

# ドローン操作が心身に与える影響

心理学科 20HP110 大栗 寛之

(指導教員：長野 祐一郎)

キーワード：自律神経活動, ストレス, 心拍数, 皮膚コンダクタンス

## 序と目的

旅客機や電車、バスなどあらゆる交通機関の運転および操縦には高度な技術と集中力が必要とされるため、しばしばストレス状態に置かれる(海野 & 橋本, 2018)。他方、心拍数 (HR) や皮膚コンダクタンス (SC) などでストレスを客観的に測定する方法が研究されている(長野, 2012; 長野, 2017; 下野, 1998)。昨今では、配送業者の人手不足などに対して、ドローンの活用が見込まれている。そこで、今回は実験室外でも測定可能な SC 測定装置を開発し、実験室外で、ドローン飛行を行った。ドローンの飛行条件の差によって生体反応すなわちストレス反応が変わるか検討した。

## 方法

実験参加者：男子大学生 7 名, 平均 21 ( $SD=1.2$ ) 歳。  
実験課題：撮影用ドローンを用いて、自由飛行と撮影飛行の 2 条件で課題を遂行してもらった。  
心理指標：小川ら(2000)による一般感情尺度を用いて、肯定的感情(PA) 8 項目, 否定的感情(NA) 8 項目, 安静状態(CA) 8 項目の計 24 項目を用いて測定した。  
生理指標：心電図(HR および RMSSD, HF Power(n.u.)), SC を測定した。  
条件配置：課題時間中に自由にドローン飛行をしてもらう自由条件と、制限時間内に可能な限り風景画像を撮影してもらう撮影条件の 2 条件を設けた。  
手続き：生理指標は、前安静期 5 分, 課題期 5 分, 後安静期 5 分の 3 期間で計 16 分間測定した。心理指標は、課題前後の 2 回回答してもらった。条件の順序をカウンターバランスし、同スケジュールを繰り返した。

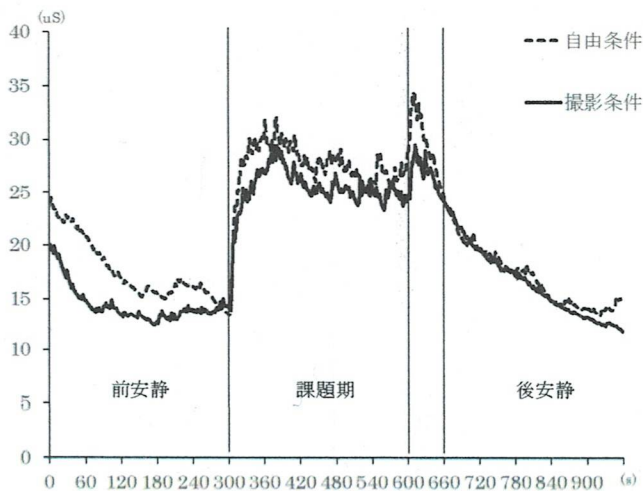


Figure1 全期間を通した SC の平均値

## 結果

完成した SC 計測器は、縦 80×横 100×高さ 20mm の大きさで収めることができた (Figure2)。重さは 123g であった。計測器を開発したことによって、実験室に限定されずに実験が可能となった。

ドローンを用いた実験では、課題内容によって差があるか分析を行った。HR と RMSSD および HF Power(n.u.), SC のそれぞれを従属変数とした 2 (飛行条件：自由, 撮影) × 3 (期間：前安静, 課題, 後安静) の 2 要因参加者内計画の分散分析を行った。HR, RMSSD および HF Power(n.u.) で有意な差は認められなかった。SC では、期間の効果のみ有意な差があった (Figure1)。また、一般感情尺度でも 2 (飛行条件：自由, 撮影) × 2 (期間：課題前, 課題後) の 2 要因参加者内計画の分散分析を行った。PA および NA, CA のすべてにおいて有意な差は認められなかった。

## 考察

統計的に有意ではないが、両条件において HR が低下していたことから、副交感神経活動が優位になった可能性が考えられた。RMSSD では自由条件で上昇、撮影条件で下降傾向があった。これについては、本来ドローン飛行で優位になりやすい副交感神経活動が、撮影課題によって抑制された可能性が考えられた。一方、課題中の SC は有意な上昇が認められ、交感神経活動の亢進によって発汗が生じたといえるが、HR や RMSSD による心臓系の反応と一貫していなかった。ドローン飛行では、心臓と汗腺活動に乖離が起きる可能性が考えられた。一般感情尺度においても有意な効果はなかったが、PA 得点は撮影条件では変化しないのに対し、自由条件ではやや下降した。NA 得点は自由条件で変わらなかったが、撮影条件では上昇した。CA 得点は両条件ともに下降した。このことから、ドローン飛行は「楽しくも緊張」するような挑戦的な課題であった可能性が考えられた。

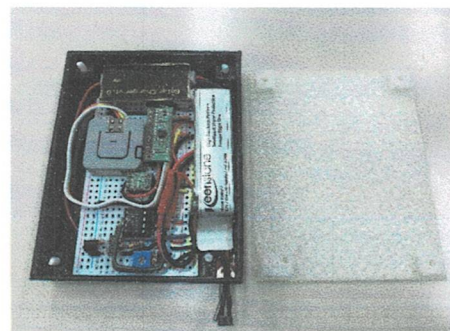


Figure2 完成した SC 計測器

ドローン操作が心身に与える影響

学籍番号 20HP110

氏名 大栗寛之

指導教員 長野祐一郎

## 序論

### 交通機関操縦中のストレス状態

飛行機、ドローン、電車、自動車など、あらゆる交通機関の操縦は高度な技術と集中力を要求されるため、操縦者はしばしばストレス状態に置かれる(海野 & 橋本, 2018)。このストレスは心理的な圧力のみならず、生理学的な反応としても現れることが知られている(Kostenko, et al, 2022)。操縦中のストレスは各交通機関の安全運航に重要な影響を及ぼすため、これを正確に測定し理解することは交通運輸システム全体において極めて重要な課題である。

### 航空機およびドローン飛行中のストレス状態

近年、ドローンの使用が急速に増加し(久保, 2017)、その操縦に伴うストレスも注目されている(Antonik, et al, 2019)。特に、ドローン操縦中の生理反応は、航空機操縦時のそれと類似していると考えられ、実際の飛行機操縦のストレス状態を模擬する貴重な代理課題となる可能性が高い。ドローン操縦は、操作の複雑さや責任の重さなど、特有のストレスが考えられるが、これらの要素は大学で行うことができる航空機操縦のシミュレーションとして大きな価値がある。したがって、ドローン操縦における心理的・生理的反応の研究は、航空機操縦中のストレス管理技術の進展に長期的に貢献するとともに、操縦関連のストレス理解を深める新たな視点を提供すると思われる。

### 心拍、心拍変動、皮膚コンダクタンスはストレス状態を示す指標となり得るか

心拍数(Heart Rate, 以下HR)、心拍変動、皮膚コンダクタンス(Skin Conductance, 以下SC)は、ストレスや感情状態を客観的に測定するための生理学的指標である(長野, 2012; 長野, 2017; 下野, 1998)。SCは精神的な発汗にともなう皮膚表面の電気の流れやすさを反映するものであり、交感神経活動の亢進を反映する(本多・長野, 2017)。心拍変動は、呼吸による心拍感覚の変動定量的に捉えたものであり、副交感神経活動の指標として広く用いられる(長野, 2017)。これらの指標は自律神経系の活動を反映し、ストレス状態を示す有用な情報を提供する。本研究では、これらの指標を用いてドローン操縦時のストレスレベルを評価し、操縦の難易度と心身の反応との関連を探索する。

### 持続可能なウェアラブル測定機器の開発

本研究では、心拍数と心拍変動の測定にはスポーツ用ウェアラブル測定器を使用し、さらに新たに開発した太陽光発電による無線SC測定器を活用した。この測定器はArduinoを基板としており、従来の商用製品に比べコストを削減しつつ必要な機能を提供する。特に、太陽光発電の統合は屋外での長時間使用における電力供給問題を解決し、環境に優しい持続可能なデバイスの実現を目指している。Arduinoの利用は、その柔軟性とコスト効率の高さから、カスタマイズされた測定機器の開発を容易にし、生体情報の測定に新たな可能性を開いている(櫻井, 2017; 長野他, 2022)。このような取り組みは、ウェアラブル測定機器の開発における新たな潮流を生み出し、今後の研究での応用が期待されることから、本研究においても積極的に取り入れた。

## 本研究の目的

本研究の主な目的は、実験に合わせた測定機器を開発することを目指した。加えて、ドローン操縦時のストレスレベルを HR、心拍変動、SC を用いて測定し、操縦環境と心理的・生理的反応との関連を解明することである。また、副次的にドローン操作の安全性向上とストレス管理に役立つ知見を提供することを目指す。

## 研究 1

### 目的

一般的に、SC を測定するには研究機器を投入する必要がある。測定機器には高額な予算が必要になることが多い。また、定期的なメンテナンスによって測定環境を維持する必要がある、測定場所が固定される。研究 2 のように、実験室外で測定できることを想定した。同時に、無線化を施して電源による計測情報の乱れを防止した。さらに、昨今では人々や環境のために持続可能な開発が盛んである。そのため、研究 1 では持続可能かつ簡便なワイヤレス SC 測定機器の開発を目指した。

