

電気工作クラブ アルデュイーノ Arduinoマイコン初級

倒立振子で自律ロボットを学ぼう

第6回

配布ソフト

スケッチ名	動作
R3_2019_1_17_touritu	倒立振子
Narita_barance_robot	倒立振子 2
_20181215_IR_MotorControl_2	リモコンで操縦の3輪車

【1】「倒立振子」バランスをとって 静止 させるには

乾電池を新しいものに取り換える

倒立振子の前・後・左・右を示すシールを貼って

スケッチと倒立振子の動きを調べ、メモしていく。

※下のパラメーターに変え、安定して立つかどうか調整、結果をメモする。

体の傾き(角度)	kAngle	100	100	100	150	150	150	
体の傾きの変化率(角速度)	kOmega	500	600	600	600	700	800	900
車輪軸の移動速度	kSpeed	80	80	80	80	80	80	
車輪軸の移動距離	kDistance	0	0	10	10	10	10	
○	×							

33	long powerScale;
34	float Run_Speed = 0, Run_Speed_K = 0;
35	const int kAngle = 150; // 150★【 0~200 まで変えて】
36	const int kOmega = 900; // 900★★★【 0~1200 まで変えて】
37	const long kSpeed = 80; // 80★★★【 0~160 まで変えてみる この値がポイント!!】
38	const long kDistance = 10; // 20★【 0~120 まで変えてみる】

(K*****)のKは「仮に定めた定数」という名まえです

ロボットによって 個々に違いがあります。調整は 昭明はいけません。

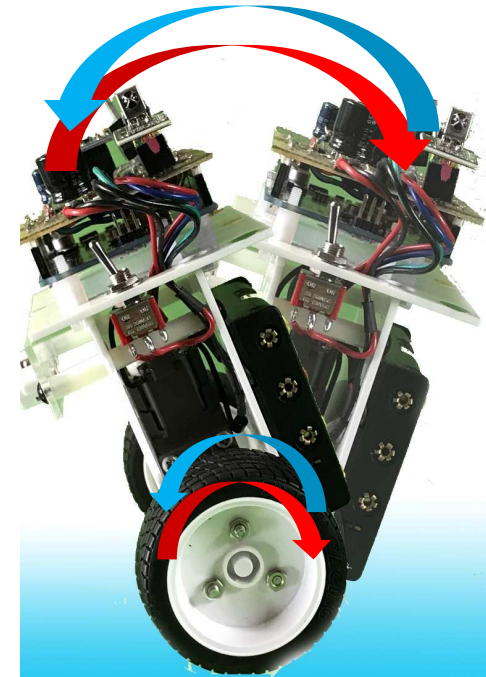
電池の消耗によっても変わってきます。 「100 転び 1 起き」と考えてください。

今回作ったロボットは上の表に示した 値 で安定するようです。

ピタッと止まる 押しても元に戻る …… このような調整を繰り返して完成です。

Omega 傾きの変化率(角速度)

Angle 体の傾き(角度)



体が静止して立つ
車輪は 回らない

体が右に傾く
車輪は右回りに
急に倒れようとする
車輪も速く回る

体が左に傾く
車輪は左回りに
急に倒れようとする
車輪も速く回る

車輪を回すことによ
って倒れないようにし
ています。そのための
最適な【 Power 】
を Arduino Uno が計算
しています

Speed 車輪軸の移動速度

Distance 車輪軸の移動距離

● ギヤボックスから異音が出る

ギヤボックスの歯車がガタガタしないようにスペーサー、プッシュ(黄色のプラ製と金属)が入っていますか?。ガリガリ、ジーのような異音が出ている場合は、油をつけてみます。

