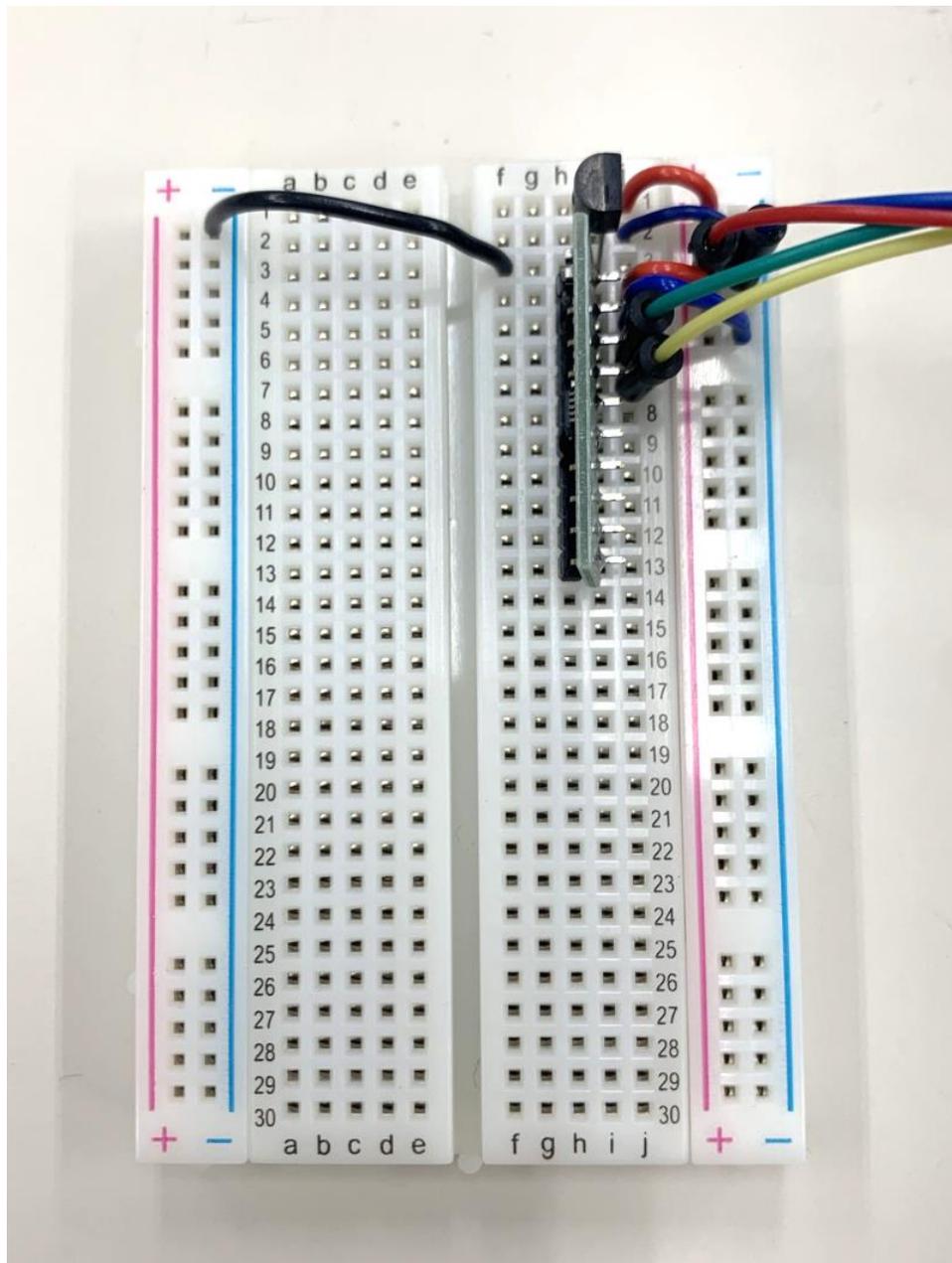


Arduinoで生理指標を測る

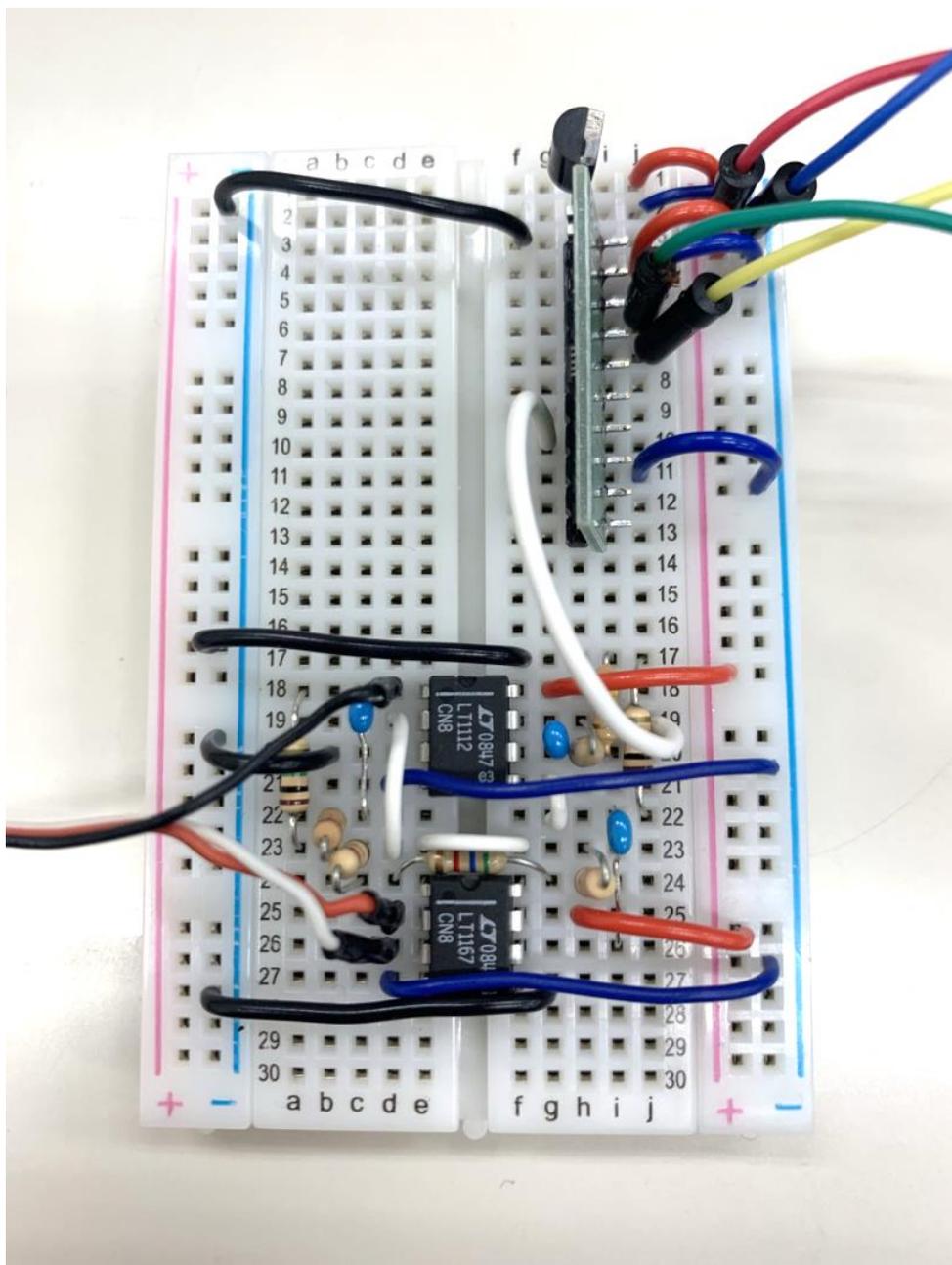
資料



資料1. ブレッドボード上の電源回路とAD変換モジュール

種別	部品名	本体	ブレッドボード	備考	
正負 電源回路	レールスプリッタ (TLE2426CLP)	1:IN	h1		
		2:COMMO	h2		
		3:OUT	h3		
	ワイヤー(レッドS)		j1	R+	
	ワイヤー(ブルーS)		j2	R-	
	ワイヤー(ブラックL)		f3	L-	
AD変換 モジュー ル	AD変換モジュール (ADS1015)	VIN	h4		
		GND	h5		
		SCL	h6		
		SDA	h7		
		ADR	h8		
		ALT	h9		
		A0	h10		回路からの 電圧入力
		A1	h11		
		A2	h12		
	A3	h13			
ワイヤー(レッドS)		j4	R+		
ワイヤー(ブルーS)		j5	R-		
ワイヤー(グリーンL)		j6		Arduino SCL	
ワイヤー(イエローL)		j7		Arduino SDA	

