

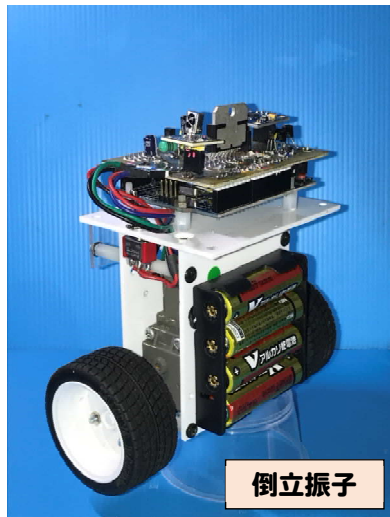
# 電気工作クラブ アルデュイーノ Arduinoマイコン初級

倒立振子で自律ロボットを学ぼう

第5回

## ◆ 配布ソフト

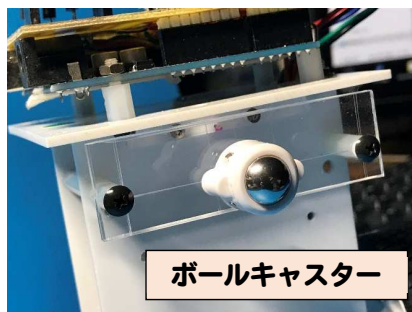
スケッチ名	動作
_20181206_IR_MotorControl_0	リモコンで操縦 3輪車スケッチ
_2018_12_07_touritu002	倒立振子のスケッチ
IRremote_test1	赤外線リモコンのコード読み取り
IRremote_test2	赤外線リモコンのコード読み取り
ChecL3GD20	センサーのチェック
Irremote.zip SPI_L3GD20.zip	Arduino Uno ライブラリー