### 方法

本研究では、屋外でも使用可能なワイヤレス SC 測定機器を開発対象とした。今回の開発にあたって、長野ら (2022) を参考に改良した。具体的には、4 点の改良を行った。まず、ワイヤレス化のために Bluetooth 通信が可能となるようにプログラムの加筆修正を行った。次に、ソーラーパネルを組み込むように電気回路を見直した。また、多くの人が実践可能なように市場において入手可能な部品等を用いるようにした。4 点目は、上記の内容をひとまとめにし、持ち運べるように設計した。

以下に、どのように改良したか具体的に記述する。1 点目のプログラムの改良では、長野ら (2022) によるプログラムを無線通信も可能なように加筆修正した。2 点目の太陽光充電化では、ソーラーパネルを用いて電源回路を見直し、増設した。これに伴って、電源電圧は 3.7V から 5V となった。3 点目の入手可能性では、入手可能な部品を使用するように選定した。4 点目のウェアラブル化では、ソーラーパネルのサイズに合わせて外装を作製し、必要部品を内部に収めるようにした。

## 結果

### 必要部品

長野ら (2022) を元に、開発するのに必要だった部品は以下の通りである。ATOM Lite (M5Stack 社製)、5V 出力昇圧 DCDC コンバーター(AE-XCL102D503CR-G)、ADS1015 使用 PGA 機能搭載 12bitAD コンバーター(AE-ADS1015)。無線状態でも電源供給を行うため、3.7V/550mAh バッテリー (Keenstone 社製) を使用した。太陽光充電を可能とするた

めに、Arduino 用ミニ lipo ソーラーバッテリー充電パネルおよび 1 W ソーラーパネル 80×100 を使用した。上記の電気回路を接続するために、ブレッドボード配線パターンタイプの片面ガラスユニバーサル基板(AZ0526)を使用した。ユニバーサル基板に ADS1015 使用 PGA 機能搭載 12bitAD コンバーターと 5V 出力昇圧 DCDC コンバーターを組み込んだ様子を Figure1-1 に示した。太陽光充電化にあたって必要な部品も Figure1-2 に示した。

マイコンや電気回路、ソーラーパネルなどを最小限に収めるために、計測機器の外装は 3D モデリングソフトウェア (テトラフェイス社製 metasequoia4) と 3D プリンター (Anker 社製 AnkerMake M5) を用いて作製した。また、実験を阻害しないような簡素な見た目とした。ソフトウェア上での 3D データの様子は Figure1-3 に示した。完成品の内部と外観の様子は Figure1-4 に示した。

