

皮膚コンダクタンスを測定する安価な回路の設計と 虚偽検出実験への応用

櫻井 優太

Designing of inexpensive circuits for measuring the skin conductance
and its application to lie detection experiment.

Yuta Sakurai

要旨

精神生理学的な研究で用いられる生理反応の測定機器は一般に高価であるため手軽に用意することができない。また、市販の機材は高度に複雑化しているため内部の仕様が明らかでないことも多い。そこで本稿では安価に製作可能な皮膚コンダクタンスの測定回路を設計し、実際に製作すると共に虚偽検出実験への応用を試みた。ブレッドボードや計装アンプLT1167, オペアンプLMC6484, 各種抵抗器やコンデンサーなど安価に入手可能な部品を組み合わせ、皮膚コンダクタンス水準 (SCL) と皮膚コンダクタンス反応 (SCR) 測定のための回路を作成した。回路によって増幅された信号はArduinoを経由してコンピュータに伝えられ、Processingによってグラフ化とファイルへの記録をおこなった。この装置を用いて虚偽検出実験をおこなったところ、虚偽に伴うSCLやSCRの変動が明瞭に確認できた。安価に使用できる電極についてなど検討事項が残ったが、自作測定器の利便性を確認する事ができた。

キー・ワード：皮膚コンダクタンス, Arduino, 虚偽検出

はじめに

精神生理学の領域の研究では、知覚、注意、記憶、運動、あるいは感情など種々の心理的活動に伴う身体反応をとらえる。指標となるのは脳波、脳血流などの中枢神経指標や、心電図、筋電図、脈波、呼吸、皮膚温度などの末梢神経、自律神経指標である。特に感情やストレスに関する研究では、それらが生じることに伴う自律神経反応を測定し、感情やストレスの強度に関連する反応についての検討や、種々の主観的体験が生じるメカニズムそのものに関する検討が行われてきた。

しかし、このような研究で用いられる機器は非常に高価であり、測定環境を整えるのに数百万円以上の費用がかかることが少なくない。心電図な

ど基礎的な指標においてもこの傾向は同様であり、精神生理学的な研究を行う上での障壁のひとつとなっている。また、この障壁は精神生理学の教育においても問題となる。測定機器を用意するのが難しく、授業の中で測定を体験する機会を得る事が難しいことや、あるいは機材を用意できてもその個数が充分でなければ、スムーズに授業を進行することは難しく、機材の順番が回ってくるまでの待ち時間が生じてしまう。

市販の機材を用いることの問題点は高価であることに加えて、高度に洗練され複雑化しているため内部の細かい仕様が明らかにされていない場合があることや、研究目的に応じた調整を加えることが難しいという点もあげられる。仕様が明らかでないことは、研究において得られた測定結果の

扱いに問題を生じさせ、教育においては測定器を「ブラックボックス」としてとらえる傾向にもつながってしまう。十分な精度の測定を行うためには、測定器そのものの原理や仕様について十分に理解しておく必要があり、このことは教育上重要でもある。また、研究目的に応じた調整という点については、特に応用的な研究を行う上で問題となる。研究の目的や対象となる場面によっては、例えば小型軽量で持ち運びが容易な機器を使用しなければならないことや、屋外などで実験を行う時には交流電源に頼らない電源、例えば乾電池などで測定をしなければならないことがある。そのような個々の実験目的の全てに適合する市販機器があるわけではない。研究目的に応じて、機材を調整したり作成したりする必要がある。市販の機器を用いるのではなく機器を自作することにメリットがある。

このような背景があるものの、機材の自作そのものに技術的または費用的な面で高いハードルがあるのであれば、結局それが障壁になってしまう。しかし、近年は以下に示すような製品の登場により、生理測定の機器を自作することが容易になってきている。

まず、Arduino (Arduino LLCおよびArduino SRL 製) や Raspberry Pi (Raspberry Pi Foundation製) などの単一基盤マイクロコンピュータが安価に入手できるようになったことが挙げられる。これらを用いる事によって、市販の測定器と同じように測定値をコンピュータに入力し、データを記録、分析できる。電子工作やプログラミングのスキルが必要になるものの、容易になってきている。書籍やwebページに豊富な情報があることも注目される。これらのマイクロコンピュータを用いた場合、その設計によっては小型軽量の機器を作ることや、乾電池で動作するように作ることも可能である。

次に、電子回路を作るために便利な製品が登場したことが挙げられる。例えばブレッドボード (ソルダーレス・ブレッドボード) は所定の配線がなされた「部品の土台」となる部品であるが、このブレッドボード上の穴に部品の端子を差し込むだけで回路を作ることができる。手間のかかる

ハンダ付けをすることなく、工具としてもニッパーやワイヤーストリッパー (導線の被覆剥き) で手軽に回路を作ることができる (ただし、ブレッドボードは部品を完全に固定しないため部品が容易に抜け落ちてしまう。従って、長期にわたって使用するには基盤にハンダ付けするのが良い)。また、オペアンプなどの増幅器が安価に入手できる点も挙げられる。生理的な反応は微弱な電位であることが多く、記録のためにはそれを大きな電位へと増幅する必要があるが、そのための部品を入手するのは容易である。ブレッドボードやオペアンプは数百円程度で入手できるものが多く、安価である。