倒立振子



リモコン三輪車



ボールキャスター

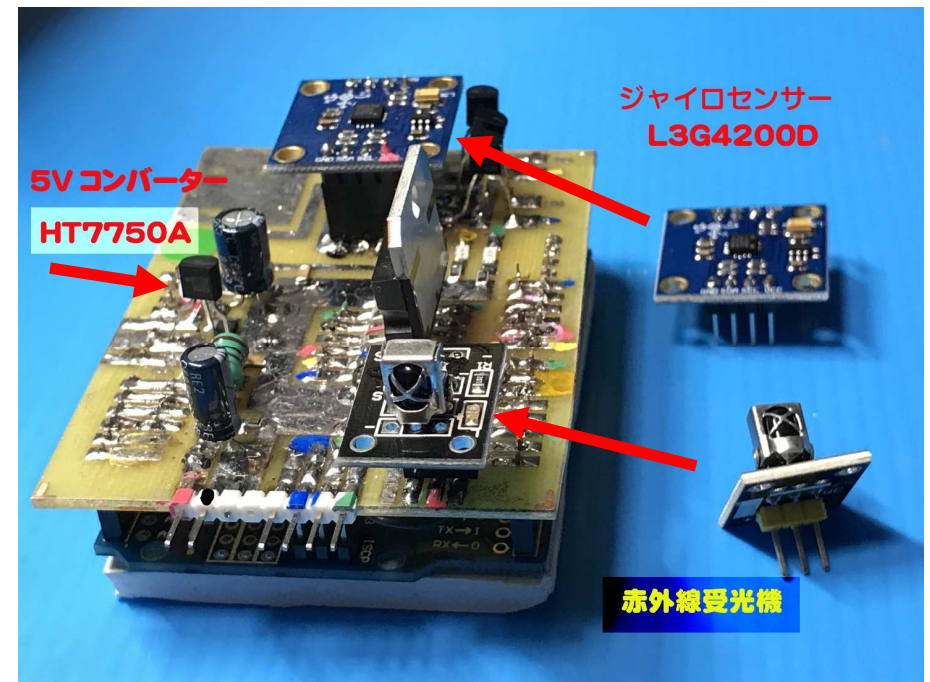
左下写真のようにキャスターを付けます。

今回は、Arduino Unoのスケッチを使って、倒立振子が自分で立つようになるか。

三輪車をリモコンで操縦できるか。

きょうは、ロボットが生まれる日です。

## 5V回路、ジャイロセンサー、リモコン受光素子



5Vコンバーター  
HT7750A

ジャイロセンサー  
L3G4200D

赤外線受光機

## 5V昇圧コンバーター回路



5V電源

乾電池電源

HT7750A

モーターが回ることによって乾電池電源の電圧が下がります。この回路によって、いつでも正確な5Vを取り出すことができます。

## L3G4200D (角速度センサー)

小さなチップに

3軸 (x, y, z 軸 Gyro Senso) 角速度センサー、  
角速度からマイコンを使って角度を求める

通信は SPI (SCLK, MISO, MOSI, SS の信号線 4 本) で行います。  
信号線は使う Arduino Uno の端子が決まっています。

## 動作をチェックするプログラム

## ChecL3GD20. ino

カーナビカメラの手ぶれ補正

ゲームコントローラ、

ロボット姿勢制御、

ヘリコプタ・ドローンの姿勢制御

などに使われる

★ SPI ライブラリーが必要★

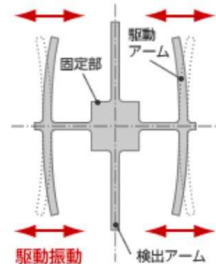
## 赤外線受光素子

★Arduino で赤外線を使うためのライブラリー★

## IRremote. zip

## ライブラリーの確認とインストール

Arduino IDE で 「スケッチ」 → 「ライブラリーをインクルード」  
ライブラリー一覧に 「SPI」 「IRremote」 があるかどうか確認でき  
ます。無かったら 「zip 形式のライブラリーをインストール」 で  
組み込むことができます。



```
void loop () {
  X = L3GD20_read(0x29);
  x = X = (X << 8) | L3GD20_read(0x28);
  Y = L3GD20_read(0x2B);
  y = Y = (Y << 8) | L3GD20_read(0x2A);
  Z = L3GD20_read(0x2D);
  z = Z = (Z << 8) | L3GD20_read(0x2C);
  x *= 0.00875; // センサー値から角速度に変換した値 0.00875 は分解能
  y *= 0.00875;
  z *= 0.00875;
  // Serial .print(" millis=");
  // Serial .print(millis());
  // Serial .print(" x=");
  // Serial .print(x);
  Serial .print(" y= ");
  Serial .println(y);
  // Serial .print(" Z=");
  // Serial .println(z); if ( y > 3 ) {
    digitalWrite (modeLED1, LOW);
    digitalWrite (modeLED2, LOW);
    digitalWrite (modeLED3, HIGH);//LED は赤色
  } if ( y < 3 ) {
    digitalWrite (modeLED1, HIGH);//LED は緑色
    digitalWrite (modeLED2, LOW);
    digitalWrite (modeLED3, LOW);
  }
  delay(500); //0.5 秒
}
```

## シリアルモニター画面

```
y= -0.25
y= -0.17
y= -0.12
y= -0.09
y= -0.13
y= -0.03
y= -0.38
y= 0.02
y= -0.15
y= 0.03
y= 0.11
y= 0.06
...
```