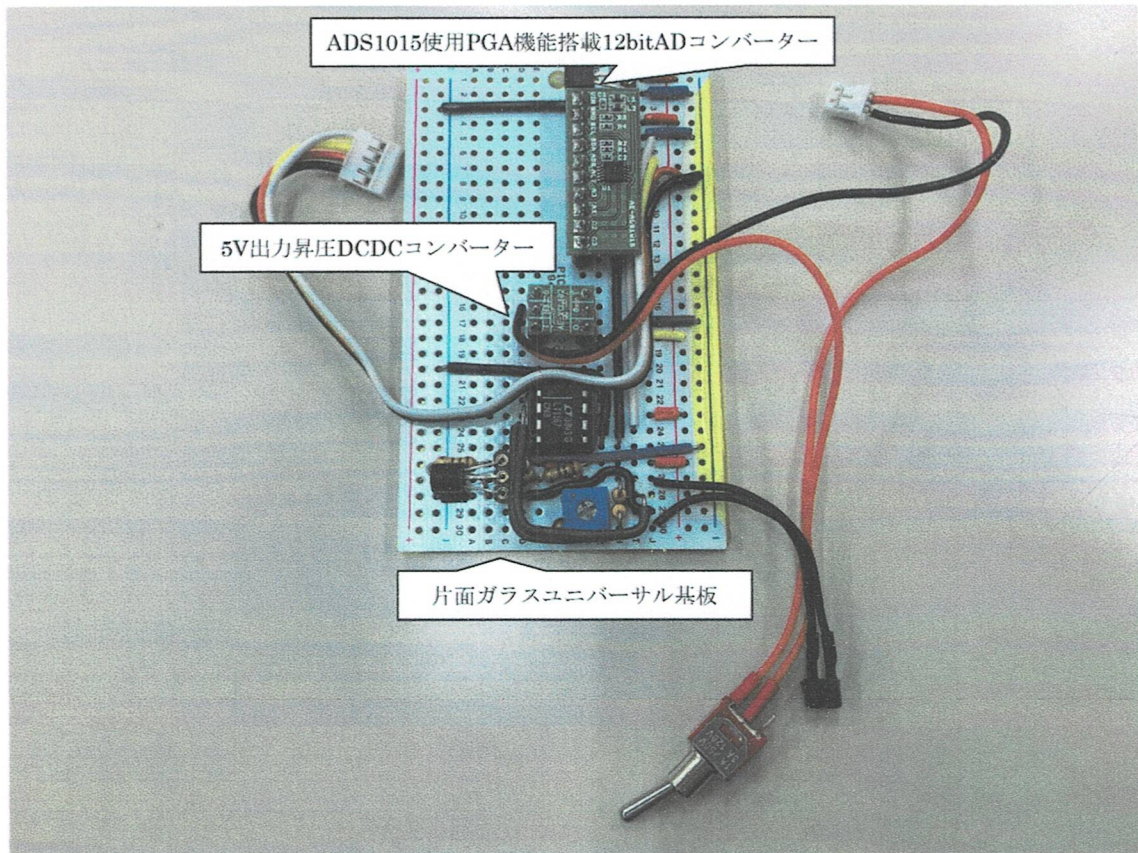


Figure1-1 SC 測定のために電子部品を回路基板に取り付けた様子

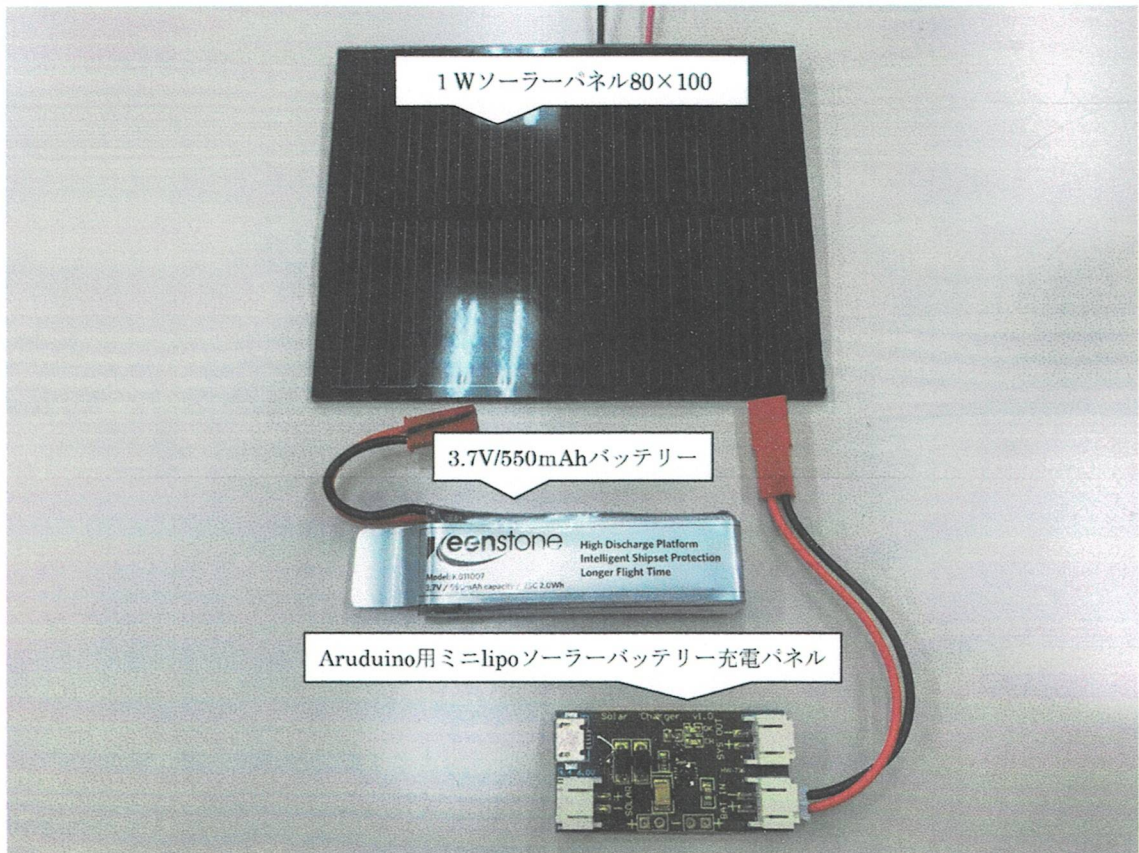


Figure1-2 ソーラーパネルとバッテリーおよび接続モジュール

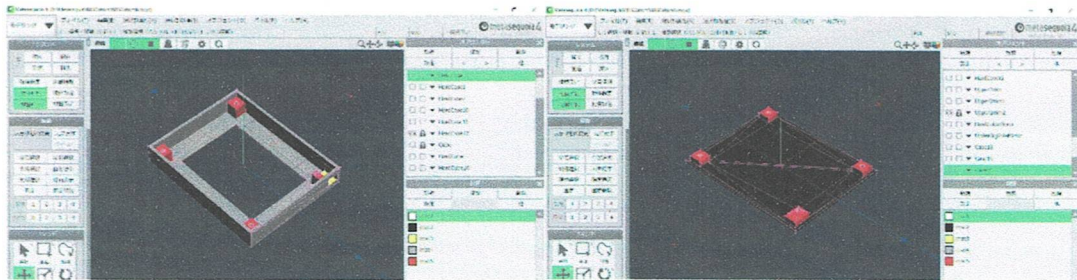


Figure1-3 外装の3Dデータ設計の様子

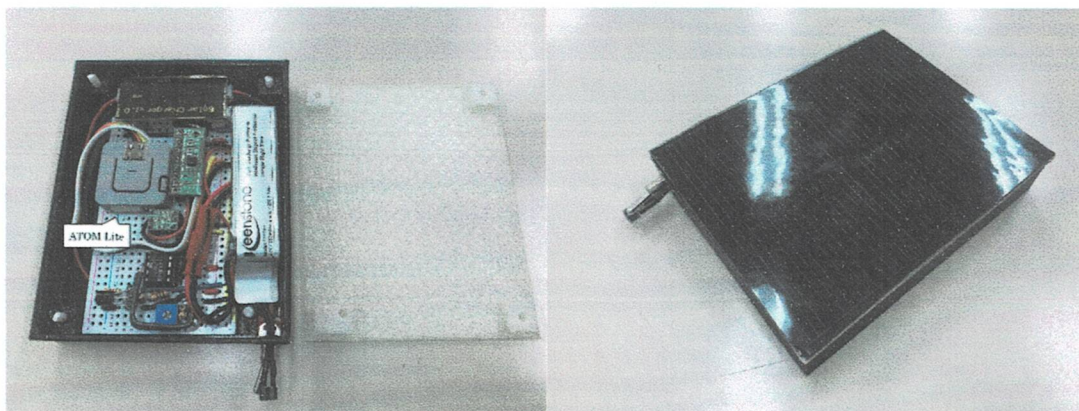


Figure1-4 計測器内部と完成品の様子

計測器マイコンには Windows 版 Arduino 開発環境 1.8.13 でプログラムの書き込みを行い、PC では Processing 開発環境 3.5.4 を用いて SC を計測し、記録できるようにした。プログラムの詳細は、付録に添付した。

#### 組立および完成品

組み立てるうえで、電源や順番に気をつけなければ、短絡や正常に計測できなくなってしまう危険があった。そのため、一度に組み立て、完成品とはしなかった。まず、ブレッドボード型基板による電気回路の試験を有線電源で運用した。同様に、マイコンの動作も検証した。次に、バッテリーと接続し、無線化を確認した。同時に、ソーラーパネルによる充電も確認した。ケースに組み込む際も、電気回路、ソーラーパネル、電池の順番に組み立て、ショートに注意した。

完成した SC 測定機器は、スイッチおよびケーブル接続部分を除いて縦 80×横 100×高さ 20mm のサイズであった。電極接続用ケーブルを除く重量は 123g であった。起動時には、ライトが光るように施した。計測機器単体では機能しないものの、PC との無線通信によって測定と、記録ができた。

#### 考察

本研究の目的は、実験環境のために持続可能かつ簡便なワイヤレス SC 測定機器の開発を目指すことであった。

結果より、Bluetooth 通信が可能なコンパクトかつ、太陽光充電によって電池駆動が可能な SC 計測器を低コストで作製することができた。また、Bluetooth 通信によって、電源による計測時のノイズを除去することに成功した。さらに、ワイヤレス化によって被験者に対する身体的制約を減らすことができた。同時に、屋外で行われることが多いドローン飛行が可能となった。固定電源を必要とする従来型の測定機器では、実験室から離れることができないなど不便な点が多かった。今回のワイヤレス化によって、ドローンに限らず他の実験の可能性を高めることができた。

縦および横の大きさは、ソーラーパネルのサイズまで抑えることに成功した。ソーラーパネルによって太陽光充電が可能となり、屋外での使用に適した測定機器となった。また、昨今の需要に合わせた、次世代のことを考えた測定機器となった。

## 研究 2

### 目的

本研究の目的は、ドローン操作における操作者の心理的および生理的反応を、実際の環境と模擬された環境で比較分析することであった。具体的には、屋外でのドローン操縦を想定し、自作のポータブル SC 計測器と、ウェアラブル心電図計を使用して、操作者のストレスレベルと感情状態を定量的に評価することを目的とした。この実験は、ドローン操作におけるストレス反応の評価、さらにはドローンの教育プログラム改善への応用を想定しておこなわれた。

### 方法

#### 実験日時・場所および状況

本実験は、2023 年 8 月下旬から 10 月上旬にかけて大学構内にて行われた。大学構内で 33×23m あった広さのうち、半分程度を使用した。実験環境の見取り図を Figure2-1 に示した。実験者と参加者に机と椅子を用意し、実験実施が可能な状態とした。参加者の体温調節のために、実験に影響を及ぼさないように、離れた場所から送風を行った。遮音は施さなかった。

#### 実験参加者

実験参加者は文京学院大学に所属する男子大学生 7 名で、平均年齢は 21 ( $SD=1.2$ ) 歳であった。独自項目から得られた回答によると、実験実施以前からドローンの経験があった人は 1 名であった。ドローンに興味があった人は 6 名であった。



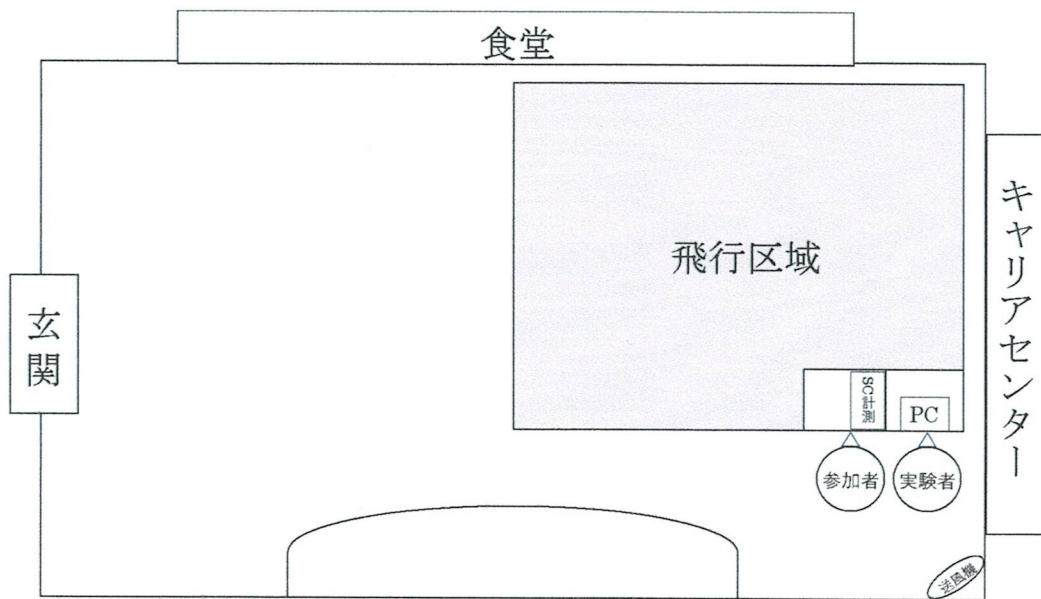


Figure2-1 実験が行われた大学構内での実験者，参加者および測定機器の様子

### 実験材料および機器

本実験では，Microsoft で使用されている風景画像とドローン（DJI 社製 MINI3PRO）を用いた。Figure2-2 に，風景画像設置の様子を示した。



Figure2-2 風景画像が設置された実験環境の様子

SC 読み取り用のソフトウェア Arduino プログラムのためにラップトップ PC (mouse 社製 F5-i5[Windows10] : 以下 PC) を用いた。同様に，計測中も用いた。SC 計測機器は，長野ら (2022) の SCC 計測機器を元に，研究 1 で太陽光充電が可能となった独自の SC 計測器を作成し，用いた。また，HR および心拍変動を読み取るために Polar 社製 H10 と ASUS 社製タブレットを用いた。Microsoft Forms (以下，Forms) で作成した質問紙に回答してもらうため，実験参加者にスマートフォンも用意してもらった。

### 心理指標

主観的な感情状態を測定するために，心理指標として一般感情尺度 (小川ら，2000) を使

用し、肯定的感情（以下 PA）8 項目、否定的感情（以下 NA）8 項目、安静状態（以下 CA）8 項目の計 24 項目に対し、「まったく感じていない」から「非常に感じている」の 4 件法で回答してもらった。1 つの課題に対し課題前と課題後の 2 回、回答を求めた。課題後では、課題中の感情を想起して回答するように教示した。

また、ドローンの経験などに関する質問と内省報告も求めた。

### 生理指標

Polar 社の使用手順に従って H10 を胸部に装着し、心電図の計測と記録を行った。得られた心電図から Root Mean Square Of Successive Differences(以下 RMSSD)と High Frequency Power(Normalized Units)（以下 HF Power(n.u.)) を算出し、取り扱った。また、右手の拇指球および小指球に電極（日本光電工業株式会社製ディスプレイ電極 F Vitroad）を装着し、指ぬき手袋で剥がれないようにしつつ SC を 1000ms 間隔で計測、記録した。

### 実験スケジュール

本実験のスケジュールを Figure2-3 に示した。本実験では、前安静 5 分、課題 5 分、後安静 5 分とした。条件の順序をカウンターバランスし、同スケジュールを繰り返した。その際、休憩 5 分間をはさんだ。前安静から課題期では、ドローンの離陸のみで問題は無かった。しかし、課題期から後安静にかけてはドローンの移動と着陸のために時間が統制できなかった。そのため、課題終了後 1 分間の時間を設け、その後 5 分間を後安静とした。

