IoT Machine-Guard

Project Report

**ACKNOWLEDGMENTS**

**ABSTRACT**

Industry 4.0 is a revolutionary development process for the efficient management of manufacturing operations. The idea was first presented by the German Government in its High-Tech Strategy 2020 paper. Internet of Things is an integral component of industry 4.0 conceptual model and emphasizes on the interconnectivity of devices and electronic gadgets so that the devices could talk to each other and synchronized data could be utilized by all concerned users. The concept has not only gained significance in the technology sector but also gained prominence in the industrial establishments and given rise to the notion of Industrial Internet of Things (IIoT). This study analyzes Industry 4.0 development process using IIoT techniques. The study focuses on the performance issues and difficulties that may arise due to the inefficient working of sensors and ineffective techniques of fault detection. The Reference Architectural Model Industry 4.0 is analyzed, in which the three-layer model provides foundational principles for industry 4.0 concepts implementation. Techniques, solutions, and strategies are recommended to handle the issues and to ensure optimized performance and smooth functioning of industry 4.0 processes using IIoT techniques.

**Table of Contents**

[**ACKNOWLEDGMENTS** iii](#_Toc502084988)

[**ABSTRACT** iv](#_Toc502084989)

[**Table of Contents** v](#_Toc502084990)

[**List of Figures** vii](#_Toc502084991)

[**CHAPTER 1: INTRODUCTION** 1](#_Toc502084992)

[1.1. Introduction to The Chapter 2](#_Toc502084993)

[1.2. Project Background 2](#_Toc502084994)

[1.3. Problem Statement 3](#_Toc502084995)

[1.4. Project Objectives 4](#_Toc502084996)

1.5. Planning for Project 4

[1.6. Project Main Focus 4](#_Toc502084997)

[1.7.](#_Toc502084998) Tasks that can the "IoT Machine-Guard" device do ………………...……………….5

[**CHAPTER 2: LITERATURE REVIEW** 7](#_Toc502085000)

[2.1. Introduction to The Chapter 8](#_Toc502085001)

[2.2. RAMI 4.0 (Reference Architectural Model Industry 4.0) 8](#_Toc502085002)

[2.3. Sensors 11](#_Toc502085003)

[2.4. Fault Detection 13](#_Toc502085004)

[2.5. Performance Analysis 14](#_Toc502085005)

[**CHAPTER 3: Building the IoT Machine-Guard Device “The Practical Part”…….**19](#_Toc502085006)

[3.1. Introduction to the chapter 20](#_Toc502085007)

3.2. Planning for the hardware design ………………………………………………………………………………………… 20

3.2.1. The block diagram of the hardware design…………………………………………………………………………20

3.2.2. The IoT Machine-Guard input/output data Flowchart ………………………………………………………..21

3.2.3. The IoT Machine-Guard Code Flowchart ……………………………………………………………….…………...22

3.2.4. The IoT Machine-Guard Circuit Schematics and Wiring Diagram…………………….…………..……..23

3.2.5. The IoT Machine-Guard Bill of Materials ………………………………………………………..…………………..24

Chapter 4 "Programming" …………………………………………………………………………………………………………….26

4.1. Introduction to the Chapter …………………………………………………………………………..………………………27

4.2. Explaining Vital Code Parts …………………………………………………………………………………………………….27

Chapter 5 "Experimenting" …………………………………………………………………………………………………………..34

5.1. Introduction to the Chapter …………………………………………………………………………………………………..35

5.2. Experiment (1) ……………………………………………………………………………………………………………………….35

5.3. Experiment (2) ……………………………………………………………………………………………………………………….40

5.4. Experiment (3) ……………………………………………………………………………………………………………………….43

chapter 6 " Research Results and Recommendations" ………………………………………………………………….50

6.1. "Introduction to the chapter"…………………………………………………………………………………………………51

6.2. Research findings and Recommendations ……………………………………………………………………………..51

References ……………………………………………………………………………………………………………………………………52

**List of Figures**

Figure 2.1: Smart Technology Infrastructure ………………………………………..…………8

[Figure 2.2: RAMI 4.0 Architectural Model 9](#_Toc502085009)

[Figure 2.3: The Working of a Wireless Sensor Network (WSN) 12](#_Toc502085010)

[Figure 2.4: Algorithm for Distributed Maintenance Planning 14](#_Toc502085011)

[Figure 2.5: Incident Response and Performance Analysis in an IIoT Environment 16](#_Toc502085012)

[Figure 2.6: Performance Improvement by Combining Process and Equipment Data 17](#_Toc502085013)

Figure 3.1. The IoT Machine-Guard Block Diagram …………………………………...……20

Figure 3.2. The IoT Machine-Guard Flowchart ………………………………………....……21

Figure 3.3. The IoT Machine-Guard Code Flowchart…………………………………...……22

Figure 3.4. The IoT Machine-Guard Circuit Schematics and Wiring Diagram………...……23

Table 3.1. The IoT Machine-Guard Bill of Materials ………………………………………..24

Figure 4.1. Code Libraries Setup ………………………………………..………….…...……27

Figure 4.2. Machine Normal Output Values Setup ………………………………….......……28

Figure 4.3. Analog and Digital Feeds Setup………. …………………………………....……28

Figure 4.4. Setup Connection to Adafruit.io web broker………………………………...……29

Figure 4.5. Keep Connected to Adafruit.io web broker & Power Control Setup...……...……30

Figure 4.6. Send Densors' Data to Adafruit.io web broker ….……………..…………...….…31

Figure 4.7. Print Current Power State of machine(ON State) and Show Sensors' Data …...…31

Figure 4.8. Print Current Power State of machine (OFF State) and Show the Reason...……32

Figure 4.9. Handle Power Messages from Adafruit.io…... ….………………………...……33

Figure 5.1. IoT Machine-Guard Serial Monitor Data Output (A.C "On" State) ……………..35

Figure 5.2. IoT Machine-Guard Serial Monitor Data Output (A.C "OFF" State) ……….…..36

Figure 5.3. IoT Machine-Guard Serial Monitor Data Output (Smoke Sensor Stimulated, Machine OFF State) …………………………………………………………………….…..37

Figure 5.4. IoT Machine-Guard Adafruit.io Data GUI Output (trial 1.1) …………………..38

Figure 5.5. IoT Machine-Guard Adafruit.io Data GUI Output (trial 1.2)…………….……..39

Figure 5.6. Online Data Output of a working CNC Machine (trial 1)……………….……..40

Figure 5.7. Online Data Output of a working CNC Machine (trial 2)………………..……..41

Figure 5.8. Online Live Data Output of a working CNC (trial 1)………………...….……..42

Figure 5.9. Online Emergency Triggers……………………………………..……….……..42

Figure 5.10. Online Data Output of a Working A.C 1 Hour Duration……….……….……..43

Figure 5.11. Online Data Output of a Working A.C 2.5 Hours Duration……….……..……..44

