

2018 年度(平成 30 年度)

創造工学セミナーⅡ

Final Report

新たな観光望遠鏡システムの開発

指導教授 金丸 隆志 准教授

チームメンバー

S5-15003 青木 俊輔

S5-15013 岩手山 彰也

S5-15015 大塚 裕斗

S5-15020 金井 聖

S5-15034 鈴木 智也

目次

論文要旨	(青木担当)	3
第1章 緒言	(大塚担当)	4
1.1	研究背景	4
1.2	研究目的	5
1.3	先行研究で開発された観光望遠鏡の動作原理	6
1.4	先行研究で開発された観光望遠鏡の構造	7
1.5	先行研究で開発されたアプリケーション	8
1.6	研究の方向性	9
1.7	本論文の構成	10
第2章 カメラの回転機構及びその構成部品	(鈴木担当)	11
2.1	使用したレンズモジュール	12
2.2	ステッピングモーターの構造と動作原理	13
2.3	使用するステッピングモーターの性能	14
2.4	ステッピングモーターの励磁方式	15
2.5	マイクロステップ駆動	17
2.6	回路	18
2.6.1	Arduino UNO	18
2.6.2	使用するモータードライバー	19
2.6.3	SPI 通信	20
2.6.4	デジタイゼーション接続	21
2.6.5	回路の製作	22
2.7	Arduino UNO プログラム	23
第3章 機体	(岩手山担当)	24
3.1	先行研究の機体	24
3.2	改善図面	25
3.3	使用素材	26
3.4	使用材料	27
3.5	加工方法	28
3.5.1	右柱	28
3.5.2	左柱	30
3.5.3	カメラボックス	32
3.5.4	上台	34
3.5.5	支え柱	35

3.5.6 中台.....	36
3.5.7 下台.....	38
3.6 中台の基板配置.....	39
3.7 カメラの上下の動作.....	40
3.8 カメラの左右の動作.....	41
3.9 新たに完成した機体.....	42
第4章 オートフォーカス (金井担当).....	43
4.1 コントラストに基づくオートフォーカス.....	44
4.1.1 コントラストの計算.....	44
4.1.2 コントラスト実験.....	45
4.2 コントラストに基づくオートフォーカスのプログラム作成.....	46
4.2.1 フォーカス状態の判断.....	46
4.2.2 回転速度の制御.....	47
第5章 結論 (青木担当).....	50
謝辞 (青木担当).....	52
参考文献.....	53
図表目録.....	55

論文要旨 (青木担当)

観光地にあるコイン式観光望遠鏡は景観を同行者と共有できない問題から減少傾向にある。しかし、訪日外国人観光客が増加している傾向から潜在的な需要は大きいと考えられる。そこで、タブレット端末を組み合わせることで、より革新的な観光望遠鏡を開発できると考えた。本研究では、先行研究で作成された観光望遠鏡の問題点を改善し、より使いやすい観光望遠鏡の開発を目指した。オートフォーカス機能の付与、新たな観光望遠鏡の機体の製作を行った。

第 1 章 緒言 (大塚担当)

1.1 研究背景

コイン式の観光望遠鏡は、昔から景観の良い観光地に必ずと言って良いほど設置されてきた。遠くの景色を拡大してみることができ、日常では得られない体験ができる。

現在も、成田空港の展望デッキや名古屋城などに設置されている。しかし、以前に比べ観光望遠鏡の設置台数は減少傾向にある。実際に、東京タワーでは、展望台スペースをより広く快適に利用するため、大展望台と特別展望台に設置されていた 37 台の有料望遠鏡を撤去し、双眼鏡のレンタルを開始した。また、2012 年 5 月に開業した東京スカイツリーには観光望遠鏡が設置されていない。

私達は、この原因として「共感」できないことが問題と考えた。近年 SNS 上でいわゆる「インスタ映え」を狙って、「いいね」で共感を得たいという行動欲求が広がりつつある。それに対し、従来の観光望遠鏡は同時に 1 人しか使用できず、同行者とも体験を共有することができない。すなわち、従来の観光望遠鏡は、現代の観光客の要求を満たしていないと言える。

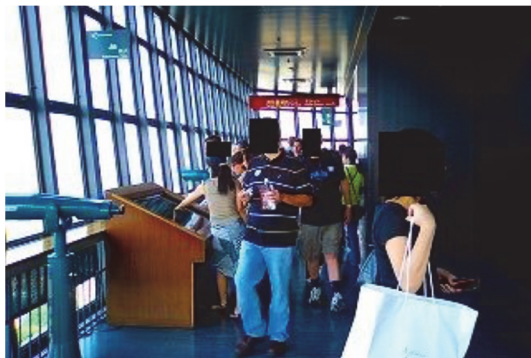


図 1.1 左)観光望遠鏡が撤去される前の東京タワー展望台
右)観光望遠鏡が撤去された後の東京タワー展望台 (1)

1.2 研究目的

本研究は、減少傾向にある観光望遠鏡に新たな価値を見出すことで観光客のニーズに応えることを目指す。

図 1.2 のグラフから分かるように日本に観光に来る外国人の数は著しく増加している。2020 年に開催される東京オリンピックや、政府が推進する「明日の日本を支える観光ビジョン」などにより数多くの観光地で外国人を見かけることが多くなっている。図 1.2 のグラフから分かるように日本に観光に来る外国人の数は著しく増加している。

外国人観光客増加に伴い、東海旅客鉄道（JR 東海）は東海道新幹線の駅構内においてヤマハが開発したスマートフォン用アプリケーション「おもてなしガイド」を活用した多言語対応サービスの開発を進めている。また、宿泊施設や空港、飲食店などには訪日外国人向けの無料 Wi-Fi が設置されている。こうしたスマートフォン等を利用し、より快適に旅行を楽しんでもらうためのサービスが増えている。

以上のことから、観光望遠鏡とスマートフォンやタブレット端末を組み合わせた新たな観光望遠鏡の需要があると考え、開発に取り組むことにした。

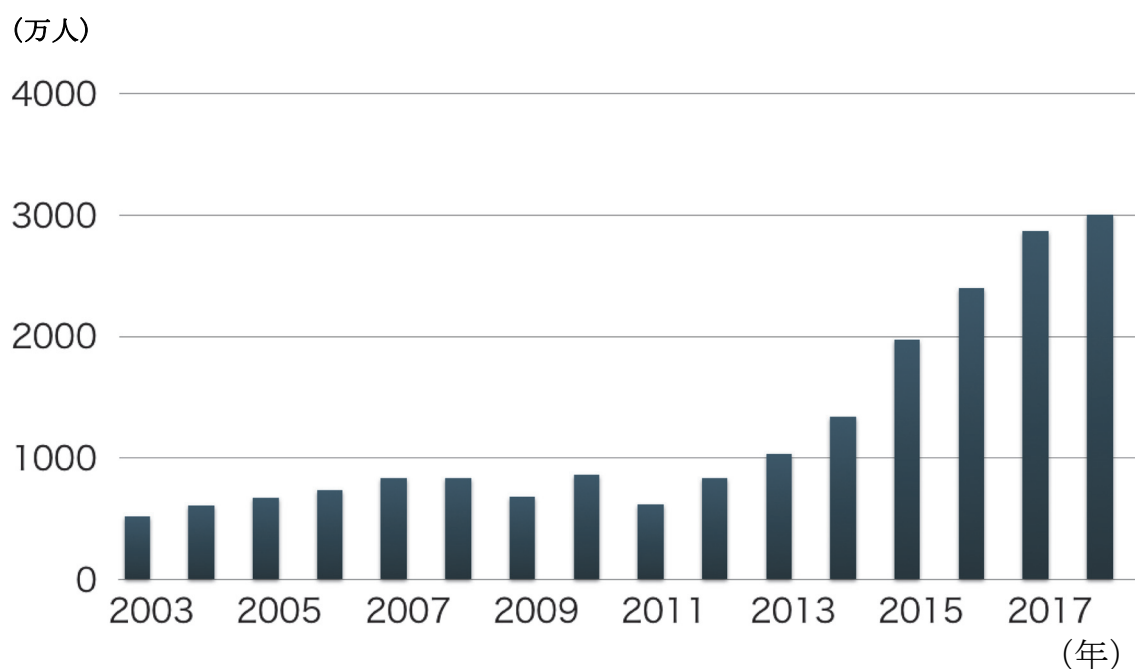


図 1.2 訪日外国人数 観光庁統計データ (2)

1.3 先行研究で開発された観光望遠鏡の動作原理

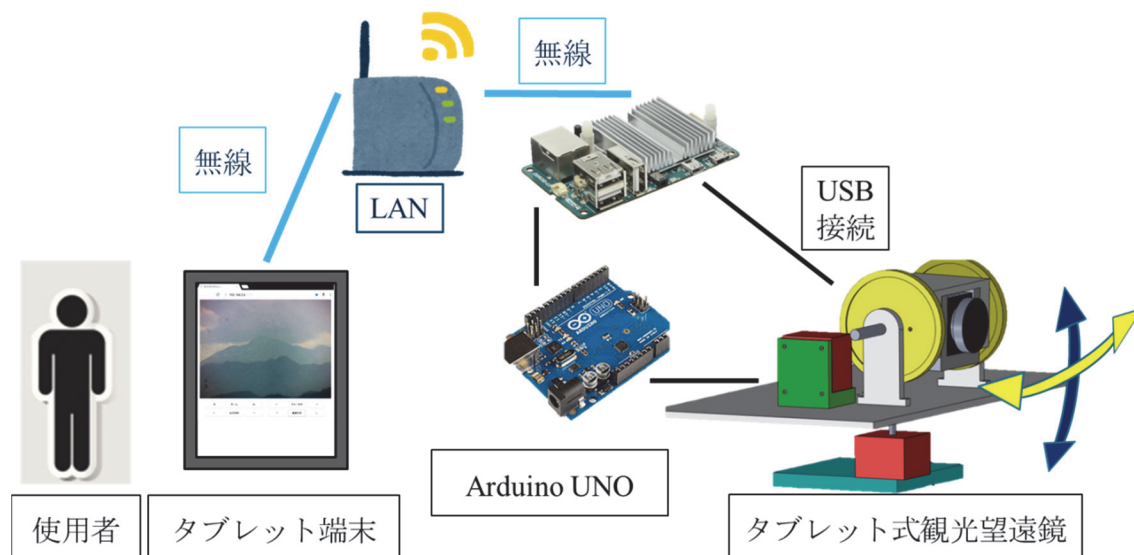


図 1.3 タブレット式観光望遠鏡の動作原理 (3)

先行研究で開発された観光望遠鏡の動作原理を図 1.3 に示す。使用者はタブレットを用いて、上下左右移動、ズームイン、ズームアウト、フォーカス調節のボタン押すことで操作を行う。その情報が無線 LAN を通して最終的に Arduino UNO に送られ、モーターの制御をする。映し出された景色も同様に、無線 LAN を通してタブレットに表示され、景色を楽しむことができる。これが先行研究で製作された観光望遠鏡の動作原理である。

1.4 先行研究で開発された観光望遠鏡の構造

先行研究で製作された観光望遠鏡の構造を図 1.4 に示す。大きさは 23cm×21cm×20cm であり、赤部分がモーター、黄部分がギアを示している。

先行研究の観光望遠鏡は主にアクリル材と 3D プリンタで作成されたパーツで構成されている。アクリル材は一番下の四角い板とその上の四角い板に、3D プリンタで作成されたパーツはカメラ周りのボックスに使用された。また、土台部分が鉄の棒とモーターのみで支えられ、カメラも厚さ 5mm の細い柱で支えられている。

この観光望遠鏡には 4 つのモーターが存在する。図 1.4 に示されている赤部分の上下左右に動作するモーターが 2 つと、レンズモジュール内部に存在するフォーカス調節とズームイン、ズームアウトを行うためのモーターが 2 つである。モーターは全てステッピングモーターで、駆動方法は全てフルステップ駆動である。

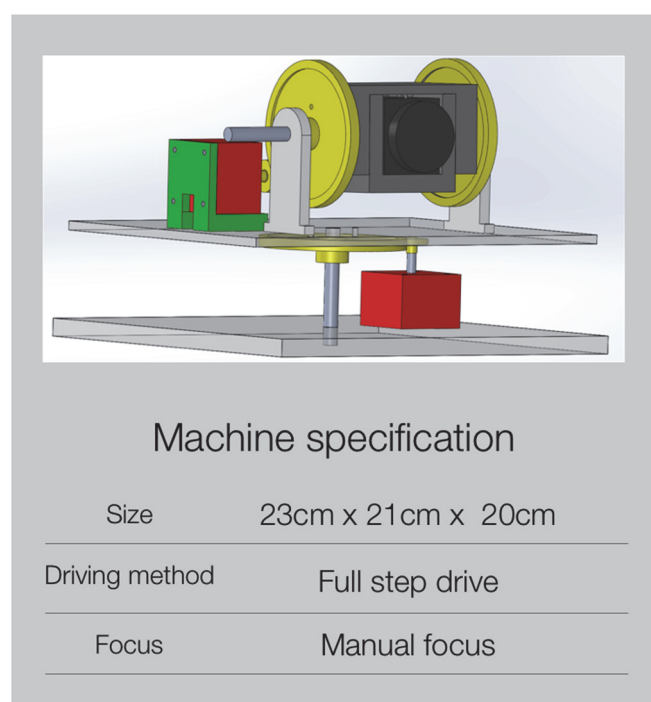


図 1.4 先行研究で開発された観光望遠鏡の CAD 全体図
(赤：モーター、黄：ギア、緑：固定具 を示している)

1.5 先行研究で開発されたアプリケーション

先行研究で開発されたアプリケーションの操作画面を図 1.5 に示す。上半分が観光望遠鏡からの映像、下半分が操作画面となっている。操作画面では、ズーム調整、フォーカス調整、水平方向移動、垂直方向移動用のモーターを操作でき、それぞれに正転、逆転、停止に対応する 3 つのボタンが存在する。回転したい方向のボタンを押すことでモーターが駆動し、それぞれの機能の停止のボタンを押すことでモーターが停止する。

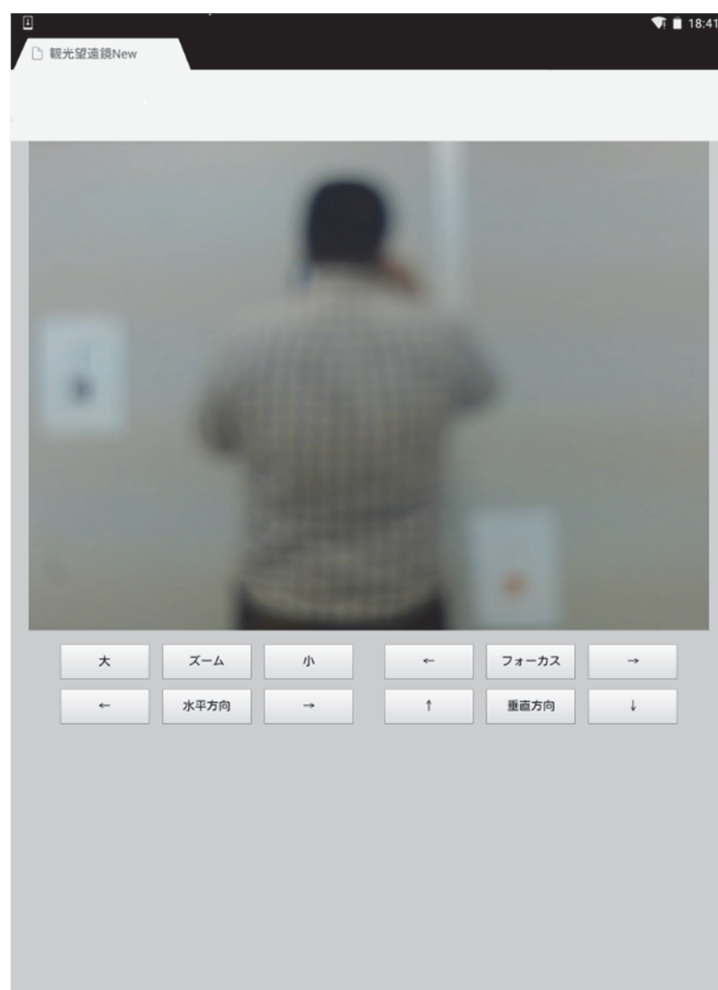


図 1.5 先行研究で開発されたアプリケーションの操作画面

1.6 研究の方向性

先行研究で開発された観光望遠鏡は以下のような問題点がある。

①上下左右に滑らかに動作しない

上下左右に移動させる際の動き方は、決して滑らかと言えない。観光望遠鏡は、数十 km～数百 km の景色を映し出す。そのため、僅かな振動でも、映像では大きな振動となり影響を及ぼす。

②機械的安定性に乏しい

土台部分が鉄の棒とモーターのみで支えられ、カメラも厚さ 5mm の細い柱で支えられているため、機械的安定性に乏しい。また、Arduino UNO やユニバーサル基板が上の台に配置されており、全体的に機構の重心が高いことも原因の一つである。

③フォーカスを合わせるのに時間がかかる

観光望遠鏡の操作は手動で行っている。そのなかでもピントを合わせる作業は、操作する人の感覚的な判断が必要となる。また、他の動作をした後、毎回調節するため時間を要し、利便性が高いとは言えない。

本研究ではこれらの問題点を踏まえ新たな観光望遠鏡システムの開発を目指し、以下のような目標を立てた。下記の点を改善することで安定性・利便性の高い観光望遠鏡を完成させることを目指す。

- 観光望遠鏡の動作の円滑化

動作する際の振動は、駆動しているモーターに原因があると考えられる。モーターの駆動を滑らかにすることで円滑化を図る。

- 機械的安定性向上

しっかりとした支えのある新しい機体を作成する。

- オートフォーカス機能の付与

オートフォーカスのプログラムを作成する。この機能により、感覚的な判断や、毎回の操作が一つ減るため、操作性向上に繋がると考えた。

- コンパクト化

この目標は上記の問題点の解消ではないが、利便性を追加するために追加した。

1.7 本論文の構成

本論文の構成は以下の通りである。第 2 章ではカメラの回転機構及びその構成部品についてまとめる。第 3 章では新しく製作した機体について、第 4 章ではオートフォーカスの方法や実験結果などを述べる。第 5 章は私達が掲げていた目標と先行研究でからの改善点を修正出来ているかを考察する。