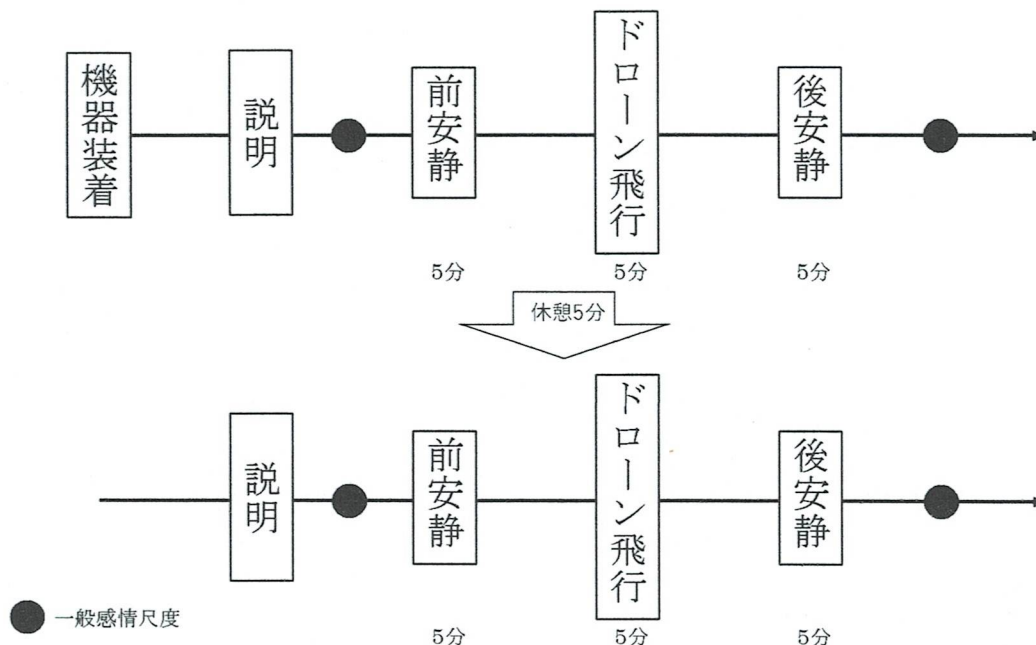


Figure2-3 実験スケジュール

### 手続き

インフォームドコンセントを取り、計測に必要な機器を装着してもらったあと、実験の説

明を行った。機器の装着では、風景画像を撮影するのに左手を多用するため、利き手・非利き手に関わらず右手にSC用電極を装着した。加えて、安静期から課題期、課題期から安静期において最小限の動作で済むように、両手の間にドローンコントローラーを配置した。

実験の説明では、実験全体を通しての姿勢と安静方法、ドローンの操縦方法と課題終了後の取り扱い、課題内容について説明した。順序効果を考慮し、カウンターバランスをとった。実験参加者は、直前までどのような課題を行うか知らない状況であった。緊張度をあげるため、撮影条件では「可能な限り、多くの風景画像を撮影してください。しかし、このドローン機体は13万円しますので、撮影のために壁に近づきすぎないように気をつけてください」と教示した。自由条件では「自由にドローンを操縦し、飛行してください」と教示した。自由条件では、値段について言及しなかった。

実験時の姿勢については、胸部に心電図計測器を装着してもらっていた。そのため、椅子に深く腰をかけ、可能な限りテーブルに近づき、肘をつけてもらった。また、無理のない範囲で足の位置も指定した。

実験の説明終了後、記録用PCに表示したQRコードからFormsの質問紙に回答するよう協力してもらった。

前安静の前には「今から実験を始めます。安静中は、ドローンのコントローラー中央にある赤点を注視するようにしてください。それでは、安静にしてください。」と教示した(Figure2-4)。安静終了後「それでは、コントローラーを手に取り、飛行を開始してください。」と教示した。課題終了後には「機体を近づけ、着陸させてください。」と言い、着陸を確認したのち「それでは、ドローンのコントローラー中央にある赤点を注視して、安静にしてください。」と教示した。



Figure2-4 ドローンコントローラーの外観

## 分析

HR, RMSSD および HF Power(n.u.)の計測データは、一度 kubios standard にて分析を行った。その後、HR, RMSSD および HF Power(n.u.)のそれぞれを従属変数として、2(飛行条件:自由,撮影)×3(期間:前安静,課題,後安静)の2要因参加者内計画の分散分析を行った。

SC では、Processing によって出力されたデータを整えた。データを整えたあと、SC を従属変数として 2(飛行条件:自由,撮影)×3(期間:前安静,課題,後安静)の2要因参加者内計画の分散分析を行った。1人の計測データに不備があったため、6人のデータを用いて分析を行った。

一般感情では、web フォームから回答してもらった結果をもとにデータ化した。データを整え、PA・NA・CAのそれぞれを従属変数として2(飛行条件:自由,撮影)×2(期間:課題前,課題後)の2要因参加者内計画の分散分析を行った。

## 結果

### HR

Figure2-5 に、全参加者の各期間における HR の平均値を示した。図から、両条件ともに課題期において心拍数が低下していた。特に、撮影条件では心拍数が大きく低下していた。課題量によって、HR に差があるかを検討した。各期間における全参加者の HR を従属変数として、2(飛行条件:自由,撮影)×3(期間:前安静,課題,後安静)の2要因参加者内計画の分散分析を行った。その結果、すべての要因の効果は有意ではなかった(飛行条件: $F(1,6)=1.72$ , *n.s.*; 期間: $F(2,12)=1.13$ , *n.s.*; 飛行条件×期間: $F(2,12)=1.72$ , *n.s.*)。つまり、HR は条件や期間に応じて変化しているとは言えなかった。

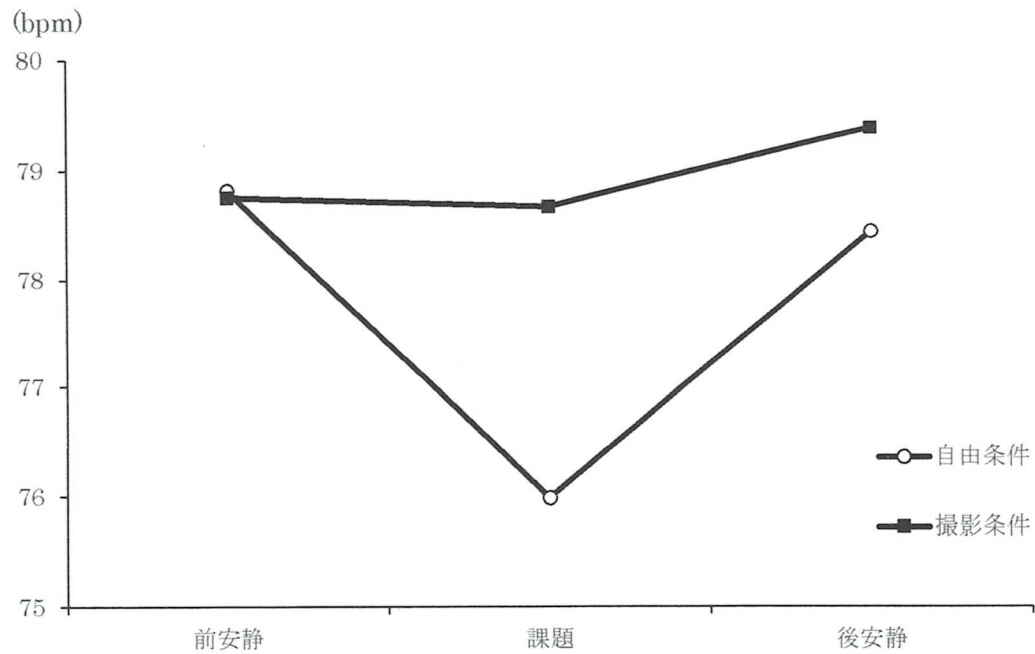


Figure2-5 各期間における HR の平均値

#### RMSSD

Figure2-6 に、全参加者の各期間における RMSSD の平均値を示した。図から、RMSSD の変化が違っていることが見て取れた。自由条件では、前安静時に低く、課題期に大きく上昇し、後安静時に再度低下していた。撮影条件では、全期間を通して高いままであったが、課題期に低下が見られた。課題量によって、RMSSD に差があるかを検討した。各期間における全参加者の RMSSD を従属変数として、同様に分散分析を行った。その結果、すべての要因の効果は有意ではなかった(飛行条件: $F(1,6)=0.89$ , *n.s.*; 期間: $F(2,12)=0.16$ , *n.s.*; 飛行条件×期間: $F(2,12)=0.99$ , *n.s.*)。つまり、RMSSD は条件や期間に応じて変化しているとは言えなかった。