以上のことから、本稿ではArduinoとブレッドボード、オペアンプを用いる生理反応測定器の製作法およびそれを用いた心理学的実験への応用例を示す。測定する生理反応としては、発表例が少ないことと、精神生理学の授業における実験実習でよく取り上げられる虚偽検出 (ウソ発見) 実験へ応用しやすいことから、皮膚コンダクタンスを選択する。

回路の設計

皮膚コンダクタンスを含む皮膚電気活動 (electrodermal activity) の測定法については、Fowles, Christie, Edelberg, Grings, Lykken, & Venables (1981) が標準的な方法について勧告を発表している。電位法や通電法などいくつかの方法が示されているが、本稿では、通電法による皮膚コンダクタンス測定に注目する。Fowles et al. (1981) の回路は、新美 (1983) により国内における部品入手性の点から検討が加えられている。これらの情報を参考に、安価かつ容易に製作できることや、Arduinoを用いてコンピュータに測定値を記録することを念頭に増幅回路を設けることなど、以下の変更を行った。

1. ホイートストンブリッジに入力する電源を乾電池から半固定抵抗で分圧し0.5 Vを得る方式にした。より精度の良い測定を行うためには定電圧ダイオードを用いて正確に0.5 Vを作る必要があるが、実験開始時に確認を行えば数十分間程度

の実験には問題がないため省略した。

2. ホイートストンブリッジ部の抵抗器を簡略化し、抵抗器を安価に入手できるものに代替した。Fowles et al. (1981) の回路では可変抵抗器によってホイートストンブリッジのバランスを取るようになっているが、固定抵抗器に変更しても約 $0 \mu\text{S} \sim 約 40 \mu\text{S}$ の範囲を測定することができたため変更した。虚偽検出実験においては充分に実用的な範囲であると思われる。

3. Arduinoのアナログ入力ポートに入力するため、計装アンプLT1167 (Liner Technology社製) とオペアンプLMC6484 (National Semiconductor社製) により、複数段の増幅回路を設けた。出力部の直前に高周波ノイズを減衰させるローパスフィルタを設けたが、測定環境によってフィルタの特性を調整する必要があると思われる。また、Arduinoのアナログ入力ポートに負電圧がかからないようにダイオードを追加した。

4. 皮膚コンダクタンス水準 (skin conductance level, SCL) だけでなく、虚偽検出実験において有用な皮膚コンダクタンス反応 (skin conductance response, SCR) を測定できるように交流成分を取りだして、さらにそれを増幅する多段階の回路を設けた。取り出された交流成分は 0 V を中心に正負に振れるが、Arduinoのアナログ入力ポ-

ードで測定可能な $0 \text{ V} \sim 5 \text{ V}$ の範囲に収めるために 2.5 V 昇圧するレベルシフト回路を設けた。

完成した回路をFigure 1に示す。回路図の描画や回路のシミュレーションにはLTspice IV ver. 4.23g (Liner Technology社製) を用いた。Figure 1ではLTspice IVによる描画の都合で半固定抵抗を固定抵抗としている。また、乾電池は 1.2 V のニッケル水素充電式電池を4本ずつ (合計8本) 用いる事を念頭に 4.8 V としているが、 1.5 V のアルカリ乾電池を3本ずつ (合計6本) 用いて 4.5 V の電源とすることも可能である。

回路の製作

以下、順にホイートストンブリッジ部とLT1167による第1段階の増幅回路 (Figure 2およびTable 1), LMC6484による第2段階の増幅とSCLの回路 (Figure 3およびTable 2), 交流成分 (SCR) の抽出とLMC6484による第2段階の増幅およびレベルシフト回路 (Figure 4およびTable 3), ローパスフィルタ回路およびダイオードによる保護回路とArduinoへの接続 (Figure 5およびTable 4) の、各部の組み立て手順を示す。導線は便宜的に色分けしているが、部品の端子を切り落としてできる線材を用いても良い。

