

可再生能源实验室气象站

初始设计报告模板

董晨希：CAD 工程师

Rowan McCullough：测试工程师兼财务经理

Ian Torp：项目经理

王疏桐：制造工程师

2025 年夏季 – 2025 年秋季



项目发起人/讲师：Carson Pete

免责声明

本报告由学生编写，是大学课程要求的一部分。虽然该项目已投入大量精力，但它并非注册工程师的成果，也未经过该行业常见的广泛验证。未经彻底、独立的测试和验证，请勿依赖或使用本报告中的信息、数据、结论和内容。大学教职员工可能以顾问、赞助人或课程讲师的身份参与了该项目，但他们对结果或结论的准确性不承担任何责任。

执行摘要

北亚利桑那大学可再生能源实验室委托我们设计一个用于学术用途的气象站。Carson Pete 教授向我们提出了方案，并明确了他的要求。该气象站将收集六种不同类型的气象数据，并允许随时远程访问。这六种不同的气象数据类型包括降雨量、气温、气压、湿度、太阳辐照度和风速。客户唯一可能要求我们不要收集的气象数据是降雨量数据。随着项目的推进，我们将与客户共同决定是否收集降雨量数据。

气象站将把所有传感器的输出数据输入到位于可再生能源实验室内的 Raspberry Pi 中。这台 Raspberry Pi 是实验室提供的，型号是 Raspberry Pi 3+。它将收集来自传感器的所有数据，然后将原始数据转换为可读数据。由于每个传感器的校准方式不同，每个传感器都需要在 Pi 中编写自己的代码来获取可读数据。有些传感器可能需要使用表格来解释数据，而有些传感器可能只需要一个公式。一旦数据可读，我们决定为每个传感器的数据创建10分钟平均值，然后将其转换为60分钟平均值，并上传到数据库和网站。这样做是为了避免Raspberry Pi存储大量数据，同时提供准确的读数。每个10分钟平均值也将上传到网站，替换之前的平均值，以便用户访问实时天气数据。

至于结构设计，实验室为我们提供了一个高大的金属桁架，我们可以在上面安装传感器。桁架顶部的平台将使用菱形横梁，并在平台顶部安装对角支撑。除风速计外，所有传感器都将安装在这个高度，以达到各自的标准。风速计将通过吊臂连接到一个单独的现有塔架上。此设计目前不支持安装雨量计，但如果客户要求，安装起来并不困难。桁架顶部的每个传感器将安装在菱形横杆的顶点，以保持三个接触点，确保安装稳定。

撰写本文时，该项目尚处于设计阶段，尚未完成任何施工或原型设计。下文将对每个感兴趣的项目进行详细说明。我们目前的预算包括约 3000 美元的现有设备、客户提供的 500 美元额外费用以及预计的 300 美元筹款。目前可预见的成本将用于制作用于安装传感器的金属横杆以及为 Raspberry Pi 制作一张新的 SD 卡。

目录

内容

免责声明	2
执行摘要	3
目录	4
1 背景.....	5
1.1 项目描述	5
1.2 交付成果	5
1.3 成功指标	6
2 要求.....	6
2.1 客户要求 (CRs)	6
2.2 工程要求(ERs)	7
2.3 品质功能展开图 (QFD)	8
3 在您的设计空间内进行研究	9
3.1 基准对比分析	9
3.2 文献综述	9
3.2.1	9
3.2.2	11
3.2.3	13
3.2.4	14
3.3 数学建模	15
3.3.1.....	15
3.3.2.....	16
3.3.3.....	16
3.3.4.....	17
4 设计理念.....	17
4.1 功能分解	17
4.2 概念生成	20
4.3 选择标准	25
4.4 概念选择	25
5 结论.....	27
6 参考.....	28
7 附录.....	30
7.1 附录 A: 伪代码	30

1 背景

本报告的这一章将讨论该项目的初始重要性。第一部分将描述项目提案并解释该项目的重要性。它还将讨论项目的预算和筹款目标。下一节将讨论课程的主要交付成果，涵盖项目的每个截止日期。本章的最后一部分将讨论该项目的成功指标。这将包括我们的目标以及项目总体的主要设计要求。

1.1 项目描述

我们受命为北亚利桑那大学 (NAU) 的可再生能源实验室 (RE Lab) 组装一个气象站。该气象站需要提供最新、准确的气象数据读数，供学术和科研使用。可再生能源实验室过去曾配备有正常运行的气象数据采集器，但如今已无法使用。此外，之前的采集工作需要教授亲自前往可再生能源实验室读取当前数据。当时既无法远程访问这些数据，也无法访问之前的资料。我们的目标是解决这个问题。我们的项目将收集数据，并进行多年的在线展示和存储。这将减轻我们从可再生能源实验室收集气象数据的麻烦。虽然我们的气象站无法提供未来天气事件的预报，但收集到的数据足以进行此类判断。气象站是重要的设施，通常用于预测未来天气事件、进行灾害防备和提供农业信息。我们的气象站将收集温度、降雨量、湿度、气压、太阳辐射和风速等数据。我们的项目最有影响力的课程是北亚利桑那大学的可再生能源课程和风能课程。学生和教师将能够访问我们的数据进行计算和其他审议。

我们为这个项目预算了约 3000 美元的现有设备费用，客户额外提供的 500 美元用于支付潜在的额外费用，以及团队预计的 300 美元筹款。团队已经可以使用这些设备，目前还没有遇到任何意外费用。可能需要支付的费用包括更换树莓派的 SD 卡。目前的 SD 卡功能正常，但我们可能需要更换它才能访问树莓派。这笔费用大约在 6 美元到 50 美元之间。由于筹款目标很低，如果团队无法找到外部赞助商，他们准备自行提供资金。

1.2 交付成果

该项目旨在交付一个功能齐全的模块化气象站，以满足北亚利桑那大学 (NAU) 附近可再生能源实验室的需求。该气象站将用于测量六个关键环境变量：温度、湿度、风速、气压、太阳辐照度和降雨量。最终交付成果包括：

- 一个完全组装的太阳能气象站，使用 Raspberry Pi 进行数据处理和无线传输。
- 通过模块化安装和标准化接口实现稳健的传感器集成，并遵循世界气象组织的选址指南。
- 一个基于云的数据显示仪表板，使用 Adafruit IO 构建，能够显示实时和历史测量值。
- 传感器校准模型以及至少一个有效的数学建模示例（例如，湿度到电压的转换或风速计算）。

- 用 Python 编写的自定义数据管理代码，用于读取、处理、平均和上传传感器值。
- 完整的传感器支撑结构 CAD 设计，关注维护、传感器更换和防风雨性能。
- 一份详细的技术报告，概述设计流程、基准比较、概念生成、建模和测试结果。

1.3 成功指标

该项目的成功将根据其功能性能及其与报告中所述工程目标的契合程度进行评估。评估标准如下：

- 测量精度：所有传感器输出在测试条件下均应符合制造商规定的精度标准。
- 数据正常运行时间：气象站应能够持续收集和传输数据，在 24 小时测试期内至少有 90% 的数据成功上传。
- 模块化和维护：传感器必须可单独更换，并在发生故障时可独立验证。
- 在线可视化：所有六个变量必须清晰地显示在 Adafruit IO 仪表板上，并带有适当的单位和标签。
- 结构稳定性：塔架和传感器支架必须能够承受弗拉格斯塔夫典型的中等强度的户外风雨和温度变化。
- 文档质量：所有代码、CAD 模型和校准方法必须清晰地组织并以最终报告格式提交。

2 要求

本节将概述气象站项目的客户需求和工程要求。客户需求由我们的客户提供，并为每个需求分配了重要级别。工程要求由团队确定，并可量化。这些需求用于创建质量功能展开 (QFD)，用于比较和识别彼此之间的关系，并帮助我们确定主要关注点。此外，QFD 还用于评估基准，以评估它们在多大程度上能够满足我们项目的需求。

2.1 客户要求 (CRS)

关键气象参数测量 - 气象站将测量温度、湿度、风速和风向、气压以及太阳辐射。也可能测量降雨量。

- 数据传输 - 收集的数据将通过互联网传输。
- 远程数据访问 - 实时数据和存储数据应可通过网页界面访问。
- 可再生能源 - 任何需要供电的组件都必须使用太阳能供电。
- 耐候性 - 气象站必须能够承受室外恶劣天气条件。
- 低维护 - 气象站每年的维护时间应少于2小时。

- 传感器精度 - 传感器读数应经过正确校准。
- 用户友好 - 用户界面应易于操作。

2.2 工程要求 (ERS)

- 长期数据存储 - 数据库应以有序的方式记录4年内的数据。
- 提高数据准确性 - 传感器读数应高度准确，误差不超过3%。数据平均值计算应准确。
- 多次风速读数 - 应在标准高度和实验室现有塔顶测量风速和风向，至少提供两次读数。
- 按照行业标准测量 - 传感器应根据行业标准正确放置。
- 正确校准 - 传感器应经过正确校准，以便从原始读数上传准确的数据，误差不超过真实值的3%。
- 所有数据类型的测量 - 监测站应记录5种数据类型，包括温度、气压、湿度、太阳辐射和风力数据。
- 低功耗要求 - 监测站应能够在实验室现有的太阳能发电条件下完全运行，目标功耗为每天0.2千瓦时或更低。