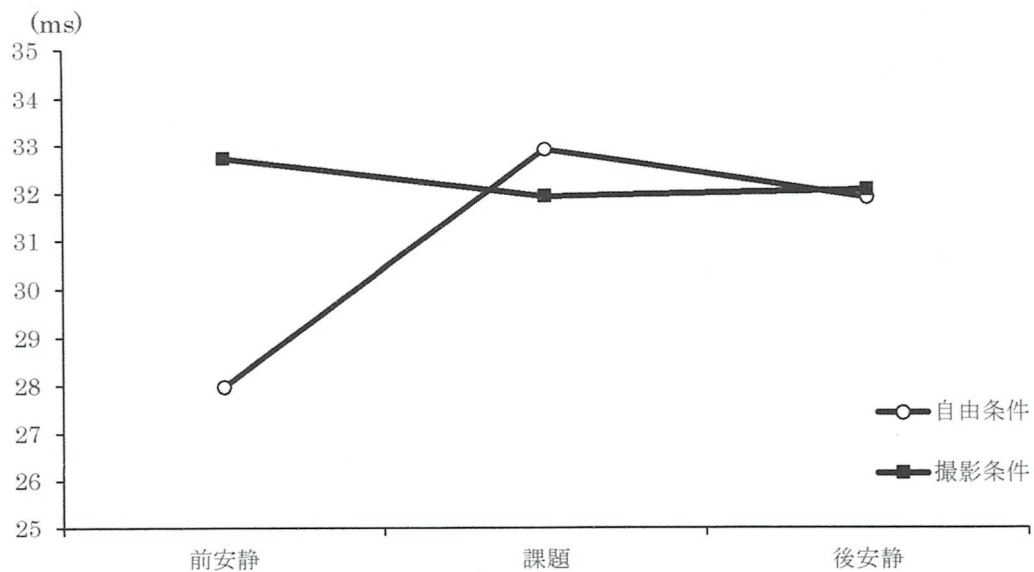


Figure2-6 各期間における RMSSD の平均値

#### HF Power (n. u.)

Figure2-7に、全参加者の各期間における Power(n.u.)の平均値を示した。条件によって、HF Power(n.u.)が異なっていることが図から見て取れた。自由条件では課題期に低下しているのに対し、撮影条件では課題期に上昇していた。課題量によって、HF Power(n.u.)に差があるかを検討した。各期間における全参加者の HF Power(n.u.)を従属変数として、同様に分散分析を行った。その結果、すべての要因の効果は有意ではなかった(飛行条件: $F(1,6)=1.12, n.s.$ ; 期間: $F(2,12)=0.06, n.s.$ ; 飛行条件×期間: $F(2,12)=0.29, n.s.$ )。つまり、HF Power(n.u.)は条件や期間に応じて変化しているとは言えなかった。

HF Power(n.u.)

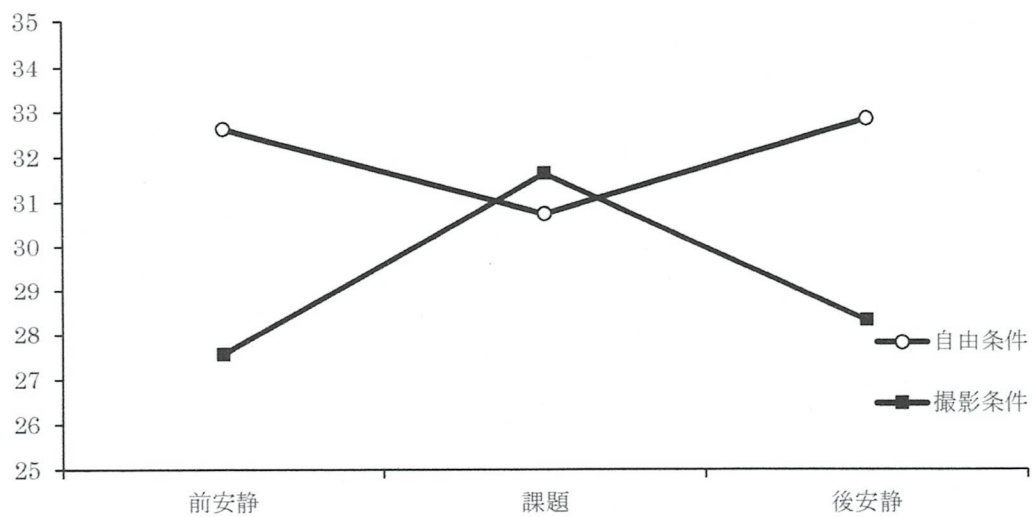


Figure2-7 各期間における HF Power(n.u.)の平均値

### SC

Figure2-8 に、全参加者の各期間における SC の平均値を示した。図から、両条件ともに課題期において SC が上昇していた。全期間を通して、自由条件で高い平均値となっていた。課題量によって SC に差があるかを検討した。各期間における全参加者の SC を従属変数として分散分析を行った。その結果、期間の効果が有意であった( $F(2,10)=39.25, p<.01$ )。飛行条件の効果と、飛行条件×期間の交互作用は有意ではなかった(飛行条件: $F(1,6)=1.70, n.s.$ ; 飛行条件×期間: $F(2,10)=0.25, n.s.$ )。期間の効果が有意であったため、Holm 法による多重比較を行ったところ、前安静より課題が高く、後安静より課題が高かった( $p<.05$ )。したがって、SC の値は飛行条件によって影響を受けず、課題期において、前安静、後安静より高いことが示された。

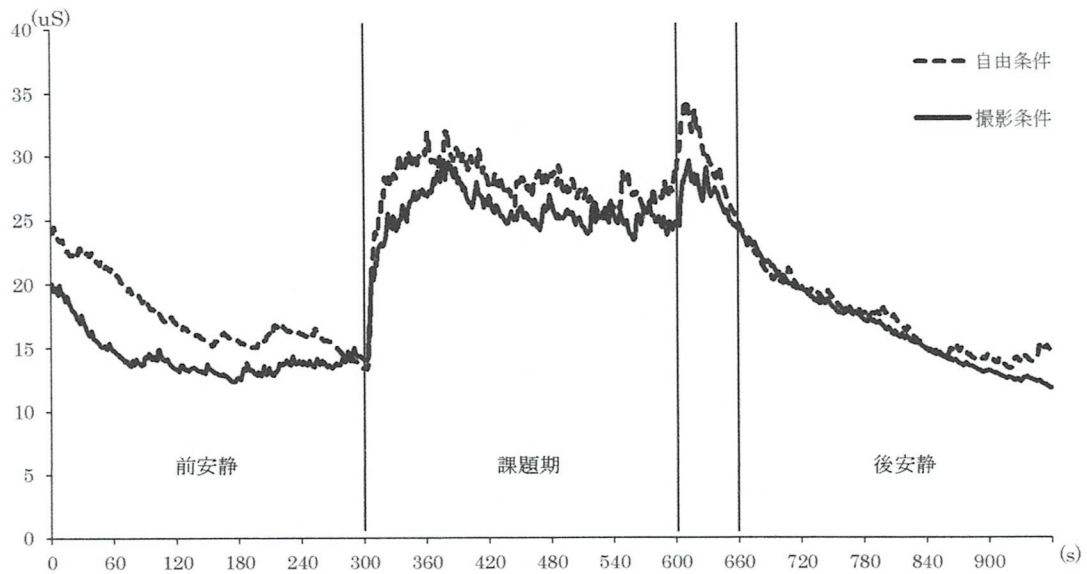


Figure2-8 全期間を通した SC の平均値

### 一般感情尺度

Figure2-9 に、全参加者の課題前後における PA の平均値を示した。図から、課題の前後で変化していた。自由条件では下がり、撮影条件では上がっていた。また、課題量によって主観的感情 (PA 得点, NA 得点, CA 得点) に差があるかを検討した。各期間における全参加者の PA 得点を従属変数として、同様に分散分析を行った結果、すべての要因の効果は有意ではなかった (飛行条件:  $F(1,6)=0.04$ , *n.s.*; 期間:  $F(1,6)=0.23$ , *n.s.*; 飛行条件 × 期間:  $F(1,6)=1.80$ , *n.s.*)。つまり、肯定的感情は条件や期間に応じて変化しているとは言えなかった。



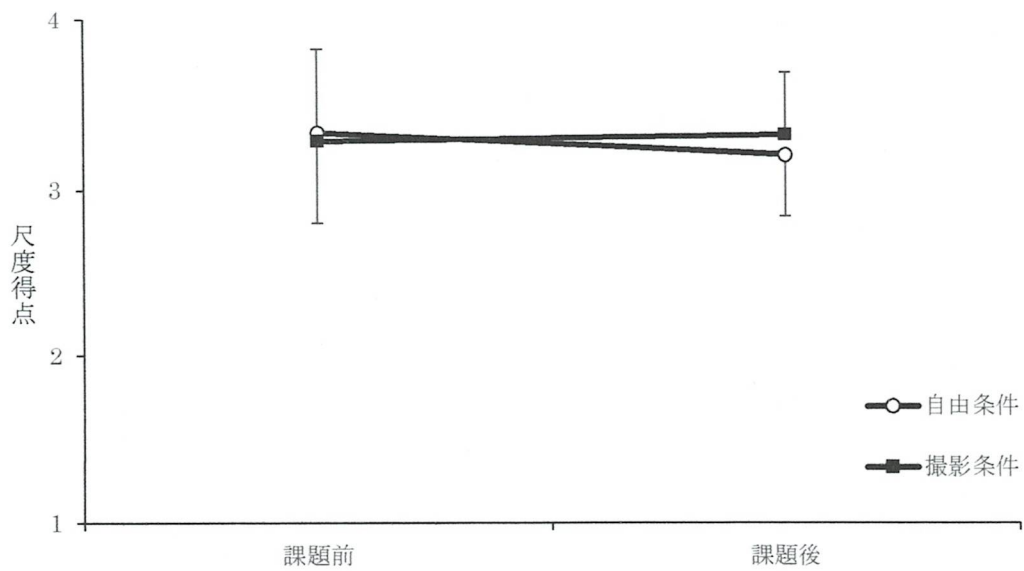


Figure2-9 課題前後における PA 得点の変化

Figure2-10 に、全参加者の課題前後における NA 得点の平均値を示した。撮影条件ではほとんど変化が見られないのに対し、自由条件では得点が上昇していた。各期間における全参加者の NA 得点を従属変数として、同様に分散分析を行った結果、すべての要因の効果は有意ではなかった(飛行条件: $F(1,6)=0.90, n.s.$ ; 期間: $F(1,6)=0.55, n.s.$ ; 飛行条件×期間: $F(1,6)=1.18, n.s.$ )。つまり、否定的感情は条件や期間に応じて変化しているとは言えなかった。

