

DM

**Web-Based Telescope Control Interface  
and Automation Architecture  
for the University of Madeira Astronomical Observatory**

MASTER DISSERTATION

**Igor da Silva Vasconcelos**  
MASTER IN INFORMATICS ENGINEERING



UNIVERSIDADE da MADEIRA

*A Nossa Universidade*

[www.uma.pt](http://www.uma.pt)

September | 2025

**Web-Based Telescope Control Interface  
and Automation Architecture  
for the University of Madeira Astronomical Observatory**

MASTER DISSERTATION

**Igor da Silva Vasconcelos**  
MASTER IN INFORMATICS ENGINEERING

SUPERVISOR  
Marko Radeta



FACULDADE DE CIÊNCIAS EXATAS E DA ENGENHARIA

MASTER OF SCIENCE DEGREE IN INFORMATICS ENGINEERING

# Web-Based Telescope Control Interface and Automation Architecture for the University of Madeira Astronomical Observatory

Author:

Igor da Silva Vasconcelos

Supervised by:

Professor Dr. Marko Radeta

Constitution of the public examination jury:

Professor Dr. David Sardinha Andrade de Aveiro, President

Professor Dr. Mary Alejandra Luiz Barreto, Member

Professor Dr. Marko Radeta, Member

**Friday 5<sup>th</sup> December, 2025**

# Resumo

Desde tempos pré-históricos, a humanidade tem-se fascinado com o céu noturno, o que levou à criação de observatórios astronómicos terrestres e extraterrestres que proporcionam conhecimentos valiosos sobre o universo observável. No entanto, os observatórios astronómicos são tipicamente dispendiosos de construir e manter, estando muitas vezes localizados em zonas remotas que limitam o acesso do público. Além disso, enfrentam desafios burocráticos que dificultam a obtenção de tempo de observação. Esta dissertação contribui com o desenho inicial de um sistema de controlo remoto do OAUMa - Observatório Astronómico da Universidade da Madeira. A novidade desta dissertação assenta em: (i) Proposta de arquitetura para o sistema de controlo remoto do OAUMa; (ii) Três testes relacionados com a comunicação com o telescópio do OAUMa; (iii) Uma interface web para o controlo do telescópio do OAUMa; e (iv) Avaliação do desempenho do sistema através de testes de usabilidade e feedback de um astrónomo profissional e do público amador. A solução proposta combina tecnologias de baixo custo, baseadas em microcontroladores (MCUs), com protocolos de comunicação normalizados, tal como o Instrument Neutral Distributed Interface (INDI), permitindo uma integração flexível entre diferentes dispositivos astronómicos. A abordagem apresentada nesta dissertação serve também como um plano inicial de como os observatórios astronómicos podem ser operados remotamente, de forma económica, escalável e adequada a uma vasta gama de observatórios, incluindo instalações amadoras e educativas.

**Keywords:** Astronomia Observacional · Internet das Coisas · Controlo Remoto · Sistemas Microeletromecânicos (MEMS) · Automação de Telescópios · Observações Remotas

# Abstract

Since prehistoric times, humankind has been fascinated by the night sky, leading to the creation of terrestrial and extraterrestrial astronomical observatories that provide valuable insights into the observable universe. However, astronomical observatories are typically expensive to construct and maintain, often situated in remote areas that limit public accessibility. Additionally, they are burdened by bureaucratic challenges, making it difficult to secure observation time. This dissertation contributes with the initial design of a remote control system for the OAUMa - Observatório Astronómico da Universidade da Madeira. The novelty of the dissertation lies in: (i) Proposed architecture for the OAUMa remote control system; (ii) Three tests regarding the communication with the OAUMa telescope; (iii) A web-based interface for controlling the OAUMa telescope; and (iv) Evaluation of the system's performance through usability and feedback from a professional astronomer and amateur audience. The proposed solution combines low-cost technologies, based on microcontrollers (MCUs), with standardized communication protocols, such as the Instrument Neutral Distributed Interface (INDI), enabling flexible integration between different astronomical devices. The approach presented in this dissertation serves also as an initial blueprint on how the astronomical observatories may be remote operated, cost-effective, scalable, and suitable for a wide range of observatories, including amateur and educational facilities.

**Keywords:** Observational Astronomy · Internet of Things · Remote Control · Micro-Electromechanical Systems (MEMS) · Telescope Automation · Remote Observations

# Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Dr. Marko Radeta, for his continuous support, expert guidance and encouragement throughout the development of this dissertation. His experience, patience and advice were really important to the successful completion of this work.

I also wish to thank the team of colleagues and collaborators who contributed their ideas, technical expertise and effort to the automation of the observatory. In particular, I am grateful to Wave-Labs for providing the necessary equipment and financial support, without which this project would not have been possible.

I extend my heartfelt thanks to the university students who actively participated in the project, generously giving their time, energy and enthusiasm. Your commitment played a vital role in the development, testing and validation of the system.

My sincere thanks also go to the university, for providing the infrastructure and facilities that enabled the installation of the observatory and the execution of this work.

I would further like to acknowledge the valuable support of the professional astronomer who dedicated his time to review the system, provide constructive feedback and share his experience. His contributions ensured the project remained grounded in real-world astronomical practices.

Finally, I am also deeply grateful to my family and friends for their support throughout this journey. Their encouragement, patience and belief in me provided the strength I needed to overcome challenges and stay motivated. Their presence and understanding played a meaningful role in helping me reach this milestone.

# Table of Contents

<b>List of Figures</b> .....	<b>vi</b>
<b>List of Tables</b> .....	<b>vii</b>
<b>1 Introduction</b> .....	<b>1</b>
<b>1.1 On potential of Madeira for Astronomical Observations</b> .....	<b>1</b>
<b>1.2 Dissertation Contributions</b> .....	<b>2</b>
<b>1.3 Dissertation Validation</b> .....	<b>2</b>
<b>1.4 Dissertation Structure</b> .....	<b>2</b>
<b>2 Literature Review</b> .....	<b>4</b>
<b>2.1 Traditional Astronomical Observatory Setups</b> .....	<b>4</b>
<b>2.2 Automation of the Dome Rotation</b> .....	<b>5</b>
<b>2.3 Automation of Telescope Control</b> .....	<b>6</b>
<b>2.4 Automation of Environmental Monitoring</b> .....	<b>7</b>
<b>2.5 Automation of Alerting Mechanisms</b> .....	<b>9</b>
<b>2.6 Automation using Imaging Technologies</b> .....	<b>10</b>
<b>2.7 Automation Protocols</b> .....	<b>13</b>
<b>2.8 Automation User Interfaces</b> .....	<b>14</b>
<b>2.9 Automation by the Communities</b> .....	<b>15</b>
<b>2.10 Automation using Affordable Technologies</b> .....	<b>15</b>
<b>2.11 Design Decisions for the Proposed System</b> .....	<b>17</b>
<b>3 Methodology</b> .....	<b>19</b>
<b>3.1 Initial Status of OAUMa</b> .....	<b>19</b>
<b>3.2 Proposed OAUMa Automation Architecture</b> .....	<b>21</b>
<b>3.2.1 Master MCU</b> .....	<b>22</b>
<b>3.2.2 Shutters MCU</b> .....	<b>23</b>
<b>3.2.3 Telescope MCU</b> .....	<b>23</b>
<b>3.2.4 All-Sky MCU</b> .....	<b>24</b>
<b>3.2.5 Power Distribution and Network Communication</b> .....	<b>25</b>
<b>3.2.6 Workflow</b> .....	<b>27</b>

<b>3.3</b>	<b>Telescope Automation</b> .....	<b>28</b>
<b>3.3.1</b>	<b>Test 1: Stellarium App</b> .....	<b>28</b>
<b>3.3.2</b>	<b>Test 2: Meade Telescope Serial Command Protocol</b> .....	<b>30</b>
<b>3.3.3</b>	<b>Test 3: INDI</b> .....	<b>32</b>
<b>3.4</b>	<b>Proposed OAUMa Telescope Control Interface</b> .....	<b>35</b>
<b>3.4.1</b>	<b>Interface Data Flow</b> .....	<b>36</b>
<b>3.4.2</b>	<b>GUI</b> .....	<b>39</b>
<b>4</b>	<b>Results</b> .....	<b>45</b>
<b>4.1</b>	<b>Evaluation Protocol</b> .....	<b>45</b>
<b>4.2</b>	<b>Feedback</b> .....	<b>47</b>
<b>4.2.1</b>	<b>User Suggestions and Resulting Improvements</b> .....	<b>47</b>
<b>4.2.2</b>	<b>SUS Results</b> .....	<b>51</b>
<b>4.2.3</b>	<b>Astronomer Feedback</b> .....	<b>51</b>
<b>5</b>	<b>Discussion</b> .....	<b>53</b>
<b>5.1</b>	<b>Comparison with the Literature</b> .....	<b>53</b>
<b>5.2</b>	<b>Study limitations and Room for Improvement</b> .....	<b>54</b>
<b>5.3</b>	<b>Future Work</b> .....	<b>55</b>
<b>6</b>	<b>Conclusion</b> .....	<b>57</b>
	<b>References</b> .....	<b>58</b>

## List of Figures

1	OAUMa - Observatório Astronómico da Universidade da Madeira . . . . .	20
2	Proposed schematic of the astronomical observatory control system. This setup assumes that there are an upper shutter and a lower shutter, and that the dome has motors installed previously . . . . .	25
3	Observation night workflow . . . . .	29
4	INDI Architecture Overview. Source: <a href="https://indilib.org/">https://indilib.org/</a> . . . . .	33
5	INDI Communication Flow. Source: <a href="https://docs.indilib.org/quick-guide/">https://docs.indilib.org/quick-guide/</a> . . . . .	34
6	System data flow overview . . . . .	37
7	Proposed telescope control system's home page . . . . .	39
8	Proposed telescope control system's control page . . . . .	40
9	Automated slew section . . . . .	41
10	Manual control section . . . . .	42
11	Proposed telescope control system's settings page . . . . .	43
12	The terms "slew" and "position" were changed to "move" and "coordinates", respectively. Additionally, a label for the object coordinates search was added for better clarity . . . . .	48
13	Explanatory tooltips were added for technical terms to enhance user understanding. . . . .	48
14	Improved error feedback was implemented for incorrect object name searches . . . . .	49
15	Improved visual clarification of keyboard support for the manual control pad . . . . .	50
16	Newly added user guide page to assist first-time users . . . . .	50

## List of Tables

1	Comparison of CCD, CMOS/sCMOS, and EMCCD Sensors in Automated Observatories	12
2	Summary of Observatory Equipment and Dome Specifications . . . . .	20

## List of Acronyms

<b>AI</b>	Artificial Intelligence
<b>AP</b>	Access Point
<b>API</b>	Application Programming Interface
<b>ASCOM</b>	Astronomy Common Object Model
<b>ASTCom</b>	Astronomic Communication Library
<b>BLE</b>	Bluetooth Low Energy
<b>CCD</b>	Charge-Coupled Device
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>DC</b>	Direct Current
<b>DEC</b>	Declination
<b>DIY</b>	Do-It-Yourself
<b>DNS</b>	Domain Name System
<b>EIRM</b>	Eliminating Image Rotation Mechanism
<b>EMCCD</b>	Electron-Multiplying CCD
<b>FO</b>	Fractional-Order
<b>FWHM</b>	Full Width at Half Maximum
<b>GUI</b>	Graphical User Interface
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTTPS</b>	Hypertext Transfer Protocol Secure
<b>ICAT</b>	IEEC Calibration and Analysis Tool
<b>ID-T</b>	Integral Derivative-Tilted
<b>IMU</b>	Inertial Measurement Unit
<b>INDI</b>	Instrument Neutral Distributed Interface

**IP** Internet Protocol

**IoT** Internet of Things

**LLM** Large Language Model

**MCU** Microcontroller Unit

**MEMS** Micro-Electromechanical Systems

**MQTT** Message Queuing Telemetry Transport

**OAUMa** Observatório Astronómico da Universidade da Madeira

**OCS** Observatory Control Systems

**PID** Proportional-Integral-Derivative

**POA** Peafowl Optimization Algorithm

**PSO** Particle Swarm Optimization

**RA** Right Ascension

**REST** Representational State Transfer

**RF** Radio Frequency

**RPI** Raspberry Pi

**SIMBAD** Set of Identifications, Measurements, and Bibliography for Astronomical Data

**SNR** Signal-to-Noise Ratio

**SUS** System Usability Scale

**TCP/IP** Transmission Control Protocol/Internet Protocol

**TCS** Telescope Control Systems

**TID** Tilt Integral-Derivative

**UPS** Uninterruptible Power Supply

**USB** Universal Serial Bus

**UTC** Coordinated Universal Time

**VDOM** Virtual Document Object Model

**VPN** Virtual Private Network

**sCMOS** Scientific Complementary Metal-Oxide-Semiconductor

# 1 Introduction

Observable universe encompasses the radius of 14.2 billion parsecs (about 46.3 billion light years) [1] while its actual size remains unknown. Indeed, humankind's fascination with the night sky has inspired centuries of exploration and discoveries [2]. Driven by curiosity about the celestial phenomena, humans developed tools and structures to observe the heavens, ranging from (terrestrial) optical, radio to (space) X, gamma ray observatories [3], marking significant advancements in science and technology and bringing the cosmos closer. Today, terrestrial astronomical observatories offer access to powerful telescopes and instruments that can unlock the mysteries of the cosmos such as its size and topology [4]. However, most public observatories require bureaucratic burdens and complex tender forms for allocating the observation times to the external visitors. Moreover, they typically consist from complex and costly equipment, needing extensive domain expertise and knowledge for their efficient operation. These facilities are often isolated to secure the dark skies for observation and reduce light pollution [5]. This results in high travel costs and limited accessibility to the general public. Removing these burdens can provide educational impact, facilitating overall exploration and discovery.

## 1.1 On potential of Madeira for Astronomical Observations

Interest in Madeira as a location for astronomy increased during the 20th century, with the establishment of new observatories and research projects [6]. Astronomical observation in Madeira first started in the 15th century when the island became an important place for Portuguese maritime exploration [6]. With its clear skies and other favorable atmospheric conditions, Madeira soon gained the attention of astronomers of all nationalities [6]. The Funchal Astronomical Observatory was inaugurated in the 18th century, being the first astronomical observatory in Portugal [6]. The island has provided a location for many different astronomical observations and studies over the centuries, which have included stellar position measurements, atmospheric phenomena and studies of the night sky [6]. Despite many challenges, such as Saharan dust and political difficulties, Madeira remains a very relevant site for such studies [6]. New astronomical technologies have been tested on the island, where sky monitoring observations have taken place [6]. More recently, with the development of new technologies and projects in the field, consolidation of the local astronomical community has taken place [6].

## 1.2 Dissertation Contributions

This dissertation contributes to the initial design and validation of a telescope remote control system for controlling the OAUMa. The approach presented in this dissertation focuses on a general architecture of the remote astronomical observatory which is cost-effective, scalable, being generalized to a wide range of observatories, including amateur and educational facilities. This work prioritizes the Internet of Things (IoT)-based remote control, which allows the setup and operation of any telescope and dome, targeted for small and medium-sized observatories with limited budgets. Objectives of the dissertation are:

- Proposed architecture for the OAUMa remote control system.
- Three tests regarding the communication with the OAUMa telescope.
- A web-based interface for controlling the OAUMa telescope.
- Evaluation of the system’s performance through usability and feedback from a professional astronomer and amateur audience.

## 1.3 Dissertation Validation

Following the iterative SCRUM software development methodology, the dissertation is developed within the multidisciplinary team of Wave Labs including a professional astronomer from abroad who is the end user and validating the proposed interface for remote control of the telescope. Iterative feedback from the domain expert allowed the identification of initial requirements as well as the end goals in the remote access to the telescope of the OAUMa. Proposed system has also been evaluated using the System Usability Scale (SUS) with 5 people from the amateur audience.

## 1.4 Dissertation Structure

The document is divided in six main parts: (i) introduction, where a brief context of the challenge in remote observational astronomy is explained; (ii) literature review, where the state-of-the-art systems performing the astronomical observations are explored; (iii) methodology, where the proposed architecture for the OAUMa remote control system and an interface to remotely control the OAUMa telescope are detailed through 3 tests; (iv) results, where the obtained feedback on usability aspects from the users and the professional astronomer is shown; (v) discussion,

comparisons against the literature in remote astronomical observatories including the limitations of the study and planned future work; and (vi) conclusion, showcasing the overall contribution of the dissertation in the revival of astronomical observatories in Madeira.

## 2 Literature Review

The increasing interest in people in exploring space and deep sky has created the need for more affordable and accessible ways to observe the universe. There are solutions that can be found when combining low-cost automation approaches, remote observatory management and control, and open-source technologies. This literature review aims to identify current solutions and their limitations, focusing on practical and scalable approaches to observatory automation. Throughout the literature review, Google Scholar <sup>1</sup> was used as the primary search tool, using keywords such as "low-cost observatory control systems", "remote telescope control software", "observatory shutter control system" and "remote observatory control system". Studies were selected based on their relevance to low-cost automation and their applicability to the proposed system. In addition, existing Large Language Models (LLM) models were used to find and summarize the relevant works such as Phind <sup>2</sup>, Consensus <sup>3</sup> and NotebookLM <sup>4</sup> because they facilitate scientific and scholar search in astronomy and information extraction.

