

FEETECH 社デジタルサーボの使用方法

■最初に■

この説明書では FEETECH 社提供のデバッグソフトと Arduino (Yun、Mega2560) を使用した動作確認、及び ArduinoUNO を使用した動作確認の為に簡易なプログラムを説明します。

■デバッグソフトの用意■

◎用意するもの

- お使いの Windows P C Windows 1 0 (64bit 版) 以外の OS 上での動作は未確認です。
- ArduinoYun または ArduinoMega2560 または ArduinoUNO (それぞれ互換品も可)
- FE-URT-1 (USB ケーブル同梱)
- FEETECH 社デジタルサーボ (この説明書は弊社販売 STS3032、STS3215 での確認です)
- サーボ用電源: 6 V 2 A 以上のもの
(サーボを複数台接続する場合は、サーボの必要電源容量に応じて適宜ご用意ください)
- 接続用ケーブル: 4 本 プレッドボード・ジャンパーワイヤ (オスメス) 等適宜

◎弊社ホームページの該当商品ページより、下記をダウンロード、解凍してください。

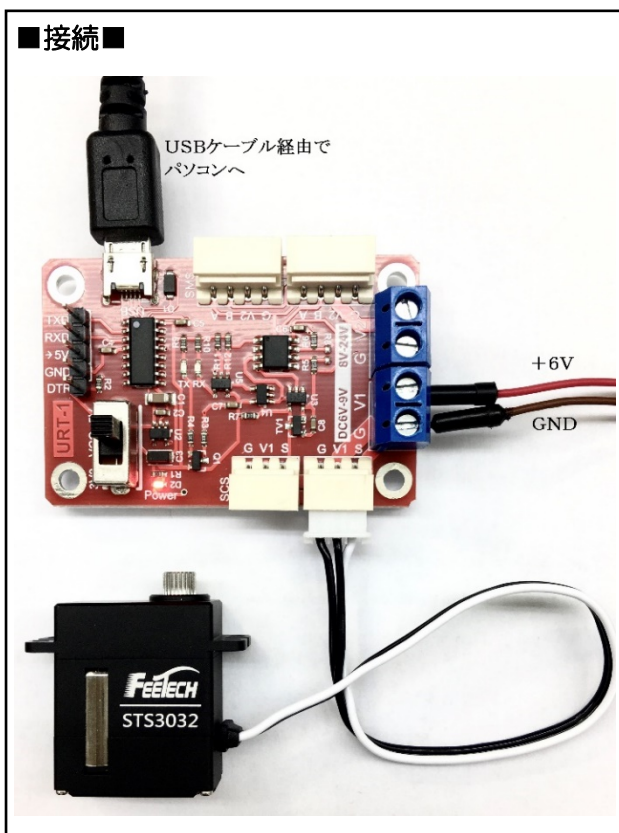
- FD1.9.8-EN_200923.zip

下記 zip ファイルはダウンロードし、解凍せずに ArduinoIDE のライブラリにインクルードしてください。

- SCServo.zip

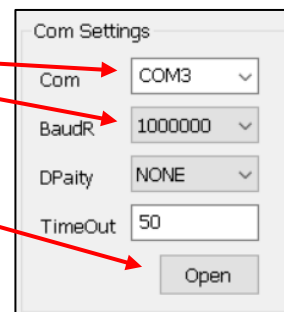
●最初に動作確認と、サーボ個別の ID の設定方法を説明します。

■接続■

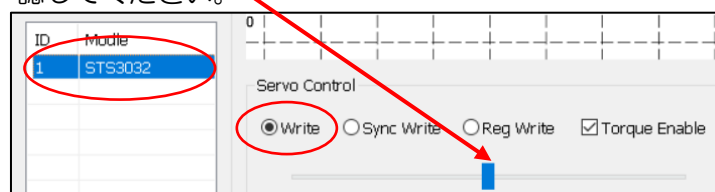
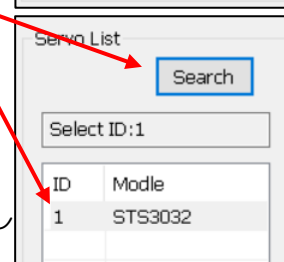


■動作確認■

- FD1.9.8-EN_200923.zip を解凍し、FD.exe を立ち上げます。
- Com ポートを確認して BaudR を 1000000 に設定し、「Open」をクリックし通信を確立します。