資料2. 電源回路とおよびAD変換モジュールの接続方法



資料3. ブレッドボード上のECG測定回路

種別	部品名	本体	ブレッドボード	備考	
被験者 電極	みのむし(ブラック)		d18	Patient GND	
	みのむし(レッド)		c25	Electrode-	
	みのむし(ホワイト)		c26	Electrode+	
増幅回路 1	計装アンプ (LT1167)	1	e24		
		2	e25		
		3	e26		
		4	e27		
		5	f27		
		6	f26		
		7	f25		
	ジャンプワイヤー(レッドM)		R+	h25	
	ジャンプワイヤー(ブルーL)		R-	d27	
	ジャンプワイヤー(ブラックL)		L-	g27	
	抵抗5.6kΩ (緑青赤金)		d24	g24 倍率設定用	
増幅回路2	オペアンプ (LT1112)	1	e18		
		2	e19		
		3	e20		
		4	e21		
		5	f21		
		6	f20		
		7	f19		
		8	f18		
		ワイヤー(レッドM)		R+	g18
		ワイヤー(ブルーL)		R-	d21
		ワイヤー(ブラックL)		L-	f17
		ワイヤー(ブラックS)		L-	b20
		ワイヤー(ホワイトM)		d19	d23
		ワイヤー(ホワイトS)		e23	f23
		抵抗30kΩ (橙黒橙金)		c23	c24
		抵抗30kΩ (橙黒橙金)		h23	h24
		抵抗12kΩ (茶赤橙金)		b22	b23
	抵抗1MΩ (茶黒緑金)		a18	a23	
	コンデンサ0.01uF(103)		c18	c22	
	ワイヤー(ホワイトS)		g21	g22	
	抵抗1MΩ (茶黒緑金)		j17	j22	
	抵抗240Ω (赤黄茶金)		i17	i20	
	抵抗15kΩ (茶緑橙金)		h19	h20	
	コンデンサ0.01uF(103)		g19	g20	
	コンデンサ0.1uF(104)		i22	i26	
出力	ワイヤー(ホワイトL)		j19	ADC-A0	
	ワイヤー(ブルーS)		R-	ADC-A1	

資料4. ECG測定回路の接続方法

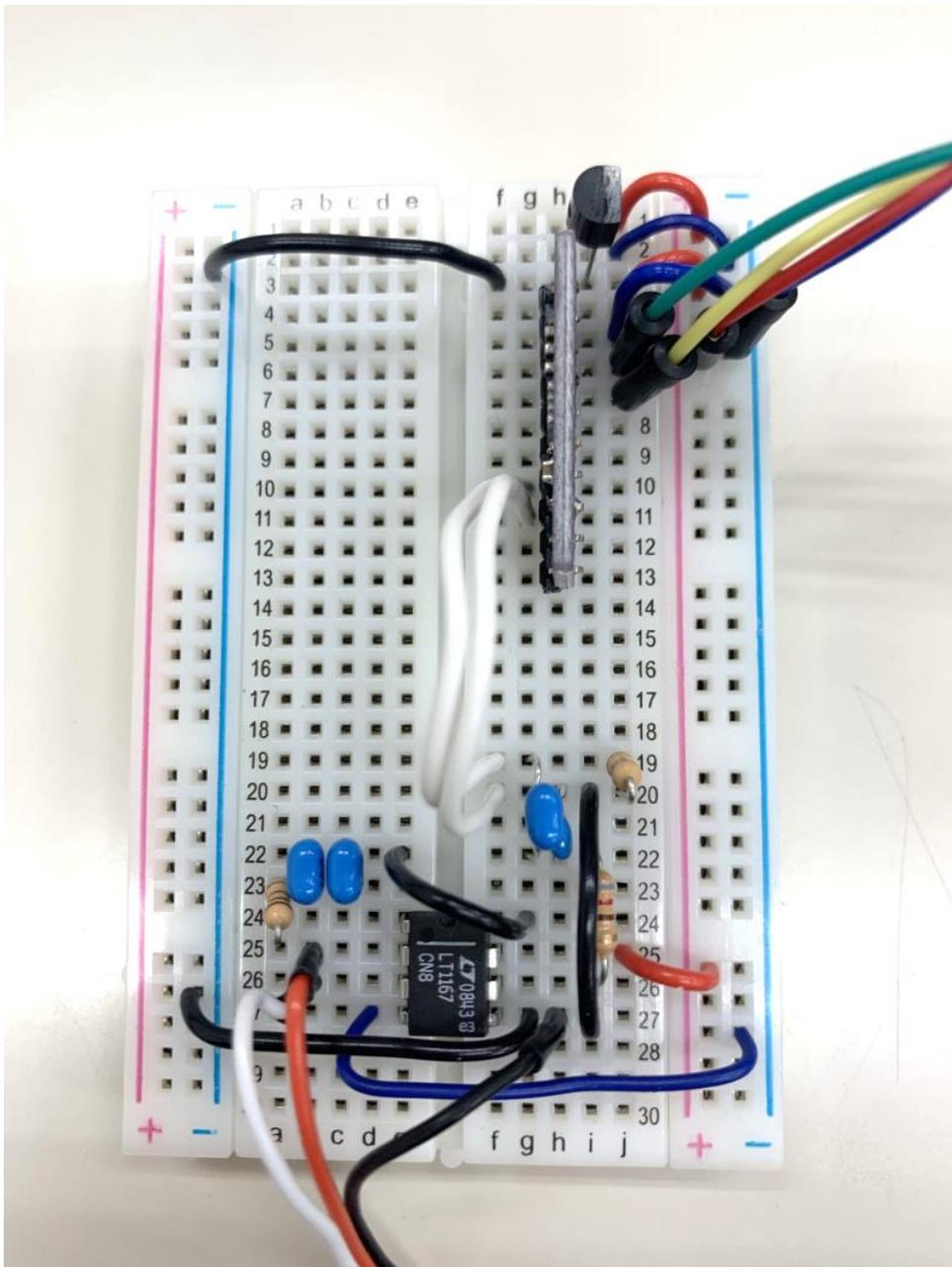
```

1 #include <Wire.h> //I2C通信ライブラリ
2 #include <Adafruit_ADS1X15.h> //ADS1015ライブラリ
3 Adafruit_ADS1015 ads; //ADS1015クラスのインスタンス化
4
5 void setup(void)
6 {
7   Serial.begin(115200); //シリアル通信開始
8   // ADS1015 gain 2/3 input range +/- 6.144V
9   ads.begin(); //ADS1015通信開始
10 }
11
12 void loop(void)
13 {
14   int16_t results; //AD変換結果
15   results = ads.readADC_Differential_0_1(); //差動入力
16
17   float multiplier = 3.0F; // デフォルトゲイン (2/3倍)における係数
18   //得られたデジタル値をmvに換算しシリアルモニタに表示
19   Serial.print(results * multiplier);
20   Serial.print(",4000,1000");
21   Serial.println(); //シリアルモニタ改行
22   delay(0);
23 }