### 2.1 Traditional Astronomical Observatory Setups

Astronomical observatories are mainly composed by two main components: the dome and the telescope [7–11]. The dome protects the equipment inside it from weather conditions, such as rain, wind and temperature changes, while the telescope serves as the main equipment for observations. Additionally, the dome helps reduce light pollution, the excessive artificial lighting that brightens the night sky affecting the telescope's imaging quality and performance.

Typically, the telescope is placed at the center of the dome, allowing the telescope to be able to point at maximum altitude and azimuth axes values. The telescope looks at the sky through the shutters, an aperture present in the dome that provides the telescope with an unobstructed view of the sky while maintaining the benefits of the dome enclosure.

One of the most relevant operational requirements linked to any observatory concerns the alignment between shutters and telescope pointing. While the telescope tracks an object, the dome must track and rotate to keep the shutters in the right direction of the telescope's optical axis. Otherwise, the line of view will be occluded by the dome structure, which will affect the observations.

---

<sup>1</sup><https://scholar.google.com/>

<sup>2</sup><https://www.phind.com/>

<sup>3</sup><https://consensus.app/>

<sup>4</sup><https://notebooklm.google.com/>

## 2.2 Automation of the Dome Rotation

Effective observations require precise alignment between the dome and the shutters. To achieve synchronization and coordination between both parts, efficient robotic operation of astronomical observatories is necessary, allowing for remote and unattended observations [9]. Nevertheless, even in low-cost control systems, precision and reliability remain important considerations [12].

Rotation of domes in astronomical observatories is usually done by electric motors [7,9,10,13] integrated into a centralized control system. In such systems, constant communication with the telescope is established to ensure the shutter aligns precisely with it [9]. This issue of synchronization is so relevant that in case the shutters do not correspond with the telescope, then the view of the sky will be partially or completely obscured, which can affect the quality of observation and its data.

The selection of the motor is dependent on the precision, torque and mechanical configuration adopted in the rotation. Stepper motors are very common since they can perform angular motions, enabling good positional control [7,10]. DC motors are generally applied when one requires rotation to be smoother and faster, although sometimes compromising accuracy [10,11,13,14].

To ensure the dome correctly aligns, encoders are frequently integrated into the rotational system of the dome [7,15–18]. These encoders provide high-resolution feedback on the dome's position, which is compared to the desired azimuth [7,17]. This system allows for error correction in the event of slippage or inertia-based overshooting.

IoT-based control architectures are being applied to dome rotation systems as well. These systems utilize networks of MCUs that offer more cost-effective solutions than industrial controllers [7,10–12,14,15,19]. These type of systems allow individual components, such as motors, sensors and limit switches, to be independently controlled, improving modularity and fault tolerance. These applications mostly rely on MCU platforms like Arduino and Raspberry Pi (RPI) [7,10,17]. The availability of hardware compatibility and strong support from their active communities contributes to the popularity of these units in amateur and professional observatories. For systems like these, the dome usually rotates by use of relays that are triggered on and off by digital signals originating from an MCU [17]. These relays act like switches to apply or remove motor power according to the logic for the control of such movement. Additionally, MCUs also often interface

with limit switches to provide indications of the home position of a dome or when rotation should cease [7].

This dissertation assumes the reader is interested in having the remote access to the telescope and assumes that the astronomical observatory has already an automated dome rotation and synchronization with the telescope. Nevertheless, it will still provide a system architecture on how to use the basic IoT devices for the automation of the dome rotation.

### 2.3 Automation of Telescope Control

The automation of telescope control systems has been a long-term goal in modern astronomy, enabling improved operational efficiency and the possibility of fully robotic observatories. Achieving precise tracking and pointing for observations requires a well structured relation between hardware and software [11,13].

Telescope axes, right ascension (RA) and declination (DEC), are usually controlled by DC motors [7,13] or stepper motors [8,14], depending on the required balance of torque, precision and cost. As a specific example, the 50/80 cm Schmidt telescope employs a twin-motor drive mechanism with two torque DC motors designed to ensure that there is no backlash in reversing the direction of motion [13]. Gearboxes, usually configured as three-stage gear trains, have been employed in some designs to provide the necessary speed reduction for smooth astronomical tracking [13]. Other configurations include two-axis systems driven by stepper motors in conjunction with timing belts for both speed reduction and finer positional precision [14].

The accuracy of telescope motion depends on feedback mechanisms, where both absolute and incremental encoders play an important role in measuring angular position and velocity [13]. These sensors form the foundation of control strategies used in Telescope Control Systems (TCS), which are the core software and hardware frameworks responsible for pointing, tracking, and calibration of telescopes [20,21].

Pointing and tracking accuracy are vital for scientific observations, since even small deviations can affect data quality. Control methods, such as Proportional-Integral (PI) controllers, are used for precise velocity regulation in tracking systems [13]. Another approach incorporates mount error modeling, such as the Tpoint system, to correct systematic mechanical misalignments and thereby

improve pointing accuracy [22]. Some designs use twin-motor differential torque drives to eliminate backlash when reversing motion, further improving performance [13].

Techniques using artificial intelligence (AI) methods for control have also been investigated to optimize telescope tracking. In that regard, controllers of the type Fractional-Order (FO), such as Tilt Integral-Derivative (TID) and Integral Derivative-Tilted (ID-T), were designed in MATLAB to minimize the error between the desired and the reference of the tracking signal [7]. Nature-inspired algorithms such as the Peafowl Optimization Algorithm (POA) and Particle Swarm Optimization (PSO) have been deployed to find appropriate controller gains and have yielded improvements over traditional controller tuning methods [7].

Automated focusing systems adjust the telescope optics, like a secondary mirror or an attached camera, until stars appear sharp. In order to achieve this, a sequence of images at different focus levels is created and then a curve to determine the optimal setting is applied [23]. In addition, some systems compensate for temperature-induced focus drift through predefined thermal models [23].

After the initial slew to the desired target, pointing accuracy is rectified by capturing an image, which allows to get the positional offset from the intended coordinates, and applying movements to correct and center the object. This approach is a core feature of modern robotic observatories [23, 24].

For the purpose of long-exposure astrophotography, auto-guiding can be used to correct small tracking errors in real time. Guiding can be performed using either the science camera itself or a separate dedicated guiding camera, depending on the system architecture [15, 23].

In this dissertation, the focus will be on providing the remote access for operating the telescope through three tests which will be presented hereinafter.

## 2.4 Automation of Environmental Monitoring

Astronomical observations require favorable weather conditions. Environmental monitoring is a fundamental component in the operation of astronomical observatories, especially those that depend on remote or robotic control. The main goals of environmental monitoring are to protect equipment, ensure that the observations have the best conditions and to allow astronomers to know the current weather status at the observatory, letting them optimize their time and logistics.

This is achieved by tracking environmental variables and hardware statuses through sensors and software.

Environmental monitoring at observatories normally goes over a wide group of meteorological and atmospheric parameters. These parameters include, among others, ambient temperature, humidity, wind speed and direction, precipitation (rain, snow and hail), fog, atmospheric pressure, ground temperature and even dust particles concentration [7, 9, 10, 18, 25–32]. Weather stations are the foundation of such systems, providing continuous measurements of key variables. These values enable remote and autonomous decision-making, allowing observatory operators to assess real-time conditions and respond appropriately, such as whether to open or close the shutters or to initiate shutdown procedures.

Apart from standard weather metrics, there are specialized sensors for more distinct monitoring. For example, cloud detection systems may be employed to check on cloud coverage, which often proves to be a determining factor in whether observations can proceed [18]. The early detection of cloud cover saves observatories from wasting valuable observing time and equipment wear. Some systems also monitor the temperature of the sky, providing additional data that can be used to assess sky transparency and quality [9]. Other environmental hazards, such as electrical storms, are monitored through storm sensors that detect changes in the local electrostatic field; such sensors may automatically initiate shutdown procedures to protect against possible lightning or power surges [9]. These monitoring systems taken together are essential for giving a correct estimate of the number of usable observing nights annually [27].

An improvement that makes environmental monitoring better, specifically regarding cloud and sky analysis, is the use of all-sky cameras [19, 27, 28, 31, 33], which capture panoramic images of the entire sky. These cameras provide information on cloud cover, meteor activity and other atmospheric phenomena [8, 9, 11]. The images are processed to generate cloud maps and identify clear regions of the sky, thereby supporting real-time decision-making for telescope pointing and scheduling [9]. Furthermore, all-sky cameras offer early warnings of approaching clouds and offer continuous evaluation of sky conditions [9, 11]. Nightly logs of sky brightness, as captured by these equipments, help distinguish between clear and cloudy nights, providing a valuable historical record for site characterization [29].

The data collected by the different sensors is normally managed by MCUs, which can store, process and even send the data to central servers to further analysis [28]. Automated algorithms, including traditional image processing and advanced machine learning techniques, are used to classify sky conditions and detect clouds with high accuracy [28, 33–35]. The evolution of the technology has allowed for the creation of better and more reliable environmental monitoring devices, simplifying the data collection process.

Throughout this dissertation, the weather control systems will be mentioned as an optional component of the remotely astronomical observatory system, allowing the reader to use these additional data to facilitate decision-making for performing the observations

## 2.5 Automation of Alerting Mechanisms

Another aspect in literature that deals with remote observations is safety. For example, during emergency situations, redundant shutter-closing mechanisms diminish the risk of single points of failure, while monitoring systems identify potential faults or irregularities, triggering timely alerts to ensure system reliability [9, 16, 25].

Sensors may be incorporated to limit excessive shutter movement, in addition to power outage protection mechanisms to ensure closure of the shutter in case of energy loss [9]. For example, Piezo velocity sensors can detect unexpected failures of the mechanism and if the shutter falls, the sensors detect a sudden increase in velocity, which can then send signals to activate a safety mechanism of the shutter <sup>5</sup>.

Other observatory components can also be tracked, such as motor temperature and current, and controller voltage to make sure they are working properly [25]. Some systems can implement a "Heartbeat" protocol, to check the state of the other components [18] or even have an automated auto-recovery from hardware faults and control system issues [36]. Additionally, power supply systems, including uninterruptible power supply (UPS) units, are also monitored by tracking battery status to ensure consistent energy delivery [9].

By integrating these hardware and software components into a unified system, astronomical observatories can achieve efficient and secure operations, even in remote and unattended environ-

---

<sup>5</sup>[https://www.technoarete.org/common\\_abstract/pdf/IJERMCE/v4/i3/Ext\\_49615.pdf](https://www.technoarete.org/common_abstract/pdf/IJERMCE/v4/i3/Ext_49615.pdf)

ments. Continuous monitoring of equipment status is essential for maximizing observatory productivity while safeguarding critical equipment.

Since the thesis provides the first step in remotely controlling the telescope, it assumes that the reader gains the practical knowledge on safety topic as well for the future optimization of the astronomical observatory

## 2.6 Automation using Imaging Technologies

Automated and remotely-operated observatories rely a lot on imaging subsystems, such as cameras, filter wheels, focusers and the associated image-processing pipelines, to gather data and to provide control for pointing, guiding and decision making. Modern automation combines imaging hardware with software frameworks and algorithms that range from low-level exposure control to higher-level image-based decision systems (scheduling, safety, target selection).

Camera systems have two partially distinct roles in automated observatories: (i) science imaging, where the camera is the primary data-gathering instrument and long exposures, sensitivity and calibration matter most; and (ii) control/feedback imaging, where the camera provides near real-time information used to guide, focus or verify mechanical actions (e.g., plate solving for pointing verification, autoguiding for tracking or cloud/sky assessment). Many modern observatories combine both roles by using a main scientific camera in conjunction with a smaller guide camera or auxiliary wide-field cameras, while software orchestrates exposures, guiding corrections, and calibrations within an automated schedule [20, 23].

Charge-Coupled Device (CCD) sensors and their high-sensitivity variants, such as Electron-Multiplying CCDs (EMCCDs), remain a mainstay for scientific imaging where high quantum efficiency, excellent charge transfer performance and low read noise are required. CCDs are typically cooled, frame-transfer or full-frame devices used for long exposures, photometry and spectroscopy. They are commonly integrated into automation frameworks where exposure sequencing, calibration frame acquisition (bias, dark, flats), and data pipelines are scheduled and executed without user intervention [20, 37], but are also used in custom systems [38]. CCDs usually impose cadence and data-flow constraints (large files, slower readout), and therefore automation frameworks must manage readout, thermal stabilization, and calibration scheduling (bias/dark/flat sequences) to preserve photometric fidelity and to support unattended operation [23]. EMCCDs are used where

very low light detection is crucial because of their gain stage and superior effective Signal-to-Noise Ratio (SNR) under some regimes [37].

Complementary Metal-Oxide-Semiconductor (CMOS) sensors and scientific CMOS (sCMOS) have matured to the point where they are widely used in astronomy. Their main advantages are fast readout, high frame rates, low power, and often lower system cost. These characteristics make CMOS ideal for real-time control tasks such as rapid target acquisition, object detection, space-debris tracking and panoramic monitoring. In automation contexts, high-frame-rate CMOS cameras feed image-processing modules (object detection, centroiding or astrometry) that can drive real-time mount corrections or inform scheduling decisions [20, 39, 40]. CMOS devices enable iterative, feedback-driven control loops (autoguider loops, high-speed verification) and support computational imaging methods (lucky imaging, de-rotation control mechanisms) that can be integrated into robotic frameworks.

Complete automation leverages image analysis for control. Plate solving is used to confirm pointing and to correct mount offsets, as it can replace or complement mechanical alignment routines, being widely used to recover from mispointing without manual intervention. Autoguiding techniques use a guide star (either a separate guide camera or a subframe of the main camera) to compute tiny corrections to the mount during exposures, preserving image sharpness for long integrations [23]. Automatic focusing algorithms evaluate image quality metrics (Full Width at Half Maximum (FWHM), contrast, Laplacian statistics) across a set of images acquired at different focuser positions to find optimal focus and several autofocus strategies and search algorithms (binary search, hill-climbing, fibonacci search) are used in automation pipelines [41].

For alt-azimuth mounts, field rotation during long exposures must be compensated. Mechanisms such as an Eliminating Image Rotation Mechanism (EIRM) combine mechanical counter-rotation with real-time control driven by camera feedback to limit field rotation artifacts. Recent work shows that carefully designed control laws and high-precision sensors can reduce residual rotation error to the arcsecond level, enabling accurate long-exposure imaging without complex post-processing [40].

Automation frameworks expose cameras and imaging functionality as first-class resources. They orchestrate exposure scheduling, guide loops, calibration sequences and image pipelines, often via driver layers such as INDI or Astronomy Common Object Model (ASCOM)/Alpaca that present device capabilities as properties and actions. These frameworks provide the glue between instru-

ment drivers, scheduler, and image-processing modules so that cameras become controllable programmatically in a structured, repeatable way [20, 23].

Table 1: Comparison of CCD, CMOS/sCMOS, and EMCCD Sensors in Automated Observatories

Sensor Type	Strengths	Weaknesses	Role in Automation
<b>CCD</b>	<ul style="list-style-type: none"> <li>- High quantum efficiency and low read noise [37]</li> <li>- Ideal for long-exposure scientific imaging</li> <li>- Mature technology, widely supported by automation frameworks</li> </ul>	<ul style="list-style-type: none"> <li>- Slower readout speed, limits high-cadence feedback loops</li> <li>- Large file sizes and data throughput considerations</li> <li>- Cooling often required</li> </ul>	<ul style="list-style-type: none"> <li>- Primary science imaging in automated observatories</li> <li>- Integration into automated exposure pipelines, calibration sequences, and unattended operations [23]</li> </ul>
<b>CMOS / sCMOS</b>	<ul style="list-style-type: none"> <li>- Fast readout and high frame rates [20, 39, 40]</li> <li>- Low power consumption</li> <li>- Supports high-speed feedback and real-time analysis</li> <li>- Often lower cost than CCDs</li> </ul>	<ul style="list-style-type: none"> <li>- Slightly higher noise and non-uniformity in some cases</li> <li>- Requires careful calibration for scientific-quality measurements</li> </ul>	<ul style="list-style-type: none"> <li>- Real-time control imaging (guiding, autofocus, sky/cloud monitoring)</li> <li>- Enables iterative closed-loop control and computational imaging methods (lucky imaging, field deroation) [40]</li> </ul>
<b>EMCCD</b>	<ul style="list-style-type: none"> <li>- Extremely low read noise and high sensitivity [37]</li> <li>- Suitable for low-light, fast events</li> <li>- Effective for detecting faint objects or weak spectral lines</li> </ul>	<ul style="list-style-type: none"> <li>- Expensive and complex</li> <li>- Limited dynamic range in some configurations</li> <li>- Gain stage may require careful calibration</li> </ul>	<ul style="list-style-type: none"> <li>- Specialized automation tasks such as faint object detection, fast photometry, adaptive optics feedback</li> <li>- Can be integrated into robotic frameworks for high-sensitivity imaging [23, 37]</li> </ul>

Automated observatories may adopt a hybrid camera approach to use the complementary strengths of different sensor technologies. Usually, a cooled CCD or EMCCD serves as the primary science camera for high-precision imaging, long exposures, photometry or spectroscopy, granting high quantum efficiency, low read noise, and excellent charge transfer performance [20, 23, 37]. As for the secondary camera, CMOS or sCMOS camera is used for real-time control and feedback. These auxiliary cameras, often smaller and faster, provide near-instantaneous images to support autoguiding loops, plate solving, cloud detection, and rapid target acquisition [39, 40].