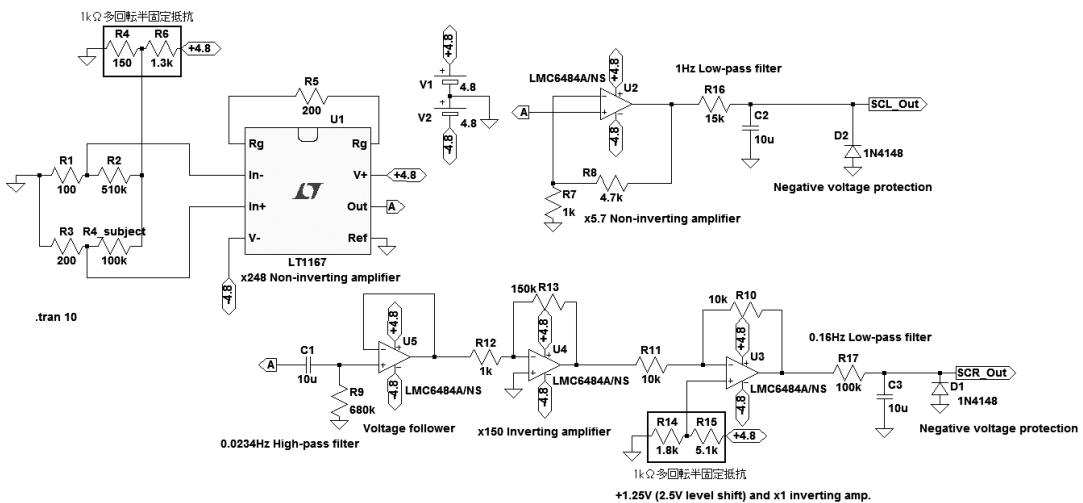


Figure 1. 皮膚コンダクタンス測定回路。

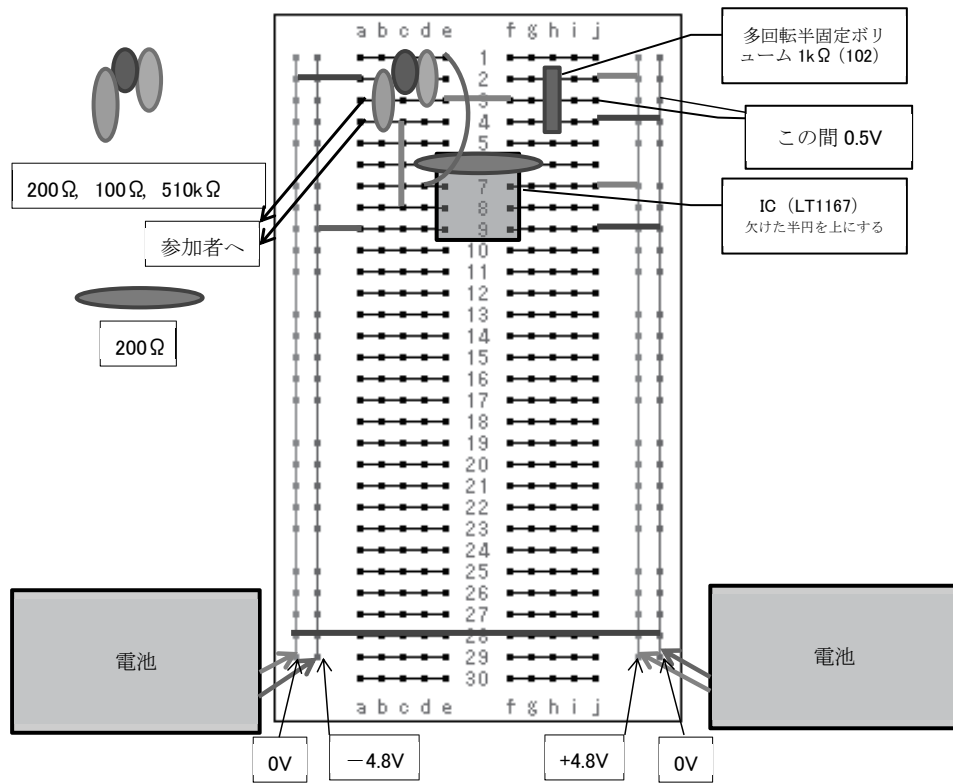


Figure 2. ホイートストンブリッジ部とLT1167による第1段階の増幅回路。

Table 1
Figure 2に含まれる部品の一覧と配線

部品	配置	備考
赤線1	j2 - 右赤 (+)	
赤線2	e3 - f3	
赤線3	c4 - c8	
赤線4	j7 - 右赤 (+)	
緑線1	左赤 (0V) - a2	
緑線2	j4 - 右青 (0V)	
緑線3	j9 - 右青 (0V)	
青線1	e1 - d7	
青線2	左青 (-) - a9	
多回転半固定ボリューム たて型3296W(1kΩ)	h2 - h4	
200Ω	b2 - b4	
100Ω	c1 - c2	
510kΩ	d1 - d3	
コネクタ付コード (みの虫×ジャンパーワイヤ)	a3およびa4	
LT1167	e6 - f6	1番ピンがe6, 8番ピンがf6
200Ω	d6 - g6	
スイッチ付き電池ボックス	プラスを右赤 (+)	
スイッチ付き電池ボックス	マイナスを右青 (0V)	
スイッチ付き電池ボックス	プラスを左赤 (0V)	
スイッチ付き電池ボックス	マイナスを左青 (-)	
緑線4	左赤 (0V) - 右青 (0V)	