X, Z は表示させてない

y が 3 以上 赤

y が 3 以下 緑

## 「記号の説明」

「X, Y, Z」 X軸 Y軸 Z軸

「x, y, z」 計算結果 角度

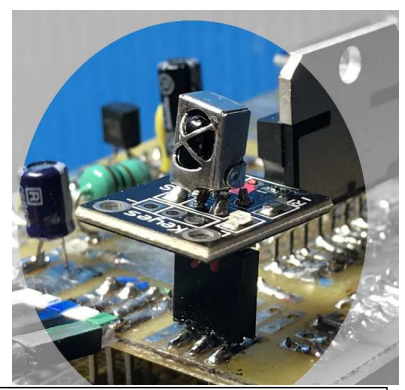
「0x28 0x29 0x2A 0x2B 0x2C 0x2D」

センサー素子の中にある データの入れ物場所

「Serial」 = シリアル データ

「Serial .print」 シリアルモニターに出力される

# リモートコントロールのための受光素子



受光素子は3ピンソケットに付けます

## リモコンボタンを調べるスケッチ

リモコンボタンを押すとコードを読み取ることができます

### ①【IRremote\_test1】

```
0011010111001010100000001111111 (AC 53 1 FE)
0011010111001010100000001111111 (AC 53 1 FE)
0011010111001010010000001011111 (AC 53 2 FD)
0011010111001010010000001011111 (AC 53 2 FD)
0011010111001010010000001011111 (AC 53 2 FD)
0011010111001010010000001011111 (AC 53 2 FD)
0011010111001010011000001001111 (AC 53 6 F9)
0011010111001010011000001001111 (AC 53 6 F9)
```

### ②IRremote\_test2では「シリアルモニター」に「0x\*\*\*\*\*」のような8けたのデータが表示されます。

```
35CA1AE5
35CA1AE5
35CA1AE5
35CA1AE5
35CA1AE5
35CA1AE5
35CA1AE5
35CA32CD
35CA32CD
35CA9A65
35CA9A65
35CA9A65
35CAEA15
35CAEA15
```

信号は8桁



ボタン	信号	意味
青	35CACC33	停止
	F50ACC33	
赤	35CACC33	自律
	F50A2CD3	
緑	35CACC33	手動
	F50AAC53	
黄	35CACC33	倒立
	F50A6C93	
↑	35CA1AE5	前へ
→	35CA6A95	右へ
←	35CAEA15	左へ
↓	35CA9A65	バック
決定	35CA32CD	停止
電源	35CA38C7	終了

ボタンを押す事により値が変わります。

8けたの数字、アルファベットを読み取ってメモしてください。

### 2進数 と 16進数 について

```
0000=0 0001=1 0010=2 0011=3 0100=4 0101=5
0110=6 0111=7 1000=8 1001=9
1010=A 1011=B 1100=C 1101=D 1110=E 1111=F
0000 0001 =01 1111 0001=F1 1010 1010=AA
```