```

資料5. ECG計測に用いたプログラム

AD変換モジュールのゲインは、デフォルトである2/3倍とし、入力レンジは±6.114V(プログラム9行目)、1bitあたり3.0mv(17行目)であった。



資料6. ブレッドボード上のEMG測定回路

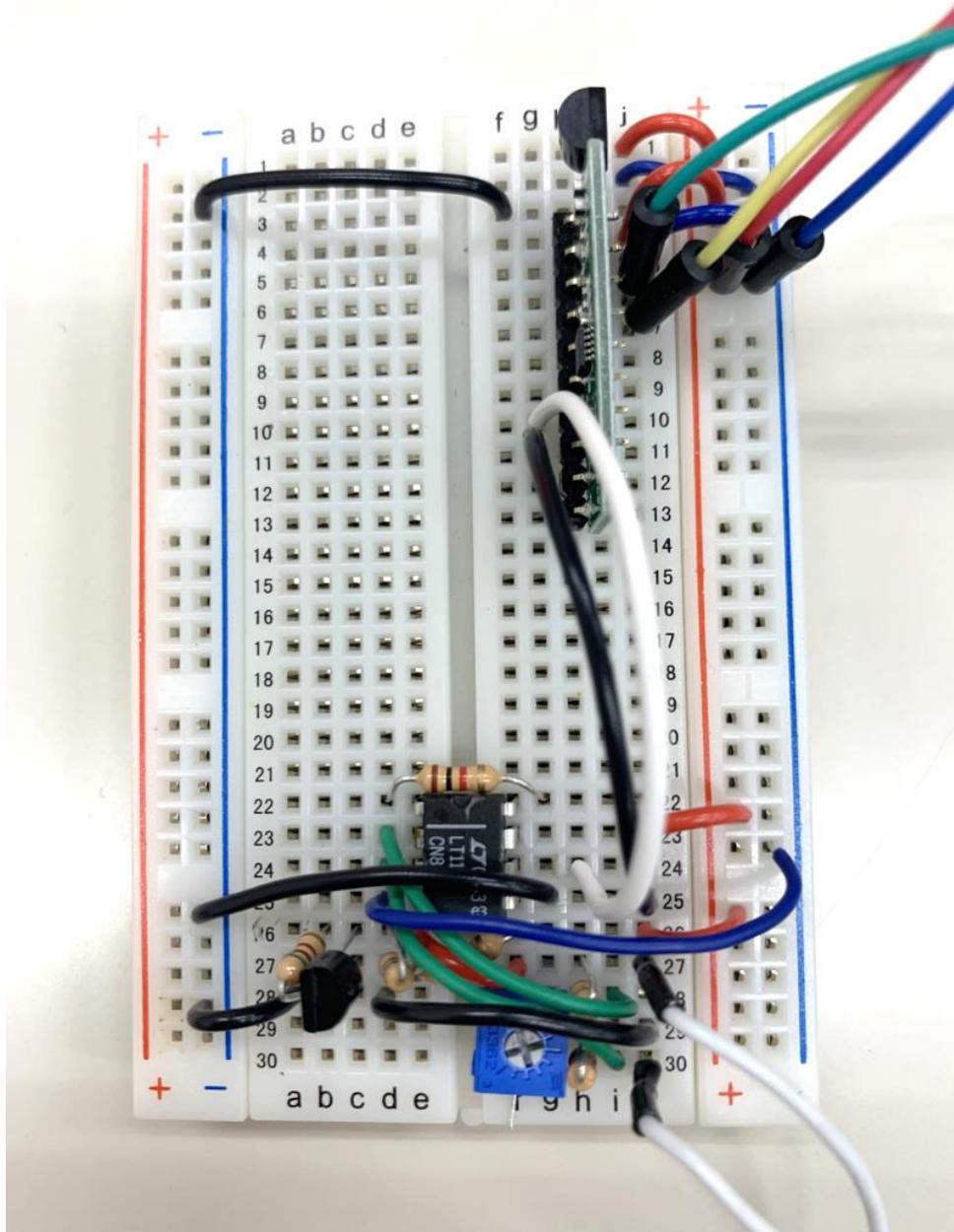
種別	部品名	本体	ブレッドボード	備考
被験者電極	みのむし(ブラック)		h27	
	みのむし(レッド)		b25	
	みのむし(ホワイト)		b26	
増幅回路	計装アンプ (LT1167)	1	e24	
		2	e25	
		3	e26	
		4	e27	
		5	f27	
		6	f26	
		7	f25	
		8	f24	
	ワイヤー(レッドS)		j25	R+
	ワイヤー(ブルーL)		d27	R-
	ワイヤー(ブラックL)		g27	L-
	ワイヤー(ブラックS)		e22	f24
	ワイヤー(ブラックL)		i20	i27
	コンデンサ10uF(106)		b22	b23
コンデンサ10uF(106)		c22	c23	
コンデンサ10uF(106)		g19	g22	
コンデンサ0.1uF(104)		h20	h22	
抵抗100Ω(茶黒茶金)		a23	a24	
抵抗820Ω(金)		i22	i26	
抵抗200Ω(赤黒茶金)		j19	j20	
出力	ワイヤー(ホワイトL)		f20	ADC-A1
	ワイヤー(ホワイトL)		f19	ADC-A0

資料7. EMG測定回路の接続方法

```
1 #include <Wire.h>
2 #include <Adafruit_ADS1X15.h>
3 Adafruit_ADS1015 ads;
4
5 void setup(void)
6 {
7   Serial.begin(115200);
8   // ADS1015 gain 16x  input +/- 0.256V
9   ads.setGain(GAIN_SIXTEEN); //ゲインを16倍に設定
10  ads.begin();
11 }
12
13 void loop(void)
14 {
15   int16_t results;
16   results = ads.readADC_Differential_0_1(); //差動入力