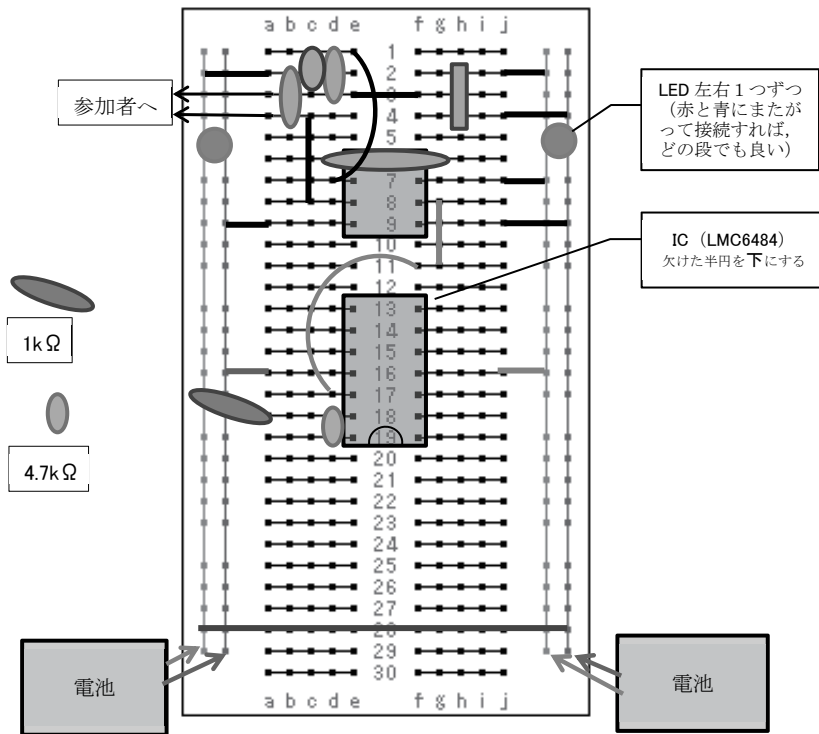


Figure 3. LMC6484による第2段階の増幅とSCLの回路。

Table 2
Figure 3に含まれる部品の一覧と配線

部品	配置	備考
赤線5	g8 - g11	
赤線6	f11 - d17	
赤線7	j16 - 右赤 (+)	
青線3	左青 (-) - a16	
LMC6484	f19 - e19	1番ピンがf19, 14番ピンがe19
1kΩ	a18 - 左赤 (0V)	
4.7kΩ	d18 - d19	
抵抗内蔵5mm赤色LED(5V用)	+端子を右赤 (+), -端子を右青 (0V)	長いリード線が+端子
抵抗内蔵5mm赤色LED(5V用)	+端子を左赤 (0V), -端子を左青 (-)	長いリード線が+端子

※LEDは通電状態を示すパイロットランプであるため、省略可能。

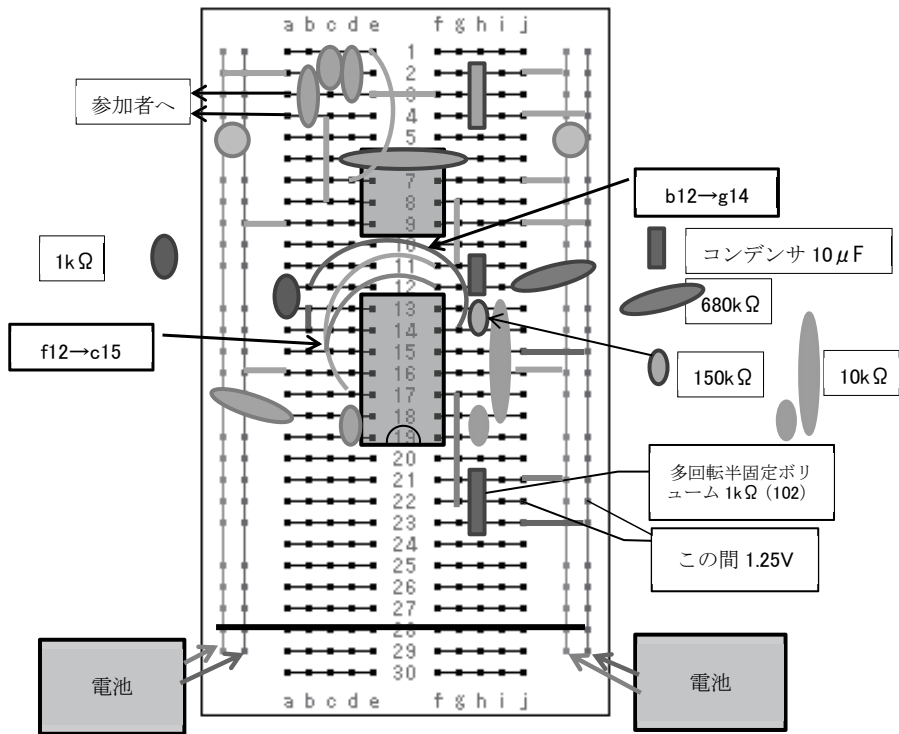


Figure 4. 交流成分 (SCR) の抽出とLMC6484による第2段階の増幅およびレベルシフト回路。

Table 3
Figure 4 に含まれる部品の一覧と配線

部品	配置
赤線8	f12 - c15
赤線9	g17 - g22
青線4	b12 - g14
青線5	b13 - b14
青線6	j15 - 右青 (0V)
1kΩ	a12 - a13
コンデンサ10μF	h11 - h12
680kΩ	j12 - 右青 (0V)
150kΩ	h13 - h14
10kΩ	i13 - i18
10kΩ	h18 - h19
多回転半固定ボリューム たて型3296W(1kΩ)	h21 - h23
赤線10	j21 - 右赤 (+)
青線7	j23 - 右青 (0V)