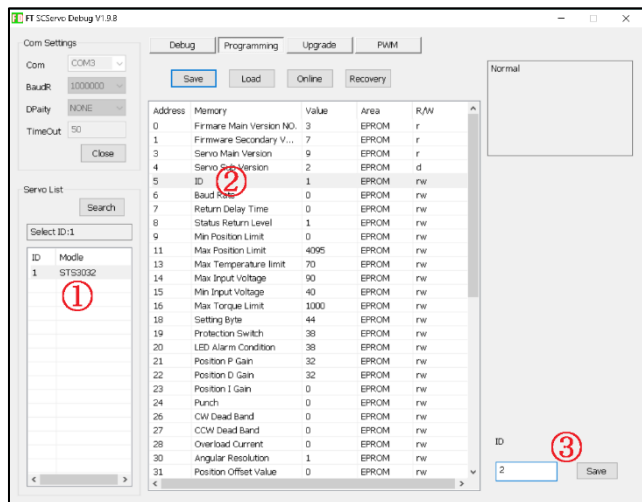


- Servo List の「Search」をクリックすると、接続されたサーボの ID が表示されます。
工場出荷時の初期値は ID=1 です。
この作業は必ず 1 台のみ接続しておこなってください。
次ページで ID の変更手順を説明します。
- 接続されたサーボ名をクリックし write にチェックを入れ、スライダーを左右に動かしサーボが動作するのを確認してください。

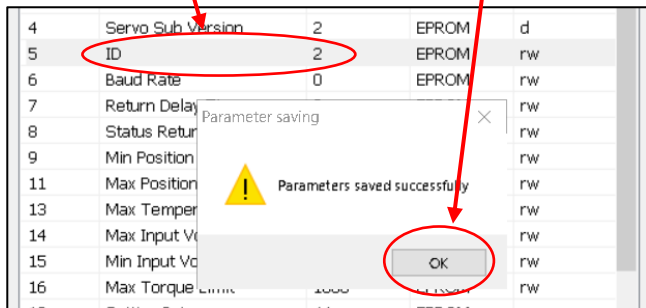


■IDの変更■

- ①上部「Programming」をクリックすると下記画面が出ますので、Servo List に表示されている接続されたサーボをクリックしてください。
- ②サーボのレジスタ内容が反映されますので「5 ID」をクリックします。
- ③右下に「ID」の文字と入力窓に「1」が表示されますので、1 を2に変更して窓の右の「Save」をクリックします。



・ID が2に変更された事を確認して「OK」をクリックします。



複数台のサーボを使用する場合は上記手順で ID を設定してください。

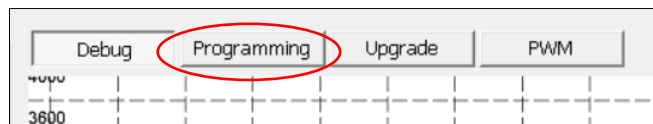
また、設定後は上部 Debug ボタンをクリックして各個体の動作確認をおこなってください。

■サーボの初期化（リカバリ）■

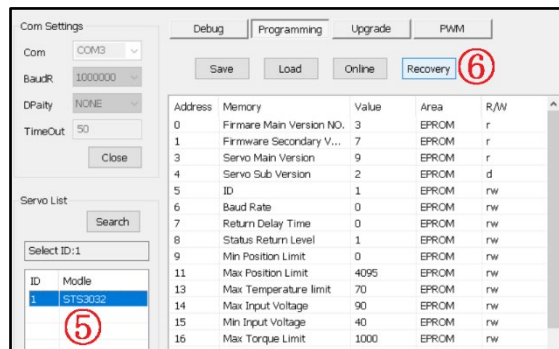
サーボ内のレジスタが自作ソフトウェアのバグなどで意図せず書き換えられてしまった場合の初期化（リカバリ）の方法を説明します。

作成者の意図する本来のコマンドを受付けなかったり、回転が止まらなかったりする様な場合にお試しください。

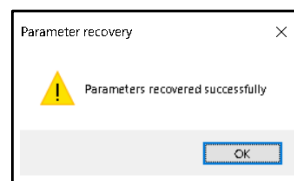
- ①サーボ、FE-URT-1、サーボ用電源、PCを接続。
- ②FD.exe を立ち上げる。
- ③前ページ「■動作確認■」の手順をおこなう。
（この時点ではサーボはコマンドを受け付けないのでスライダーは操作できません）
- ④上部「Programming」をクリック。