Figure 5.12. Online Vibration Data Output of a Working A.C 2 Hours Duration .……..…..45

Figure 5.13. Online Current Data Output of a Working A.C 2 Hours Duration .………..…..45

Figure 5.14. Online Temperature Data Output of a Working A.C 2 Hours Duration …..…..46

Figure 5.15. Online Smoke Data Output of a Working A.C 2 Hours Duration ………….…..46

**CHAPTER 1:**

**INTRODUCTION**

* 1. **Introduction to The Chapter**

The term industry 4.0 represents the advanced trend of data exchange and automation of processes, particularly in the context of manufacturing technologies. The main areas of focus under this head include the Internet of Things (IoT), cyber-physical systems, cognitive computing, and cloud computing. Baena, Guarin, Mora, Sauza, and Retat (2017) assert that the concept of industry 4.0 emerged from the initiatives of industry professionals, academics, and the German Government. The primary focus of this concept is to enhance the competitiveness of the manufacturing sector. The concept emphasizes on accomplishing this objective by the integration of Information and Communication Technologies (ICT) in the industrial production. Zezulka, Marcon, Vesely, and Sajdl (2016) argue that the industry 4.0 will bring interconnectivity into three key factors. First, it will bring digitization of simple to complex economic networks. Second, the digitization will be extended to the offers of products and services. Third, it will create new possibilities in the context of marketing models. The most promising technology that will drive all these factors is the Internet of Things (IoT). The concept is now further extended to Services and People. In an industry 4.0 setup, communication entities will be in a position to communicate with one another through IoT and utilize the synchronized data through the systems’ life cycle. This study conducts the performance analysis in an industrial 4.0 environment through the use of Industrial Internet of Things (IIoT) techniques.

* 1. **Project Background**

The pace of technology change has been increasing with the passage of time. The software applications and newly-built systems become obsolete within 5 to 10 years. The new trend that has further added to this pace is the underlying operating system technology. It is also updated by vendors on a periodic basis. All these advancements have led the IT professionals to focus on IIoT and the issues of cybersecurity. The clients will not embrace the IIoT technology if it means the complete replacement of the existing systems. The software developers and system designers will require unleashing the power of the already-developed systems. It has given rise to the notion of control systems that is the central theme of IIoT. The technology is heavily dependent on process data for maintenance, operation, and optimization. The technology obtains this data from the control systems.

IIoT is a promising platform, and it will enable industrial engineers to host their applications at a centralized level. The cloud computing environment in this context will bring increased uptime and greater efficiencies. IIoT will converge the people, data, and smart machines. It is expected to bring transformations in industrial efficiency, productivity, and operations through the use of rich datasets. A key benefit of IIoT is to enable the machines to talk to each other reliably and promptly. The communication systems in an industry 4.0 environment enable the services, entities, and resources to be connected and share and utilize data among all stakeholders (Zezulka, Marcon, Vesely, & Sajdl, 2016). However, it will require building the capability of collecting data from sensors. The collection of data is impacted by a variety of factors such as the performance of the sensors, humidity level, and the temperature. Hence, there is a need for performance analysis in an industrial 4.0 environment as how IIoT can be implemented effectively in real time, and what corrective and preventive measures should be adopted for its smooth operations.

* 1. **Problem Statement**

The cloud computing environment and web-based applications have enabled real-time access of data through the desktop computers, notebooks, tablets, smartphones, and other electronic gadgets. An employee, a customer, or a supplier can always connect to the corporate network of an industrial organization. IIoT will bring new features in this interconnected world where the devices will also be able to talk to each other and can also be controlled remotely.

However, the industrial field is, still, suffering many problems such as fires, total loss of machines, lack of enough experienced machine-guards and investigators, and lack of data measurement and analysis to estimate machine life and quality.

The project tries to investigate the efficiency of a newly-made “IoT Machine-Guard” device.

The research will focus on the extent those new devices can add to the advancement of the whole industrial system, in general, and industrial safety specifically, around the globe.

* 1. **Project Objectives**

Based on the project background and the problem statement, this project aims to present a thorough investigation of a newly-made “IIoT Machine-Guard” device, planning to the project, conducting research, tasks distribution amongst team researchers, selecting suitable parts for the project, wiring, coding, and testing procedures.

* 1. **Planning for Project**

Based on the project background and the problem statement, I tried to find a new device that has the ability to read machine-connected sensors’ data outputs at once, and to have communication abilities too.

A device that can read data outputs from the sensors, set triggers if any sensor data exceeds the normal level of any machine-default output levels, and to upload theses data to a web broker.

A web broker is a web-based IoT platform that combines the use of both MQTT communication protocol for data transfer, and cloud computing for live GUI data display and data analysis as well.

In my search for a good broker, we liked the Adafruit broker the most. It costs around 15$/month with an infinite number of feeds, gauges and data storage, which in turn stimulated me to start conducting and applying the practical part of our research.

* 1. **Project main focus**

During my pilot study, I noticed an increasing lack of a good dependable device that can guard most industrial machines; track and analyze machinery data outputs, protect machines from damage or even fires caused by malfunction or obsolete machines in a factory.

For all mentioned reasons, and for the great advancement in the field of IIoT, I thought of building a monitoring and protection device and named it “IoT Machine-Guard.” That can be attached to many electrical machines for both monitoring data outputs, and for safety and quality issues.

* 1. **Tasks that can the “IoT Machine-Guard” device do**

The “IoT Machine-Guard” can perform the following tasks:

1. It has a connected Temperature Sensor that can measure temperature of values of 0-1024 Degrees Celsius, display the temperature of machine, detect any overheating, trigger an alarm, and cut off current from machine automatically for protection.
2. It has a connected Vibration Sensor that can display the vibration level of machine, detect any extra vibration (caused by malfunction, broken engine parts, or unstable current source), trigger an alarm, and cut off current from machine automatically for protection.
3. It has a connected Sound Sensor that can display the sound level of machine, detect any extra strange sound frequencies (caused by malfunction, broken engine parts, or unstable current source), trigger an alarm, and cut off current from machine automatically for protection.
4. It has a connected Smoke Sensor that can display the smoke level of machine, detect any extra smoke levels around machine’s high-current hot spots and motors (caused by malfunction, broken engine parts, or extra current withdrawal), trigger an alarm, and cut off current from machine automatically for protection.
5. It has a connected Current Sensor that can display the current withdrawal and Watt consumption levels of machine, detect any over-current levels drawn by machine (due to malfunction, broken engine parts, or extra current withdrawal), trigger an alarm, and cut off current from machine automatically for protection.
6. It has a connected Power Relay that functions as an on/off switch to control the power of machine.
7. The device can compare all sensors’ data inputs to predefined normal level for each sensor and control the power of machine accordingly.
8. The device can also send these data outputs from the sensors to [Adafruit.io](http://adafruit.io/) web broker in a ratio of 1 data reading/4 seconds.
9. The state of the interactive online power switch changes according to machine power status, and it can also be used to power on/off a machine.