17
18   float multiplier = 0.125F; //ゲイン16倍時の係数
19   Serial.print(results * multiplier);
20   Serial.print(",100,-100");
21   Serial.println();
22   delay(0);
23 }
```

資料8. EMG計測に用いたプログラム

AD変換モジュールのゲインは、16倍とし、入力レンジは±0.256V(プログラム9行目)、1bitあたり0.125mv(18行目)であった。



資料9. ブレッドボード上のSCC測定回路

種別	部品名	本体	ブレッドボード	備考	
定電圧回路	電圧リファレンス (LM385Z-1.2)	1:Anode	c28		
		2:Cathode	c27		
		3:NC	c26		
	抵抗510Ω (緑茶茶金)		b28	b26	
	抵抗750Ω (紫緑茶金)		d27	d26	
	抵抗100Ω (茶黒茶金)		e27	f26	
	ワイヤー(レッドS)		e26	f27	この間に 0.5V出る
ワイヤー(ブラックL)		d28	j29		
ワイヤー(レッドS)		j26	R+		
ワイヤー(ブラックS)		a28	L-		
ブリッジ回路	抵抗510kΩ (緑茶黄金)		h28	h27	
		抵抗200Ω (赤黒茶金)		h30	h29
	VR(10K) 調節用抵抗	左	f28		
		中央	g29		
		右	接続しない		
	みのむし①		j27		
	みのむし②		j30		
ワイヤー(グリーン)		i28	d23	SCC出力	
ワイヤー(グリーン)		i30	d24		
増幅回路 1	計装アンプ (LT1167)	1	e22		
		2	e23		
		3	e24		
		4	e25		
		5	f25		
		6	f24		
		7	f23		
		8	f22		
	ワイヤー(レッドS)		j23	R+	
ワイヤー(ブルーL)		d25	R-		
ワイヤー(ブラックL)		g25	L-		
抵抗1kΩ (茶黒赤金)		d22	g22	50.4倍	
出力	ワイヤー(ホワイトL)		h24	ADC-A0	この間にSCC波形 が出力される
	ワイヤー(ブラック)		j25	ADC-A1	

