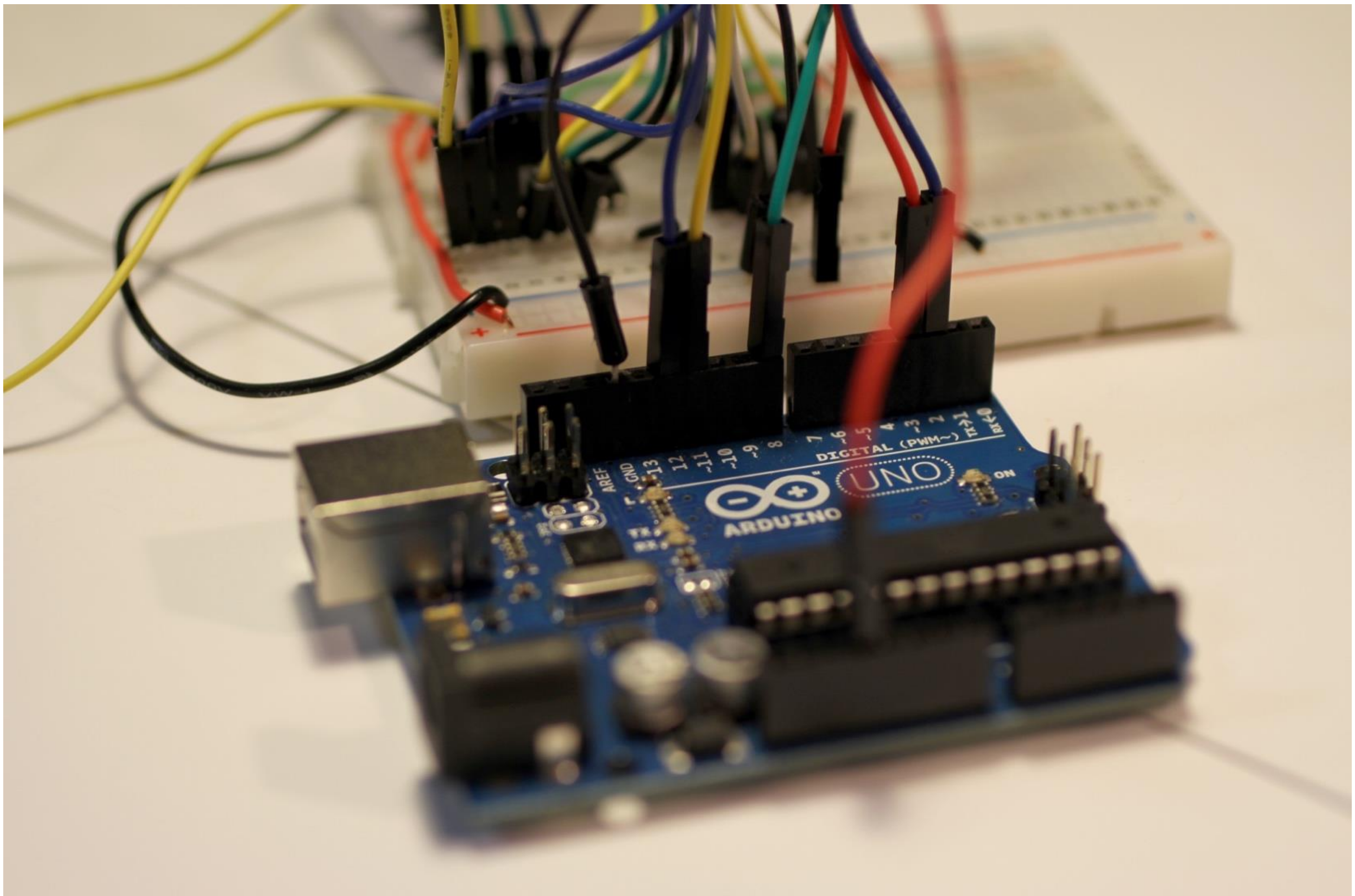
Energy Management in Low Power Wireless Sensor Networks

Preliminary Design



[1] *Cover page image kindly provided by Visual Hunt [1]*

Energy Management in Low Power Wireless Sensor Networks Team 14

Authors
Peter Bouvy(21299044)
Yung Ren Chin(21247413)
Aaron Hurst(21325887)
Khanh Tan (Jamie) Phan(21326604)
Matthew Ramanah(21317297)
Jake Sacino(21132001)
Andy Ta(21317377)