This dual-camera configuration allows automation frameworks to maintain a high level of observing efficiency and reliability. While the science camera executes scheduled exposures and calibration sequences, the CMOS guide camera continuously monitors the mount’s pointing and tracks

stars to apply corrections or evaluates the sky conditions to make safety decisions. By combining these sensors, the system can control different aspects, such as maintaining focus, adjusting for mount drift and compensating for field rotation with mechanisms like the EIRM [40]. This type of setup maximizes the observatory's operational flexibility and ensures that both scientific objectives and mechanical constraints are respected, even under fully unattended remote operation [20, 23].

While the focus of this dissertation is the remote access to the telescope, the dissertation limits itself to the telescope navigation and has the real-time imagery as future works.

## 2.7 Automation Protocols

Remote control is the main characteristic of modern astronomical observatory operations, allowing both professional and amateur astronomers to access and manage telescopes and instrumentation from any location. The increasing automation of observatories relies heavily on well-structured remote control and communication systems, which are essential for efficient and safe operation of telescopes, cameras, domes and auxiliary devices.

Remote observatory control involves operating equipment over long distances, using the internet in general [7, 14, 42]. This capability enables astronomers not only to perform observations from locations far away from the observation site but also opens up access for educational outreach and public engagement: users can actively participate in the observation of celestial objects without being on site, provided a network connection exists.

This is made possible through the application of combined communication protocols, software frameworks, and intuitive user interfaces. Communication with users from observatory systems may be provided through web servers by using Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) protocols [25, 43, 44], or through lower-level network protocols like Transmission Control Protocol/Internet Protocol (TCP/IP) [7, 38]. Message Queuing Telemetry Transport (MQTT) is a lightweight publish-subscribe network protocol used in some setups for efficient messaging between devices [14]. Wireless methods of communication also exist, such as Wi-Fi, to further extend the flexibility of remote control [10].

Standardized communication protocols like ASCOM <sup>6</sup> and INDI <sup>7</sup> are central in facilitating interoperability among the varied hardware and software components, easing the integration and

---

<sup>6</sup><https://ascom-standards.org/>

<sup>7</sup><https://indilib.org/>

monitoring of observatory systems [11, 45]. Such frameworks allow the utilization of a wide range of devices from different manufacturers and also support automation and scripting for advanced observation routines.

The Astronomic Communication Library (ASTCom) is a communications framework that allows the nodes of an autonomous telescope system to exchange data efficiently in real time [43]. Grounded on the WebSocket protocol, ASTCom defines a common interface for signal serialization and transmission, thus allowing every parameter of the telescope controller to be remotely accessed without impairing real-time performance [43]. The framework provides communication between the STMicroelectronics 32-bit microcontroller STM32, acting as the main real-time controller, the RPI 4, and external client applications, where the RPI acts as a relay between controller and external systems [43]. Basically, ASTCom enables full monitoring of every signal from the controller, thus allowing automated routines and human operators to oversee, make changes in, and manage the system while maintaining the integrity and stability of the control loop.

In this dissertation, the HTTPS and INDI communication protocols are used, given their robustness and generalizability to any kind of telescope.

## 2.8 Automation User Interfaces

Since the nature of the systems for remote observatories is very complex, intuitive and user-friendly GUIs need to be developed. GUIs ensure that in a real-time mode, feedback on the status of observatories is given, management of devices is simplified, and the user can monitor and control observations [7, 9, 13, 19, 38, 46]. Modern observatories integrate GUIs with web-based platforms, permitting users to access them from any device connected to the internet [11, 47]. For instance, advanced systems enable users to submit observation requests by specifying parameters such as target coordinates, instrument configuration, and exposure settings, through a dedicated web portal [36].

These platforms can also implement role-based access control, assigning different privileges to users, thereby supporting a wider user community while maintaining operational security [47]. Mobile applications further extend remote control capabilities, allowing administrators and users to monitor telescope status, view schedules and manage observations from smartphones or tablets [47]. Some observatories even explore the use of virtual reality applications, providing immersive simulations of telescope operations, improving educational and outreach experiences [47].

In this dissertation, the custom front-end GUI is showcased using modern and accessible web UI technologies, allowing users to interact with the telescope through any internet browser.

## 2.9 Automation by the Communities

The amateur astronomy community has been involved in a very important way in the development of low-cost observatory projects by sharing experience, designs, and software that increased access to astronomical research [10, 19, 48]. These sources are not peer-reviewed, but they contain very valuable, field-tested strategies that are absolutely relevant for small observatories and Do-it-Yourself (DIY) projects. Some of those amateur projects successfully converted automotive components, such as wiper motors combined with timing belts and pulleys, to achieve strong and inexpensive vertical shutter motion <sup>8</sup>. Other projects describe the use of battery-powered linear actuators charged by solar panels, thus making it possible for an observatory shutter to run independently from external power grids <sup>9</sup>. Reed switches or Hall-effect sensors are often used to detect the dome's home position and trigger recalibration routines, providing a reliable and affordable strategy for repositioning the dome with decent accuracy <sup>10</sup>. DIY designs may become more sophisticated and incorporate proportional-integral-derivative (PID) control algorithms to smooth dome rotation and reduce mechanical wear [17].

This dissertation follows the DIY principles and contributes to the open-source community by providing the generalizable interface to remotely operate the telescope.

## 2.10 Automation using Affordable Technologies

The increasing accessibility and affordability of electronic components has allowed the development of low-cost astronomical observatories, which combine affordable hardware with open-source software to make astronomy more inclusive and practical. These systems are designed to be portable and easily installable, making them particularly suitable for smaller observatories, educational institutions and outreach initiatives [7, 14, 42]. Their cost-effectiveness also enables detailed observations and data collection in remote or challenging environments, supporting research that would otherwise be prohibitively expensive [10, 49, 50].

<sup>8</sup><https://www.skybadger.net/projects/shutter/shutter.shtml>

<sup>9</sup><https://stargazerslounge.com/topic/374969-diy-automation-of-pulsar-27m-dome-rotation-shutter-and-weather-monitor/>

<sup>10</sup><https://stargazerslounge.com/topic/374969-diy-automation-of-pulsar-27m-dome-rotation-shutter-and-weather-monitor/>

The use of MCUs is a central enabler of this democratization, to the extent that it allows educational institutions, amateur astronomers, and research projects access to and automation of observation tools at a fraction of the traditional cost by controlling motors, sensors and other components [13,19]. Among these, the RPI stands out as a primary computing platform for remote observatories thanks to its ability to run full operating systems, provide network connectivity, and interface with a wide range of hardware [11,14]. Such flexibility and power have inspired projects ranging from professional observatories to fully 3D-printed open-source telescopes and miniature domes controlled via headless remote sessions <sup>11</sup> <sup>12</sup>.

Open-source software has its roots in minimizing these costs. Standard protocols, such as ASCOM and INDI, ensure that hardware and software components can talk to each other [11,45], thus enabling a single interface to control telescopes, cameras, domes and more [11,45]. Open-source observatory and telescope control systems like TALON [9] offer all-inclusive sets of tools for telescope management, image acquisition, atmospheric monitoring, and alarm handling. Some projects are focused on open-source data processing, such as IEEC Calibration and Analysis Tool (ICAT), for calibration and analysis of images [9]. These enhance image quality, correct distortions, and provide support for accurate telescope pointing with advanced calibration and alignment algorithms [14].

Further reductions in the cost of production, customization and maintenance have also been promoted by the adoption of 3D printing for such items as telescope tubes, mounts and even complete domes [14]. Where low-cost systems are remarkably advantageous, they have been confronted with some challenges regarding long-term performance and durability when compared to commercial alternatives. Reliable and accurate data will, therefore, be essential for wider use of the system [50,51]. Active community involvement is required for continued open-source observatory development, including the creation of comprehensive documentation and support systems that provide guidance to users on implementation and troubleshooting [51,52].

This dissertation provides an initial step in remotely operating the astronomical observatory, leveraging affordable technologies to make the system accessible and practical for a wide range of users.

---

<sup>11</sup><https://www.instructables.com/Pi-lomar-3D-Printed-Working-Miniature-Observatory-/>

<sup>12</sup><https://3dprint.com/250384/telescope-prime-features-raspberry-pi-and-3d-printed-parts/>

## 2.11 Design Decisions for the Proposed System

After reviewing the literature and discussing possible approaches with a professional astronomer, several important design decisions were made regarding the technological side of the proposed system. It has been identified that the INDI protocol provides a generalized way in which access to and control of various types of telescopes and equipment can be easily done. At the same time, existing mobile applications like Stellarium Mobile can be adopted to enhance user experience, while raw telescope commands, which are made possible through the Meade Telescope Serial Command Protocol, can also be used as an alternative to directly control the telescope.

The first design decision was about the computational backbone of the system. MCUs seemed particularly useful, since they are more affordable, energy efficient and versatile. They are used in many observatory automation projects, especially in amateur and semi-professional contexts, where they handle tasks such as device control, data acquisition and peripheral integration. For this project, their affordability and adaptability made them an ideal choice for building custom control units capable of managing multiple observatory components.

The second design decision focused on actuation and sensing. Motors and sensors represent the physical interface between software commands and mechanical motion, and the literature emphasizes their core role in dome and shutter automation. Stepper or DC motors, when paired with position encoders, limit switches and other sensors, enable precise movement control and provide continuous feedback. This feedback is crucial for safe operation, ensuring that the dome maintains alignment with the telescope and that shutter states (open or closed) are always known. Without such mechanisms, risks of mechanical collisions or misalignment would significantly increase.

While MCUs, sensors and motors form the hardware foundation, the choice of software framework ended up to be the most important decision. The INDI protocol appeared especially advantageous for the proposed system. Since the system was intended to run on a RPI, compatibility with a UNIX-based environment was a decisive factor, and INDI met this requirement. More importantly, INDI abstracts device specific details through a driver based architecture. Each driver exposes properties that are shared across devices of the same class. This abstraction simplifies automation logic, since scripts developed for one telescope or camera can be reused with minimal changes for another. Another strength of INDI lies in its extensibility and modularity. New devices can be

integrated simply by adding their drivers, without altering the core system. This grants that the observatory can evolve with future hardware upgrades, while reducing long-term maintenance.

For development purposes, the telescope control system was implemented and tested on a personal computer running Ubuntu Linux. This choice offered three main benefits: (i) rapid iteration with desktop development tools; (ii) strong environment similarity to the RPI deployment; and (iii) access to INDI's device simulators. The simulators, more specifically in this case the Telescope Simulator, allowed the system's logic to be tested incrementally, even in the absence of physical hardware. This accelerated progress while reducing risks. Once the system was ready to deploy, migration to the RPI would require minimal adjustments beyond environment setup and performance tuning.

These design decisions form the foundation for the proposed system. By combining cost-effective MCUs, reliable sensing and actuation mechanisms and the flexibility of the INDI framework, the architecture achieves both affordability and scalability. It is designed not only for the OAUMa observatory but also as a blueprint for other facilities seeking accessible and modular approaches to telescope automation.

## 3 Methodology

This dissertation applies the DIY principles to automate the operations of the OAUMa. It focuses on providing an interface to remotely control the telescope of the OAUMa and also provides a possible implementation guide for a remote control system that enables efficient management of the observatory's equipment and facilitates astronomical observations from off-site locations. The proposed approach is generalizable to other not yet automated observatories.

### 3.1 Initial Status of OAUMa

The OAUMa, built in 2024 and located on the rooftop terrace of the University of Madeira, remains a manually operated facility (Figure 1). Its current setup requires the physical presence of an operator, since none of the primary systems are automated.

The dome's shutters, consisting of two parts (upper and lower), must be opened manually. The upper shutter is operated by rotating a mechanical handle, while the lower shutter is released by slowly manually releasing the metallic cables until it reaches approximately a 120° open position. The dome's rotation is powered by a motor that dates back nearly 25 years. This motor is controlled via a simple handheld device, requiring an operator to physically command the dome to rotate left, right or stop the motion.

The telescope is mounted at the center of the dome, on top of a steel base. It too requires manual control through its dedicated digital hand controller, which allows users to do many actions, like pointing to a celestial object, and save many settings important for observations and calibration purposes. Furthermore, the telescope is equipped with an access point (AP), that allows connection and communication with external software, such as Stellarium<sup>13</sup> and SkySafari Plus<sup>14</sup>, among others, providing additional control and visualization tools to the telescope control.

Nevertheless, the access to the observatory is restricted. Entry to the rooftop terrace requires a key, as does entry into the observatory dome itself, limiting usage mainly to authorized people, making public access difficult. Moreover, some equipment remains underutilized: the observatory's CCD camera is currently not installed, and the solar panel system is not in operation, as the facility is now directly powered by a 220V electrical line from the university.

---

<sup>13</sup><https://stellarium.org/>

<sup>14</sup><https://skysafariastronomy.com/>

In summary, the OAUMa's existing equipment and infrastructure include:

Table 2: Summary of Observatory Equipment and Dome Specifications

Component	Specification / Description
<b>Dome</b>	Sirius School Observatory Dome <sup>15</sup> Total Height: 3.25 m Wall Height: 1.5 m Diameter: 3.5 m Door Size: 1.15 m (H) × 1.0 m (W) Aperture Width: 1.0 m Open Past Zenith: 400 mm
<b>Dome Rotation</b>	Motorized with 12 V DC wiper motor and 12 wheels
<b>Power Supply</b>	4 W Solar Panel
<b>Base</b>	5 × 5 × 0.12 m reinforced concrete platform
<b>Shutter Mechanism</b>	Upper shutter: manual handle Lower shutter lid: manual operation
<b>Telescope</b>	MEADE LX200, 30 cm, with tripod, column, and accessories Includes integrated MCU and built-in AP
<b>Camera</b>	CCD SBIG 7XE with Autoguider (no filter wheel) CCD camera rotator
<b>Other</b>	Spare 12 V DC wiper motor



(a) Observatory building



(b) Telescope inside observatory building

Figure 1: OAUMa - Observatório Astronómico da Universidade da Madeira

### 3.2 Proposed OAUMa Automation Architecture

While the dissertation focuses on remote control of the telescope of OAUMa, the system proposed throughout this dissertation is also centered around the development of an affordable, IoT-based automated control solution for astronomical observatories. The goal is to provide an architecture design that can be both scalable and adaptable for other observatories that are not automated.

In discussions with the professional astronomer, the focus of this system is to enable remote operation of the observatory through a web-based interface. With this interface, the users should be able to monitor and control the observatory from any location, as long as there is access to internet. Besides the interface, the main operational objectives include the automation and synchronization of the dome's rotation, the opening and closing of the shutters, and the precise control of the telescope.

In addition, the system emphasizes the importance of safety mechanisms to prevent equipment damage and to protect users. By using sensors and control units, the observatory can operate autonomously while maintaining situational awareness of weather conditions, positional data and equipment status. At the end, this proposed system aims to make observations more accessible, efficient and safe for all types of users, with or without professional backgrounds.

The automation of the observatory was designed after consultation with a professional astronomer and the project's team, ensuring that both scientific usability and long-term maintainability were prioritized in the system architecture. The goal of the automation project is to transform the observatory into a low-cost remotely operable facility, capable of conducting scheduled observation sessions, adapting to weather conditions and protecting the equipment without requiring human presence.

From a functional perspective and by taking in consideration the process of performing an observation, from the beginning until the end, the automation of the observatory can be divided into four major subsystems: dome rotation control, shutters control, telescope control and environmental monitoring. Each subsystem has its own dedicated MCU, which provides modularity and distributed processing, ensuring that the failure of one of the components doesn't compromise the rest of the system.

The system is therefore composed of four MCUs, each with distinct objectives:

- Master MCU – central coordinator and dome rotation controller.
- Shutters MCU – control of upper and lower dome shutters.
- Telescope MCU – management of telescope peripherals such as cameras and filter wheels.
- All-Sky MCU – environmental awareness through sky imaging and weather monitoring.

This architecture follows a client–server communication model, where the Master MCU functions as the client and supervisory node, while the remaining MCUs act as servers responsible for local device control. Consequently, the system becomes more scalable, since it facilitates the addition of new subsystems through new MCUs, while also providing a clear separation of concerns, where the Master MCU coordinates the necessary actions for the tasks and the other MCUs take care of the actions themselves.

### 3.2.1 Master MCU

The Master MCU is the central piece of the automation system. Its primary role is to coordinate all subsystems, ensuring that the dome, shutters, telescope and environmental monitoring components operate in synchrony. In practice, this MCU establishes communication with each of the secondary MCUs, requesting data and issuing commands as necessary. By design, the Master MCU functions as the bridge between the remote user interface and the physical observatory hardware.

Besides coordinating tasks, the Master MCU is also responsible for the control of the dome’s rotation, including the alignment between the shutters and telescope’s pointing direction. The strategy used to take care of this alignment problem is to utilize inertial measurement units (IMUs) mounted on both the dome and the telescope. By continuously comparing their orientations, the Master MCU calculates and executes the required dome adjustments to make sure both IMUs match, meaning that both the shutters and the telescope are aligned. This functionality is more important for long observing sessions, where the telescope’s azimuth may change significantly over the observation time.