資料10. SCC測定回路の接続方法

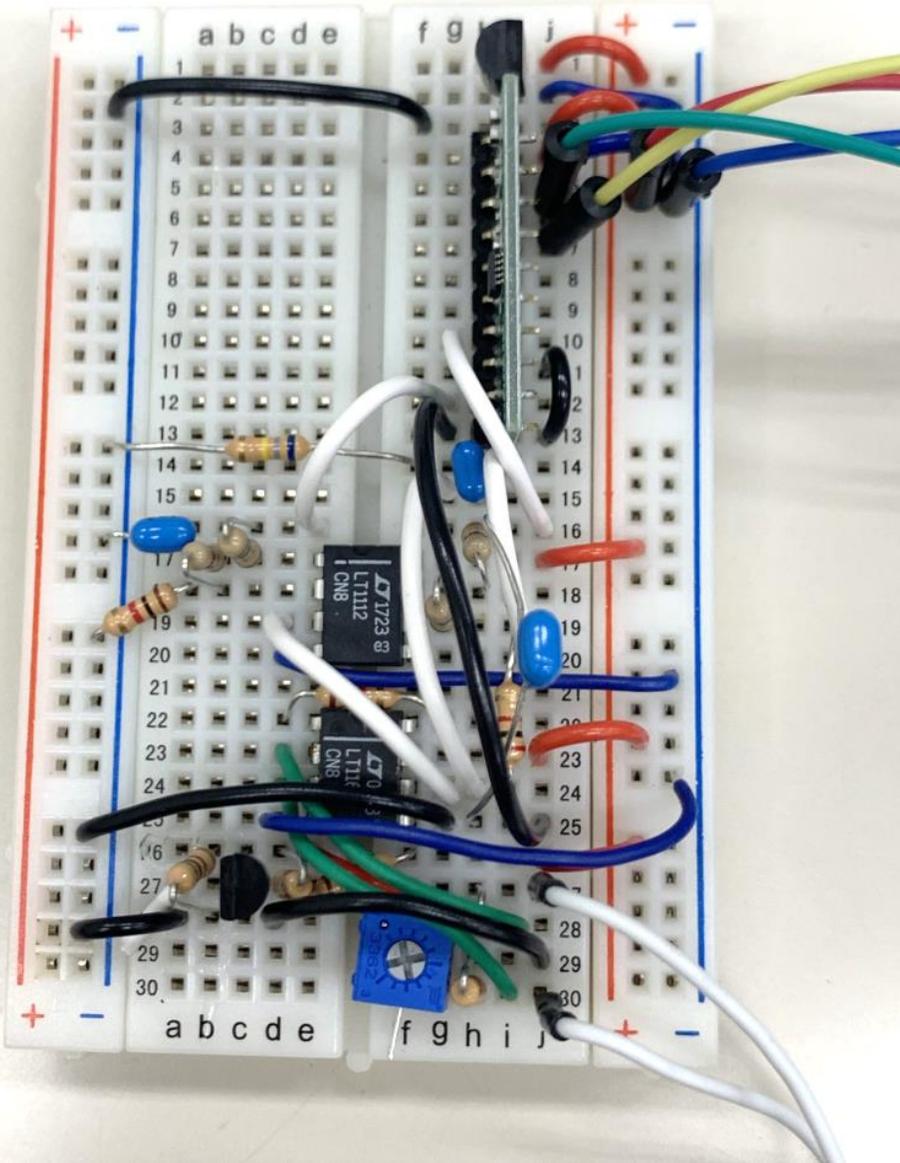
```

1 #include <Wire.h>
2 #include <Adafruit_ADS1X15.h>
3 Adafruit_ADS1015 ads;
4
5 void setup(void)
6 {
7   Serial.begin(115200);
8   // ADS1015 gain 8x input +/- 0.512V
9   ads.setGain(GAIN_EIGHT); //ゲインを8倍に設定
10  ads.begin();
11 }
12
13 void loop(void)
14 {
15   int16_t results1, results2;
16   results1 = ads.readADC_Differential_0_1(); //差動入力
17
18   float multiplier = 0.25F; //ゲイン8倍時の係数
19   Serial.print(results1 * multiplier);
20   Serial.print(",");
21   Serial.print("194,"); //24k Ohm = 42uS
22   Serial.print("-7,"); //1M Ohm = 1uS
23   Serial.println();
24
25   delay(200);
26 }

