

この資料は電車のおもちゃを TV のリモコンで動かせるように改造した内容をまとめたものです。ファームウェアの部分を中心に動作が不安定な部分もありますが、工作に興味を持つ人の参考用に公開します。

●モーター電源

モータードライバ IC の使い方の問題か、モーター電源が 1.5V では動作しなかったため、3V の電源電圧で動作させてます。モーター制御で使用する PWM 信号のデューティは最大約 50% で動作させていますが、もしかしたらモーター寿命という点ではよろしくないかもしれません。

●マイコン関係

・マイコンは Arduino のブートローダー書き込み済みの Atmega328P 3.3V 内蔵 8MHz 発振版を使用してます。UART はデバッグ時に使用しますが、115Kbps では文字化けするため 9600bps に速度を下げています。

・マイコン電源が 3.3V のため、プログラムの書き込みは TTL-232R-3V3 など 3.3V の IO 電圧に対応したケーブルを使う必要があります。

・ArduinoIDE での書き込み用デバイス設定は「Arduino Pro or Pro Mini」に設定しています。

・リモコン信号受信は Arduino の IRremote ライブラリを使用しています。信号受信の確認をポーリングで処理しているためか、信号を複数回受信することがあります。（1 回ボタンを押したただけなのに 2 回押したと判断する）リモコンのリポート信号の処理が関係しているかもしれませんが未確認です。

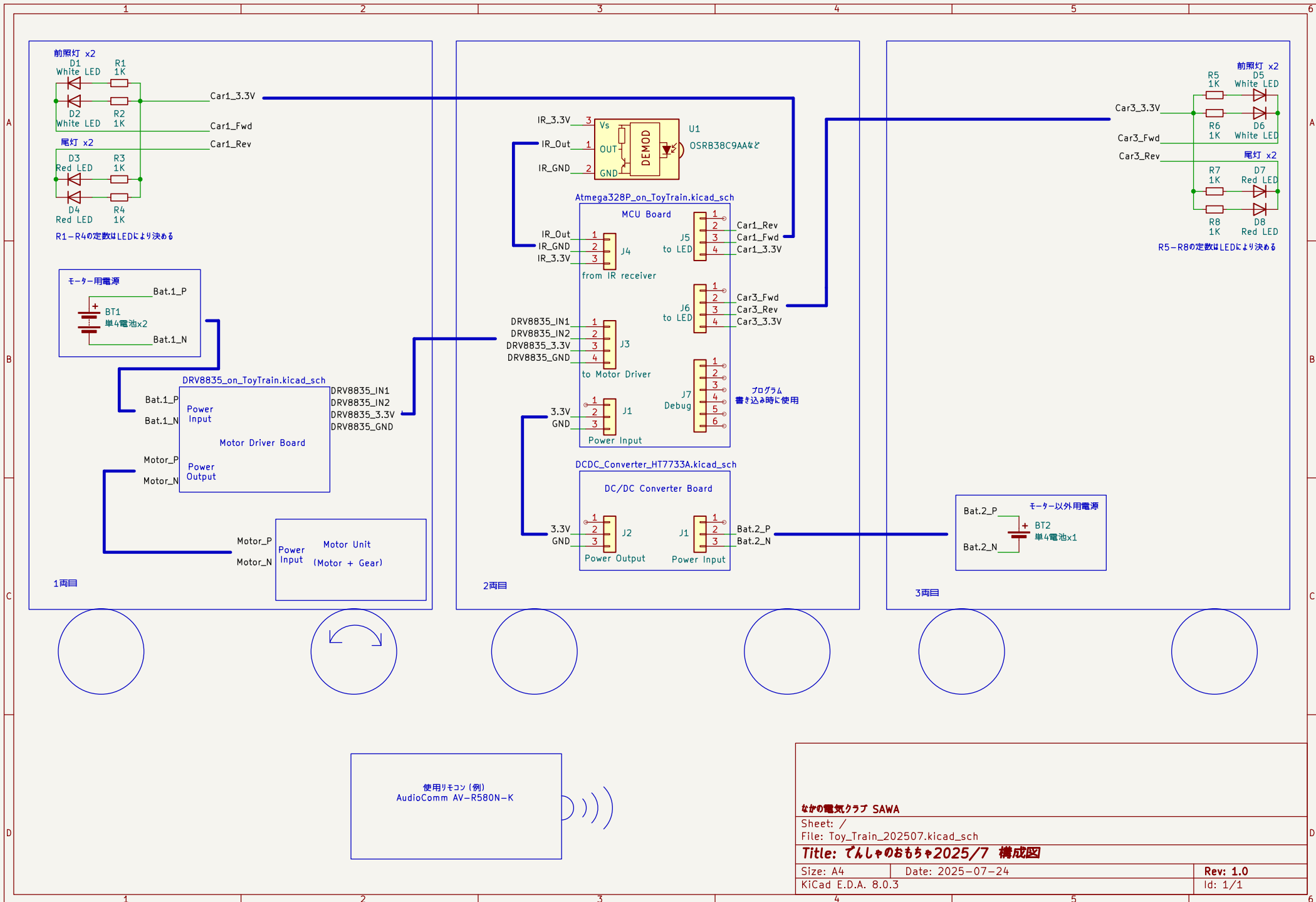
・リモコンは TV 用のマルチリモコン(オーム電機 AV-R580N-K)を使用しています。フォーマットは NEC で、カスタマーコードが 0x50 で動作する設定にしています。

