MxD Final Report Project 16-02-07

New Digital Tricks for Trusted Friends	
Principle Investigator / Email Address	Andrew Dugenske/dugenske@gatech.edu
Project Team Lead	Georgia Tech
Project Designation	DMDII 16-02-07
UI LABS Contract Number	0320170007
Project Participants	Steelcase
MxD Funding Value	$240,567.87
Project Team Cost Share	$311,680.90
Award Date	04/03/2017
Completion Date	12/31/2018

SPONSORSHIP DISCLAIMER STATEMENT: This project was completed under the Cooperative Agreement W31P4Q-14-2-0001, between U.S. Army - Army Contracting Command - Redstone and UI LABS on behalf of the Digital Manufacturing and Design Innovation Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Problem Statement

The Industrial Internet of Things (IIoT) provides great promise for manufacturing. Several sources indicate that it will have over a $10 trillion effect on society over the next few years, with manufacturing seeing 27% of the effect. With increased power of sensors, processors and bandwidth, exponentially more data can be obtained from the factory floor.

Unfortunately, a roadblock to the realization of the IIoT is the lack of communication to and among manufacturing equipment. Some estimates indicate that less than 5% of equipment on the factory floor provide data to manufacturing software applications, and only a small percentage of the data provided is actually acted upon. This is especially true of legacy machines. Many machines currently used in manufacturing facilities are decades old. As a result, they have little to no modern computer control or instrumentation.

Manufacturers want the ability to obtain data easily and affordably from legacy equipment. However, a simple, ‘plug and play’ solution has been elusive. There have been attempts to develop software and hardware equipment adapters, but they have suffered from proprietary lockout, resistance to the high cost of installing computers at each workstation, and extensive configuration requirements.

How was problem addressed.

The purpose of this project is to address this problem by developing and delivering an IoT retrofit kit design that allows manufacturers to exchange data with their legacy machines using an open architecture and standards in a low-cost and easily configurable manner. This is done by using robust single board computers, a unique sensor network, readily accepted manufacturing standards, open source software and an innovative, configurable software application.

The retrofit kit enables manufacturers to easily obtain process parameter data from legacy equipment and send control signals to the equipment. This will allow DMDII members to exchange essential digital information with their old “friends” – their legacy equipment – and thus integrate them into the Industrial Internet of the Things (IIoT) and improve factory operations.

DMDII members can make use of the retrofit kit in a variety of ways. This project will generate all the necessary plans and software needed to produce the kits from readily available, low-cost hardware. Since the kits will make use of standard manufacturing protocols, they can easily be connected to factory software applications. Members can build and install their own kits, or make use of third-party suppliers. Wipro Technologies, a global software supplier and project team member, plans to offer the kit and integration to various software platforms through their network.

We’ve established a strong team of experts from Georgia Tech, Wipro Technologies, and Steelcase. As a team that includes a major research university, a global software supplier and multi-billon dollar manufacturers, we’re confident that we will deliver a successful solution.

Summary of Project Outcomes

The project team was successful in developing several versions of the IoT retrofit kit, defining a standards-based decoupled architecture that can be used to exchange data with the kits, writing a web-based application that was used as a test harness and publishing data to a commercial application (ThingSpeak). Retrofit kits were installed on a variety of equipment at Steelcase factories in Alabama and Michigan. The kits published data to the decoupled digital architecture that was installed in the cloud (Amazon Web Services) and the data was displayed in the web-based application developed for the project and ThingSpeak.

Recommended Next Steps

Recommend next steps include the following:

- Enhance and promote the Decoupled Digital Architecture to reduce the cost and complexity of exchanging data among manufacturing equipment and applications.
- Develop improved versions of the IoT retrofit kit to expand its capabilities to address a wider variety of implementations.
- Encourage MxD partners to develop gateways that will interface their software and equipment to the architecture. Recommend that MxD partners test kits and the architecture in their factory.

I. PROJECT DELIVERABLES

The following list includes all the deliverables created through this project. The files that support the deliverables are contained in separate subdirectories of a private github repository for access by MxD. Readme files throughout the repository provide guidance about the files contained in the repository hierarchy.

#	Deliverable Name	Description	Deliverable Type
1	Architecture Definition	In order to exchange data among the retrofit kits and factory applications, a Decoupled Digital Architecture was used.	Documentation for the architecture is located in the /Architecture subdirectory of project git repository.
2	Design Data Package	An extensive set design of files (e.g. electrical schematics, mechanical designs, software, bill-of-materials (BOM), part descriptions, photos, images) needed to construct and program the retrofit kits and subassemblies have been generated.	The design files are located in the "/Design Data Package" subdirectory of the project git repository.
3	Test Harness	A test harness that provides status about the messages received from the kits that mimics a factory software application was developed.	The files used and produced for the test harness are included in the "/Test Harness" subdirectory of the project git repository.