```

資料11. SCC計測に用いたプログラム

AD変換モジュールのゲインは8倍とし、入力レンジは±0.512V(プログラム9行目)、1bitあたり0.25mv(18行目)であった。42uSおよび1uSに対応する固定抵抗を回路に接続し、得られた出力電圧をプログラムに入力した(21~22行目)。



資料12. ブレッドボード上のSCL・SCR測定回路

種別	部品名	本体	ブレッドボード	備考	
入力	ワイヤー(ホワイトL)		i24	g15	SCR回路へ
	ワイヤー(ホワイトL)		i24	d19	SCL回路へ
増幅回路 2	オペアンプ (LT1112)	1	e17		
		2	e18		
		3	e19		
		4	e20		
		5	f20		
		6	f19		
		7	f18		
		8	f17		
	ワイヤー(レッドS)		j17	R+	
	ワイヤー(ブルーL)		d20	R-	
	抵抗15kΩ (茶緑橙金)		c16	c17	SCL部分
	抵抗1kΩ (茶黒赤金)		a18	L-	
	抵抗4.7kΩ (黄紫赤金)		b17	b18	
	コンデンサ106		a16	L-	SCR部分
	抵抗100kΩ (茶黒黄金)		g18	g19	
	抵抗100kΩ (茶黒黄金)		h16	h18	
	抵抗2kΩ (赤黒赤金)		i19	i25	
	抵抗680kΩ (青灰黄金)		f14	L-	
コンデンサ106		h14	h15		
コンデンサ106		i16	h25		
ワイヤー(ホワイト)		i14	j20		
出力	ワイヤー(ホワイト)		e16	ADC-A0	SCL出力
	ワイヤー(ホワイトL)		j16	ADC-A2	SCR出力
	ワイヤー(ブラック)		j25	ADC-A3	SCL/SCR 信号GND
	ワイヤー(ブラック)		ADC-A3	ADC-A1	

資料13. SCL・SCR測定回路の接続方法

```

1 #include <Wire.h>
2 #include <Adafruit_ADS1X15.h>
3 Adafruit_ADS1015 ads;
4
5 void setup(void)
6 {
7   Serial.begin(115200);
8   // ADS1015 gain 2x input +/- 2.048V
9   ads.setGain(GAIN_TWO);
10  ads.begin();
11 }
12
13 void loop(void)
14 {
15   int16_t results1, results2;
16   results1 = ads.readADC_Differential_0_1(); //差動入力
17   results2 = ads.readADC_Differential_2_3();
18
19   float multiplier = 1.0F;
20   Serial.print(results1 * multiplier);
21   Serial.print(",");
22   Serial.print(results2 * multiplier);
23   Serial.print(",");
24   Serial.print("1111,"); //24k Ohm = 42uS
25   Serial.print("-41,"); //1M Ohm = 1uS
26   Serial.println();
27
28   delay(200);
29 }

```

資料14. SCL/SCR計測に用いるプログラムと測定例(目), 1bitあたり1mv(19行目)であった。AD変換モジュールのゲインは2倍とし, 入力レンジは±2.048V(プログラム中9

```

1 #include "BluetoothSerial.h" //BT通信ライブラリ
2 #include <Wire.h>
3 #include <Adafruit_ADS1X15.h>
4 Adafruit_ADS1015 ads;
5 BluetoothSerial SerialBT; //BluetoothSerial クラスのインスタンス化
6 long t1,t0; //時間管理用変数
7 int cnt=0; //加算回数
8 int16_t results;
9 float multiplier = 0.125F;
10 float mv, average, sum; //平均値算出用変数
11
12 void setup(void)
13 {
14   pinMode(10, OUTPUT); //内蔵赤色LEDを出力モードに
15   digitalWrite(10, LOW); //内蔵赤色LEDを点灯
16
17   Serial.begin(115200);
18   SerialBT.begin("EMG32"); //Bluetooth device name
19   Wire.begin(32, 33); //32端子をSDA, 33端子をSCLとしてI2C通信開始
20
21   // ADS1015 gain 16x input +/- 0.256V
22   ads.setGain(GAIN_SIXTEEN);
23   ads.begin();
24 }
25
26 void loop(void)
27 {
28   results = ads.readADC_Differential_0_1(); //作動入力
29   mv=abs(results * multiplier); //筋電位の絶対値を得る
30   sum+=mv;
31   cnt++;
32
33   t0=t1;
34   t1=millis()/10; //10ms間隔で平均mv数を表示
35   if(t1!=t0){
36     average=sum/cnt;
37     SerialBT.print(average);
38     SerialBT.print(",100,0");
39     SerialBT.println();
40     sum=0;
41     cnt=0;
42   }
43 }