\* The [Adafruit.io](http://adafruit.io/) web broker works as a GUI that includes gauges, line charts, toggle switches, and other useful data display tools.

It also stores all data readings of all feeds in a browsable & downloadable database. A dedicated Excel sheet can be downloaded for each feed containing feed ID, value, date, time, and location.

Those database registers can be used for data analysis that, in turn, enables factory managers/owners to perform quality inspection, machine history tracking, machine health investigation, and predict the best time for machine elimination and disposal.

1. Research conductor has created a GUI dashboard to display the data outputs in a form of Gauges and Line Charts on Adafruit.io web broker.
2. Research conductor has also recorded a special trigger for each data feed on the web broker, which in turn will send an e-mail in case any data feed exceeds the pre-defined normal value of each sensor.

**CHAPTER 2:**

**LITERATURE REVIEW**

## 2.1. Introduction to The Chapter

The chapter presents a critical review of the academic literature related to different aspects of industry 4.0 using IIoT techniques. The project reveals the work done so far in this domain and identifies the gap in the extant literature. This identification of gap develops the rationale and conceptual framework for the current study. Big data, working of sensors, performance analysis, and fault detection are critical themes of the project.

## 2.2. RAMI 4.0 (Reference Architectural Model Industry 4.0)

The Internet of Things and Services operate in an environment where devices talk to each other and create an interconnected technological ecosystem. It creates a highly sophisticated and smart technology infrastructure as shown in 2.1 below:

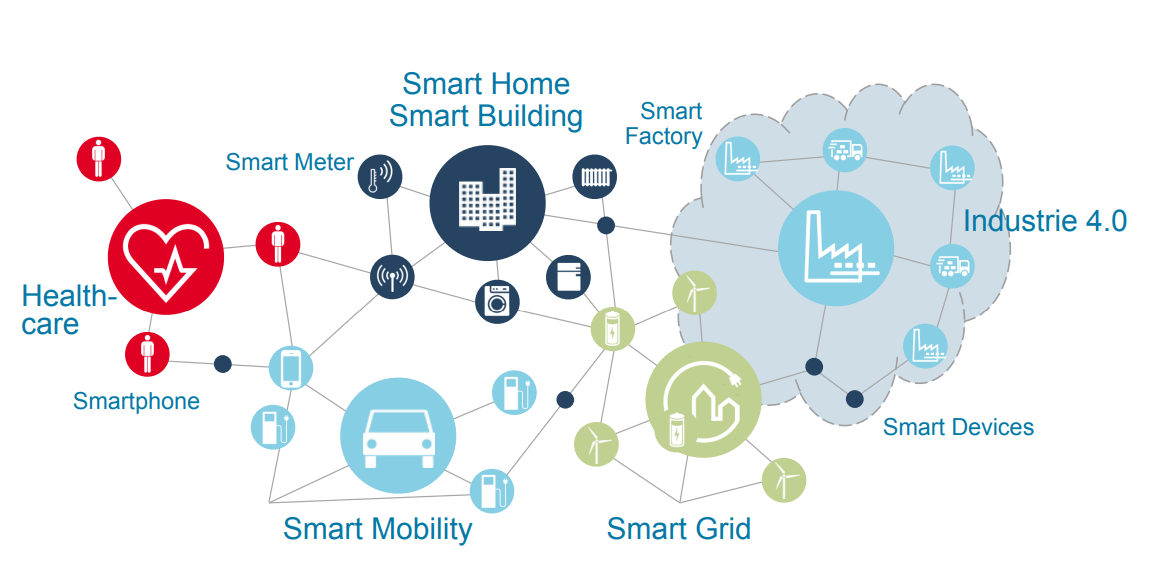


Figure 2.1: Smart Technology Infrastructure

**Source: (Schweichhart, 2017)**

The development of a smart technology infrastructure requires defining precise communication structures. It necessitates developing a common language with its unique grammar, vocabulary, culture, and semantics. An ideal solution to meet these prerequisites is RAMI 4.0. It is an architectural model that is based on a three-dimensional map. The map highlights some approaches that should be adopted in an industry 4.0 environment. A key emphasis in RAMI 4.0 architecture is the assurance that all the participants and stakeholders will understand each other. It is a mandatory requirement in the IIoT context where the devices need to talk to each other. The conceptual model of RAMI 4.0 is shown in Figure 2.2 below:

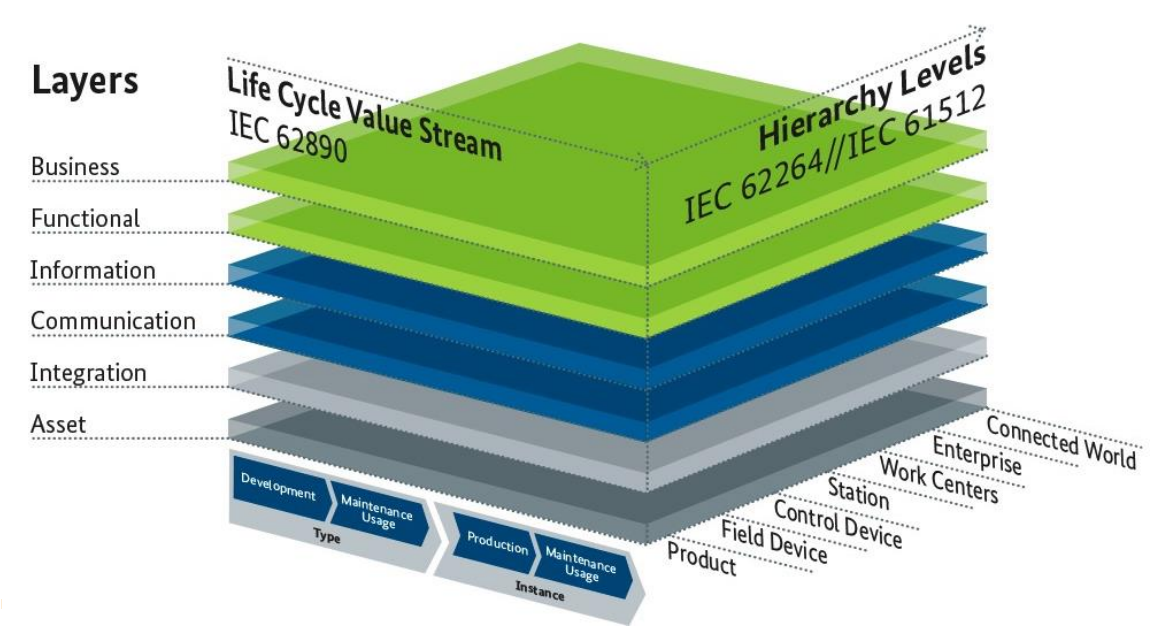


Figure 2.2: RAMI 4.0 Architectural Model

**Source: (Schweichhart, 2017)**

Industry 4.0 is used for three mutually interconnected factors:

1. Digitization and integration of any simple technical – economical relation to complex technical – economical complex networks
2. Digitization of products and services offer
3. New market models

In addition to the above 3 factors, and for purposes of such a complex production – market networks the leading institution and firms in Germany – the leading country of the Industry 4.0 activities and ideas - developed and published the RAMI 4.0 (Reference Architecture Model Industry 4.0) and the Industry 4.0 Component models in the last year.