2.3 品质功能展开图(QFD)

System QFD				Project: RE Weather Station QFD							
				Date: 6/16/2025							
1	Long Term Data Storage										
2	Increased Data Accuracy			1							
3	Multiple Windspeed Readings			1	3						
4	Measured at Industry Standards			1	9	9					
5	Proper Callibration			3	9	3	9				
6	Temp, Pressure, Humidity, Wind spd/dir, Solar Irradiance			3	9	3	9	9			
7	Low Power Requirement			3	1	1	1	1	9		
				</							

3 在您的设计空间内进行研究

3.1 基准对比分析

我们选取的第一个对标对象是 Flagstaff Pulliam 机场气象站 [7]。该气象站通过 Weather.gov 网站公开数据，并提供该地区的天气预报。虽然我们的气象站不提供未来天气预报，但将能够实时提供准确的气象数据。机场气象站的数据包括温度、湿度、风速、气压、露点、能见度和体感温度（Heat Index），还显示其经纬度与海拔高度，并明确标注数据的更新时间（每 30 分钟更新一次）。此外，它提供了三天和七天的历史记录，这也是我们希望在自己的网站中实现的功能。该网站界面清晰、用户友好，是我们团队在网站设计中重视的特性之一。它通过折线图和表格来展示采集到的所有气象数据，这也正是我们计划采用的展示方式。因此，该气象站将作为我们网站设计的参考蓝本，也可作为我们气象数据准确性的校验标准。如果我们的数据与其偏差较大，则需要重新评估我们的测量方法。

第二个对标对象是 Tempest 家用气象站 [12]，这是一款市售的消费级产品，售价约为 350 美元。该设备可测量我们多数计划采集的气象参数，并配套一款手机 App，可以实时查看设备数据及 10 天天气预报。尽管其价格合理且设备便携，但该设备似乎并未配备足够的传感器来实现其声称的全部测量功能。例如，该设备外观中并无任何风速计或风向标，却声称可以测得风速和风向。此外，该 App 虽然用户界面友好、信息丰富，但不支持历史数据记录功能。一些用户评论认为，其部分“测量值”可能并非真实由设备获取，而是从其他来源调用的数据。无论其数据准确性如何，这款性价比较高的产品为我们提出了明确的目标：我们需要在性能或功能方面实现超越，才能证明我们设计的合理性与价值。

第三个对标对象是 Davis Vantage Pro2 气象站，这是一款高端产品，广泛应用于农业、科研和高校等领域。它的主要特点包括：集成温度、湿度、风速等多种传感器；支持最远 300 米的无线数据传输；采用太阳能与电池双供电，适合野外部署。设备采用塔式结构，风速计安装在顶部，以减少干扰，并支持紫外线等模块的拓展。其模块化设计与高稳定性表现，与我们项目的目标高度一致，是我们系统架构与性能设计的重要参考对象。

3.2 文献综述

3.2.1 Ian Torp

[1] Setra Systems, Inc, “Barometric Pressure Sensors / Setra Systems,” Setra.com, 2025. <https://www.setra.com/product/pressure/barometric> (accessed Jun. 18, 2025).

这个网站是气压传感器的销售平台。为了完成我们的项目，我们专门用它来帮助确定压力传感器的一些基准。它展示了5种不同的型号，并根据压力接头、电气端子、输出、介质兼容性、热效应、补偿温度范围、工作温度、精度满量程（FS）、PSI 量程、传感技术和示例应用等标准对它们进行了比较。当我们开始调试我们提供的气压传感器时，这些信息将更加有用。该网站还讲解了一些气压传感器的基础知识，这有助于我们明确该传感器的重要性。

[2] N. US Department of Commerce, “Standards and Policy,” [www.weather.gov. https://www.weather.gov/coop/standards](https://www.weather.gov/coop/standards).

这是一个政府运营的网站，拥有许多实用资源。本项目使用的资源是 NWSI 10-1302: NWS 气候观测的要求和标准。该文件讨论了气象站的场地和暴露标准，以及更具体的气温测量标准和降水测量标准。它还讨论了土壤温度测量标准和蒸发皿蒸发测量标准，尽管这些信息对本项目来说并非有用。我们在这里了解到，温度计必须能够测量-20华氏度到115华氏度之间的温度，降水测量必须能够达到20英寸（约51厘米）。对每个传感器标准的讨论为我们在整个设计过程中的决策提供了参考。

[3] World Meteorological Organization, *Guide to meteorological instruments and methods of observation., Volume 1*. Geneva, Switzerland: World Meteorological Organization, 2008.

这是我迄今为止最重要的资料来源。本书的每一章都阐述了一种不同的测量方法及其标准。每一章都讨论了用于测量每种气象特征的不同传感器和工具。本书深入探讨了从标准到校准的方方面面。我已经用完了第一章到第七章。这些章节依次为：概述、温度测量、气压测量、湿度测量、地面风测量、降水测量和辐射测量。这是一个至关重要的资料来源，它将继续帮助我们理解和发展气象站。

[4] ISO-CAL, “What is a Pyranometer? 10 Important Points to Consider.,” [isocalnorthamerica.com, Jan. 16, 2023. https://isocalnorthamerica.com/what-is-a-pyranometer/](https://isocalnorthamerica.com/what-is-a-pyranometer/)

该网站旨在帮助我们更好地了解太阳总辐射计。它讨论了太阳辐照度的确切含义及其用途，例如气候学和太阳能发电。它描述了太阳辐照度计测量漫射和直射阳光，以记录到达单位面积表面的太阳能量。通常，该测量单位为 $\frac{W}{m^2}$ 。该网站还简要介绍了两种常见的太阳总辐射计以及校准的重要性。由于他们试图推销校准服务，因此没有直接讨论如何校准。该网站使我们能够以比上述书籍更简单的方式理解太阳总辐射计。

[5] C. Gittins, “Considering the energy consumption of a Raspberry Pi,” *IOT Insider*, Sep. 23, 2024. <https://www.iotinsider.com/news/considering-the-energy-consumption-of-a-raspberry-pi/>

这个网站帮助我确定了我们气象站的整体功耗。它被用于我的第一个数学建模练习，这将在本报告的3.3节中讨论。该网站表示，树莓派在正常负载下通常功耗为5W。

[6] NiuBol, “How much power does a weather station use?,” [Niubol.com, 2024. https://www.niubol.com/Product-knowledge/How-much-power-does-a-weather-station-use.html](https://www.niubol.com/Product-knowledge/How-much-power-does-a-weather-station-use.html).

我也用这个网站完成了我的数学建模计算。他们在这里讨论了气象测量传感器通常的功耗。这个功耗非常低，因为每个传感器预计每小时仅消耗0.3瓦。NiuBol本身是一家销售气象站的公司，但它仍然是一个有用的计算和比较来源。

[7] US, “National Weather Service,” *Weather.gov*, 2025. <https://forecast.weather.gov/MapClick.php?lat=35.19814000000002&lon=-111.65112499999998> (accessed Jun. 18, 2025).

我使用这个气象站作为我的基准气象站。我选择的基准气象站是弗拉格斯塔夫普利亚姆机场的气象站。该气象站将作为我们网站设计和每个传感器精度的基准。我们会将我们的读数与他们的读数进行比较，希望读数相似。该基准已在上文第 3.1 节中讨论过。

[8] J. Portilla, “Python Bootcamps: Learn Python Programming and Code Training,” Udemy, 2019. <https://nau.udemy.com/course/complete-python-bootcamp> (accessed Jun. 22, 2025).

我使用这门 Udemy 课程学习 Python。我没有任何 Python 编程经验，需要从头开始。我一直使用这个资源来学习编程，并且会继续这样做。这个项目需要大量的编程工作，所以这对我来说是一个至关重要的资源。

[9] Raspberry Pi, “Build Your Own Weather Station,” [Raspberrypi.org](https://projects.raspberrypi.org/en/projects/build-your-own-weather-station/0), 2017. <https://projects.raspberrypi.org/en/projects/build-your-own-weather-station/0> (accessed Jun. 27, 2025).

到目前为止，Raspberry Pi 网站在本项目的设计方面提供了很大的帮助。该网站讨论了如何使用 Raspberry Pi 出售的一款特定套件制作一个气象站。我们将以此为大纲，指导我们自己编写项目代码，因为它详细讨论了气象站从头到尾的整个设置过程。我们不会照搬它，因为我们没有用到的套件。

[10] Raspberry Pi, “Raspberry Pi Documentation - Getting Started,” [www.raspberrypi.com](https://www.raspberrypi.com/documentation/computers/getting-started.html). <https://www.raspberrypi.com/documentation/computers/getting-started.html> (accessed Jun. 27, 2025). I used this website to help me understand how to access and use a Raspberry Pi.

当我尝试访问提供的设备时，很遗憾，它上面有密码。幸运的是，这个网站讨论了一些解决方法，包括完全重置或直接更换设备内的SD卡。

[11] Raspberry Pi, “Raspberry Pi OS,” [Raspberry Pi](https://www.raspberrypi.com/software/), 2025. <https://www.raspberrypi.com/software/> (accessed Jun. 27, 2025).