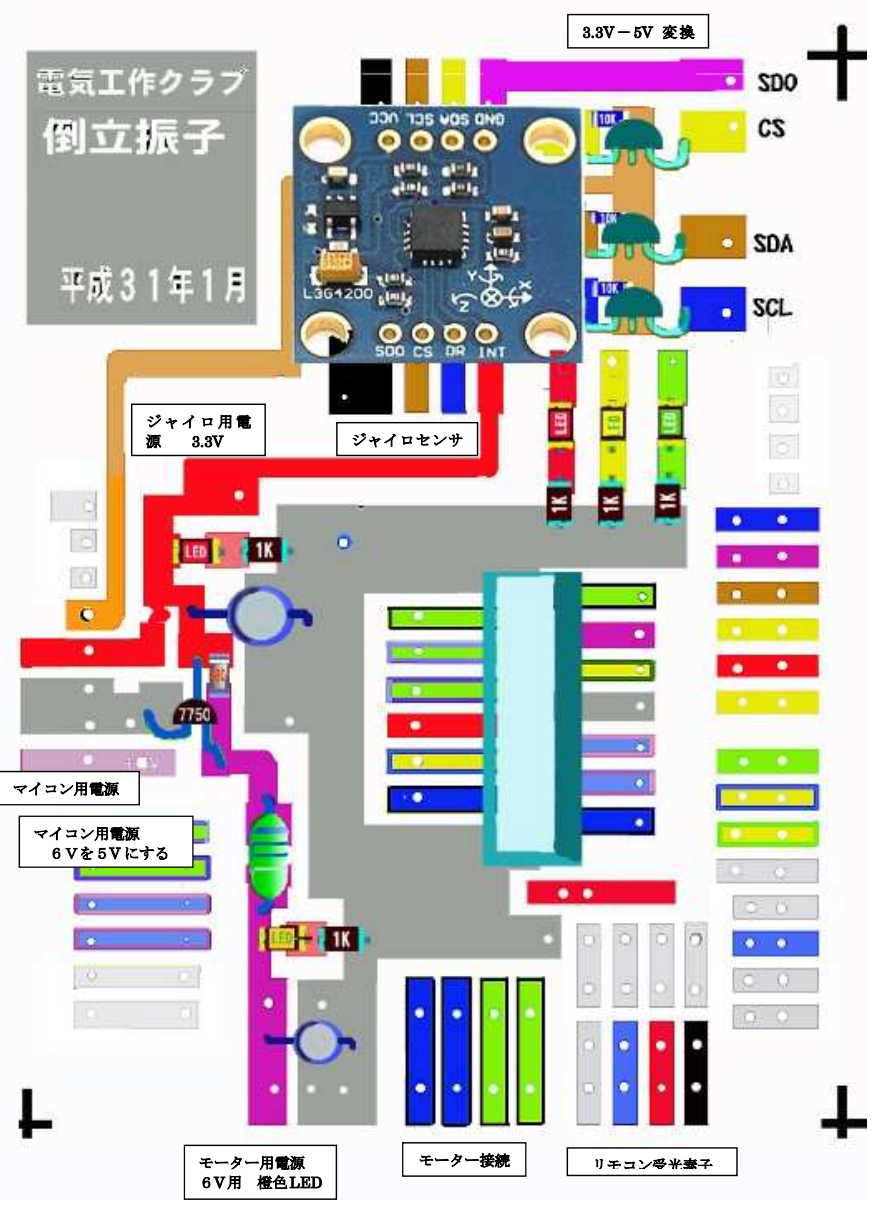
リモコンから出る赤外線 発光=「1」消灯「0」の信号です  
「AA 5A 8F」と表示された信号は 16進数「HEX」の値です

