Smart IoT Platform

ABY IBERKLEID, Brandeis University, USA
JAMES PETULLO, Brandeis University, USA
(JEFF) PANG LIU, Brandeis University, USA
SHUBHAM KAUSHIK, Brandeis University, USA

The Internet of Things (IoT) Platform we are developing is a scalable and flexible solution designed to capture, store, analyze, and act on data from a wide range of embedded system devices. Our platform aims to support simple and complex devices, including Temperature Sensors, Andon Lights, Edge-based Computer Vision Modules, and Smart Gloves for human motion tracking. The system follows a hub-and-spoke architecture. IoT devices communicate bidirectionally with a hub using MQTT, relaying data to a cloud-based API for real-time visualization, configuration, and automated actions through an intuitive UI.

The embedded devices we develop will serve their primary sensing or actuating functions and enable remote reconfiguration through non-volatile storage updates. The hub, built on a Raspberry Pi, will function as a remote storage device and MQTT server, facilitating efficient telemetry collection and device management. The platform's frontend will allow users to install, configure, and monitor devices while setting rules for automated responses to sensor data. An integrated AI/ML optimizer will leverage fleet telemetry data to enhance system performance by dynamically improving key operational metrics.

Our IoT platform enables seamless device integration, real-time analytics, and intelligent automation. It aims to support a broad spectrum of industrial applications and ensure reliability, scalability, and efficiency across diverse use cases.

ACM Reference Format:
Aby Iberkleid, James Petullo, (Jeff) Pang Liu, and Shubham Kaushik. 2018. Smart IoT Platform. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Project Description

We are developing a versatile IoT platform that allows users to integrate custom embedded system modules to capture, store, analyze, and act on data through an intuitive user interface. The platform enables seamless data visualization, sensor configuration, and automated actions, such as triggering an email alert when a sensor data point meets a predefined condition.

Modules for Industrial IoT. Our platform will incorporate embedded system device modules, including sensors and actuators, designed for various industrial IoT applications. We aim to develop simple and complex devices to demonstrate the platform's flexibility. Simple devices like temperature sensors and Andon lights focus on basic monitoring tasks. Meanwhile, advanced modules, such as edge computer vision systems and smart gloves for motion tracking, showcase the platform's ability to support various use cases.

System Architecture. The platform follows a hub-and-spoke, event-driven architecture with multiple microservices. A central hub connects various devices, relaying data to the cloud for further processing. Unlike single-device solutions, our approach supports multiple device modules, ensuring scalability and adaptability.

Authors' Contact Information: Aby Iberkleid, aiberkleid@brandeis.edu, Brandeis University, Waltham, MA, USA; James Petullo, jamespetullo@brandeis.edu, Brandeis University, Waltham, MA, USA, larst@affiliation.org; (Jeff) Pang Liu, pangliu@brandeis.edu, Brandeis University, Waltham, MA, USA; Shubham Kaushik, kaushiks@brandeis.edu, Brandeis University, Waltham, MA, USA.

ACM 2476-1249/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

Devices communicate with a hub within the same network, which consists of essential microservices, including an MQTT broker, a backend server, and a database. The hub then transmits data to a cloud-based API, where the frontend dashboard provides real-time insights and control.

Decive Framework. The embedded devices serve two primary functions:

(1) Sensing/Actuating: Each device performs its core function, capturing data or executing a physical action.
(2) Bidirectional Communication: Devices communicate through MQTT, sending real-time data while also receiving configuration updates. Users can remotely adjust device settings by modifying variables in non-volatile storage. For example, a device can update Wi-Fi credentials programmatically by altering its config file, such as using micro-python's *machine.nvs* to modify *wpa_supplicant.conf* on a Raspberry Pi.

2 Final Deliverables

The final delivery of our IoT platform will include several key components, ranging from simple sensor modules to advanced AI-powered systems. These components will enable seamless data collection, analysis, and automation across various industrial IoT applications.

Simple Device Modules. Our platform will support stand-alone or integrated sensor-actuator modules for basic data capture tasks. These modules will feature:

- Simple sensors (e.g., temperature and humidity sensors)
- Soldered components for reliability
- 3D-printed enclosures for durability and protection

Example Simple Device: *Temperature/Humidity model*, a sensor that measures temperature and humidity at configurable intervals. Users can adjust settings such as power consumption, MQTT Quality of Service (QoS) levels, and other sensor-specific parameters.

Complex Device Modules. The platform will support basic sensors and advanced modules integrating edge computing and machine learning capabilities. These devices will be built with:

- Complex sensors (e.g., Edge ML-powered cameras)
- Soldered components for robust connectivity
- 3D-printed cases

Example Complex Devices: (a) *Smart Glove Module*, a wearable device that recognizes hand gestures and publishes gesture data to the platform. (b) *Edge Computer Vision Module*, a camera-based module capable of analyzing visual data and transmitting relevant sensor information based on its training.

