Internet of Things (IoT) for Live Monitoring and Analysis of 3D Printers

Anthony Palmer, Trevin Pointer, Hunter Hinshaw and Terry Guo

ABSTRACT: To support research and education in Internet of Things (IoT) security and smart manufacturing, a preliminary research testbed was built recently at Tennessee Tech. This testbed was designed to implement a Machine Health Monitoring (MHM) system on the 3D printers located in the iMakerSpace at Tennessee Tech.

Key Words: Industrial Internet of Things (IIoT), smart/digital manufacturing, digital thread, cyber security, Machinery Health Monitoring (MHM), Predictive Maintenance (PdM), Artificial Intelligence (AI).

INTRODUCTION

BACKGROUND

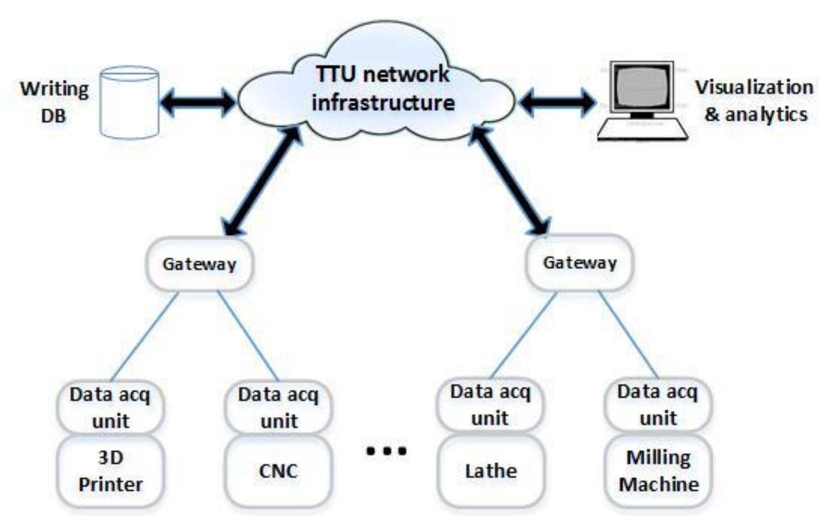
Facing increasing global connectivity and IoT usage, the need arises to ensure integrity between devices and that all devices are up and running. Thus, Machine Health Monitoring (MHM) was created.

MOTIVATION

Real-time fault detection is a must for 3D printers. If a print begins to go bad, it wastes time and money for whoever is printing with every second of faulty printing.

CONTRIBUTIONS

Building off of a previous iteration of the system, the current implementation is a rather large change. The system was originally designed to run through Amazon's Azure service, but now runs off of a small localized server. The sensors' scripts have been updated to properly read data, format it, and send it to the new server. The server itself was a large task, taking a large amount of backend scripting. The website hosted by the server was made completely in-house, and is designed for efficiency and modularity. The entire hardware system was overhauled for a more tidy and manageable approach.



A reference testbed architecture

SYSTEM DESCRIPTION

MAJOR ELEMENTS

The system runs primarily from three devices: a Raspberry Pi 3 that hosts a web server and database, a Raspberry Pi 3 controlling the primary sensors, and a Raspberry Pi 0w running a Pi Camera.

The server Pi hosts the server through Laravel, which is a web framework. Laravel itself implements scripting from JavaScript, Blade, PHP, and SQL to create a website with data visualization, along with a backend database.

The sensor Pi currently runs two sensors: a thermocouple and a humidity sensor. Both of these sensors are developed by Adafruit. The sensor Pi also has an accelerometer and vibration sensor attached, which will be implemented in the future. The sensor data is retrieved by Python scripts containing Adafruit libraries for each sensor, respectively. This data is then pushed to the server at regular intervals.

The camera Pi is also connected to the server; however, it has its own separate interface nested within the website. The camera runs off of a series of shell scripts that enable it to have an interface with built-in camera controls. The primary use of this so far has been to start and stop the camera at any given time. The camera streams the footage live to the server, and as such does not need to be on if it is not being used actively. This prevents overheating on the Pi itself.

PRELIMINARY RESULTS

WEB ACCESS

Data can be accessed (up to the last 60 entries) through the web server's IP address. Options are available to view the data in real time (i.e. seeing new readings appear as they are captured by the system) or to see it in a stable setting.

The website is designed to host a more modular setup of the system. Sensors are grouped within a specific printer, and new printers can be added at any time. A specific sensor can also be added to the website with ease. The only information required is the IP address of the sensor, the name, and the type of data it is storing.

The server is also considered to be self-maintaining. The server is constantly pinging each IP address connected to it to determine if the sensors are operational or not. It has three cases: a check mark for if the sensor is online and sending data, a warning symbol for if the sensor is online but no data is being received, and a red X for if the sensor is offline.

Real-time data visualization

REAL-TIME VISUALIZATION OF DATA

Shown below is a sample reading from the humidity sensor in our rig. While this reading is not from the inside of a 3D printer, it serves as a valid example of the sensor and visualization at work.

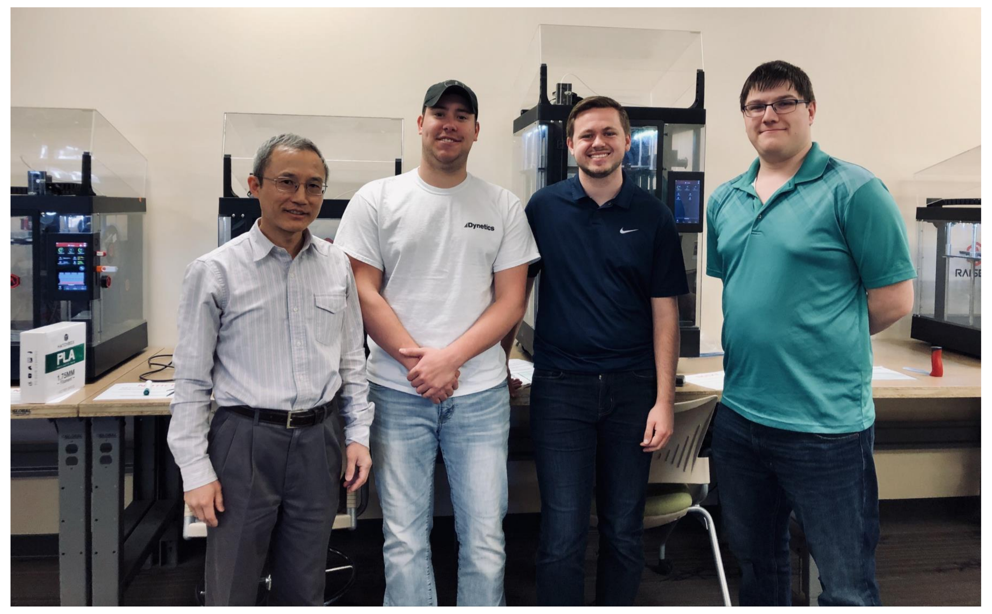
FUTURE WORK

ANALYTICS

- Fault Detection
- Fault Prediction
- Further sensor additions

SENSOR MOUNT

- Design of easy-use brackets



ACKNOWLEDGMENTS

Thank you to the Center for Manufacturing Research for sponsoring the research and to the iMakerSpace for allowing us to use their space and printers for testing and development.

REFERENCES

[1] https://learn.adafruit.com/dht/dht-circuitpython-code
[2] https://github.com/FaBoPlatform/FaBo9AXIS-MPU9250-Python
[3] https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008
[4] https://laravel.com/docs/5.2/quickstart