In addition to alignment synchronization, the Master MCU manages the overall workflow of an observation session. This includes initializing the system, verifying weather and safety conditions (via the All-Sky MCU), coordinating the opening of the shutters, aligning the dome with the telescope and monitoring status throughout the observation. If the conditions become unfavorable,

like cloud coverage increases or precipitation is detected, the Master MCU is can initiate a safety protocol to stop the observation.

Through its role as both a controller and coordinator, the Master MCU ensures that the observatory operates as a cohesive and adaptive system, while still allowing individual subsystems to function independently.

### **3.2.2 Shutters MCU**

The Shutters MCU is dedicated to controlling the upper and lower shutters of the dome. By isolating this functionality into a dedicated MCU, the system ensures that shutter operation remains independent of other tasks, reducing the risk of communication bottlenecks or delays in timed operations such as weather emergency closures.

From a hardware perspective, the Shutters MCU is directly connected to the shutter motors, to be able to control the opening and closing mechanism. It also interfaces with limit switches that allow to determine whether the shutters are fully opened or fully closed, in order to prevent mechanical stress or damage that could occur if the motors were to run beyond their intended range of motion. In addition, the Shutters MCU is equipped with an IMU. The IMU gives orientation data that enables the system to verify whether the shutters are properly aligned with the telescope's optical axis. The data from this IMU is transmitted to the Master MCU, which integrates it with the Telescope MCU's IMU orientation data to coordinate dome rotation and shutter alignment.

Finally, from a systems integration point of view, the Shutters MCU functions as a server node within the overall communication framework. The Master MCU issues commands (e.g., open, close, stop or query state), and the Shutters MCU executes these commands locally while reporting back the current status of the shutters.

### **3.2.3 Telescope MCU**

The Telescope MCU takes care of the telescope's imaging equipment, specifically the astronomical camera and the filter wheel, as well as an IMU. This specialization ensures image gathering tasks are streamlined, without overloading Master MCU.

At its core, the Telescope MCU manages the astronomical camera, responsible for capturing both scientific images and guiding frames. In addition, this MCU also controls the filter wheel for the astronomical camera, which allows switching between optical filters during observation

sessions. This capability is essential for photometric measurements and multi-wavelength studies, where observations must be repeated across different spectral bands.

In addition to imaging functions, the Telescope MCU incorporates an IMU that provides orientation data. This information is used for verifying the telescope’s pointing direction when synchronizing dome and shutter with the telescope’s azimuth. By reporting its orientation to the Master MCU, the Telescope MCU enables dynamic adjustments of dome position to make sure that the telescope’s line of sight remains unobstructed throughout an observation session.

The Telescope MCU also operates as a server node. The Master MCU issues commands, such as “capture image”, “switch filter” or “report orientation”, which the Telescope MCU translates into device instructions.

### **3.2.4 All-Sky MCU**

The All-Sky MCU focuses on environmental and sky monitoring, being the observatory’s primary safeguard for weather awareness and observation readiness. By providing constant data, the All-Sky MCU enables both automated systems and human operators to make informed decisions about when to initiate, continue or terminate an observation session.

One of the main responsibilities of the All-Sky MCU is the control of the all-sky camera, a wide angle imaging device that captures the full dome of the sky. The images are used to evaluate cloud coverage, sky brightness and general observing conditions in real time.

In addition to the all-sky camera, the All-Sky MCU is integrated with a weather station, which measures atmospheric parameters such as temperature, humidity, wind speed and pressure. These measurements are required for ensuring safe operation, as high winds or excessive humidity can pose risks to the telescope and sensitive electronic equipment. Together with the sky images, the weather data creates a comprehensive profile of the local environment.

The system can be configured to automatically stop telescope operations and close shutters if unfavorable weather conditions are detected. At the same time, operators can remotely review the sky images and sensor readings before overriding or confirming system decisions. This hybrid approach ensures both safety and flexibility in observatory operations.

The data generated by the All-Sky MCU can also contribute to long term observatory performance analysis. By storing sky and weather logs, researchers can evaluate seasonal observing

conditions, track climate trends at the observatory site and optimize scheduling for future projects. This makes the All-Sky MCU not only a monitoring unit but also a data collection instrument for site characterization studies.

Finally, the All-Sky MCU functions as a server node. It processes sky images and sensor readings, then makes the data available to the Master MCU through an Application Programming Interface (API) endpoint.

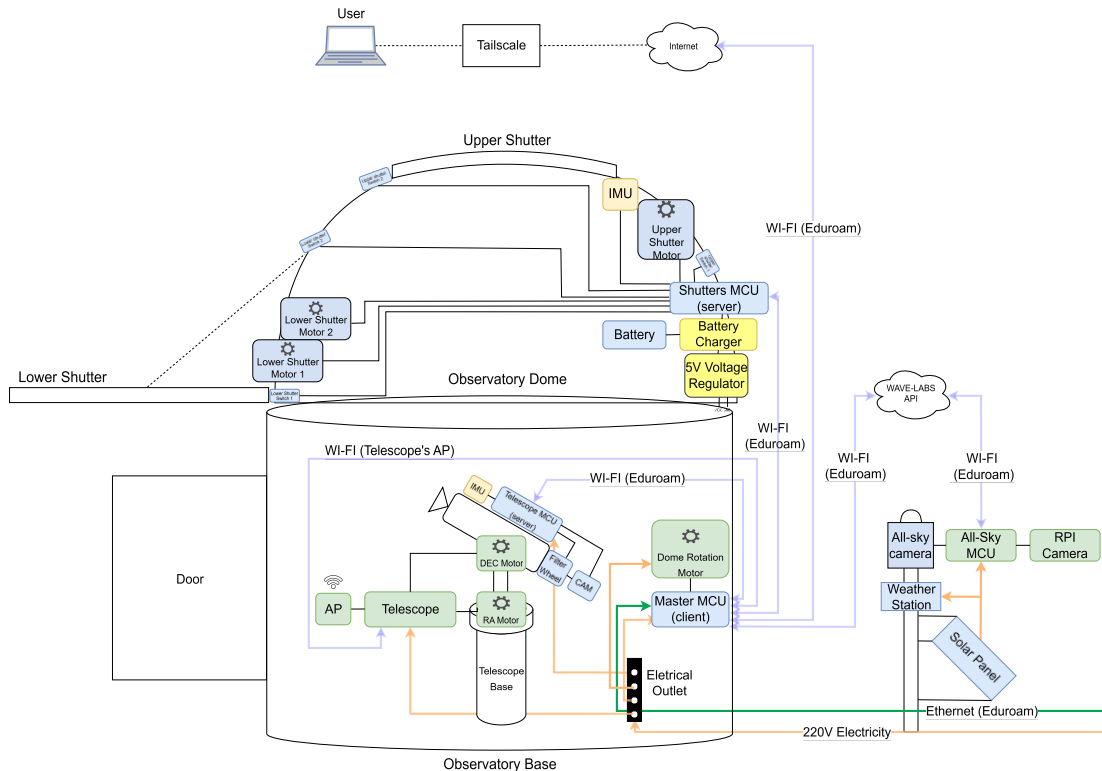


Figure. 2: Proposed schematic of the astronomical observatory control system. This setup assumes that there are an upper shutter and a lower shutter, and that the dome has motors installed previously

### 3.2.5 Power Distribution and Network Communication

In addition to the control of individual components by the MCUs, the system design must also take in consideration the power distribution and network communication. Devices such as motors, sensors and controllers require power to work, while network communication is necessary for coordination between the nodes.

Since the shutters are composed of two primary components, the upper and lower shutter, and rotate together with the dome, an electrical system capable of supplying power to the shutters

without being disrupted by the dome's continuous rotation must be designed. The challenge lies in ensuring a reliable power connection between a stationary source (the base of the observatory) and the rotating structure (dome), as shown in Figure 2, since a direct power cable from the base to the dome would lead to entanglement due to the dome's capability of rotating more than 360° in both directions.

Transferring power to rotating systems solutions, like slip rings or rotary joints, are usually expensive, prone to wear and require frequent maintenance, which lies beyond this project's budget and maintenance constraints. Instead, the concept of "home position", as described in prior works [7], was chosen.

The "home position" corresponds to a predefined fixed position of the dome, where electrical connection from the base to the dome is established. The mechanism contains a spring contact on the dome that makes contact with a stationary terminal, completing the circuit, allowing power to reach the upper part of the observatory. Consequently, power to the dome is only available when it reaches this position. This behavior is similar to how a trolley bus draws power: only when its overhead connector aligns correctly with the power line does it receive electricity. In the case of the observatory dome, the "home position" is the only alignment in which the shutters can be powered, ensuring that the wiring remains untangled and safe from mechanical damage. Operationally, this means the dome must always rotate to the "home position" before starting any shutter movement. At the start of an observation session, the dome first moves (if not already there) to the home position to open the shutters. Similarly, at the end of the session, it must return to this position before the shutters can be closed. This procedure guarantees a secure and repeatable method for energizing the shutter system while preventing electrical or mechanical complications during dome rotation.

In order to enable the interaction between the MCUs, an AP is incorporated to the architecture, to provide internet connectivity and local communication. This makes so commands and data can be exchanged between subsystems without external dependences. The Master MCU is equipped with both Wi-Fi and Ethernet interfaces, allowing for interaction with the telescope's AP and for over the internet external user interaction.

For remote access, all MCUs are part of a Virtual Private Network (VPN) service from Tailscale. With this, each MCU has their static Internet Protocol (IP) address, which simplifies device dis-

covery, coordination and control. This eliminates the need of port forwarding or dynamic Domain Name System (DNS) configurations and provides encrypted communication across the internet. As a result, access to each subsystem does not compromise network integrity or expose endpoints to the public internet.

### 3.2.6 Workflow

The workflow of a typical observation night, as seen in Figure 3, can be described as a sequence of interactions between the user, the Master MCU and the other MCUs:

1. System initialization and condition check: The user begins by opening the observatory's web control interface. Before initiating any observation, the user verifies if the conditions are favorable. This information is provided by the All-Sky MCU, which continuously monitors sky coverage and weather conditions. If the sky is clear and safe for operation, the user may proceed.
2. Opening the observatory: The first operational step involves preparing the dome. While the dome is in its "home position", the Master MCU sends a command to the Shutters MCU to open the shutters. Once the shutters are fully open, the Shutters MCU reports back to the Master MCU, confirming that the system is ready for telescope operations.
3. Telescope setup and slewing: After configuring essential parameters such as date and time, the user requests the telescope to slew to a target star. This command is relayed through the Master MCU, which communicates with the telescope via its AP, sending the necessary instructions and receiving real-time feedback from the telescope.
4. Dome synchronization with the telescope: While the telescope slews to the target, the Master MCU periodically queries the IMU data from both the Telescope MCU and the Shutters MCU. Using these values, the Master MCU calculates the relative azimuthal difference and commands the dome to rotate until the shutters are properly aligned with the telescope's pointing direction. This process ensures that the telescope always has an unobstructed view of the sky.
5. Optional camera operations: During the observation, the user may request additional operations to the Master MCU, such as changing the filter wheel position, in order to capture images in different spectral bands, which are then passed to the Telescope MCU to execute them.

6. Ending the observation: Once the observing session is complete, the user requests the telescope to park. As soon as the telescope begins its parking sequence, the Master MCU commands the dome to return to its "home position". When the dome has reached home, the Master MCU instructs the Shutters MCU to close the shutters, securing the observatory.

Throughout the session, the All-Sky MCU continues to monitor external conditions. If sudden unfavorable weather or excessive cloud coverage is detected, the All-Sky MCU immediately notifies the Master MCU. In such cases, the Master MCU initiates an emergency shutdown, stopping ongoing observations, parking the telescope, returning the dome to its "home position" and closing the shutters to protect the equipment.

This arrangement combines decentralization with centralized coordination. Each MCU remains autonomous in handling its own hardware, but the Master MCU ensures coordination. Such a design simplifies system management by distributing responsibilities, improves scalability by allowing new MCUs or sensors to be integrated with minimal interference and isolates potential points of fault. When combined with secure VPN access and internet communication, the result is a remotely operable observatory system that balances automation, accessibility and reliability.

### 3.3 Telescope Automation

The telescope is operated using its dedicated hand controller, AudioStar<sup>16</sup>. AudioStar comes with a database of astronomical objects that the user can choose from to track, as long as the telescope is correctly aligned and the target is above the horizon. Additionally, the interface allows users to change and setup settings, like time, date and geographic location. The hand controller also allows users to manually control the telescope, without needing to know coordinates or specify an object to track.

However, the goal is to automate this system so that physical input is no longer needed, enabling remote control of the telescope. Therefore, the automated system must be capable of replicating the same functionalities that AudioStar provides.

#### 3.3.1 Test 1: Stellarium App

The initial step on automating the telescope started with searching and evaluating existing solutions that were capable of remotely control the telescope. After some initial research and

<sup>16</sup><https://www.astroshop.eu/telescopes/10/a,Teleskope.GoTo.Software=AudioStar>

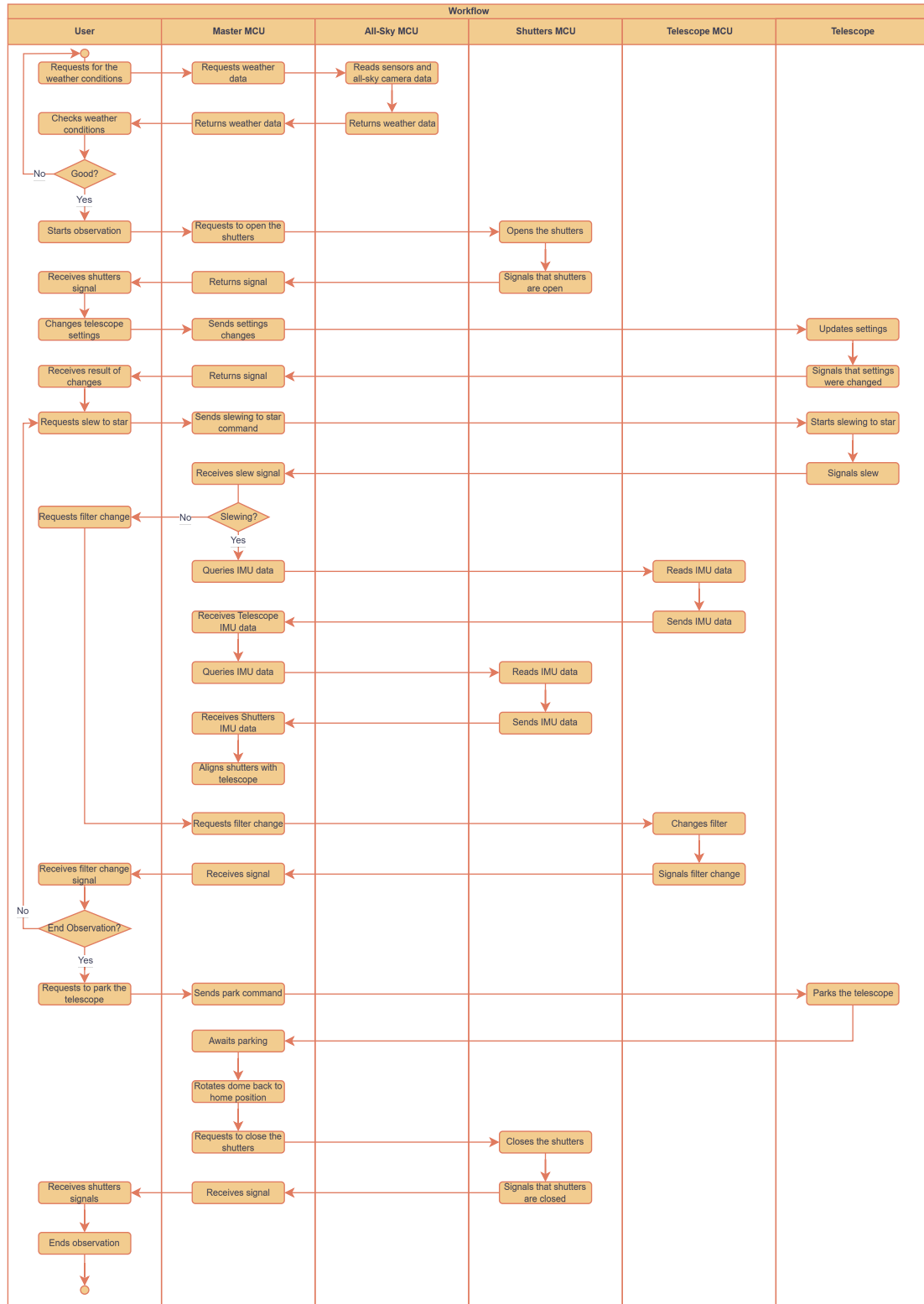


Figure 3: Observation night workflow

discussion with the project's team, the first chosen testing software was Stellarium<sup>17</sup>. Stellarium is an open-source planetarium software that provides real-time sky simulations, offers support for

<sup>17</sup><https://stellarium.org/pt/>

telescope control and it is often used by amateur astronomy communities, which made it a good starting point.

An initial test was done to connect the Stellarium mobile application to the telescope's AP. The connection was established successfully, with the software recognizing the telescope and executing the commands correctly. This short test confirmed that the telescope was able to communicate with external software, through its AP, serving as a proof of concept, showing that remote connection and control were possible.

