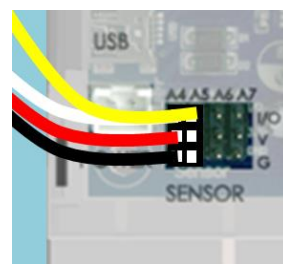
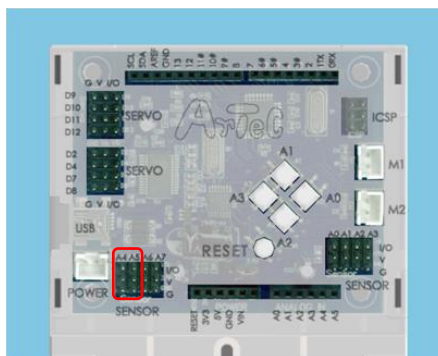
センサー部品	TCS3414CS
動作電圧	3.3~5V
インターフェイス	I2C
測定対象物距離	約 10mm

3.3. 接続方法

- ① 別売の『086882 カラーセンサー接続コード (4 芯 30cm)』を使用してください。
- ② カラーセンサー接続コードの白のコネクタをカラーセンサーに、黒のコネクタを Studuino に接続します。
- ③ A4 と A5 にまたがるように接続します。(他のコネクタに接続しても使用できません)。
- ④ 使用するときは基板上的のスライドスイッチを ON にし、LED を点灯させてください。



カラーセンサー

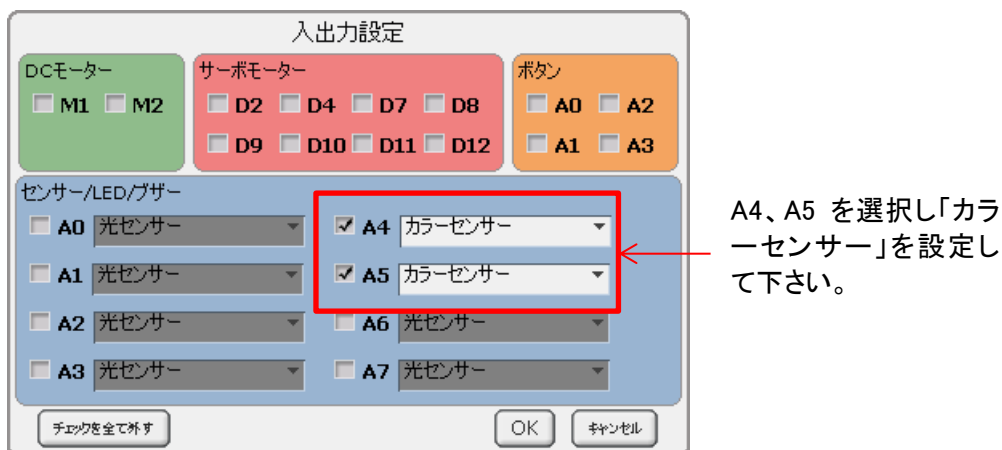


センサー接続コードの向きに
注意しましょう。

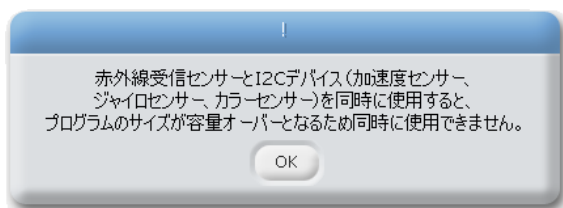
A4 に白の線、A5 に黄色の線
にがつながるように接続しま
す。

3.4. 設定

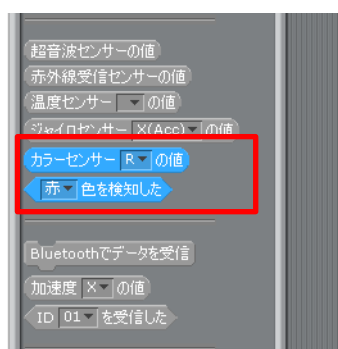
- ① 入出力設定ダイアログの「センサー/LED/ブザー」で、カラーセンサーを接続している Studuino 基板のコネクタ名 (A4、A5) にチェックを入れ、▼をクリックして「カラーセンサー」を選択し、OK をクリックしてください。



※ブロックプログラミング環境では、赤外線受信センサーと I2C デバイス (加速度センサー、ジャイロセンサー、カラーセンサー) を同時に使用した場合、プログラムのサイズが Studuino 基板のメモリのサイズを超過するため、入出力設定ダイアログで赤外線受信センサーと I2C デバイスを選択すると、下図のメッセージを表示し両センサーを同時に使用できないようにしています。



- ② 赤枠の中のブロックが有効になります。



3.5. カラーセンサーの値について

カラーセンサーは、物体の色の成分（赤色、青色、緑色）を計測し、その値と色度座標と物体の色を検出します。カラーセンサー R の値では、それぞれの値を調べます。色の成分（赤色、青色、緑色）値は 0~100 の整数値で、色度座標は実数です。検出する色は、赤、緑、青、白、黄、茶、黒です。それぞれの値は、テストモード時に表示されるセンサー・ボードで確認できます。

センサー・ボード	
[A0] 未接続	*
[A1] 未接続	*
[A2] 未接続	*
[A3] 未接続	*
[A4/A5] カラーセンサー(R)	3
[A4/A5] カラーセンサー(G)	5
[A4/A5] カラーセンサー(B)	3
[A4/A5] カラーセンサー(C)	14
[A4/A5] カラーセンサー(X)	0.3
[A4/A5] カラーセンサー(Y)	0.3
[A4/A5] 検知した色	-
[A6] 未接続	*
[A7] 未接続	*

物体の色の各値が表示されます

センサー・ボードに表示される R、G、B の値はセンサーが検知した色成分（赤色、青色、緑色）の値で、X、Y は色度座標の値です。上記 5 つの値を評価して、物体の色を検出します。

3.6. カラーセンサーを使用したプログラミング例

赤色を検知したブロックの使用例を下図に示します。下図のプログラムは、カラーセンサーが検知した色を LED で表示するプログラムです。A0 に赤色 LED、A1 に緑色 LED、A2 に青色 LED が接続されています。

The screenshot shows the Arduino IDE interface. On the left, the 'Components' panel has 'カラーセンサー R' selected. The main workspace contains a program starting with '制御スタート' followed by a 'ずっと' loop. Inside the loop, there are three 'もし' (if) blocks: 1. 'もし 赤色を検知した なら' (if detected red, then) with actions 'LED A0 を点灯' (turn on red LED), 'LED A1 を消灯' (turn off green LED), and 'LED A2 を消灯' (turn off blue LED). 2. 'もし 緑色を検知した なら' (if detected green, then) with actions 'LED A0 を消灯' (turn off red LED), 'LED A1 を点灯' (turn on green LED), and 'LED A2 を消灯' (turn off blue LED). 3. 'もし 青色を検知した なら' (if detected blue, then) with actions 'LED A0 を消灯' (turn off red LED), 'LED A1 を消灯' (turn off green LED), and 'LED A2 を点灯' (turn on blue LED). On the right, the 'センサー・ボード' (Sensor Board) window shows the following values: [A0] LED (*), [A1] LED (*), [A2] LED (*), [A3] 未接続 (*), [A4/A5] カラーセンサー(R) 2, [A4/A5] カラーセンサー(G) 7, [A4/A5] カラーセンサー(B) 3, [A4/A5] カラーセンサー(C) 15, [A4/A5] カラーセンサー(X) 0.3, [A4/A5] カラーセンサー(Y) 0.3, [A4/A5] 検知した色 緑 (green), [A6] 未接続 (*), [A7] 未接続 (*).