第2章 カメラの回転機構及びその構成部品

(鈴木担当)

本研究で開発する観光望遠鏡の機体システムは図 2.1 のような構成をしている。本章では、レンズモジュール、モーター、回路、Arduino のプログラムについて解説する。

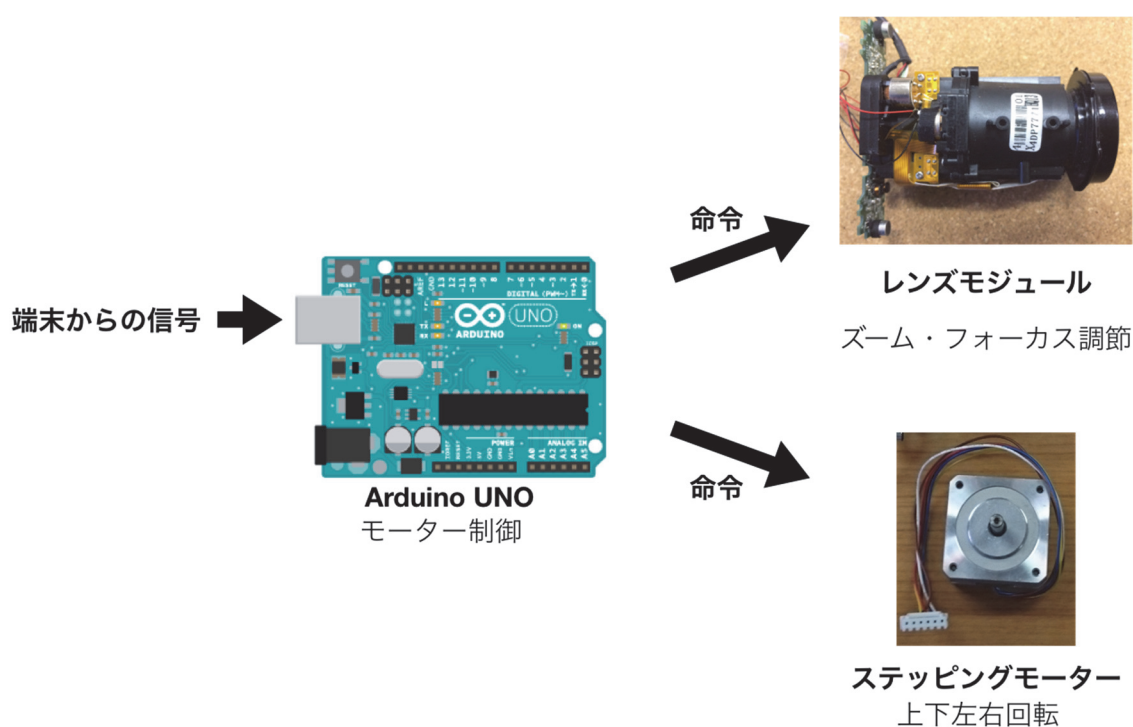


図 2.1 機体システムの概要

2.1 使用したレンズモジュール

まず、使用したレンズモジュールについて説明する。図 2.2 は K 社から提供されたレンズモジュールであり、フォーカスやズームを調節するためのステッピングモーターが取り付けられている。このレンズモジュールの映像を USB 接続で取得するため、本研究ではレンズモジュールに適した UVC 規格の画素センサーを取り付けている。図 2.3 はレンズモジュールに取り付ける Logicool 社製 C920t 内蔵の 1/4 インチ画素センサーである。

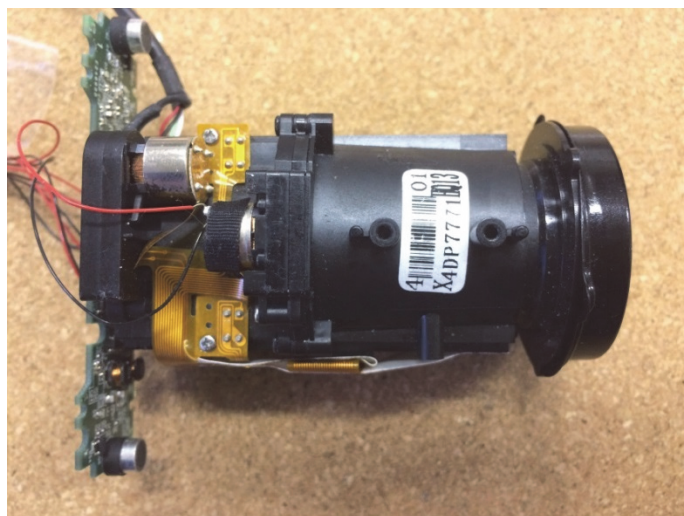


図 2.2 使用するレンズモジュール

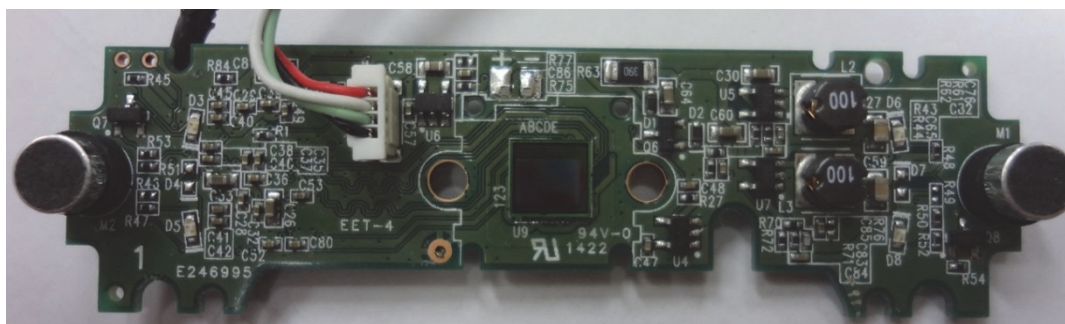


図 2.3 Logicool 社製 C920t 内蔵の 1/4 インチ画素センサー

2.2 ステッピングモーターの構造と動作原理

ステッピングモーターとは、パルス信号によって回転角度・回転速度を正確に制御できるモーターである。その特徴を表 2.1 にまとめる。また、図 2.4 は典型的なステッピングモーターの断面図を示したものである。本研究では、観光望遠鏡の上下左右の向きを操作するステッピングモーター2つとレンズモジュールに取り付けられているフォーカス・ズームを調節するステッピングモーター2つで、合計4つのステッピングモーターを用いる。

表 2.1 ステッピングモーターの特徴 (4)

特徴 1	回転角度と回転速度がパルス信号の回数と周期によって決まる 回転角度＝ステップ角×パルス数 回転速度[rpm]＝（ステップ角/360）×パルス速度[Hz]×60[s]
特徴 2	通電状態で大きな静止トルクを得ることができ、非通電状態でも小さな静止トルクが得られる
特徴 3	パルスを止めるとその位置で停止するため正確な回転角度制御に適している
特徴 4	機械的な接触子がなく長寿命である
特徴 5	基本ステップ角度が0.72° の5相ステッピングモーターと、基本ステップ角度が1.8° の2相ステッピングモーターが広く使われている。

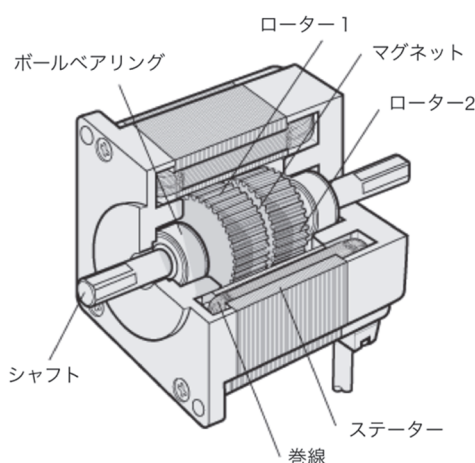


図 2.4 ステッピングモーターの断面図 (5)

2.3 使用するステッピングモーターの性能

本研究で観光望遠鏡の水平及び垂直方向の回転に2つ使用するステッピングモーターは、RS社製191-8299である。その性能を表2.2に、外見を図2.5に示す。

表2.2 使用するステッピングモーターの性能 (6)

名称	RS191-8299
ステップ角	1.8[°]
保持トルク	0.1[Nm]
配線数	6
定格電圧	12[Vdc]
フレームサイズ	40[mm]×40[mm]
軸径	5[mm]
許容電流値	400[mA]
巻線配置	ユニポーラ
シャフトの長さ	53[mm]
深さ	33[mm]
抵抗/フューズ	30[Ω]



図 2.5 RS 社 191-8299 (6)

この 2 つのほか、本研究ではレンズモジュールに内蔵されている 2 つを加え合計 4 つのステッピングモータを操作することで観光望遠鏡の操作を行う。

2.4 ステッピングモーターの励磁方式

各相のコイルへ決まった順番で電流を流す方式を励磁方式という。励磁方式には、1 相励磁、2 相励磁、1-2 相励磁の 3 種類が存在する。この励磁方式を利用用途に応じて使い分ける。

本研究で作成した観光望遠鏡の上下左右の向きを操作するステッピングモーターは、マイクロステップ駆動方式と呼ばれる 2 相励磁での電流パターンを電氣的に細分化した励磁方式を用いた。マイクロステップ駆動方式については、2.5 で説明する。また、レンズモジュールに取り付けられているフォーカス・ズームを調節するステッピングモーターも、2 相励磁方式を使用した。

● 1 相励磁

図 2.6 のように $A \rightarrow B \rightarrow A' \rightarrow B'$ の 4 つの端子に順に電流を流す方式が 1 相励磁である。励磁の順を $B' \rightarrow A' \rightarrow B \rightarrow A$ とするとモーターは逆回転する。1 相励磁は最も単純な方式であるが、トルクが弱く高速回転には向いていない。また、モーター回転中の減衰振動が大きく、脱調などの同期外れを起こしやすい。

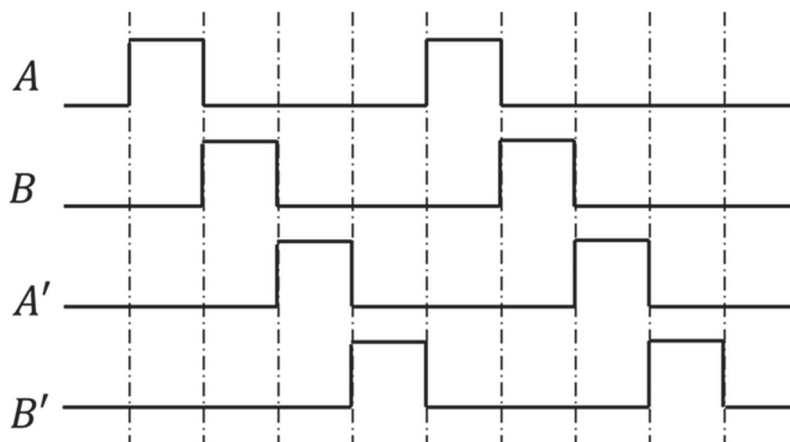


図 2.6 1 相励磁のパルス (7)

- 2 相励磁

図 2.7 のように 2 相ずつ励磁位相を進める方式である。パルス幅が 1 相励磁の 2 倍となり、1 相励磁に比べて回転が安定して、大きなトルクが得られるが消費電力も 2 倍になる。

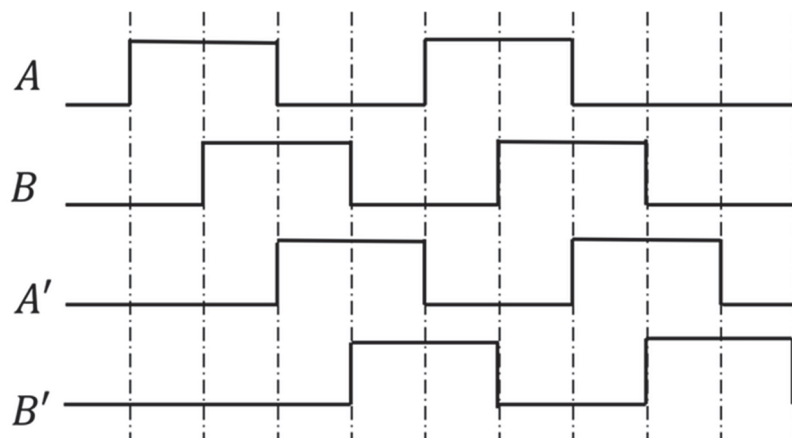


図 2.7 2 相励磁のパルス (7)

- 1-2 相励磁

1 相励磁と 2 相励磁を交互に繰り返す方式で、ハーフステップ駆動ともいう。モーターのステップ角分解能は、1 相や 2 相励磁方式の半分になるため、細かいモーター制御が可能になる。しかし、2 相励磁と比べてトルクが弱い。

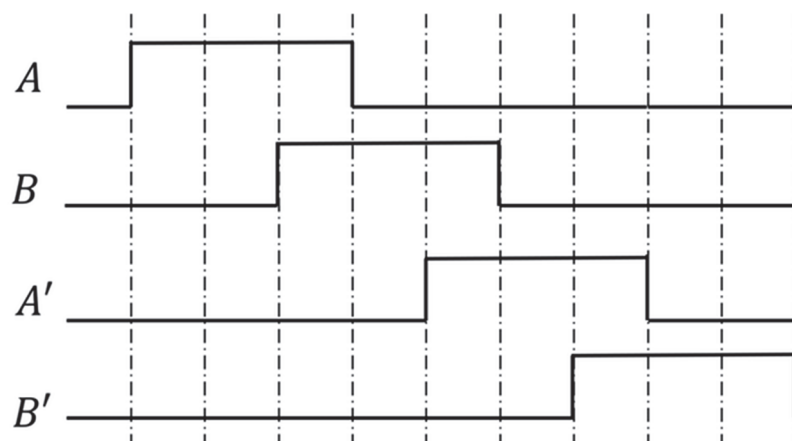


図2.8 1-2相励磁のパルス (7)

2.5 マイクロステップ駆動

本研究では、観光望遠鏡を滑らかに動かすためにマイクロステップ駆動方式を採用する。マイクロステップ駆動は、低速領域の低騒音化・低振動化を目的にステップ角を電氣的に細分化する技術である。

フルステップ駆動の場合、ローターは回転する際、比較的大きな慣性モーメントを有している。停止状態からステータの磁極を変え、1ステップ回転させる状態を考える。ローターは加速度的に角速度を増し、1ステップ回転した所で最高速度に達する。しかし、ステータの磁極はそのままなのでローターはそこで停止しようとするが、自身の慣性モーメントによりオーバーラン、アンダーランを繰り返し、減衰をして停止する（図2.9）。これが滑らかに滑らない原因である。

図2.10はマイクロステップ駆動とフルステップ駆動の電流パターンを比較したものである。マイクロステップは角相の電流を階段状に増加、同様に減少させて相間で回転子を吸引する力を徐々に変えることでステップ角を小さくすることができる。

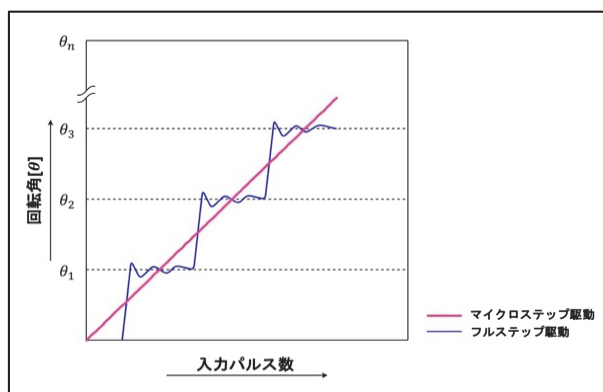


図2.9 マイクロステップ駆動とフルステップ駆動の電流パターンの比較 (8)(9)

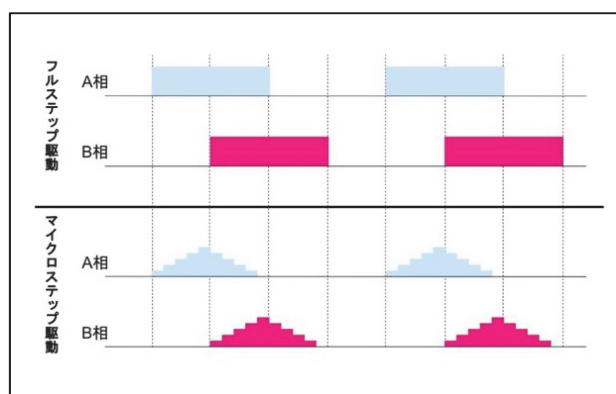


図2.10 マイクロステップ駆動とフルステップ駆動の比較 (8)(9)

2.6 回路

ここでは、観光望遠鏡のために作成した回路について述べる。最初に、ステッピングモーターを操作するために用いたマイコンボード Arduino UNO について解説する。

2.6.1 Arduino UNO

Arduino とは、AVR マイコン、入出力ポートを備えた基板、C 言語風の Arduino 言語による統合開発環境から構成されるシステムのことであり、初心者でも簡単に扱えるマイコンボードである。下記に外観(図 2.11)、仕様(表 2.3)を示す。

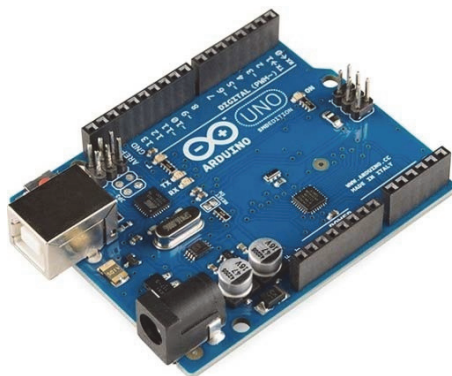


図 2.11 Arduino UNO (10)

表 2.3 Arduino UNO (10)

マイコン	ATmega328
動作電圧	5V
入力電源電圧(推奨範囲)	7～12V
デジタル I/O ピン	14 本(うち 6 本は PWM として使用可能)
アナログ入力ピン	6 本
I/O ピンの最大 DC 電流	40mA
3.3V ピンの最大 DC 電流	50mA
フラッシュメモリー	32KB(ATmega328)
SRAM	2KB(ATmega328)
EEPROM	1KB(ATmega328)
クロック周波数	16MHz