- ⑤該当するサーボ ID を選択し、
- ⑥「Recovery」をクリックします。



- ⑦下記画面が表示されればリカバリ成功です。



- ⑧上部「Debug」をクリックし、前画面に戻りスライダーを動かして動作を確認してください。

■ArduinoYun、 Arduino Mega2560 での動作確認■

ライブラリを使用しての動作確認を説明します。

ライブラリは全て Serial1 （２番目のシリアルポート）が使用できる事が前提として書かれています。

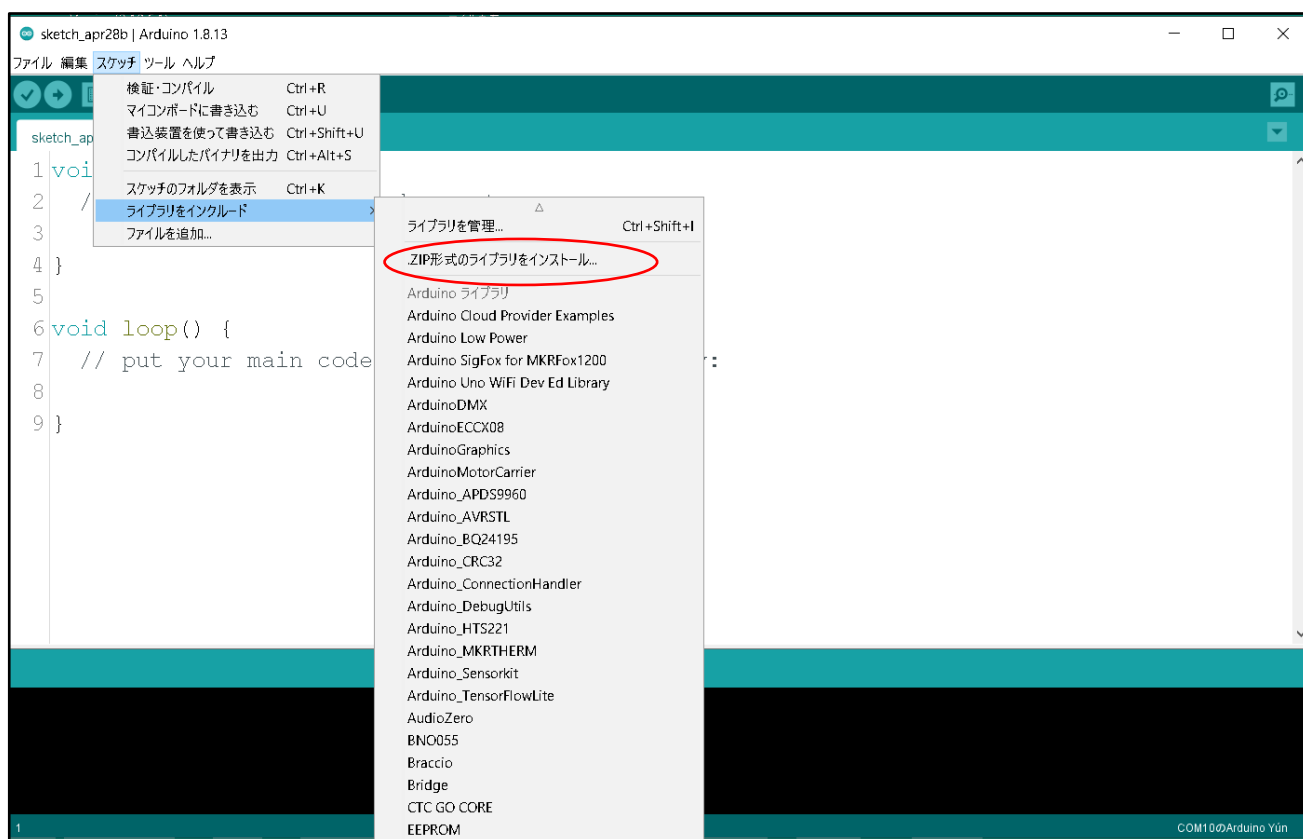
■接続■（ブレッドボード・ジャンパーワイヤ（オスメス）等を使用すると便利です）

FE-URT-1	DTR	GND	+5V	RXD	TXD
	↓	↓	↓	↓	↓
ArduinoYun	無接続	GND	5V	RX	TX
ArduinoMega2560	無接続	GND	5V	RX1	TX1