Preliminary Design

Project Partner:
Mr Mark Callaghan, ATAMO

Supervisor:
Mr Marcus Pham

Unit Coordinator:
Dr Sally Male

Group Meeting Day and Time:
Thursday 4pm

Word Count:
3392

Word Limit:
2500+ 914(from requirements) = 3414

Version 2.0

[1] *Cover page image kindly provided by Visual Hunt [1]*

Revision History

Date	Version[2]	Description	Author
12/09/2017	1.0	Significant updates to template. Personalised document while maintaining standardised format. Introduction	Jake Sacino
13/09/2017	1.1	Design Architecture	Jamie Phan
13/09/2017	1.2	Volume IV summary, Resources	Matthew Ramanah
14/09/2017	1.3	Volume VI summary, editing	Andy Ta
15/09/2017	1.4	Volume V summary	Peter Bouvy
15/09/2017	1.5	Volume II summary, relevant requirements, editing	Aaron Hurst
15/09/2017	1.6	Volume I summary	Yung Ren Chin
15/09/2017	2.0	Final Edits	Peter Bouvy

[2]*Incrementing the version by 0.1 denotes a minor change; incrementing by 1.0 denotes a significant change*

Contents

List of Figures

List of Tables

1 Introduction

The proliferation of data monitoring technologies has enabled multifarious insights into environmental conditions, system reliability, consumer behaviour, and a myriad of other fields and industries [2]. Sensors are one such technology, able to detect events or changes in their environment and send this information to other electronics. Energy storage poses a constraint on sensor operation in rural environments; a lack of grid connectivity exposes the importance of sagacious energy management.

1.1 Overview

Cloud Seven Consultants (CSC) has been contracted by ATAMO (the client) to investigate energy management in low-power wireless sensor networks (WSN). The project requires integration of a third party WSN with an ATAMO Arduino based development platform. Sensor data must be measured and periodically reported over the WSN to a cloud-based database. The firmware in the sensor must provide for reliable communications while keeping tight constraints on battery energy usage.

1.2 Purpose

The purpose of this document is to outline a high-level preliminary design of the WSN for ease of comprehension.

1.3 Structure & Contributions

CSC has identified three core sub-systems within the WSN – Hardware, Kernel and Front-End – as recommended by the client [3]. Individual sub-teams have been assigned to each core sub-system. The Hardware team is tasked with designing and fabricating the interposer printed circuit board (PCB) for the duinoPRO-Dusty module. The Kernel team is tasked with developing the firmware, communication packet structure for sensor motes and optimising battery usage. The Front-End team is tasked with data visualisation via the integration of cloud services.

This report is a high-level overview of the whole-system design. Detailed design decisions are outline in individual team volumes as listed in Table 1. Report sections cover the design architecture, summarise individual volumes, requirements, resources, system cost and top three risks.

Table 1: CSC's Individual Contributions

Volume	Contributor	Sub-team	Content
I	Yung	Hardware	Hardware Design & Communication Protocol
II	Aaron	Hardware/Kernel	Sensor Driver & Configuration
III	Jamie	Kernel	System Integration & State Management
IV	Matt	Kernel	Mesh Network Integration
V	Peter	Kernel	Network Gateway
VI	Andy	Front-End	Cloud Integration
VII	Jake	Front-End	Graphical User Interface

1.4 Definitions, Acronyms, and Abbreviations

API	Application Programmers Interface
AWS	Amazon Web Services
GUI	Graphical User Interface
HTTP	Hypertext transfer protocol
IP	Internet Protocol
PCB	Printed Circuit Board
TCP	Transmission Control Protocol
The Client	ATAMO
UART	Universal Asynchronous Receiver/Transmitter
WSN	Wireless Sensor Network

2 Design Architecture

This system implements a site-based wireless sensor network (WSN) which uses the Transmission Control Protocol/Internet Protocol (TCP/IP) to remotely transmit data to a client/developer for analysis whilst also being capable of remote configuration by the client/developer.