●その他

・基板データの部品配置や配線は、ユニバーサル基板上に配線した試作品と同じように配置してるため、最適な配置ではない部分があります。

・電源は配線の取り回しを考えてモーター用とその他用に分けていますが、一つにまとめるのがシンプルで正解かも知れません。

以上です



なかの電気クラブ SAWA

Sheet: /
File: Toy_Train_202507.kicad_sch

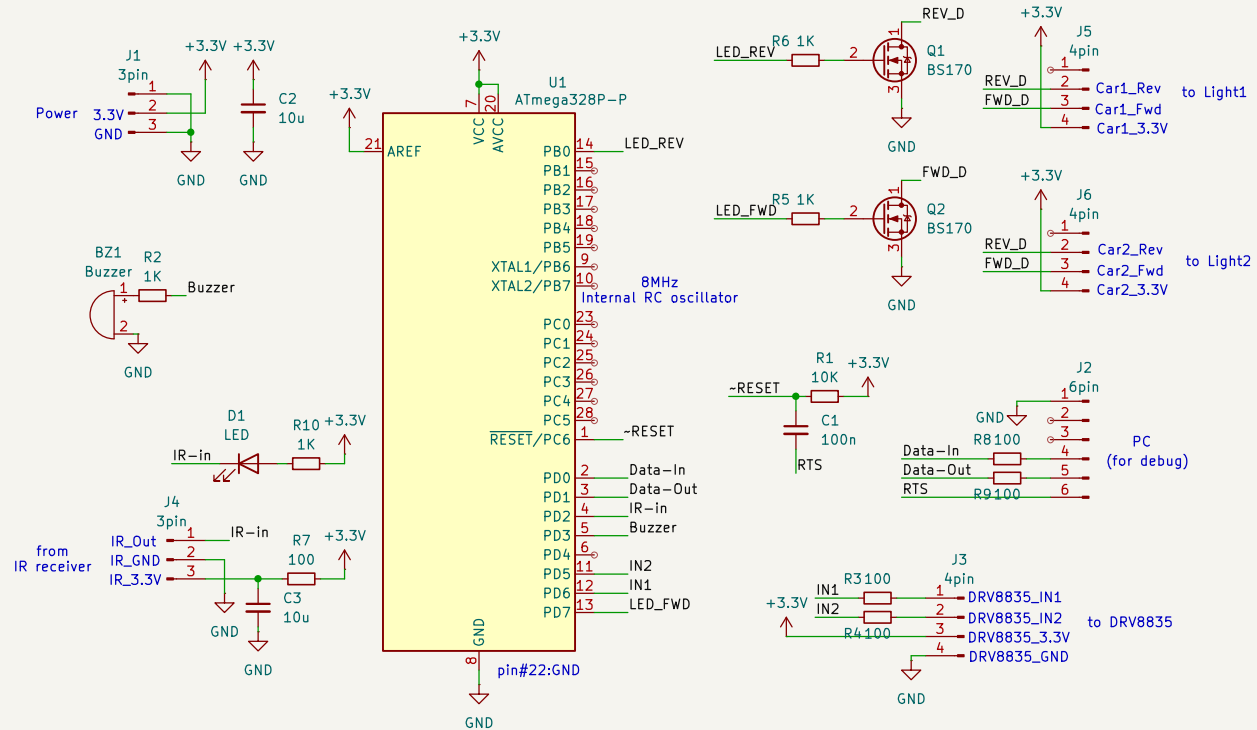
Title: てんしゃのおもちゃ2025/7 構成図

Size: A4 Date: 2025-07-24

KiCad E.D.A. 8.0.3

Rev: 1.0

Id: 1/1



なかの電気クラブ SAWA

Sheet: /

File: Atmega328P_on_ToyTrain.kicad_sch

Title: Main Circuit for ToyTrain (Atmega328P/Arduino)

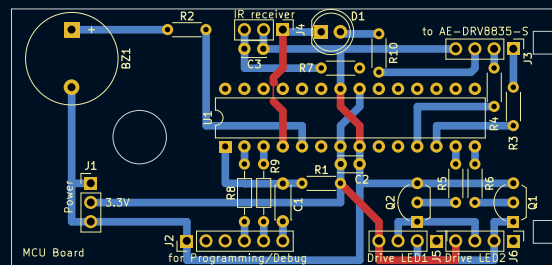
Size: A4

Date: 2025-07-24

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



なかの電気クラブ SAWA

Sheet:

File: Atmega328P_on_ToyTrain.kicad_pcb

Title: Main Circuit for ToyTrain (Atmega328P/Arduino)

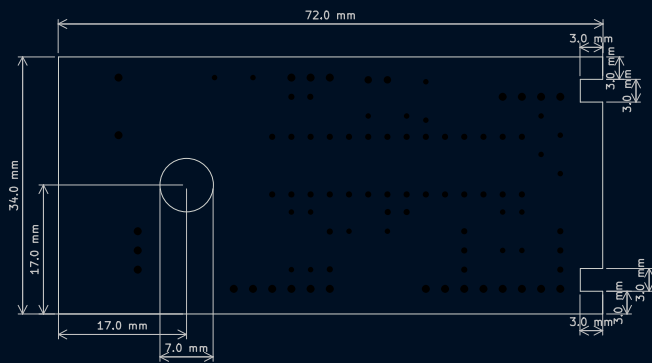
Size: A4

Date:

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



なかの電気クラブ SAWA

Sheet:

File: Atmega328P_on_ToyTrain.kicad_pcb