- ・ F E－U R T－1 基板上のスイッチは「5V」側にセットします。
- ・ サーボ、電源の接続は１ページ目をご参照ください。

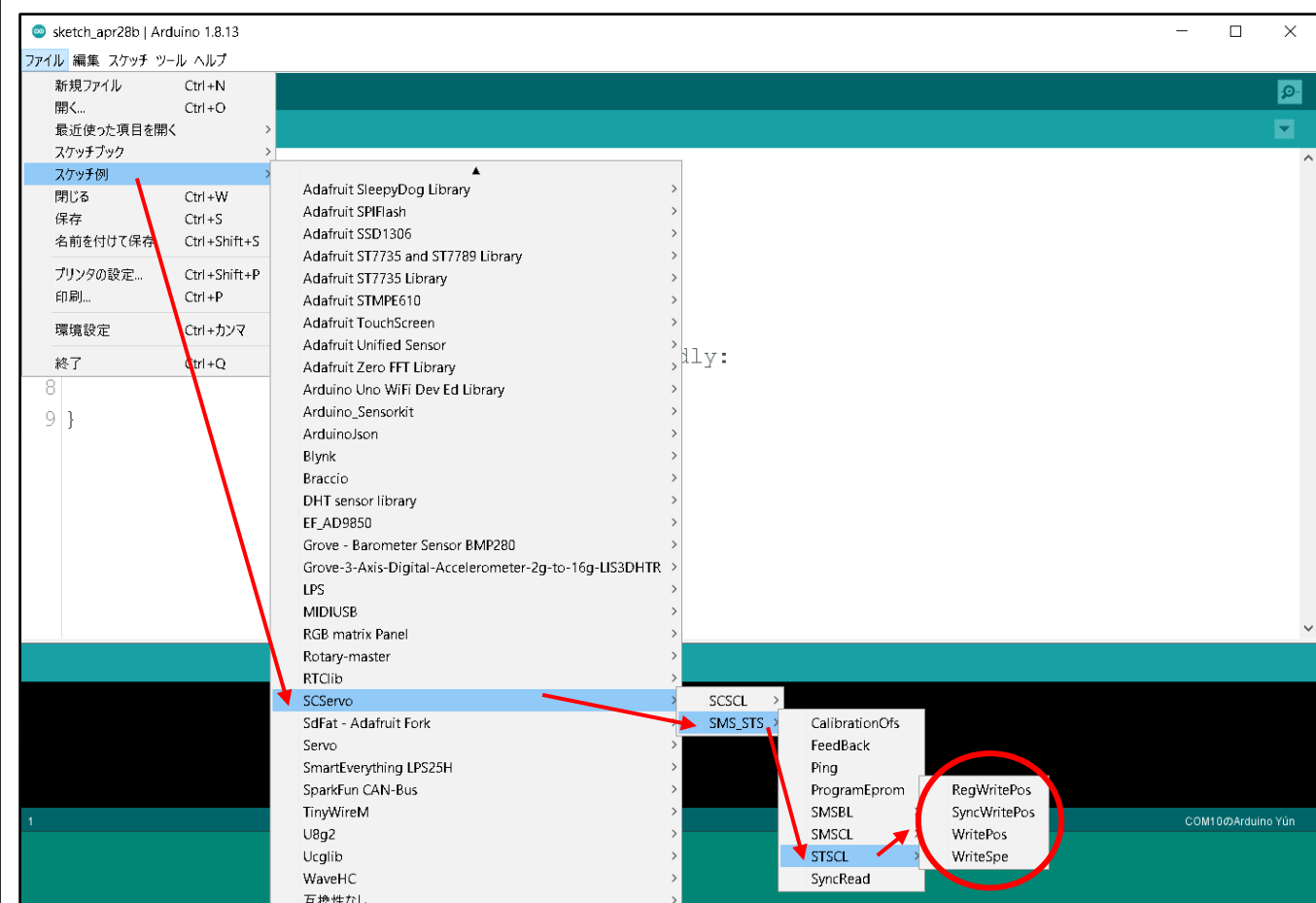
■ライブラリのインクルード■

Arduino IDE を立ち上げ、「スケッチ」⇒「ライブラリをインクルード」⇒「ZIP 形式のライブラリをインストール」の順に進み、ダウンロード済みの SCSServo.ZIP を指定します。