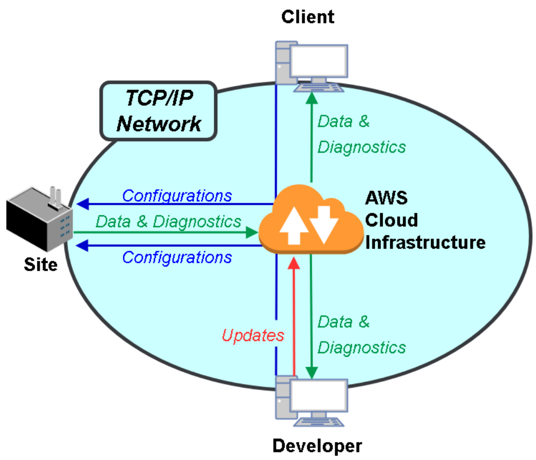


Figure 1. Transactions between system end-points

Figure 1 shows the end-points of the system and their transactions. Sensor data and diagnostic information are pushed from the site to the cloud infrastructure hosted on Amazon Web Services (AWS), which provides storage and analysis services. Commands can be sent to the site to configure sensor-hosts and updates can be patched by developers for the cloud services. Figure 2 provides a more detailed view, exposing key sub-systems required for these transactions.

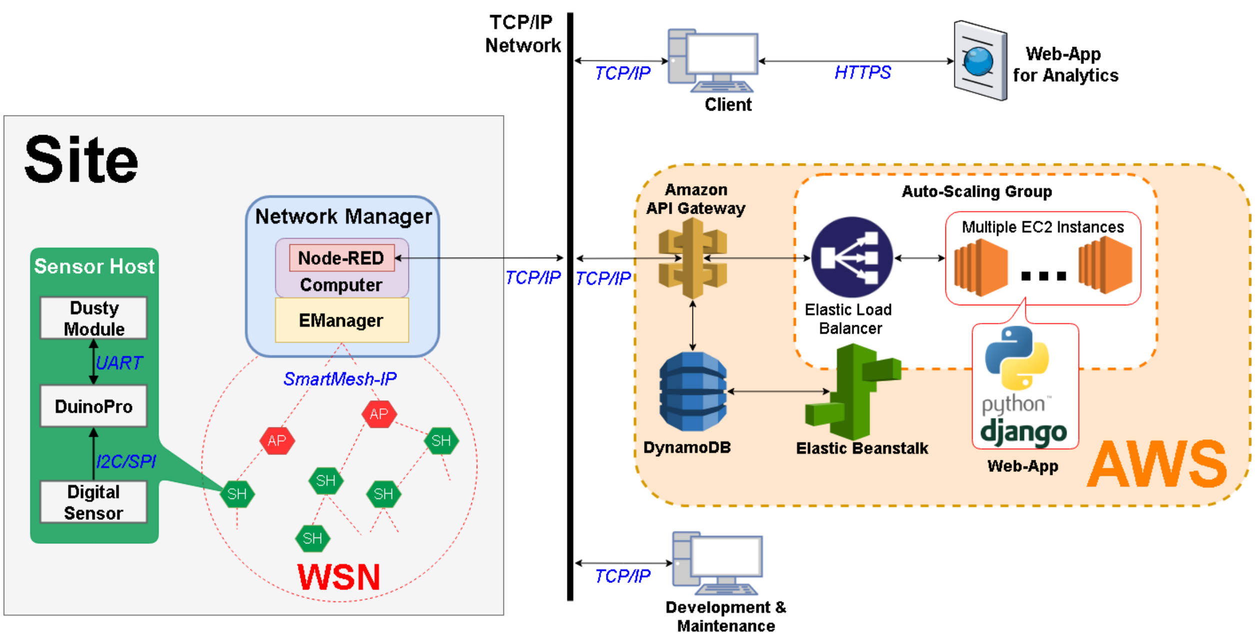


Figure 2. System architecture showing key sub-systems and interfaces

Sensor-hosts are placed within the site to create a self-forming multi-hop mesh network that interfaces with the client/developers with TCP/IP. The sensor-hosts within the WSN are AVR-based microcontrollers (duinoPRO) with digital sensors and Dusty modules which provide communication using Linear Technology's SmartMesh-IP network protocol. Sensor-hosts sample data and diagnostics, and transmit/forward the payload to the centralised network-manager node. The network manager uploads this data using the Node-RED service to AWS.