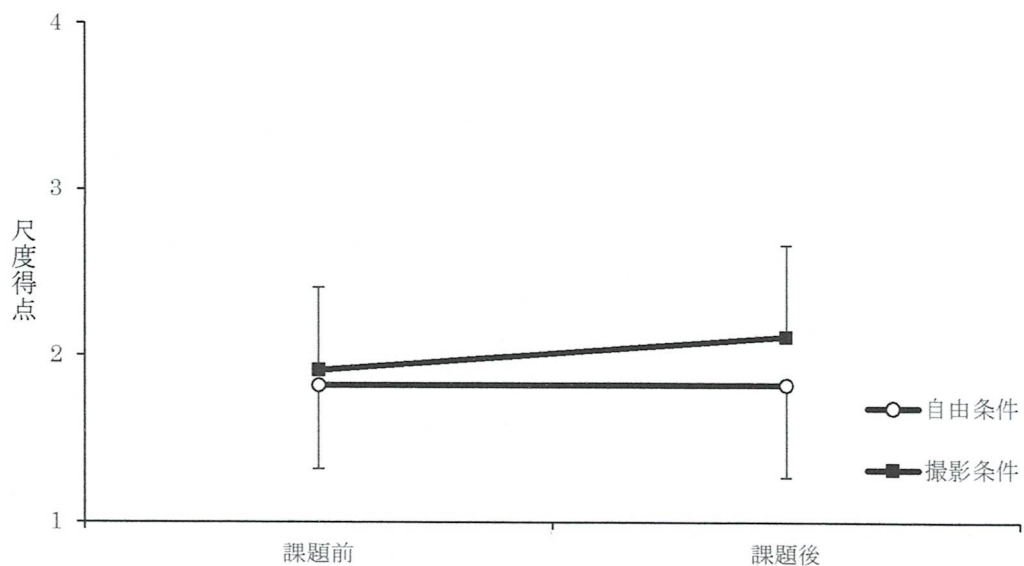


Figure2-10 課題前後における NA 得点の変化

Figure2-11 に、全参加者の課題前後における CA 得点の平均値を示した。両条件とも減少していたものの、撮影条件では大きく減少していた。各期間における全参加者の CA 得点を従属変数として、同様に分散分析を行った結果、すべての要因の効果は有意ではなかった(飛行条件: $F(1,6)=3.44, n.s.$ ; 期間: $F(1,6)=2.86, n.s.$ ; 飛行条件×期間: $F(1,6)=2.88, n.s.$ )。つまり、安静状態は条件や期間に応じて変化しているとは言えなかった。

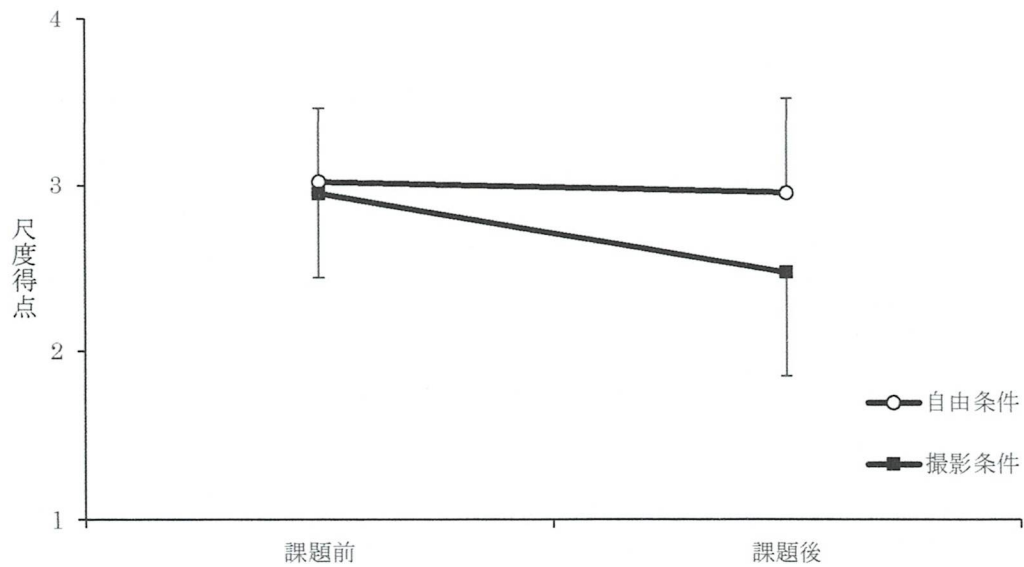


Figure2-11 課題前後における CA 得点の変化

### 内省報告

内省報告から得られた回答の一部を Table2-1 に示した。

Table2-1 実験参加者からの内省報告

参加者	報告内容
参加者2	安定感がすごいあって、操作前に感じていた不安は操作後には殆ど無くなった。軽く自由飛行中にも、撮影を行った。
参加者3	画面と機体をどちらも見て操作するのが難しく、写真をカメラに映す課題ではどっちに動かせばいいのかわからないことがあったから。
参加者4	安静期間中に車のクラクションが聞こえて少し驚いた。
参加者5	後安静中にハエが手の甲に止まり気になってしまいました。途中で駆除したのですが実験に影響が出ていないか不安に思いました。また、ドローンの操作は少しぎこちなかったと考えます。
参加者6	コントローラーの緑の点灯、点滅が安静時若干目障りだった
参加者7	操縦しやすく感じた 特に2回目の試行では慣れたせいかわかしく飛ばせた

## 考察

本研究の目的は、ドローン操作における操作者の心理的および生理的反応を、実際の環境と模擬された環境で比較分析することであった。

### ドローン飛行中の HR・RMSSD

結果から、考えられることを以下に記述する。統計的に有意ではないものの、どちらの条件でも HR が課題中に低下していた。HR の低下は副交感神経活動が優位になった可能性が示される。このことから、ドローン操作は副交感神経活動優位な状態を作り出しやすい可能性が考えられる。RMSSD は、副交感神経活動を表す指標である。統計的に有意ではないが、自由条件で上昇、撮影条件でやや下降傾向が認められた。これは本来、副交感神経活動有意な状態を導きやすいドローン飛行課題であるが、撮影という課題を付与することにより副交感神経活動が抑制される可能性を示唆している可能性がある。副交感神経活動が抑制されると、HR が下がりにくくなるので、HR の変動からもその可能性が高いといえる。類似した研究として鉄道運転シミュレーションではあるが、中川ら (2019) によると通常運転では HR の低下と心拍変動の上昇が確認された。停車操作という課題時には HR の上昇と心拍変動の低下が確認された。今回のドローン実験と近い結果であったといえるだろう。乗り物を細かく操作する際は、このような反応が一般的に生じやすい可能性がある。

### ドローン飛行中の SC 変化

SC については、課題中に上昇していた。これは、課題が交感神経活動の亢進を介して、発汗を生じさせた事を示している。交感神経優位の状態とも言えるが、これは HR や RMSSD の変化と一貫しない。つまり、心臓と汗腺の活動は、ドローン飛行中には乖離が起きる可能性がある。また、分散分析結果から飛行条件によって  $\epsilon$  q w な差は表れなかった。そのため、条件による違いは無いと思われた。

### ドローン飛行中の感情変化

PA 得点は、ほぼ横ばいだが、自由条件でやや下がるのに対し、撮影ではおおよそ平行で

あった。撮影により楽しさが少しだけ生じた可能性が考えられた。NA 得点では、自由条件はほとんど変わらなかったが、撮影条件では上昇した。課題があることによって「楽しくも緊張」というような、挑戦的な状況だったのではないかと考えられた。CA 得点では、どちらも下がるが撮影条件での下降が大きかった。やはり、撮影課題は自由飛行に比べ負荷が大きく、リラックス感情が低下していた可能性が考えられた。

### まとめと展望

以上のことから、ドローン飛行は本質的に HR を低下させるため、心臓に関しては副交感神経活動優位の課題といえる。一方で、交感神経活動を前提とした汗腺活動の増大も同時に起きた。したがって、心臓は副交感神経優位だが、汗腺は交感神経優位といった逆説的な状態を特徴とする課題といえる。このような課題として、鏡映描写課題が知られている。長野 (2004) の報告によると、単独での鏡映描写課題中の HR は低下していた。他方、Berntson ら (1996) の実験によると、錯視課題においては交感神経と副交感神経の活動が一致しなかったとされている。上記に挙げた鏡映描写課題や錯視課題は知覚運動学習もしくは感覚運動学習と言われている。特定の動作目標に向けて感覚と運動を適合させるという点において、ドローン飛行課題も感覚運動学習であり類似した生理的変化を生じた可能性が高い。