4. ジャイロ・加速度センサー

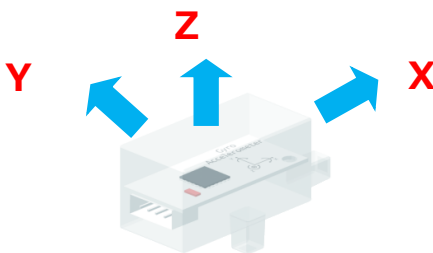
4.1. 概要

ジャイロ・加速度センサーは、InvenSense 社の MPU-6050 を搭載した、三軸ジャイロセンサー・三軸加速度センサーモジュールです。

4.2. 仕様

センサー部品	MPU-6050
動作電圧	2.4~3.4V
インターフェイス	I2C
ジャイロ	三軸±250、±500、±1000、±2000dps（初期設定 ±250dps）
加速度	三軸±2g、±4g、±8g、±16g（初期設定 ±2g）

X・Y・Z 軸方向は基板の上に記載の通りです。

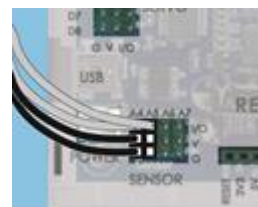
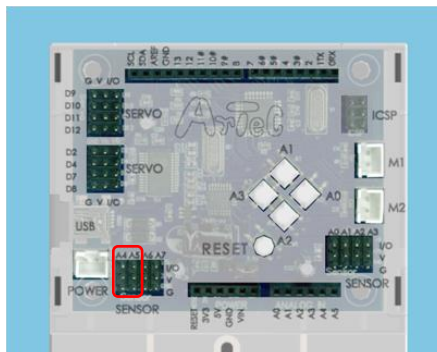


4.3. 接続方法

- ① 別売の『153127 センサー接続コード（4 芯 50cm）』を使用してください。
- ② センサー接続コードの白のコネクタをジャイロ・加速度センサーに、黒のコネクタを Studuino に接続します。
- ③ A4 と A5 にまたがるように接続します。（他のコネクタに接続しても使用できません）。
信号線（灰色の線）が上側になるように接続して下さい。



ジャイロ加速度センサー

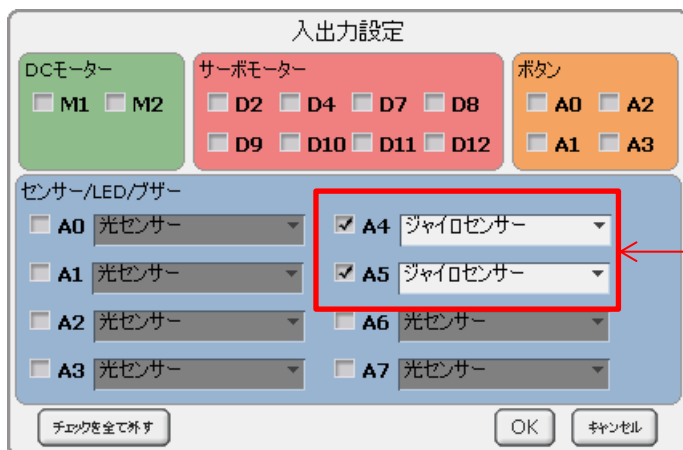


センサー接続コードの向きに
注意しましょう。

4 本線のコードを使用して、
A4 と A5 にまたがるように接
続します。

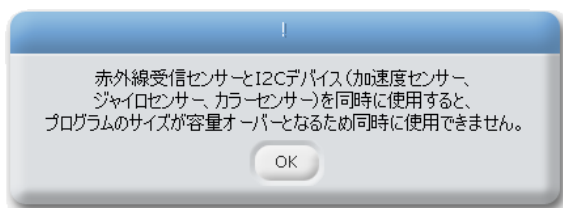
4.4. 設定

- ① 入出力設定ダイアログの「センサー/LED/ブザー」で、ジャイロセンサーを接続している Studuino 基板のコネクタ名 (A4、A5) にチェックを入れ、▼をクリックして「ジャイロセンサー」を選択し、OK をクリックしてください。

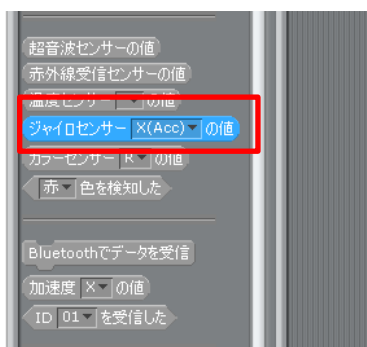


A4、A5 を選択し「ジャイロセンサー」を設定して下さい。

※ブロックプログラミング環境では、赤外線受信センサーと I2C デバイス (加速度センサー、ジャイロセンサー、カラーセンサー) を同時に使用した場合、プログラムのサイズが Studuino 基板のメモリのサイズを超過するため、入出力設定ダイアログで赤外線受信センサーと I2C デバイスを選択すると、下図のメッセージを表示し両センサーを同時に使用できないようにしています。



- ② 赤枠の中のブロックが有効になります。



4.5. ジャイロセンサーの値について

ジャイロセンサーは、物体の X・Y・Z 軸方向の傾きや角速度を検知します。ジャイロセンサーブロックでは、その値を調べます。値は、0～100 の整数値です。値は、テストモード時に表示されるセンサー・ボードで確認できます。



センサー・ボードに表示される加速度 X～Z はセンサーが検知した傾きで、角速度 X～Z はセンサーが検知した角速度です。

4.6. プログラミング例

ジャイロセンサー X(Acc) の値 ブロックの使用例を下図に示します。

下図のプログラムの内容は、タッチスイッチが押されたら、DC モーターを 2 つ使用した車型ロボットを 45°回転させるプログラムです。



0.1 秒に一回値を測定し、それを 1/10 した値を積算して何度回転したかを計算しています。

① 0 dps の時の値は 50, +250 dps の時の値は 100 なので、 $(100 - 50) / 250 = 0.2$ が 1dps に相当します。したがって $0.2 * 45 = 9$ が 45°回転したときの値です。