Shortly after the initial test, a presentation about the automation of the observatory was held for a group of university students who would later collaborate in the project's development. During the discussion session, ideas and suggestions were proposed, one of them being to communicate directly with the telescope through its own serial command protocol instead of using an intermediary software, like Stellarium. While Stellarium provides a convenient GUI and built in telescope control features, it introduces a dependency into the system, which limits customization of the software, since the users are limited to what Stellarium provides.

In contrast, direct communication with the telescope through its native protocol provides more customization and control, allowing full access to all supported commands in the protocol. This approach not only allows for more customization, but also facilitates integration into a dedicated web-based control platform. However, this increases the time needed for the automation development process. Ultimately, the decision to prioritize direct communication over intermediary software represented a critical design choice that aligned with the project's objective of developing a customizable and autonomous observatory control system.

### **3.3.2 Test 2: Meade Telescope Serial Command Protocol**

The Meade Telescope Serial Command Protocol, derived from the LX200 protocol, is a communication protocol used to control Meade telescopes, including the LX200 and ETX series. It enables control of the telescope functions via serial communication.

The grammar's syntax is simple consisting of the <command-prefix> ("#" or ":"), followed by a <statement> (the command), followed by a <command-suffix> ("#"). Commands are case-sensitive across the grammar.

Many programmers simply use the ":" character alone as the <command-prefix>. This is perfectly valid. However, since there is no return value from many of the LX200 commands, there is no way to tell if the command was actually accomplished or if there was an error. Worse yet, there is no specific way to tell whether the command was even received. If the programmer begins each of the commands with a <command-prefix> of "##:" then the hash-mark will act as a terminator to any awaiting input. This may trigger a pending error, but it is possible to resync to the command processing state machine at this point and continue. Another method is to send a hash-mark by itself if no response has been received and listen for return characters. Either of these methods should work fine.

For the second test, a custom and simple software interface was developed to check if the protocol worked as expected. The interface contained a control panel that allowed users to control the telescope in four directions (north, south, east and west) and to input astronomical object coordinates for targeted slewing. The interface communicated with the telescope through serial commands existent in the protocol's documentation and supported by the telescope. A connection with the telescope was once again successfully established. The directional control worked correctly with little to no delay. However, when trying to move the telescope to specific coordinates, the telescope would move incorrectly or not move at all. An important thing to note is that up until this point, the telescope was not fully functional due to an internal circuit error that lasted months to fix, which, at the time, was thought that it could have been a reason for the problems during the test. This raised concerns about the accuracy or validity of the movements. Several hypotheses were proposed to explain these inconsistencies:

- Telescope alignment was not completed: without correct initial alignment, any movement to absolute coordinates would be inaccurate due to reference frame mismatch.
- Incomplete setup parameters: incorrect or missing setup values may have caused incorrect internal coordinate conversions.
- Invalid or restricted coordinates: the telescope might include built-in safety constraints that reject coordinates outside mechanical limits to prevent damage.
- Dependency on hand controller: the serial protocol might rely on the hand controller for some configurations.

Although the telescope was not working correctly, an attempt to fix one of the problems, the alignment and calibration of the telescope and its internal coordinate system, was done. The attempt was scheduled during nighttime hours to facilitate visibility of reference stars. However, unfavorable weather conditions, in this case dense cloud coverage of the sky, made it impossible to identify and lock onto the stars required for the alignment.

The custom interface was deemed successful, even though the targeted slewing functionality didn't work was expected, since the problem was considered to be on the telescope and not on the custom software. As a result, it became clear that a more advanced version of the interface could be developed. However, using this protocol introduces a significant limitation: it is only compatible to the telescopes that support Meade's specific command set.

After looking through research and discussing with the professional astronomer, INDI surged as an excellent option to overcome this limitation. As highlighted in existing literature, INDI offers a standardized communication protocol that supports a wide range of astronomical equipment regardless of brand or model. A major advantage of INDI is the INDI Library, a full implementation of the INDI wire protocol for UNIX-based systems. This makes it compatible with platforms such as the RPI, which is ideal for affordable and scalable observatory automation projects. By transitioning to INDI, the system will be capable of integrating various devices, regardless of the brand, facilitating future upgrades and modular expansion.

### 3.3.3 Test 3: INDI

INDI is a protocol designed to control astronomical instruments such as telescopes, CCD cameras, focusers, filter wheels, domes and weather stations. It was developed with the goal of standardizing communication between astronomy software and hardware across different manufacturers.

At its core, INDI provides a client-server architecture where devices (hardware) are controlled by INDI drivers running on a server and clients (software interfaces) connect to that server to issue commands or receive data.

The main components of the INDI system are the following:

- INDI Server: acts as a bridge between the client software and the hardware devices. Runs device drivers that directly communicate with the connected hardware (e.g., telescope mount or camera) and listens on a network port and accepts connections from clients.

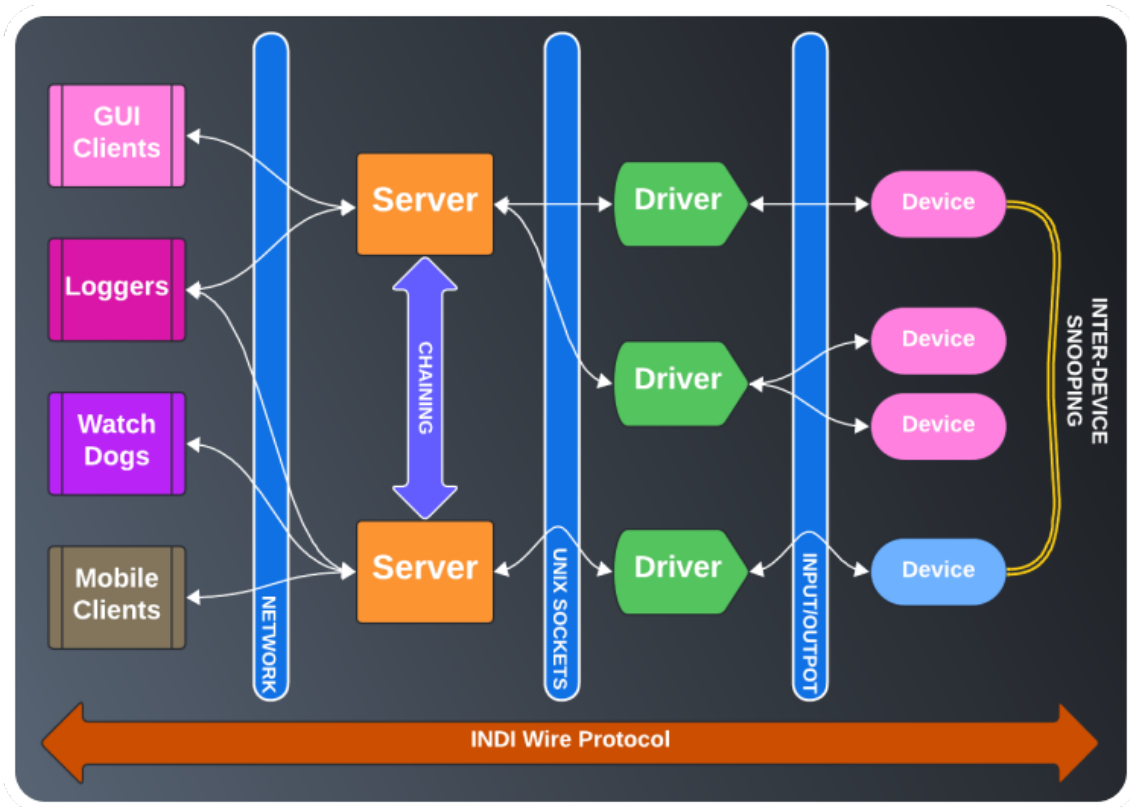


Figure 4: INDI Architecture Overview. Source: <https://indilib.org/>

- INDI Drivers: each hardware device is controlled by a specific INDI driver. Drivers expose the capabilities of the device (e.g., slew, track, capture, focus). They provide a standardized set of properties and commands through the INDI protocol, even if the underlying device APIs differ.
- INDI Clients: graphical or non-graphical interfaces used by users to control devices. These clients send requests (like “slew to coordinate” or “start exposure”) and receive status updates, images or telemetry. Examples: KStars/Ekos, INDIWeb, CCDciel, SkyCharts or custom web interfaces.
- Communication: INDI uses a XML-based wire protocol to communicate between clients and the server. Messages are sent over TCP/IP, allowing both local and remote control of observatory equipment. Clients do not need to be on the same machine as the INDI server, meaning that they can control devices over a local network or the internet. The INDI communication flow can be seen in Figure 5.

## INDI Communication Flow

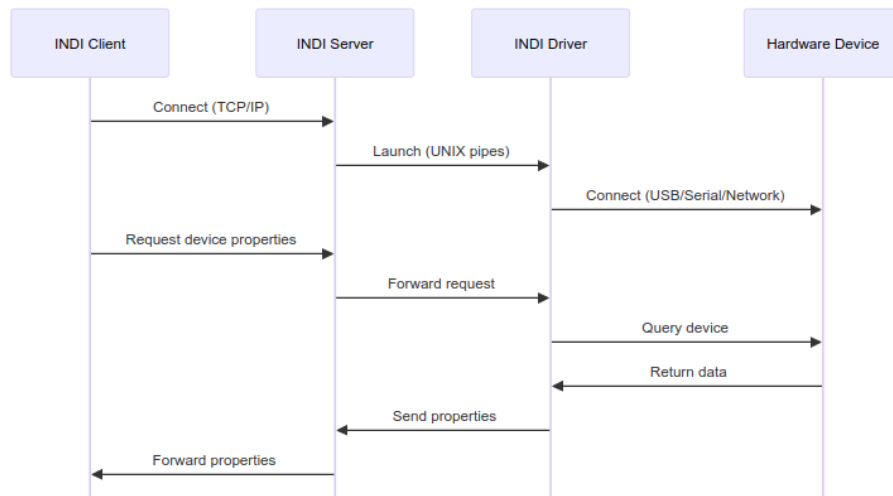


Figure. 5: INDI Communication Flow. Source: <https://docs.indilib.org/quick-guide/>

The main focus of the INDI protocol is that it is a communication model that works based on properties. The devices are controlled by updating device properties instead of direct commands as in traditional serial protocols, such as the Meade.

One of the major advantages of INDI is how easy it is to add new hardware. It's possible to create new INDI drivers for new devices by extending base classes from the INDI Library. Each driver should declare the properties supported by the device and how the device reacts when the properties are updated. This makes the system easily scalable by allowing the addition of new devices without the need to change the core of the software. This modularity allows INDI to scale to many device types, even allowing virtual or simulated devices to be created for development and testing.

To make full use of INDI's capabilities, a custom control interface can be developed that translates the control and communication model into intuitive and accessible controls for all types of users.

In summary, the goal from the first test was to verify and guarantee that the telescope could be controlled remotely. The second test aimed to explore direct communication with the telescope using its native serial command protocol, revealing limitations that necessitated a more flexible

solution. The third test focused on implementing the INDI protocol, which offers a standardized and extensible framework for device control.

The following section presents the proposed OAUMa Telescope Control Interface, a web platform that uses the INDI protocol for device communication, combined with a user-friendly interface. Additionally, the problem mentioned with the telescope during the second test was fixed before the development of this interface, meaning that the telescope was fully functional during this third test.

### 3.4 Proposed OAUMa Telescope Control Interface

The telescope control interface is designed to run on a MCU, specifically on a RPI. Taking in consideration the computational resources of the MCU, the technologies chosen should balance performance and usability.

The front-end of the telescope control system was developed using React, a popular JavaScript library. Several reasons led to this choice. First, React offers efficient rendering through its Virtual Document Object Model (VDOM), allowing even weaker hardware, such as the RPI, to handle responsive interactions. The second reason is that React works using components, which allows the interface to be easily extended with new features, while providing reusable code. In third, React supports dynamic updates without the need for a full page reload, which is essential for tasks that require constant update of information, like knowing the current position of the telescope while it is slewing. Lastly, React was picked because of previous experience using the framework, which reduces the implementation time. Other technologies were considered. Angular, while powerful, is a heavier framework with a higher learning curve and a larger runtime footprint which made it less suitable for the chosen hardware. Vue.js is similar to React in terms of reactivity, but provides less specialized libraries, which limits long term extensibility.

The technology used for the back-end of the interface was Flask, a web framework written in Python. In this case, Flask is the intermediary between the client and the INDI server, which handles API requests manages the responses sent by the INDI server. Its architecture allows for quick prototyping and deployment, which is important for a iterative development and testing strategy. Additionally, since it's written in Python, it has access to the INDI Library, which contains the INDI functionalities needed for this system. Alternatives such as Django and Node.js/Express were also considered. In this case, Django introduces unnecessary complexity for the API required

in this project, while Node.js with Express would increase development effort due to the need to manage asynchronous event loops present on both front-end and back-end, without offering huge advantages.

The main goal of this interface is to replicate the functionalities present in the telescope's AudioStar hand controller. Additionally, the interface should be designed to be intuitive to both amateur and professional users.

It's important to note that the possible features of the interface are limited by the INDI driver that is designed for the telescope, in this case the LX200 telescope. While INDI provides a standardized communication framework, some commands supported by the hardware may not be fully exposed by the driver. However, the system maintains the objective of being interoperable with other telescopes.

A web platform was chosen over a desktop application simply because of accessibility. Desktop software requires installation on each individual device, which creates technical barriers and increases maintenance. On the other hand, a web application can be accessed from any location as long as the user has internet connection, without the need to install anything. Moreover, the architecture of a web platform facilitates collaboration between multiple users at the same time, which can be beneficial for educational institutions by allowing students to access the system concurrently with appropriate access and control privileges.

In summary, React and Flask offer an efficient, maintainable and scalable solution that is compatible with the RPI hardware limitations and usability needs. In addition, the interface corresponds to a web platform, which allows for a wider accessibility of users due to its architecture.

### 3.4.1 Interface Data Flow

The proposed system follows a multi-tier architecture, in order to ensure a clear separation of concerns, where each tier has its role and communicates with the adjacent tiers through standardized interfaces. The system's data follows a repetitive flow, where commands are sent from the user to the telescope and the telescope's feedback is sent back to the user.

At a high level, the data flow, as shown in Figure 6, can be summarized as:

1. User Interaction: the user issues a command via the React interface (e.g., "Slew to RA: 10h20m, DEC: +20").

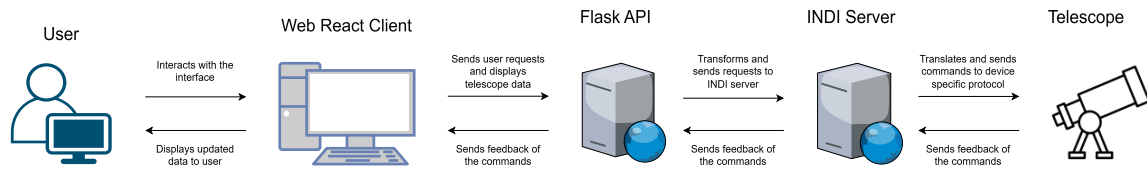


Figure. 6: System data flow overview

2. **Backend Processing:** the command is sent to the Flask API, where it is validated and translated into INDI protocol requests.
3. **Device Communication:** the API forwards the request to the INDI server, which communicates with the appropriate telescope driver;
4. **Hardware Execution:** the Meade LX200 telescope receives the command via its AP and executes the requested action.
5. **Feedback Loop:** Response from the telescope flows back through the INDI server and Flask API to update the interface.

#### ***User Layer (Control Interface)***

The user layer is represented by the web interface and it's the point of interaction between the user and the control system. The principal features of this layer are the following:

- **Responsive Design:** the UI adjusts to different screen sizes and input methods (touch, keyboard, mouse), providing a consistent user experience across devices ranging from smartphones to desktop monitors.
- **Command Execution and Feedback:** users can perform the desired actions, such as slewing the telescope or changing telescope settings. The interface then displays the outcome of these commands to confirm success or provide error messages.

#### ***Client Layer (React Web Application)***

The primary function of the client layer is to translate user actions into API requests, which are sent over HTTP to the Flask backend server. The main responsibilities of the client layer are the following:

- Command Submission via HTTP Requests: user actions, such as slewing to specific object coordinates, are converted into HTTP requests that are directed at the Flask API. Afterwards, the API manages the communication with the INDI server and the telescope itself.
- Input Validation and Error Prevention: user input, like coordinate formats, is validated before reaching the back-end, with the goal of reducing errors and unnecessary backend processing.

### ***Backend Layer (Flask API)***

This layer is represented by the Flask API, which acts as a bridge between the React web application and the INDI server. This is where the user requests are sent to, validated and then converted into INDI protocol commands. The main functions of this layer are the following:

- Command Reception and Processing: the Flask API receives HTTP requests from the React web client, which are then parsed and converted into INDI commands that the INDI server understands.
- Input Validation and Operational Safety: before sending the translated commands to the server, the user inputs go through a second validation phase to ensure that the parameters are correct and that the telescope won't be damaged or work beyond its capabilities.
- RESTful API Design: the Flask API exposes Representational State Transfer (REST) endpoints that follow the design principles of the REST architectural style. This standardization facilitates maintainability and integration.

### ***INDI Server Layer***

The INDI server is the connector between the API and the telescope itself. It implements the INDI protocol and is hosted on the same RPI as the Flask API, in order to reduce possible network latency. Core responsibilities of this layer are the following:

- Device Abstraction: the INDI server provides a consistent interface for many astronomical instruments. The abstraction given by this protocol allows for the Flask API to send generic commands without needing to know specific details of each device, enabling interoperability across different hardware.
- Driver Management: the server manages the INDI driver that was designed for each group of telescopes, in this case the Meade LX200 driver for the Meade LX200 telescope, allowing it to translate commands for the correct telescope.