要在更换或擦除 SD 卡后真正激活 Raspberry Pi，我们需要上传 Raspberry Pi 操作系统。本网站讨论了如何上传操作系统，并提供了操作系统的下载和文档。这将用于完全访问 Pi。

3.2.2 Rowan McCullough

[3] World Meteorological Organization, *Guide to meteorological instruments and methods of observation*, Volume 1. Geneva, Switzerland: World Meteorological Organization, 2008.

这份资源已与我们团队的几位成员共享，因为它包含大量对我们项目至关重要的信息。我主要利用这份资源来查找和了解我们将要使用的所有类型传感器的气象测量行业标准。此外，它还概述了每种传感器的不同子类型及其优缺点。这份资源还为我提供了下面第 3.3 节中所示的特定温度计输出计算公式。

[12] “Tempest Weather Station,” Tempest, 2024. <https://shop.tempest.earth/products/tempest> msc1kid=1236cca887f915b3bbd3a4f8024b2f08&utm_source=bing&utm_medium=cp

c&utm_campaign=Bing-Search-B2C-NB-Broad US&utm_term=home%20weather%20station&utm_content=Weather%20Station (accessed Jun. 18, 2025).

此来源是 Tempest 家用气象站的购买页面；这是我们之前提到的基准测试之一。它用于了解气象站的功能和组件。它还包含有助于验证产品性能的客户评论。

[13] A. Overton, “A Guide to the Siting, Exposure and Calibration of Automatic Weather Stations for Synoptic and Climatological Observations,” 2009. Accessed: May 01, 2024. [Online]. Available: <https://www.rmets.org/sites/default/files/2019-02/aw-s-guide.pdf>

该资料来自皇家气象学会一位会员发表的一篇论文，概述了各种气象传感器的校准和长期考量。它既可以作为其他校准研究的交叉参考，也可以作为长期保持准确读数的潜在障碍的指南。它还包含大量关于数据归档的信息，可能对我们日后有所帮助。

[14] Wisconsin DNR, “Calibration & barometric pressure | Wisconsin DNR,” *dnr.wisconsin.gov*. <https://dnr.wisconsin.gov/topic/labCert/BODCalibration2.html>

威斯康星州自然资源部的网站讨论了气压以及基于环境和海拔因素的读数变化。由于大多数气压数据都会根据海拔高度进行校正，因此在将我们的读数与附近站点的读数进行比较以及校准数据库时，了解这一过程非常重要。

[15] R. Coquilla, J. Obermeier, and B. White, “American Wind Energy Association,” 2007. Accessed: May 16, 2023. [Online]. Available: <https://research.engineering.ucdavis.edu/wind/wp-content/uploads/sites/17/2014/03/AWEA-2007-Final-Paper.pdf>

这篇研究论文由加州大学戴维斯分校的研究生发表。它重点介绍了校准风速计时的不确定度因素、可能产生的各种不确定度根源以及如何计算它们。论文还包含风洞中风速计计算的测试数据。这是我们理解风速计校准的重要资源，也是我完成不确定度计算自学作业的指南。

[16] “Login - CAS - Central Authentication Service,” *Udemy.com*, 2025. Available: <https://nau.udemy.com/course/statistics-intro/learn/lecture/35671972#overview>. [Accessed: Jun. 25, 2025]

这门 Udemy 课程是我复习统计学知识以及不确定性如何产生和计算的主要自学资源。我没有用到所有章节，只用到了第三、四、八和十章节。这些课程让我对统计过程有了重要的基础理解，从而帮助我编写了一个 MATLAB 程序来计算风速计读数的不确定性。

[17] EngineerItProgram, “Experimental Uncertainty,” *YouTube*, Mar. 18, 2013. Available: https://www.youtube.com/watch?v=xJBta_HWTRc. [Accessed: Jun. 26, 2025]

这段 YouTube 视频搭配 Udemy 课程，让我更详细地了解了工程师如何在设计和实验中计算和处理不确定性。这对我的自学作业很有帮助，也有助于我们处理数据中不可避免的不确定性。

3.2.3 王疏桐

[18] National Instruments, “NI DAQ Sensor Calibration Guide,” *National Instruments*, 2021. Accessed: May 10, 2024. [Online]. Available: <https://www.ni.com/en-us/innovations/sensor-calibration.html>

本技术指南概述了校准数据采集系统中传感器的程序和最佳实践，尤其适用于使用 NI USB DAQ 或 Raspberry Pi 等微控制器的情况。它帮助我们了解了校准漂移以及推荐的重新校准频率。

[19] Adafruit Industries, “How to Connect Weather Sensors to Raspberry Pi,” *Adafruit Learning System*, 2022. Accessed: Apr. 25, 2024. [Online]. Available: <https://learn.adafruit.com/pi-weather-station/overview>

本在线教程提供了硬件接线图和 Python 库，用于将天气传感器（例如 DHT22、BMP180）连接到 Raspberry Pi 开发板。它指导我们完成了传感器设置和初始数据采集阶段。

[20] M. A. Islam and R. Haque, “IoT-Based Real-Time Weather Monitoring System Using Raspberry Pi,” in *Proc. Int. Conf. on IoT and Applications*, 2020, pp. 44 – 49.

这篇会议论文展示了一个基于 Raspberry Pi 的物联网气象站的完整原型，并讨论了存储和可视化长期气象数据的方法。它帮助我们验证了设计方案，并启发了我们的数据存储方法。

[21] S. Pandey et al., “Analysis of Weather Monitoring Systems Using Cloud Integration,” *Journal of Sensor Networks and Data Communication*, vol. 10, no. 2, pp. 100 – 105, 2021.

本文重点介绍了气象站与云平台（如 AWS 和 ThingSpeak）的集成。它支持我们使用在线仪表盘进行数据可视化和远程监控的决策。

[22] Ambient Weather, “Installation Guide for Ambient Weather WS-5000,” *Ambient Weather*, 2023. Accessed: May 15, 2024. [Online]. Available: <https://ambientweather.com/ws5000manual>

本商业用户手册描述了室外气象站的传感器放置、校准和防护外壳要求。它为我们关于如何屏蔽和安装组件以防风雨和降水的决策提供了参考。

[23] Texas Instruments, “Thermal Design Considerations for Enclosures,” Application Report SPRA953, 2009. Accessed: July 14, 2025. [Online]. Available: <https://www.ti.com/lit/an/spra953/spra953.pdf>

本技术应用说明探讨了外壳材料、表面颜色和太阳辐射暴露如何显著影响室外设备的内部温度。它强调了最大限度地减少热量吸收的重要性，并提出了诸如反射涂层和遮阳板等设计策略。该资源指导了我们为减少被动式太阳加热而做出的气象站外壳材料和表面处理决策。

[24] IPC, “Standard for Determining Current-Carrying Capacity in Printed Board Design (IPC-2152),” IPC, 2009. Accessed: July 14, 2025. [Online]. Available: <http://www.ipc.org/TOC/IPCT-2152.pdf>

这份行业标准文件概述了根据电子系统的功耗和热环境估算其内部温升的方法。它重点介绍了外壳设计和气流如何影响组件的寿命和性能。我们参考该文件评估了气象站的内部加热风险，并支持有关通风以及传感器和处理器板的空间布局的决策。

3.2.4 董晨希

[25] “Printed humidity sensors,” Encyclopedia.pub. <https://encyclopedia.pub/entry/7493>.

这篇文章概述了三种主要的湿度传感器类型：电容式、电阻式和印刷式。它对于区分它们的工作原理和基本应用特别有用。这篇资源帮助我了解了哪些传感器最适合长期环境监测。例如，电容式传感器以其在精度、耐用性和响应时间方面的平衡而闻名，使其成为我们站点设计的首选。

[26] “Review of Printed Humidity Sensors,” MDPI Sensors. <https://www.mdpi.com/2079-4991/13/6/1110>

这篇科学论文对印刷湿度传感器进行了详细的技术比较，包括其制造方法和传感机制。虽然我们并不打算在设计中使用印刷传感器，但本文为我们理解成本效益权衡和短期使用优势提供了一个重要的基准。这对于概念评估以及电阻式和电容式传感器的比较尤其有用。

[27] “Capacitive vs Resistive Humidity Sensors,” Encyclopedia.pub. - <https://encyclopedia.pub/entry/7493>

本节重点比较电容式传感器和电阻式传感器。它详细解释了电容式传感器通过检测介电常数的变化来工作，而电阻式传感器则通过跟踪吸湿性材料电阻的变化来工作。我在计算相对湿度时使用了此资料，该公式基于电容，假设传感器遵循线性介电行为。

[28] BOMS Review - Comparison of sensing layers - <https://www.ias.ac.in/article/fulltext/boms/045/0238>

本资源探讨了湿度传感器的构造材料及其对响应时间和精度的影响。这在考虑校准策略时非常有用，因为这些材料随时间推移的漂移会影响长期性能。它也验证了我们在最终部署计划中考虑传感器更换周期的决定。

[29] “Humidity and Dew Point,” National Weather Service. https://www.weather.gov/media/epz/wxcalc/dewpoint_rh.htm

这款由政府托管的计算器工具用于根据温度和相对湿度数据建模相对湿度和露点。我参考它来验证我的传感器输出计算。它还能深入了解传感器如何将环境数据解读成有意义的天气测量数据。