The payload standard that is sent throughout the network is shown in Figure 3

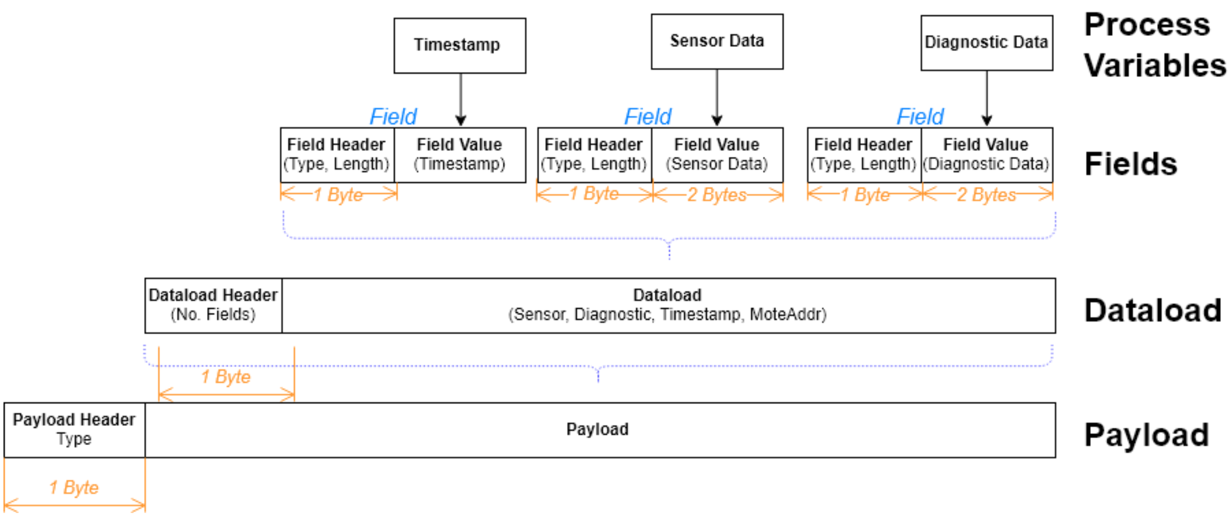


Figure 3. Payload structure and evolution from process variables.

The sub-systems in AWS integrate with one another to perform two key tasks:

1) Parse data sent from the WSN and store it in a structured format for querying, and
2) Serve data and controls through a web-application accessible through a HTTP-browser.

The web-application accesses the cloud database, DynamoDB, to process the data for analysis and visualisation. Further, it exposes interfaces that allow users to configure the WSN remotely. The same payload structure is used for this task, with each configuration parameter represented as one field. The web-application is designed with Python Django and orchestrated with Elastic Beanstalk. Orchestration in this context refers to the automatic-deployment and automatic-scaling of infrastructure to serve web-application to clients. This includes virtual servers for hosting the application (EC2), load balancing, and the DynamoDB interface for querying. Additionally, Elastic Beanstalk provides a deployment service that allows developers to rapidly patch the web-application.

Finally, the Amazon API Gateway service provides a means of authentication and interprets and directs in/out-bound streams.

3 Summary of Volumes

A brief summary of the scope and key results for each of the seven individual volumes associated with this report is presented below.

3.1 Volume I: Hardware Design & Communication Protocol

This volume covers the design of the interposer PCB required to adapt the Dusty module to the duinoPRO. CSC has chosen API UART Mode 2 as the hardware communication protocol due to its energy-efficiency. In addition, CSC has chosen EasyEDA as the PCB vendor to fabricate the PCB as it meets design criteria and is cost effective.

3.2 Volume II: Sensor Driver & Configuration

This volume covers designs for two classes of subroutines. The first class is for interfacing with the sensor module connected to the duinoPRO microcontroller. This task has been divided between high- and low-level routines for sensor communication. The second class is responsible for configuration of the mote – loading default parameters from non-volatile to program memory and updating these whenever configuration packets are received from the network manager. This volume also includes the project timeline.