② 測定値から 0 dps の時の値 50 を引いています。測定値は dps (degree per second)、つまり 1 秒当たりの回転角度です。この例では 0.1 秒ごとに測定しているので、測定値を 1/10 にしています。

③ 実際には①や②の処理に時間がかかるため、ここで 0.1 秒待つと、測定間隔が 0.1 秒を越えてしまいます。この 0.1 秒を少し減らすなどして、正確な角度回転するように調整してください。

5. 温度センサー

5.1. 概要

温度センサーは、基板に温度センサーIC を搭載し、周囲の温度を電圧に変換して温度を測定するセンサーです。

5.2. 仕様

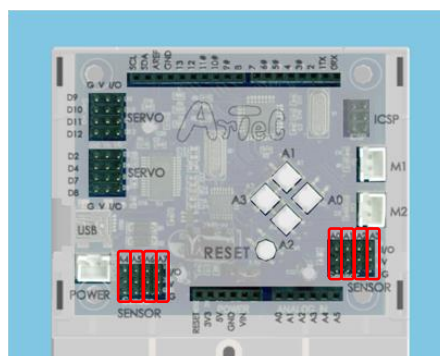
センサー部品	MCP9700
動作電圧	2.3~5.5V
動作温度	-40 ~ +125℃
精度	±4℃ (max.)、 (0℃~+70℃において)

5.3. 接続方法

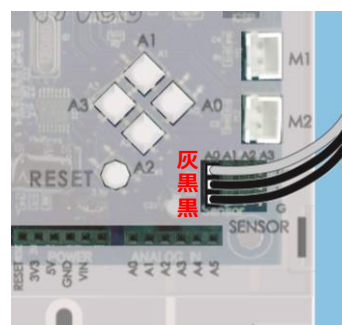
- ① 別売の『153125 センサー接続コード (3 芯 15cm)』または『153126 センサー接続コード (3 芯 30cm)』を使用してください。
- ② センサー接続コードの白のコネクタを温度センサーに、黒のコネクタを Studuino に接続します。
- ③ A0~A7 に接続できます。信号線 (灰色の線) が上側になるように接続して下さい。



温度センサー



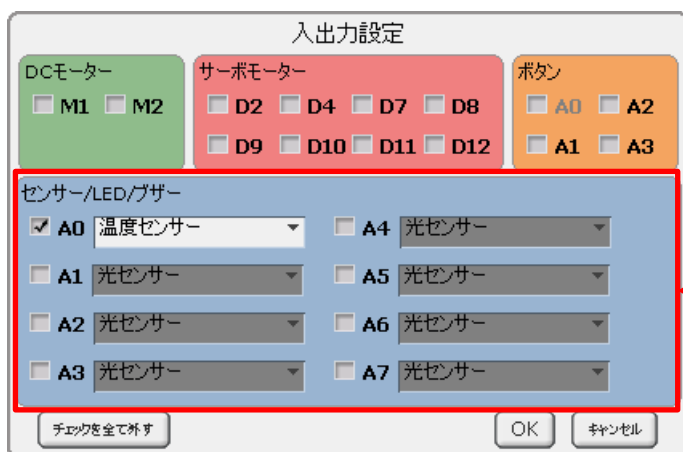
A0~A7 に接続できます。



センサー接続コードの向きに注意しましょう。上側に灰色のコードがくるように接続します。

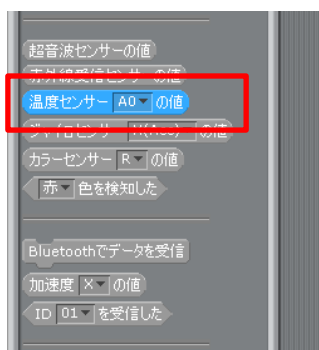
5.4. 設定

- ③ 入出力設定ダイアログの「センサー/LED/ブザー」で、温度センサーを接続している Studuino 基板のコネクタ名 (A0~A7) にチェックを入れ、▼をクリックして「温度センサー」を選択し、OK をクリックしてください。以下では、温度センサーを A0 に接続しているものとして説明します。



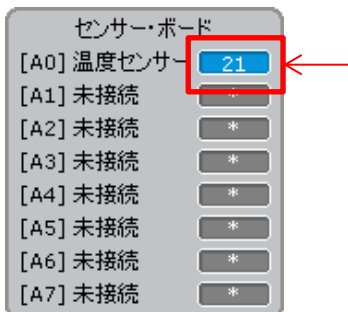
温度センサーが接続されている Studuino 基板のコネクタ名を選択し「温度センサー」を設定して下さい。

- ④ 赤枠の中のブロックが有効になります。



5.5. 温度センサーの値について

温度センサーは、周囲の温度を検知します。**温度センサー A3 の値** ではその値を調べます。値は、-40℃～125℃で、テストモード時に表示されるセンサー・ボードで確認できます。



5.6. プログラミング例

温度センサー A3 の値 ブロックの使用例を下図に示します。下図のプログラムは、温度が25℃より高い場合に LED を点灯するプログラムです。



6. 赤外線リモコン受信センサー

6.1. 概要

赤外線リモコン受信センサーは、基板に赤外線リモコン受信モジュールを搭載し、赤外線リモコンが発した赤外線を受信し、そのデータを出力するセンサーです。

6.2. 仕様

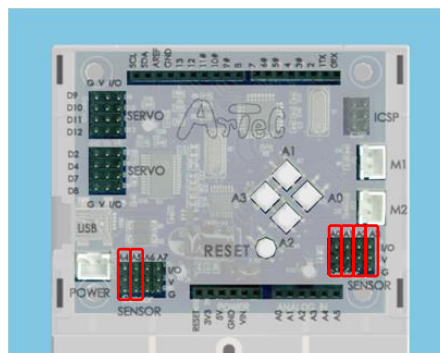
動作電圧	3.3~5V
キャリア周波数	38kHz
ピーク波長	940nm

6.3. 接続方法

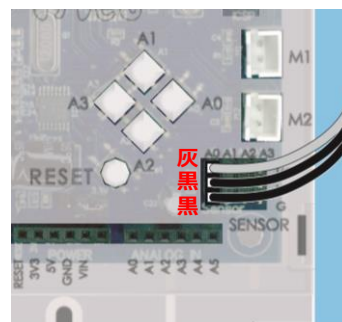
- ① 別売の『153125 センサー接続コード (3 芯 15cm)』または『153126 センサー接続コード (3 芯 30cm)』を使用してください。
- ② センサー接続コードの白のコネクタを赤外線リモコン受信センサーに、黒のコネクタを Studuino に接続します。
- ③ A0~A5 に接続できます。信号線 (灰色の線) が上側になるように接続して下さい。