Hubs (Raspberry Pi Servers). The hub will function as a remote storage device and MQTT server, ensuring efficient bidirectional communication between devices and the cloud. Key capabilities include:

- Retrieving device status and telemetry on demand
- Sending configuration updates to devices remotely

Frontend Dashboard. The platform will include a user-friendly web dashboard for managing devices, analyzing data, and automating workflows. Users will be able to:

- Install and configure new devices
- Visualize real-time and historical sensor data
- Set up automated actions (e.g., "Send an email when a data point meets a condition")

API Layer. The API will act as the bridge between the frontend and backend databases. It will handle:

- Data retrieval and storage operations
- Communication between system components

- Secure authentication and user management

AI/ML Optimizer. The platform will integrate an AI/ML optimizer that leverages telemetry data to enhance system efficiency. It will:

- Analyze fleet-wide device performance
- Adjust system parameters dynamically to improve key performance indicators (KPIs)

3 Project Phases and Roadmap

Phase 1: Proof of Concept (Completed Successfully). This phase aimed to validate the feasibility of our IoT platform by developing a prototype module. The module generated a random number between 1 and 5, transmitted it over MQTT to the hub, and logged the data. The reconfiguration mechanism was also tested by sending a command to modify the range, generating numbers between 6 and 10 instead. Successfully completing this phase demonstrated the viability of our approach. To ensure modularity and adherence to SOLID principles, we created a structured code package to serve as a foundation for the subsequent phases.

Phase 2: Simple Device Modules. Building on the proof of concept, this phase involves implementing basic sensor modules like temperature and humidity sensors. These modules will:

- Accurately capture and report sensor data
- Log data efficiently within the framework
- Support remote reconfiguration via MQTT

Phase 3: Complex Device Modules. We will extend the framework in this phase to support more advanced devices, including multiple sensors and edge ML capabilities. The objectives are to:

- Ensure complex devices, such as a computer vision (CV) module, accurately analyze and transmit data
- Maintain the framework's core functionality—data logging and remote reconfiguration
- Integrate multiple sensors into unified, intelligent modules

Phase 4 (Optional): API Optimization and Frontend Enhancements. Once we achieve seamless data capture and reconfiguration, we will focus on optimizing the database performance and expanding frontend capabilities. This may include:

- Improving query efficiency for real-time analytics
- Enhancing the dashboard UI/UX for better device management
- Expanding automation features for more complex workflows

4 Workload Division

Each team member will be responsible for a sensor module, an actuator module, or a complex device.

Aby.

- *Simple Actuator Module:* Andon Light (Multi-LED Semaphore for Manufacturing)
- *Simple Sensor Module:* Step Counter/Traffic Volume Gauge (Flex Sensor by Spectra)
- *Frontend Development*
- *API Development*
- *Hub Implementation*

James.

- *Simple Actuator Module:* LCD Display (Displays received data/messages from config commands)
- *Simple Sensor Module:* Temperature/Humidity Sensor Module
- *AI/ML Optimizer*
- *API Development*

- *Database/Storage Management*

Jeff.

- *Complex Device:* Smart Glove Module
- *Deep Learning/Reinforcement Learning Model Training*
- *Gesture Recognition Implementation*

Shubham.

- *Complex Device:* Computer Vision Module (ESP32 Camera Module)
- *Database Expertise*

5 List of Devices

Core Computing Modules.

- Raspberry Pi 4 & 5
- ESP-32
- Raspberry Pi Pico
- Coral USB Accelerator (Provided by Ju – Features
- Edge TPU for ML inferencing)
- Edge ML Vision Module
- Edge ML Audio Module

Sensors.

- Temperature Sensor
- Humidity Sensor
- Spectra Symbol Flex Sensor 2.2" (×5) (For Smart Glove)
- MPU6050 (IMU) (×1) (For Smart Glove – Motion Tracking)
- FSR402 (Force Sensors) (×2) (For Smart Glove – Pressure Detection)

Actuators.

- LCD Screen (For displaying messages and data visualization)
- LEDs (For Andon Light and status indicators)

Supporting Equipment.

- Stable Battery for Raspberry Pi Pico (For Smart Glove)
- Battery Holders (AAA or AA, depending on design choices)
- Lithium Rechargeable Batteries (For portable and power-efficient modules)

References

[1] Instructables. *ESP32-CAM Web Server and Getting Started Guide*. Available at: http://instructables.com/ESP32-CAM-WEB-Server-and-Getting-Started-Guide/.

[2] Alex Xu. *System Design Interview – An Insider's Guide*. Independently Published, 2020.

[3] Espressif Systems. *ESP-IDF Programming Guide for ESP32*. Available at: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html.

[4] Serhat Küçükdermenci. *Sign Language Voice Converter Design Using Raspberry Pi for Impaired Individuals*. ResearchGate, 2023. Available at: https://www.researchgate.net/publication/373690087_Sign_language_voice_convertor_design_using_Raspberry_pi_for_impaired_individuals.

[5] Simone Salerno. *Smart Glove with Flex Sensors*. Edge Impulse Documentation, 2023. Available at: https://docs.edgeimpulse.com/experts/novel-sensor-projects/flex-sensors-hci.

[6] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. *Multiagent Cooperation and Competition with Deep Reinforcement Learning*. PLOS ONE, Vol. 12, No. 4, 2017. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0172395.