2.6.2 使用するモータードライバー

観光望遠鏡に使用するステッピングモーターは、観光望遠鏡の向きを操作する 2 つとレンズモジュールに取り付けられているフォーカス・ズームを調節する 2 つで合計 4 つである。フォーカス・ズーム調整用のステッピングモーターは、図 2.12 左に示す低電圧用のモータードライバー TI 社 DRV8834 を用いて制御する。

先行研究では、観光望遠鏡の向きを操作する 2 つのステッピングモーター用のモータードライバーとして TOSHIBA 社 TB6674PG を使用していたが、2.5 で述べたマイクロステップ駆動方式を導入するために、秋月通商製 L6470 ステッピングモータードライバーキットへ変更した。

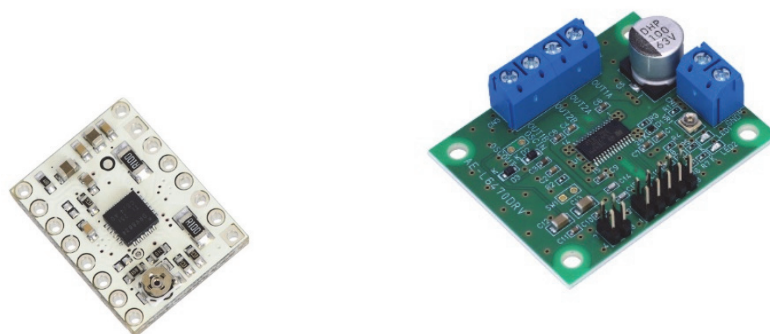


図 2.12 TI 社 DRV8834 と秋月通商製 L6470 (11)(12)

表 2.4 各モータードライバーの仕様 (11)(12)

名称	DRV8834	L6470	単位
モーター電圧(min)	2.5	8	V
モーター電圧(max)	10.8	45	V
モーター電流(ave)	1.5	3	A
モーター電流(peak)	2.2	7	A

L6470 の通信方式は、SPI 通信と呼ばれる通信方式である。SPI 通信について 2.6.3 で解説する。

2.6.3 SPI 通信

SPI はコンピュータ内部で使用するデバイス同士の通信方式であり、比較的低速なデータ転送を行うデバイスに利用される。また、信号線は 4 本で構成され、一つのデバイスを接続する場合は SS を固定することで 3 本の信号線のみで通信できる。

- SCK (Serial Clock)
- MISO (Master In Slave Out)
- MOSI (Master Out Slave IN)
- SS (Slave Select)

今回使用する L6470 の制御は、SPI 通信によって Arduino UNO から行われる。

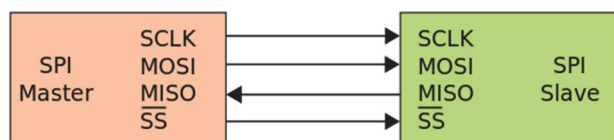


図 2.13 SPI バスでの接続例

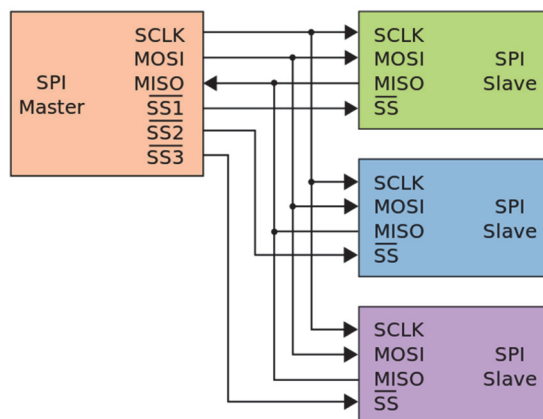


図 2.14 SPI バスでの複数デバイスの接続例

2.6.4 デイジーチェーン接続

観光望遠鏡の向きを操作する2つのステッピングモーターを動かす際に、L6470を2台使用した。L6470を2台使用する際に、Arduino UNOとデイジーチェーン接続を行いSPI通信によって制御した。

デイジーチェーン接続とは、複数の電気・電子機器を数珠つなぎにする、あるいは、全部まとめて1つの輪にするような接続方法である。今回使用するデイジーチェーン接続の回路図を以下に示す。

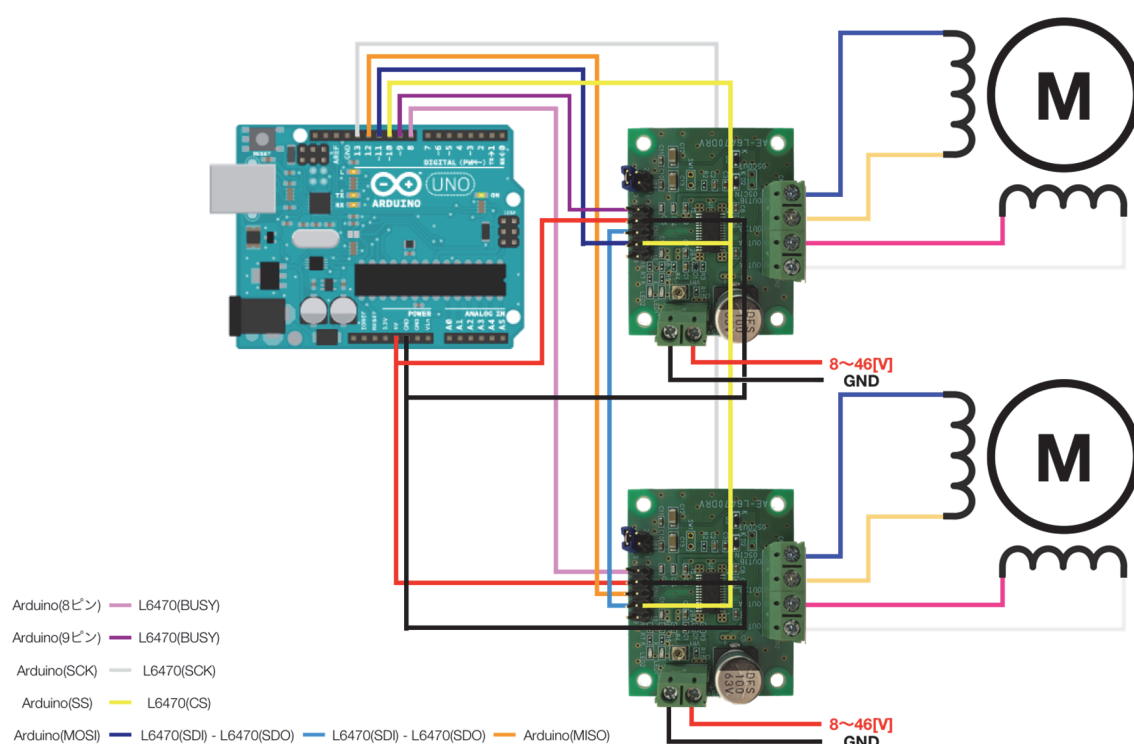


図 2.15 デイジーチェーン接続の回路図 (13)

2.6.5 回路の製作

観光望遠鏡に載せるための回路を Arduino 用ユニバーサル基板 両面黒色レジスト [UB-ARD01] で作成した。図 2.16 は今回製作した自作基板を示している。また、図 2.17 はその回路図を示している。これらの回路を使用し、マイクロステップ駆動方式でステッピングモーターを動作させることができた。その結果、観光望遠鏡が上下左右に滑らかに動作するようになり、今回の研究目的の 1 つである観光望遠鏡の動作の円滑化を達成することができた。

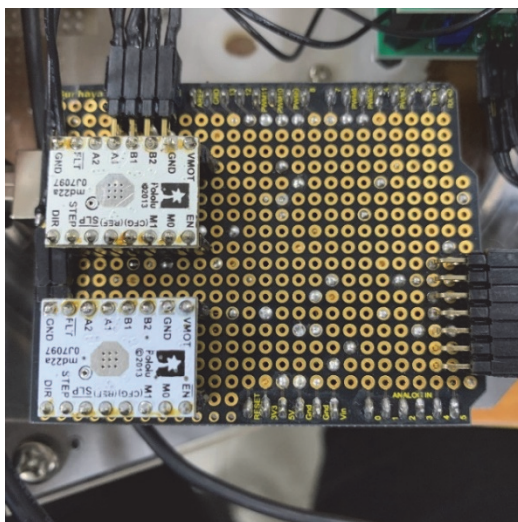


図 2.16 自作基板

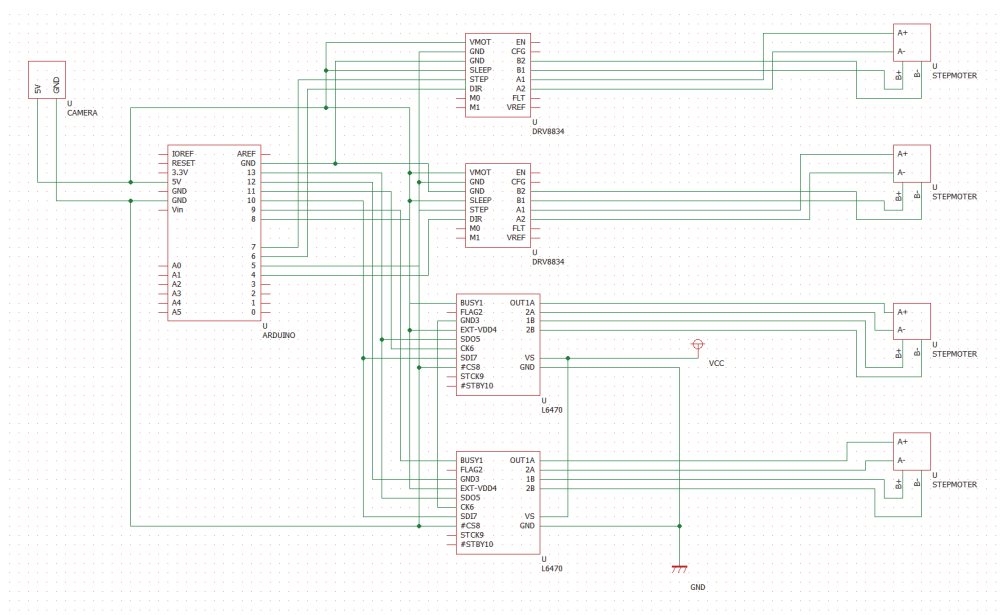


図 2.17 回路図全体

2.7 Arduino UNO プログラム

今回使用した Arduino UNO のプログラムでは、パソコンから命令されたコマンドによって、4 台のステッピングモーターの制御を行っている。

レンズモジュールに取り付けられているフォーカス・ズームを調節する 2 つのステッピングモーターはモータードライバー DRV8834 を使用して制御される。また、観光望遠鏡の向きを操作する 2 つのステッピングモーターは L6470 ステッピングモータードライバーキットを使用して制御される。

DRV8834 は、DIRPIN、STEPPEN の HIGH/LOW で正転、逆転、ステッピングモーターの速度を制御している。DIRPIN は、ステッピングモーターの回転方向を変化させるピンである。HIGH の時に正転し、LOW の時に逆転をする。STEPPEN は、ステッピングモーターの回転速度を制御するピンである。1 度 HIGH にすると、1 ステップ進み、速度を上げるには、短い間隔で ON/OFF 繰り返す必要がある。今回の Arduino UNO のプログラムでは、2ms 間隔で速度を変化させることができる。

また、L6470 の制御は SPI 通信によって Arduino UNO から行われている。あらかじめ L6470 の仕様書に記されているコマンドを Arduino UNO から、送信することで正転、逆転、ステッピングモーターの速度を変化させている。詳しいプログラムは、付録に記されている。

第 3 章 機体 (岩手山担当)

3.1 先行研究の機体

先行研究で開発された観光望遠鏡は機体的安定性に乏しかったため、新たな観光望遠鏡の機体の開発を行った。

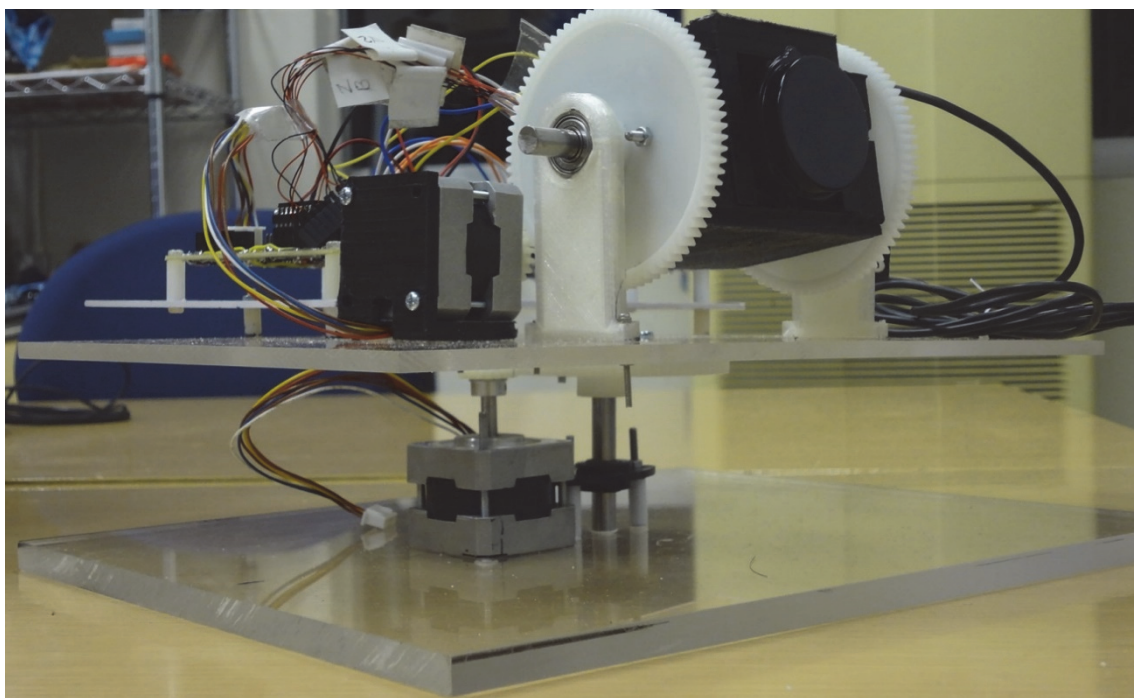


図 3.1 先行研究で製作された観光望遠鏡

3.2 改善図面

1.6 で前述した問題点より、「土台を強化すること」、「カメラを支える柱の数を増やすこと」、「システム全体の重心を下げること」を目指すことにした。

機体の改善を行うにあたり、まず、3次元CADソフトウェアのSolidWorksを用いて以上3つの問題を解決した完成予想図面(図3.2)を製作した。

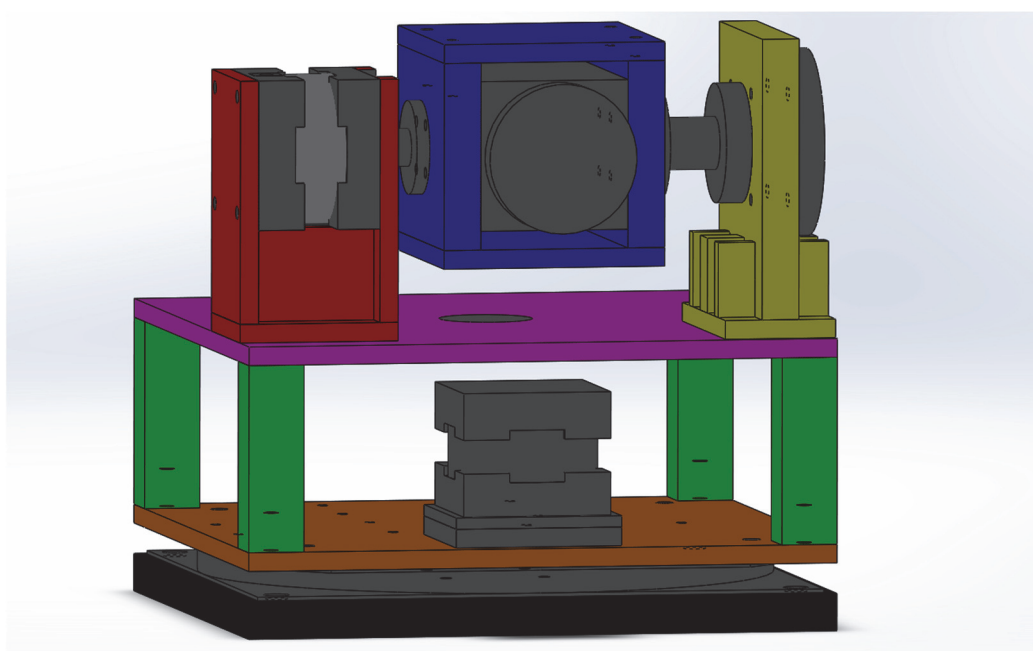


図 3.2 SolidWorks で製作した完成予想図面

(赤：左柱、黄：右柱、青：カメラボックス、桃：上台、

緑：支え柱、橙：中台、黒：下台)

3.3 使用素材

観光望遠鏡の機体製作にはアクリル材を使用する。アクリル材はプラスチックの 1 種である。美しい透明性を持ち、光線透過率が 93% でガラスよりも透き通っているという特徴がある。また、切断、穴あけ、曲げ、接着、彫刻、印刷加工が可能である。耐久性に優れ、破損してもガラス材のように飛び散らないという特徴もある。

アクリル材にはキャスト製法と押し出し製法の 2 種類の製法によるものが存在する。それぞれの違いを表 3.1 に示した。

表 3.1 キャスト製法と押し出し製法によるアクリル材の違い

	キャスト製法	押し出し製法
価格	高価	安価
長所	高硬度、切削加工に向く	板厚精度高。溶剤に溶ける
短所	厚み交差が大きい	耐薬品性に弱い
活用例	写真たて、アクリルケース	大型水槽、レーザー彫刻

今回は、機械加工に向いているキャスト製法のアクリル材を使用した。

3.4 使用材料

機体を製作するために使用した使用材料を表 3.2 に示す。

表 3.2 使用材料と個数

使用材料	個数
アクリル板 160mm×160mm×5mm	2
アクリル板 160mm×160mm×10mm	1
アクリル板 300mm×400mm×5mm	1
アクリル板 300mm×400mm×10mm	1
アクリル板 300mm×400mm×3mm	1
アクリル板（黒） 30mm×40mm×5mm	1
アクリル板（黒） 30mm×40mm×10mm	1
アクリル棒 15mm×15mm×1000mm	1
カメラ	1
モーター	2
M6 ねじ	4
M4 ねじ	8
M3 ねじ	14
ベアリング 穴直径 10 mm	1
シャフトホルダー 穴直径 5 mm	1
シャフトホルダー 穴直径 10 mm	1
シャフト 直径 10 mm 長さ 40 mm	1
ギア 歯数 80	1
ターンテーブル 150 mm角型ベアリング	1