RAMI 4.0 represent different look from different aspects, vertical axis that shows layers in Figure 2 shows the look from different aspects (market aspect, a look from a perspective of functions, information communication). The other axis (right in the horizontal level) describes function position of the components in the Industry 4.0. In this axis, there is specified the functionality of the components, no any specification for implementation but the function assignment only.

The asset layer represents physical components, for example ideas, archives, and metal parts. The integration layer makes provision of information on the asset (HW/SW, Components) in a form which is available for computer processing. It contains elements related to information technology such as (RFID, Sensors, Actuators), also connecting persons is a part of this layer, persons are connected via (Human Machine interconnection) – HMI. The information layer provides description of rules and events, generates run time for processing of events, and ensures data integrity. The functional layer provides formal description of functions and creates platform of horizontal integration of various functions. The business layer ensures integrity of functions, contains legal and regulatory framework conditions, and enabling modeling of rules which system has to follow.

The function of layers is also depicted in horizontal axis in Figure 2 in the left-hand side horizontal axis. The axis is divided into 2 parts, Type & Instance, a type is a representation of a product initial idea. It includes the submission of new orders, development of parts, testing them and finally the prototype. After a prototype has been accomplished and all tests succeeded the type is prepared for serial production, now here where the instant part takes place, each one of the produced parts of the (Type) is called an Instance of that Type e.g. has a unique serial number.

In Industry 4.0 Component Model, the application’s developer goal is to help producers and system integrators to create HW-SW components for industry 4.0. A lot has been discussed about the communication systems, like internet of things and internet of services & people, and this model describes it all. It represents in more details what is industry 4.0 and its sub-connection systems in addition it explains and lists industry 4.0 factors and specifications of how to transform a regular component in any industry to an industry 4.0 component. Any component that needs to be an industry 4.0 component must be a part of electronic container (Shell) of secured data during all life cycle, and the data must be available to all entities of technical production chain.

Components or objects (component of a machine, machine, SW, etc.) in a non-industry 4.0 component, is an unknown object or individually known, anonymous entity. Its only objective is to function by itself regardless of what another objects condition is. So, in order for this object to become more socialized and have interaction with other object (industry 4.0 concept) this object must be a part of something called an administration shell (data container), and this administration shell consists of various objects (things), and all of these things are connected to the (data container ), and administration shells are all connected, now all (things) in different administration shells are inter connected and can send and receive data from another administration shell, Hence, internet of things.

There are several key requirements of an industry 4.0 model. The network of an Industry 4.0 environment must be designed in a way that the communication between any 2 endpoints is possible. It must be possible to define the concept of an industry 4.0 component in such a way that it can meet requirements with different areas. It must be possible that an information of an Industry 4.0 component (thing) can be kept in the object itself or in a higher-level IT system.

## 2.3. Sensors

IIoT is regarded as an innovative and interactive technology. The technology provides interconnectivity to a set of devices (for instance smartphones) and a set of sensors (Hossain & Muhammad, 2016). When this interconnectivity is established, the system enables recording, collecting, sharing, and transmitting data for business intelligence and data analytics.

A sensor (node) is the basic element of a Wireless Sensor Network (WSN), and the network in this context works as follows. A request is sent by the user to the node of the gateway. The gateway sends a message to the sensor node. The sensor node sends the processed data back to the gateway node. Eventually, the gateway node sends a message to the user who can then access, view, and analyze the data. It is shown in Figure 2.3 below:

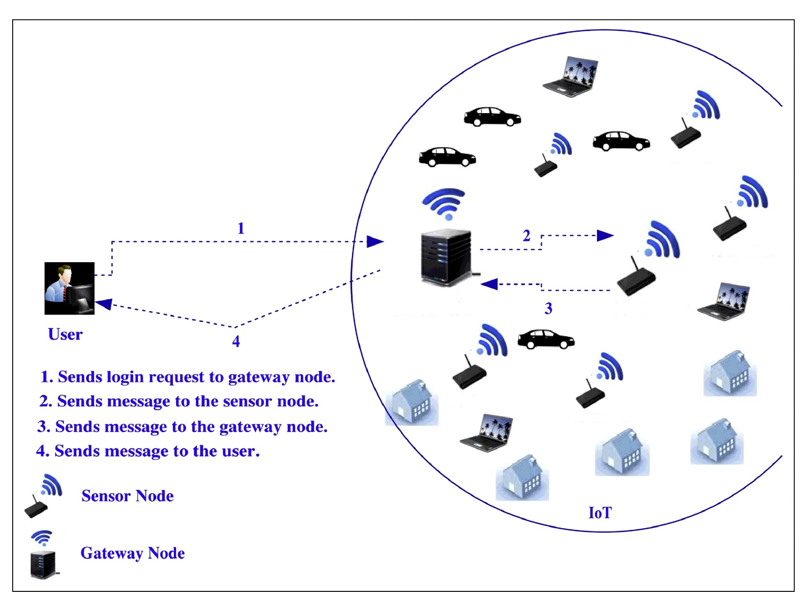


Figure 2.3: The Working of a Wireless Sensor Network (WSN)

A sensor is a device that interacts with its surroundings, it receives information (input) of what it was designed to detect, for instance there are sensors that can detect temperature, vibrations, light, sound, etc. And the (output) is usually a signal that is transferred to a human readable context either.

One of the interesting sensors, is motion sensors that detects if any movement is occurring around it, the sensors receives its input by first producing or sending a certain type of energy, such as microwaves, ultrasonic waves or light beams and detect when the flow of energy is interrupted by something entering its path, when this happens the sensor make an input out of it and the produces the corresponding output.

Multiple kinds of networks are there, one of which is wireless sensor network WSN, it’s a network of an environment where sensors are connected wirelessly and can communicate with each other. A sensor (node) is the basic element of a WSN, one of the problems of the WSN or any network is power saving, so the strategy followed in WSN that leads to power saving is duty-cycle strategy where a node can periodically switch between sleep and active modes. If a node is collecting data, it is assigned to a time slot (the part where it is awake and collecting), after the collecting is finished and only that time it can go back to sleep to save power, a node or sensor also must be awake if its parent has been assigned to a time slot (awake time) to help it.