赤外線リモコン
受信センサー



A0~A5 に接続できます。



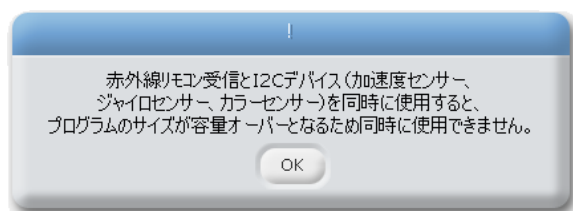
センサー接続コードの向きに注意しましょう。上側に灰色のコードがくるように接続します。

6.4. 設定

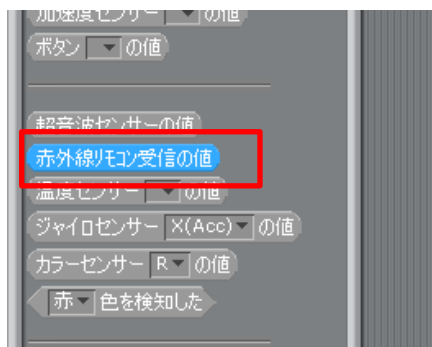
- ① 赤外線リモコン受信センサーは、A0～A5 のいずれかに設定することができます。入出力設定ダイアログの「センサー/LED/ブザー」の A0～A5 のいずれかにチェックを入れ、▼をクリックして「赤外線リモコン受信」を選択し、OK をクリックしてください。以下では、赤外線リモコン受信センサーを A0 に接続しているものとして説明します。



※ブロックプログラミング環境では、赤外線リモコン受信センサーと I2C デバイス（加速度センサー、ジャイロセンサー、カラーセンサー）を同時に使用した場合、プログラムのサイズが Studuino 基板のメモリのサイズを超過するため、入出力設定ダイアログで赤外線リモコン受信と I2C デバイスを選択すると、下図のメッセージを表示し両センサーを同時に使用できないようにしています。

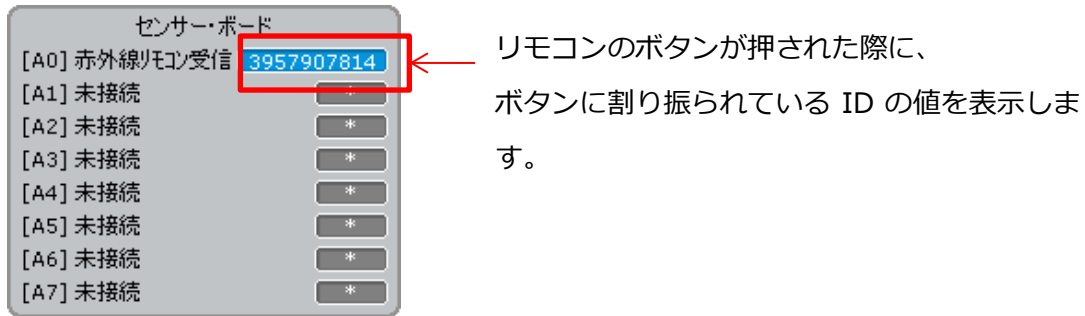


- ② 赤枠の中のブロックが有効になります。



6.5. 赤外線リモコン受信ブロック値について

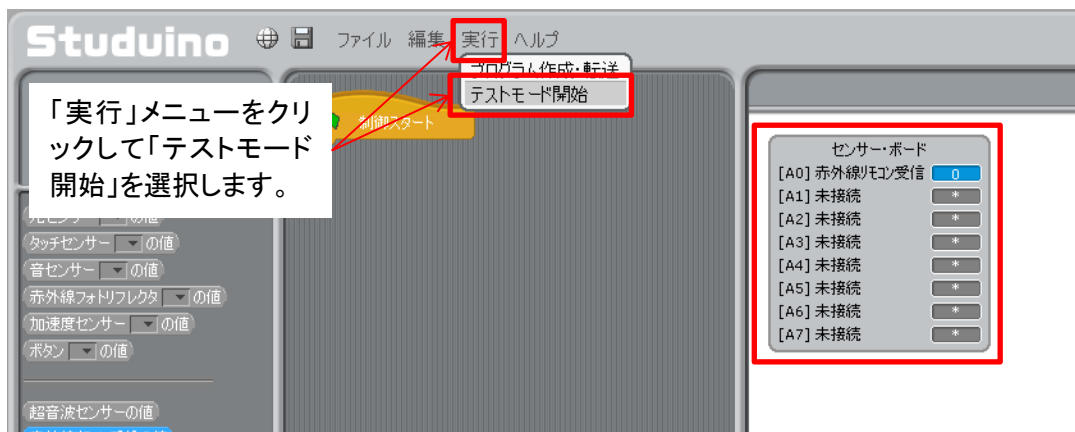
赤外線リモコン受信センサーは、リモコンのボタンが押された際に押されたボタンの ID を検知します。赤外線リモコン受信ブロックでは、その ID の値を調べます。ID の値は、0～4294967296 の整数値です。値は、テストモード時に表示されるセンサー・ボードで確認できます。押されたりモコンボタンの ID の値がそのまま表示されます。