4	Experience with Thingspeak	ThingSpeak was used to demonstrate the ability of the retrofit kits to publish data to a commercial application.	A report describing the experience and the software developed are included in the /ThingSpeak subdirectory of the project git repository.

Software Deliverables

The software for each deliverable is located in the deliverable's repository directory. Please see the readme.md located in the repository for specifics about the software.

Implementation Guide

The implementation guides for the deliverables are located in the deliverable's repository directory. Please see the readme.md files located in the repository for specifics about the guides.

II. PROJECT REVIEW

Use Cases & Problem Statement

The primary barriers to implementing IIoT retrofit kits in the past have been cost and time. Hence, our proposal addressed these two barriers directly.

As indicated in the original PPD and through our market analysis, manufacturers are unwilling to pay more than $1,000 per machine for connectivity. Truthfully, manufactures desire a $100 - $200 hardware solution. In order to achieve this price target, we plan to make use of commercial-off-the-shelf hardware and open source software whenever possible.

There have been several attempts to develop adapter boxes that allow communication with various pieces of manufacturing equipment. The adapter boxes are typically developed with proprietary communication protocols to allow manufacturing execution or SCADA software companies to communicate with the equipment. The adapters were typically sold at cost, or included in the software sale so that the software companies can make it up from revenue of their core software product.

The project targeted the following primary use cases:

1. **Monitoring the throughput and tool usage of large punch presses.**
   Steelcase makes use of large punch presses to produce components for their products from sheet steel. The presses strike large tools to produce patterns in the steel. After a certain number of strikes, the tools must be refurbished. The retrofit kits were used to count the number of strikes on tools that are used in the press. Figure 1 shows a retrofit kit mounted to one of the presses.

Figure 1. A project IoT retrofit kit monitoring a 400 ton press at Steelcase.

2. **Capital Asset Motor Monitoring**
   There are key capital assets in the Steelcase factory that are driven by a single large industrial motor. If the motor fails unpredictively, production and delivery are affected negatively. Retrofit kits were used to monitor motors in key locations in the factory. The kits collected vibration, current and temperature data to determine if the motor was trending toward a fault. Figure 2 shows a retrofit kit mounted to a critical capital asset motor.

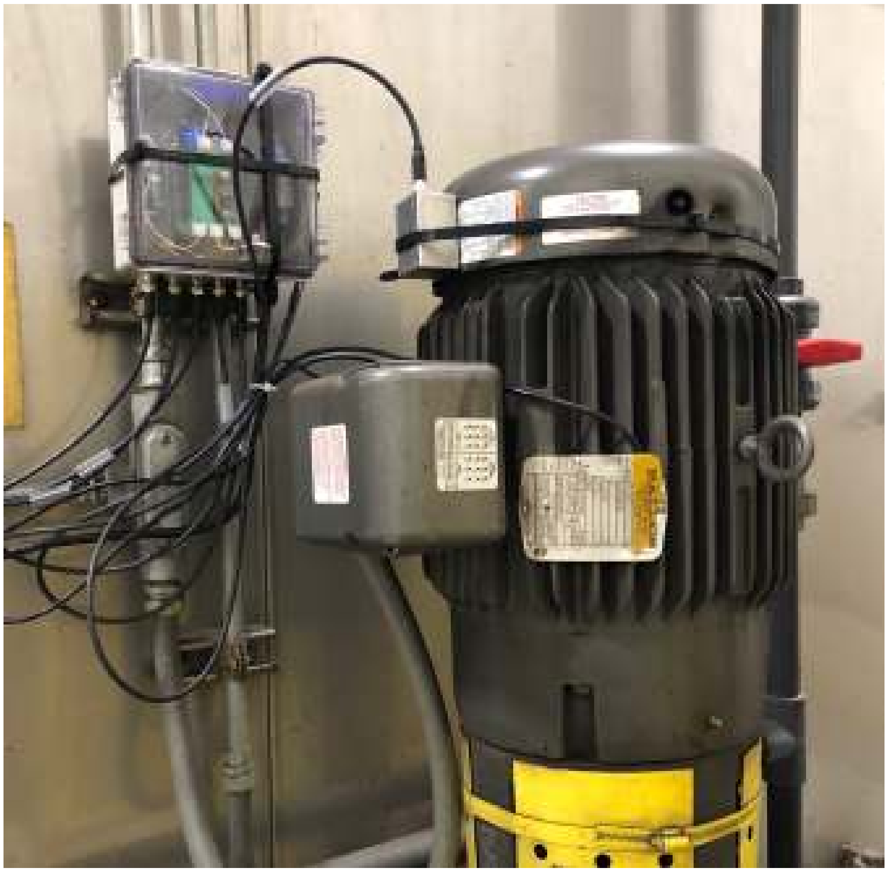

Figure 2. A retrofit kit mounted to a critical motor in the Steelcase factory.