また、本実験参加者の多くが以前からドローンに興味はあったものの、未経験であった。ドローン操縦に慣れていなかったという側面もあり、このような結果となったと考える。順序効果を考慮したうえで実験を実施した。しかし、撮影条件から自由条件を行った一部参加者は自由飛行中でも撮影飛行のような動きが見られた。これが、真に自由な飛行かと問われれば難しいところである。同様に、撮影条件から行った場合、ドローンが高額であるということを知り、自由条件に影響を与えてしまった可能性は否めない。さらに、ドローン飛行のために実験室を離れて実験を実施した。しかし、広い空間を完全に統制することができなかった。そのため、不意な音などによって生体反応に影響があった可能性を否定できない。また、実験参加者の内省報告からも同様の意見が見受けられた。これは、実験環境に適した設備を借り、用意することができなかったことが挙げられる。本実験を参考に実験を行う場合は、事前の環境準備を強く勧めたい。

### 総合考察

本論文では、実験に合わせた測定機器の開発と、ドローン操縦時の生理・心理反応を解明することを目指した。さらに、ドローン操作の安全性向上とストレス管理に役立つ知見を提供することを目標とした。

屋外でも使用可能な SC 測定装置の開発により、屋外環境での心拍数や心電図の測定が可能となり、屋内でのドローン実験を通して、統制された環境外での実験に大きな足がかりを築くことができた。Bluetooth 通信のワイヤレス化による計測機器の改良に成功したが、計測の不安定さが課題として残った。今後は測定環境の統制や、計測器自体のさらなる改良が課題となるだろう。

研究 2 では、ドローン飛行実験から心拍数の低下や副交感神経活動の活動亢進、さらには汗腺活動の増大といった、交感神経と副交感神経の活動の乖離が観察された。これはドローン飛行が感覚運動学習に近い特性を持ち、心臓と汗腺の活動が異なる反応を示す可能性を示唆している。したがってドローン操縦時のストレスを、計算課題やスピーチ課題などの一般的なストレス課題と同様に捉えることは難しいだろう。この知見は、ドローン操縦者の育成や学習方法の改善に役立つ可能性がある。

技術の進歩により、多くの人がドローン飛行や撮影を楽しめるようになったが、この分野はまだ一部の人にとって敷居が高い可能性が示された。今後ドローンがより身近な存在になれば、本研究とは異なる結果が得られる可能性もある。近年は、配送業や農薬散布、従業員のストレス管理などの分野でのドローンの活用が盛んに報告されている。本研究の成果がこれらのドローン活用に寄与できることを願っている。

## 引用

- Berntson, G. G., Cacioppo, J. T., & Fieldstone, A. (1996). Illusions, arithmetic, and the bidirectional modulation of vagal control of the heart. *Biological psychology*, 44(1), 1-17.
- Kostenko, A., Rauffet, P., & Coppin, G. (2022). Supervised classification of operator functional state based on physiological data: Application to drones swarm piloting. *Frontiers in Psychology*, 12, 770000.
- 久保大輔. (2017). 無人航空機システム (ドローン) の歴史と技術発展. *計測と制御*, 56(1), 12-17.
- 本多麻子, 長野祐一郎 (2017) 温熱系. 堀忠雄, 尾崎久記 (監修), 坂田省吾, 山田富美雄 (編) 生理心理学と精神生理学 第 I 巻. 北大路書房, 京都, pp.207-222.
- 小川時洋, 門地里絵, 菊谷麻美, & 鈴木直人. (2000). 一般感情尺度の作成. *心理学研究*, 71(3), 241-246.
- 長野祐一郎. (2004). 競争型鏡映描写課題における心臓血管反応. *生理心理学と精神生理学*, 22(3), 237-246.
- 長野祐一郎. (2012). フィジカルコンピューティング機器を用いたストレス反応の測定. *ストレス科学研究*, 27, 80-87.
- 長野祐一郎, 櫻井優太, & 鈴木里砂. (2022). Arduino で生理指標を測る. *生理心理学と精神生理学*, 40(1), 114-124.
- 中川千鶴, & 渡部貴浩. (2019). 生理指標を活用した運転時の心身状態の推定・呼吸周期を基準とする HRV における副交感神経活動推定の検討. *人間工学*. *人間工学会第 60 回大会講演集*, pp. 1D1-2.
- 櫻井優太. (2017). 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. *愛知淑徳大学論集心理学部篇*. *愛知淑徳大学論集*. *心理学部篇*, 7, 27-38.

- 下野太海, 大須賀美恵子, & 寺下裕美. (1998). 心拍・呼吸・血圧を用いた緊張・単調作業ストレスの評価手法の検討. *人間工学*, *34*(3), 107-115.
- 海野遥香, & 橋本成仁. (2018). 生活道路走行時のドライバーの歩行者に対するストレス意識に関する研究. *都市計画論文集*, *53*(3), 1355-1361.
- Yu, D., Antonik, C. W., Webber, F., Watz, E., & Bennett, W. (2019). Multi-modal physiological sensing approach for distinguishing high workload events in remotely piloted aircraft simulation. *Human-Intelligent Systems Integration*, *1*, 89-99.

## 付録

### 使用されたプログラム内容

記録用 PC で使用されたプログラム内容を Figure3-1 に示した。1 行目から 32 行目では、主にライブラリのインポートと変数の初期化を行った。Processing のシリアル通信ライブラリをインポートした。シリアルポート、ポート名、ボーレート、データプロットのサイズ、データ配列、前回のデータ配列、色配列を定義した。データカウント、出力ファイル名、バッファ変数、書き込み用の PrintWriter オブジェクト、開始時刻を記録する変数を初期化した。34 行目から 46 行目では主に、setup 関数を使用し描画領域の設定とシリアルポートの初期化をした。SC 描画領域を 600x500 ピクセルで設定した。指定されたポート名とボーレートでシリアルポートを開くようにした。現在のタイムスタンプをファイル名に使用し、新しいファイルをデータ保存用に作成するようにした。48 行目から 104 行目では、draw 関数を使用してデータの読み取り、描画、保存をした。600 回の更新ごとに背景をクリアし、テキストを描画するようにした。シリアルポートからのデータを読み取り、それを加工してマイクロジーメンス (uS) 単位に変換した。SC 描画領域に読み取ったデータをプロットし、線を描画した。また、現在時刻と経過時間も描画した。5 回に 1 回の間隔でデータを csv ファイルに保存するようにした。

```

GetSC231207memo      subroutine1
import processing.serial.* // Processingのシリアル通信ライブラリをインポート
Serial myPort; // シリアル通信用の変数を宣言
String portName = "COM11"; // Bluetoothシリアルポートの名前
int baudRate = 115200; // ボーレート (通信速度)
// データプロットエリアの幅と高さ
int dataPlotWidth = 640;
int dataPlotHeight = 200;
// データの配列を初期化
float[] data = new float[4]; // 現在のデータを格納する配列
float[] dataNv = new float[4]; // 未使用のデータの可能性がある別の配列
float[] prevData = new float[4]; // 直前のフレームでのデータを格納する配列
// グラフの描画に使用する色を設定
color[] colors = {
  color(255, 0, 0), // 赤
  color(0, 0, 255), // 青
  color(0, 0, 0), // 黒
  color(0, 0, 0), // 黒
};
int datacount = 0; // データのカウント
String OutFileName; // 出力ファイル名
PrintWriter writer; // ファイル出力用の変数
int startMillis; // プログラム実行開始時刻
void setup() // プログラムの初期設定
{
  size(600, 500); // ウィンドウサイズの設定
  myPort = new Serial(this, portName, baudRate); // シリアルポートの初期化
  OutFileName = getTimestamp() + ".csv"; // 出力ファイル名の生成 (タイムスタンプを使用)
  writer = createWriter("data/" + OutFileName); // ファイルの作成
  buf2 = ""; // バッファを初期化
  startMillis = millis(); // プログラム実行開始時刻を取得
}
void draw() // プログラムの描画処理
{
  // 一定間隔で画面をクリアし、テキストを表示
  if ((datacount % 600) == 0) {
    background(255); // 画面をクリア
    fill(0, 0, 0); // テキストの色を黒に設定
    textSize(12); // テキストのサイズ
    text("50 uS", 10, 20); // テキストを表示
  }
  while (myPort.available() > 0) // シリアルポートからデータを受信
  {
    String inString = myPort.readStringUntil('\n'); // 改行までのデータを読み取り
    if (inString != null) // 読み取ったデータが null でない場合
    {
      inString = trim(inString); // 受信したデータを整形
      float dv = float(inString); // 文字列を浮動小数点数に変換
      float uS = map(dv, -4.5, 202, 1.03, 42.37); // データの範囲を変換して新しい単位で表現
      // グラフ描画
      stroke(colors[0]); // グラフの線の色を設定
      int xp = datacount % 600; // グラフの横軸上の位置を計算
      prevData[0] = data[0]; // 直前のフレームでのデータを記憶
      data[0] = map(uS, 0, 50, 0, 500); // 現在のデータを新しい単位に変換し、縦軸上の位置を計算
      line(xp - 1, 500 - prevData[0], xp, 500 - data[0]); // 直前のデータと現在のデータを線で結び、グラフを描画
      // データ保存
      String stamp = getTimestamp(2); // タイムスタンプ (時間情報) を取得
      String buf = stamp + "," + uS; // CSV形式のデータ行を作成
      println(buf); // コンソールにデータを表示
      buf2 = buf2 + buf + "\n"; // データをバッファに追加 (改行を含む)
      // 一定間隔でファイルに書き込み
      if (datacount % 5 == 0) { // 5フレームごとにファイルにデータを書き込む
        writer.print(buf2); // バッファになまったデータをファイルに書き込む
        writer.flush(); // ファイルに残っているデータを書き込み
        buf2 = ""; // バッファをクリア
      }
      // 現在の時間と経過時間を描画
      drawCurrentTime(); // 現在の時間を描画
      drawElapsedTime(); // 経過時間を描画
      datacount++; // データカウントをインクリメント
    }
  }
}
// プログラムの終了時に実行される処理
void stop() {
  // ファイルに残っているデータを書き込み
  writer.flush();
  // ファイルを閉じる
  writer.close();
  // 親クラスのstopメソッドを呼び出す
  super.stop();
}