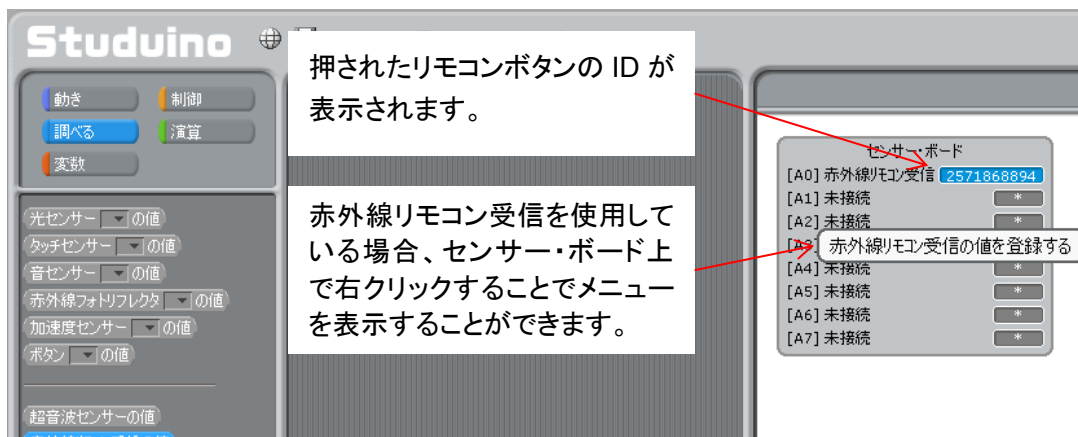
6.6. プログラミング例

赤外線リモコン受信センサーが検知するリモコンボタンの ID を定数として登録できます。

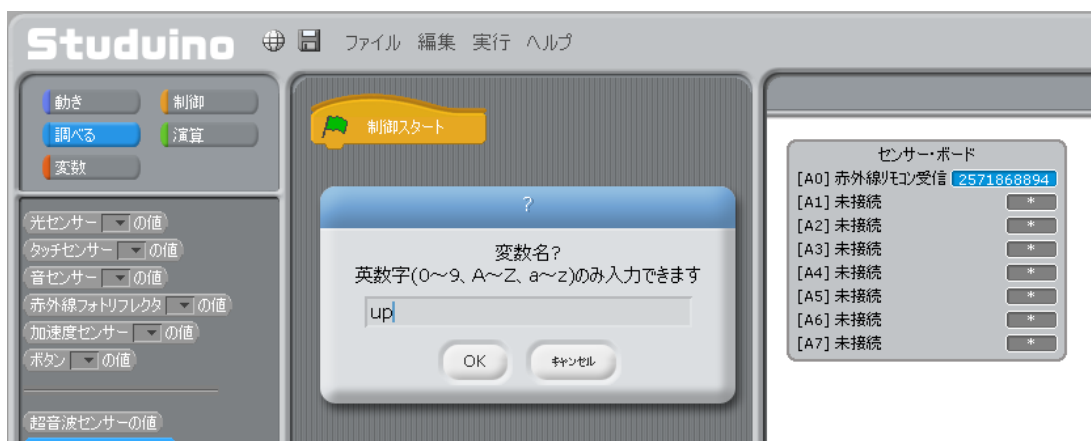
- ① 「実行」メニューから「テストモード開始」を選択しセンサー・ボードを表示します。



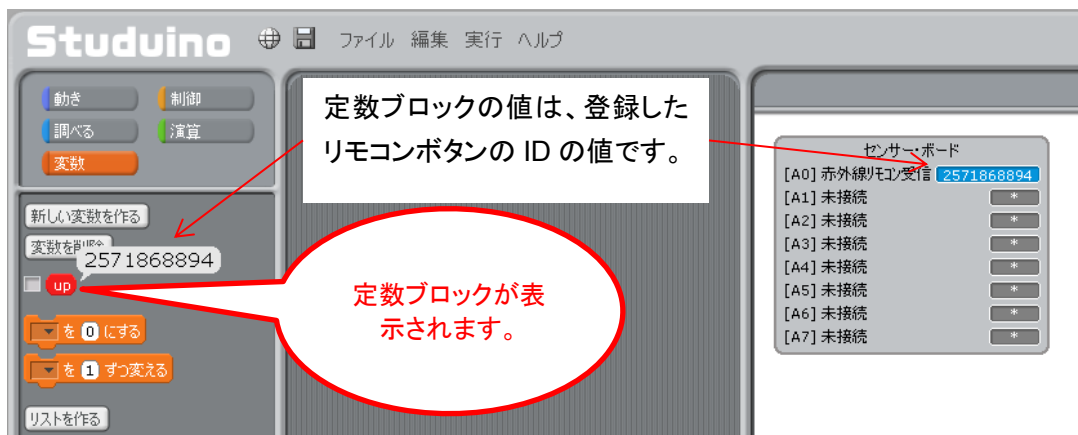
- ② リモコンのボタンを押して、センサー・ボードにボタンの ID が表示されたら、センサー・ボード上で右クリックし、「赤外線リモコン受信の値を登録する」メニューを選択します。



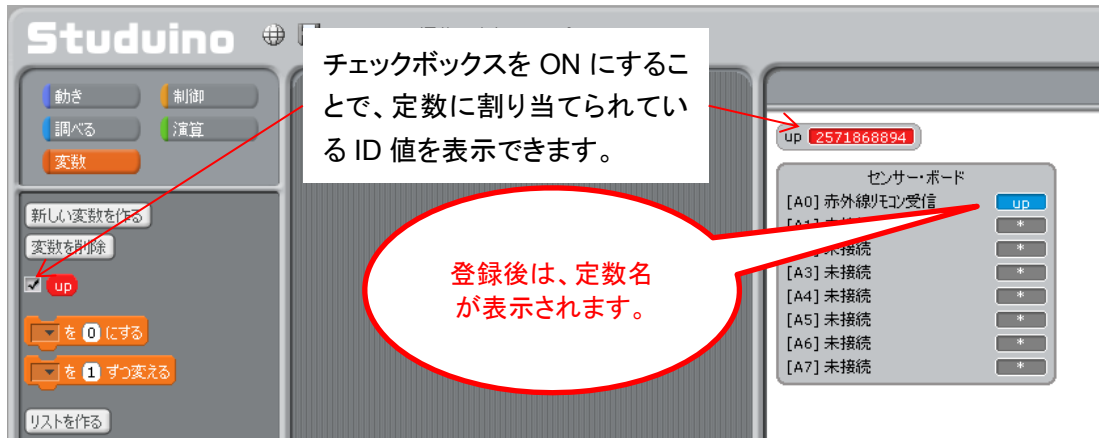
- ③ リモコンボタンの ID 値に付ける名前を入力し、OK ボタンを押します。



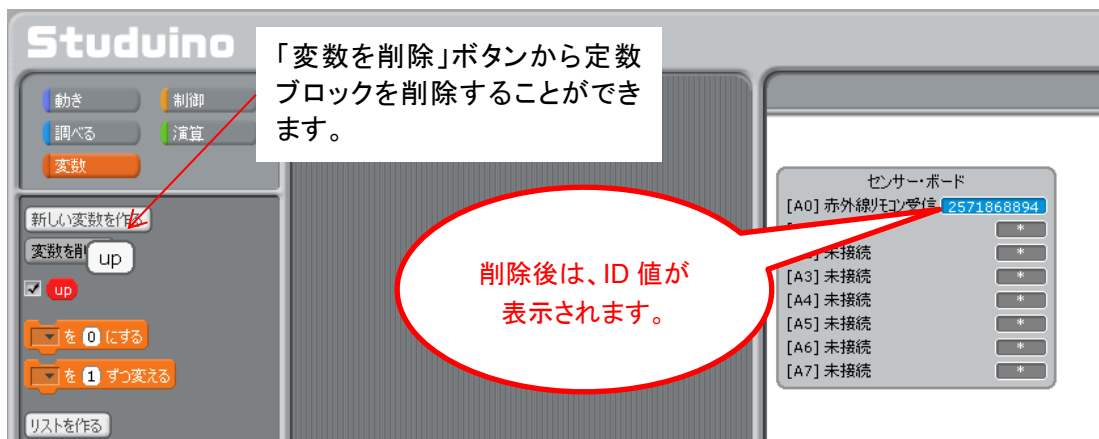
- ④ 変数カテゴリに③で設定した名前の定数ブロックが表示されます。



- ⑤ リモコンボタンの ID 値を登録すると、センサー・ボード上では登録された定数名が表示されます。また、定数に登録された ID 値は、定数ブロックの横のチェックボックスを ON にすることで確認することができます。