3.5 加工方法

加工に使用した工具は帯鋸、フライス盤、ボール盤、タップである。これらの工具を用いて切削加工、穴あけ加工、ねじ穴の製作を行った。

機体の加工は、図 3.2 の 7 つの部品に分けて行い、それぞれを完成させた後にこれらを合わせて作り上げた。

以下、3.5.1～3.5.7 ではそれぞれの部品の詳しい加工方法を説明する。

3.5.1 右柱

ここでは右柱の製作方法を記す。右柱は、カメラボックスを右側から支える台である。

まず、アクリル板 30mm×40mm×5mm とアクリル板 30mm×40mm×10mm から帯鋸で必要なサイズを切り取り、その後フライス盤を用いて切削加工を行った。また、右柱のベアリングを取り付けるために M4 のねじ穴を製作した。M4 のねじ穴はボール盤を用いて図面の寸法通り穴をあけた後、タップを用いて製作した。ねじ穴の製作を手動で行ったのは、アクリル材が急激な回転による加工で中のねじ穴が潰れたり、割れる可能性があったためである。すべてのパーツの完成後、アクリル用ボンドを用いてそれらを接着した。図 3.3.1～3.3.4 はそれぞれ加工を行ったパーツと個数を示しており、これらを組み合わせた後、購入したベアリングを図 3.3.3 のパーツにねじ止めしたものが図 3.3 の完成図である。カメラボックスを支える際には直径 10 mm、長さ 40 mm のシャフトをベアリングに通し使用する。

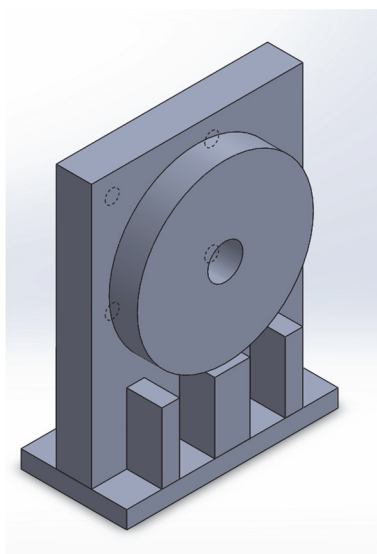


図 3.3 右柱完成図

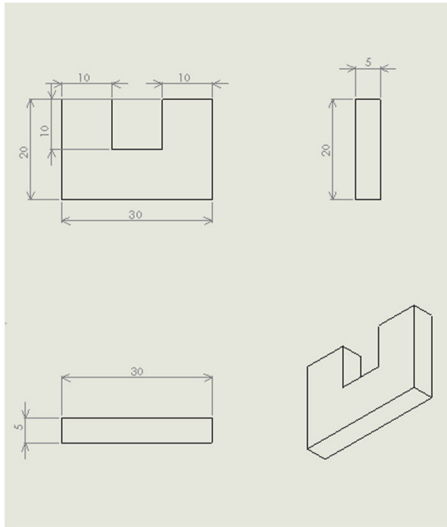


図 3.3.1 パーツ 1 × 2

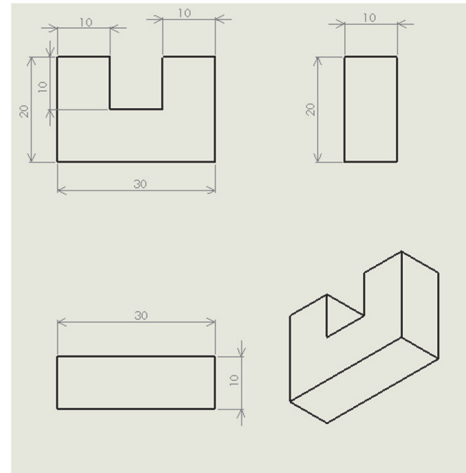


図 3.3.2 パーツ 2 × 1

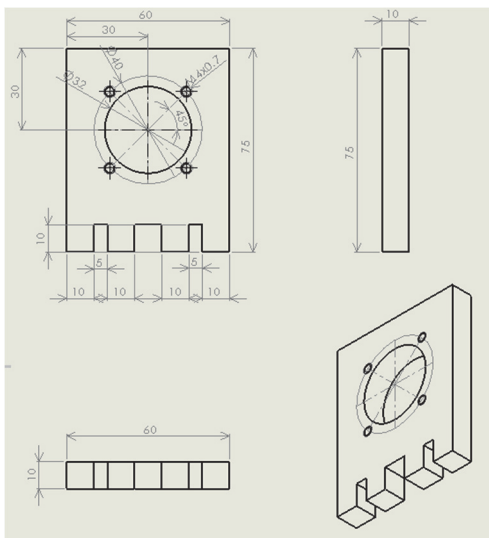


図 3.3.3 パーツ 3 × 1

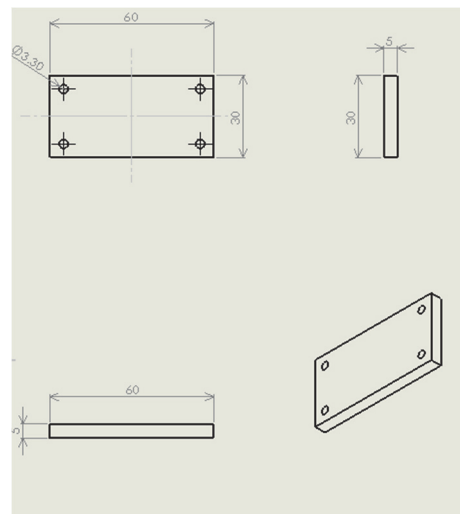


図 3.3.4 パーツ 4 × 1

3.5.2 左柱

ここでは左柱の製作方法を記す。左柱はカメラボックスをモーターで直接左側から支える台である。

まず、アクリル板 30mm×40mm×5mm とアクリル板 30mm×40mm×10mm から帯鋸で必要なサイズを切り取り、その後フライス盤を用いて切削加工を行った。また、モーターを通すための U 字のパーツはボール盤で製作し、その他はフライス盤の刃をドリルに変更して正確な寸法の穴をあけた。すべてのパーツの製作後、アクリル用ボンドを用いて接着した。図 3.4.1～3.4.4 はそれぞれ加工を行ったパーツと個数を示しており、図 3.4 はモーターを図 3.4.1、3.4.2 に M3 ネジで固定した完成図である。

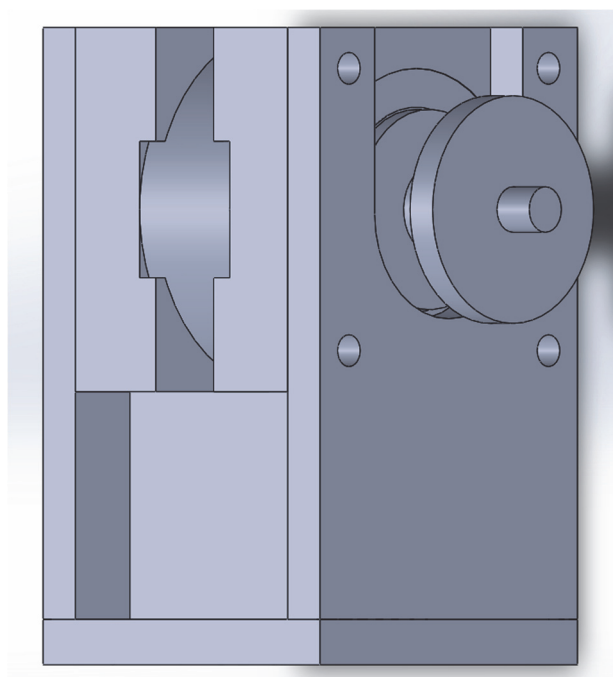


図 3.4 左柱完成図

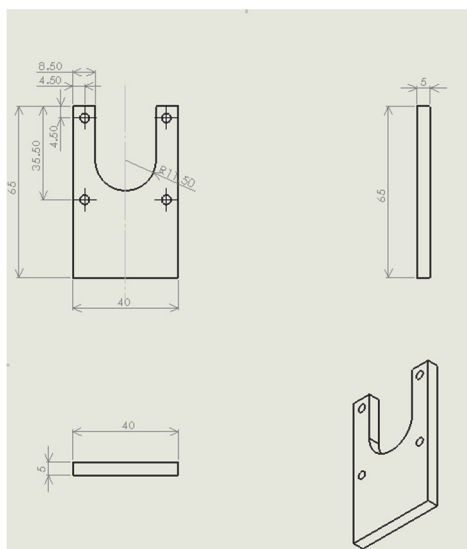


図 3.4.1 パーツ 1 × 1

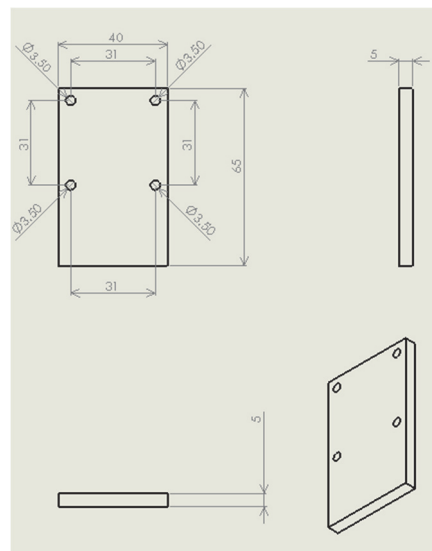


図 3.4.2 パーツ 2 × 1

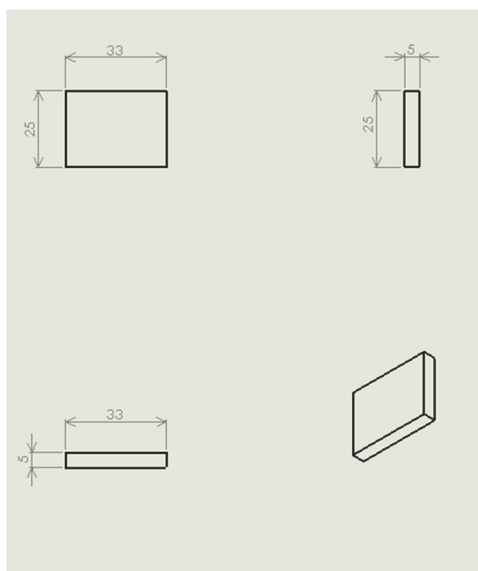


図 3.4.3 パーツ 3 × 2

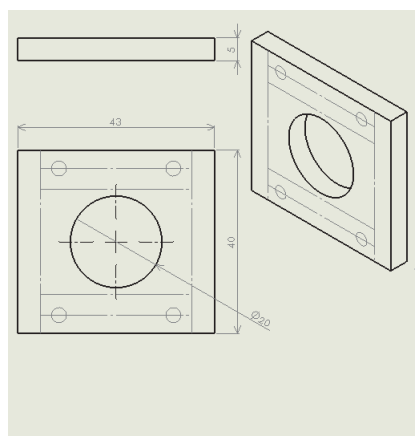


図 3.4.4 パーツ 4 × 1

3.5.3 カメラボックス

ここではカメラボックスの製作方法を記す。カメラボックスの上下にウレタン吸収剤を接着し、挟み込むことでボックス内にレンズモジュールを固定する。左柱にあるモーターとシャフトホルダーを直接ねじで固定することでモーターの動きに連動し、上下に動く仕組みである。

まず、アクリル板（黒） $30\text{mm} \times 40\text{mm} \times 5\text{mm}$ とアクリル板（黒） $30\text{mm} \times 40\text{mm} \times 10\text{mm}$ から帯鋸で必要なサイズを切り取り、フライス盤を用いて切削加工をした。また、カメラボックスは左右にシャフトホルダー固定用のねじ穴が必要であった。このことから、フライス盤の刃をドリルに変更した後に穴をあけて、タップを用いて手動で M3 のねじ穴を製作した。すべてのねじ穴が完成した後、それぞれのパーツをアクリル用ボンドで接着してカメラボックスを作った。右柱から伸びるシャフトと左柱から伸びるモーターがカメラボックスの左右のシャフトホルダーに固定される。図 3.5 は左右のシャフトホルダーを取り付けた完成図である。

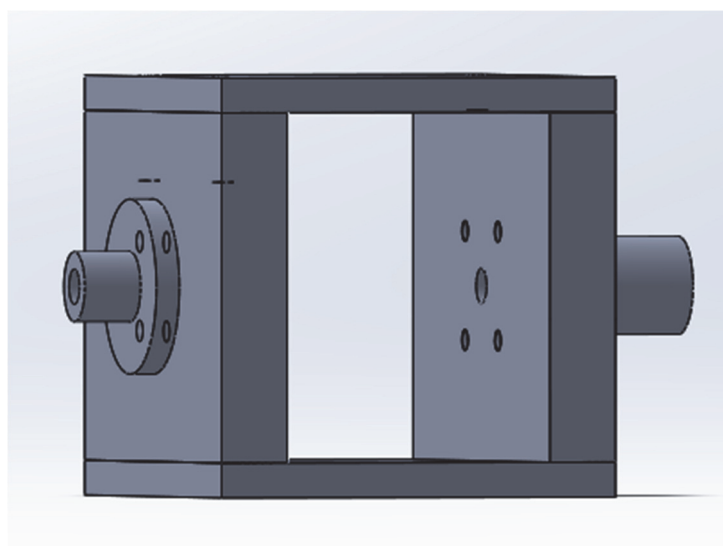


図 3.5 カメラボックス完成図

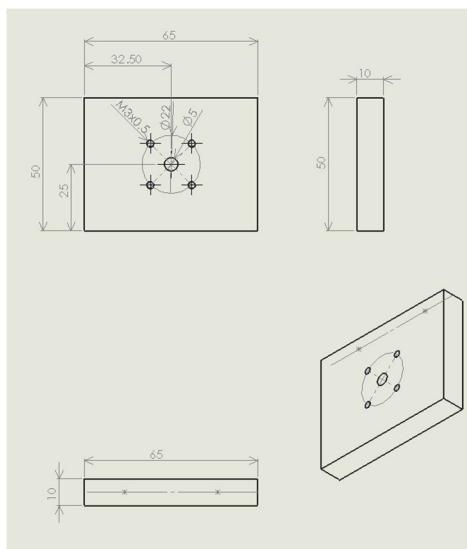


図 3.5.1 パーツ 1 × 1

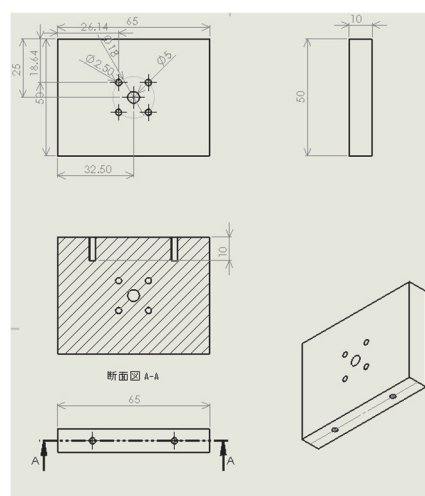


図 3.5.2 パーツ 2 × 1

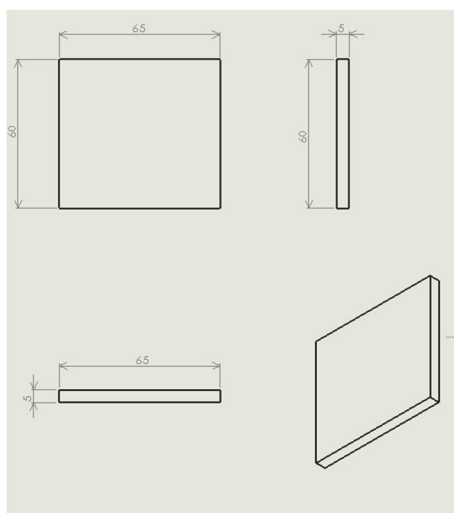


図 3.5.3 パーツ 3 × 2

3.5.4 上台

上台は、160mm×160mm×50mm のアクリル板にボール盤で 26mm と 20mm の穴を開けて作成した。上台の中央の穴はカメラの配線と中台に置かれている電子配線を接続するために開けた。また、20mm の穴はモーターの配線と中台に置かれている基板に接続するために開けた。

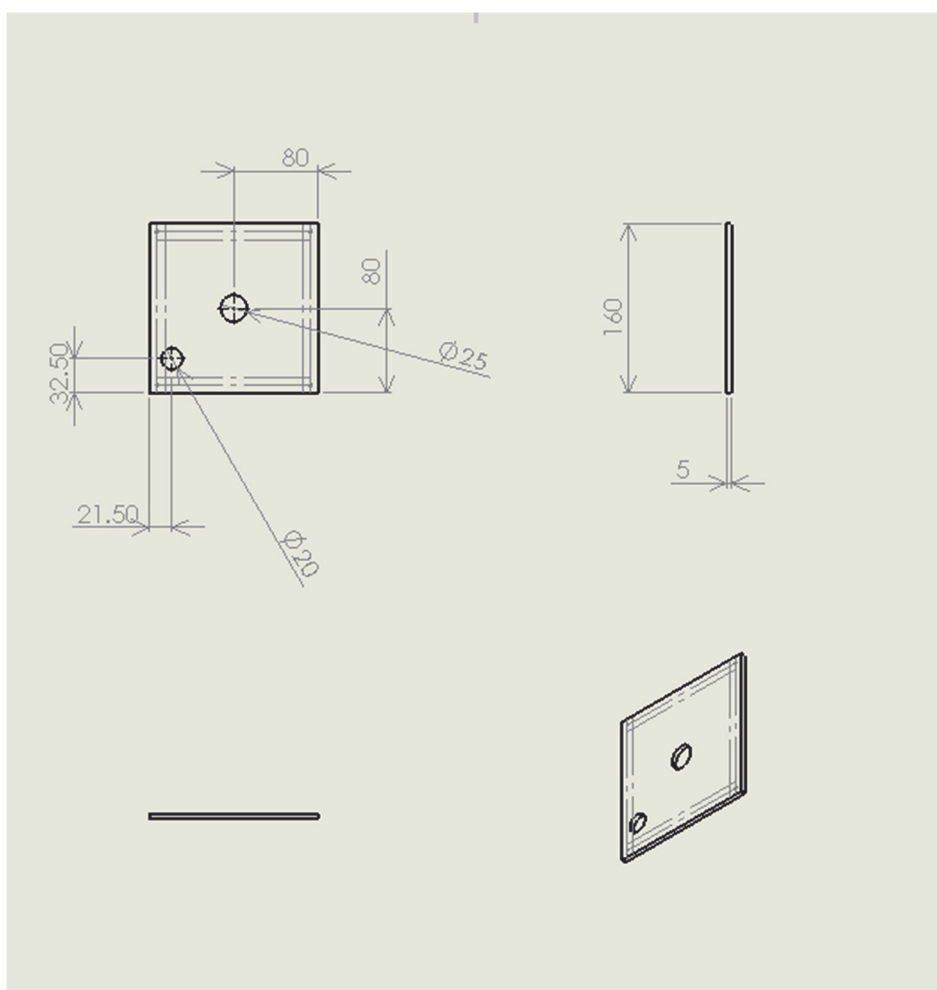


図 3.6 上台の完成図

3.5.5 支え柱

ここでは支え柱の製作方法を記す。まず、15 mm×15 mm×1000 mmの亚克力棒から帯鋸で必要なサイズを切り取った。その後、フライス盤で旋盤加工を行うことで15 mm×15 mm×50 mmの柱を4つ製作した。また、中心にボール盤で深さ10 mmの穴をあけ、タップを用いて深さ10 mmのM4のねじ穴を製作した。これを支え柱として使用した。