■スケッチ例の選択■

「ファイル」⇒「スケッチ例」⇒「SCServo」⇒「SMS_STS」⇒「STSCCL」と進むと、下図のように4例が表示されます。



- ・ファイル⇒スケッチ例⇒SCServo⇒SMS_STS⇒STSCCL
- ⇒RegWritePos : ID1,ID2 を個別書き込みで動作（個別パラメータ設定）
- ⇒SyncWritePos : ID1,ID2 を同期書き込みで動作（同一パラメータ設定）
- ⇒WritePos : ID1,通常書き込み
- ⇒WriteSpe : ID1,通常書き込み 多回転定速モード

■ArduinoUNO の使用■

今回の例ではサーボをハードウェアシリアル、表示やキーボード入力をソフトウェアシリアルでおこないます。ライブラリを使用したスケッチ例は「Serial1」を全て「Serial」に書き換える必要があります。スケッチ内で「Serial」を使用してデータを表示したい場合は「SoftwareSerial」を使用します。これは、SoftwareSerial がサーボの必要とする 1000000bps をサポートしていない為です。また、サーボ内のレジスタデータを参照するなどの為にソフトウェアシリアル機能を使用する場合は USB-シリアル変換モジュール（FT232RL 超小型USBシリアル変換モジュールなど）が別途必要になります。

■接続■

FE-URT-1	DTR	GND	+5V	RXD	TXD
	↓	↓	↓	↓	↓
ArduinoUNO	無接続	GND	5V	ORX	1TX

ArduinoUNO	GND	+5v	D2	D3
	↓	↓	↓	↓
USB-シリアル変換	GND	5V	TXD	RXD

- ・FE-URT-1 基板上のスイッチは「5V」側にセットします。
- ・サーボ、電源の接続は1 ページ目をご参照ください。

下記の例では、ボリュームを回すとサーボ（ID=1）が回転し、ボリュームで設定した位置で止まります。また、PCのキーボードから0～9の数字キーを押し続ける事で「数字×455」の位置に移動します。数字キーを離すとボリュームで設定した位置に戻ります。