- ⑥ 作成した定数ブロックは、「変数削除」ボタンから削除することができます。



赤外線リモコン受信の値 ブロックの使用例を下図に示します。下図のプログラムは、DC モーターを2つ使用した車型ロボットをリモコンボタンで前進、後退させるプログラムです。リモコンの↑ボタンを forward 定数に、↓ボタンを back 定数に割り当てており、ループの先頭で赤外線リモコン受信センサーの値を取得し、センサー値と定数が等しい場合に、DC モーターを回転させます。

The screenshot shows the Studuino IDE interface. The main workspace contains a program starting with a '制御スタート' (Control Start) block, followed by a 'ずっと' (Forever) loop. Inside the loop, there are two conditional blocks: 'もし IRRemote = forward なら' (If IRRemote = forward) and 'もし IRRemote = back なら' (If IRRemote = back). Each conditional block contains a sequence of actions: setting the speed of DC Motor M1 and M2 to 100, setting their rotation direction (forward for 'forward', reverse for 'back'), a 0.1-second delay, and releasing the motors. The right sidebar shows the 'センサーボード' (Sensor Board) configuration, with the IR remote sensor set to 'forward'.

7. 超音波距離センサー

7.1. 概要

超音波距離センサーは、基板に超音波の送信部と受信部を搭載し、超音波が物体で反射して戻ってくる時間を利用して距離を測定するセンサーです。

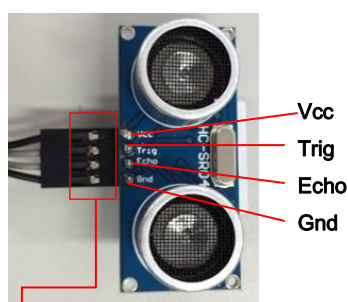
7.2. 仕様

超音波センサモジュール	HC-SR04
動作電圧	4.4V~5V
測定距離範囲	2~400cm
測定角度範囲	<15°

※電池が消耗すると正しい値が測定できない場合があります。その際は新しい電池に交換してください。

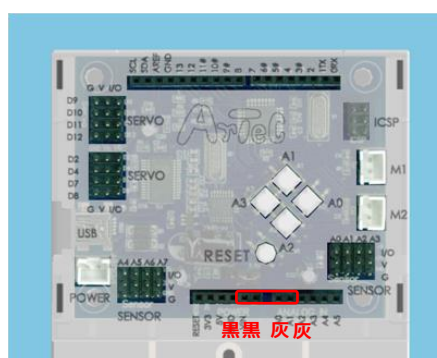
7.3. 接続方法

- ① 別売の『086881 超音波距離センサー接続コード (4 芯 30cm)』を使用してください。
- ② 超音波距離センサー接続コードのメスコネクタを、黒のコードが「Vcc」「Gnd」に、灰色のコードが「Trig」「Echo」になるように接続します。メスコネクタの金属部分が上を向くようにします。
- ③ 超音波距離センサー接続コードのオスコネクタを、黒のコードが「GND」「VIN」に、灰色のコードが「A0」「A1」になるように Studuino に接続します。



超音波距離センサー

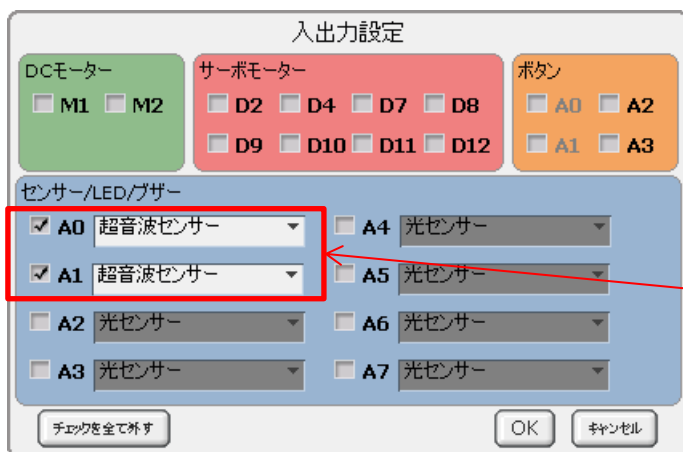
メスコネクタの金属部分が上を向くように接続します。



センサー接続コードの向きに注意しましょう。
黒：GND
黒：VIN
灰：A0
灰：A1
に接続します。

7.4. 設定

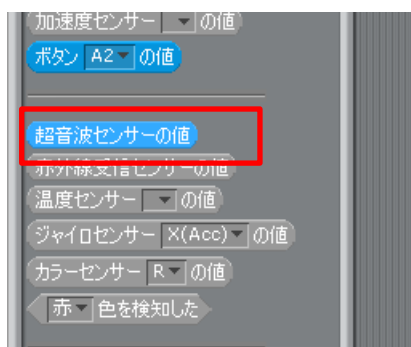
- ① 入出力設定ダイアログの「センサー/LED/ブザー」のA0、A1 にチェックを入れて、▼をクリックして超音波センサーを選択し、OK をクリックしてください。



センサーのA0、A1 のチェックボックスにチェックを入れて「超音波センサー」を選択します。

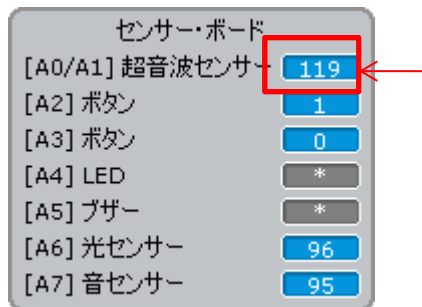
※超音波センサーは、A0、A1 にチェックが入っているときのみ、選択項目に表示されます。

- ② 赤枠の中のブロックが有効になります。



7.5. 超音波センサーブロック値について

超音波センサーの値 は、ブロック実行時の超音波センサーの値を調べるブロックです。0~400の整数値で単位は cm です。下図は、テストモード時に表示されるセンサー・ボードでの超音波センサーの値をあらわしています。