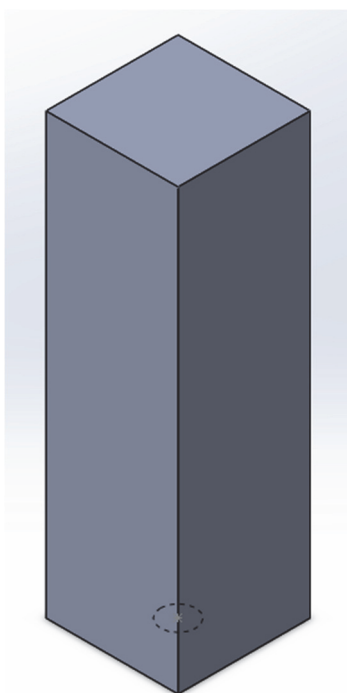


図 3.7 支え柱の完成図

3.5.6 中台

ここでは中台の製作方法を記す。中台の上には Arduino UNO などの電子基板とモーターが取り付けられ、下にはターンテーブルが取り付けられる。

まず、160mm×160mm×5mm のアクリル板に卓上ボール盤とボール盤で穴を開けたものを製作した。また、あけた穴からモーターの駆動部を下台にあるギアとかみ合わせるための高さ 8 mm の調整台が必要だった。このため、アクリル板 300mm×400mm×5mm、アクリル板 300mm×400mm×3mm から必要なサイズを帯鋸で切り取った後にフライス盤で寸法を調整した。また、穴あけを行ってできたものをアクリル用ボンドで接着し、8 mm の高さ調整台を製作した。図 3.8.1～3.8.3 を組み合わせてできたものが図 3.8 の完成図である。

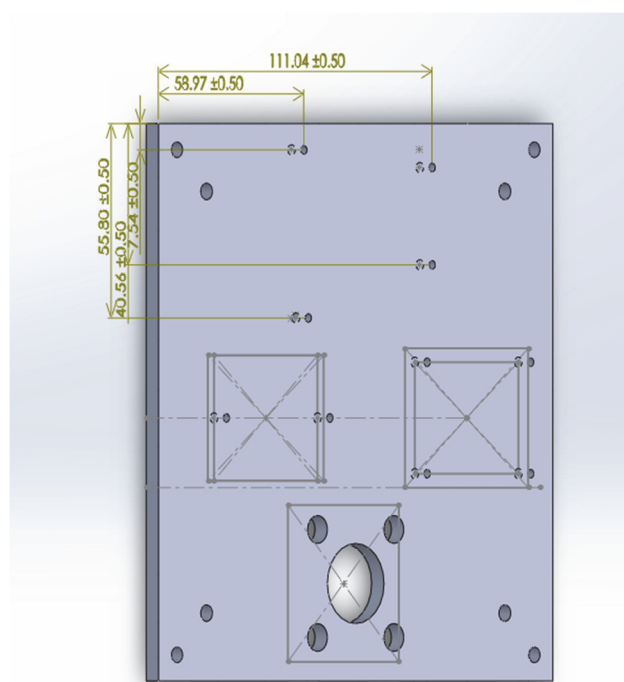


図 3.8 中台完成図

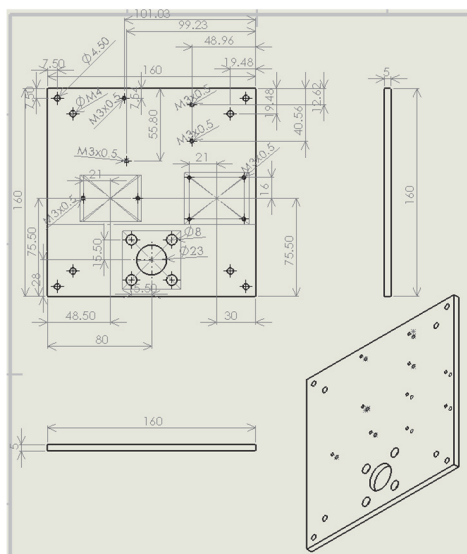


図 3.8.1 パーツ 1 × 1

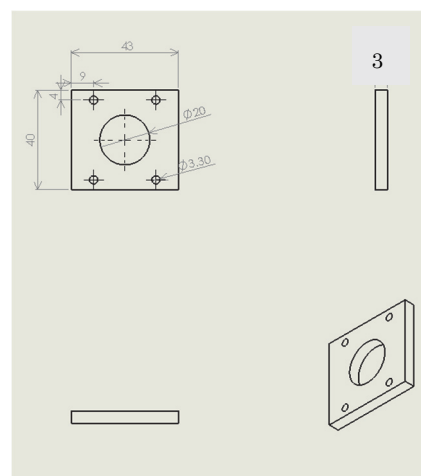


図 3.8.2 パーツ 2 × 1

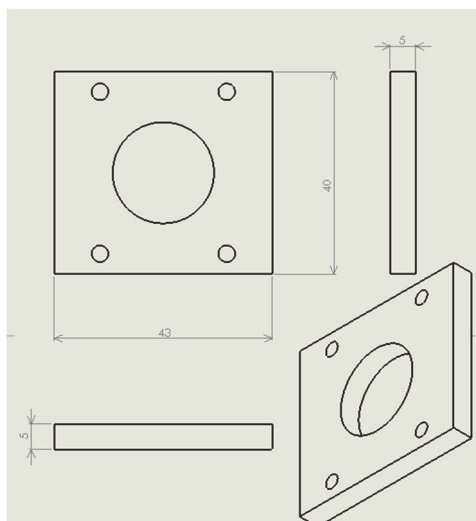


図 3.8.3 パーツ 3 × 2

3.5.7 下台

下台は、アクリル板 160mm×160mm×10mm にボール盤で穴を開けて作成した。この上に直径 80 mm、歯数 80 のギアと 50 mm 角型ベアリングターンテーブルを固定することで、ギアの周りをモーターが円を描くように滑らかに移動し、カメラの横移動ができるようになる。

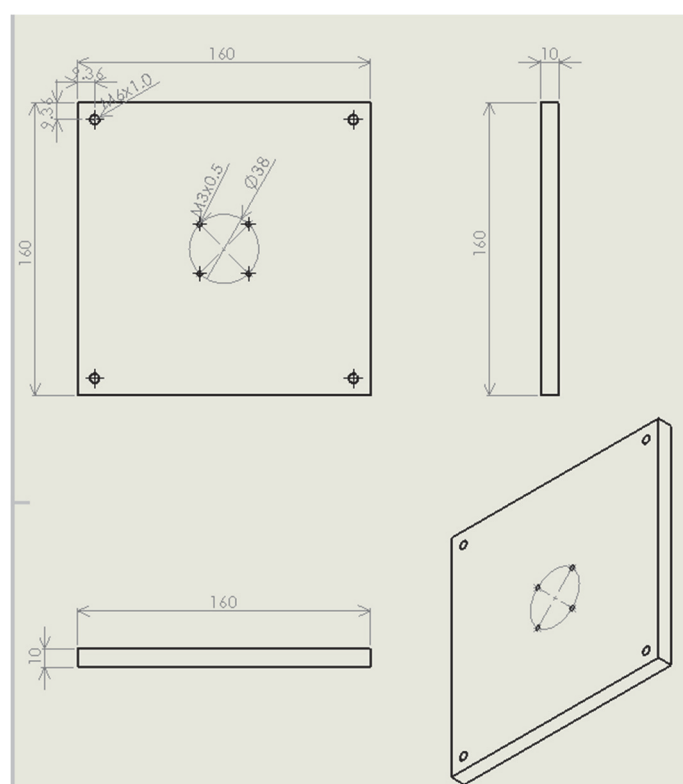


図 3.9 下台完成図

3.6 中台の基板配置

図 3.8 の中台に Arduino UNO、ユニバーサル基板×2、モータードライバー×2 を配置した。基板をねじ止めするために、アクリル板上にねじとスペーサーで固定を行った。また、スペース確保のため、モータードライバーを2段に重ね、Arduino UNO の上にユニバーサル基板を設置した。図 3.10 は基板の配置図である。

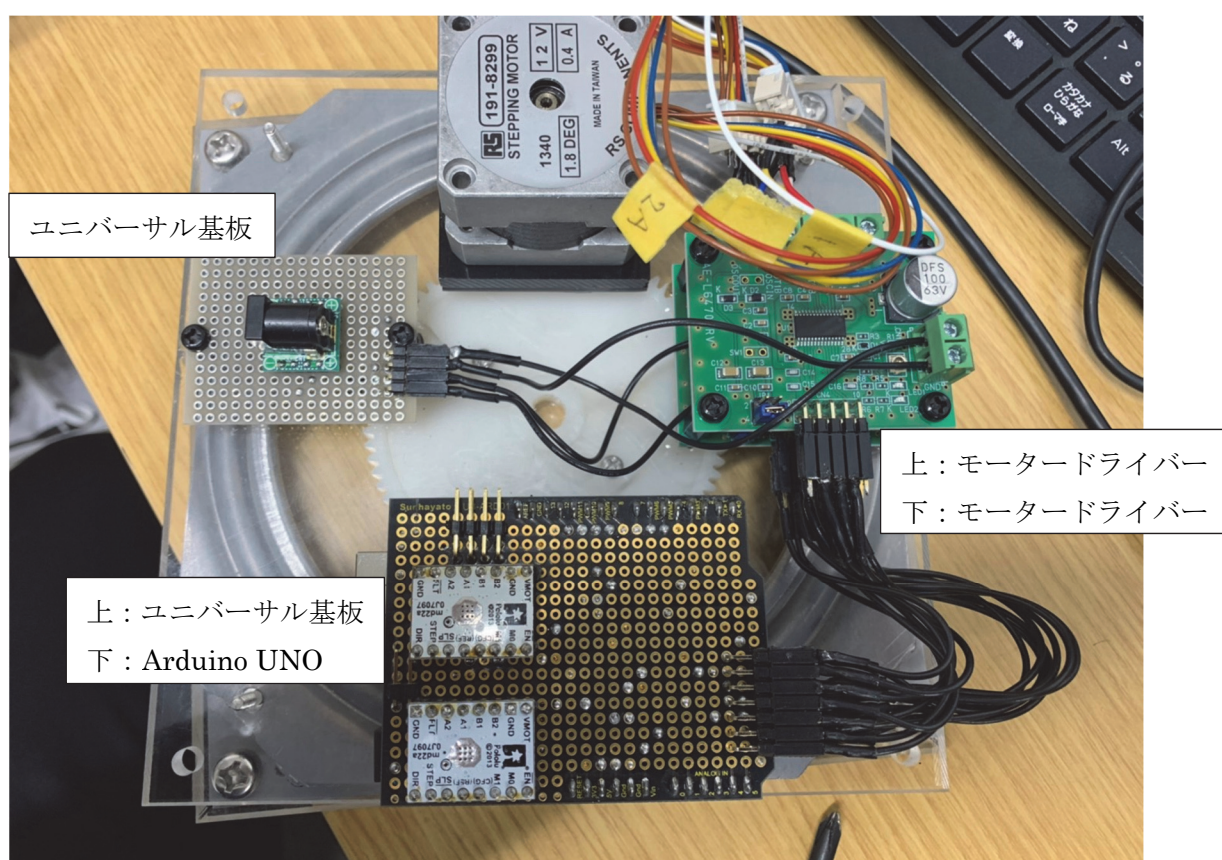


図 3.10 基板の配置図

3.7 カメラの上下の動作

ここではカメラの上下の動作原理について説明する。カメラの上下の動作は図 3.3～3.5 で製作した部品によって行われる。まず、Arduino UN0 から送られた正転、逆転の信号をモーターが受け取る。これにより、モーターが図 3.11 のように回転し、中に入っているカメラはその景色を得ることができる。

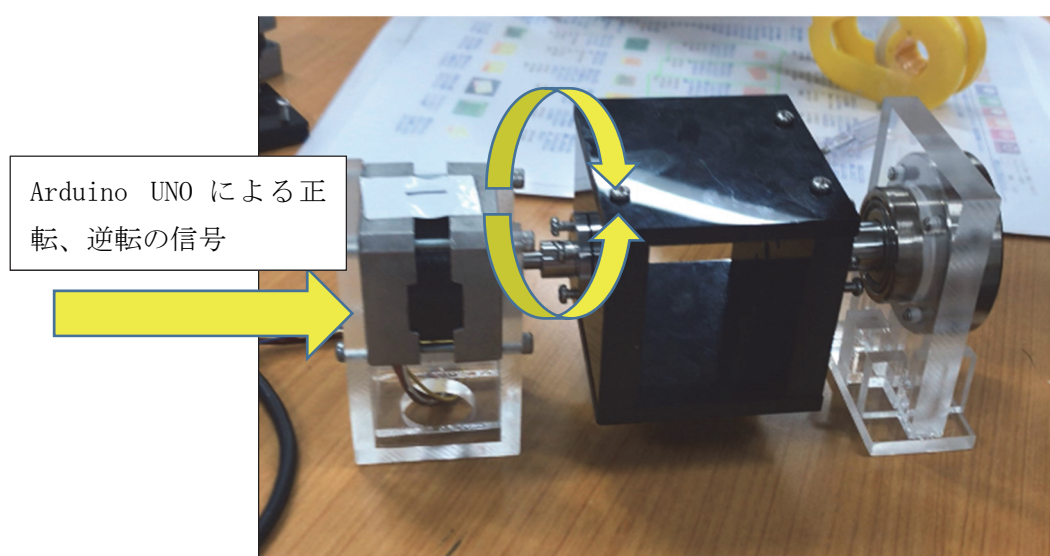


図 3.11 カメラの上下の動作原理

3.8 カメラの左右の動作

ここではカメラの左右の動作原理について説明する。カメラの左右の動作は図 3.8 と図 3.9 で作成した部品によって行われる。図 3.8 と図 3.9 の部品は図 3.12 のようにターンテーブルに取り付けられる。ターンテーブルの上部分に図 3.8 の中台、下部分に図 3.9 の下台という配置になっている。また、図 3.9 の下台の中心には直径 80 mm、歯数 80 のギアが取り付けられている。これにより、モーターがギアの周りを動作して図 3.8 の中台に設置されているカメラは左右の景色を得ることができる。

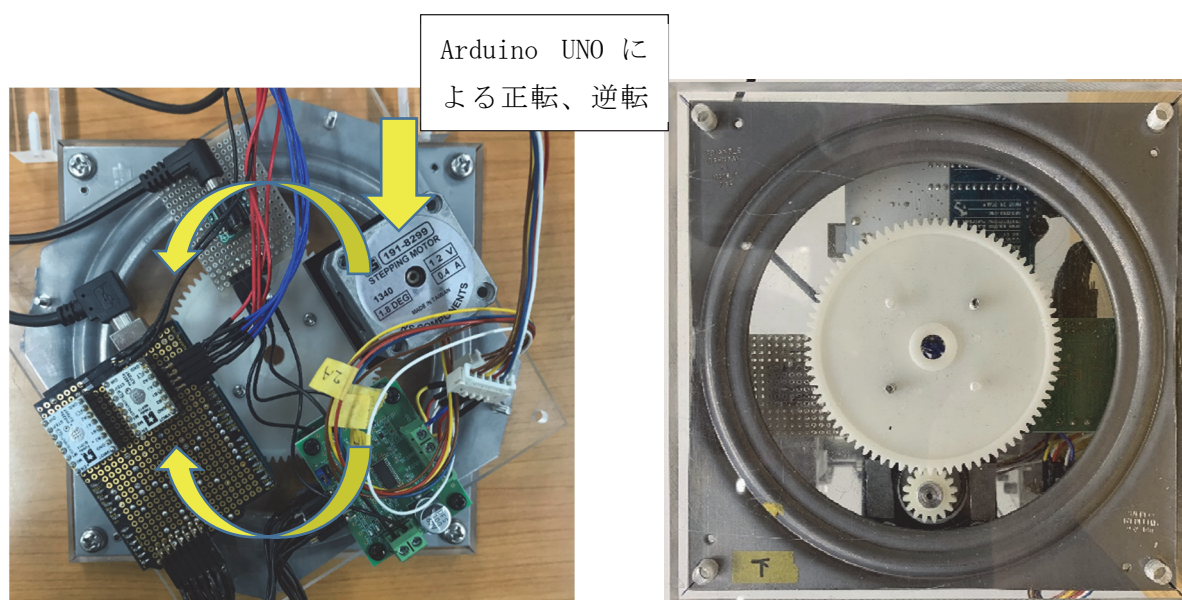


図 3.12 カメラの左右の動作原理 左)表 右)裏

3.9 新たに完成した機体

図 3.3～3.9 のパーツを使い機体を作り上げることで、一番下の台に重さのあるターンテーブルを固定し、重心の低い部分に基板を配置することができた。さらに、カメラを左右からの軸でしっかりと固定することができた。これにより前述した 1.6 の課題、「土台を強化すること」、「カメラを支える柱の数を増やすこと」、「システム全体の重心を下げること」を達成することができたといえる。また、新たに製作した機体の大きさが全体で 160 mm×160 mm×160 mm となったのでコンパクト化にも成功した。