Objectives

The goals for the retrofit kit are as follows:

- The cost of the hardware should be less than $250 in low quantities.
- The kit should be able to receive data from a wide variety of digital and analog sensors.
- The kit should be able to send information about its operation.
- Whenever possible the kit should be developed using open source hardware and software.
- In case of damage, the kit should be easily replaceable in the factory.
- The kit should be easy to configure.
- The time to deploy the kit should be approximately one hour.
- The deployment of the kit should require little or no programming.

Technical Approach

The purpose of this project was to develop an IoT retrofit kit that is capable of collecting data from sensors that are attached to legacy equipment and transmit the data to an application on the network. An architecture for exchanging data with the retrofit kit and applications to display the data received by the kit were also developed and demonstrated.

Retrofit Kit, First Generation

To keep costs low and functionality high, a Linux single board computer, BeagleBone Black wireless (BBBW), was used as the heart of the retrofit kit. The BBBW is a highly capable, low-cost, open-source computer that runs embedded Debian Linux. It can run a variety of programming languages, and has over 64 programmable general purpose input/outputs (GPIOs) that are exposed through two headers. The GPIOs allow the BBBW to communicate using several protocols such as: ADC, CAN, digital inputs or outputs, Ethernet, HDMI, I^2C, SPI, UART, USB and WIFI. Figure 3 shows an image of a BBBW wireless.

Figure 3. BeagleBone Black wireless (beagleboard.org)

The BBBW headers provided easy access to the GPIOs, but a method to securely attach wires to the headers was needed. For the initial versions of the kits, we made use of the Grove BBBW base cape which plugs into the BBBW as shown in figure 4. Latched female Grove connectors attach to the cape, providing an assured connection to the devices that interface to the board.

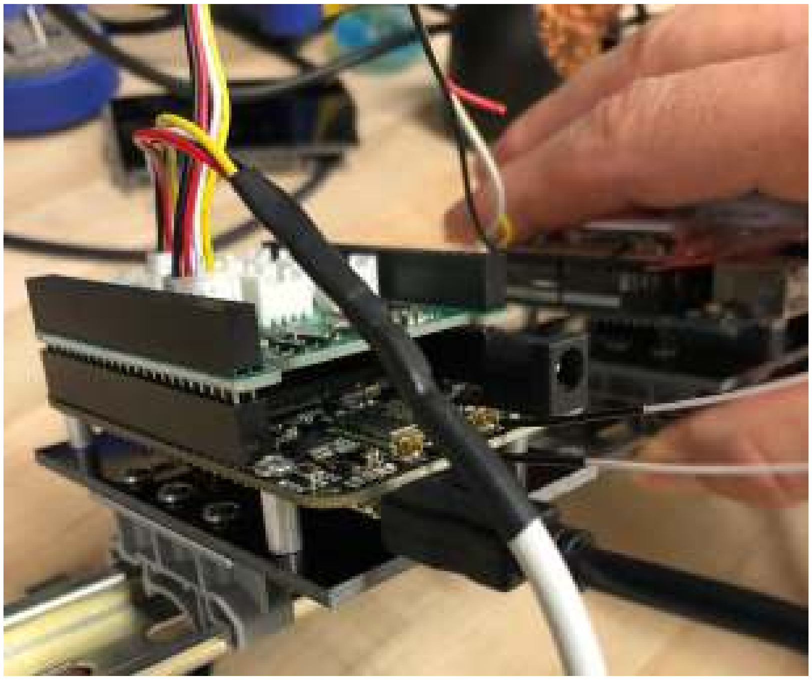

Figure 4. Grove base (green board) connected to BBB (black board).

The BBBW and the cape were placed in an industrial plastic box to protect the components as shown in figure 5. To simplify cable management, clamping strain relief devices were used. Power was provided to the kit through the use of a 5 volt wall charger. Network connectivity was accomplished through the BBBW's WIFI. Sensors were attached to the kit using Grove cables which contain four conductors—2 for signal and 2 for power. Wiring diagrams for the kits and sensors are included in the Design Data Package.

Figure 6 shows this version of the kit mounted to a press at Steelcase in which it counted press strokes using a reed switch. A magnet was attached to the press ram which triggered the reed switch. The kit sent message to the architecture each time the switch was actuated which provided a count of throughput for the press.

Figure 5. Enclosure containing the first version of the retrofit kit.

Figure 6. Retrofit kit attached to 400 ton press at Steelcase.