7.6. プログラミング例

超音波センサーの値 ブロックの使用例を下図に示します。下図のプログラムの内容は、DC モーターを2つ使用した車型ロボットを物体との距離が5cm以内になるまで速さ100で前進させるプログラムです。



図のように、条件判定のブロックの中に入れて使用できます。

8. ブルートゥースモジュール

8.1. 概要

ブルートゥースモジュールには、「ブルートゥースモジュール RBT-001（品番：86873）」と「ロボット用ブルートゥースモジュール（品番：86876）」と2種類があります。いずれも、SPP（Serial Port Profile）に対応し、Studuino とはシリアル（UART）接続します。

8.2. 仕様

品番	86873	86876
モジュール	ブルートゥースモジュール RBT-001	ロボット用ブルートゥースモジュール
動作電圧	2.5～3.3V	3.3～5V
バージョン	V2.0 準拠	V2.0+EDR 準拠
プロファイル	SPP	SPP
インターフェイス	UART	UART
通信距離	約 10m	約 10m
Bluetooth 認証	取得済み	未取得
技術適合証明書	取得済み	未取得

※電池が消耗すると正しい値が測定できない場合があります。その際は新しい電池に交換してください。

【ご注意ください】

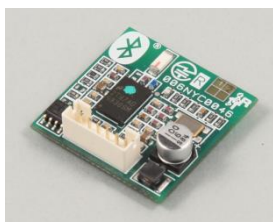
ロボット用ブルートゥースモジュール（品番：86876）については、技術適合証明を受けていないため電波暗室や電波障害を起こさない十分に広い敷地・建屋内で使用しなければなりません（平成18年総務省告示第173号にある試験設備の要件を満たす必要があります）。

上記条件以外の場所で利用すると、日本の法律に違反する可能性があります。

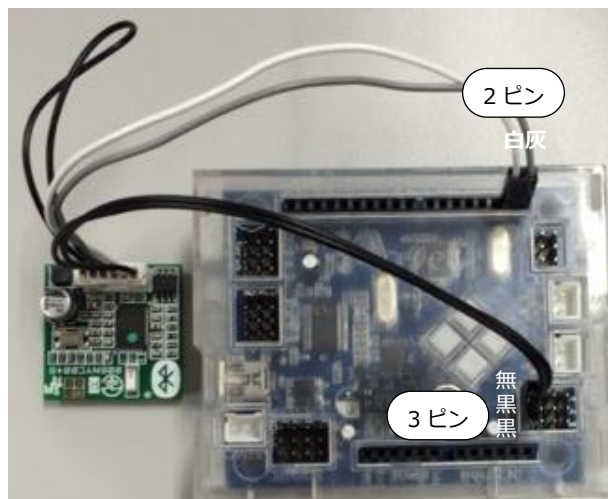
8.3. 接続方法

8.3.1. ブルートゥースモジュール RBT-001 (品番 : 86873)

- ① 別売の『86884 ブルートゥースモジュール RBT-001 接続コード (4 芯 15cm)』を使用してください。
- ② 白のコネクタをブルートゥースモジュール RBT-001 に接続します。
- ③ 2 ピンのコネクタの灰色のコードを Studuino の『0RX』に白のコードを『1TX』に接続します。
- ④ 3 ピンのコネクタの黒のコード 2 本を Studuino の A0～A7 のいずれかの『V』『G』に接続します。



ブルートゥースモジュール
RBT-001



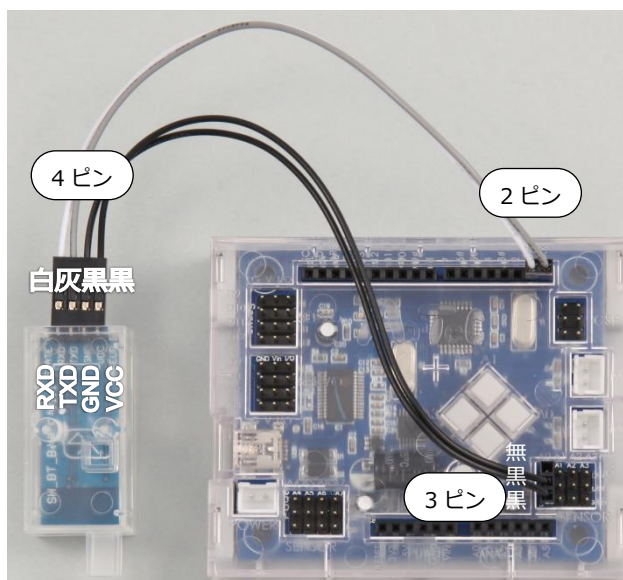
センサー接続コードの向きと接続場所に注意しましょう。

8.3.2. ロボット用ブルートゥースモジュール (86876)

- ① 別売の『86883 ロボット用ブルートゥースモジュール接続コード (4 芯 15cm)』を使用してください。
- ② 4 ピンのコネクタの白のコードをロボット用ブルートゥースモジュールの『RXD』に、黒のコードを『VCC』に接続します。
- ③ 2 ピンのコネクタの灰色のコードを Studuino の『0RX』に白のコードを『1TX』に接続します。
- ④ 3 ピンのコネクタの黒のコード 2 本を Studuino の A0～A7 のいずれかの『V』『G』に接続します。



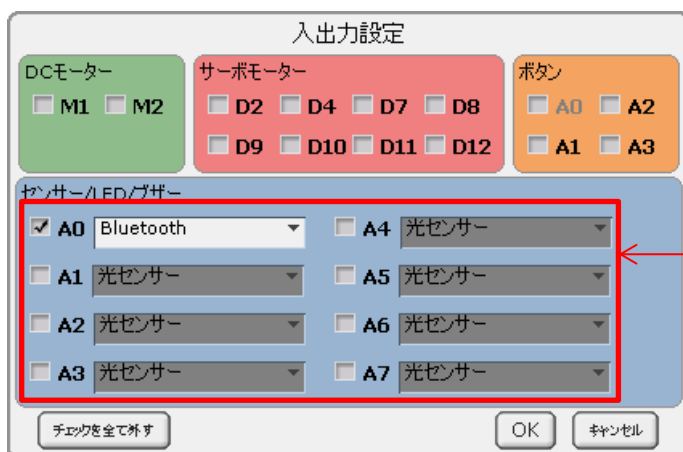
ロボット用
ブルートゥースモジュール



センサー接続コードの向きと接続場所に注意しましょう。

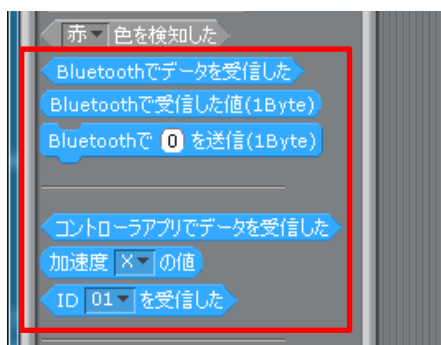
8.4. 設定

- ① ブルートゥースモジュールは、A0～A7 のいずれかに設定することができます。入出力設定ダイアログの「センサー/LED/ブザー」のA0～A7 のいずれかにチェックを入れ、▼をクリックして Bluetooth を選択し、OK をクリックしてください。以下では、Bluetooth を A0 に接続しているものとして説明します。