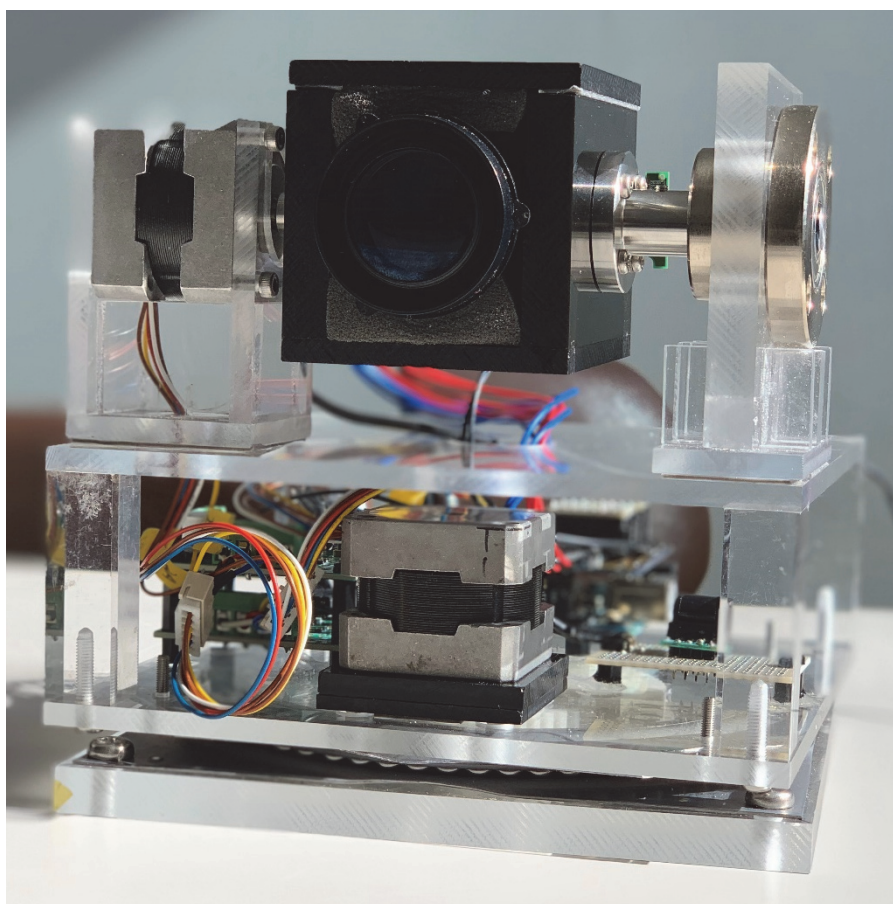


図 3.13 新たに完成した機体

第4章 オートフォーカス (金井担当)

オートフォーカスとはカメラのフォーカスを自動的に合わせる機能である。近年販売されているスマートフォンやカメラのほとんどもに装備されている。オートフォーカスの方式は大きくアクティブ方式とパッシブ方式の2つに区分される。

アクティブ方式はレーダーと同様の原理で、対象物に赤外線や超音波などを照射し、その反射波が戻ってくる時間や照射角度によって距離を検出する方法である。メリットは明かりが少ない環境でも使用可能なことである。反対に、風景写真などの遠距離のピントは、反射波が届かないことや、届いても微弱なため、フォーカスを合わせにくいという欠点がある。

一方、パッシブ方式はレンズに映る光を測定する方式である。取り込んだ映像を元にフォーカスを合わせるため、遠距離へのピントを合わせやすいというメリットがある。しかし、暗い場所など光が少ないものにピントを合わせるのを苦手とする。

今回は望遠レンズを使用するため、パッシブ方式のコントラストオートフォーカスを採用した。

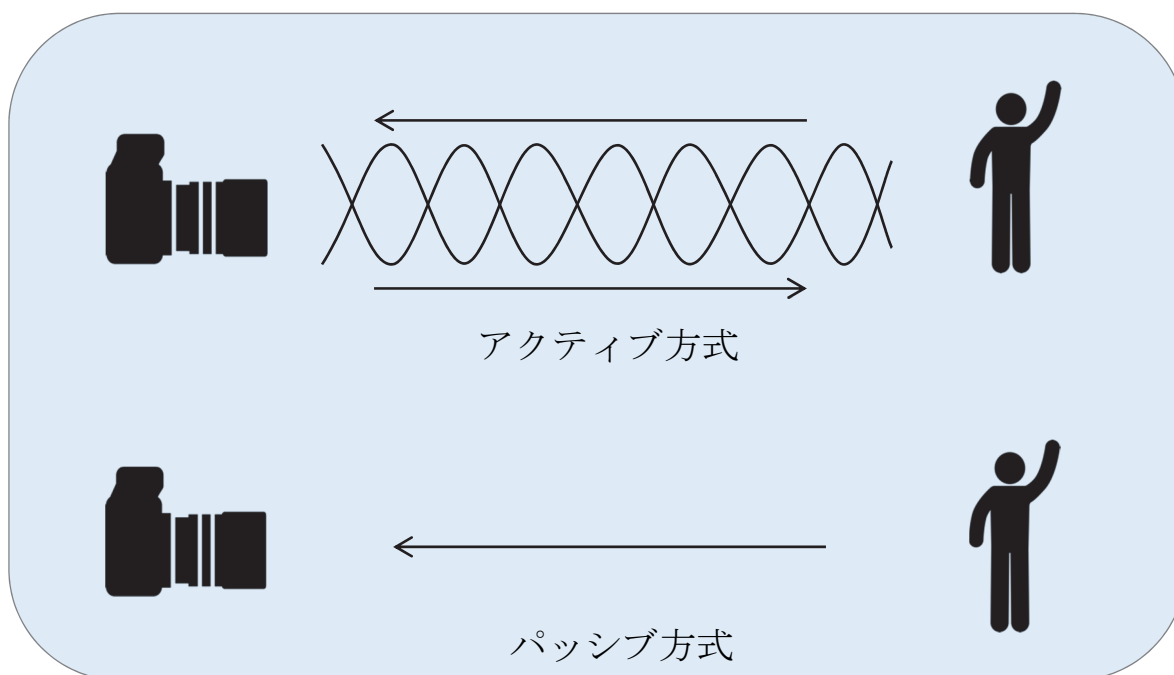


図 4.1 オートフォーカス方式の違い

4.1 コントラストに基づくオートフォーカス

コントラストオートフォーカスは、画像のコントラストを用いたオートフォーカスである。

ピントは、収差という光のズレをなくすことでピントがあった状態となる。収差があると、像のシャープさが失われ、コントラストが悪くなってしまう。これを利用して、コントラストが高くなる方向へレンズを自動で動かすことでフォーカスを合わせる。

4.1.1 コントラストの計算

本研究では、Michelson コントラストによりコントラストを計算する。

Michelson コントラスト C は、グレースケール化された画像に対して、その最大輝度、最小輝度 L_{max}, L_{min} を用いて次式で計算される。

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}}$$

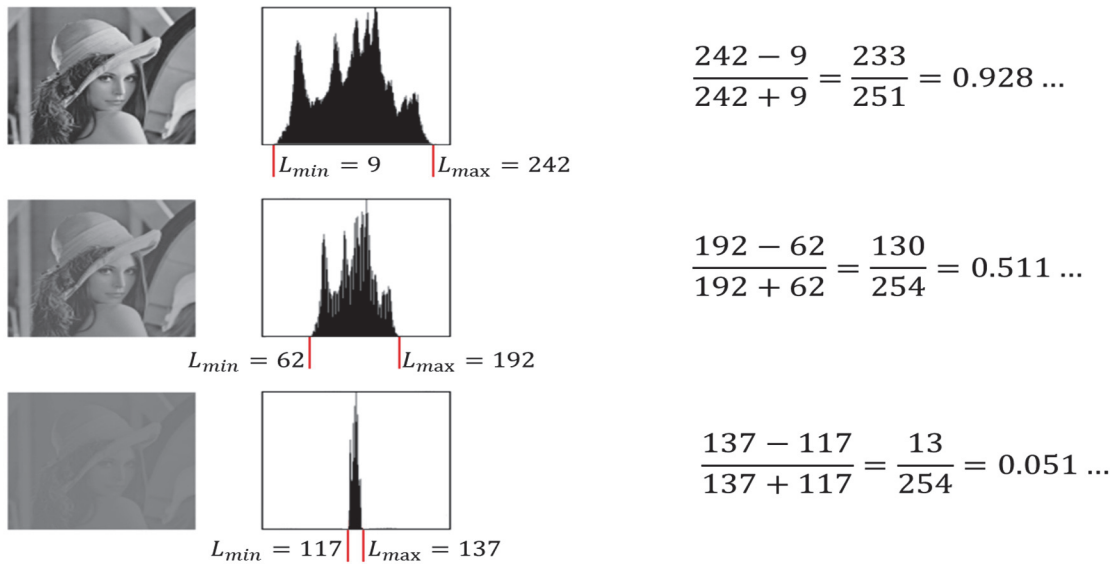


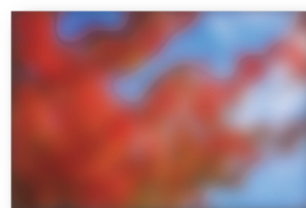
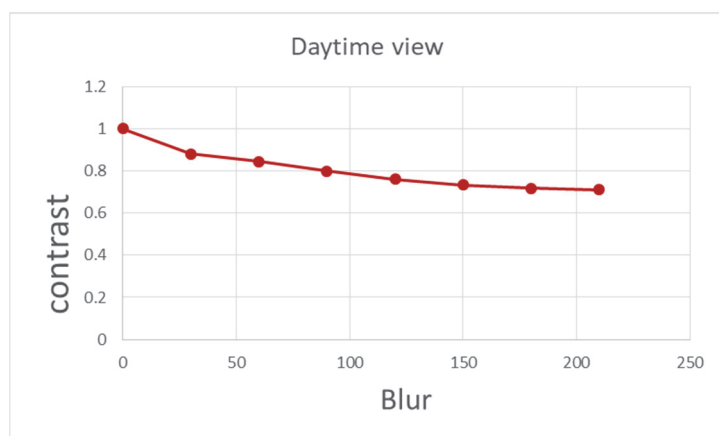
図 4.2 コントラスト計算の例 (15)

4.1.2 コントラスト実験

図 4.3 と図 4.4 は、人工的に画像をぼかし、収差がある状態を実現した際のコントラストの違いを示したものである。

人工的にぼかすためガウスぼかしを用いた。ガウスぼかしは、周囲の指定された半径の画素の値から平均値を算出し、置き換えることで画像をぼかす処理である。指定するぼかし半径が大きいほど、ぼやけた画像となる。横軸の Blur はこのぼかし半径の数値となっている。今回はぼかし半径を 30 ずつ変えて計算を行った。

明るめの画像と、暗めの画像で再現を行った。どちらの画像も画像がぼけるにつれて、コントラストの値が小さくなっているのがわかる。

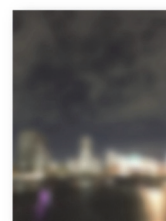
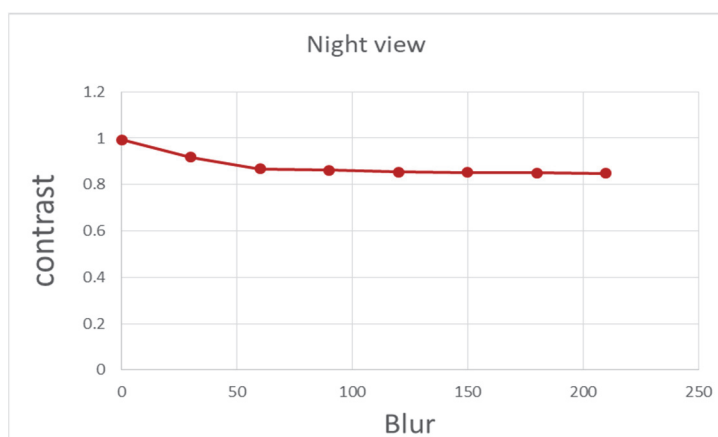


ぼかし半径 210



ぼかし半径 0

図 4.3 人工的に画像をぼかした際のコントラストの変化(明るい画像)



ぼかし半径 210



ぼかし半径 0

図 4.4 人工的に画像をぼかした際のコントラストの変化(暗い画像)

4.2 コントラストに基づくオートフォーカスのプログラム作成

オートフォーカスのプログラムは大きく

- 1、フォーカス状態の判断
- 2、回転速度の制御

という手順を繰り返し行う。

詳しいプログラムは付録に添付したため、ここではこの2つの手順について大まかに説明する。

4.2.1 フォーカス状態の判断

今回のプログラムは新旧二枚の画像を比較することで、フォーカスの状態を判断する。古い画像のコントラストより新しい画像のコントラストが大きい場合、モーターはフォーカスが合う方向へと回転していると言える。

逆に、新しい画像のコントラストより古い画像のコントラストが大きい場合、モーターはフォーカスが合わない方向へ回転していることになる。この場合モーターを今とは逆回転させるように信号を送る。

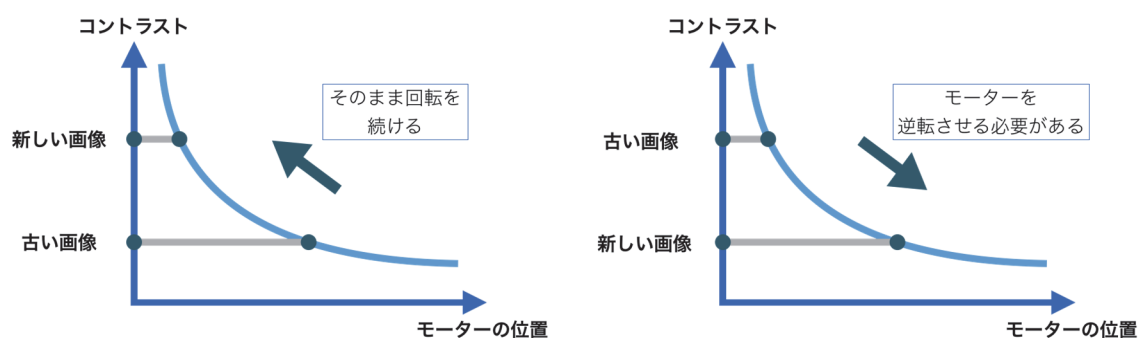


図 4.5 モーターの回転方向イメージ図

4.2.2 回転速度の制御

レンズモジュールのモーターは、1ステップの時間を調節することで速度を調節する。

今回作成したオートフォーカスは、常にフォーカスを調節するモーターが駆動している。また、コントラストを解析している間もモーターは駆動しているため、モーターの回転速度の違いにより、オートフォーカスの特性が変わる。

精度の違いを示すため1ステップの時間が2msと18msの時のオートフォーカスを比較した。

図4.6の景色をフォーカスが合っていない状態からプログラムを動作させ、計測を行った。図4.7は、1ステップ2msのグラフ、図4.8は、1ステップ18msのグラフである。横軸は、時間ステップ、縦軸はコントラストを示している。また、オレンジのラインは、ピントが合っている状態の値を示している。モーターが早い場合、大きくピントが合っていない状態から、ピントが合うまで比較的短時間で行うことができる。しかしモーターの速度が速すぎるため、ブレがとても大きい。逆に、モーターが遅い場合、ピントが合うまで非常に時間がかかる。一方で、ピントが合った状態でのブレがあまりないことがわかる。



図 4.6 実験に用いた風景

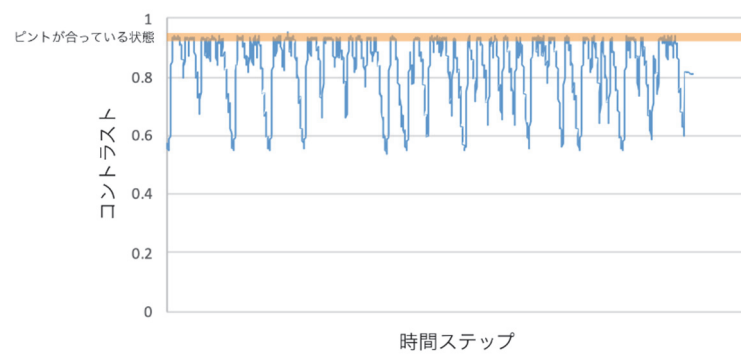


図 4.7 1 ステップ 2ms の場合のオートフォーカス



図 4.8 1 ステップ 18ms の場合のオートフォーカス

この問題を解決するため、コントラストの値によってモーターの速度を変化させるようにした。

図 4.9 は、速度変化の分布を示したものである。コントラストが低い場合、モーターを速く移動させるようにし、コントラストが高い場合、より精密なコントラストを実現するため、モーターの速度が遅くなるよう設定した。

図 4.10 は図 4.7、図 4.8 と同じ実験を行った際のグラフである。速度を変化させることで素早くピントが合い、ピントがあった状態で安定していることがわかる。