## 2.4. Fault Detection

Upasani, Bakshi, Pandhare, and Lad (2017) argue that the integration of computing infrastructure with sensors is gaining increased prominence on the industry shop-floors. However, the authors emphasize that maintenance planning is a critical activity of decision making that can minimize unplanned downtime. The projects present a distributed algorithm that can detect the possibility of faults in the system in the context of job-shop manufacturing (Figure 4). This project highlights that this algorithm is a preferable approach over Particle Swarm Optimization and Memetic Algorithm. It is because these algorithms are based on centralized heuristics, and the distributed algorithm adopts a decentralized approach.

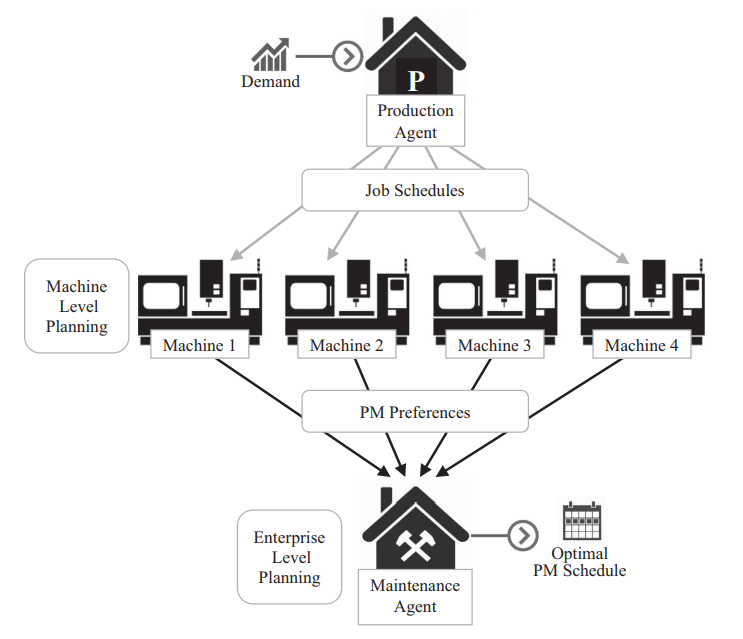


Figure 2.4: Algorithm for Distributed Maintenance Planning

## 2.5. Performance Analysis

The manufacturing sector has entered a new era with the concept of Industry 4.0. The new era is driven and governed by digital revolution. The new age is considered as the fourth major transformation of the manufacturing sector. The key trends that led to the concept of industry 4.0 include the lean revolution that happened in the decade of 70s, the increased trend of outsourcing in the decade of 90s, and the boom of automation and computerization that began in the 2000s. Silicon Partners (2017) present a hypothetical model of a smart factory to illustrate performance analysis in an IIoT – Industry 4.0 environment. In the model, there is a chemical plant that is being used for mass scale production of chemicals. In the reserves of the chemical plant, there are hazardous as well as highly reactive chemicals. The smart factory is built such that the chemical plant’s components are all connected by actuators and sensors. These components include exhausts as well as pipelines. The environment of Integrated Connections assists the factory in controlling the chemical levels and maintaining the stocks. All this is accomplished without much human intervention.

On one day, there is a sudden incident in which a pipeline of the chemical plant had an unexpected leak. Subsequently, a hazardous gas began to spread in the air. Since the pipelines of the chemical plant were connected with advanced sensors, the supervisor monitoring the chemical plant immediately became aware of the issue. It was possible because the advanced sensors were providing real time data transfer and updates to the supervisor. The data was also supplied and transferred to the actuators. The actuators performed the required action based on the information and commands received in the data transfer. The required action in this example translated to the chemical plant releasing a gas to dissolve the harmful effects of the hazardous gas. The actuators also performed the action of sealing the vault of the pipeline so that there is no further emission of hazardous gas from the pipeline. All these actions brought the situation back to normalcy. The performance system of the smart factory was not limited to the corrective actions. The data from the incident of the chemical plant was also analyzed critically by the senior management of the organization. It was analyzed for the causes of the pipeline leak, improving the precautionary measures, and devising effective solutions for addressing these issues. The example indicates that how performance analysis can be carried out in an industry 4.0 IIoT environment in which the devices can talk to each other and they are connected by sensors and actuators. The example demonstrates that IIoT reduced the likelihood of human error in the smart factory and provided quick and relevant response to the incident as shown in Figure 2.5.

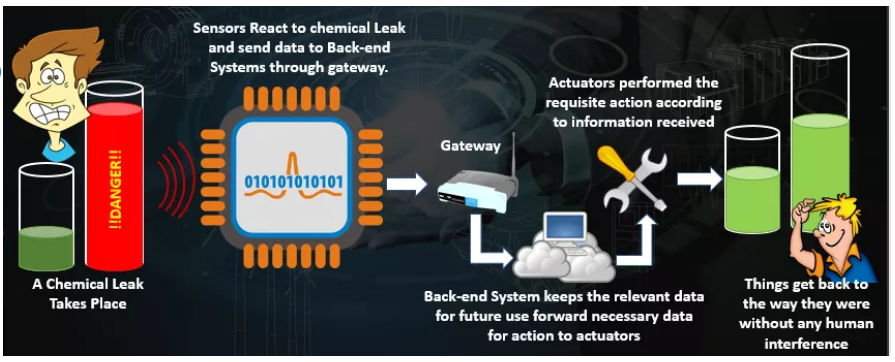


Figure 2.5: Incident Response and Performance Analysis in an IIoT Environment

**Source: (Silicon Partners, 2017)**

Rio (2017) argues that the new revolution in the manufacturing sector is propelled by the intersection of IIoT and enterprise asset management. The author asserts that there are certain factors that have increased the significance of adopting IIoT. The first factor is predictive maintenance so that unplanned downtime could be reduced and hence reducing the up time of the equipment in the manufacturing plant. In the performance management in IIoT, the predictive maintenance program anticipates failures and takes corrective actions. In the traditional approach, the data used to be pulled from the control systems and the data owners just had the processed data. However, in an IIoT environment, the data is not pulled from the control systems. Instead, it is pulled from the sensors directly installed on the equipment. It provides an opportunity to the administrators to have equipment-related data besides the process data. The combination of the equipment related data and the process data enables the supervisors to make innovations and improvements in the domain of asset performance management. A typical workflow combining the process data and equipment data in an IIoT environment in shown in Figure 2.6 below:

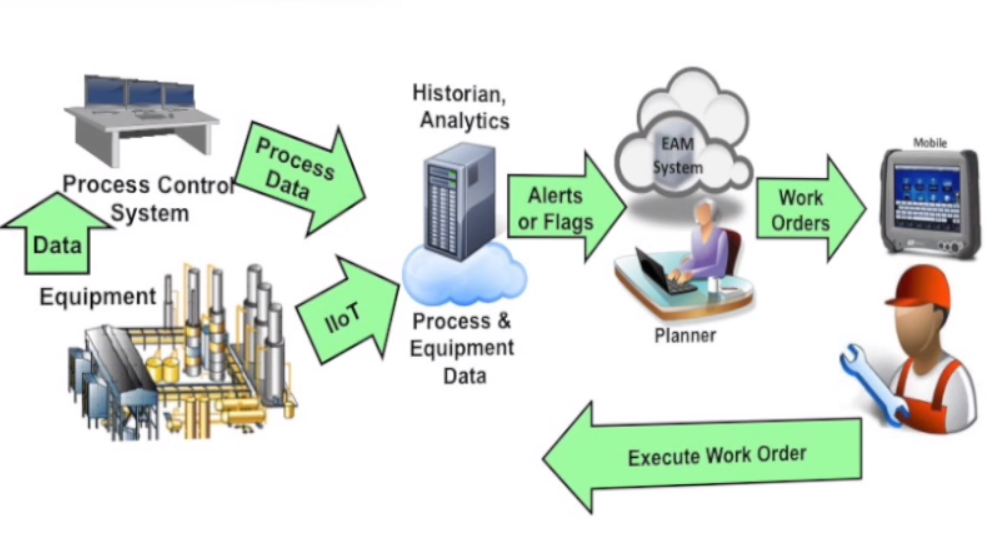


Figure 2.6: Performance Improvement by Combining Process and Equipment Data

**Source: (Rio, 2017)**

Another area of focus is the end users. The maintenance department and the technicians in a conventional model perform their tasks in separate silos. Hence, it usually takes a long time to communicate issues. The equipment issues are communicated personally or through emails that represents an ad-hoc mode of communication. In an industry 4.0 paradigm, the recommended approach is to send alerts to the integrated system when there is an unusual occurrence in the connected network. The flags and alerts become visible throughout the integrated system. An automated work order may also be generated for the plant to be performed by the actuators (Ehret & Wirtz, 2017). It can bring significant improvements in the performance and output of reliability engineers. It is because the predictive maintenance feature will forecast the exact date of an equipment failure that resolves the issues of ad-hoc communication. The system also assists the maintenance technicians in the task accomplishment. In the conventional system, a paper work order is given to a maintenance technician. The technician performs the task, writes in own handwriting on the work done, and the work order is then submitted to a data entry person. Due to a significant manual work involved in this process flow, the trust and confidence on the system is reduced.

In a highly mobile environment of IIoT, the data collection is an integral part of the work process of the maintenance technician. It enhances the accuracy of the whole system. People are more willing to use the system as a tool of scheduling and planning. Hence, the industry 4.0 setup improves the overall reliability of the system (Wollschlaeger, Sauter, & Jasperneite, 2017). IIoT highly recommends automated management of business processes. The flags and alerts are generated in the system on a regular basis. It ensures the prompt response and immediate solution of the issues. In the manufacturing sector, the production unit has its own set of quality indicators, and the maintenance unit has its own set of metrics. In the conventional setup, the two units work as separate business units. The communication is rare or occasional between the two units. There is no synchronization of the communication. In an IIoT setup, the combination of process data and equipment related data allows for new aspects of process optimization and brings effectiveness in the asset performance management.

**CHAPTER 3:**

**(Building the IoT Machine-Guard Device)**

***“The Practical Part”***

## Introduction to The Chapter

In this chapter are the practical procedures conducted by the team for building the *“IoT Machine-Guard”* device.

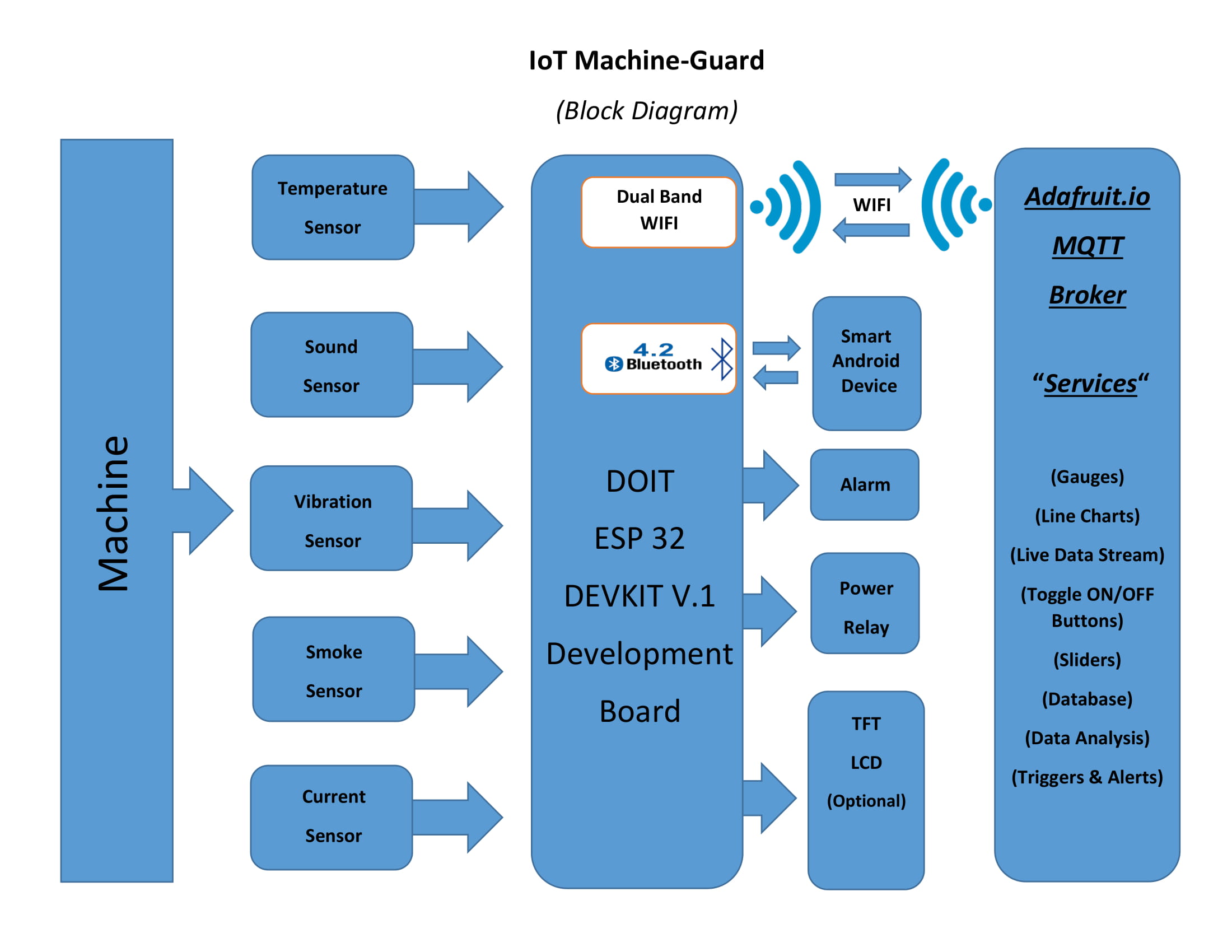
It lists the hardware block diagram, input/output data run ways, and the programing code input/output flowchart.