```

Figure3-1 記録用 PC で使用されたプログラム



同様に、記録用 PC で使用されたサブルーチンプログラムを Figure3-2 に示した。1 行目～9 行目では、`getTimestamp` 関数を使用して現在のタイムスタンプを生成するようにした。現在の年、月、日、時、分、秒を取得。その後、`nf()`関数を使用して各時間要素に必要な場合には先行ゼロを追加し、年月日と時分秒を"YYYYMMDD\_HHMMSS"形式で組み合わせた。10 行目から 28 行目では `getTimestamp2` 関数を使用して、現在時刻のタイムスタンプを生成するようにした。その際、現在の時、分、秒を取得した。`nf()`関数を使用して、時間要素に先行ゼロを追加し、"HH:MM:SS"形式でタイムスタンプを作成した。31 行目から 43 行目では `drawCurrentTime` 関数を用いて、現在時刻の描画するようにした。具体的には、`getTimestamp2` 関数を使用して現在時刻のタイムスタンプを取得し、描画領域に矩形を描き、その上に現在時刻を表示するようにした。46 行目～58 行目では `drawElapsedTime` 関数を使って、経過時間の描画した。`getElapsedTime` 関数を呼び出して経過時間を取得したのち、描画領域に矩形を描き、その上に経過時間を表示させた。61 行目から 76 行目では、`getElapsedTime` 関数でプログラム開始時からの経過ミリ秒を計算をした。経過時間を時、分、秒に分解し、"HH:MM:SS"形式で文字列を生成するようにした。

```

GetSC231207memo  subroutine1
1 // タイムスタンプを生成するサブルーチン
2 String getTimestamp() {
3 // 年、月、日、時、分、秒を取得
4 int y = year();
5 int m = month();
6 int d = day();
7 int h = hour();
8 int min = minute();
9 int s = second();
10
11 // 各時間コンポーネントに必要なに応じて先頭にゼロを追加するために nf() を使用
12 String timestamp = nf(y, 4) + nf(m, 2) + nf(d, 2) + "_" + nf(h, 2) + nf(min, 2) + nf(s, 2);
13
14 return timestamp; // タイムスタンプを返す
15 }
16
17 // 現在の時間のフォーマットを生成するサブルーチン
18 String getTimestamp2() {
19 // 時、分、秒を取得
20 int h = hour();
21 int min = minute();
22 int s = second();
23
24 // 各時間コンポーネントに必要なに応じて先頭にゼロを追加するために nf() を使用
25 String timestamp = nf(h, 2) + ":" + nf(min, 2) + ":" + nf(s, 2);
26
27 return timestamp; // タイムスタンプを返す
28 }
29
30 // 現在の時間を描画するサブルーチン
31 void drawCurrentTime()
32 {
33 // 現在の時間のフォーマットを取得
34 String stamp2 = getTimestamp2();
35
36 // 背景の四角形を描画
37 fill(255): noStroke(); // 色を白に設定し、枠線なしで四角形を描画
38 rect(450, 5, 130, 50); // 四角形を描画 (位置: x=450, y=5, 幅: 130, 高さ: 50)
39
40 // テキストを描画
41 fill(0): stroke(0): textSize(24); // 塗りつぶしの色を黒、枠線の色を黒、テキストサイズを24に設定
42 text(stamp2, 450, 30); // テキストを描画 (内容: stamp2, 位置: x=450, y=30)
43 }
44
45 // 経過時間を描画するサブルーチン
46 void drawElapsedTime()
47 {
48 // 経過時間のフォーマットを取得
49 String elapsedTime = getElapsedTime();
50
51 // 背景の四角形を描画
52 fill(255): noStroke(); // 色を白に設定し、枠線なしで四角形を描画
53 rect(450, 55, 130, 50); // 四角形を描画 (位置: x=450, y=55, 幅: 130, 高さ: 50)
54
55 // テキストを描画
56 fill(0): stroke(0): textSize(24); // 塗りつぶしの色を黒、枠線の色を黒、テキストサイズを24に設定
57 text(elapsedTime, 450, 60); // テキストを描画 (内容: elapsedTime, 位置: x=450, y=60)
58 }
59
60 // 経過時間を取得するサブルーチン
61 String getElapsedTime() {
62 // プログラム実行開始からの経過時間をミリ秒で取得
63 int elapsedMillis = millis() - startMillis;
64
65 // 秒、分、時に変換
66 int s = elapsedMillis / 1000;
67 int m = s / 60;
68 int h = m / 60;
69
70 // 残りの秒数と分数を計算
71 s = s % 60;
72 m = m % 60;
73
74 // フォーマットを整えて返す
75 return nf(h, 2) + ":" + nf(m, 2) + ":" + nf(s, 2);
76 }
77
78
79
80
81

```

Figure3-2 記録用 PC で使用されたサブルーチンプログラム

実際に、SC 測定器でマイコンに書き込まれたプログラム内容を Figure3-3 に示した。1 行目から 4 行目では、使用するライブラリと変数の定義した。M5StickC, Bluetooth, Wire, Adafruit ADS1X15 のライブラリをインポートし、Bluetooth 通信、チップ ID、チップ名用の文字列、Adafruit ADS1015 センサーのインスタンスを定義した。6 行目から 34 行目では、`setup` 関数を使用してデバイスの初期設定を行った。具体的には、チップ ID を取得し、それを元に Bluetooth デバイス名を設定した。また、ATOM Lite の初期化、画面の設定、黒で画面をクリアし、Bluetooth デバイス名とバージョン情報を表示。Bluetooth 通信を開始し、ADS1015 センサーのゲインを設定、I2C 通信を開始し、センサーを初期化するようにした。36 行目から 54 行目では、`loop` 関数を使用して測定値の取得と送信を行った。ADS1015 センサーから差動入力値を読み取り、乗数 (0.125F) をかけた。読み取った値をシリアルポートと Bluetooth 経由で送信し、100 ミリ秒ごとに処理を繰り返した。

```

SCBTa02memo
1 #include <M5Atom.h> // M5Atomライブラリ
2 #include <BluetoothSerial.h> // BluetoothSerialライブラリ
3 #include <Wire.h> // Wireライブラリ
4 #include <Adafruit_ADS1X15.h> // Adafruit_ADS1X15ライブラリ
5
6 BluetoothSerial SerialBT; // BluetoothSerialオブジェクトの宣言
7 uint64_t chipid; // ESP32のチップID
8 char chipname[256]; // チップの名前を格納する配列
9 Adafruit_ADS1015 ads; // ADS1X15オブジェクトの宣言 (12-bit版)
10 unsigned long previousMillis = 0; // 最後に値が更新された時間を保存する変数
11 const long interval = 1000; // 更新間隔 (ミリ秒)
12
13 // M5AtomのLEDに色を設定する関数
14 CRGB dispColor(uint8_t r, uint8_t g, uint8_t b) {
15 // RGB値からCRGB構造体を生成する関数
16 return (CRGB)((r << 16) | (g << 8) | b);
17 // 24ビットのRGB値を組み立ててCRGB構造体として返す
18 }
19
20 void setup() { // Arduinoの初期設定を行う関数
21 chipid = ESP.getEfuseMac(); // ESP32のチップIDを取得
22 sprintf(chipname, "M5Atom_%04X", (uint16_t)(chipid >> 32));
23 // チップの名前を生成
24 M5.begin(true, true, true); // M5Atomの初期化
25 SerialBT.begin(chipname); // BluetoothSerialの初期化
26 Serial.println("begin"); // シリアルモニタに"begin"と表示
27
28 ads.setGain(GAIN_SIXTEEN); // ADS1X15のゲインを設定
29 Wire.begin(26, 32); // I2C通信の開始
30 ads.begin(); // ADS1X15の初期化
31 Serial.println("Wirebegin"); // シリアルモニタに"Wirebegin"と表示
32
33 M5.dis.drawpix(0, dispColor(100, 100, 100)); // M5AtomのLEDを点灯
34 }
35
36 void loop() { // Arduinoのメインループ処理
37 unsigned long currentMillis = millis(); // 現在の時間をミリ秒単位で取得
38
39 if (currentMillis - previousMillis >= interval) {
40 // 最後の更新から1秒以上経過した場合
41 previousMillis = currentMillis; // 現在の時間を保存
42
43 // 変数の宣言: 16ビット整数型のresults、浮動小数点数型のmultiplier
44 int16_t results; // 16ビット整数型のresults
45 float multiplier = 0.125F; // 浮動小数点数型のmultiplier
46 results = ads.readADC_Differential_0_1();
47 // ADC (アナログ・デジタル・コンバータ) からデータを読み取り、resultsに格納
48
49 Serial.println(results * multiplier); // シリアルモニタに結果を出力
50 SerialBT.println(results * multiplier); // Bluetoothシリアルに結果を出力
51 Serial.print("200,"); // 固定のデータをシリアルに出力
52 Serial.print("-7,"); // 固定のデータをシリアルに出力
53 }
54 }

```

Figure3-3 ATOM Lite に書き込まれたプログラム