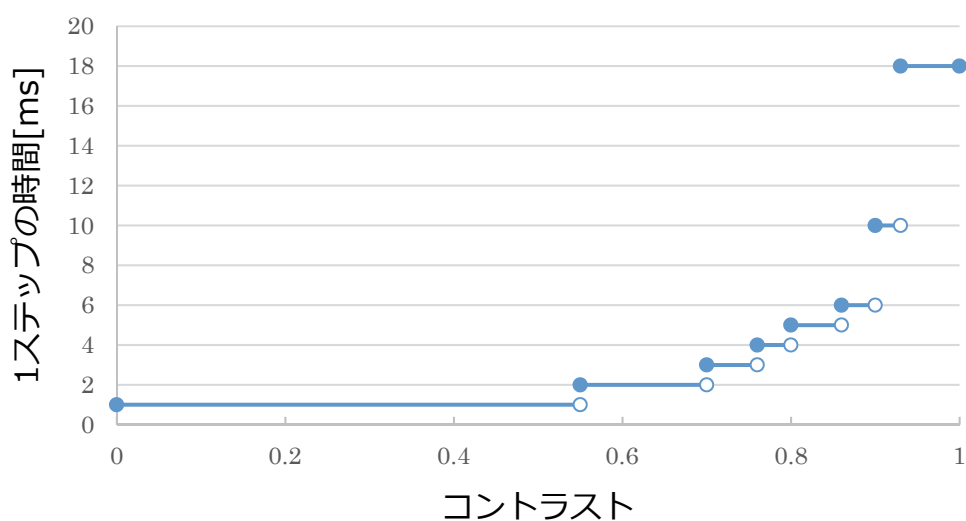


図 4.9 コントラストの値に応じたモーターの回転速度

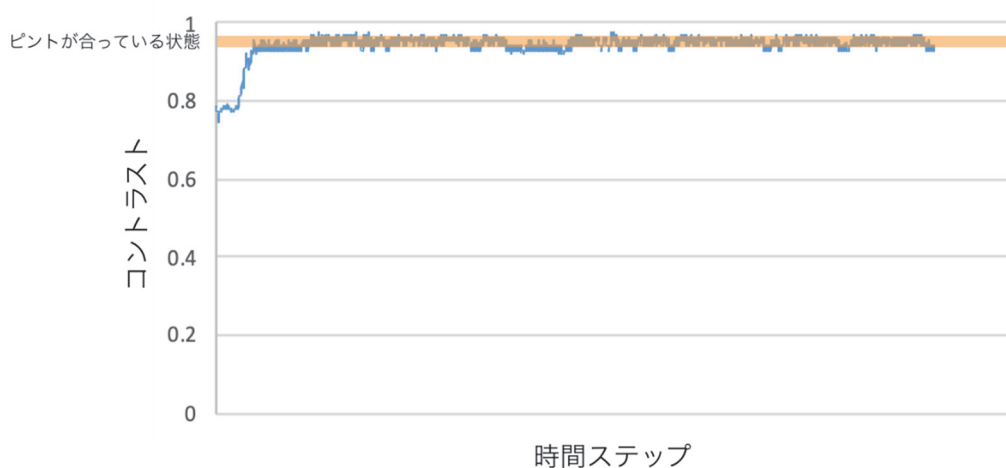


図 4.10 回転速度を可変式にした場合のオートフォーカス

第 5 章 結論 (青木担当)

先行研究で開発された観光望遠鏡の問題点を改善し、新たな観光望遠鏡システムの開発を研究目的とした。表 5.1 は先行研究で開発された観光望遠鏡の問題点を示している。

表 5.1 先行研究で開発された観光望遠鏡の問題点

問題点 1	上下左右に滑らかに動作をしない
問題点 2	機械的安定性に乏しい
問題点 3	フォーカスを合わせるのに時間がかかる

表 5.1 の問題点を改善するための目標を表 5.2 に示した。

表 5.2 問題点に対しての目標

目標 1	観光望遠鏡の動作の円滑化
目標 2	機械的安定性の向上・コンパクト化
目標 3	オートフォーカス機能の付与

これらの目標を達成するように、開発した観光望遠鏡を図 5.1 に、開発した観光望遠鏡で撮影された景色を図 5.2 に示した。また、本研究での観光望遠鏡システムの概要を図 5.3 に示す。



図 5.1 開発した観光望遠鏡



図 5.2 開発した観光望遠鏡で撮影された景色

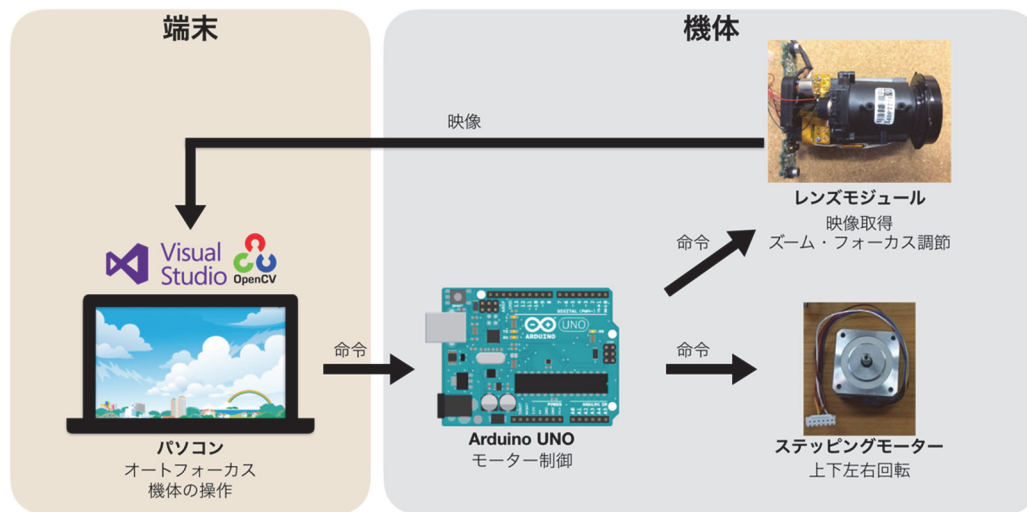


図 5.3 本研究での観光望遠鏡システムの概要図

次に、表 5.2 の目標 1 から 3 の達成状況について述べる。

- 目標1. 新たに回路を製作し、ステッピングモーターの駆動方式を、フルステップ駆動方式からマイクロステップ駆動方式に変更した。その結果、ステッピングモーターをより微小なステップ角で動作させることに成功し、観光望遠鏡が滑らかに動作するようになった（2 章）。
- 目標2. 先行研究で開発された機体の機構は重心が高く、その上に重量のあるものを載せているため、安定性に乏しかった。完成した機体（図 4.1）では、中台に Arduino UNO などの基板を載せ、重心の低い機構を製作することで安定が向上した。新たな観光望遠鏡のサイズが 160mm×160mm×160mm と先行研究で開発された観光望遠鏡（図 1.4）よりも小型になり、本研究の目標である小型化を実現した（3 章）。
- 目標3. ピント調節用のモーターをコントラストが高くなる方向へ動かすプログラムを作成し、新たな観光望遠鏡にパッシブ方式のコントラストオートフォーカス機能を付与させることができた。また、コントラストの値によってモーターの速度が変化するような変速機能を取り入れ、より精密なオートフォーカスを実現することができた（4 章）。

本研究で開発した観光望遠鏡は、目標 1 から目標 3 を達成することができた。また、観光客のニーズに合った機能を付与させるなど、より革新的な機能を付与させることが今後の課題である。

謝辞 (青木担当)

研究活動を通して熱心なご指導をして頂いた先進工学部機械理工学科金丸隆志准教授に深く感謝を申し上げます。

また、部品加工で細かく指導してくださった工学院大学 19 号館(ふらっと)所属の先生方に感謝申し上げます。

そして知能機械研究室に在籍する先輩方やご指導いただいた関係者の皆様に感謝いたします。

参考文献

- (1) 東京タワー展望台のご案内. (引用日 2019 年 1 月 22 日)
<https://www.tokyotower.co.jp/observatory/>.
- (2) 国土交通省 観光庁 訪日外国人数 観光庁統計データ. (引用日: 2019 年 1 月 24 日.)
http://www.mlit.go.jp/kankocho/siryou/toukei/in_out.html.
- (3) 吉野晃平 ” タブレット端末を用いた観光望遠鏡システムの開発” . 工学院大学修士論文, 2016.
- (4) ステッピングモーターのための基礎概要. (引用日: 2019 年 1 月 27 日.)
http://sekkei.if.land.to/item_stepping.html.
- (5) ステッピングモーターの構造. (引用日: 2019 年 1 月 30 日.)
http://www.orientalmotor.co.jp/tech/reference/stepping_motor01/.
- (6) RS 社 191-8299. (引用日: 2019 年 2 月 1 日.)
<http://uk.rs-online.com/web/p/stepper-motors/1918299/>.
- (7) ステッピングモーターの仕組みを理解しよう. (引用日: 2019 年 2 月 2 日.)
<https://dotstud.io/blog/stepper-motor-nefrybt-control/>.
- (8) 谷腰欣司「ステッピングモーターの実用技術 モーターの基礎から駆動メカニズムまで」. 電波新聞社, 2006.
- (9) 坂本正文「ステッピングモーターの使い方」. オーム社, 2003.
- (10) ArduinoBoardUno. (引用日: 2019 年 1 月 29 日.)
<https://www.arduino.cc/en/Main/ArduinoBoardUno>.
- (11) DRV8834 Low-Voltage Stepper Motor Driver Carrier. (引用日: 2019 年 1 月 29 日.)
<https://www.pololu.com/product/2134>.

- (12) 秋月電子通商 L6470 使用 ステッピングモータードライブキット.
(引用日: 2019 年 2 月 2 日.)
<http://akizukidenshi.com/catalog/g/gK-07024/>.
- (13) L6470 デイジーチェーン接続のテスト. (引用日: 2019 年 1 月 22 日.)
<http://spinelify.blog.fc2.com/blog-entry-81.html>.
- (14) 吉野晃平, 他 5 名 ” Android 観光望遠鏡の開発～レンズと Android を組み合わせ、次世代に向けて何ができるか～”. 工学院大学卒業論文, 2014.
- (15) opencv.jp - OpenCV: ヒストグラム (Histograms) サンプルコード.
(引用日 2018 年 12 月 26 日.)
<http://opencv.jp/sample/histogram.html>.
- (16) L6470 取扱説明書.
http://akizukidenshi.com/download/ds/akizuki/AE-L6470_20190118.pdf.

図表目録

図 1.1	左) 観光望遠鏡が撤去される前の東京タワー展望台 右) 観光望遠鏡が撤去された後の東京タワー展望台 (1)	p4
図 1.2	訪日外国人数 観光庁統計データ (2)	p5
図 1.3	タブレット式観光望遠鏡の動作原理 (3)	p6
図 1.4	先行研究で開発された観光望遠鏡の CAD 全体図	p7
図 1.5	先行研究で開発されたアプリケーションの操作画面	p8
図 2.1	機体システムの概要	p11
図 2.2	使用するレンズモジュール	p12
図 2.3	Logicool 社製 C920t 内蔵の 1/4 インチ画素センサー	p12
図 2.4	ステッピングモーター の断面図 (5)	p13
図 2.5	RS 社 191-8299 (6)	p14
図 2.6	1 相励磁のパルス (7)	p15
図 2.7	2 相励磁のパルス (7)	p16
図 2.8	1-2 相励磁のパルス (7)	p16
図 2.9	マイクロステップ駆動とフルステップ駆動の電流パターンの比較 (8)(9)	p17
図 2.10	マイクロステップ駆動とフルステップ駆動の比較 (8)(9)	p17
図 2.11	Arduino UNO (10)	p18
図 2.12	TI 社 DRV8834 と秋月通商製 L6470 (11)(12)	p19
図 2.13	SPI バスでの接続例	p20
図 2.14	SPI バスでの複数デバイスの接続例	p20
図 2.15	デイジーチェーン接続の回路図 (13)	p21
図 2.16	自作基板	p22
図 2.17	回路図全体	p22
図 3.1	先行研究で製作された観光望遠鏡	p24
図 3.2	SolidWorks で製作した完成予想図面	p25
図 3.3	右柱完成図	p28
図 3.3.1	パーツ 1 × 2	p28
図 3.3.2	パーツ 2 × 1	p28
図 3.3.3	パーツ 3 × 1	p28
図 3.3.4	パーツ 4 × 1	p28
図 3.4	左柱完成図	p30
図 3.4.1	パーツ 1 × 1	p31

図 3.4.2	パーツ 2 × 1	p31
図 3.4.3	パーツ 3 × 2	p31
図 3.4.4	パーツ 4 × 1	p31
図 3.5	カメラボックス完成図	p32
図 3.5.1	パーツ 1 × 1	p33
図 3.5.2	パーツ 2 × 1	p33
図 3.5.3	パーツ 3 × 2	p33
図 3.6	上台の完成図	p34
図 3.7	支え柱の完成図	p35
図 3.8	中台完成図	p36
図 3.8.1	パーツ 1 × 1	p37
図 3.8.2	パーツ 2 × 1	p37
図 3.8.3	パーツ 3 × 2	p37
図 3.9	下台完成図	p38
図 3.10	基板の配置図	p39
図 3.11	カメラの上下の動作原理	p40
図 3.12	カメラの左右の動作原理 左)表 右)裏	p41
図 3.13	新たに完成した機体	P42
図 4.1	オートフォーカス方式の違い	p43
図 4.2	コントラスト計算の例 (15)	p44
図 4.3	人工的に画像をぼかした際のコントラストの変化(明るい画像)	p45
図 4.4	人工的に画像をぼかした際のコントラストの変化(暗い画像)	p45
図 4.5	モーターの回転方向イメージ図	p46
図 4.6	実験に用いた風景	p48
図 4.7	1 ステップ 2ms の場合のオートフォーカス	p48
図 4.8	1 ステップ 18ms の場合のオートフォーカス	p48
図 4.9	コントラストの値に応じたモーターの回転速度	p49
図 4.10	回転速度を可変式にした場合のオートフォーカス	p49
図 5.1	開発した観光望遠鏡	p50
図 5.2	開発した観光望遠鏡で撮影された景色	p50
図 5.3	本研究での観光望遠鏡システムの概要図	p51

表 2.1	ステッピングモーターの特徴 (4)	p13
表 2.2	使用するステッピングモーターの性能 (6)	p14
表 2.3	Arduino UNO (10)	p18
表 2.4	各モータードライバーの仕様 (11)(12)	p19
表 3.1	押し出し製法とキャスト製法の違い	p26
表 3.2	使用材料と個数	p27
表 5.1	先行研究で開発された観光望遠鏡の問題点	p50
表 5.2	問題点に対しての目標	p50

2018 年度(平成 30 年度)

創造工学セミナーⅡ

Final Report

プログラム付録

指導教授 金丸 隆志 准教授

チームメンバー

S5-15003 青木 俊輔

S5-15013 岩手山 彰也

S5-15015 大塚 裕斗

S5-15020 金井 聖

S5-15034 鈴木 智也

目次

第1章 最終的な Visual Studio のプログラム

P3～12

第2章 最終的な Arduino のプログラム

P13～22

第 1 章 最終的な Visual studio のプログラム

```
#include "stdafx.h"
#include <opencv2/opencv.hpp>
#include <windows.h>

#ifdef _DEBUG
#pragma comment(lib, "opencv_world341d.lib")
#else
#pragma comment(lib, "opencv_world341.lib")
#endif

HANDLE arduino;
bool Ret;

using namespace cv;
using namespace std;

int
main()
{
    //1.ポートをオープン
    arduino = CreateFileA("COM4", GENERIC_WRITE | GENERIC_READ, 0,
NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

    if (arduino == INVALID_HANDLE_VALUE) {
        printf("PORT COULD NOT OPEN¥n");
        system("PAUSE");
        exit(0);
    }
}
```

```
}
```

```
DWORD dwSendSize;
```

```
VideoCapture cap(0);
```

```
if (!cap.isOpened()) {
```

```
    cout << "camera open error" << endl;
```

```
}
```

```
Mat frame;
```

```
cap >> frame; //1 フレーム取得
```

```
Mat gray;//グレースケールを入れておくための mat
```

```
cvtColor(frame, gray, CV_BGR2GRAY);
```

```
double minVal, maxVal;
```

```
minMaxLoc(gray, &minVal, &maxVal); // 輝度値の最小と最大を見つけます.;
```

```
double c1 = (maxVal - minVal) / (maxVal + minVal);
```

```
//std::cout << "c1=" << c1 << std::endl;
```

```
int w = 1;
```

```
int sendCommand = 1;
```

```
int f1 = 1;
```

```

int f2 = 1;
int sendCommand2 = 1;

int g1 = 0;
int h1 = 0;
int i1 = 0;
int j1 = 0;
int g2 = 0;
int h2 = 0;
int i2 = 0;
int j2 = 0;

int v1 = 0;
int w1 = 0;
int v2 = 0;
int w2 = 0;

while (1)
{

    cap >> frame; //1 フレーム取得

    cvtColor(frame, gray, CV_BGR2GRAY);

    minMaxLoc(gray, &minVal, &maxVal); // 輝度値の最小と最大を見つけ
    ます.;

    double c2 = (maxVal - minVal) / (maxVal + minVal);

    double c3 = c2 - c1;
    //cout << "c1=" << c1 << ", c2=" << c2 << ", c3=" << c3 << endl;

    cout << c2 << endl;

```

```

if (c3 < 0) {
    w = -w;

    sendCommand = 1;

}

else {
    sendCommand = 0;

}

//DWORD dwsendSize;
string Y;//Y を宣言

DWORD lengthOfSent = 1; // 送信する文字数
if (w == 1) {
    Y = "u";//このコマンドを送信

}

else { // w== -1
    Y = "t";//このコマンドを送信

}

if (sendCommand == 1) {
    Ret    =    WriteFile(arduino,    Y.c_str(),    lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
    //cout << "コマンド送信" << ", w=" << w << endl;
    if (!Ret) {
        cout << "send faild" << endl;
        CloseHandle(arduino);
    }
}

```

```

        system("PAUSE");
        exit(0);
    }
    sendCommand = 0;
}

```

```
double
```

```

    fs1 = 0.93,
    fs2 = 0.90,
    fs3 = 0.86,
    fs4 = 0.80,
    fs5 = 0.76,
    fs6 = 0.70,
    fs7 = 0.55;

```

```

string X;//
if (c2 >= fs1) {
    X = "x";//このコマンドを送信
    f1 = 1;
}
else if ((c2 < fs1) && (c2 >= fs2)) {
    X = "y";//このコマンドを送信
    f1 = 2;
}

else if ((c2 < fs2) && (c2 >= fs3)) {
    X = "z";//このコマンドを送信
    f1 = 3;
}
else if ((c2 < fs3) && (c2 >= fs4)) {
    X = "&";//このコマンドを送信
    f1 = 4;
}