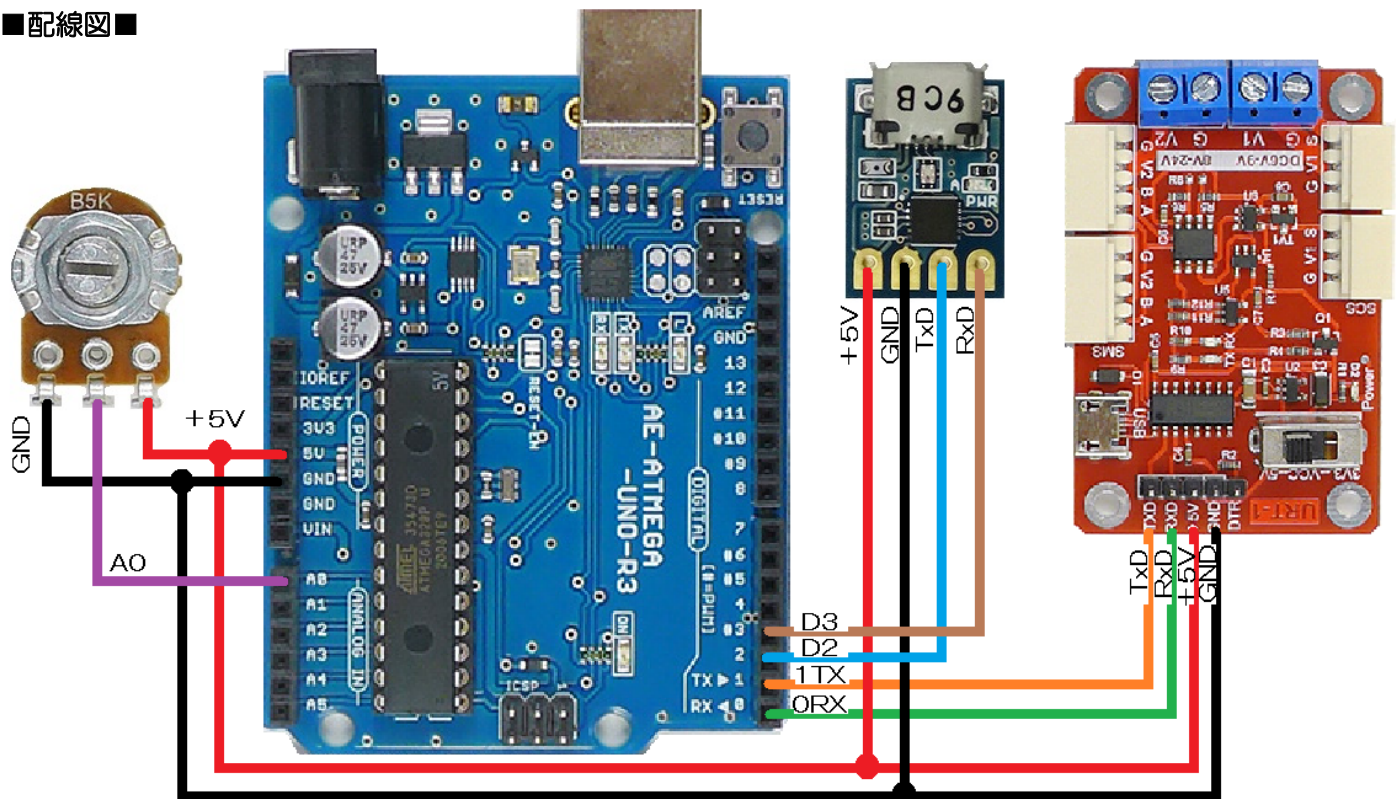
（戻る時は受信バッファをクリアする為に少しタイムラグがあります）

ArduinoUNO のシリアルモニタはサーボの入出力に使用していますので、ソフトウェアシリアルの入出力にはターミナルソフト（Tera Term など）を適宜ご用意ください。

ArduinoIDE（開発ソフト）のシリアルポート番号は ArduinoUNO のシリアルポート番号に合致させます。

また、ターミナルソフトのシリアルポート番号はソフトウェアシリアルが接続されているUSB-シリアル変換モジュールのシリアルポート番号に合致させます。

■配線図■



■サンプルスケッチ■

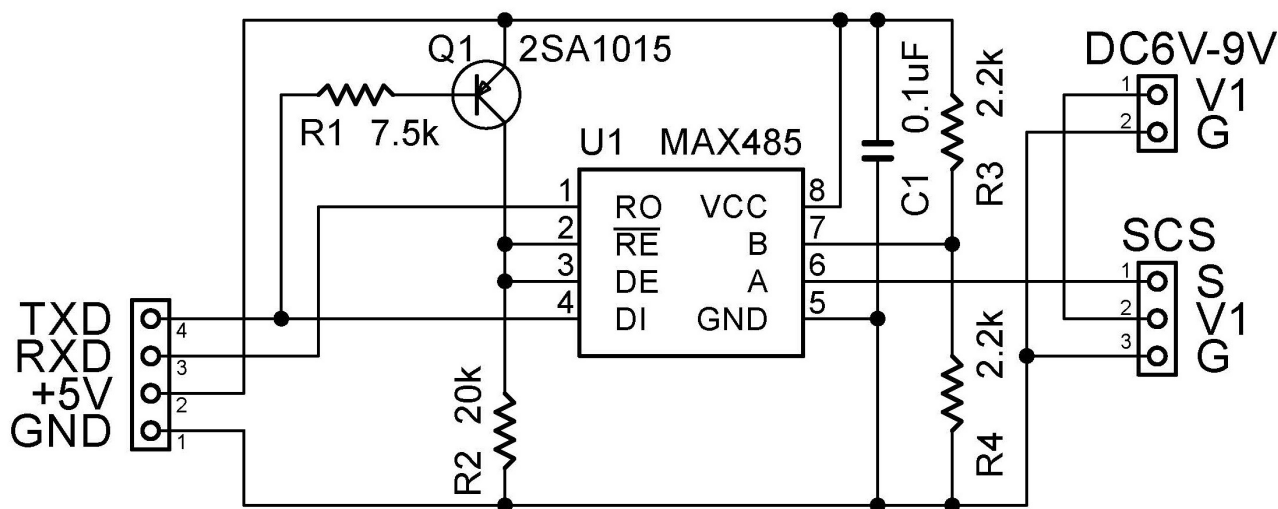
スケッチを書き込む時は ArduinoUNO の「ORX」は切り離しておいてください。動作させる時に接続します。

```
//CHECK_ALL.ino
#include <SCServo.h>
#include <SoftwareSerial.h>
SoftwareSerial softSerial(2, 3); // RX, TX
SMS_STS sms_sts;
unsigned int SERVO_POS = 0;
void setup() {
    Serial.begin(1000000);
    softSerial.begin(9600);
    sms_sts.pSerial = &Serial;
    delay(1000);
}
void loop() {
    sms_sts.WritePosEx(1, SERVO_POS, 0, 254);
    RCV_DATA();
    INPUT_POS();
}
void INPUT_POS(void)
{
    if(softSerial.available()) {
        char data = softSerial.read();
        if((data >= 0x30)&&(data <= 0x39)) {
            SERVO_POS = (data - 0x30) * 455;
        }
    }
    else{
        SERVO_POS = analogRead(0) * 4;
    }
}
```