```

資料15. EMGの無線計測に用いるプログラム Bluetooth化された計測装置は, PCのBluetoothデバイス検索機能を用い, 18行目で指定された名前を検出, 登録することで利用可能となる。プログラム中では, EMG測定の際絶対値に変換し(29行目), さらに10msの平均mV数を送信する形式(33~42行目)としている。

```

1 #include <WiFi.h> //Wifiライブラリ
2 #include <WiFiMulti.h> //複数のアクセスポイントを管理するためのクラス
3 #include <HTTPClient.h> //HTTP通信ライブラリ
4 #include <Wire.h>
5 #include <Adafruit_ADS1X15.h>
6
7 WiFiMulti wifiMulti; //WiFiMultiクラスのインスタンス化
8 Adafruit_ADS1015 ads;
9
10 void setup(void)
11 {
12   pinMode(10, OUTPUT);
13   digitalWrite(10, LOW);
14
15   Serial.begin(115200);
16   Wire.begin(32, 33); //I2C connection
17
18   ads.setGain(GAIN_EIGHT); // ADS1015 gain 8x
19   ads.begin();
20
21   delay(4000); //4秒待機した後に通信開始
22   wifiMulti.addAP("MySSID", "MyPassword"); //WiFiアクセスポイントの登録
23 }
24

```

```

25 void loop(void)
26 {
27   int16_t results;
28   float multiplier = 0.25F;
29   float SC;
30   results = ads.readADC_Differential_0_1(); //作動入力
31   SC=results * multiplier;
32
33   if((wifiMulti.run() == WL_CONNECTED)) { //登録したアクセスポイントに接続
34     HTTPClient http; //HTTPClientをインスタンス化
35     Serial.print("[HTTP] begin...\n");
36     String URL="http://hogehoge.ne.jp/hogehoge/getsc.php";
37     URL+="?SC="+String(SC);
38     http.begin(URL); //HTTP通信開始
39
40     Serial.print("[HTTP] GET...\n");
41     Serial.println(URL);
42     int httpCode = http.GET(); //URLをGET方式でリクエスト
43
44     if(httpCode > 0) {
45       Serial.printf("[HTTP] GET... code: %d\n", httpCode);
46       if(httpCode == HTTP_CODE_OK) { //通信に成功したら
47         String payload = http.getString(); //通信結果をpayloadへ格納
48         Serial.println(payload);
49       }
50     } else { //失敗したらエラーとコードを表示
51       Serial.printf("[HTTP] GET... failed, error: %s\n",
52         http.errorToString(httpCode) .c_str());
53     }
54     http.end(); //HTTP通信終了
55   }
56   delay(1000); //1秒ごとに繰り返す
57 }

```

資料16. SCCをWi-Fi経由でサーバーに送信するプログラム

通信に必要なライブラリを呼び出し(1~3行目), 接続先Wi-FiルータのSSIDとパスワードを指定してアクセスポイントとしてESP32をスタートし(22行目), 1秒間隔で測定されたSCC値をサーバーにHTTPプロトコルで送信している(33~56行目)。送信の際, データはURLのあとに "getsc.php?SC=34.56" などのようにパラメータとして渡し, GET方式で送信した(36~38行目)。

```
1 <?
2 $SCval=$_GET['SC']; //Arduinoから送られたデータを受け取る
3 $timestr=strftime("%Y%m%d_%H%M%S"); //タイムスタンプ作成
4
5 if($SCval!=""){
6     $fp=fopen("SCdata.csv","a"); //追加形式でファイル開く
7     fprintf($fp,"%s,%s\n",$timestr,$SCval); //タイムスタンプとデータを保存
8     fclose($fp); //ファイル閉じる
9     echo"saved SC value!"; //Arduino側へ文字列送信
10 }
11 ?>
12
13
```

PHP Hypertext Preprocessor file length : 423 lines : 13 Ln : 13 Col : 1 Pos : 424 Windows (CR LF) UTF-8 INS

資料17. Wifi経由受信したSCCデータをサーバーに記録するプログラム

GET方式で送信されたSCC値は2行目で取得され、タイムスタンプとともにサーバ上のディスクに記録された(5~10行目)。記録の際はファイルを追加方式で開き(7行目)、新たに送信されてきたデータがタイムスタンプとともにカンマ区切りで保存される方式を用いた