「Bluetooth」は、A0～A7 で一つだけ選択することができます。

- ② 赤枠の中のブロックが有効になります。



Bluetoothでデータを受信した、Bluetoothで受信した値(1Byte)、Bluetoothで 0 を送信(1Byte)

ブロックは、ブルートゥース機器と 1Byte データを送受信するための基本ブロックになります。コントローラアプリでデータを受信した、加速度 X の値、ID 01 を受信したブロックは、弊社提供のコントローラアプリからのデータを受信するためのブロックです。

8.5. PC、タブレット端末（スマートフォン）等との接続方法（ペアリング）

PC やタブレット端末との通信を行うには、ペアリングが必要です。ペアリングの際に PIN コードの入力を求められますので、以下のコードを入力してください。

RBT-001 (86873)	0000
ロボット用Bluetoothモジュール (86876)	1234

8.6. ロボット用Bluetoothモジュールを使用したプログラミング例

8.6.1. Bluetooth基本ブロックを使用したプログラミング例

下図に Android アプリと Studuino 基板側のプログラムを記します。本作例では、Android 端末から Studuino 基板に対して、LED の ON/OFF 命令とサーボモーターの角度を送信し、Studuino 基板から Android 端末に対して、ジャイロセンサーの値を送信しています。

【Android アプリ】

LED ON/OFF 命令とサーボモーターの角度を Studuino 基板へ送信します

ジャイロセンサーの値を Studuino 基板から受信します

LED ON/OFF		ジャイロセンサーの値	
LED (A1)		加速度X軸	52
LED (A2)		加速度Y軸	37
LED (A3)		加速度Z軸	73
サーボモーター(D9)		角速度X軸	49
		角速度Y軸	50
		角速度Z軸	49

【Studuino 基板のプログラム】

制御スタート

init 関数をコールする

ずっと

getBluetoothCommand 関数をコールする

sendGyroValue 関数をコールする

0.1 秒待つ

init 関数

gyroID を 150 にする

LED A1 を 消灯

LED A2 を 消灯

LED A3 を 消灯

getBluetoothCommand 関数

もし Bluetoothでデータを受信した なら

Bluetoothで受信した値(1Byte) を command に追加する

もし command の長さ = 2 なら

もし command の 1 番目 = 200 なら

サーボモーター D9 を command の 2 番目 度にする

もし command の 1 番目 = 201 なら

LED A1 を 点灯

でなければ

LED A1 を 消灯

もし command の 1 番目 = 202 なら

もし command の 2 番目 = 1 なら

LED A2 を 点灯

でなければ

LED A2 を 消灯

もし command の 1 番目 = 203 なら

もし command の 2 番目 = 1 なら

LED A3 を 点灯

でなければ

LED A3 を 消灯

command の長さ = 0 まで繰り返す

1 番目を command から削除する

sendGyroValue 関数

Bluetoothで gyroID を送信(1Byte)

Bluetoothで ジャイロセンサー X(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Y(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Z(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー X(Gyro) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Y(Gyro) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Z(Gyro) の値 を送信(1Byte)

ジャイロセンサーの値を Android 端末に送信します

LED とサーボモーターの情報を Android 端末から受信します

Studuino 基板のプログラムの各処理を説明します。本作例では、Android 端末から Studuino 基板へ送信するデータは 2Byte 単位で、1Byte 目は命令 ID、2Byte 目は命令データです。Studuino 基板から Android 端末へ送信するデータは、1Byte 単位です。

sendGyroValue 関数は、最初に

ジャイロセンサーの値を示す gyroID (150) を送信した後、ジャイロセンサーの X・Y・Z 軸の加速度と角速度の値 (0~100) を 1Byte 単位でそれぞれ送信します。

sendGyroValue 関数

Bluetoothで gyroID を送信(1Byte)

Bluetoothで ジャイロセンサー X(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Y(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Z(Acc) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー X(Gyro) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Y(Gyro) の値 を送信(1Byte)

Bluetoothで ジャイロセンサー Z(Gyro) の値 を送信(1Byte)

getBluetoothCommand 関数は、Android 端末から受信したデータがあれば、command リストにデータを保存します。データを 2Byte 受信したら、1Byte 目の命令 ID によって処理の場合分けをします。本作例では、命令 ID は次のように決めています。

- ・ 200 : サーボモーターの制御
- ・ 201 : A1 接続の LED の制御
- ・ 202 : A2 接続の LED の制御
- ・ 203 : A3 接続の LED の制御

2Byte 目の命令データを使用して各処理を終了後、command リストを空にします。

8.6.2. コントローラアプリを使用したプログラミング例

コントローラアプリでデータを受信した、**ID 01**を受信した、**加速度 X**の値のブロックの使用例を下図に記します。このブロックは、弊社提供の Android 端末用コントローラアプリを使用することを前提としています。下図のプログラムは、**コントローラアプリでデータを受信した**ブロックを使用しコントローラアプリからデータを受信後、受信したデータの種類によってロボットを制御します。**加速度 X**の値のブロックは、コントローラアプリから送信される Android 端末の傾きセンサーの値（0～100）を取得します。**ID 01**を受信したブロックは、コントローラアプリから送信される ID アイコンが押されているかどうかの状態を取得します。









アプリの操作方法については、[Bluetooth アプリの使用方法](#)をご参照ください。

8.7. テストモード中の動作について

テストモードではコントロールアプリの値を調べることができません。

Bluetooth を用いた（Android 端末と Studuino 基板の）通信と、PC と Studuino 基板の間のテストモードの通信はどちらもシリアル通信を行っており、テストモード中は PC と Studuino 基板の間の通信が優先されてしまうためです。

テストモード中の動作は以下のとおりとなります。

ブロック	テストモード中の動作
  	これらのブロックを調べると、常に「はい」となります。
 	これらのブロックを調べると、常に 0 となります。
	このブロックを実行してもデータの送信を行いません。

9. お問い合わせ先

株式会社アーテック お客様相談窓口

お電話によるお問い合わせ **072-990-5656**

Eメールによるお問い合わせ **support@artec-kk.co.jp**