[30] Ma, H. et al., “Graphene-Based Humidity Sensors,” arXiv. <https://arxiv.org/abs/2410.02255>

本文介绍了一种利用石墨烯制成的新型印刷湿度传感器。虽然目前仍处于研究阶段，但它展现了超灵敏柔性传感器的未来发展方向，有望增强智能可穿戴设备或高分辨率大气监测。我借此将我们正在评估的现有传感器方案与未来空间站迭代中可能实现的方案进行对比。

[31] NOAA Sensor Siting Standards - <https://www.weather.gov/coop/standards>

官方指南概述了湿度传感器的正确安装协议，包括高度（距地面 1.25-2 米）、避免阳光直射以及相对于植被的位置。

[32] WMO Guide to Meteorological Instruments - https://library.wmo.int/index.php?lvl=notice_display&id=12407

在这里，我了解了湿度传感器的长期稳定性、校准要求以及可接受的误差范围。我特意参考了它的建议，以确保我的自动气象系统中相对湿度数据的准确性。

3.3 数学建模

3.3.1 电源要求 - Ian Torp

估算气象站的功耗至关重要。为此，我们研究了树莓派和气象传感器在正常负载下的典型功耗。树莓派在正常负载下通常功耗为 5W [5]，而每个传感器每小时通常仅功耗 0.3W [6]。要计算出每天的功耗，我们需要计算出每天的用电量。每天有 24 小时，所以我们将计算结果乘以每个传感器的 0.3W 和树莓派的 5W。由于我们总共可能有 7 个传感器，因此公式变为：

$$24\text{hrs} \times 7(.3\text{W}) + 24\text{hrs} \times 5\text{W} \quad (1)$$

这意味着气象站每天消耗 0.1704 千瓦时电能，相当于每年 62.192 千瓦时。这非常低的功耗。一块简单的 60 瓦小型太阳能电池板，即使每天只有 5 小时的日照，也足以支持该气象站的日常运行。一块 60 瓦的太阳能电池板，即使每天只有 5 小时的日照，也能产生约 300 瓦时的电能，足以支持气象站的日常运行。

3.3.2 温度计输出计算 - Rowan McCullough

我们的站点预计将使用电子温度计（PRT），它通过纯金属的电阻来测量温度。纯金属的电阻与整个气象范围内的温度几乎呈线性关系 [3]。不同的电子温度计根据其使用的金属进行分类，最常见的是 Pt100。Pt100 传感器还有子类别，这些子类别按标准公差分类，例如 IEC 60751 或 ASTM E1137 [3] [13]。这些标准定义了用于将输出转换为温度值的电阻和灵敏度曲线。PRT 的电阻可以用下面的 Callender-Van Dusen 方程表示。

$$R = R_0 \left(1 + At + Bt^2 + C(t - 100)t^3 \right) \quad (2)$$

在该方程中， R_0 表示 0°C 时的电阻， t 表示温度（摄氏度）。系数 A 、 B 和 C 取决于标准规范。假设符合 IEC 60751 规范，当温度低于零度时， $R_0=100\Omega$ 、 $A=3.908 \times 10^{-3}^\circ\text{C}$ 、 $B=-5.80 \times 10^{-7}^\circ\text{C}$ 和 $C=4.27 \times 10^{-12}^\circ\text{C}$ ；当温度高于零度时， $C=0^\circ\text{C}$ 。当温度高于 -40°C 时， C 项可以忽略不计，因此温度反比方程可以简化为：

$$t = -\frac{A}{2B} + \frac{A}{2B} \sqrt{1 - \frac{4B}{A^2} \left(1 - \frac{R}{R_0}\right)} \quad (3)$$

因此，假设输出读数为 110 Ω，我们的温度读数约为 25.34 ° C。

3.3.3 热管理考虑因素 - 王疏桐

户外电子设备容易受到阳光直射、温度波动、空气流通不足等因素的影响，导致内部温度升高，过热会影响传感器精度，加速元器件老化。

焦耳加热

可以使用以下热阻模型估算总功率损耗引起的内部温升：

$$\Delta T = P * R_{\theta JA} \quad (4)$$

- ΔT：器件温升（° C）
- P：总功耗（W）
- R_{θ JA}：器件或系统结至环境热阻（° C/W）

结果：

如果传感器总功耗为 7.1W，热阻为 10° C/W，则温升约为 71° C。如果环境温度为 35° C，则外壳内部温度可能高达 106° C。

太阳辐射加热

在户外应用中，太阳辐射会显著提高暴露外壳的温度。吸收功率可按下式计算：

$$Q_{solar} = G \cdot A \cdot \alpha \quad (5)$$

- G：太阳辐射强度（典型值：800 - 1000 W/m²）
- A：受光面积（m²）
- α：材料吸收系数（黑色：接近0.9，白色：接近0.3）

结果：

0.1m² 的黑匣子接收：

$$Q = 1000 * 0.1 * 0.9 = 90W$$

需要反光涂层或遮阳设计

3.3.4 湿度传感器输出计算 - 董晨希

对于电容式湿度传感器，相对湿度（%RH）通常由电容的变化决定：

$$RH = \frac{C - C_{\text{dry}}}{C_{\text{wet}} - C_{\text{dry}}} \times 100 \quad (6)$$

Where:

C 为测量电容

C_{dry} = 0% RH 时的电容

C_{wet} = 100% RH 时的电容

传感器示例:

C_{dry} = 180 pF, C_{wet} = 250 pF, Measured C = 216pF

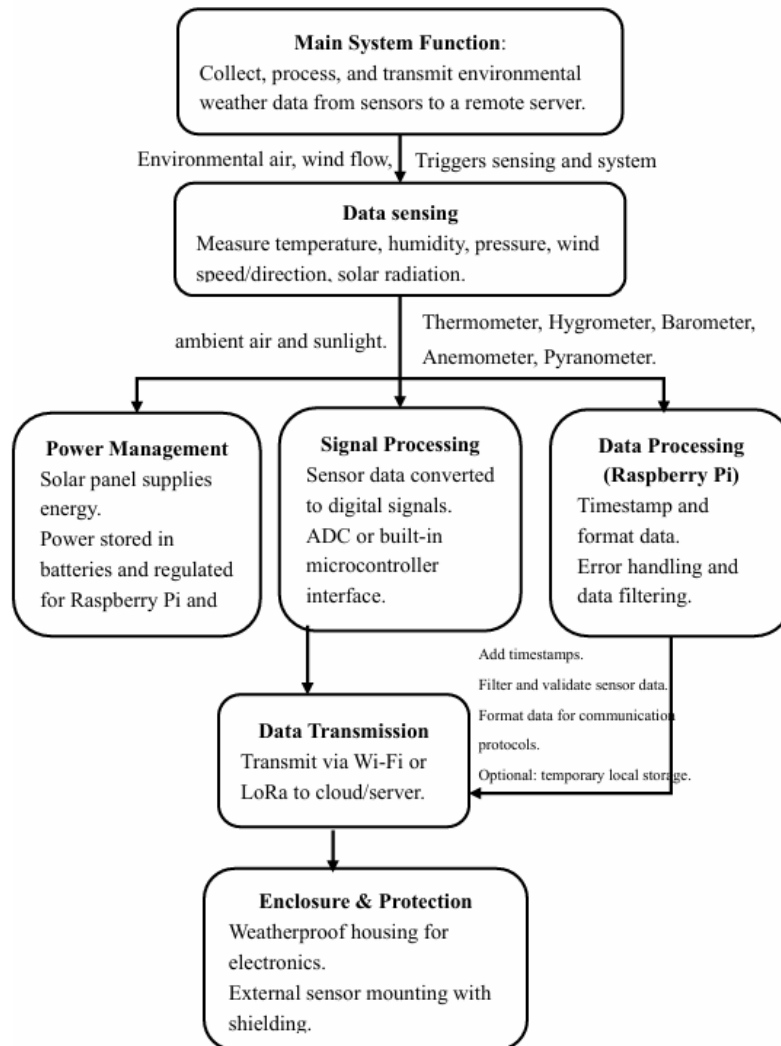
$$RH = \frac{216 - 180}{250 - 180} \times 100 = \frac{36}{70} \times 100 = 51.4\% \quad (7)$$

通过将数值与制造商的校准曲线进行比较，我们完成了验证。该模型使我们能够验证传感器灵敏度，以满足±2 - 3% RH 的传感器公差，并指导我们进行自动校准，以补偿随时间推移产生的漂移。

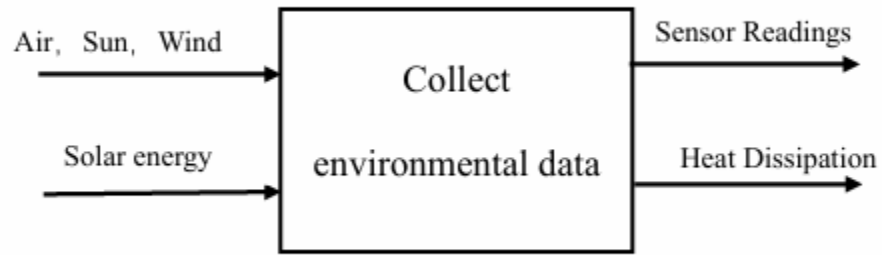
4 设计理念

4.1 功能分解

功能分解图对于我们的气象站项目至关重要。它将复杂的系统分解为传感、信号处理、数据管理、数据传输、电源管理和结构保护等关键子功能，帮助我们明确如何实现每一项工程需求。这种结构化方法使我们能够更高效地分配任务，提前识别技术挑战（例如模数转换、太阳能调节等），并确保子系统之间的有机集成。此外，功能图还清晰地展示了从环境输入到数据输出的逻辑路径，使设计更易于验证和测试。分解模型还具有良好的模块化，方便后续的调试、升级或组件更换。