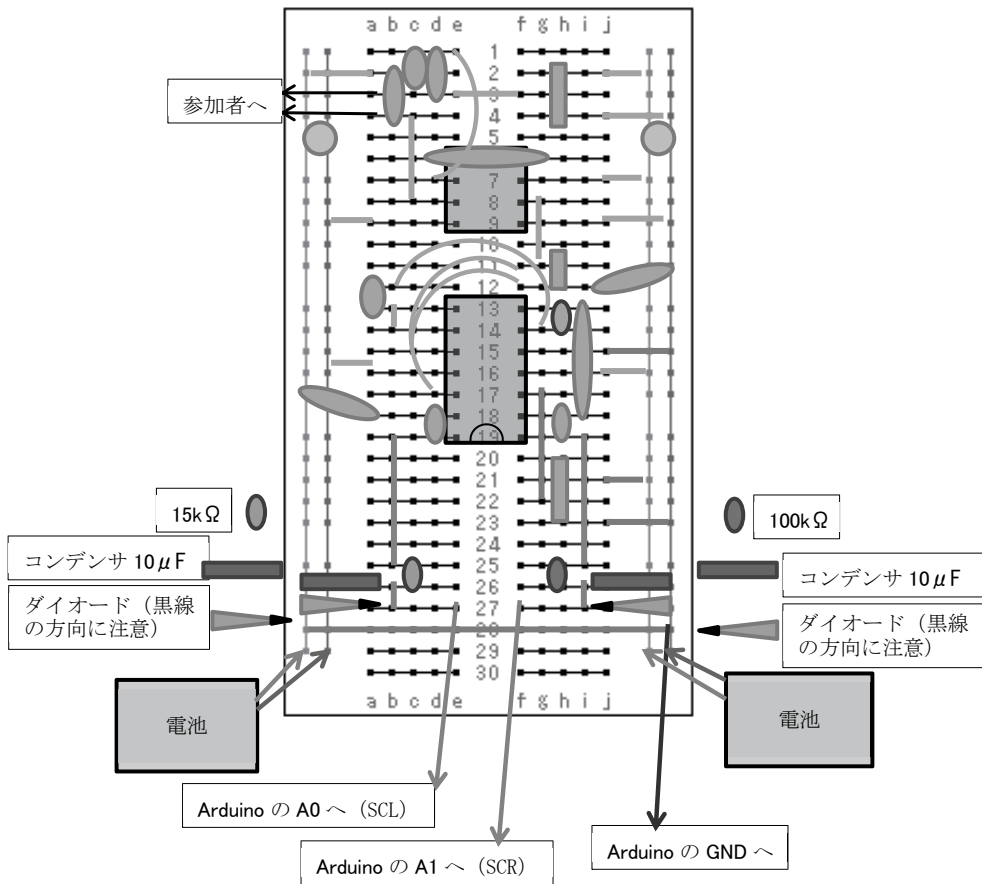


Figure 5. ローパスフィルタ回路およびダイオードによる保護回路とArduinoへの接続。

Table 4
Figure 5に含まれる部品の一覧と配線

部品	配置	備考
赤線10	b19 - b25	
赤線11	i19 - i25	
赤線12	b26 - b27	
赤線13	i26 - i27	
15kΩ	c25 - c26	
100kΩ	h25 - h26	
コンデンサ10μF	左赤 (0V) - a26	
コンデンサ10μF	j26 - 右青 (0V)	
スイッチングダイオード 1N4148	左赤 (0V) - a27	カソード端子 (黒線) がa27
スイッチングダイオード 1N4148	j27 - 右青 (0V)	カソード端子 (黒線) がj27
Arduino アナログポートA0	e27	
Arduino アナログポートA1	f27	
Arduino GND	左赤または右青の0Vいずれか	

基本的に、各部の組み立てにおいてはICや半固定抵抗など大型の部品を先に取り付けて、その部品を避ける形で導線の長さを決めて配線を行うと良い。ホイートストンブリッジ部の組み立て

(Figure 1) においては、j3と右の青線 (0 V) の間が0.5 Vになるように、テスターを見ながら半固定抵抗を調整する。同様に、SCR部のレベルシフト回路 (Figure 4) においては、j22と右

の青線 (0 V) の間が1.25 Vになるように、テスターを見ながら半固定抵抗を調整する。

今回の製作において用いた部品は、Arduinoを含めて合計約4,000円で入手でき、非常に安価に測定回路を作ることができた。

なお、筆者は学生と共に1時限の授業4週間(90分間の授業4回、合計6時間)を用いて、測定の原理を説明しながら、3人1組のグループで1台ずつの回路を製作することに成功している。実際に製作を行いながら原理を学び、最終的には虚偽検出実験で回路を実用できたため、非常に高い教育効果を得た。

電極について

測定に用いる電極と、それを参加者に装着するときに用いる電極ペーストについては今回最も苦慮した点である。皮膚電気活動について測定法から研究への応用まで詳細に記述されている新美・鈴木(1986)による記述を参照したが、電極の自作は材料の入手、製作共に困難であった。電極ペーストについては材料となる局方寒天末が既に入手困難であった。

そのため、今回は脳波測定用の銀塩化銀電極と、市販の電極ペースト(今回はParker Laboratories社製sigma cream)を使用することにした。ジャンパコード・みの虫クリップを用いて回路のa3とa4に接続部を作り(Figure 2)、電極のリード線をつなぎ、電極ペーストを塗布して参加者の指に電極を装着した。大学などの研究室に脳波計などの測定器が導入されていれば電極や電極ペーストを流用するのは比較的容易であると思われるが、そうでない場合、電極や電極ペーストを購入するには高額のコストがかかる。使い捨ての電極を用いることも可能であると思われるが、市販品のものには必ずしも安価ではないため、多数の測定を行うときには多くの費用がかかる。電極や電極ペーストを安価に用意する手段については今後の検討課題である。