- **Status Monitoring and Broadcasting (Polling-Based):** the INDI server is regularly checking for devices changes, which are then updated through the driver properties over the INDI protocol. Afterwards, the back-end receives these changes and updates the interface with the new data.

### *Telescope Layer (Physical Device)*

The telescope layer corresponds to the physical Meade LX200 telescope itself. The communication between the RPI and the telescope is done via its built-in Wi-Fi module, which creates a dedicated AP. The main functions of the telescope layer are the following:

- **Command Execution:** the telescope interprets and performs all the commands it receives, such as slewing to specific object coordinates.
- **Status Reporting Feedback Loop:** the telescope regularly transmits telemetry data in a polling process, every second by default. After executing the received commands, the results are sent back to the INDI server, ensuring that the previous layers have accurate and updated telescope information.

### 3.4.2 GUI

One of the primary objectives in developing the interface of the proposed telescope control system is to create an intuitive and user-friendly experience for the users, regardless of their prior experience with a telescope. The interface is designed to offer a modernized control interface and to ideally reduce the technical complexity of telescope operation without the need for expert assistance.

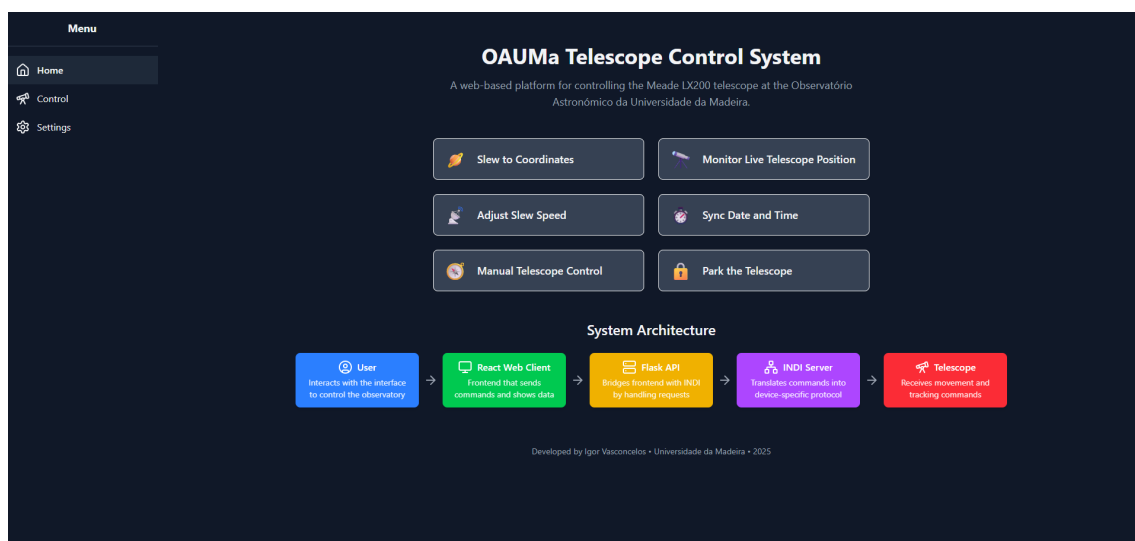


Figure 7: Proposed telescope control system's home page

The homepage, as seen in Figure 7, showcases a description of the control system. It provides an overview of the observatory's functionalities and explains the overall system architecture. The homepage acts as a starting point that offers contextual information about the system and its automation capabilities.

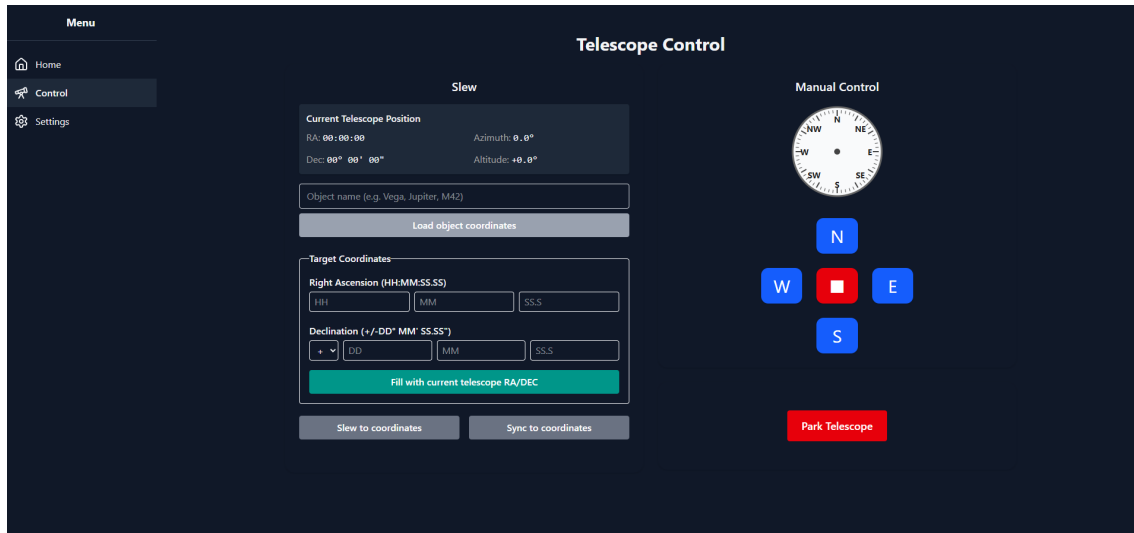


Figure 8: Proposed telescope control system's control page

The control page, as shown in Figure 8, contains the telescope movement control actions and it can be divided into two areas: the automated and precise control section (left half of the page), and the manual control section (right half of the page). The first allows users to perform more precise actions, based on target object coordinates, and the second is for direct manual adjustments, respectively.

In the precise control section (Figure 9), the system displays regularly updated data of the telescope's current position, which includes RA, DEC, azimuth and altitude, with a refresh rate of one second. Additionally in this section, can search for celestial objects by name or catalog identifier. This search first looks through a local catalog of astronomical objects and their positions, but if the object is not found, the system queries the Set of Identifications, Measurements and Bibliography for Astronomical Data (SIMBAD) database. Once the search object is found, its coordinates are populated in the target coordinates form, otherwise, an error shows up. The users can also manually input their wished target coordinates, without the need to look for an astronomical object. This form supports standard formatting for both coordinates (e.g., RA as 16:23:45.62, DEC as +62<sup>o</sup> 45' 13"), as well as some other formats. Afterwards, the users can press the "Slew to coordinates"

**Slew**

**Current Telescope Position**

RA: 00:40:53      Azimuth: 15.3°  
 Dec: 60° 14' 06"      Altitude: +6.7°

cassiopeia

Load object coordinates

**Target Coordinates**

**Right Ascension (HH:MM:SS.SS)**

00      40      30.24

**Declination (+/-DD° MM' SS.SS")**

+      60      14      6.00

Fill with current telescope RA/DEC

Slew to coordinates      Sync to coordinates

Figure. 9: Automated slew section

button, which then checks with the target coordinates are valid before starting the slewing process. These checks include:

1. Verifying that the target is above the horizon.
2. Ensuring the target's altitude remains within the predefined minimum and maximum limits.

If any of the conditions are not met, the slew operation is aborted and the user is warned about the situation. Additionally, the system changes the tracking mode of the slew based on the selected object: solar rate, for the Sun; lunar rate, for the Moon; and sidereal rate for the other celestial objects. Finally, a synchronization option is available and allows users to align the telescope based on the target coordinates picked.

The manual control section of the control page (Figure 10) allows for directional movement control over the telescope. At the top of this section, a compass display is shown with the goal of giving users visual feedback on their directional actions. This feedback is accompanied by toast notifications, that also indicate which direction the telescope is moving to.

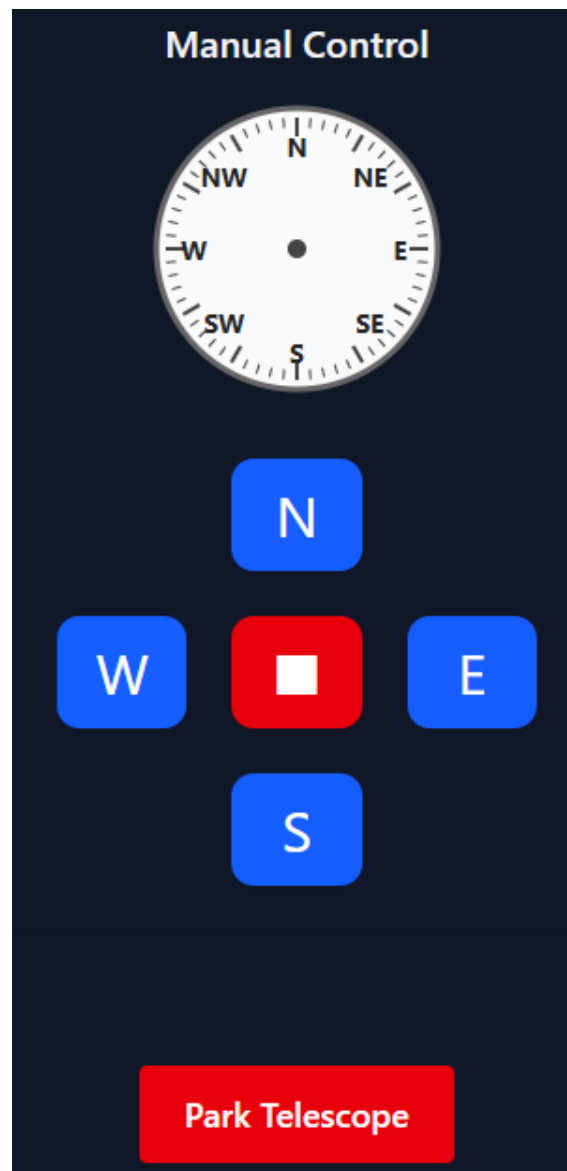


Figure.10: Manual control section

Below the compass display, the manual movement pad is showcased. The manual movement pad allows users to move the telescope along the cardinal directions: North, South, East, and West. The system supports simultaneous multi-directional movement (e.g., North-East) for greater flexibility and fine control. This pad is complemented with keyboard integration, improving usability for users who prefer non-mouse input. Safety mechanisms are built into the manual controls to prevent accidental damage to the telescope. For example, if a movement would go beyond the predefined altitude limits, the corresponding directional button becomes disabled, forcing the user to move the telescope to the opposite direction, back within the altitude limits. Due to the telescope's one second polling interval, movement beyond the limits can happen, especially if the user holds the

direction button down. This was taken in consideration when defining the maximum altitude limit, in order to protect the telescope and equipment around it. At the center of the control pad there is a stop button that doesn't affect the manual control movement (since the directional buttons are held down instead of tapped), but can be used to abort any movement, which is more useful for the precise controlled slews.

Finally, at the bottom of this section there is a park telescope button that makes the telescope return to its starting position. When parking the telescope with the hand controller, after the telescope reaches its initial position, the only action permitted afterwards is to turn off the telescope. When this functionality is done through the interface, a timer, of one minute, starts counting to give the user feedback on the parking process. Once the user chooses to park the telescope, the connection to the telescope is terminated, which prevents any further actions until the telescope is turned back on and the connection is established. Consequently, this timer is necessary to ensure that the telescope ends the parking process, while reducing the risk of unintentional operations during it.

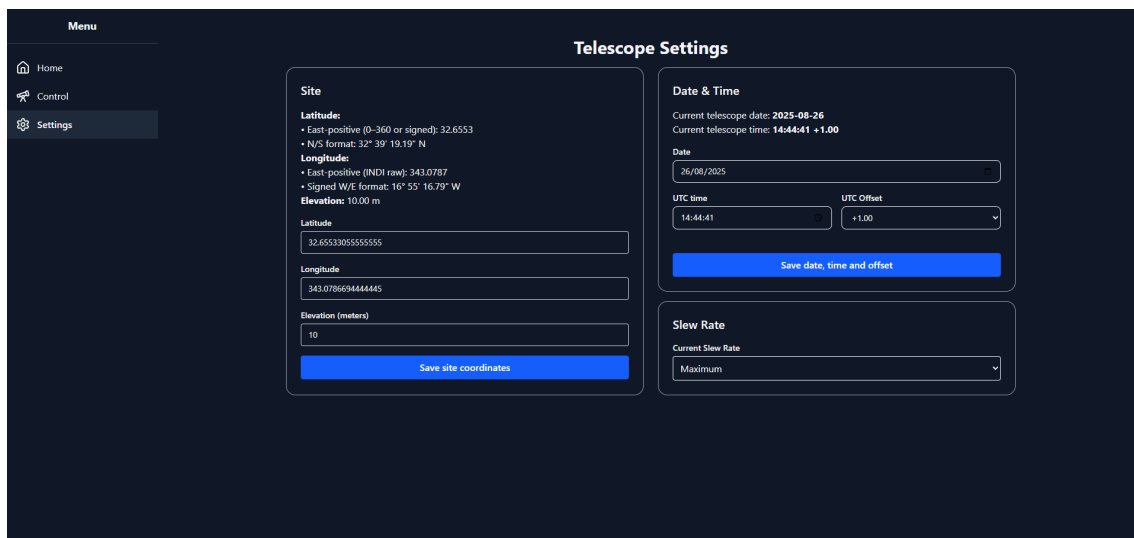


Figure. 11: Proposed telescope control system's settings page

The settings page of the proposed control system (Figure 11) contains actions that are used less frequently compared to telescope movement and object targeting. These options are typically required during the initial setup of the telescope. For clarity purposes, the page is divided into three sections, each representing a different category of settings.

The first section, the one on the left, contains the settings related to site information. Telescopes require their geographic position information in order to correctly map the sky and point to the desired objects. This includes latitude, longitude and elevation. These values most of the times don't change once configured, but if the telescope often changes its location, this setting becomes more frequently used. Additionally, the site data is saved in non-volatile memory, ensuring that it persists across power cycles and system reboots.

The second section, the one on the top-right, focuses on date and time settings. It contains the telescope's current date, time and time offset values, which are crucial for accurately map the sky at that moment. Contrary to the site settings, these parameters must always be filled in the initial setup, since the telescope, or at the least the one used in the project, doesn't retain this information.

The last section, the one on the bottom-right, is destined to change the slew rate of the slews. The slew rate determines the angular speed at which the telescope moves when repositioning. This parameter primarily impacts manual movement controls, enabling users to switch between slow, precise adjustments and faster slewing speeds for covering larger sky areas quickly.

The system provides feedback, through a toast notification, everytime the user saves new values, reassuring the user that the update was successful and to help prevent confusion or redundant re-entry of data.

## 4 Results

### 4.1 Evaluation Protocol

To validate the proposed system, a usability study was conducted. The evaluation had two main objectives:

- Check whether users without previous domain expertise in astronomy could successfully operate the system.
- Collect qualitative and quantitative feedback on the interface’s usability and intuitiveness.

To evaluate the interface, a set of tasks was created for the participants to perform, simulating a possible observation workflow. Five participants, three men and two women, all portuguese, ages ranging from 22 to 55 years old, three with an informatics engineering background while the other two with no technology related background, all with no prior experience in telescope operation or observatory software, participated. The testing was done inside the observatory, near the telescope, since the system was not ready yet for total remote control. Tasks were not measured through quantitative metrics, only qualitative, through observation and the participants’ comments. Participants were provided with minimal context about the system and were told to rely primarily on the interface to complete each task.

To better understand the users’ cognitive processes, the *Think Aloud* method [53] was utilized. Participants were asked to verbalize their thoughts and any difficulties encountered during each task. This method revealed not only where users struggled, but also the causes of confusion, such as ambiguous terminology or unclear visual cues.

The tasks were created to follow a possible observation workflow, while testing the usability of the interface. Even though the tasks created are really straight forward, without much room for possible errors, there are tasks that are designed to create error situations. The task list used during the evaluation was the following:

1. Navigate to the telescope *Settings* page.
2. Indicate the current date, time, and Coordinated Universal Time (UTC) offset of the telescope.
3. Request the correct date, UTC time, and UTC offset from the facilitator. Enter the values in the interface, save them, and report whether the task was successful.

4. Identify the telescope's current slew rate.
5. If not already selected, set the slew rate to the highest available option.
6. Indicate the telescope's current geographic location (latitude, longitude, and elevation).
7. Enter the following values and save them:
  - Latitude: 32°39'19.19" N
  - Longitude: 16°55'16.79" W
  - Elevation: 10 m
8. Navigate to the telescope *Control* page.
9. Indicate the telescope's current pointing coordinates.
10. Move the telescope using the manual control pad and describe what happens.
11. Move the telescope downwards until it points towards the ground. Report what happens.
12. Change the slew rate to  $3x$  and move the telescope again using the manual control pad. Describe what happens.
13. Enter the following coordinates, command the telescope to slew to them, and describe the outcome:
  - Right Ascension: 01:1.0
  - Declination: +62<sup>o</sup> 00' 00"
14. Command the telescope to slew to the star *Cruz*. Report what happens.
15. Command the telescope to slew to the star *Polaris*. Report what happens.
16. Command the telescope to *Park*. Report what happens.

