Development Specification

Show the statistics of the weather monitoring station

Version 1.0
Last modified on 2023-Sep-15
Number of pages 7

Document History		
Date	**Description**	**Author(s)**
04-Jun-09	Version 0.01: Created the initial version of the doc.	Emma
15-Jan-10	Version 0.02: Initial version is moved into the new text document template	Ruben
31-Jan-13	Version 1.0: Updated according to text document template version 1.4	Arman
2013-Nov-28	Changed the template's logo and footer	Nora

Table of Contents

1. Introduction

1.1. Document identifier

PRJ-02-SPC

1.2. Scope

Our project,"Weather monitoring station " is a IOT project that

- Using Raspberry Pi
- Collects weather data such as temperature and humidity
- Compares data to norms
- Create an alerting mechanism for low or high temperature and humidity with LED indicator and Buzzer.
- Turning on cooling system
- Technical documentation

The project will be focused on:

- Reading accuracy data
- Not to fail in runtime

The project is constrained by:

- Returning only temperature and humidity
- Not good at connecting to the sensor every time
- Sometimes failed to read from sensor
- Limited Sensor Integration
- Budget Constraints
- Project Timeline

Key milestones:

- Raspberry Pi Setup
- Sensor Integration
- Data Collection
- Alerting Mechanism
- LED Indicator
- Buzzer Integration
- Cooler Integration
- Testing
- Documentation

1.3. Definitions of Terms and Acronyms

- LED : Light emitting diode
- GPIO: General-purpose input/output
- BLE: Bluetooth low energy
- BTSMP1: Sensor
- Buzzer: An audio signaling device
- I2C: Inter-Integrated circuit.communication protocol
- GROUND: Common reference point for electrical circuits
- PWM: Pulse Width Modulation
- Power Pins: Provide electrical power to a device or component.

1.4. References

Key documents and links supporting this proposal are listed below as reference:

- GITHUB Link: https://github.com/WeatherSensor/Weather_Team
- Sensor differences: https://www.raypcb.com/dht11-vs-dht22/
- Intro To Raspberry PI:
  https://www.argenox.com/library/bluetooth-low-energy/using-raspberry-pi-ble/
- Define Raspberry PI: https://raspberryexpert.com/find-raspberry-pi-ip-address/

2. Problem Statement

2.1. Overview of Solution

High-Quality Sensors: We have integrated a range of high-precision sensors to measure various weather parameters, including temperature, humidity.
Data Acquisition and Processing: The station collects data continuously and processes it in real time. Data is recorded and stored for further analysis.
Reliable Communication: To transmit data to a central server or database, our station employs reliable communication methods such as Wi-Fi.
Alarm System: Our station includes a buzzer and cooler to provide immediate alerts for extreme weather conditions.

2.2. Source Code Organization

File Structure:

- Naming conventions: All files should have descriptive names that reflect their contents.
- Grouping related files: Related files should be grouped together in subdirectories based on their functional area or component.
- Code file format: Code files should use the appropriate file format, such as .cpp .h for C++.

Code Style:

- Indentation: Code should be indented using a consistent number of spaces or tabs to indicate code blocks.
- Line length: Lines of code should be limited to a maximum length, typically around 80-100 characters, to improve readability.
- Code comments: Comments should be added throughout the code to explain its functionality,
- Consistency: The code style should be consistent throughout the codebase, with a single style guide used by all developers.

Code Documentation:

- In-line comments: Comments should be added throughout the code to explain the purpose and function of each section of code.
- Docstrings: Functions and classes should include docstrings.

3. Current Status (As of 05.09.2023)

As of the latest update, our weather monitoring station project has reached an advanced stage of development, with approximately 90% of the planned work successfully completed. The project team has made substantial progress in implementing key features and functionalities. Notably, we have integrated critical components such as weather sensors, data acquisition systems, and communication infrastructure, enabling the station to collect and transmit real-time weather data reliably. Additionally, we have added essential elements like a buzzer and cooler to enhance the station's capabilities and ensure optimal performance. With most of the core functionality in place, we are now focused on fine-tuning and optimizing the system, conducting thorough testing, and preparing for deployment. This significant milestone brings us closer to achieving our goal of delivering a robust and efficient weather monitoring solution to our stakeholders.

4. Dependencies, Assumptions, Risks

Dependencies:

Software dependencies:

- Raspberry Pi Imager
- Other libraries such as pigpio,functional,csignal,regex,thread,string

Hardware dependencies:

- Raspberry PI

Data dependencies:

- A dataset of temperature and humidity(these should be taken from the sensor)
- Convert hexadecimal data that was taken from the sensor to a decimal

Assumptions:

- Weather Patterns.
- Data Accuracy.
- Stable Environment.

Risks:

- We had risks related to temperature`s minus.
- The program may fail to connect to the sensor.
- We had risks related to code that was not improved as well,but now we fixed it.
- Data Inaccuracy.
- Operational Risks.

5. Schedule and Effort Estimations

Task	Description	Start Date	End Date	Team members	Estimated effort(in days)

1. Project preparation	Examine the project	18.06.2023	25.06.2023	Ani Tamaryan(team lead), Karlen Matevosyan, Arman Saghatelyan,Marieta Sahakyan,Diana Marikyan,Nella Nersisyan	7
2.Meeting with mentor	Meet with mentor Taron Kalashyan to discuss the project and finalize the plan.	26.07.2023	26.07.2023	All team members	1
3.Meeting with team at Zealous Lab	Meet with a team to work together and distribute the work between teammates.	27.07.2023	27.07.2023	All team members	1
4.Sensor Integration	Connect the sensor to the raspberry pi with BLE.Implement c++ code to read data from sensor	28.07.2023	29.07.2023	All team members	2
5.Data collection	Create a data collection script to read temp and humidity from the sensor at regular intervals	01.08.2023	03.08.2023	All team members	3
6.Alerting Mechanism	Define threshold values for temp and humidity that trigger alerts	04.08.2023	06.08.2023	All team members	3
7.LED Indicator	Connect LEDs to GPIO pins of the Raspberry pi.Implement c++ code for that	08.08.2023	14.08.2023	All team members	6
8.Buzzer Integration	Connect buzzer to GPIO pins of the raspberry pi and implement code for that	16.08.2023	18.08.2023	All team members	3

9.Testing	Improve and optimize code for the maximum	20.08.2023	01.09.2023	All team members	12