Title: Main Circuit for ToyTrain (Atmega328P/Arduino)

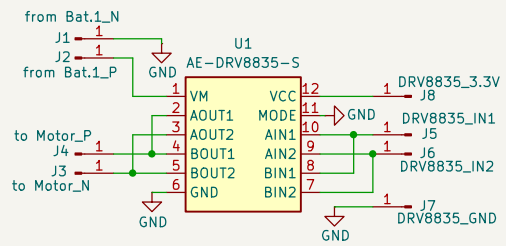
Size: A4

Date:

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



なかの電気クラブ SAWA

Sheet: /

File: DRV8835_on_ToyTrain.kicad_sch

Title: Motor Drive Circuit (AE-DRV8835-S)

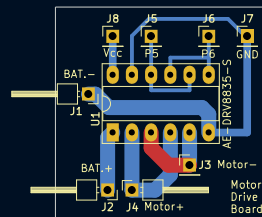
Size: A4

Date: 2025-07-24

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



なかの電気クラブ SAWA

Sheet:

File: DRV8835_on_ToyTrain.kicad_pcb

Title: Motor Driver Circuit (AE-DRV8835-S)

Size: A4

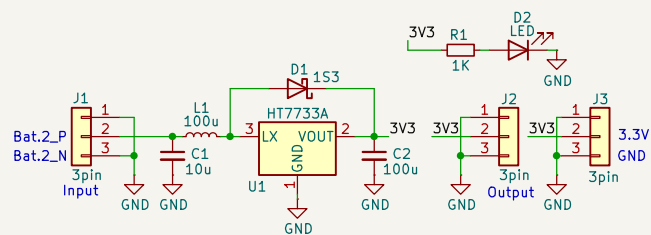
Date: 2025-07-21

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1





なかの電気クラブ SAWA

Sheet: /

File: DCDC_Converter_HT7733A.kicad_sch

Title: Step-Up DC/D Converter Circuit

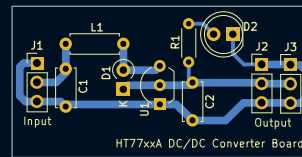
Size: A4

Date: 2025-07-24

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



なかの電気クラブ SAWA

Sheet:

File: DCDC_Converter_HT7733A.kicad_pcb

Title: Step-Up DC/DC Converter Board

Size: A4

Date: 2025-07-21

Rev: 1.0

KiCad E.D.A. 8.0.3

Id: 1/1



//リモコン受信プログラム：信号はNEC format、IRremoteライブラリのSimpleReceiverを使用
//でんしゃのおもちゃ内蔵基板用 元ファイルSimpleReceiver_with_control_train.inoとはI/Oportが異なる

```
#include <Arduino.h>  
#include "TinyIRReceiver.hpp"
```

```
const byte Pin_DRV8835_IN1 = 5; //DRV8835 IN/IN mode  
const byte Pin_DRV8835_IN2 = 6; //DRV8835 IN/IN mode  
const byte Pin_buzzer = 3; //圧電素子  
const byte Pin_Led_Fwd = 7; //LED forward  
const byte Pin_Led_Rev = 8; //LED reverse  
bool stat_forward = true; //true:前進 false:後退  
int stat_speed = 0; //0:停止 1,2:速度アップ  
bool stat_change_speed = false; //速度変更の有無 true:有 false:無  
bool stat_LED_On = true; //true LED on false:LED off
```

```
void setup() {  
  initPCIInterruptForTinyReceiver(); // Enables the interrupt generation on change  
  of IR input signal  
  Serial.begin(9600);  
  Serial.println("Cont_Toy-Train_Motor_LED.ino");  
  Serial.println("receive address 0x50");  
  Serial.println("Adrs Button : Function");  
  Serial.println("0x17 Power : Stop motor");  
  Serial.println("0x09 Change Input : LED on/off");  
  Serial.println("0x19 CH up : Speed up (FWD)");  
  Serial.println("0x18 CH down : Speed down (FWD)");  
  Serial.println("0x12 CH up : Speed up (REV)");  
  Serial.println("0x15 CH down : Speed down (REV)");  
  // Serial.println("in setup");  
  pinMode(Pin_buzzer, OUTPUT);  
  digitalWrite(Pin_buzzer, 0);  
  pinMode(Pin_Led_Fwd, OUTPUT);  
  pinMode(Pin_Led_Rev, OUTPUT);  
  digitalWrite(Pin_Led_Fwd, 0);  
  digitalWrite(Pin_Led_Rev, 0);  
  pinMode(Pin_DRV8835_IN1, OUTPUT);  
  pinMode(Pin_DRV8835_IN2, OUTPUT);  
  Cont_Motor_Break();  
  delay(100);  
  Cont_Motor_FreeRun();  
}
```

```
void loop() {  
  Check_IR_signal();  
  Led_OnOff();  
  Control_motor_speed();  
  delay(250);  
}
```

```
void Control_motor_speed() {  
  int data_pwm = 0;  
  if (stat_speed == 0) {  
    data_pwm = 0;  
  }  
  if (stat_speed == 1) {  
    data_pwm = 75;  
  }  
  if (stat_speed == 2) {  
    data_pwm = 100;  
  }  
  if (stat_speed == 3) {  
    data_pwm = 140;  
  }  
}
```

```
if (stat_change_speed == true) { //モーター制御（速度・回転方向）で変更発生時の処理  
  if (stat_forward == true) {  
    Cont_Motor_forward(data_pwm);  
    Serial.print(" dir=forward speed=");  
    Serial.println(stat_speed);  
    digitalWrite(Pin_Led_Fwd, 1);  
    digitalWrite(Pin_Led_Rev, 0);  
  }  
  if (stat_forward == false) {  
    Cont_Motor_backward(data_pwm);  
    Serial.print(" dir=backward speed=");  
    Serial.println(stat_speed);  
    digitalWrite(Pin_Led_Fwd, 0);  
    digitalWrite(Pin_Led_Rev, 1);  
  }  
  stat_change_speed = false;  
  delay(100);  
}
```

```
void Check_IR_signal() {  
  if (TinyReceiverDecode()) { //受信信号がNECフォーマットの場合  
    printTinyReceiverResultMinimal(&Serial);  
    buzzer_Xms(1000, 10);  
    byte data_adrs = TinyIRReceiverData.Address;  
    byte data_cmd = TinyIRReceiverData.Command;  
    Serial.print(F("Adrs=0x"));  
    // Serial.print(TinyIRReceiverData.Address, HEX);  
    Serial.print(data_adrs, HEX);  
    Serial.print(F(" Cmd=0x"));  
    // Serial.print(TinyIRReceiverData.Command, HEX);  
    Serial.print(data_cmd, HEX);  
    Serial.println(' ');  
    if (data_adrs == 0x50) { //アドレスが合っていたら処理する  
      if (data_cmd == 0x09) { //テスト用：LED on/offをトグル動作  
        stat_LED_On = !stat_LED_On;  
        buzzer_Xms(5500, 10);  
      }  
      if (data_cmd == 0x17) { //停止  
        stat_speed = 0;  
        Cont_Motor_Break();  
        delay(10);  
        Cont_Motor_FreeRun();  
        buzzer_Xms(7000, 10);  
        buzzer_Xms(6000, 10);  
        buzzer_Xms(5000, 10);  
      }  
      if (data_cmd == 0x19) { //前進：速度アップ  
        buzzer_Xms(4500, 10);  
        if (stat_speed == 0 || stat_forward == true) {  
          if (stat_speed < 3) {  
            stat_speed++;  
            stat_change_speed = true;  
            stat_forward = true;  
            buzzer_Xms(7000, 10);  
          }  
        }  
      }  
      if (data_cmd == 0x18) { //前進：速度ダウン  
        buzzer_Xms(4000, 10);  
        if (stat_speed >= 1 || stat_forward == true) {  
          if (stat_speed > 0) {  
            stat_speed--;  
            stat_change_speed = true;  
            buzzer_Xms(7000, 10);  
          }  
        }  
      }  
    }  
  }  
}
```

```

    }
  }
}
if (data_cmd == 0x12) { //後退:速度アップ
  buzzer_Xms(3500, 10);
  if (stat_speed == 0 || stat_forward == false) {
    if (stat_speed < 3) {
      stat_speed++;
      stat_change_speed = true;
      stat_forward = false;
      buzzer_Xms(7000, 10);
    }
  }
}
if (data_cmd == 0x15) { //後退:速度ダウン
  buzzer_Xms(3000, 10);
  if (stat_speed >= 0 || stat_forward == false) {
    if (stat_speed > 0) {
      stat_speed--;
      stat_change_speed = true;
      buzzer_Xms(7000, 10);
    }
  }
}
}
}
}
}
}