● 電池の交換をしてみる

ギヤボックスでモーターへの負荷は小さくなっていますが、大きな電流が流れます。

電池が消耗しても、モーターの回転がスムーズにタイヤへ伝わらないと 立ちません。

● タイヤの回り方をよく観察する

フリーでタイヤを回転させたとき、逆回転していることがあります。片方の配線をチェンジさせれば同じ向きに回るはずですが。



倒立振子を調整するためには、スケッチの数値を変えて調整します。
そのとき
倒立振子の 前 後 左 右 を知るために
色シールを貼っておきます。

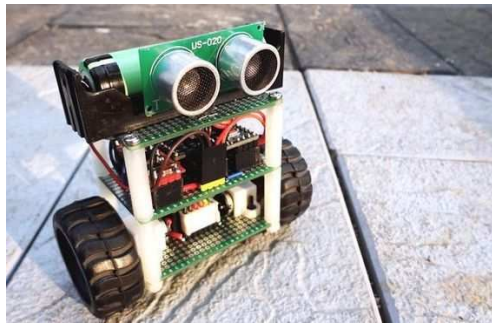
- 白 前
- 黄色 後
- 赤 右モーター側
- 緑 左モーター側

電気工作クラブでは「倒立振り子」を通して、自律ロボットの学習をしてきました。

世界のあちこちで、さまざまな「倒立振り子」が作られています。これからも、「倒立振り子」もっともっと調べていってください。

外国語で書かれているものも、日本語に翻訳してくれるアプリもあります。部品も、インターネットで見つけることができます。

これからは、興味を持ったことやってみみたいと思ったこと、あきらめることなく挑戦していってください。



2018Arduino 初級講座 材料費

品名	単価	数	金額	備考
arduino UNO	390	1	390	USBコード付
ブレッドボード	136	1	136	
L3G4200D	180	1	180	センサー
ギヤ ギヤボックス	570	1	570	
ギヤ スポーツタイヤ	405	1	405	
トランジスタ 3 2N4000 HT7750A	30	3	90	センサー・5V電源用
LM297	110	1	110	モーター制御IC
基板転写フィルム	60	1	60	
LED 4色 各2本	5	8	40	
乾電池 単3 4本×2 2本	23	10	230	モーター・リモコン
電池BOX 単3 4本用	85	2	170	
スイッチ 3p	40	1	40	電源用
赤外線受光器	50	1	50	
ピン端子 4P×2	20	2	40	センサー取り付け
ピン端子 3P	10	3	30	
プラスチック板(70×85×2 mm)	75	3	225	アクリル白(キャスト)
2.54 mmピッチ端子	20	1	20	UNOとの接続
ガラスエポキシ片面基板	45	1	45	
ボックスケース大	108	1	108	
プラスチック容器小	20	1	20	
コンデンサー、チップ抵抗・LED・ケーブル ボールキャスター・ナイロンスペーサー(8) 他雑費。テキスト印刷用紙等				

※この値段には 送料、振込手数料は含まれていません

◆ 「倒立振り子」コントロール R3_2019_1_17_touritu

1	//*****
2	/* 2019.1.17 電気工作教室 Arduino 初級講座
3	// 倒立振り子 スケッチ ver006
4	//*****
5	#include <IRremote.h>
6	#include <SPI.h>
7	#define MAX_TIME 150 // max ms between codes
8	int receiver = 2; //
9	IRrecv irrecv(receiver);
10	decode_results results; byte countS = 0;
11	//long zeroOmega1 = 0;
12	int modeLED1 = 7; //モード 1_LED 緑色
13	int modeLED2 = 8; //モード 2_LED 黄色
14	int modeLED3 = 9; //モード 3_LED 赤色
15	int RMA = 14; //A0 : 右 A
16	int RMB = 15; //A1 : 右 B
17	int REN = 5; //右イネーブル
18	int LMA = 16; //A2 : 左 A
19	int LMB = 17; //A3 : 左 B
20	int LEN = 6; //左イネーブル
21	int recOmega1[10];
22	int omega1 = 0;
23	int power;
24	int fwdBck = 0;
25	int ry;
26	long vE5 = 0;
27	long xE5 = 0;
28	long R;
29	long theta1 = 0;
30	long sumPower = 0;
31	long sumSumP = 0;
32	long lastPressTime = 0;
33	long powerScale;
34	float Run_Speed = 0, Run_Speed_K = 0;
35	const int kAngle = 150; //★【倒れてしまう場合 0~200 まで変えて】
36	const int kOmega = 600; //★★★【倒れてしまう場合 0~700 まで変えて】
37	const long kSpeed = 80; //★★★【倒れてしまう場合 0~100 まで変えてみる この値がポイント!!】
38	const long kDistance = 10; //★【倒れてしまう場合 0~120 まで変えてみる】
39	boolean right = false;
40	boolean left = false;
41	
42	void L3GD20_write(byte reg, byte val) {
43	digitalWrite(10, LOW);
44	SPI.transfer(reg);
45	SPI.transfer(val);
46	digitalWrite(10, HIGH);
47	}
48	byte L3GD20_read(byte reg) {