3.3 Volume III: System Integration & State Management

System integration ensures all sub-systems and their interfaces are considered from a holistic standpoint. System integration is managed using design patterns and principles that manages software scope/encapsulation and error handling. This volume further details the state management of the AVR-based microcontroller; the state variables and transitions made during the device's operational lifetime. In particular, sleep management is implemented to ensure optimized battery usage.

3.4 Volume IV: Mesh Network Integration

Individual motes will interface with the network using Linear Technology's SmartMesh IP network, consisting of a number of mesh systems between motes that relay data to the network manager. When attempting to join the network, each mote will implement a duty cycle management system that prolongs the battery life of the mote in line with the client's requirements. Operational battery life will then be managed by a complex scheduling routine designed to minimise average current while still maintaining reliable sampling at a user configurable rate.

3.5 Volume V: Network Gateway

This volume discusses the design of the gateway. This device collects and processes all data sent to the network manager from the motes and uploads it to the cloud database. It also sends any data and commands specified by the user to the motes in the local network. CSC has chosen to implement the SmartMesh IP Embedded Manager due to its simplicity to configure and the limited time frame in which to develop this system. The application running on the Gateway will be programmed in Python 2.7 using the Smart Mesh SDK running on a Raspberry Pi 3.

3.6 Volume VI: Cloud Integration

Upon the successful data extraction from the WSN, the network manager will store its datasets into a NoSQL cloud database. CSC has chosen AWS as the cloud service provider due to its inherent modularity and flexibility in developing basic applications when compared to IBM Watson.

In utilising the AWS resources, CSC has chosen Amazon DynamoDB for its ease of deployment and schema-less characteristics. The implementation of DynamoDB satisfies all the relevant mandatory and aspirational requirements. This volume also includes configuration management.

3.7 Volume VII: Graphical User Interface

This report pertains to the development and design of the project's web-based graphical user interface (GUI), the link between the end user and the information that is being carried over the network. This includes the selection of the optimal web application framework, the design of the visual interface, and the deployment of the web application. Although the web application's integration with the database will be considered, it excludes design decisions specific to the selection of the database's infrastructure.

4 Requirements

The extent to which key requirements are expected to be achieved based on the preliminary design is discussed below. The full list of updated design requirements is included in Appendix A – Updated Requirements Table.

Requirement M03 is addressed by the Preliminary Design documentation as a whole.

Volume II details how requirements M07 and M08 will be achieved with regard to obtaining the data readings from the sensors. It also describes how requirement M11 will be addressed. However, it is also noted that M11 may not be fully realised in the first-test prototype due to prioritising meeting and testing most mandatory requirements at the expense of delaying the implementation of M11.

Volume II also details how A03 will be achieved from the perspective of receiving and updating configuration parameters at the sensor motes.

Volume V discusses how requirements A01 and A03 will be satisfied using a flexible packet structure. It also highlights how the configuration of the network manager and gateway will satisfy requirements M06 and A02.

5 Resources

Due to the complex nature of the architecture, CSC has leveraged a multitude of resources to assist in the design and implementation of the project. To maintain transparency and quality assurance, the team leveraged GitHub to establish a version control system for all documents used throughout the project [4].

5.1 Hardware

PCB design will be achieved through EagleCAD, enabling the team to produce a schematic and layout that aligns with electrical protocols and manufacturing standards [5]. EagleCAD was chosen due to the team's previous experience with the software and the portability of the final design. EasyEDA, a web-based electronic design automation tool, was selected for the fabrication of the PCB due to its ease of use and competitive pricing [6].

5.2 Network

The team has selected a DC2274A-A device to act as the Network Manager. The device was chosen for to its compatibility with the Smartmesh IP that our wireless sensor network implements, as well as the portability associated with a bidirectional USB connection to the gateway [7]. A Raspberry Pi will act as the gateway between the network manager and the database due to its compact size and cost.

5.3 Front End

CSC has selected AWS as the cloud services platform for the front-office portion of the project. AWS promotes compatibility with many software applications that are able to leverage for cloud-based tasks [8]. Implementation of the database will be achieved using DynamoDB, Amazon's NoSQL database service that handles the routing of data requests, enabling the user to scale the dataset while still maintaining speed and reliability [9]. The team also intends to leverage Node-RED and Node.js to visualise, troubleshoot and optimise the flow of data when needed [10]. Development of the GUI will be achieved by utilising AWS Elastic Beanstalk, enabling simple management of individual applications in the AWS Cloud without having to interface the infrastructure behind each application [11].