Retrofit Kit, Second Generation

Through experimentation and testing on the factory floor, the team learned how to improve the retrofit kit to better server potential customers. Figure 7 shows a picture of the second generation kit and figure 8 shows an exploded image of the kit. Some of the improvements that were made included:

- The kit was made more rugged, modular and dust resistant to make the kits more durable in a factory environment.
- Multiple small enclosures were combined into a single large enclosure that contained DIN rail attachment points.
- All the components in the enclosure were attached via a DIN rail.
- The clamping strain reliefs were replaced by cable glands and M8 style industrial connectors. The connectors contained pigtails which allowed secure wiring to the BBBW. M8 cables were used to connect the sensor to the kit.
- The low voltage transformers were mounted inside the enclosure to the DIN rail and power was provided by a single 110v power cord attached to the enclosure through a cable gland.
- A DIN rail adapter assembly was developed for the BBBW which allow the BBBW to be rigidly attached to the DIN rail, but easily removed for maintenance.
- DIN mounted relays were used to electrically isolate proximity sensors from the BBBW and associated electronics.

Figure 7. The second generation of the retrofit kit with DIN rail mounted devices and standardized connectors.

Figure 8. Exploded version of the retrofit kit parts.

Retrofit Kit, Third Generation

The project team continued to learn and improve the retrofit kit from factory installations. Figure 9 shows an updated kit that includes a custom BBBW cape developed to address issues associated with the Grove cape and cables. The new cape also provided deterministic sampling of signals through an included microcontroller.

The Grove BBBW base cape used in the previous kit generation was found to be unreliable. The digital and analog signals that were presented to the cape through the Groove connectors were not being reliably delivered to the BBBW. This was confirmed by comparing the input signals to the voltages measured independently on the pins of the BBBW. The cape delivered the correct signal intermittently which was especially troublesome. The Grove female connectors were also found to be unreliable since the connector pins were crimped to the conductors in a non-robust manner. The cable connections were suitable for hobby projects, but were not sturdy enough for manufacturing environments.

Sampling of signals at greater than 10 kHz was needed to collect current and vibration data from the sensors attached to manufacturing equipment. However, the BBBW runs Debian Linux which is not a real-time operating system so it is not capable of collecting high frequency data at deterministic sampling rates.

To address these issues and others, the custom cape included the following features:

- A Teensy microcontroller was incorporated to accomplish deterministic analog sampling up to 1 Mhz.
- The addition of half-wave rectification circuits to accommodate sinewave generating current sensors.
- Routing of signals to the Teensy or the BBBW through solderable jumpers.
- Improved wire connections through the use of screw-down micro-header terminal blocks.
- A dedicated header for ground and voltage connections.

Configuration software running on the single board computer allow a user to map external signals to definitions outlined in manufacturing standards (i.e. MT Connect data items, or MQTT CAMX event messages), nearly eliminating the need to produce code for a simple application. An open source application, Node-RED, is used to read the data from the BBBW's GPIOs and convert the data into JSON messages that adhere to the Decoupled Digital Architecture.

Figure 9. The third generation retrofit kit with custom BBBW cape (green board).

The Teensy microcontroller can be programmed to sample 8 analog signals up to a rate of 1 Mhz. The sampled data is sent to the BBBW at regular intervals via a UART connection. The BBBW receives the data via the Node-RED flow running on the BBBW. The Node-RED flow converts the array received from the Teensy into a JSON message and publishes it to the architecture's MQTT message broker. Specifics about the custom cape can be found in the

"Design Data Package/Subassemblies/BBB Cape, GTMI IoT" directory of the supporting file repository.

Decoupled Digital Architecture

The Decoupled Digital Architecture for Manufacturing is a high performance, low cost, standards-based, superior alternative to tightly coupled systems that are prevalent in manufacturing. The benefits are gained through the use of an extremely scalable publish and subscribe message broker technology which dramatically reduces the number of unique point-to-point communication paths.

The architecture can accommodate a wide variety of devices, protocols and applications. It is very secure, easy to maintain, simple to install and built upon prevalent IoT standards which makes it well suited for workcell or enterprise implementations. A conceptual representation of the architecture is shown in figure 1.