黑箱模型

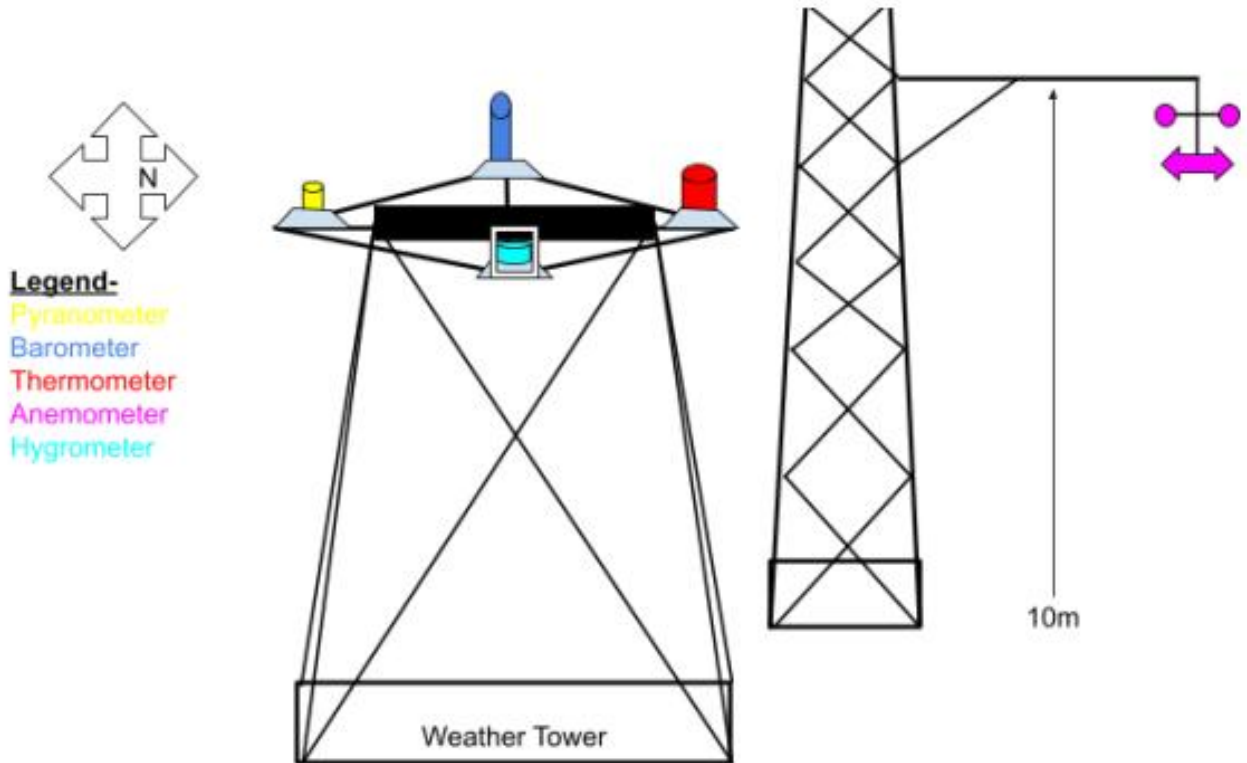


4.2 概念生成

在我们的项目中，适用于概念生成的两个主要系统是传感器的结构布局 and 用于处理数据的编码策略。每个团队成员都为这两个系统设计了一个概念。这些概念如下所示。被确定为中间概念的概念被选为基准，以便与其他概念进行比较，并在第 4.4 节中所示的 Pugh 图上进行比较。

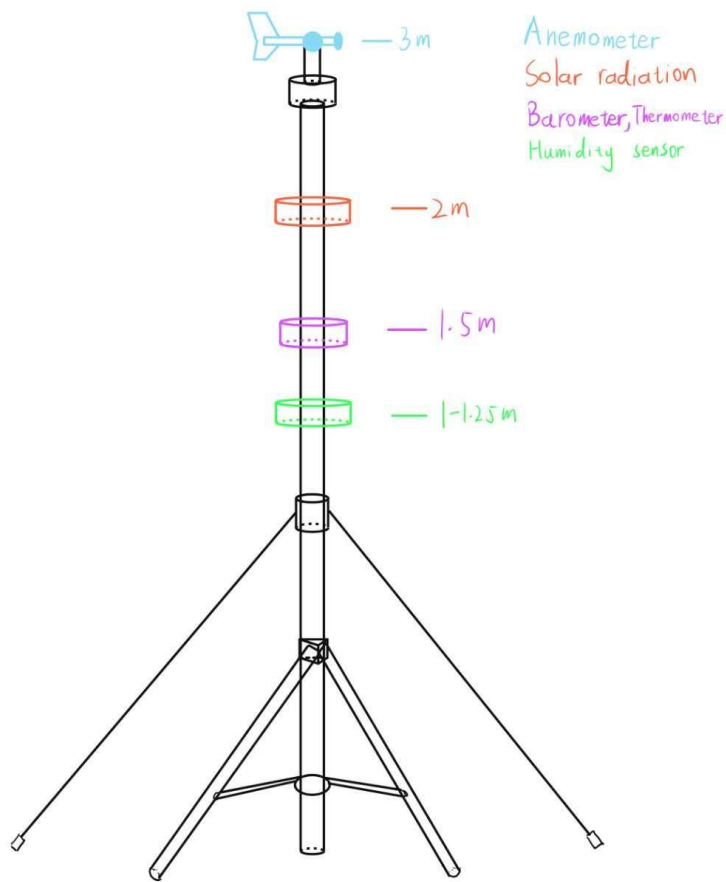
4.2.1 结构概念:

基准 (Rowan):



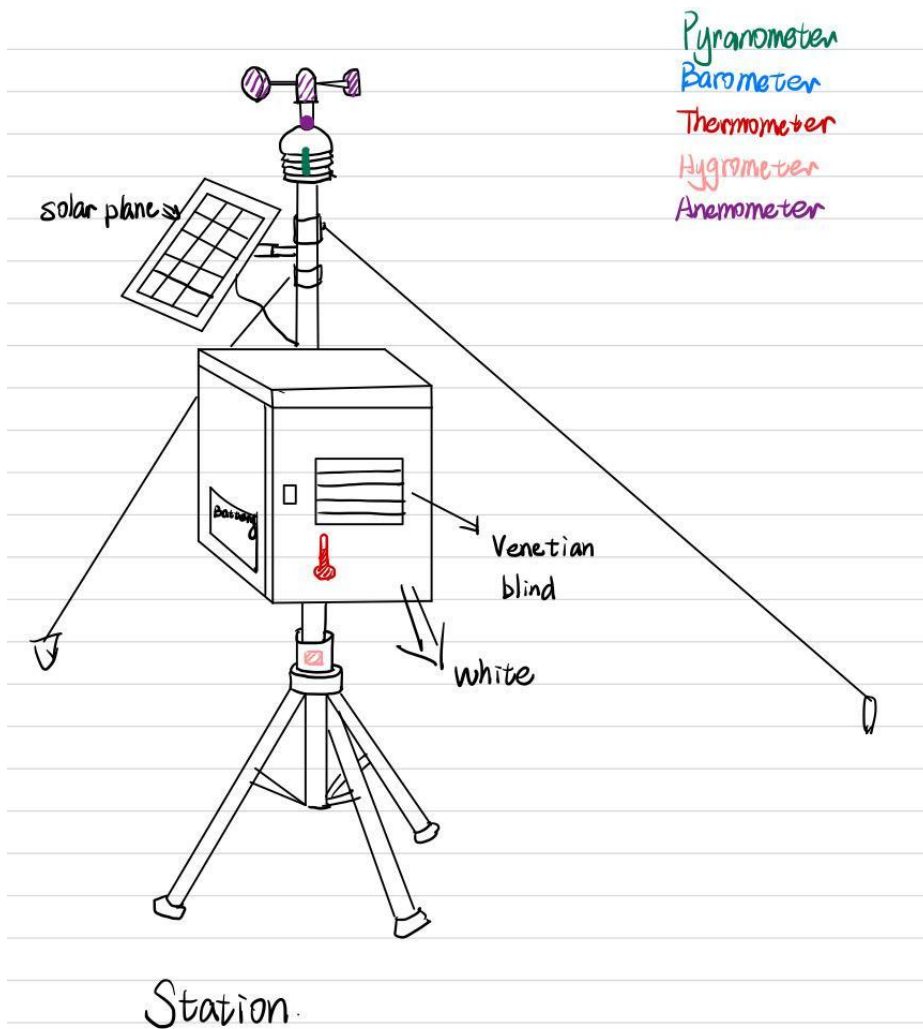
该结构概念利用了可再生能源实验室外部现有的小型平台以及现有的风速计塔。该平台将采用菱形横梁，并在平台顶部安装对角支撑。预计安装在此高度的传感器（包括除风速计以外的所有传感器）将安装在其顶点，以提供三个接触点来保持稳定性。风速计将安装在现有塔上十米高的行业标准高度的吊杆上。任何需要额外安装的传感器都可以安装在合适的材料内，同时与其他传感器保持足够远的距离以避免干扰。该概念的优点包括符合行业标准、成本低廉和经久耐用。缺点包括建造和安装横梁和吊杆可能存在困难，以及横梁的角部安装在靠近头部高度的位置存在潜在的安全隐患。