6 System Cost

The WSN comprises of hardware, software and human resource costs [12]. The hardware cost solely consists of the interposer PCB fabrication used to attach the Dusty to the duinoPRO. The software costs are the Asian-pacific marginal provisioned throughput rates used in data transactions. Details information about the PCB vendor or cloud-base service provider and service cost are tabulated in Table 2.

Table 2: System Cost

	Vendor or Service Provider	Type of service	Details	Cost
Hardware	EasyEDA	PCB Fabrication	Fabricating PCB with dimension of 29.27 x 28.54mm and castellated mounting holes.	$33.81
Software	AWS DynamoDB	Provisioned throughput cloud database storage	Write capacity unit	$0.00074/hr
			Read capacity unit	$0.000148/hr
Human Resources	CSC	Engineering design	The engineering costs pertaining to the development of the WSN	$120/hr

7 Top Three Risks

Risk	Consequence
Incorrect termination of a pin required for a critical dusty module feature	Significant time delays to project; possibility of unmet deliverables
Testing environment does not mimic real world operating conditions	Systematic underrepresentation of expected battery lifetime
Inappropriate handling of electronics damages components	Significant time delays to project; possibility of unmet deliverables

8 Conclusion

In recognising the importance of the client's definition of quality, the design decisions have systematically prioritised the mandatory and most desired aspirational requirements respectively.

9 References

[1] G. Levine. (2017). *Arduino* [Image]. Available: https://visualhunt.com/photo/126201/.].

[2] M. Yen, "Wireless sensor network monitors structural data and environmental contaminants," (in English), *Materials Performance,* Article vol. 47, no. 3, pp. 19-20, Mar 2008. Available: <Go to ISI>://WOS:000253665300005

[3] A. Ta, "ELEC5552 Team 14 Minutes 170810 Week 2," 2017, Available: ELEC5552 LMS Team 14 File Exchange.

[4] GitHub. (2017). *How Developers Work*. Available: https://nodered.org/about/.].

[5] Autodesk. (2017). *PCB Layout Software for Every Engineer*. Available: https://www.autodesk.com/products/eagle/features.].

[6] Easy EDA. (2017). *An Easier EDA Experience*. Available: https://easyeda.com.].

[7] Linear Technology. (2017). *SmartMesh IP USB Network Manager*. Available: http://www.linear.com/solutions/5744.].

[8] Amazon. (2017). *Cloud Computing with Amazon Web Services*. Available: http://www.linear.com/solutions/5744.].

[9] Amazon. (2017). *Amazon DynamoDB Documentation*. Available: https://aws.amazon.com/documentation/dynamodb/.].

[10] Node-RED. (2017). *Flow-based Programming*. Available: https://nodered.org/about/.].

[11] Amazon. (2017). *What Is AWS Elastic Beanstalk?* Available: https://nodered.org/about/.].

[12] P. Bouvy, "ELEC5552 Team 14 Minutes 170824 Week 4.2," 2017, Available: ELEC5552 LMS Team 14 File Exchange.

[13] J. Phan, "ELEC5552 Team 14 Minutes 170907 Week 6.2," 2017, Available: ELEC5552 LMS Team 14 File Exchange.

10 Appendices

Revision History

Date	Version	Description	Author
12/09/2017	1.0	Initial project requirements	Andy Ta
13/09/2017	1.1	Updated configuration flexibility aspirational requirement [13]	Andy Ta

10.1 Appendix A – Updated Requirements Table

Table 3: Mandatory Requirements

ID	Requirements	Business Need(s)
M01	Functional Operating Conditions - Operational ambient temperature: -40 to 70°C - Fully IP rated (IP68 or above) - Input Battery Voltage 3.0 to 3.6 V - Humidity: 10-90%	Continual system functionality in expected operating environment
M02	Safety: - The system must be safe to operate, install, and maintain with necessary precautions taken to ensure the system will not cause injury or damage.	Reduce danger to human life to as low as reasonably practicable
M03	Design strategy: - A clear and defined strategy for implementation of the system must be provided.	Ease of implementation, maintenance and enhancement
M04	Operational Lifetime: - Sensor devices will operate for no less than 1 year for every 1000 mAh of battery capacity supplied.	Affordability of deploying large numbers of sensors and instrumentation
M05	System Network: - The system must be capable of receiving data remotely from at least 3 sensor devices, with each device separated by a maximum of 50m in open air - The sensor hosts must be discoverable and able to synchronise to the WSN given they are within 50 m (in open air) of another SmartMesh-IP enabled host (sensor host or network manager).	Provides additional data points to alleviate domain-based errors Automates the process of data procurement