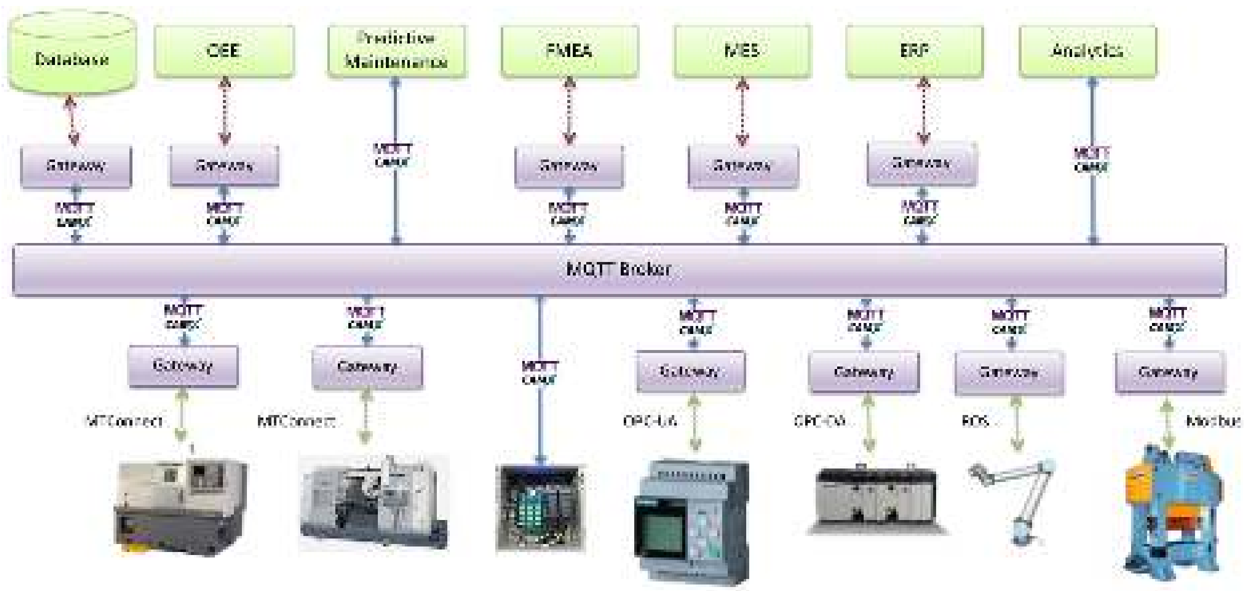

Figure 10. Decoupled Digital Architecture.

Data exchange is accomplished via a publish and subscribe message broker that transfers messages bi-directionally among clients using MQTT. The architecture exchanges standardized messages (based upon the IPC CAMX Standards) which contain manufacturing parameter data and are represented in Javascript Object Notation (JSON). Definitions of the messages can be found in the Messages section of this site. Routing of messages is accomplished through broker topics which are described in the Topics section of this site.

Gateways (as shown in figure 1) are used to convert client communication protocols into the standardized and holistic protocol defined by the architecture. Gateways receive data from machines in various ways (e.g. IPC CAMX, Modbus, MTConnect, OPC-UA) convert the data into standard JSON messages and publish them to the broker using MQTT. Gateways can also receive data from the broker and place it into a database or a format required by an application.

MQTT brokers can be implemented using open-source, leased, purchased or hosted software provided by a variety of vendors.

The messages exchanged in the architecture are modeled after various manufacturing data standards (e.g. MTConnect, CAMX) and structures used in commercial applications. Four base parameters: assetId, dateTime, dataItemId and a value are contained in each message as shown in figure 11. Information beyond the base parameters is added by including additional parameters.

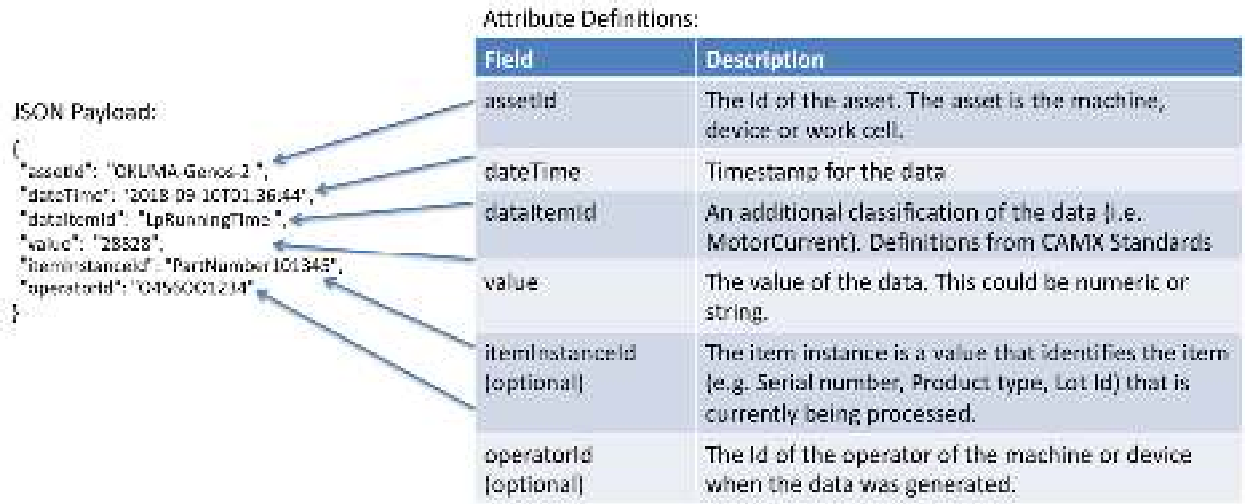

Figure 11. Attribute definitions for standard messages.