The tasks were designed to cover the following aspects:

1. Navigation through the interface, including the *Settings* and *Control* pages.
2. Retrieval and modification of telescope configuration parameters.
3. Manual and precise telescope movement, including moving to coordinates and named celestial objects.
4. Interpretation of system feedback and error handling.

## 5. Evaluation of interface clarity and discoverability of advanced features.

At the end of the testing session, participants were asked to share their opinion on the system and to complete a questionnaire, the SUS, providing their overall experience with the system.

In addition to user testing, the system was reviewed by the professional astronomer. Due to logistical constraints, the astronomer was unable to participate in live testing. Instead, a demonstration video was prepared, showcasing the system. The astronomer provided qualitative feedback, highlighting strengths, potential mistakes and opportunities for improvement.

## 4.2 Feedback

### 4.2.1 User Suggestions and Resulting Improvements

After task completion, participants were asked to comment on their experience with the system. The feedback was overall positive, but opportunities for improvement were also identified, mainly related to terminology, error messaging, and visual guidance. Due to the small number of participants of the usability tests, each suggestion and comment done by the users was found to be almost mandatory to implement. Additionally, because of time constraints, these changes were not again tested with another usability test session, creating an uncertainty about the actual impact of the changes, if they made sense both in an amateur and professional context, if they represented situations that would rarely happen in a larger group of participants or even with users with different characteristics. The suggestions given by the participants were the following:

- Replace the terms “slew” and “position” simpler ones.
- Add explanatory tooltips for technical terms.
- Improve error feedback for invalid object searches.
- Clarify that the manual control pad can also be operated via keyboard arrows.

Based on the suggestions given, the following improvements were made to the interface:

Figure 12 shows the modifications that were done regarding the user comprehension of the terms “slew” and “position” and interface clarity. The technical term “slew” was replaced with “move” and “position” was updated to “coordinates”, in order to use terminology that is more commonly used. After some analysis, this change could be due to the fact that none of the participants are native English speakers, meaning they are not familiar with the terms, especially the term “slew”.

**Move to Coordinates**

**Current Telescope Coordinates**

Right Ascension: 04:37:34      Azimuth: 354.96°  
 Declination: 57° 33' 26"      Altitude: +0.61°

**Search for object coordinates**

Object name (e.g. Vega, Jupiter, M42)

Load coordinates

**Target Coordinates**

**Right Ascension (HH:MM:SS.SS)**

HH    MM    SS.S

**Declination (+/-DD° MM' SS.SS)**

+ ▾    DD    MM    SS.S

Fill in with current telescope RA/DEC

Move to coordinates      Sync to coordinates

Figure. 12: The terms “slew” and “position” were changed to “move” and “coordinates”, respectively. Additionally, a label for the object coordinates search was added for better clarity.

However, a question arises: is replacing the correct technical terms with more commonly used terms scientifically and professionally acceptable?

Additionally, a descriptive label was added to the object coordinates search field to provide clarify its purpose, reducing potential confusion for users.

The latitude of the observation site. Positive values indicate locations in the Northern Hemisphere, while negative values indicate locations in the Southern Hemisphere.

**Latitude:**

- DMS format: 32° 39' 19.19" N
- Decimal degrees (-90...+90): 32.655°

Figure. 13: Explanatory tooltips were added for technical terms to enhance user understanding.

Users struggled with knowing what each term meant, but through the tasks' description they were able to identify and continue with the test. The addition of explanatory tooltips, illustrated in Figure 13, comes with the purpose of explaining technical terms present in the interface that may be unfamiliar to novice users and consequently reducing possible confusion.

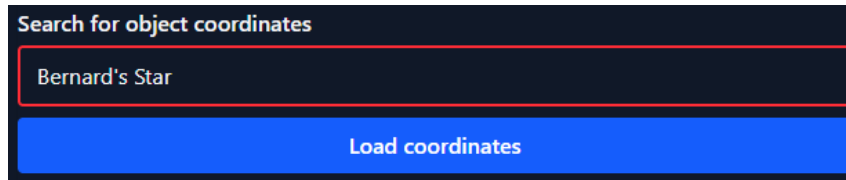


Figure. 14: Improved error feedback was implemented for incorrect object name searches

The improved error feedback, shown in Figure 14, tackles the visual feedback problem when a user entered incorrect astronomical object names. Although toast notifications were implemented, one of the users during the test missed them when they repeatedly wrote the wrong star. They commented that they were focused looking at the search bar, leading them to miss the notification. To help address this, the system now highlights the object name input field with a red border when an error occurs, signaling to the user that the input was not correct or recognized.

None of the users realized that the manual control pad could also be operated using the keyboard arrow keys. This lack of discoverability led the participants to only interact with the directional controls with the mouse. To address this issue, a contextual tooltip that informs the users about the possibility of using the keyboard was added below the manual control pad, as shown in Figure 15.

Finally, a user guiding page, as shown in Figure 16, was added to help new or unexperienced users learn how to use the interface. It is divided in four sections: control, settings, example workflow and tips.

The control section showcases instructions on how to use the functionalities present in the control page. It also details accepted coordinate formats, validation rules and the use of the stop button to abort telescope movement. The settings section guides the user through configuration of the different settings of the settings page and the accepted formats. The example workflow section offers a step-by-step procedure for conducting an observation. Finally, the tips section provides best practices for safe, efficient and precise observations.

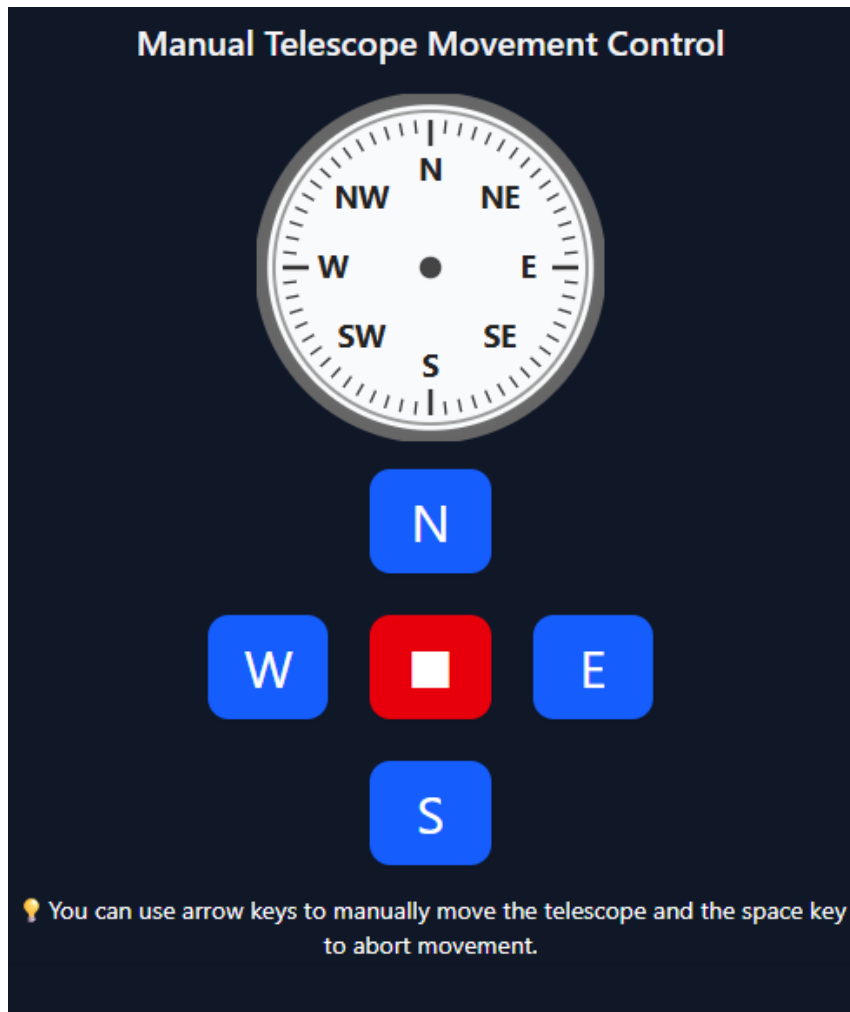


Figure. 15: Improved visual clarification of keyboard support for the manual control pad

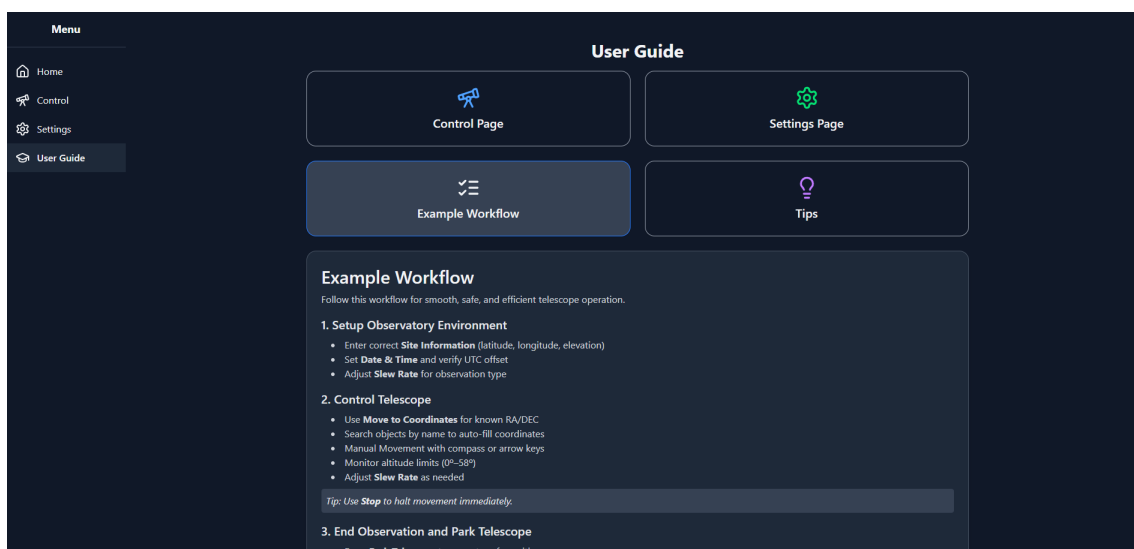


Figure. 16: Newly added user guide page to assist first-time users

### 4.2.2 SUS Results

Following task completion, participants completed the standardized SUS questionnaire. The results were the following:

- Mean SUS Score = 79.5
- Range of Scores = 42.5–92.5

A mean score of 79.5 indicates that the system is above average and generally works well, though there may be room for improvements. This suggests that the interface was overall accepted well and generally intuitive by the users without previous knowledge in astronomy and telescope control.

However, the range of SUS scores (42.5–92.5) suggests that not all the users had the same experience when using the interface. After analysing the users' comments, through direct observation during the tasks and through the SUS results, a relation can be established, where lower scores were typically associated with difficulty in understanding technical terminology and recovering from errors, while higher scores were related to the easiness in learning the interface and understanding the logical organization of the controls.

All in all, the SUS results showcased that the interface was generally intuitive and accessible for unexperienced users, even though there was room for improvement and refinement.

### 4.2.3 Astronomer Feedback

The proposed interface, the same one that the participants of the usability tests used, was also reviewed by the professional astronomer. Overall, the interface was positively received, being well-designed and functionally complete for telescope control. The astronomer only pointed out an error in the site coordinates, where the latitude should have said "North-positive" instead of "East-Positive", as it is displayed in Figure 11.

Beyond the small error, the astronomer emphasized that this interface only represents a part of the requirements needed for real world observation. The proposed interface, in fact, only represents the control of a telescope, whereas in a real work situation, a more complex interface, that not only control the telescope, but controls the rest of the observatory is necessary. While acknowledging that such interface is beyond the scope of this thesis, the astronomer wanted to highlight its importance for future development and astronomy needs.

Despite this limitation, the astronomer identified a possible usage for the proposed interface. Specifically, it could be a component of a web-based interface for remote telescope access, that allows observatory owners to provide observing time to external users without granting them direct access to the local computer connected to the instruments. This use case was considered highly valuable as the astronomer expressed genuine interest in becoming the first user of such a system.

## 5 Discussion

### 5.1 Comparison with the Literature

Similar to the approaches and solutions proposed by the different authors mentioned, this automation proposal aims to create an observatory remote control system, specifically using affordable IoT technologies. The sections 2.2-2.4 expose existing solutions to automate individual subsystems, such as dome rotation using a combination of motors and positional devices like encoders, telescope accuracy control using feedback mechanisms and optimization algorithms or environmental monitoring through different sensors and devices, like weather stations and all-sky cameras. All these projects are really important and appropriate within the objectives and context of automation, but may not provide a full observatory automation solution. On the other hand, this work proposes a possible approach for remote observatory control, through an IoT and INDI-based framework. Additionally, its modular architecture can be used as a reference blueprint for any astronomer or institution that wants to automate their observatory, even if just a part of it.

Professional observatory automation solutions usually have high-quality, high-precision hardware that is supported by vendors, facilitating the overall automation process by reducing integration and maintenance efforts, but with increased costs. Alternatively, low-cost and DIY projects, such as the ones showcased in the section 2.10, decrease the financial costs, but require more knowledge and time for the automation, since they have to take in consideration more details, like wiring complexity, synchronization of the equipment and long-term reliability. The proposed design also follows the affordable and cost-efficient ideals by using consumer-grade components, but also with the downside of needing technical expertise during setup and for maintenance of the system.

Similarly to interfaces described in the section 2.8, the proposed system adopts a web-based telescope approach. This approach offers two advantages: (i) it allows observatory owners to give access to their telescope without exposing the host computer or local network; and (ii) download of any software is required by the end users, lowering the entry barrier. This is important for collaborative contexts, like schools, where telescope time is shared among the students across different locations.

However, this proposed approach also has its constraints. In this solution, the telescope must have a network communication interface in order to be able to communicate with the web interface. Even though, telescope mounts that support Ethernet or Wi-Fi are becoming more common, legacy

systems may require additional middleware to provide the same functionalities. Furthermore, other works have looked into more advanced topics. For example, section 2.5 describes projects with weather sensitive safety mechanisms, while section 2.6 includes projects that have used plate-solving techniques for automated star alignment. These capabilities are not present in this initial automation proposal, but remain important areas for future development. Despite these limitations, the modular low-cost architecture combined with INDI provides flexibility and scalability, allowing the system to be easily extendable with new equipment.

## 5.2 Study limitations and Room for Improvement

In telescope operation with the physical presence of a person, the hand controller is the main interface for telescope control. During startup, the controller asks the user to input the date, time and daylight savings status. With this information, combined with the stored site data, the telescope calculates its position in the sky and creates a reference frame for the following slewing operations. However, replicating the same input initialization process through the designed revealed to be problematic. More specifically, when the same parameters asked by the hand controller during initialization were transmitted through the proposed interface, the telescope failed to correctly recalculate its position in the sky. As a result, the telescope had a wrong reference frame of the sky and the telescope's position, which caused slewing actions to point to incorrect areas of the sky and, in extreme cases, the telescope would move towards dangerous positions, risking colliding and damaging the equipment. Consequently, it became clear that the initialization of the telescope had to be done using the telescope's hand controller in order for the telescope to work correctly, which compromises the full remote control objective of the project. This represents a critical area that requires further investigation, but due to time constraints it was not possible to explore it in greater depth.

Another limitation is that the telescope still requires a manual alignment procedure before it can be controlled remotely. The alignment consists of pointing the telescope towards reference stars, allowing the system to calibrate its internal coordinates, which requires the presence of a person. At the moment, this action can't be done by the system without user intervention.

In addition, even though the INDI driver used for this telescope provides the ability to change the telescope's mount configuration, such feature was not implemented, because the current observatory hardware and control logic are designed specifically for an altazimuth mount configuration.

Supporting alternative mount types would require not only software modifications but also significant changes in the telescope's physical configuration.

Finally, the telescope focusing feature remains a limitation. While the INDI driver exposes focuser properties, it was not implemented due to time constraints.

### 5.3 Future Work

This dissertation provides a blueprint for a possible automation architecture of the OAUMa and a web-based telescope control interface. Although the system developed shows the viability of an IoT-based architecture integrated with INDI, there are still areas that need to be improved.

The first area of improvement is related to automated alignment and calibration. As mentioned previously, at the moment, telescope initialization and alignment require manual intervention. Introducing alignment strategies, such as plate-solving techniques, can be used to automatically identify the telescope's orientation by analyzing the sky.

Related to this is the challenge of telescope initialization. As mentioned before, the proposed system showed occurrences of misalignment when trying to replicate the initialization process. Future work should investigate possible solutions to this problem, ensuring that coordinate synchronization of the telescope is reliable across observation sessions.

Another existing concern is related to the support for different mount types and focusing capabilities. The system was developed using an alt-azimuth mount configuration, but it should be extended to support other mounts, like the equatorial mount that would require the addition of an equatorial wedge to the telescope's base. For the focusing, the system should add focuser control, existent in the INDI driver properties, to the interface, so it contains all the basic telescope control functionalities.

Expansion of observatory functions is also essential. Besides developing a broader interface that controls the whole observatory instead of just the telescope, other functionalities are important to add, such as scheduling. Observation scheduling would allow users to predefine targets and times, transforming the observatory into a semi-autonomous system capable of executing observation sessions with minimal human intervention. Additionally, providing collaboration features by allowing multiple users with role-based access control could be useful for public observation opportunities and for educational purposes.