设计 1 (晨希):



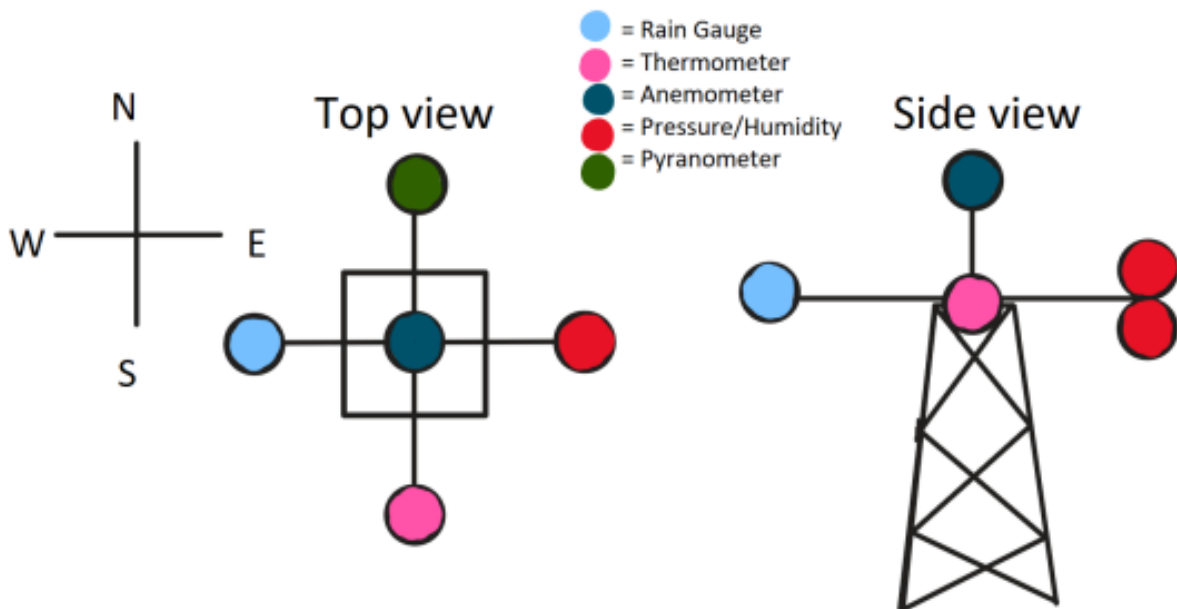
结构方案采用模块化塔式结构，高度约3米，并通过三脚架支撑和线缆加固。传感器分层安装，方便独立布置，减少相互干扰。风速计安装在顶部（3米），确保气流畅通。太阳辐射计安装在南侧2米高处，确保充足的光照。气压计和温度计安装在对面1.5米高处。湿度传感器安装在最底层（1-1.25米），置于具有通风和辐射屏蔽的外壳内，避免阳光直射和热干扰。所有传感器的数据线均从主杆内部走线，既防风防雨，又美观整洁。模块化结构设计方便更换、校准和运输。其主要缺点是风速计高度未达到10米，不符合风测量的行业标准，可能会影响数据的准确性。

设计 2（疏桐）：



该结构概念描绘了一个紧凑型集成气象站，它构建在一个稳定的三脚架平台上，并配有垂直安装杆，以最大限度地增加传感器的垂直放置空间。该设计将所有必要的传感器（包括风速计、风向标、日射强度计和太阳能电池板）整合到一个中央桅杆上。温度、湿度和气压传感器安装在防辐射的史蒂文森屏内，以减少太阳加热干扰。箱体也漆成白色，并配有百叶窗以防止过热。这种设置的优点包括占地面积小、易于部署以及集中式电源/数据路由。然而，主要的缺点是风速计的高度不符合10米的风力测量行业标准，这可能会影响数据的准确性。

设计 3 (Ian):



此概念与 基准 类似，因为它采用了安装在实验室外部现有平台上的安装系统。主要区别在于传感器的安装系统，此概念使用横杆，每个横杆的末端都安装一个传感器。由于压力和湿度传感器的暴露要求相似，因此它们将彼此叠放安装。主要区别在于风速计的放置位置，此概念将其放置在平台正上方的垂直杆上。此概念的优点与 基准 类似，并且安装更方便。缺点包括缺乏稳定性和耐用性，每个传感器支架只有一个接触点，并且风速计的高度未达到行业标准的 10 米。

4.2.1 伪代码概念

基准 (Ian):

我编写的伪代码只是一个简单的概述，描述了树莓派中每个传感器的处理流程。这些流程可以在附录A中查看。这段基于Python的代码读取传感器数据，并相应地完成每个传感器的校准计算。我的想法是将这些数据点存储在树莓派中每个传感器的相应部分中。我建议每5分钟将这些数据上传到网站一次，以提供清晰且最新的信息。我还建议树莓派应该连接外部存储设备，这样我们就可以在本地备份所有收集到的数据，保存大约一年。

这个设计很简单，对于外部程序员来说也很容易理解。它还能确保在网站出现错误时，我们有备份数据。遗憾的是，这种设计也更昂贵，因为它需要购买外部硬盘。这种设计需要大量的存储空间，因为5分钟的上传速度比弗拉格斯塔夫普利亚姆机场气象站的数据上传速度要快得多。这也将导致比其他设计更高的功耗。

设计 1 (晨希):

我提出了一个基于 Python 的脚本：每 10 分钟读取一次传感器数据，并将数据同步上传到本地数据库和服务器。每个传感器的数据独立上传，而不是一起上传，这样当一个传感器发生故障时，不会影响其他传感器的数据。同时，一天过去后，全天的数据会在本地数据库中额外保留一天，以应对数据丢失。另一种方案是根据环境变化动态调整数据记录频率：当几个数据几乎相同或差异不大时，降低记录频率；当环境变化较大时，则加快记录频率。

设计 2 (疏桐):

此设计采用结构化方法，使用基于 Python 的工具处理和可视化环境天气数据。数据每 30 秒从本地 CSV 文件或远程云源收集一次。每个数据输入都包含时间戳、温度、湿度和气压读数。这些数据会立即使用库进行清理，以处理缺失或无效的条目，并确保日期时间格式的一致性。(pandas)

每 30 秒，系统会将一条新记录添加到一个短期缓冲区，该缓冲区最多可存储 20 个校准读数。每分钟，系统都会根据这些读数计算一个滚动平均值，以反映当前状况。此实时值使用或等库渲染为实时图表，并可选择上传到使用 Dash 或 Streamlit 等工具构建的简易仪表板。(matplotlib/seaborn)

除了实时更新外，该系统还支持对长期摘要进行分层平均。每隔10分钟，系统会根据过去10个1分钟值计算平均值，并将其存储在缓冲区中。收集到6个10分钟的数据块后，再次计算平均值，生成最终的每小时摘要，该摘要可以存储用于历史分析或以PDF报告形式下载。

可视化包括：

- 折线图显示温度随时间的变化。
- 用于比较不同间隔的气压值的条形图。
- 显示温度和湿度的多线图。
- 每个图表都经过标记，颜色样式清晰，并自动更新以反映最新读数。

该系统设计支持定时更新和可选的仪表板界面。用户可以选择传感器类型或时间窗口来查看特定的数据集，并且可以将整个过程配置为使用 CRON 作业或 Python 中的基于时间的触发器自动重复。

设计 3 (Rowan):

本设计将利用简单的每小时平均值（由六个十分钟平均值计算得出）进行数据库上传。传感器每 30 秒读取一次，最多保存 20 个读数，然后取平均值并作为实时读数发送到网站。每次获取新读数时，第 20 个读数将被删除并替换；每分钟一个新的实时读数将替换网站上显示的读数。这将为任何想要在访问我们网站时查看当前天气值的用户提供实时、准确的读数。每 20 个读数后，将获取一个单独的平均值并保存到缓冲区，该缓冲区将收集这些平均值，直到一小时过去。然后，每小时的平均值将上传到数据库进行存储。这可以避免 Raspberry Pi 上一次性存储过多数据，同时仍能为网站提供实时和长期数据。此过程的基本概述如下，实际代码结构请参见附录 A。