Additional information about the Decoupled Digital Architecture can be found in the "Architecture Definition" directory of the supporting file repository.

Test Harness

The project team made use of a Test Harness application prior to the beginning of the project to test IoT devices on the factory floor and display information that is of use to manufacturing engineers. The application was extended to address the functionality of the IoT kits developed through this project. To enable the concept of interconnected systems, a cloud infrastructure was developed as a demonstrable reference implementation and test harness for the retrofit kit. The retrofit kits publish data to the MQTT message broker and the test harness subscribes to the message broker as described in the Architecture section of this document. The test harness acts as a basic manufacturing software application that may be used in conjunction with the retrofit kits to provide manufacturers value.

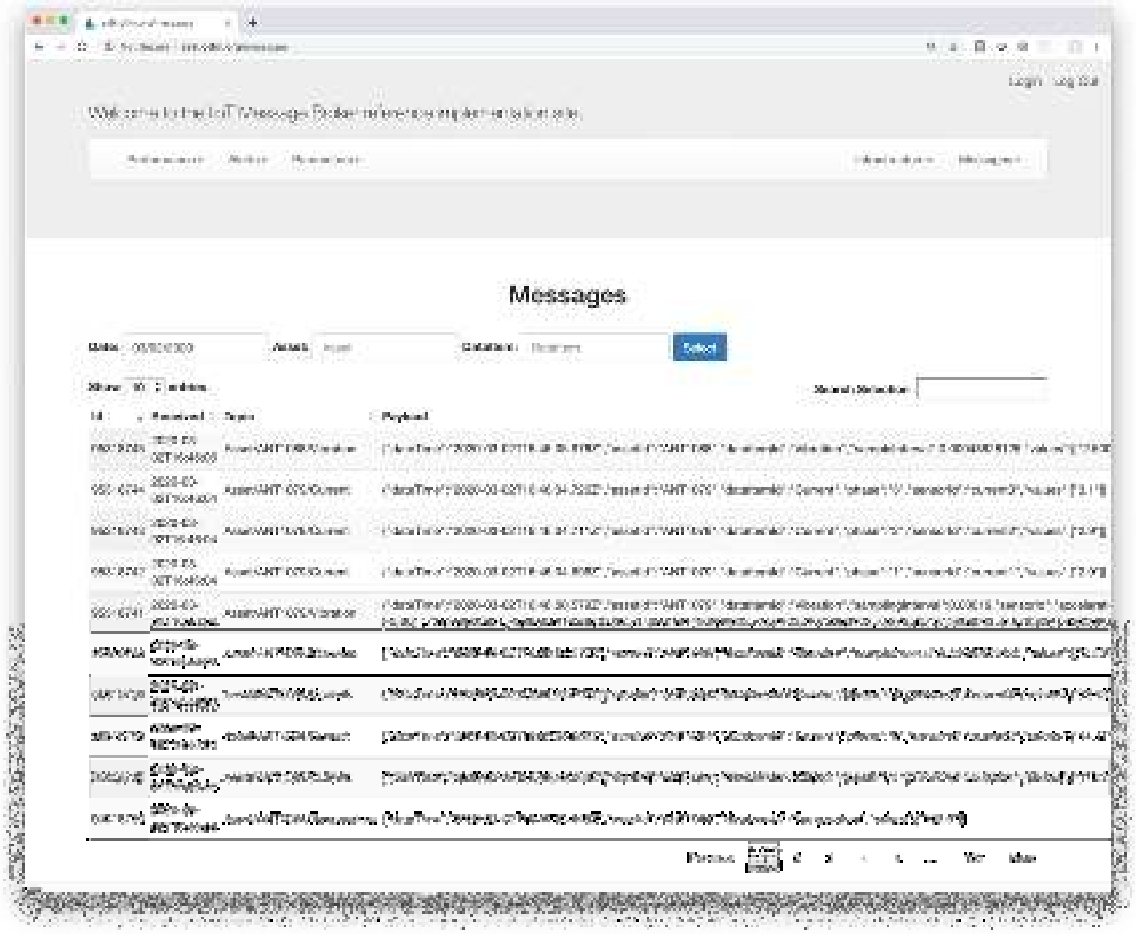

Figure 12. Message produced the retrofit kits are display in the test harness application.