## Planning for the hardware design:

The following diagrams list all the steps done for planning the hardware build.

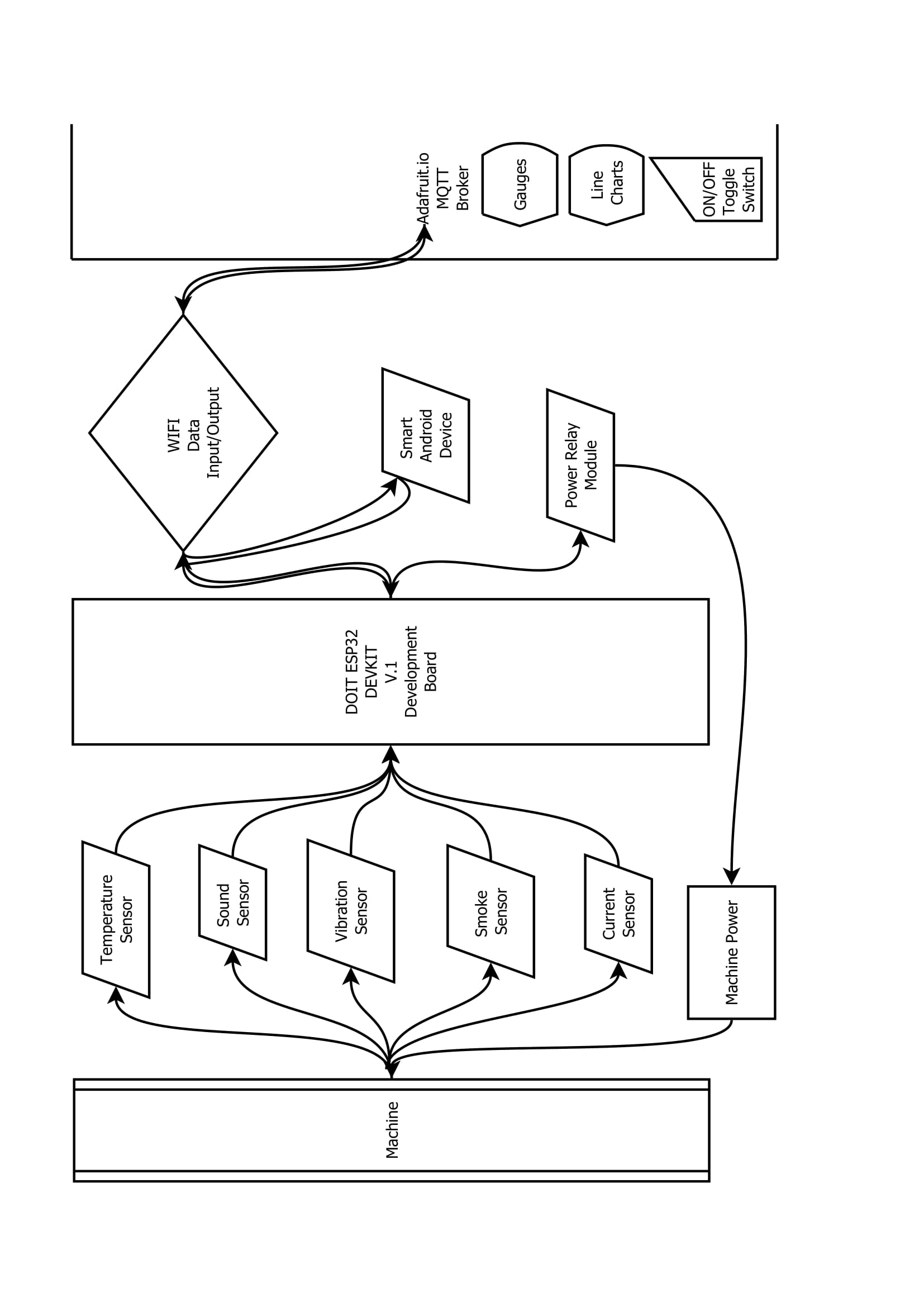
* + 1. In Figure (3.1) is the block diagram of the hardware design.

Figure 3.1: The IoT Machine-Guard Block Diagram



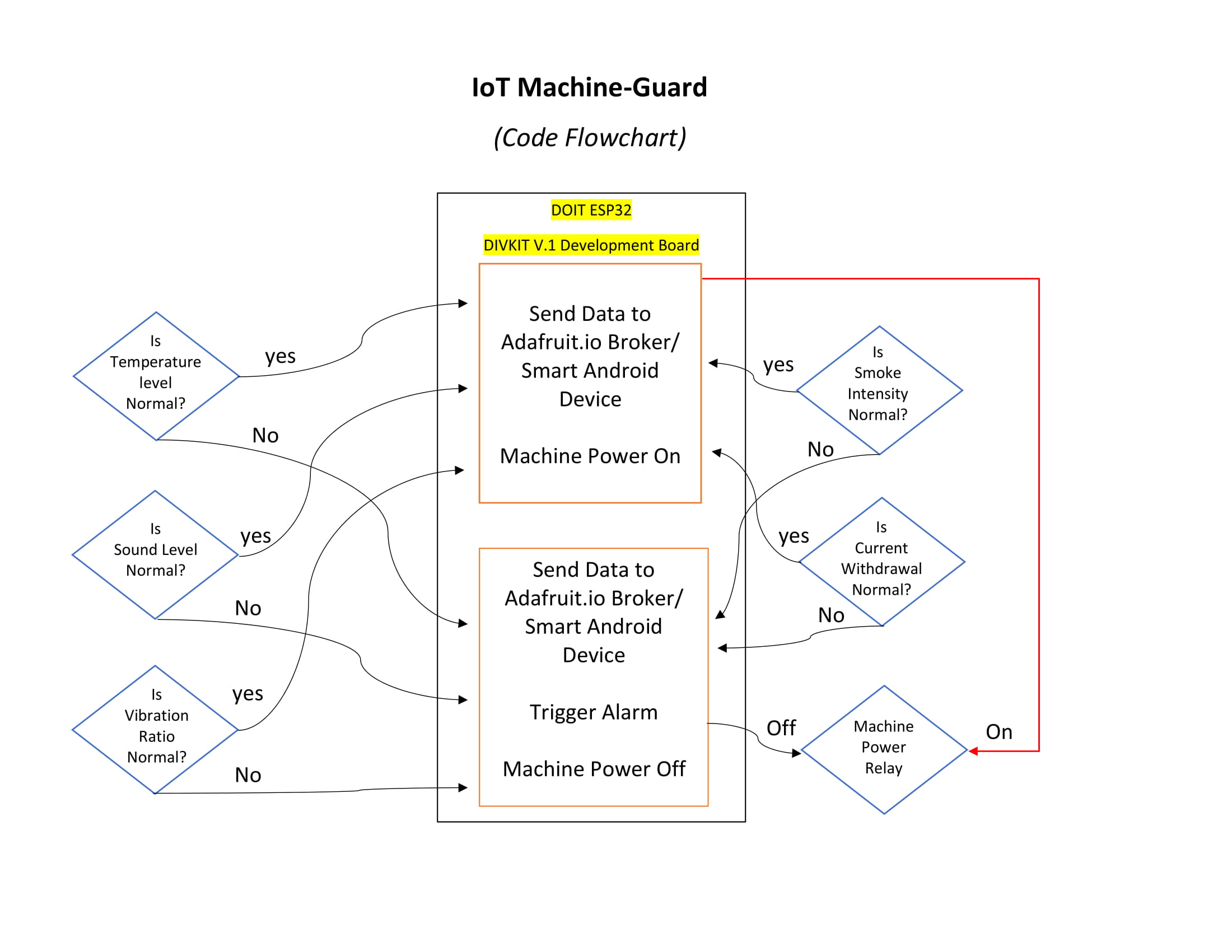
* + 1. In Figure (3.2) is the *“IoT Machine-Guard”* input/output data Flowchart.