1010 1010 は AA ではなく 0xAA と表します



電気工作クラブ  
倒立振り子

平成31年1月



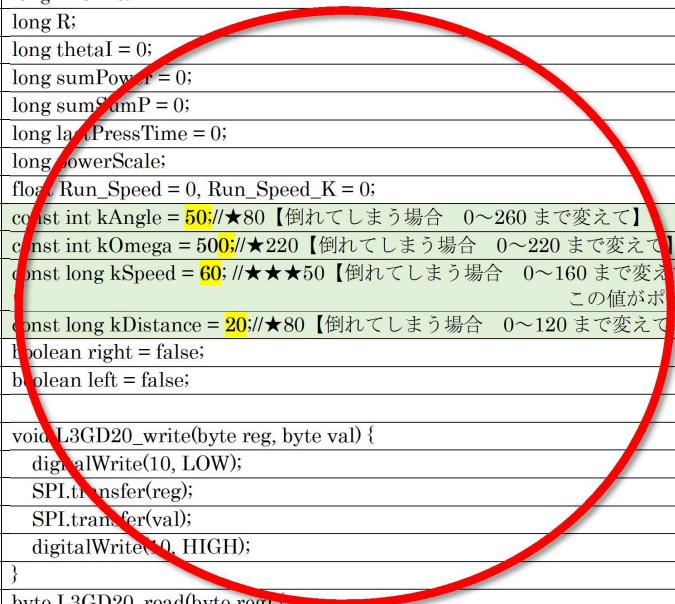
◆ 「倒立振り子」コントロール

2018\_12\_07\_touritu002

```

1 //*****
2 /* 2018.12.7 電気工作教室 Arduino 初級講座
3 // 倒立振り子 スケッチ ver002
4 //*****
5 #include <IRremote.h>
6 #include <SPI.h>
7 #define MAX_TIME 150 // max ms between codes
8 int receiver = 2; //D2
9 IRrecv irrecv(receiver);
10 decode_results results; byte countS = 0;
11 //long zeroOmegaI = 0;
12 int modeLED1 = 7; //モード 1_LED 青又は緑
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 recOmegaI[10];
22 int omegaI = 0;
23 int power;
24 int fwdBck = 0;
25 int ry;
26 long vE5 = 0;
27 long xE5 = 0;
28 long R;
29 long thetaI = 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 = 50; //★80 【倒れてしまう場合 0~260 まで変えて】
36 const int kOmega = 500; //★220 【倒れてしまう場合 0~220 まで変えて】
37 const long kSpeed = 60; //★★★50 【倒れてしまう場合 0~160 まで変えてみる
   この値がポイント!!】
38 const long kDistance = 20; //★80 【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) {

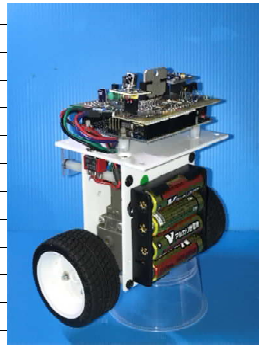
```



```

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

```



暴走します ケガなどに注意

```

99   if (results.value == 0x35CA1A85) {
100     Run_Speed_K = -15;
101     Run_Speed = Run_Speed + Run_Speed_K; // ↑キーの“↑キー”を押してる間、-15を
      加算し続けます。
102     power = Run_Speed;
103   }
104   else if (results.value == 0x35CA9A65) {
105     Run_Speed_K = 15;
106     Run_Speed = Run_Speed + Run_Speed_K; // ↓キーの“↓キー”を押してる間、-15を
      加算し続けます。
107     power = Run_Speed;
108   }
109
110   if (right == true) {
111     digitalWrite(modeLED1, LOW);
112     digitalWrite(modeLED2, LOW);
113     digitalWrite(modeLED3, HIGH); // 赤点灯
114     analogWrite(REN, 140); // ★★ORG140
115     digitalWrite(RMA, HIGH);
116     digitalWrite(RMB, LOW);
117     analogWrite(LEN, 140); // ★★ORG140
118     digitalWrite(LMA, LOW);
119     digitalWrite(LMB, HIGH);
120     delay(30); //
121   } else if (left == true) {
122     digitalWrite(modeLED1, HIGH); // 緑点灯
123     digitalWrite(modeLED2, LOW);
124     digitalWrite(modeLED3, LOW);
125     analogWrite(REN, 140); // ★★ORG140
126     digitalWrite(RMA, LOW);
127     digitalWrite(RMB, HIGH);
128     analogWrite(LEN, 140); // ★★ORG140
129     digitalWrite(LMA, HIGH);
130     digitalWrite(LMB, LOW);
131     delay(30);
132   }
133   //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
134   } else if (power > 0) {
135     digitalWrite(modeLED1, HIGH); // 緑点灯
136     digitalWrite(modeLED2, LOW);
137     digitalWrite(modeLED3, LOW);
138     analogWrite(REN, power); // 正転時 右モーターのパワー
139     digitalWrite(RMA, HIGH);
140     digitalWrite(RMB, LOW);
141     analogWrite(LEN, power); // 正転時 左モーターのパワー
142     digitalWrite(LMA, HIGH);
143     digitalWrite(LMB, LOW);
144   }
145   //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
146   else if (power < 0) {
147     digitalWrite(modeLED1, LOW);