```

```

else if ((c2 < fs4) && (c2 >= fs5)) {
    X = "#";//このコマンドを送信
    f1 = 5;
}
else if ((c2 < fs5) && (c2 >= fs6)) {
    X = "%";//このコマンドを送信
    f1 = 6;
}
else if ((c2 < fs6) && (c2 >= fs7)) {
    X = "@";//このコマンドを送信
    f1 = 7;
}
else if (c2 < fs7) {
    X = "$";//このコマンドを送信
    f1 = 8;
}

if (f1 == f2) {
    sendCommand2 = 0;
}
else {
    sendCommand2 = 1;
}

if (sendCommand2 == 1) {
    Ret    =    WriteFile(arduino,    X.c_str(),    lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
    //cout << "コマンド送信 2" << ", x=" << X << endl;
    if (!Ret) {
        cout << "send faild" << endl;
        CloseHandle(arduino);
        system("PAUSE");
        exit(0);
    }
    sendCommand2 = 0;
}

```

```

}
c1 = c2;
f2 = f1;
imshow("取得画像", frame);
//imshow("ヒストグラム", frame);

```

```

string G;
string H;
string I;
string J;
string V;
string W;

```

```

if (GetAsyncKeyState(VK_UP)) { //↑ + sift キー
    cout << "↑↑↑" << endl;
    G = "g";
    g1 = 1;
}
else {
    G = "a";
    g1 = 0;
}

```

```

if (g1 != g2) {
    Ret = WriteFile(arduino, G.c_str(), lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
    cout << "コマンド送信 3" << ", control=" << G << endl;
    if (!Ret) {
        cout << "send faild" << endl;
        CloseHandle(arduino);
    }
}

```

```

        system("PAUSE");
        exit(0);
    }
}

if (GetAsyncKeyState(VK_DOWN)) {//↓ + sift キー
    cout << "↓↓↓" << endl;
    H = "h";
    h1 = 1;
}
else {
    H = "a";
    h1 = 0;
}

if (h1 != h2) {
    Ret = WriteFile(arduino, H.c_str(), lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
    cout << "コマンド送信 3" << ", control=" << H << endl;
    if (!Ret) {
        cout << "send faild" << endl;
        CloseHandle(arduino);
        system("PAUSE");
        exit(0);
    }
}

if (GetAsyncKeyState(VK_RIGHT)) {//→ + sift キー
    cout << "→→→" << endl;
    I = "i";
    i1 = 1;
}
else {

```



```

        I = "b";
        i1 = 0;
    }

    if (i1 != i2) {
        Ret    =    WriteFile(arduino,    I.c_str(),    lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
        cout << "コマンド送信 3" << ", control=" << I << endl;
        if (!Ret) {
            cout << "send faild" << endl;
            CloseHandle(arduino);
            system("PAUSE");
            exit(0);
        }
    }

    if (GetAsyncKeyState(VK_LEFT)) { //← z キー
        cout << "←←←" << endl;
        J = "j";
        j1 = 1;
    }
    else {
        J = "b";
        j1 = 0;
    }

    if (j1 != j2) {
        Ret    =    WriteFile(arduino,    J.c_str(),    lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
        cout << "コマンド送信 3" << ", control=" << J << endl;
        if (!Ret) {
            cout << "send faild" << endl;
            CloseHandle(arduino);
            system("PAUSE");

```

```

        exit(0);
    }
}

if (GetAsyncKeyState('Z')) { // x キー
    cout << "zoom in" << endl;
    V = "v";
    v1 = 1;
}
else {
    V = "s";
    v1 = 0;
}

if (v1 != v2) {
    Ret = WriteFile(arduino, V.c_str(), lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
    cout << "コマンド送信 3" << ", control=" << V << endl;
    if (!Ret) {
        cout << "send faild" << endl;
        CloseHandle(arduino);
        system("PAUSE");
        exit(0);
    }
}

if (GetAsyncKeyState('X')) { // → + sift キー
    cout << "zoom out" << endl;
    W = "w";
    w1 = 1;
}
else {
    W = "s";
    w1 = 0;
}
}

```

```

        if (w1 != w2) {
            Ret = WriteFile(arduino, W.c_str(), lengthOfSent,
&dwSendSize, NULL); //これが送信の信号
            cout << "コマンド送信 3" << ", control=" << I << endl;
            if (!Ret) {
                cout << "send faild" << endl;
                CloseHandle(arduino);
                system("PAUSE");
                exit(0);
            }
        }

        c1 = c2;
        f2 = f1;
        g2 = g1;
        h2 = h1;
        i2 = i1;
        j2 = j1;

        v2 = v1;
        w2 = w1;

        int key = cvWaitKey(1);
        if (key == 115) {
            break;
        }

    }

    return 1;
}

```

第二章 最終的な Arduino のプログラム

/*

/*

シリアルモニタ コマンド

L6470

- a 1 台目 回転停止
- b 2 台目 回転停止 cvRelease
- c 両方 回転停止
- d 1 台目 指定方向に連続回転
- e 2 台目 指定方向に連続回転
- f 両方 指定方向に連続回転
- g 1 台目 正転に設定
- h 1 台目 逆転に設定
- i 2 台目 正転に設定
- j 2 台目 逆転に設定
- k 1 台目 スピードアップ
- l 1 台目 スピードダウン
- m 2 台目 スピードアップ
- n 2 台目 スピードダウン
- o 回転速度 遅い
- p 回転速度 普通
- q 回転速度 早い

カメラ

- r フォーカス ストップ
- s ズーム ストップ
- t フォーカス 正転
- u フォーカス 逆転
- v ズーム 正転
- w ズーム 逆転

*/

```

#include <SPI.h>

//上下左右
#define PIN_SPI_MOSI 11
#define PIN_SPI_MISO 12
#define PIN_SPI_SCK 13
#define PIN_SPI_SS 10
#define PIN_BUSY 9
#define PIN_BUSY2 8
    int inputchar;
    int dia = 1;
    int dia2 = 1;
    int spd = 1000;
    int spd2 = 1000;

//カメラ
//int pl= 9;
#define STEPPIN1 7
#define DIRPIN1 6
#define STEPPIN2 5
#define DIRPIN2 4
//#define
int PulseWidth = 1; // 1 パルスの長さ msec
#define STEPS 200 // 1 回転あたりのステップ数
#define STEPNUM 1 // 何ステップずつ動かすか
#define DEBUG true
    int zoomState = 0;
    int focusState = 0;

```

```

void setup(){
//上下左右
    pinMode(MOSI, OUTPUT);
    pinMode(MISO, INPUT);
    pinMode(SCK, OUTPUT);
    pinMode(SS, OUTPUT);
    pinMode(PIN_BUSY, INPUT_PULLUP);
    pinMode(PIN_BUSY2, INPUT_PULLUP);

//カメラ
    pinMode(STEPPIN1, OUTPUT);
    pinMode(DIRPIN1, OUTPUT);
    pinMode(STEPPIN2, OUTPUT);
    pinMode(DIRPIN2, OUTPUT);

    SPI.begin();
    SPI.setDataMode(SPI_MODE3);
    SPI.setBitOrder(MSBFIRST);
    Serial.begin(115200);
    digitalWrite(PIN_SPI_SS, HIGH);

//1 台目
    L6470_setparam_acc(0x030); // [R, WS] 加 速 度 default 0x08A (12bit)
    (14.55*val+14.55[step/s^2])
    L6470_setparam_dec(0x030); // [R, WS] 減 速 度 default 0x08A (12bit)
    (14.55*val+14.55[step/s^2])
    L6470_setparam_maxspeed(0x2c); // [R, WR] 最 大 速 度 default 0x041 (10bit)
    (15.25*val+15.25[step/s])
    L6470_setparam_minspeed(0x001); // [R, WS] 最 小 速 度 default 0x000 (1+12bit)
    (0.238*val[step/s])
    L6470_setparam_fsspd(0x3FF); // [R, WR]μ ステップからフルステップへの切替点速度
    default 0x027 (10bit) (15.25*val+7.63[step/s])
    L6470_setparam_kvalhold(0xFF); // [R, WR] 停 止 時 励 磁 電 圧 default 0x29 (8bit)

```

```

(Vs[V]*val/256)
    L6470_setparam_kvalrun(0xFF); //[R, WR] 定速回転時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_kvalacc(0xFF); //[R, WR] 加速時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_kvaldec(0xFF); //[R, WR] 減速時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_stepmood(0x07); //ステップモード default 0x07 (1+3+1+3bit)

//2 台目
    L6470_setparam_acc2(0x030); //[R, WS] 加 速 度 default 0x08A (12bit)
(14.55*val+14.55[step/s^2])
    L6470_setparam_dec2(0x030); //[R, WS] 減 速 度 default 0x08A (12bit)
(14.55*val+14.55[step/s^2])
    L6470_setparam_maxspeed2(0x2c); //[R, WR] 最 大 速 度 default 0x041 (10bit)
(15.25*val+15.25[step/s])
    L6470_setparam_minspeed2(0x001); //[R, WS] 最 小 速 度 default 0x000 (1+12bit)
(0.238*val[step/s])
    L6470_setparam_fsspd2(0x3FF); //[R, WR]μ ステップからフルステップへの切替点速度
default 0x027 (10bit) (15.25*val+7.63[step/s])
    L6470_setparam_kvalhold2(0xFF); //[R, WR] 停 止 時 励 磁 電 圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_kvalrun2(0xFF); //[R, WR] 定速回転時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_kvalacc2(0xFF); //[R, WR] 加速時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_kvaldec2(0xFF); //[R, WR] 減速時励磁電圧 default 0x29 (8bit)
(Vs[V]*val/256)
    L6470_setparam_stepmood2(0x07); //ステップモード default 0x07 (1+3+1+3bit)
}

```

```

void loop(){

```

```

//カメラ

```

```

if (zoomState == 1)
    { digitalWrite(DIRPIN1, HIGH);
      digitalWrite(DIRPIN2, LOW);
    }
else if (zoomState == 2)
    {digitalWrite(DIRPIN1, LOW);
      digitalWrite(DIRPIN2, HIGH);
    }
if (focusState == 1)
    digitalWrite(DIRPIN2, HIGH);
else if (focusState == 2)
    digitalWrite(DIRPIN2, LOW);

if (zoomState != 0){  focusState = 0;
                      digitalWrite(STEPPIN1, HIGH);
                      digitalWrite(STEPPIN2, HIGH);
                      delay(2);
                      digitalWrite(STEPPIN1, LOW);
                      delay(2);
                      digitalWrite(STEPPIN1, HIGH);
                      digitalWrite(STEPPIN2, LOW);
                      delay(2);
                      digitalWrite(STEPPIN1, LOW);
                      delay(2);
                      }

if (focusState != 0){digitalWrite(STEPPIN2, HIGH);
                      delay( PulseWidth);
                      digitalWrite(STEPPIN2, LOW);
                      delay( PulseWidth);}

```



```

// シリアルポートより 1 文字読み込む
inputchar = Serial.read();

if(inputchar != -1 ){// 読み込んだデータが -1 以外の場合 以下の処理を行う
    switch(inputchar){
//L6470
// stop
        case 'a':
            L6470_softstop0();
            break;

        case 'b':
            L6470_softstop20();
            break;

        case 'c':
            L6470_softstop0();
            L6470_softstop20();
            break;

//run
        case 'd':
            L6470_run(dia,spd);
            break;

        case 'e':
            L6470_run2(dia2,spd2);
            break;

        case 'f':
            L6470_run(dia,spd);
            L6470_run2(dia2,spd2);
            break;
    }
}

```

```

//dia
case 'g':
    dia = 1;
    L6470_run(dia,spd);
    break;

case 'h':
    dia = 0;
    L6470_run(dia,spd);
    break;

case 'i':
    dia2 =1;
    L6470_run2(dia2,spd2);
    break;

case 'j':
    dia2 = 0;
    L6470_run2(dia2,spd2);
    break;

//speed
case 'k':
    spd += 1000;
    L6470_run(dia,spd);
    break;

case 'l':
    spd -= 1000;
    L6470_run(dia,spd);
    break;

case 'm':
    spd2 += 1000;
    L6470_run2(dia2,spd2);

```

```

        break;
case 'n':
    spd2 = 1000;
    L6470_run2(dia2,spd2);
    break;

case 'o':
    spd = 1000;
    spd2 = 1000;
    break;

case 'p':
    spd = 5000;
    spd2 = 5000;
    break;

case 'q':
    spd = 10000;
    spd2 = 10000;
    break;

//カメラ
case 'r':
    focusState = 0;
    break;

case 's':
    zoomState = 0;
    break;

case 't':
    focusState = 1;
    break;

```

```
case 'u':

    focusState = 2;

    break;

case 'v':
    zoomState = 1;
    break;

case 'w':
    zoomState = 2;
    break;

case 'x':
    PulseWidth = 18;

    break;

case 'y':

    PulseWidth = 10;

    break;

case 'z':
    PulseWidth = 6;
    break;

case '&':
    PulseWidth = 5;

    break;

case '#':
```

```

        // pl= 9;
        PulseWidth = 4;

        break;

    case '%':

        // pl= 20;
        PulseWidth = 3;
        break;

    case '@':

        PulseWidth = 2;
        break;

    case '$':

        PulseWidth = 1;
        break;

    }
} else { // 読み込むデータが無い場合は何もしない
    }
}

```


新たな観光望遠鏡システムの開発

指導教員 金丸 隆志 准教授

①S5-15003 青木 俊輔 ②S5-15013 岩手山 彰也

③S5-15015 大塚 裕斗 ④S5-15020 金井 聖 ⑤S5-15034 鈴木 智也

1. 緒言

観光地にあるコイン式観光望遠鏡は減少傾向にある。従来の観光望遠鏡は遠くの景色を拡大してみることができ、日常では得られない体験ができる。しかし、同時に1人しか使用できず、景色を共有することができない。近年 SNS 上で共感を得たいという行動欲求が広がりつつある。従来の観光望遠鏡は、現代の観光客の要求を満たしていないと言える。

しかし、訪日外国人観光客が増加している傾向から潜在的な需要は大きいと考えられる。そこで、スマートフォンなどの端末と観光望遠鏡を組み合わせることで、現代の観光客のニーズに合った新たな観光望遠鏡システムの開発を目指す。

2. 先行研究の観光望遠鏡

文献(2)の観光望遠鏡は、タブレットを用いて望遠鏡の操作とレンズモジュールからの景色を眺めることができる。動作原理を図1に示す。

文献(2)の観光望遠鏡は、画面上にある上下左右移動、ズーム、フォーカス調節用のボタンを押すことで操作できる。その情報が無線 LAN を通して最終的に Arduino UNO に送られ、望遠鏡に設置されたモーターを制御する。映し出された景色も同様に、無線 LAN を通してタブレットまで送られる。これが先行研究で作成された観光望遠鏡の動作原理である。

本研究では、この動作原理をもとに新たな観光望遠鏡を製作した。

3. 新たな観光望遠鏡システムの製作

観光望遠鏡は数キロから数百キロメートル先の景色を見るため、小さな振動やガタつきが映像に影響をもたらす。そこで、機械的安定性の高い機体を製作した(図2左)。機体は、アクリル板を加工して製作した。サイズは 160x160x160mm と片手で持ち歩けるほどの大きさとなっている。太い支えや、搭載される基板を低い位置に設置することでガタつきのない頑丈な機体となっている。

上下左右の駆動にはステッピングモーターを使用している。1/128 マイクロステップ駆動方式を取り入れることで、スムーズで低振動な回転を実現した。

私たちが製作した観光望遠鏡の動作原理を図3に示す。

図1との大きな違いとして操作性向上のためオートフォーカス機能が追加されたことが挙げられる。使用者はパソコンの十字キー、ズーム調整用のボタンを押すことで、上下左右、ズームの調節作業を行う。フォーカスの調節はパソコン内で動作しているプログラムが取得した映像をもとに自動で調節を行う。フォーカスの調節は観測地を変えるたびに必要となるためこの機能は使用者の負担を軽減させることができる。

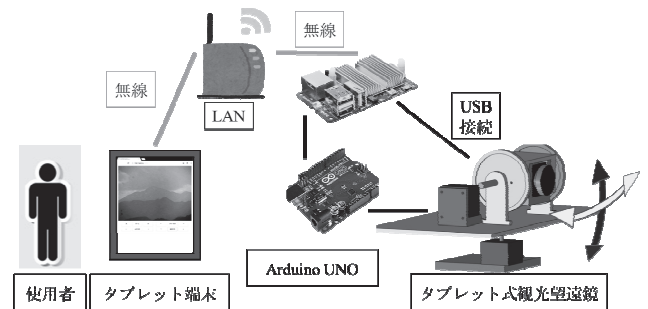


図1 先行研究の観光望遠鏡の動作原理

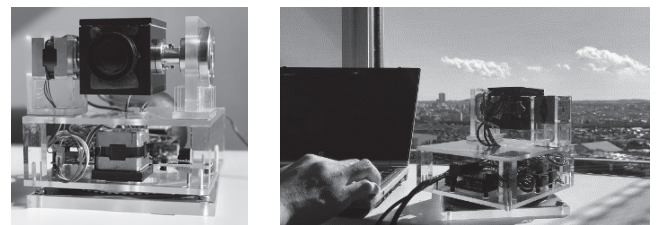


図2 左) 新たに製作した観光望遠鏡

右) 新たに製作した観光望遠鏡での撮影風景

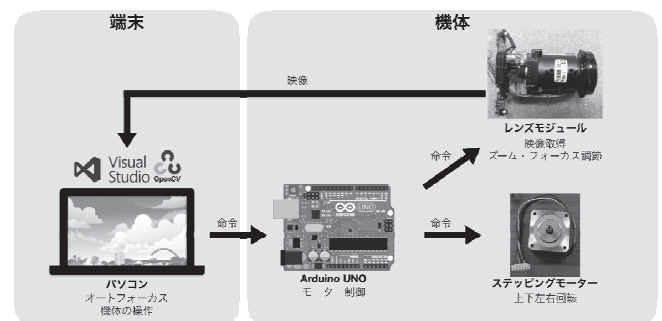


図3 新たに製作した観光望遠鏡の動作原理

4. 結言

本研究で製作した観光望遠鏡システムは、操作性に優れ、望遠鏡として快適に使用することができるようになった。今後の課題として、観光客のニーズに合ったより新規性のある機能の実装が求められる。

<参考文献>

- (1) 吉野晃平・他5名, “Android 観光望遠鏡の開発～レンズと Android を組み合わせ、次世代に向けて何ができるか～,” 工学院大学卒業論文, 2014.
- (2) 吉野晃平 “タブレット端末を用いた観光望遠鏡システムの開発,” 工学院大学修士論文, 2016.