IoT Enabled Smart Inventory Testing & Validation Document

`

Adil Saldanha | Ammar Rehan | Melika Salehi | Wency Go

April 8, 2020

Group 110

Contents

Change Log

Date	Author	Summary of change	Section pointer
Feb. 9, 2020	A.R.	Changed validation document for low bandwidth communication and end device	Validation section
Feb. 9, 2020	A.R.	Changed validation document for low bandwidth communication and end device	Testing section
Feb. 8 2020	M.S.	Edge device added to test section	Testing section
Jan. 29, 2020	A.S	Load cell calibration method added	Testing section
April 4, 2020	A.R.	Made additions to validation methods/concerns	Validation Section
April 4, 2020	A.R.	Made additions to testing methods/concerns	Testing Section
April 4, 2020	A.S.	Added testing method for power test	Testing Section
April 4, 2020	A.S.	Added additional testing method to check sensitivity is sufficient	Testing Section
April 7, 2020	M.S.	Corrected A.R. validation method	Validation Section
April 7, 2020	M.S.	Added additional testing methods for end device, communication & edge device	Testing Section
April 8, 2020	M.S.	Added additional validation methods	Validation Section

Validation

Module	Requirement	Validation Method	Results
POWER	End and edge devices must continuously operate; powered through reliable source	Edge device powered through wall socket	Being plugged into the wall showed to be reliable, no outages were observed in 24 hour period.
		End device powered through wall socket	
		Edge device powered through battery / USB	
		End device powered through battery/USB	Being powered through USB showed uninterrupted operation for the 8 hour duration of test
END DEVICE	Send and receive data through a low-bandwidth, low-power communication protocol to the edge device	Zigbee protocol is low bandwidth and low power	Device was able to send and receive signals wirelessly via Zigbee protocol
LOW BANDWIDTH COMMUNICATION	Wireless communication	Use XBee modules to identify transmitted and received data	Displays if reading (receiving) and sending(pinging) data was successful for each connected device.
	Scalable	Manual experimentation of adding multiple end devices to singular edge device and test connection reliability using XCTU	Devices can be added to list of connected devices. Pinging addresses are successfully looped.
LOAD CELL	Must measure to 0.1 gram accuracy	Use an existing scale accurate to 0.1g weigh items and after load cell is calibrated compare measurements of items of load cell to existing scale and make sure items weigh the same.	The scale is accurate for lower masses however as the weight on the scale increases the accuracy goes down which shows the scale doesn't obey a linear model as initially thought.
	In a meeting held in January the client said they would be happy with 780g load limit (the current spec of the load cell)	To check that this accuracy works for our product, we will use 10g smarties packets incrementally count the number of packets on the scale. We have 70 packets which amounts to 700g. If the scale counts all packets successfully the accuracy is acceptable. Use different amounts of water in a 1 litre water bottle (around the 800ml mark) to test the upper limit of the scale.	The scale successfully measures around 700ml. After this the accuracy drops significantly.
EDGE DEVICE	Send data to cloud via high-bandwidth protocol; must be able to submit data to the IoT HUB	Check it can send sample data to do hub by running script provided by Microsoft	Azure HUB has an existing web based UI which can display the metrics of data received in real time

	Send and receive data via low-bandwidth protocol; must be able to receive data through zigbee protocol	Script to check timely reception via request-response protocol	It can be visually confirmed for short strings. Can also use MS excel compare cell value function
	Must be able to operate for 8-10 hours without interruption	Increase the run time after each of time blocks of 1 hrs, 3hrs, 5hrs, 8hrs & 10hrs have completed runs without issue Test wifi downtime handling by turning off the wifi in intervals of 5min, 15 min & 30 min	Confirmation can be made through already existing Azure HUB UI. The IoT HUB UI shows edge device connection activity.
CLOUD COMPUTING	Aggregate data onto Microsoft Azure	View data values, with time stamps and source information on Azure	Power Bi, an azure compatible extension, was used to view data values, time stamps and source of data. Azure IoT Hub itself does not store data without using an additional database or data streaming extension.
DATA TRANSFER	Communication between end and edge device, must be wireless	Transferring signals containing message string between end and edge device via zigbee protocol	The zigbee protocol was tested between the end device arduino and a PC computer. This showed a successful communication protocol. Results are expected to be the same for transferring between the end device and edge device using the same zigbee supported xbee modules
	Communication between edge device and cloud must be wireless	Transferring message strings between edge device and azure cloud, via WI-FI	Data sent by edge device was successfully received on azure platform in real time.
EASE OF USE	End user mist be able to configure inventory settings	Implement a simple UI on end device to allow manual experimentation of changing device parameters such as product ID, inventory changeover quantity, product unit weight	
ROBUST DATA HANDLING	End device must be able to handle interruptions in data handling and processing.	In event of outage, store last couple values read from sensor on board, and successfully forward to edge once operation resumes. Confirm that forwarded to edge device is received through request- response protocol	
	Edge device must be able to handle interruptions in data handling and processing	In event of outage, store last couple received values from end device and push successfully to cloud once operation resumes.	