データのサンプリングと記録

作成された回路をArduinoにつなぎ、さらに

USBでコンピュータに接続する。Arduinoには資料1のプログラムをArduino Software (IDE) ver. 1.6.5を用いて書き込んだ。アナログポートの値は10回サンプリングを行って移動平均を求めることにより平滑化しているが、ノイズの程度により適宜調整することができる。得られた測定値はシリアル通信によってコンピュータへと伝えられる。

シリアル通信で伝えられるデータは、Processing ver. 3.0.1で資料2のプログラムを実行することにより受信し、同時にグラフ化とファイルへの記録を行った。

プログラム中に「換算」として示される部分は、Arduinoによって得られる0 ~ 1023の値からSCLやSCRの値(μS 単位)に換算している部分である。抵抗器の誤差などで値が変わるため、新しく測定器を作成したときには、キャリブレーターとなる抵抗器を参加者の電極につながるみの虫クリップに接続し、いくつかの抵抗値を実測して換算式の値を調整する必要があると思われる。

このプログラムは、画面内をクリックすることによって「マーク」を付けることができる。例えば虚偽検出実験においては参加者に対して質問をしたタイミングでマークを付けておき、マークの前後の反応の大きさを分析することができる。

Processingによって読み取られるsettings.txtの内容は資料3に示した。「COM3」の部分はArduinoを接続したときにArduino Software (IDE)に示されるポート番号に応じて修正する必要がある。2行目および3行目は表示画面のサイズ(横、縦のドット数)であり、測定に用いるコンピュータに応じて適宜調整することができる。

虚偽検出実験への応用

本装置を用いて簡単な虚偽検出実験を行った結果を以下に示す。この実験ではトランプのカード6枚のうちから参加者に任意のカードを引かせて、その番号を覚えさせた。続いて、選ばれなかった5枚から1枚を取り除き、この4枚に参加者が選んだカードを含めてよくシャッフルし、取り除いた1枚を先頭に加えた。この手順によって、1枚

目は必ず「ウソをつかない」試行になるように調整した。

カードを1枚ずつ呈示し「あなたを選んだカードはこれですか？」の質問に対して、必ず「いいえ」と答えるように指示した。2枚目から6枚目のカードのいずれかで、参加者は1回ウソをつことになる。1枚につき20秒の間隔を開けてカードを呈示した。6枚の呈示が終了したら、カードの並び順を維持した状態で2回目、3回目と同様のカード呈示を行い、全て「いいえ」と答えさせた。カードが呈示されたタイミングで測定画面をクリックしてマークを付けた。この手続きを3回繰り返して、ウソに伴うSCLやSCRの変動が認

められるか検討した。

男子学生1名がこの実験の参加者となって得た測定値をFigure 6に示す。実験開始直後の1枚目や最後の6枚目にSCL・SCRの反応があったが、3回の試行で共通して、3枚目のカード（3つめのマーク以降）において大きなSCLとSCRの変動が認められた。実際にこの実験中において虚偽のカードとなっていたのが3枚目であり、虚偽に伴う反応を鋭敏にとらえる事ができたといえる。しかし、波形には細かいノイズが乗っていて、場合によっては波形を明確にとらえるのを妨げてしまう可能性が示された。今回設計した増幅回路のノイズ対策にはさらなる検討の余地がある。

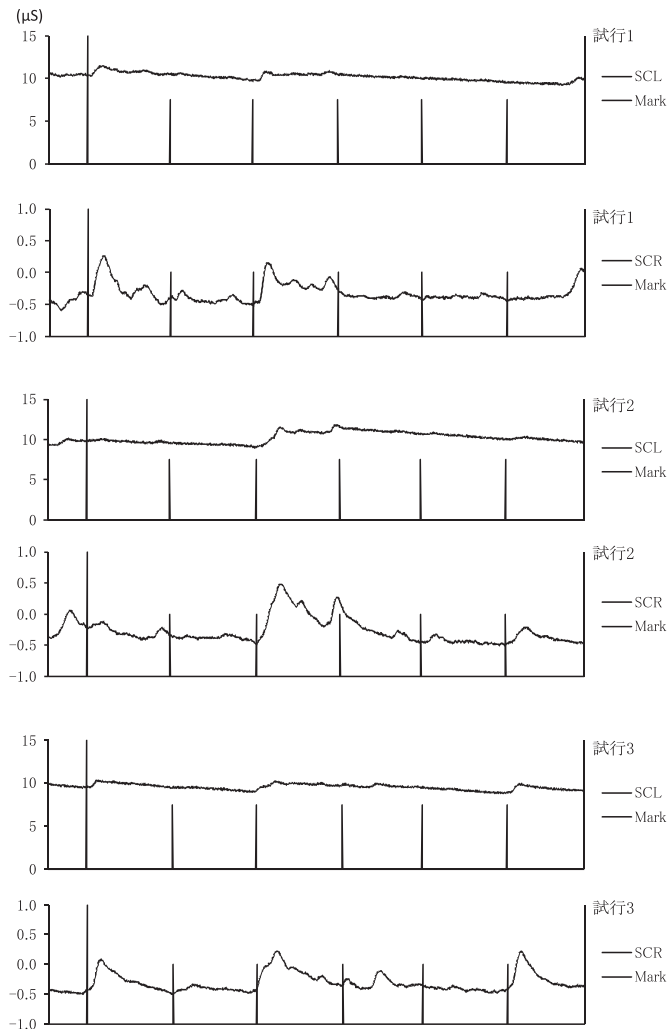


Figure 6. 虚偽検出実験への応用例。
(Markはカードが呈示されたタイミングを示す。20秒間隔でカードを呈示した。)