- 每 30 秒：

将校准的原始数据添加到 20 个读数的缓冲区。

- 每 1 分钟：

计算最近 20 个读数的滚动平均值。

将其作为当前“实时”值上传到网站。

将其存储在 10 分钟缓冲区中。

- 每 10 分钟：

当收集到 10 个 1 分钟值时，简单取平均值（等权重）。

将其添加到每小时缓冲区。

- 每 60 分钟：

当 6 个十分钟区块填满后，取平均值，形成每小时摘要。

发送到数据库。

4.3 选择标准

4.3.1 结构标准

结构设计的选择标准直接源于我们的工程需求以及客户需求（如果这些需求适用于该系统）。虽然我们无法在没有实体部件的情况下对大多数设计进行计算，但我们进行了小组讨论，以最好地确定每种设计在多大程度上实现了这些要求，以及它们在我们的决策矩阵中应占多大的权重。这些标准包括耐候性、安全合规性、低维护成本、易于安装、可读取多个风速以及符合行业标准。

4.3.2 伪代码标准

与结构标准类似，这些标准是从我们的工程需求和客户需求中提取的，前提是它们与该系统相关。虽然我们的实际代码对于许多标准都无法量化，但我们了解它们完成每个目标的程度，并且能够确定它们对每个标准的完成情况，并在决策矩阵中权衡每个标准。

4.4 概念选择

4.4.1 结构概念选择

每个设计都被放入 Pugh 图表中，排名前两个的设计进入决策矩阵。

Pugh Chart

Criteria	DATUM (Rowan)	Design 1 (Chenxi)	Design 2 (Shutong)	Design 3 (Ian)
Weather Durability	0	-1	1	-1
Safety Compliance	0	1	1	0
Low Maintenance	0	0	-1	0
Easy Installation	0	0	-1	0
Multiple Windspeed Readings	0	0	0	0
Measured at Industry Standards	0	-1	-1	-1
	Totals	-1	-1	-2
	Rank	2	2	4

概念选择的第一步是使用Pugh图表。首先选择一个基准，然后根据每个设计在满足每个标准方面比该基准更好或更差进行排名。1代表更好，-1代表更差，0代表相同。然后，将这些数字相加，得出设计的排名。由于设计一和设计二并列第二，因此经过团队讨论后，我们决定将设计一移至决策矩阵。

Criteria	Weights	Design DATUM (Rowan)		Design 1 (Chenxi)	
		Unweighted Score	Weighted Score	Unweighted Score	Weighted Score
Weather Durability	0.2	75	15	75	15
Safety Compliance	0.1	90	9	100	10
Low Maintenance	0.2	90	18	70	14
Easy Installation	0.15	50	7.5	30	4.5
Multiple Windspeed Readings	0.1	100	10	100	10
Measured At Industry Standards	0.25	80	20	60	15
Sum	1	485	79.5	435	68.5

在这个决策矩阵中，我们为每个标准分配了权重，总分为 1。然后，我们根据每个设计在 100 分中的标准完成情况为每个设计打分。将这些分数乘以每个标准的权重，得出每个设计 100 分中的分数。获胜设计是基准设计，得分为 79.5。

4.4.2 伪代码概念选择

使用与结构系统相同的过程，将每个伪代码设计放入 Pugh 图表中，以确定两个最佳设计，然后将其用于决策矩阵。

Pugh Chart

Criteria	DATUM (Ian)	Design 1 (Chenxi)	Design 2 (Shutong)	Design 3 (Rowan)
Long Term Data Storage	0	0	-1	1
Increased Data Accuracy	0	1	0	0
Remote Data Access	0	0	1	1
User Friendly	0	-1	0	0
Low Power Requirement	0	1	-1	0
Data Transmission	0	0	1	0
	Totals	1	0	2
	Rank	2	3	1

在将每个伪代码设计与基准进行比较后，设计 1 和设计 3 排名最高，并进入决策矩阵。

Criteria	Weights	Design 1 (Chenxi)		Design 3 (Rowan)	
		Unweighted Score	Weighted Score	Unweighted Score	Weighted Score
Long Term Data Storage	0.25	75	18.75	85	21.25
Increased Data Accuracy	0.25	90	22.5	70	17.5
Remote Data Access	0.25	80	20	90	22.5
User Friendly	0.1	60	6	75	7.5
Low Power Requirement	0.1	80	8	80	8
Data Transmission	0.05	90	4.5	90	4.5
Sum	1	475	79.75	490	81.25

在为每个标准分配权重之后，我们会根据每个设计对该标准的完成情况，以 0 到 100 的等级对每个设计进行评分。然后将这些分数乘以每个标准的权重，得出每个设计的总分（满分 100 分）。获胜的设计是设计 3，得分为 81.25。

5 结论

本文档详细介绍了北亚利桑那大学可再生能源实验室气象站的设计和开发进展。该气象站被要求提供气温、湿度、气压、太阳辐照度、风速和降水等气象数据测量。降水测量的要求并非一成不变，因为客户希望我们暂时考虑这一可能性。该气象站预计将收集准确的科学数据，并允许在任何有互联网连接的地方访问其数据。

Carson Pete 教授特别要求数据通过互联网连接传输和查看。他还要求该站使用可再生能源供电，由于该站位于可再生能源实验室，因此这个要求很简单。此外，它还应足够耐用，能够承受所有户外天气条件。每个传感器都应精确且维护成本低，以便该站即使在无人操作的情况下也

能收集准确的数据。上传数据的网站也应方便用户使用，并能够至少存储一年的数据。由于该项目的成本相对较低，我们预计不会超出预算。

为了满足这些要求，我们进行了设计选型。最终的设计方案主要包含三个方面。结构方面，设计涉及可再生能源实验室提供的金属桁架以及其中一座塔楼。六英尺高的桁架顶部将固定一个菱形横杆，使其顶点安装的每个传感器都具有三个接触点。目前的设计不包含降水测量装置。温度计、气压传感器、日射强度计和湿度传感器将安装在该横杆顶部。这成功地符合了它们各自的测量标准。为了满足风速计的标准，风速计将安装在距离现有塔楼10米的吊臂上。该设计决策的下一个方面是伪代码。伪代码使用简单的每小时平均值，该平均值由六个十分钟平均值计算得出，用于数据库上传。传感器将每30秒读取一次，最多保存20个读数，然后取平均值并作为实时读数发送到网站。每次获取新的读数时，最早的读数都会从 Raspberry Pi 中删除。每分钟都会将新的实时读数上传到网站，替换之前的读数。这提供了实时信息，只要有互联网连接，就可以在任何地点访问。每小时的平均值会被上传到数据库，并无限期保存。这可以防止 Raspberry Pi 上存储过多数据。

6 参考

- [1] Setra Systems, Inc, “Barometric Pressure Sensors | Setra Systems,” *Setra.com*, 2025. <https://www.setra.com/product/pressure/barometric> (accessed Jun. 18, 2025).
- [2] N. US Department of Commerce, “Standards and Policy,” *www.weather.gov*. <https://www.weather.gov/coop/standards>
- [3] World Meteorological Organization, *Guide to meteorological instruments and methods of observation., Volume 1*. Geneva, Switzerland: World Meteorological Organization, 2008.
- [4] ISO-CAL, “What is a Pyranometer? 10 Important Points to Consider.,” *isocalnorthamerica.com*, Jan. 16, 2023. <https://isocalnorthamerica.com/what-is-a-pyranometer/>
- [5] C. Gittins, “Considering the energy consumption of a Raspberry Pi,” *IOT Insider*, Sep. 23, 2024. <https://www.iotinsider.com/news/considering-the-energy-consumption-of-a-raspberry-pi/>
- [6] NiuBo1, “How much power does a weather station use? ,” *Niubol.com*, 2024. <https://www.niubol.com/Product-knowledge/How-much-power-does-a-weather-station-use.html>
- [7] US, “National Weather Service,” *Weather.gov*, 2025. <https://forecast.weather.gov/MapClick.php?lat=35.19814000000002&lon=-111.65112499999998> (accessed Jun. 18, 2025).
- [8] J. Portilla, “Python Bootcamps: Learn Python Programming and Code Training,” *Udemy*, 2019. <https://nau.udemy.com/course/complete-python-bootcamp> (accessed Jun. 22, 2025).
- [9] Raspberry Pi, “Build Your Own Weather Station,” *Raspberrypi.org*, 2017. <https://projects.raspberrypi.org/en/projects/build-your-own-weather-station/0> (accessed Jun. 27, 2025).