ID	Requirements	Business Need(s)
M06	Internet Connectivity: - The system must be capable of sending sensor data to a cloud-based database system from an internet-enabled gateway.	Continual data access
M07	Data validity: - Sensor data that is sent through the network (from sensor to gateway) must be valid, such that errors can be detected and stopped before transmission.	Reliable data integrity
M08	Data Specification: - Data that is transmitted from each host must contain the timestamp of the sensor data, the source of the sensor data, and the sensor data itself - The sensor data must be precise to at least 16-bits of resolution.	Provides value where the time and location of sensor data is of practical significance
M09	Data currency and timeliness: - The user must be able to view current data from the sensors from an internet-enabled gateway with a maximum of a two (2) minute delay	Provides near real-time access to monitored data
M10	Client Application: - A clear and relevant user interface that can access and interpret the cloud-based database must be provided for demonstration purposes.	Provide convenient and coherent access to monitored data
M11	Portability: - Capable of adapting any sensor module that is designed to the ATAMO duinoPRO specification.	Provide cross-functionality across a wide range of end-uses
M12	Hardware Interface - Interface between dusty module and duinoPRO must conform to ATAMO's standard protocol	Provide cross-functionality across a wide range of end-uses

Table 4: Aspirational Requirements

ID	Ranking	Requirements	Business Need(s)	Client Priority
A01	1	Data Specification: - In addition to sensor data of at least 16-bit resolution, diagnostic data of 16-bit resolution should be transmitted through the network	Provide critical data on the current condition of the system Accommodate planned downtime and potential maintenance strategies	5
A02	2	System Network: - The system will be capable of receiving data remotely from at least 5 sensor devices, with each device separated by a maximum of 50m in open air	Increased network capacity enables improved data resolution across the monitored site	5
A03	3	Configuration Flexibility - The system design will allow for each mote to have and receive different configurations from the network manager. Configuration parameters will include, but are not limited to, sensor sampling interval and diagnostic data sampling interval.	Ability to flexibly dispatch the network throughout a plant. Ensuring that each mote can be optimally configured to its environment and purpose	5
A04	4	Synchronise Sensor Readings - Ensure that sensor data readings occur at predictable and synchronised times at each node	Reliable and timely data acquisition	4
A05	5	Authentication - Authentication will be provided to a minimum of two users on a single-tenant platform - The users will have identical permissions	Facilitates the procurement of confidential sensor data Demonstrates the potential scalability of the system	4
A06	6	Operational Lifetime - Sensor devices will operate for no less than 2.5 years for every 1000 mAh of battery capacity supplied.	Low and infrequent system maintenance costs High reliability sensing	3
A07	7	Minimise System Cost - Ensure total system cost is no more than $60	Affordability of deploying large numbers of sensors and instrumentation	2

ID	Ranking	Requirements	Business Need(s)	Client Priority
A08	8	Amplified User Experience - Graphical display of time series data - Consideration of colour theory, visual hierarchy, and typography	Ability to rapidly and easily discern trends and irregularities in sensor data	2
A09	9	Virtual Manager - The system should incorporate SmartMesh IP VManager-based network management to replace local embedded-based network management - A remote x86 based virtual machine with the SmartMesh-IP VManager installed should be capable of performing network management functions when connected to a SmartMesh-to-IP gateway that is local to the WSN site	Ability to more easily scale and reconfigure network	1
A10	10	Database Scalability - The database design may be capable of accommodating for data sent by hundreds of sensors - This scalable database design should be incorporated into the cloud system with considerations for the 'Big Data' movement	Ability to install extensive monitoring equipment throughout a plant and manage the associated data	1