まとめ

ブレッドボードや計装アンプLT1167, オペアンプLMC6484, 各種抵抗器やコンデンサーなど安価に入手可能な部品を組み合わせ、皮膚コンダクタンス水準 (SCL) と皮膚コンダクタンス反応 (SCR) 測定のための回路を作成することができた。手軽に実験を始めることができるだけでなく、このような機材作成そのものを精神生理学の授業に取り入れ可能であることが大きな利点である。今後は、他の生理反応をとらえる装置の設計・製作や、今回の試作から得られた、安価な電極の必要性や、十分なノイズ対策などの点について、さらに検討する必要がある。

謝 辞

今回の機材設計にあたって、文京学院大学 長野祐一郎先生より大変貴重な情報を賜りました。ここに感謝を申し上げます。

引用文献

- Fowles, D. C., Christie, M. J., Edelberg, R., Grings, W. W., Lykken, D. T. & Venables, P. H. (1981). Publication Recommendations for Electrodermal Measurements. *Psychophysiology*, 18, 232-239.
- 新美 良純 (1983). 精神生理学会 (アメリカ) 勧告による皮膚コンダクタンス測定器の試作心理学研究, 54, 325-327.
- 新美 良純・鈴木 二郎 (編) (1986). 皮膚電気活動 星和書店

資料1 Arduinoに書き込んだプログラム

```
// センサーの値を格納する配列
int sensors[2];
// 受信するシリアル通信の文字列
int inByte;

// 移動平均のためのサンプリング回数
const int numReadings = 10;
// アナログポートから読み取った値を格納する変数
int readingsA[numReadings];
// 現在読み取った値のインデックス番号
int readIndexA = 0;
// 合計値
int totalA = 0;
// 平均値
int averageA = 0;

int readingsB[numReadings];
int readIndexB = 0;
int totalB = 0;
int averageB = 0;

void setup() {
  Serial.begin(9600);

  // アナログ読み取り値を0で初期化
  for (int thisReading = 0; thisReading < numReadings;
  thisReading++) {
    readingsA[thisReading] = 0;
    readingsB[thisReading] = 0;
  }

  // センサーの値と受信する文字列を初期化
  sensors[0] = 0;
  sensors[1] = 0;
  inByte = 0;
  // 通信を開始
  establishContact();
}

void loop() {
  // 最も古い値を合計値から引く
  totalA = totalA - readingsA[readIndexA];
  // アナログポートの値を読み取る
  readingsA[readIndexA] = analogRead(0);
  // 合計値に加算する
  totalA = totalA + readingsA[readIndexA];
  // インデックス番号を次に進める
  readIndexA = readIndexA + 1;

  // インデックス番号が配列の最後に来た場合
  if (readIndexA >= numReadings) {
    readIndexA = 0; //インデックス番号を先頭に戻す
  }
  // 平均値の計算
  averageA = totalA / numReadings;

  // A1ポートについても同様の処理をする
  totalB = totalB - readingsB[readIndexB];
  readingsB[readIndexB] = analogRead(1);
  totalB = totalB + readingsB[readIndexB];
  readIndexB = readIndexB + 1;

  if (readIndexB >= numReadings) {
    readIndexB = 0;
  }

  averageB = totalB / numReadings;
}
```

```
// サンプリングを安定させるためのディレイ
delay(1);

if (Serial.available() > 0) {
  // 受信した文字列を読み込み
  inByte = Serial.read();
  // 移動平均済みの値を変数に格納
  sensors[0] = averageA;
  sensors[1] = averageB;
  // カンマ区切りで値を送出
  Serial.print(sensors[0]);
  Serial.print(",");
  Serial.println(sensors[1]);
}
}

void establishContact() {
  // Processingから何か文字が送られてくるのを待つ
  while (Serial.available() <= 0) {
    // 初期化用の文字列
    Serial.println("0,0");
    delay(300);
  }
}
}
```