- [10] Raspberry Pi, “Raspberry Pi Documentation – Getting Started,” *www.raspberrypi.com*. <https://www.raspberrypi.com/documentation/computers/getting-started.html> (accessed Jun. 27, 2025).
- [11] Raspberry Pi, “Raspberry Pi OS,” *Raspberry Pi*, 2025. <https://www.raspberrypi.com/software/> (accessed Jun. 27, 2025).
- [12] “Tempest Weather Station,” *Tempest*, 2024. https://shop.tempest.earth/products/tempest?msclkid=1236cca887f915b3bbd3a4f8024b2f08&utm_source=bing&utm_medium=cpc&utm_campaign=Bing-Search-B2C-NB-Broad-US&utm_term=home%20weather%20station&utm_content=Weather%20Station (accessed Jun. 18, 2025).
- [13] A. Overton, “A Guide to the Siting, Exposure and Calibration of Automatic Weather Stations for Synoptic and Climatological Observations,” 2009. Accessed: May 01, 2024. [Online]. Available: <https://www.rmets.org/sites/default/files/2019-02/aws-guide.pdf>
- [14] Wisconsin DNR, “Calibration & barometric pressure | Wisconsin DNR,” *dnr.wisconsin.gov*. <https://dnr.wisconsin.gov/topic/labCert/BODCalibration2.html>
- [15] R. Coquilla, J. Obermeier, and B. White, “American Wind Energy Association,” 2007. Accessed: May 16, 2023. [Online]. Available: <https://research.engineering.ucdavis.edu/wind/wp-content/uploads/sites/17/2014/03/AWEA-2007-Final-Paper.pdf>
- [16] “Login – CAS – Central Authentication Service,” *Udemy.com*, 2025. Available: <https://nau.udemy.com/course/statistics-intro/learn/lecture/35671972#overview>. [Accessed: Jun. 25, 2025]
- [17] EngineerItProgram, “Experimental Uncertainty,” *YouTube*, Mar. 18, 2013. Available: https://www.youtube.com/watch?v=xJBta_HWTRc. [Accessed: Jun. 26, 2025]
- [18] National Instruments, “NI DAQ Sensor Calibration Guide,” *National Instruments*, 2021. Accessed: May 10, 2024. [Online]. Available: <https://www.ni.com/en-us/innovations/sensor-calibration.html>
- [19] Adafruit Industries, “How to Connect Weather Sensors to Raspberry Pi,” *Adafruit Learning System*, 2022. Accessed: Apr. 25, 2024. [Online]. Available: <https://learn.adafruit.com/pi-weather-station/overview>
- [20] M. A. Islam and R. Haque, “IoT-Based Real-Time Weather Monitoring System Using Raspberry Pi,” in *Proc. Int. Conf. on IoT and Applications*, 2020, pp. 44–49.
- [21] S. Pandey et al., “Analysis of Weather Monitoring Systems Using Cloud Integration,” *Journal of Sensor Networks and Data Communication*, vol. 10, no. 2, pp. 100–105, 2021.
- [22] Ambient Weather, “Installation Guide for Ambient Weather WS-5000,” *Ambient Weather*, 2023. Accessed: May 15, 2024. [Online]. Available: <https://ambientweather.com/>

[ws5000manual](#)

[23] Texas Instruments, “*Thermal Design Considerations for Enclosures*, ” Application Report SPRA953, 2009. Accessed: July 14, 2025. [Online]. Available: <https://www.ti.com/lit/an/spra953/spra953.pdf>

[24] IPC, “*Standard for Determining Current-Carrying Capacity in Printed Board Design (IPC-2152)*, ” IPC, 2009. Accessed: July 14, 2025. [Online]. Available: <https://www.ipc.org/TOC/IPCT-2152.pdf>

[25] “Printed humidity sensors,” Encyclopedia.pub. <https://encyclopedia.pub/entry/7493>.

[26] “Review of Printed Humidity Sensors,” MDPI Sensors. <https://www.mdpi.com/2079-4991/13/6/1110>

[27] “Capacitive vs Resistive Humidity Sensors,” Encyclopedia.pub. - <https://encyclopedia.pub/entry/7493>

[28] BOMS Review - Comparison of sensing layers - <https://www.ias.ac.in/article/fulltext/boms/045/0238>

[29] “Humidity and Dew Point,” National Weather Service. https://www.weather.gov/media/epz/wxcalc/dewpoint_rh.htm

[30] Ma, H. et al., “Graphene-Based Humidity Sensors,” arXiv. <https://arxiv.org/abs/2410.02255>

[31] NOAA Sensor Siting Standards - <https://www.weather.gov/coop/standards>

[32] WMO Guide to Meteorological Instruments - https://library.wmo.int/index.php?lvl=notice_display&id=12407

7 附录

7.1 附录 A: 伪代码

设计 1(晨希)-

主要思路:

每个传感器独立上传。

本地备份 24 小时。

基于环境稳定性的自适应采样率。

初始化:

加载所有传感器驱动程序。

连接 SQLite 数据库和服务端。

定义环境稳定性阈值。

每 10 分钟：

传感器：

```
data = read(sensor)

timestamp = get_time()

if sensor == "humidity":
    if abs(data - last_value) > 10 or data > 100:
        flag = "outlier"
    else:
        flag = "valid"
    save_local(sensor, data, timestamp, flag)
else:
    save_local(sensor, data, timestamp)

upload(sensor, data, timestamp)
```

每10分钟采集一次数据。

湿度数据增加异常值判断。

所有数据首先保存到本地数据库，然后上传到远程服务器。

日常维护：

```
delete_local_data(older_than=24_hours)
```

```
generate_daily_report()
```

每天清理超过 24 小时的本地缓存，并生成当天的传感器运行报告。

自适应采样率逻辑：

```
if variance(sensor_data) < threshold:
    set_interval(longer) # Environment is stable → reduce sampling frequency
else:
```

```
set_interval(shorter) # Drastic changes in the environment → Increase sampling frequency
```

通过判断数据变化程度来确定采样间隔。

环境稳定性 → 节能

环境波动 → 提高分辨率

设计 2(疏桐)-

脚本: weather_visualization.py

从 CSV 文件加载数据

```
file_path = "weather_data.csv" data = read_csv(file_path)
```

清理数据

将“时间戳”转换为日期时间格式,删除缺失值的行

过滤温度: 将值保持在 -50 至 60° C 之间

过滤气压: 将值保持在 800 至 1100 hPa 之间

过滤最近 24 小时的数据

```
now = current_datetime() start_time = now - 24 hours recent_data = data where timestamp >= start_time
```

绘制温度图

创建图形 (10x4) 绘制时间戳与温度的关系图, 标签: “温度 (° C)”, 标题: “过去 24 小时的温度”, 将 x 轴旋转 45 度, 添加网格和图例, 另存为 “temperature_last24h.png”

绘制压力图

创建图形 (10x4) 绘制时间戳与压力的关系图, 标签: “压力 (hPa)”, 标题: “过去 24 小时的压力”, 将 x 轴旋转 45 度, 添加网格和图例, 另存为 “pressure_last24h.png”

输出确认消息

打印: “图表已保存: temperature_last24h.png 和 pressure_last24h.png”

设计 3 (Rowan)-

Buffers

```
rolling_buffer = [] # Up to 20 raw readings (~10 min at 30s intervals)
```

```

ten_minute_buffer = []      # 1 average per minute, keep 10
hourly_buffer = []         # 1 average per 10 minutes, keep 6
LOOP every 30 seconds:

    data = read_and_calibrate_all_sensors()

    # Maintain rolling buffer

    IF length of rolling_buffer == 20:

        Remove oldest

    Append data to rolling_buffer

    # Every 1 minute: upload rolling average for real-time display

    IF time_since_last_minute == 60 sec:

        rolling_avg = average_buffer(rolling_buffer)

        upload_to_website(rolling_avg)

        Append rolling_avg to ten_minute_buffer

    # Maintain 10-minute buffer

    IF length of ten_minute_buffer == 10:

        ten_min_avg = average_buffer(ten_minute_buffer)

        Append ten_min_avg to hourly_buffer

        Clear ten_minute_buffer

    # Every 60 minutes (6x10min)

    IF length of hourly_buffer == 6:

        hourly_avg = average_buffer(hourly_buffer)

        store_hourly_avg_to_database(hourly_avg)

        Clear hourly_buffer

```

基准 (Ian):

IMPORTANT NOTE CODE IS HIGHLY DEPENDANT ON WHAT SENSOR IS BEING USED

Anemometer: will produce signals as it rotates. These signals will likely occur twice per rotation. With this, wind speed will be calculated as:

$Circumference = 2 * \pi * radius$

```
Rotations = signals/2
```

```
Speed = (rotations * circumference) / time
```

```
#this will then convert to kmph or mph through simple conversion math.
```

For rainfall: the signal will be produced when one bucket fills enough to tip over. Knowing the tipping point of the bucket is essential for calculating the overall rainfall.

```
bucket = .227mm #this is example size not real
```

```
count = 0
```

```
def bucket_tipped():
```

```
    global count
```

```
    Count = count+1
```

```
    Print (count*bucket)
```

```
Rainfall= bucket_tipped/time
```

```
#time should measure from start of rain till end
```

```
#Reset function here
```

```
Temperature:
```

The thermometer can output many things depending on what it is. Assuming it outputs current it will be in mA

```
Reading = data input #in mA
```

```
Temp = # reading altered via reference table or specific formula.
```

#if using reference table, indexing and slicing very important. Additionally, the table will be uploaded to the Raspberry Pi.

```
Humidity
```

Similar to temperature, it will produce voltage or amps.

The code will take those outputs and convert them either using a reference table or a specific formula, likely provided in the instructions for the device.

```
Humidity = (input) + #modification based on input. Either formula or reference table
```

```
Barometric Pressure
```

Again, similar to temperature and entirely dependent on what sensor we have.

```
Pressure = (input) #modified with formula or table
```

Solar irradiance

#Will likely need a voltage reader, then convert that voltage to W/m^2 through calibration/ specific formula

Irradiance= (V #modified by formula) in (W/m^2)

#All of this data will be collected internally into separate tables specific to each sensor. Then, every 5 min (or sooner for certain devices) it will upload this to the website where it will then be sorted properly and displayed in a user-friendly manner.