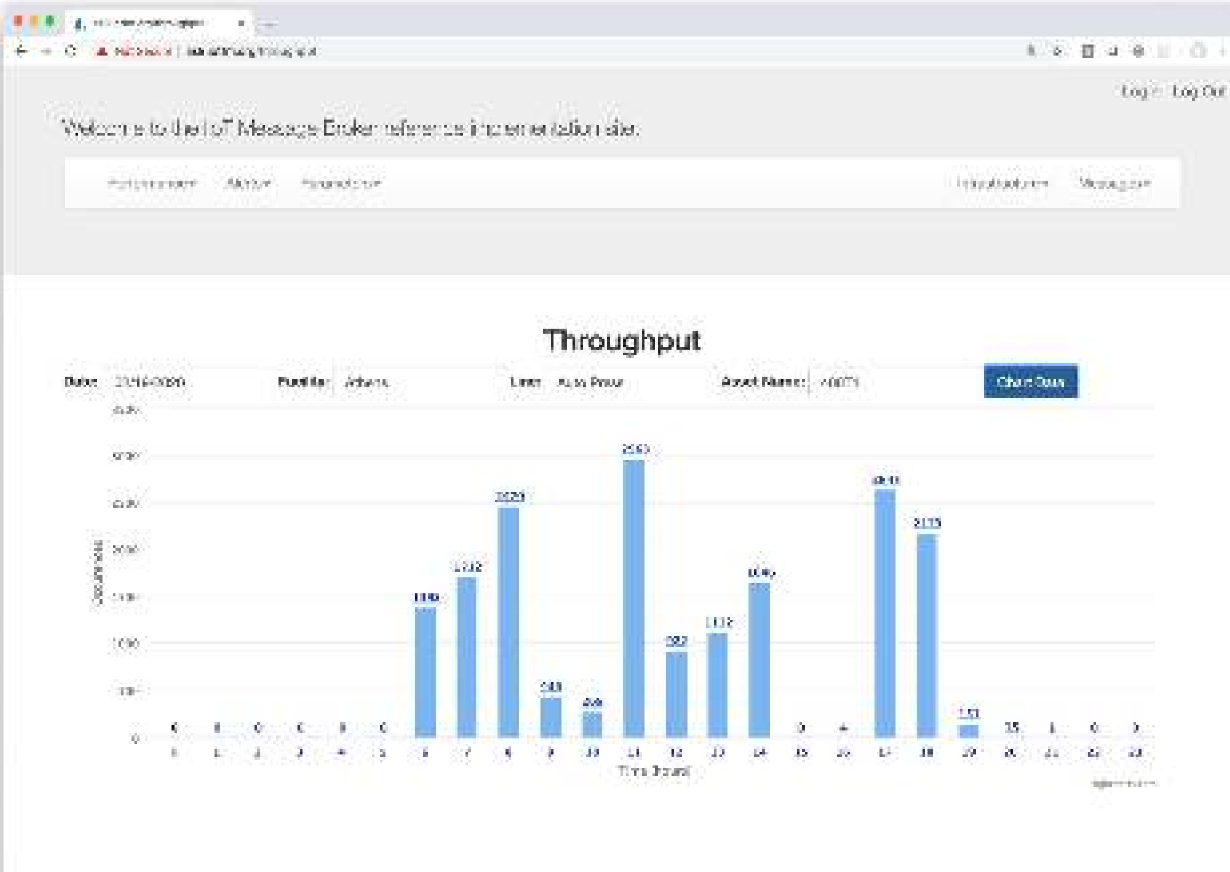

Figure 13. Test harness displaying throughput.

Additional information about the Test Harness can be found in the "Test Harness" directory of the supporting file repository.

Experience with Thingspeak

ThingSpeak was used to demonstrate the ability of the retrofit kits to publish data to a commercial application. ThingSpeak is a web service developed by MathWorks Inc. that allows user to collect and store data in the cloud. The web service also provides applications for users to analyze and visualize data in MATLAB. A full report about the project team's experience with ThingSpeak and the software used to interface with ThingSpeak are included in the "Experience with Thingspeak" directory of the supporting file repository.

Figure 14. Sample project data displayed in ThingSpeak.

Planned Benefits

- Solution kits in future technology will be much cheaper, enabling adoption by manufacturers.
- Plug-and-play solutions will enable manufacturers to focus on connecting the machines to a wider IIoT solutions.
- Open standards used in future technology will enable ubiquitous communication with third party OEMs.
    - Open software will be made available to the DMDII commons, enabling developers to extend code from the future technology and no longer have to start from scratch.
- Lower costs of integration through developing a unified solution.
    - Using low cost components
    - Using open source software
- Wider adoption.
    - Publishing to Digital commons
    - Joint projects with DMDII members

III. ANTICIPATED KPI'S & METRICS

The table below outlines the key performance indicators and metrics used to evaluate the success of the project outcomes in comparison to the current state and proposed goals.