Ultimately, it is important to note that the current system represents only the initial stage of the proposed automation approach. While this dissertation has focused on a telescope control interface and on an automation architecture blueprint, the full implementation of the architecture remains future work.

## 6 Conclusion

The dissertation describes the initial design and implementation of a remote control system for the OAUMa. The work showcased an architectural blueprint of a possible observatory automation, as well as a web-based interface for telescope control.

This work provides four main contributions. First, it describes an observatory remote control architecture that uses IoT technologies, integrated into subsystems controlled by different MCUs and the INDI communication protocol. Second, it explains the decisions behind the process of the technologies chosen for telescope communication and control through three different tests. Third, it showcased the functional web interface developed for both experienced and unexperienced users. Finally, it validates the architecture through feedback received by the professional astronomer accompanying the project during the dissertation and the first interaction with developed interface through the feedback from both the astronomer and the participants of the usability tests.

The project only represents the initial design for transforming the OAUMa into a remotely accessible observatory. Future work should fix the problems found during the design process of this system, as well as improve and expand the system to be able to fully remotely control the observatory.

In conclusion, this dissertation offers an automation architecture blueprint for how observatories may be operated in a remote, cost-effective and scalable manner. By combining low-cost technologies with standardized communication protocols, it contributes not only to the revitalization of the OAUMa but also by making astronomy more accessible.

## References

- [1] P. Halpern and N. Tomasello, "Size of the observable universe," *Advances in Astrophysics*, vol. 1, no. 3, pp. 135–137, 2016. [Online]. Available: [https://www.researchgate.net/profile/Nick-Tomasello/publication/311954421\\_Size\\_of\\_the\\_Observable\\_Universe/links/59f26eda0f7e9beabfcc63a8/Size-of-the-Observable-Universe.pdf](https://www.researchgate.net/profile/Nick-Tomasello/publication/311954421_Size_of_the_Observable_Universe/links/59f26eda0f7e9beabfcc63a8/Size-of-the-Observable-Universe.pdf)
- [2] F. E. Roach, *The light of the night sky*. Springer Science & Business Media, 2013, vol. 8.
- [3] D. Leverington, *Observatories and telescopes of modern times*. Cambridge University Press, 2017.
- [4] J.-P. Luminet, J. R. Weeks, A. Riazuelo, R. Lehoucq, and J.-P. Uzan, "Dodecahedral space topology as an explanation for weak wide-angle temperature correlations in the cosmic microwave background," *Nature*, vol. 425, no. 6958, pp. 593–595, 2003. [Online]. Available: <https://www.nature.com/articles/nature01944>
- [5] R. A. B. D. Cruz and P. Tucio, "Review of potential astronomical optical observatory sites in the philippines," in *IOP Conference Series: Earth and Environmental Science*, vol. 880, no. 1. IOP Publishing, 2021, p. 012008. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1755-1315/880/1/012008/meta>
- [6] P. Augusto, "Madeira: 300 years of an astronomical site," *Journal for the History of Astronomy*, vol. 55, no. 1, pp. 47–75, 2024. [Online]. Available: <https://journals.sagepub.com/doi/full/10.1177/00218286231220367>
- [7] M. Ismail, S. Elbanna, M. A. El-Dabah, and F. Elnagahy, "A complete affordable control system for remote astronomical observing accompanied by an intelligent controller," *IEEE Access*, vol. 10, pp. 97217–97236, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9887914>
- [8] T. Brothers, "45 years of telescope automation at mit's wallace astrophysical observatory," *RTSRE Proceedings*, vol. 1, no. 1, 2018. [Online]. Available: <https://rtsre.org/index.php/rtsre/article/view/37>

- [9] J. Colomé, I. Ribas, X. Francisco, K. Casteels, D. Fernández, J. Isern, X. Palau, and J. Torra, “The oadm robotic observatory,” *Advances in Astronomy*, vol. 2010, no. 1, p. 183016, 2010. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1155/2010/183016>
- [10] L. R. Sonsa, F. Morgado-Dias, and P. Augusto, “Low cost automated astronomical observatory for remote operation: A requisite analysis,” in *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. IEEE, 2017, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8008969>
- [11] P. Cuartas-Restrepo, N. Gaviria-Gómez, and J. Galvez-Serna, “Design of a low cost control system for an astronomical observatory,” *Ingeniería y Ciencia*, vol. 16, no. 31, pp. 103–127, 2020. [Online]. Available: [http://www.scielo.org.co/scielo.php?pid=S1794-91652020000100103&script=sci\\_arttext&tlng=en](http://www.scielo.org.co/scielo.php?pid=S1794-91652020000100103&script=sci_arttext&tlng=en)
- [12] S. T. Gutiérrez, C. I. Fuentes, and M. A. Díaz, “Introducing sost: An ultra-low-cost star tracker concept based on a raspberry pi and open-source astronomy software,” *IEEE Access*, vol. 8, pp. 166 320–166 334, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9179736>
- [13] T. S. Kumar and R. N. Banavar, “Design and development of telescope control system and software for the 50/80 cm schmidt telescope,” *Optical Engineering*, vol. 52, no. 8, pp. 081 607–081 607, 2013. [Online]. Available: <https://www.spiedigitallibrary.org/journals/optical-engineering/volume-52/issue-8/081607/Design-and-development-of-telescope-control-system-and-software-for/10.1117/1.OE.52.8.081607.full>
- [14] G. S. A. Robin, A. K. Shukla, and R. Jolivot, “Remotely-controlled low-cost digital telescope,” *Engineering Journal*, vol. 23, no. 3, pp. 141–156, 2019. [Online]. Available: <https://www.engj.org/index.php/ej/article/view/3003>
- [15] R. Suszynski, “Stand-alone station for deep space objects astrophotography,” in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*. IEEE, 2009, pp. 333–336.
- [16] M. K. Srivastava, A. Ramaprakash, M. P. Burse, P. A. Chordia, K. S. Chillal, V. B. Mestry, H. K. Das, and A. A. Kohok, “Telics—a telescope instrument control system for small/medium sized astronomical observatories,” *Publications of the*

- Astronomical Society of the Pacific*, vol. 121, no. 884, p. 1112, 2009. [Online]. Available: <https://iopscience.iop.org/article/10.1086/644648/meta>
- [17] A. Kumar, A. S. Sengupta, and S. Ganesh, “Autonomous dome for robotic telescope,” *arXiv preprint arXiv:1604.08370*, 2016.
- [18] M. Reefer, O. Alfaro, S. Foster, P. Plavchan, N. Pepin, V. Banaji, M. Vidaurri, S. Webster, S. Banaji, J. Berberian *et al.*, “Asynchronous object-oriented approach to the automation of the 0.8-meter george mason university campus telescope in python,” *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 8, no. 2, pp. 027002–027002, 2022.
- [19] J. Han, C. Wang, D. Fan, C. Cui, S. Li, L. Mi, Z. Li, Y. Xu, B. He, C. Li *et al.*, “Amateur public observatory i: The observatory and hardware integration system,” *Astronomy and computing*, vol. 25, pp. 89–93, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2213133718300209>
- [20] Z.-Y. Wang, G.-Y. Zhang, J. Wang, Z. Geng, Z.-Y. Zhu, M.-H. Jia, Q. Zhang, Z.-H. Zheng, K. Ge, J.-Y. Gu *et al.*, “Racs2: a framework of a remote autonomous control system for telescope observation and its application,” *Frontiers in Astronomy and Space Sciences*, vol. 10, p. 1104150, 2023. [Online]. Available: <https://www.frontiersin.org/journals/astronomy-and-space-sciences/articles/10.3389/fspas.2023.1104150/full>
- [21] P. Hickson, “Ocs: A flexible observatory control system for robotic telescopes with application to detection and characterization of orbital debris,” in *First International Orbital Debris Conference, Sugarland, Texas*, 2019. [Online]. Available: <https://www.hou.usra.edu/meetings/orbitaldebris2019/orbital2019paper/pdf/6066.pdf>
- [22] A. J. Castro-Tirado, “Robotic autonomous observatories: a historical perspective,” *Advances in Astronomy*, vol. 2010, no. 1, p. 570489, 2010. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1155/2010/570489>
- [23] T.-O. Husser, F. V. Hessman, S. Martens, T. Masur, K. Royen, and S. Schäfer, “pyobs-an observatory control system for robotic telescopes,” *Frontiers in Astronomy and Space Sciences*, vol. 9, p. 891486, 2022. [Online]. Available: <https://www.frontiersin.org/journals/astronomy-and-space-sciences/articles/10.3389/fspas.2022.891486/full>

- [24] D. E. Rosenberg, R. O. Gray, J. Mashburn, A. W. Swenson, C. E. McGahee, and M. M. Briley, “Robotic spectroscopy at the dark sky observatory,” *Publications of the Astronomical Society of the Pacific*, vol. 130, no. 988, p. 064503, 2018.
- [25] G.-y. Zhang, J. Wang, P.-y. Tang, M.-h. Jia, J. Chen, S.-c. Dong, F. Jiang, W.-q. Wu, J.-j. Liu, and H.-f. Zhang, “An autonomous observation and control system based on epics and rts2 for antarctic telescopes,” *Monthly Notices of the Royal Astronomical Society*, vol. 455, no. 2, pp. 1654–1664, 2016. [Online]. Available: <https://academic.oup.com/mnras/article/455/2/1654/1106707>
- [26] D. H. Kasper, T. G. Ellis, R. R. Yeigh, H. A. Kobulnicky, H. Jang-Condell, M. Kelley, G. J. Bucher, and J. S. Weger, “Remote operations and nightly automation of the red buttes observatory,” *Publications of the Astronomical Society of the Pacific*, vol. 128, no. 968, p. 105005, 2016.
- [27] J.-N. Yoon, Y. S. Lee, C.-H. Kim, Y. Kim, H.-S. Yim, W. Han, and J. H. Jeong, “Statistical analysis on weather conditions at chungbuk national university observatory in jincheon, korea,” *Journal of Astronomy and Space Sciences*, vol. 29, no. 4, pp. 397–405, 2012.
- [28] T. K. Sharma, P. Parihar, R. K. Banyal, M. M. Kemkar, U. Stanzin, and G. Anupam, “Infrared scanning cloud monitor for the indian astronomical observatory, hanle,” *Journal of Astronomical Instrumentation*, vol. 6, no. 04, p. 1750008, 2017.
- [29] R. Honeycutt, B. Adams, D. Swearingen, and W. Kopp, “Devices for observatory automation,” *Publications of the Astronomical Society of the Pacific*, vol. 106, no. 700, p. 670, 1994.
- [30] I. Selim, N. Ahmed, and M. Lashin, “Automated of 74-inch kottamia telescope dome based on fuzzy logic due to online weather station data,” *International Journal of Computer Applications*, vol. 975, p. 8887, 2016.
- [31] A. Castellón, G. Larrubia, R. Castillo, A. J. Castro-Tirado, E. Fernández-García, C. Pérez del Pulgar, A. Reina, and S. Castillo-Carrión, “Dome control with fuzzy logic,” *Frontiers in Astronomy and Space Sciences*, vol. 9, p. 924871, 2022.
- [32] P. Márquez, O. Martínez, M. Gaug, and J. M. Miranda, “Risk assessment of environmental conditions affecting the useful lifetime of cherenkov telescopes at el roque de los muchachos

- observatory,” *Publications of the Astronomical Society of the Pacific*, vol. 134, no. 1039, p. 095002, 2022.
- [33] G. Lim, D. Kim, S. Lim, M. Im, H. Choi, J. Park, K.-H. Park, J. Park, C. Muskaan, D. Kim *et al.*, “The robotic maa0 0.7 m telescope system: Performance and standard photometric system,” *Publications of the Astronomical Society of the Pacific*, vol. 136, no. 5, p. 055001, 2024.
- [34] X. Li, B. Wang, B. Qiu, and C. Wu, “An all-sky camera image classification method using cloud cover features,” *Atmospheric Measurement Techniques*, 2022. [Online]. Available: <https://amt.copernicus.org/articles/15/3629/2022/amt-15-3629-2022.html>
- [35] M. Mommert, “Cloud identification from all-sky camera data with machine learning,” *The Astronomical Journal*, vol. 159, 2020. [Online]. Available: <https://iopscience.iop.org/article/10.3847/1538-3881/ab744f/meta>
- [36] I. Price, J. Nielsen, C. Lidman, J. Soon, T. Travouillon, and R. Sharp, “Converting the anu 2.3 telescope to fully automated operation,” *Publications of the Astronomical Society of Australia*, vol. 41, p. e057, 2024.
- [37] S. Tulloch and V. Dhillon, “On the use of electron-multiplying ccds for astronomical spectroscopy,” *Monthly Notices of the Royal Astronomical Society*, vol. 411, no. 1, pp. 211–225, 2011.
- [38] A. Mansour, F. Elnagahy, and M. Ismail, “An implementation of remotely automated control to the 14-inch telescope system,” in *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, 2020, pp. 1–6.
- [39] P. Ghadekar, O. Borude, V. Gavali, D. Gholap, A. Mali, and C. Rathad, “Artificial intelligence-guided telescope control system using jetson-nano,” in *2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, vol. 1. IEEE, 2024, pp. 54–59.
- [40] W. Zhou, B. Li, C. Rao, J. Huang, J. Qian, P. Jiang, and X. Li, “High accuracy eliminating image rotation control system for optical telescope,” *Control Engineering Practice*, vol. 161, p. 106342, 2025.

- [41] I. Helmy, A. Hamdy, D. Eid, and A. Shokry, “Autofocusing optimal search algorithm for a telescope system,” *Journal of Astronomical Instrumentation*, vol. 10, no. 03, p. 2150012, 2021.
- [42] K. Watanabe, M. Otani, H. Tanaka, and M. Okyudo, “The simple and compact remote telescope system,” in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, 2010, pp. 1129–1134. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/5447285>
- [43] P. Bartkowiak, R. Patelski, M. Kwiatkowska, and D. Pazderski, “System architecture for development and supervision of robotic astronomical telescope,” *Pomiary Automatyka Robotyka*, vol. 26, 2022.
- [44] T. Husser and F. Hessman, “Using open source software and open standards for operating robotic telescopes,” in *Revista Mexicana de Astronomia y Astrofisica Conference Series; SAO/NASA Astrophysics Data System: Huelva, Spain*, vol. 51, 2019, pp. 15–23. [Online]. Available: [https://www.astroscu.unam.mx/rmaa/RMxAC..51/PDF/RMxAC..51\\_articulo06.pdf](https://www.astroscu.unam.mx/rmaa/RMxAC..51/PDF/RMxAC..51_articulo06.pdf)
- [45] L. Suchockas, “Automating data management system of 1.5-meter telescope at tartu observatory,” *Baltic Astronomy*, vol. 29, no. 2, pp. 179–188, 2020. [Online]. Available: <https://www.ims.ut.ee/www-public2/at/2020/msc/atprog-courses-magistrit55-loti.05.036-laimonas-suchockas-text-20200520.pdf>
- [46] Z. Wang, Y. Tian, J. Li, Z.-H. Cao, and Y.-H. Zhao, “A study on universal observation control system and its application for lamost,” *Research in Astronomy and Astrophysics*, vol. 21, no. 6, p. 149, 2021. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1674-4527/21/6/149/meta>
- [47] T. Franks, D. Hake, D. Babcock, K. Meah, T. Ullery, D. McHugh, and D. Herr, “Software design, development, and implementation for an automated radio telescope,” in *2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. IEEE, 2023, pp. 116–121.
- [48] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, “Seti@ home: an experiment in public-resource computing,” *Communications of the ACM*, vol. 45, no. 11, pp. 56–61, 2002. [Online]. Available: <https://dl.acm.org/doi/fullHtml/10.1145/581571.581573>

- [49] A. J. Calderwood, R. A. Pauloo, A. M. Yoder, and G. E. Fogg, “Low-cost, open source wireless sensor network for real-time, scalable groundwater monitoring,” *Water*, vol. 12, no. 4, p. 1066, 2020. [Online]. Available: <https://www.mdpi.com/2073-4441/12/4/1066>
- [50] S. Hobbs, D. Paull, and J. Clarke, “Marsobot: design and performance characterization of a low-cost, ground-based multispectral camera on an open source rover,” *International Journal of Remote Sensing*, vol. 38, no. 12, pp. 3635–3655, 2017. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/01431161.2017.1302109>
- [51] T. Aidukas, R. Eckert, A. R. Harvey, L. Waller, and P. C. Konda, “Low-cost, sub-micron resolution, wide-field computational microscopy using opensource hardware,” *Scientific reports*, vol. 9, no. 1, p. 7457, 2019. [Online]. Available: <https://www.nature.com/articles/s41598-019-43845-9>
- [52] J. M. Branning Jr, K. A. Faughnan, A. A. Tomson, G. J. Bell, S. M. Isbell, A. DeGroot, L. Jameson, K. Kilroy, M. Smith, R. Smith *et al.*, “Multifunction fluorescence open source in vivo/in vitro imaging system (openivis),” *Plos one*, vol. 19, no. 3, p. e0299875, 2024. [Online]. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0299875>
- [53] M. W. Van Someren, Y. F. Barnard, J. A. Sandberg *et al.*, “The think aloud method: a practical approach to modelling cognitive processes,” *London: AcademicPress*, vol. 11, no. 6, 1994.