```

```

void buzzer_Xms(int freq, int time) {
  tone(Pin_buzzer, freq);
  delay(time);
  noTone(Pin_buzzer);
}

```

```

void Led_OnOff() {
  if (stat_LED_On == false) {
    digitalWrite(Pin_Led_Fwd, 0);
    digitalWrite(Pin_Led_Rev, 0);
    Serial.print("      Both LED off : ");
  }
  if (stat_LED_On == true && stat_forward == true) {
    digitalWrite(Pin_Led_Fwd, 1);
    digitalWrite(Pin_Led_Rev, 0);
    Serial.print("      Reverse LED on : ");
  }
  if (stat_LED_On == true && stat_forward == false) {
    digitalWrite(Pin_Led_Fwd, 0);
    digitalWrite(Pin_Led_Rev, 1);
    Serial.print("      Forward LED on : ");
  }
  Serial.println(stat_speed);
}

```

```

void Cont_Motor_Break() { //ブレーキ
  analogWrite(Pin_DRV8835_IN1, 255);
  analogWrite(Pin_DRV8835_IN2, 255);
}

```

```

void Cont_Motor_FreeRun() { //空転
  analogWrite(Pin_DRV8835_IN1, 0);
  analogWrite(Pin_DRV8835_IN2, 0);
}

```

```

void Cont_Motor_forward(int value) { //正転
  analogWrite(Pin_DRV8835_IN1, value);
  analogWrite(Pin_DRV8835_IN2, 0);
}

```

```

void Cont_Motor_backward(int value) { //逆転
  analogWrite(Pin_DRV8835_IN1, 0);
  analogWrite(Pin_DRV8835_IN2, value);
}

```