49	byte ret = 0;
50	digitalWrite(10, LOW);
51	SPI.transfer(reg 0x80);
52	ret = SPI.transfer(0);
53	digitalWrite(10, HIGH);
54	return ret;
55	}
56	*****void setup ()*****
57	void setup () {
58	Serial .begin(115200);
59	irrecv.enableIRIn();// 受信を開始する
60	
61	pinMode(modeLED1, OUTPUT);
62	pinMode(modeLED2, OUTPUT);
63	pinMode(modeLED3, OUTPUT);
64	pinMode(RMA, OUTPUT);
65	pinMode(RMB, OUTPUT);
66	pinMode(LMA, OUTPUT);
67	pinMode(LMB, OUTPUT);
68	pinMode(LMA, OUTPUT);
69	pinMode(LMB, OUTPUT);
70	pinMode(LEN, OUTPUT);
71	for (int i = 0 ; i < 10 ; i++) {
72	recOmegaI[i] = 0;
73	}
74	pinMode(10, OUTPUT);
75	digitalWrite(10, HIGH);
76	SPI.begin();
77	SPI.setBitOrder(MSBFIRST);
78	SPI.setDataMode(SPI_MODE3);
79	SPI.setClockDivider(SPI_CLOCK_DIV2);
80	L3GD20_write(0x20, B11001111);
81	L3GD20_write(0x23, B00000000);
82	delay(300);
83	*****void loop ()*****
84	void loop () {
85	chkAndCtl();
86	if (irrecv.decode(&results)) { // 赤外線リモコンのボタンが放された時の処理
87	irrecv.resume(); // Receive the next value
88	lastPressTime = millis();
89	Run_Speed_K = 0; // リモコンボタンが放された時 0 に戻す
90	irrecv.resume(); // Receive the next value 一旦リセット
91	}
92	if (results.value == 0x35CA6A95) // 右へ
93	{
94	right = true;
95	}
96	else if (results.value == 0x35CAEA15) { // 左へ
97	
98	left = true;
99	}