| SCALABILITY | System should be designed to be scalable for future development and production | The software must be modular and follow standard documentation practices, so it can be expanded and built upon. Such that it allows for adding new functionality and multiple end devices. Zigbee protocol allows for star network topology. Implement star topology with two end devices and one edge device using the end-to-edge communication software framework. | |

Testing

Module	Test	Testing Method	Results
END DEVICE	XBee Communication to the Edge Device	Send an incrementing value through the XBee communication protocol to see if the edge device is able to receive and display data in real-time.	The values were received in real-time by the edge device and worked any time as long as the end device was transmitting data.
	Receiving message check	Checks if ping message received is what is expected. If it is, then sends data, otherwise sends "incorrect code"	The serial monitor shows data sent via serial. Accurately sends data if received ping message was as expected.
	Request-response protocol	Wait for edge device to ping, send receipt confirmation, wait for edge device to confirm	
	Robust data handling ;Load sensor disconnected i.e. garbage values	Script to recognize if the load sensor is disconnected i.e. broken	
	Simulated power outage i.e. device restarting	Script to recognize there is a power outage, by caching time stamps	
LOW BANDWIDTH COMMUNICATION	Receiving message check	Receiving messages are checked if starting and ending identifiers present in the message. Also checks if index values for start and end of message are correct.	Displays "Serial Read Failed" if unable to receive expected response. Otherwise continues reading.
	Sending message check	Check if message successfully sent from edge device if end device confirms receipt of message.	Displays if pinging was successful or not.
	Check edge device parameter validity	Check if Edge device is set at the correct parameters by reading and displaying those parameters. Also checks if user tries to add invalid parameters.	Displays if existing or user-entered parameters are valid. Returns "Invalid" if not.
	Confirm end device ID registration with edge device	Assign a fixed device ID to end device xbee chip	
	Performance in retail environment; physical obstacles in retail environment such as metal shelves and dividers alter performance	Test the device in a retail store, to check if connection can become spotty and how this is impacted through different hours in a day (peak hour vs. slow sales)	

LOAD CELL	Hardware connections	Use a 'beep' test to ensure all wires are connected. Also check that arduino is reading appropriate values; Serial monitor gives "no device connected" or "-1" if connections are not made	All wires beeped showing connections were made by all wires. After soldering connections on Arduino are more stable as expected. Plugging wires into headers causes issues with the electrical connection
	Load cell calibration	Use an existing scale to weigh items and adjust calibration factor in calibration code, until load cell gives correct reading of item weight. (Repeat for multiple items for a more reliable calibration factor).	Getting the correct calibration factor can take some time. The end goal will be for the weighing scale to calibrate itself after being given a known weight.
		Use self-calibration code to run calibration with different weights, then sue an existing scale to compare results	
EDGE DEVICE	Data quality	Use Power Bi for viewing data values submitted to cloud. Check there are no garbage values. Use Power Bi's excel report generator to compare values.	Was tested with sample data. No garbage values.
	Data completion	Ensure all API requests sent are going through. Firewall/ network config can alter this. Use Power Bi's excel report generator to compare data sent vs. data inserted	Not yet tested over secure network. Tested for a small batch of sample data of 30 values.
	Multi-device (star network) topology	Configure xbee chip for 2 end device connection, using request-response data handling protocol	
		Simulate 5 end device connection by cycling through device IDs on two end devices	
		Test reliability of star topology in using request-response data handling	
	Power outage handling	Recognize power outage by caching time stamps	