```

```

148 digitalWrite(modeLED2, LOW);
149 digitalWrite(modeLED3, HIGH); //赤点灯
150 analogWrite(REN, - power); // 正転時 右モーターのパワー
151 digitalWrite(RMA, LOW);
152 digitalWrite(RMB, HIGH);
153 analogWrite(LEN, - power); // 正転時 右モーターのパワー
154 digitalWrite(LMA, LOW);
155 digitalWrite(LMB, HIGH);
156 }
157 // delayMicroseconds(3600);
158 }
159 void chkAndCtl0 {
160 // omegaI = 0;
161 // for ( i = 0 ; i < 10 ; i++ ) {
162 // omegaI = omegaI + analogRead(A5) - zeroOmegaI;
163 // delayMicroseconds(10);
164 // }
165 // omegaI = omegaI / 10;
166 R = 0;
167 results.value = 0;
168 right = false;
169 left = false;
170 for ( int i = 0 ; i < 45 ; i++ ) {
171 ry = ( (L3GD20_read(0x2B) << 8) | L3GD20_read(0x2A) );
172 // ry = ( (L3GD20_read(0x29) << 8) | L3GD20_read(0x28) ); //★★ センサーの取付
// 方向を替えた場合
173 R = R - ry; //★★ 【 R = R + ry で正逆反転する】
174 delayMicroseconds(90);
175 }
176 megaI = R * 0.00875 / 45 + 4 ; //★★★ 【 +0 ~ +10 で調整 ORG=0】
177 if ( abs(omegaI) < 2 ) {
178 omegaI = 0;
179 }
180 recOmegaI[0] = omegaI;
181 thetaI = thetaI + omegaI;
182 countS = 0;
183 for ( int i = 0 ; i < 10 ; i++ ) {
184 if ( abs(recOmegaI[i]) < 4 ) {
185 countS++;
186 }
187 }
188 if ( countS > 9 ) {
189 thetaI = 0;
190 vE5 = 0;
191 xE5 = 0;
192 sumPower = 0;
193 sumSumP = 0;
194 }
195 for ( int i = 9 ; i > 0 ; i-- ) {
196 recOmegaI[i] = recOmegaI[i - 1];
197 }

```

```

198 powerScale = (kAngle * thetaI / 100) + (kOmega * omegaI / 100) + (kSpeed * vE5 / 1000)
+ (kDistance * xE5 / 1000);
199 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
200 sumPower = sumPower + power + 1;
201 //★★ 【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)】
202 sumSumP = sumSumP + sumPower;
203 vE5 = sumPower;
204 xE5 = sumSumP / 1300;
205 }
206
207
208
209
210

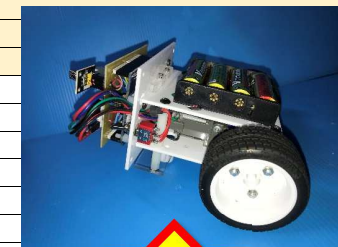
```

◆ 赤外線リモコンで操作する 3 輪車  
20181206\_IR\_MotorControl\_0

```

1 //*****
2 /* 2018.12.07 電気工作教室 Arduino 初級講座
3 // リモコンカー スケッチ ver0
4 //*****
5 #include <IRremote.h>
6 int receiver = 2; //デジタル 2 番ピン
7 IRrecv irrecv(receiver);
8 decode_results results;
9 #define MAX_TIME 150
10 int modeLED1 = 7; //LED 青又は緑
11 int modeLED2 = 8; //LED 黄色
12 int modeLED3 = 9; //LED 赤色
13 int RMA = 14; //A0 : 右モータードライバーA 端子へ
14 int RMB = 15; //A1 : 右モータードライバーB 端子へ
15 int REN = 5; //右モータードライバーイネーブル端子へ
16 int LMA = 16; //A2 : 左モータードライバーA 端子へ
17 int LMB = 17; //A3 : 左モータードライバーB 端子へ
18 int LEN = 6; //左モータードライバーイネーブル端子へ
19 float Run_Speed = 0, Run_Speed_K = 0;
20 int power;
21 unsigned long now = millis();
22 long lastPressTime = 0;
23 int state = HIGH;
24
25 void setup() {
26 // Serial.begin(9600);
27 irrecv.enableIRIn(); // 受信を開始する
28 pinMode(modeLED1, OUTPUT);
29 pinMode(modeLED2, OUTPUT);
30 pinMode(modeLED3, OUTPUT);
31 pinMode(REN, OUTPUT);
32 pinMode(RMA, OUTPUT);