Metric	Baseline	Goal	Results	Validation Method
Remote access of machine state	Not available	Machine to cloud	Achieved	Demonstration through web-application
Ability to Remotely control machine state	Available, but only on specific OEM devices	Available to all kinds of machines through open standards	Did not achieve due to safety concerns	None
Capture machine states at varying rates	Available, but only on specific OEM devices	Available to all kinds of machines through open standards	Achieved	Demonstration through web-application
Security	Not applicable	Makes use of acceptable security protocols	Achieved	The retrofit kit and architecture make use of widely-accepted authentication techniques.
Open standards	Not available	Available	Achieved	The retrofit kit and architecture make use of a variety of open standards: HTTP, MQTT, JSON. No proprietary technology is required by the architecture.
Plug and play capability	Requires high customizing	Same kit can be reused for multiple kinds of machines	Achieved	The same retrofit kit has been used to monitor multiple machines with only slide programming modifications.
Kit cost $500 - $1000	Current costs too high	$200-$300 USD	Achieved	The BOM cost of the kit without sensors is approximately $300.
Flexible and reprogrammable	Custom protocols/OEM constrained	Open standards to enable 3rd party access	Achieved	The kit can be reprogrammed using open source IDEs (Arduino) and web interfaces (Node-RED).

IV. TECHNOLOGY OUTCOMES

TECHNOLOGY DELIVERABLES

An extensive set of files (over 700) were produced to document the project deliverables. To organize, describe and easily deliver the files, a github repository was been established at this location:

https://github.com/edu-gatech-fis/DMDII-16-02-07-Final-Report

Throughout the repository, 'readme' files have been included to help the user navigate the hierarchical structure. The top level of the repository contains the following repositories that represent the major deliverables and activities:

/Architecture

This directory contains the definition of the Decoupled Digital Architecture that was used in the project to exchange data.

/Design Data Package

This extensive directory contains the files (e.g. electrical, images, mechanical, software, bills of material) for the kits and kit subassemblies that were developed for the project.

/Documents

This directory contains general documents for the project.

/Test Harness

This directory contains the software used to develop the test harness for the project.

/Thingspeak

This directory contains information about how the retrofit kits sent data to ThingSpeak and the graphing of the data in ThingsSpeak.

V. TRANSITION PLAN

Potential Commercial Plan

Georgia Tech continues to make use of the decoupled digital architecture and the legacy retrofit kits in its research projects and educational programs. The architecture is being used in projects funded by companies such as Boeing, Delta Airlines, Ford Motor Company, Ingersoll-Rand, Moog, Steelcase and Thyssen-Krupp and the legacy kits are being used in several of their factories and labs. Georgia Tech will continue to promote the architecture and kits used.

The latest version of the retrofit kit was designed for flexibility and industrial survivability. A NEMA hard plastic enclosure was fitted with a DIN rail to simplify mounting devices in the enclosure. Adapters were built that made it easy to mount parts such as circuit boards to the DIN rail. Cable glands and M8 connectors were used to attach sensors and power cords to the enclosure. All the sensors made use of M8 cables so they could quickly be attached and detached from the enclosure. This flexibility was important while the project was ongoing so that changes could be made quickly to accommodate customer needs. However, parts such as the M8 cables, M8 connectors and DIN mounted devices added significant to the cost of the kit.

Due to the success of the architecture and kits, companies are now searching for reliable commercial suppliers of the hardware kits since producing the kits are typically not part of their core mission. They desire to purchase from a reliable company as a 'part number' versus ad hoc builds. Companies have also identified a wide-variety of uses for the kits, so the purchase price is very important since the purchase volume could be large. Companies that sell the kits, however, will need to make a reasonable margin from the sale in order for their business to be sustainable. The BOMs listed for the kits produced for this project only contain the part prices and don't include costs such as assembly. By developing kits that target a specific use (e.g. vibration measurement), costs can be reduced by eliminating parts such as quick disconnect connectors and replacing them with snap-in-place strain relief mechanisms.

Georgia Tech has been discussing potential commercialization of the kits with companies such as Real Time Automation (Pewaukee WI), Prime Manufacturing Technologies (Savage, MD) and Factory Right (Atlanta, GA). Developing a reliable and capable retrofit kit that can be sold for a reasonable profit has been the subject of the discussions.

VI. ACCESSING THE TECHNOLOGY

No Background Intellectual Property (software, designs, data, etc.) is needed to use or modify the technology outcomes.

VII. APPENDICES

Please see the Technology Deliverables section about the github repository that contains supporting files.