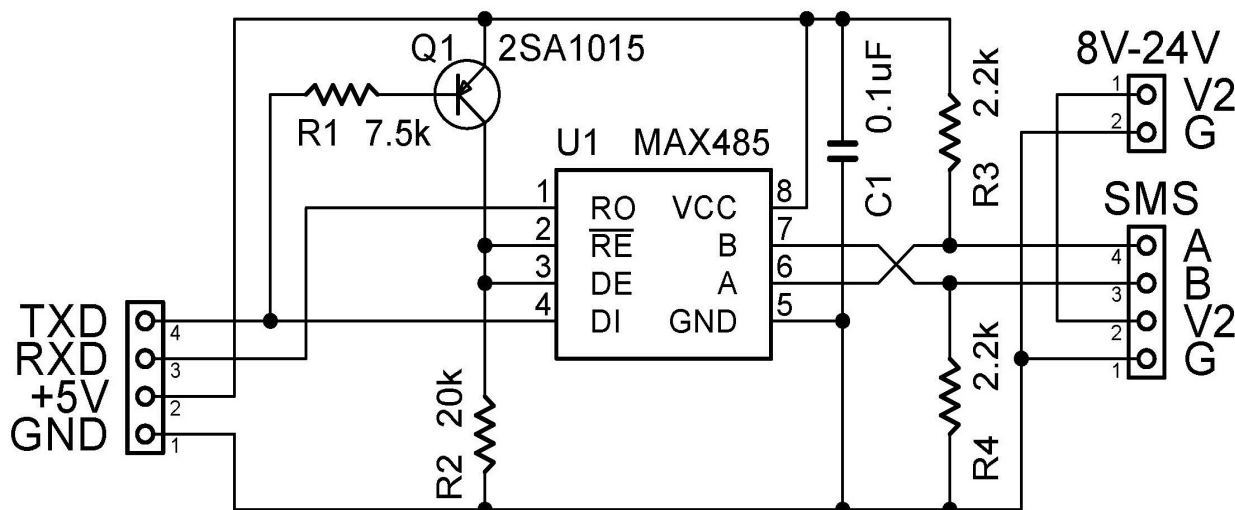
```
//続き
void RCV_DATA(void) {
    int Pos;
    if(sms_sts.FeedBack(1) != -1) {
        Pos = sms_sts.ReadPos(-1);
        softSerial.print("Position:");
        softSerial.print(Pos);
        delay(10);
    }else{
        softSerial.print("FeedBack err");
        delay(10);
    }
    Pos = sms_sts.ReadPos(1);
    if(Pos != -1) {
        softSerial.print("    Servo position:");
        softSerial.print(Pos);
        delay(10);
    }else{
        softSerial.print("    read position err");
        delay(10);
    }
    softSerial.println();
}
```