```



暴走します ケガなどに注意



33	pinMode(RMB, OUTPUT);
34	pinMode(LEN, OUTPUT);
35	pinMode(LMA, OUTPUT);
36	pinMode(LMB, OUTPUT);
37	}
38	
39	void loop() {
40	if (irrecv.decode(&results)) { // 赤外線リモコンのボタンが放された時の処理
41	irrecv.resume(); // Receive the next value
42	Run_Speed_K = 0; // リモコンボタンが放された時 0 に戻す
43	if (state == HIGH) {
44	state = LOW;
45	}
46	lastPressTime = millis();
47	irrecv.resume(); // Receive the next value 一旦リセット
48	}
49	if (state == LOW && millis() - lastPressTime > MAX_TIME) {
50	state = HIGH;
51	}
52	//前進
53	if (results.value == 0x35CA9A65) { //構造上前後を入れ替えてある
54	analogWrite(REN, power); // 前進時 右モーターのパワー
55	analogWrite(LEN, power); // 前進時 左モーターのパワー
56	digitalWrite (modeLED1, LOW); //
57	digitalWrite (modeLED2, LOW);
58	digitalWrite (modeLED3, HIGH); //赤色点灯
59	digitalWrite(RMA, LOW);
60	digitalWrite(RMB, HIGH);
61	digitalWrite(LMA, LOW);
62	digitalWrite(LMB, HIGH);
63	Serial .print(" power="); Serial .println(power);
64	Run_Speed_K = 15;
65	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。
66	power = 10 + Run_Speed;
67	// results.value = 0;
68	delay(100);
69	}
70	
71	// 後退
72	if (results.value == 0x35CA1AE5) {
73	analogWrite(REN, power); // 後退時 右モーターのパワー
74	analogWrite(LEN, power); // 後退時 左モーターのパワー
75	digitalWrite (modeLED1, HIGH); //青色点灯
76	digitalWrite (modeLED2, LOW); //黄色
77	digitalWrite (modeLED3, LOW);
78	digitalWrite(RMA, HIGH);
79	digitalWrite(RMB, LOW);
80	digitalWrite(LMA, HIGH);
81	digitalWrite(LMB, LOW);
82	

83	Serial .print(" power="); Serial .println(power);
84	Run_Speed_K = 15;
85	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。
86	power = 10 + Run_Speed;
87	// results.value = 0;
88	delay(100);
89	}
90	// 右に曲がる
91	if (results.value == 0x35CA6A95) {
92	digitalWrite (modeLED1, LOW);
93	digitalWrite (modeLED2, HIGH);
94	digitalWrite (modeLED3, HIGH); //赤点灯
95	analogWrite( REN, 0); //★★
96	digitalWrite(RMA, LOW);
97	digitalWrite(RMB, HIGH);
98	analogWrite(LEN, 80 ); // ★★ORG140
99	digitalWrite(LMA, LOW);
100	digitalWrite(LMB, HIGH);
101	delay(100);
102	// 0.1秒
103	}
104	
105	//左に曲がる
106	if (results.value == 0x35CAEA15) {
107	digitalWrite (modeLED1, HIGH); //青 or 緑点灯
108	digitalWrite (modeLED2, HIGH);
109	digitalWrite (modeLED3, LOW);
110	analogWrite( REN, 80 ); // ★★ORG140
111	digitalWrite(RMA, LOW);
112	digitalWrite(RMB, HIGH);
113	analogWrite( LEN, 0); //★★
114	digitalWrite(LMA, LOW);
115	digitalWrite(LMB, HIGH);
116	delay(100);
117	// 0.1秒
118	}
119	
120	if (results.value == 0x35CA32CD) { //停止
121	digitalWrite(RMA, HIGH);
122	digitalWrite(RMB, HIGH);
123	digitalWrite(LMA, HIGH);
124	digitalWrite(LMB, HIGH);
125	analogWrite(REN, 255);
126	analogWrite(LEN, 255);
127	}
128	}