資料 2 Processingのプログラム

```
import processing.serial.*;
Serial myPort; // シリアルポート

String lines[]; //設定ファイルの行番号
int screen_height,screen_width; //画面サイズ

int xpos = 0; //線の描画位置 (x座標値) (0で初期化)
float maxHeight; //画面半分の高さ

PFont font; //フォント

int scl = 0, scr = 0; //SCLとSCRの値 (0で初期化)
int sclYpos0 = 0, scrYpos0 = 0; //SCLとSCRの直前のY座標値 (0で初期化)
int mark = 0; //マウスによるマーク

String Data[]; //出力用配列
int dcnt = 0; //データのカウンタ

void setup(){
  //settings.txtより画面サイズ, シリアルポート番号の読み取り
  lines = loadStrings("settings.txt");

  //settings.txtの1行目に基づいてシリアルポートを開く
  myPort = new Serial(this, lines[0], 9600);
  myPort.clear(); //受信データをクリア

  //settings.txtの2行目, 3行目から情報から画面サイズを指定
  screen_width=int(lines[1]);
  screen_height=int(lines[2]);

  size(800,600); //画面サイズを仮設定
  surface.setSize(screen_width, screen_height); //画面サイズ初期化

  //画面縦幅の2分の1を求める (各値の最大高さとなる)
  maxHeight = float(screen_height) / 2.0;

  font = loadFont("AgencyFB-Reg-24.vlw"); //フォント
```

```
ト読み込み

Data=new String[32000]; //データ出力用配列の初期化

Data[dcnt]="SCL,SCR,Mark"; //出力ファイルのヘッダ
dcnt++; //データカウンタを進める
saveStrings("result.csv", Data); //ヘッダを出力する

refresh(); //画面初期化

frameRate(10); //秒間フレームレートを10に設定 (サンプリングレート10Hz)

// 改行コード(¥n)が受信されるまで、シリアルメッセージを受けつづける
myPort.bufferUntil(¥n);
}

void draw(){
  int sclYpos, scrYpos; //描画用Y座標
  float scl2, scr2; //換算後のジューメンズ値

  String str; //データ出力用文字列

  if(xpos == 0){refresh();} //初回およびxposが0に戻った時の処理, 画面初期化

  //SCLを1023 (最大値) に対する割合として求めY座標とする
  sclYpos = int(maxHeight - ( float(scl) / 1023.0 ) * maxHeight );

  //SCRを1023 (最大値) に対する割合として求め, さらに画面半分の高さを足してY座標とする
  scrYpos = int(maxHeight - ( float(scr) / 1023.0 ) * maxHeight) + maxHeight);

  //換算
  scl2 = round( (float(scl) * 0.036034 + 0.330606 ) * 100.0 ) / 100.0;
  scr2 = round( ( float(scr) - 512.0 ) / 397 ) * 100.0 ) /100.0;

  stroke(255, 255, 255); fill(255, 255,255); rect(0, 30, 100, 40); //SCLの値を消す
  textFont(font,24); fill(0, 102, 153); text("SCL " + scl2, 10, 60); //SCLの値を表示する (uS)

  //SCRの値を消す
  stroke(255, 255, 255); fill(255, 255, 255); rect(0, maxHeight + 1, 100, 40);
  //SCRの値を表示する (uS)
  textFont(font,24); fill(0, 102, 153); text("SCR " + scr2, 10, maxHeight + 30);

  stroke(0,102, 153);
  line(xpos-1,sclYpos0,xpos,sclYpos); //SCL (上部) 描画
  line(xpos-1,scrYpos0,xpos,scrYpos); //SCR (下部) 描画
  xpos++; //x座標値を進める

  sclYpos0 = sclYpos; //直前の座標値をストックする
  scrYpos0 = scrYpos;

  if(mark==1000) //マークがONだった場合
  {
    //マーク値を加えて出力する
    Data[dcnt]=str(scl2) + "," + str(scr2) + "," +
```

```

str(mark);
  mark =0; //マークをリセット
}
else //マークがOFFだった場合
{
  Data[dcnt]=str(scl2) + "," + str(scr2);
}

dcnt++; //データカウンタを進める

//10サンプル (1秒) ごとに値を出力する
if(dcnt % 10 == 0){
  str=" , "; //最後に空行を追加する
  Data[dcnt]=str;
  saveStrings("result.csv", Data);
}

if(xpos>screen_width){xpos=0;} //描画位置が画面右端
に到達した場合にxposをリセットする
}

void refresh(){ //画面の初期化 (xposが画面右端 (上限)
に達した時も実行)
  background(255); //白で背景初期化
  stroke(0,0,0); //255, 100, 100); //描画色を指定
  textFont(font,24); //テキストのフォント,文字サイズを
指定
  fill(0,0,0); // 102, 153); //
  text(lines[0], 10, 30); //comポート名を表示 (座標10,60)

  //画面中央の水平線を描画する
  line(0, screen_height / 2, screen_width, screen_height
/ 2);
}

```

```

void mousePressed(){
  //マウスボタンが押された時の処理
  stroke(0,0,0);
  line(xpos,0,xpos,screen_width); //縦ラインを引く
  mark = 1000; //マーク変数を更新
}

void serialEvent(Serial myPort) {
  // シリアルバッファを読み込み
  String myString = myPort.readStringUntil('\n');
  // 空白文字など余計な情報を消去
  myString = trim(myString);
  // コマ区切りで複数の情報を読み込む
  int sensors[] = int(split(myString, ','));
  // 読み込んだ情報の数だけ、配列に格納
  if (sensors.length == 2) {
    scl = sensors[0];
    scr = sensors[1];
  }
  // 読み込みが完了したら、次の情報を要求
  myPort.write("A");
}

```

資料3 Processingの実行時に必要な settings.txtの内容

```

COM3
1024
650

```