■実験用簡易インターフェイスの自作■（FE-URT-1 を使用しない方法）
信号名はFE-URT-1 基板上の信号に相對します。

●SCSシリーズ、STSシリーズデジタルサーボ用インターフェイスです。



●SMSシリーズデジタルサーボ用インターフェイスです。



この回路の場合、TXD、RXDは5V TTLレベルです。（3.3V不可）

サーボへの配線ですが、メーカー仕様書にはコネクタの詳細は記載されておりません。

受け側（基板側コネクタ）のピン径は0.7mmΦ（実測値）ですので、お客様ご自身で工夫して接続してください。

簡易的にブレッドボードジャンパーピンなどをお使いいただく事は可能かと存じますが、接触の確実性は保証いたしかねます事をご了承くださいませ。

注）上記回路や接続方法は簡易的なものです。実験用途での使用確認はいたしておりますが、機器への組み込み用途などには冒頭で紹介いたしましたFE-URT-1のご使用をお勧めいたします。

ご使用はご自身の責任（オウンリスク）にてお願いいたします。

この回路を用いた場合のいかなる不具合、不利益も弊社は責任を負わないものとします。

■補足：コマンドの送出■

直接マイコンなどで制御する場合に参考にしてください。

- ・パリティチェックなし、データ長8bit、1ストップビット、フロー制御なし、ボーレート 1000000bps
- ・表中のパケットデータ長は「ID 番号位置を0」とした場合の「チェックサムより前のデータ」の長さです。
（データが10個あればパケットデータ長は9になります）
- また、チェックサムは、ヘッダーとチェックサムを含まないデータを全て加算して下位8bitのみを残し、ビット反転したものです。
 $CHECKSUM = \sim((byte)((ID + \text{データ長} + \text{コマンド} + \dots) \& 0xFF))$
- F E E T E C H社のホームページには簡単に16進コマンドを生成できるエクセルファイルが用意されています。詳細はそちらをご覧ください。

・ステップコントロール

S C S シリーズ	ヘッダー		I D 番号	パケット データ長	コマンド	レジスタ 先頭番地	位置情報バイト		時間情報バイト		速度情報バイト		チェックサム
							上位	下位	上位	下位	上位	下位	
16進	FF	FF	01	09	03	2A	00	00	03	E8	00	00	DD
10進	255	255	1	9	3	42	0	0	3	232	0	0	221
送出 16 進コマンド			FF FF 01 09 03 2A 00 00 03 E8 00 00 DD										

SMS, STS シリーズ	ヘッダー		I D 番号	パケット データ長	コマンド	レジスタ 先頭番地	位置情報バイト		時間情報バイト		速度情報バイト		チェックサム
							下位	上位	下位	上位	下位	上位	
16進	FF	FF	01	09	03	2A	00	00	E8	03	00	00	DD
10進	255	255	1	9	3	42	0	0	232	3	0	0	221
送出 16 進コマンド			FF FF 01 09 03 2A 00 00 E8 03 00 00 DD										

・マルチターンコントロール（時間情報バイトは動作に影響を与えません：非機能）

SMS STS シリーズ	ヘッダー		I D 番号	パケット データ長	コマンド	レジスタ 先頭番地	加速度 情報 バイト	位置情報 バイト		時間情報 バイト		速度情報 バイト		チェックサム
								下位	上位	下位	上位	下位	上位	
16進	FF	FF	01	0A	03	29	32	E0	2E	00	00	5F	00	29
10進	255	255	1	10	3	42	50	224	46	0	0	95	0	41
送出 16 進コマンド			FF FF 01 09 03 29 32 E0 2E 00 00 5F 00 29											

・読み込みコマンド

レジスタ先頭番地から読込バイト数だけのデータを読み込みます。
下の例では0x38 番地からの位置情報（2バイト）を取得します。

表記	ヘッダー		I D 番号	パケットデータ長	コマンド	レジスタ先頭番地	読込バイト数	チェックサム
16進	FF	FF	01	04	02	38	02	BE
10進	255	255	1	4	2	56	2	190
送出 16 進コマンド			FF FF 01 04 02 38 02 BE					

受信データの例：
FF FF 01 04 00 07 00 E3 (07, 00 = 0x0007 = 位置7)
FF FF 01 04 00 F2 0F F9 (F2, 0F = 0x0FFF2 = 位置4082)
(受信データの詳細についてはメーカーページ、メーカーサンプルソフトをご参照ください)

本説明書の詳細はメーカーページでご確認ください。

(株)秋月電子通商