100	if (results.value == 0x35CA1AE5) // 前へ
101	Run_Speed_K = -15;
102	Run_Speed = Run_Speed + Run_Speed_K; // 十字キーの “↑キー” を押してる間、-15 を加算し続けます。
103	power = Run_Speed;
104	}
105	else if (results.value == 0x35CA9A65) // 後ろへ
106	Run_Speed_K = 15;
107	Run_Speed = Run_Speed + Run_Speed_K; // 十字キーの “↓キー” を押してる間、-15 を加算し続けます。
108	power = -Run_Speed;
109	}
110	
111	if (right == true) {
112	digitalWrite (modeLED1, LOW);
113	digitalWrite (modeLED2, LOW);
114	digitalWrite (modeLED3, HIGH); // 赤点灯
115	analogWrite(REN, 140); // ★★ORG140
116	digitalWrite(RMA, HIGH);
117	digitalWrite(RMB, LOW);
118	analogWrite(LEN, 140); // ★★ORG140
119	digitalWrite(LMA, LOW);
120	digitalWrite(LMB, HIGH);
121	delay(30);
122	} else if (left == true) {
123	digitalWrite (modeLED1, HIGH); // 青 or 緑点灯
124	digitalWrite (modeLED2, LOW);
125	digitalWrite (modeLED3, LOW);
126	analogWrite(REN, 140); // ★★ORG140
127	digitalWrite(RMA, LOW);
128	digitalWrite(RMB, HIGH);
129	analogWrite(LEN, 140); // ★★ORG140
130	digitalWrite(LMA, HIGH);
131	digitalWrite(LMB, LOW);
132	delay(30);
133	
134	// 以下バランスをとって立ったり前に進む時のモーター制御
135	} else if (power > 0) {
136	// Serial.print(" power f = "); Serial.println(power);
137	digitalWrite (modeLED1, HIGH); // 緑点灯
138	digitalWrite (modeLED2, LOW);
139	digitalWrite (modeLED3, LOW);
140	analogWrite(REN, power); // 正転時 右モーターのパワー
141	digitalWrite(RMA, HIGH);
142	digitalWrite(RMB, LOW);
143	analogWrite(LEN, power); // 正転時 左モーターのパワー
144	digitalWrite(LMA, HIGH);
145	digitalWrite(LMB, LOW);
146	}
147	// 以下バランスをとって立ったり後に進む時のモーター制御
148	else {

```

149 // Serial.print(" power B = "); Serial.println(power);
150 digitalWrite(modeLED1, LOW);
151 digitalWrite(modeLED2, LOW);
152 digitalWrite(modeLED3, HIGH);//
153 analogWrite(REN, -power);// 正転時 右モーターのパワー
154 digitalWrite(RMA, LOW);
155 digitalWrite(RMB, HIGH);
156 analogWrite(Len, -power);// 正転時 右モーターのパワー
157 digitalWrite(LMA, LOW);
158 digitalWrite(LMB, HIGH);
159 }
160 // delayMicroseconds(3600);
161 }
162 void chkAndCtl0 {
163 // omegaI = 0;
164 // for ( i = 0 ; i < 10 ; i++ ) {
165 // omegaI = omegaI + analogRead(A5) * zeroOmegaI;
166 // delayMicroseconds(10);
167 // }
168 // omegaI = omegaI / 10;
169 R = 0;
170 results.value = 0;
171 right = false;
172 left = false;
173 for ( int i = 0 ; i < 45 ; i++ ) {
174 ry = ( (L3GD20_read(0x2B) << 8) | L3GD20_read(0x2A) );
175 // ry = ( (L3GD20_read(0x29) << 8) | L3GD20_read(0x28) );//★★★【センサーの取付
// 方向を替えた場合】
176 R = R + ry; //★★★【 R = R - ry で正逆反転する】
177 delayMicroseconds(90);// ★★★ORG 90 を変えてみる (45~100)
178 }
179 omegaI = R * 0.00875 / 45 ;//★★★【 +0~ +10 で調整 ORG==0】
180 // Serial.print(" omegaI = "); Serial.println(omegaI);
181 if ( abs(omegaI) < 2 ) {
182 omegaI = 0;
183 }
184 recOmegaI[0] = omegaI;
185 thetaI = thetaI + omegaI;
186 countS = 0;
187 for ( int i = 0 ; i < 10 ; i++ ) {
188 if ( abs(recOmegaI[i]) < 4 ) {
189 countS++;
190 }
191 }
192 if ( countS > 9 ) {
193 thetaI = 0;
194 vE5 = 0;
195 xE5 = 0;
196 sumPower = 0;
197 sumSumP = 0;
198 }

```

```

199 for ( int i = 9 ; i > 0 ; i-- ) {
200 recOmegaI[i] = recOmegaI[i - 1];
201 }
202 powerScale = ( kAngle * thetaI / 100 ) + ( kOmega * omegaI / 100 ) + ( kSpeed * vE5 / 1000 )
+ ( kDistance * xE5 / 1000 );
203 power = max ( min ( 95 * powerScale / 100 , 255 ) , -255 );//MAX=2つの数値のうち、大きい
// ほうの値を返す
204 sumPower = sumPower + power + 1;
205 //★★【前後に移動する場合、前行「power」の後に、適当な数値 (1~5程度の整数)】
206 sumSumP = sumSumP + sumPower;
207 vE5 = sumPower;
208 xE5 = sumSumP / 1000;
209 }
210 //*****

```

- 故障したり、わからないこと、困ったときは
 - ※ 090-7869-3680 【柴田】
 - または、事務室 0532-41-3330
- LINE ID検索 ja2guj グループ「ゆめたまご」
 - グループ 参加歓迎、退会自由です。

■ 2月以降 月1回 以下のように
「Andoroid 倒立振子」の勉強会を行います

2月16日(土) 午前10時から11時30分
3月16日(土) 同上

- ◆ 「倒立振子」がうまく動作しない
 - ◆ 改造 故障診断 修理をしたい
 - ◆ Arduino に関する事など
- 自主参加です。途中からでも OK です。