Figure 3.2: The IoT Machine-Guard Flowchart



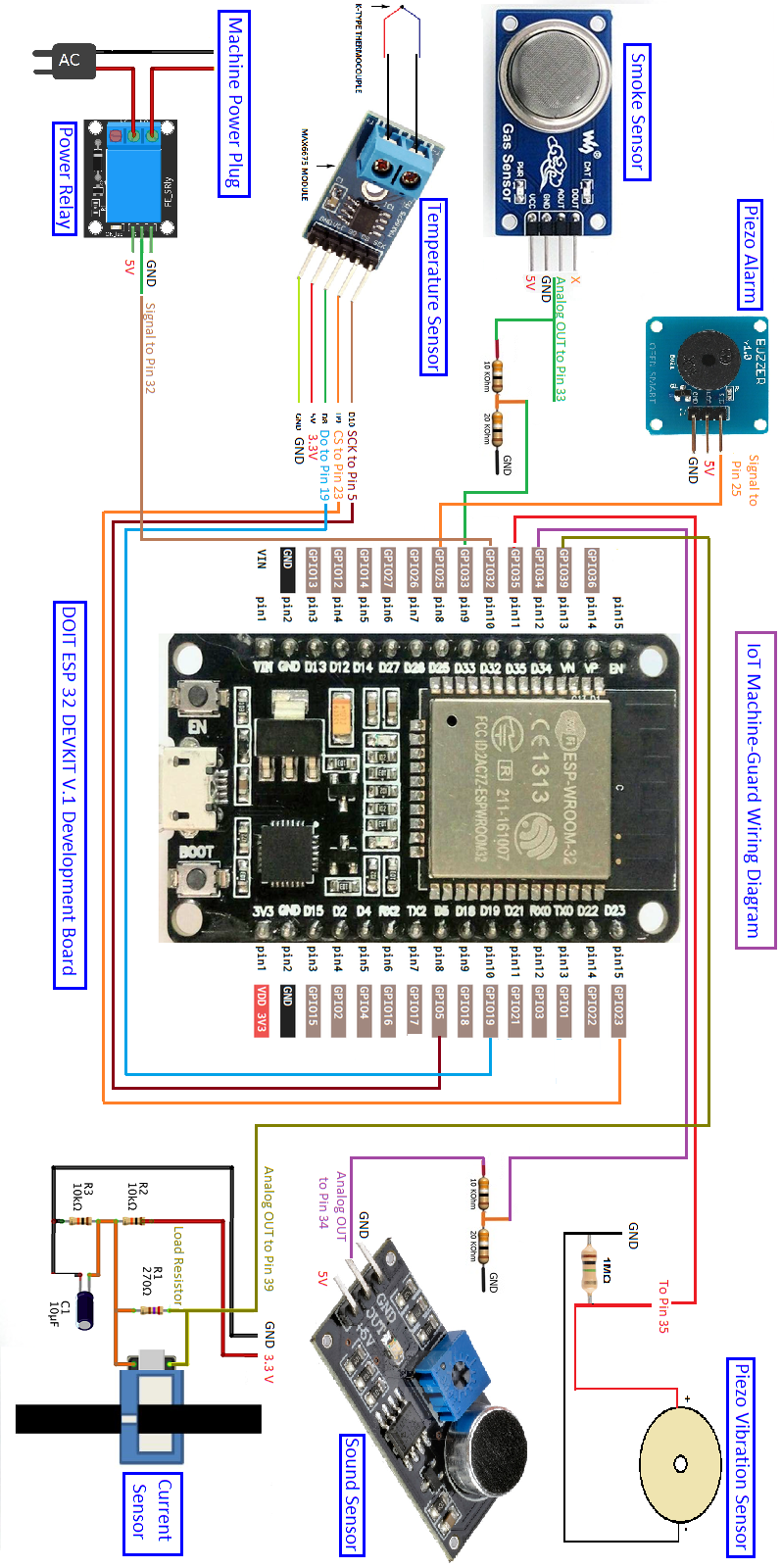
* + 1. In Figure (3.3) is the *“IoT Machine-Guard”* Code Flowchart.

Figure 3.3: IoT Machine-Guard Code Flowchart



* + 1. In Figure (3.4) is the *“IoT Machine-Guard”* Circuit Schematics and Wiring Diagram.

**Figure 3.4: IoT Machine-Guard Circuit Schematics and Wiring Diagram**



* + 1. In table (3.1) is the *“IoT Machine-Guard”* Bill of Materials.

**Table 3.1: IoT Machine-Guard Bill of Materials**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Item | Quantity | Specifications |
| 1 | DOIT ESP32 DEVKIT V.1 Development Board | 1 | * **Wireless connectivity**   + **WiFi:** 150.0 Mbps data rate with HT40   + **Bluetooth:** BLE (Bluetooth Low Energy) and legacy Bluetooth * **Processor:** Tensilica Xtensa Dual-Core 32-bit LX6 microprocessor, running at 160 or 240 MHz * **ROM:** 448 KB * **SRAM:** 520 KB * **Low Power:** ensures that you can still use ADC conversions, for example, during deep sleep. * **Peripheral Input/Output:** peripheral interface with DMA that includes capacitive touch, ADCs (Analog-to-Digital Converter), DACs (Digital-to-Analog Converter), I²C (Inter-Integrated Circuit), UART (Universal Asynchronous receiver/Transmitter), SPI (Serial Peripheral Interface), I²S (Integrated Interchip Sound), RMII (Reduced-Media-Independent Interface) and PWM (Pulse-Width Modulation). * **Arduino IDE compatible** |
| 2 | Thermocouple Temperature  Sensor with MAX IC | 1 | SPI serial port outputs temperature value - Measurement range: 0 ~ +1024'C - Resolution: 12 bits - On-chip cold junction compensation - High-impedance differential input - Thermocouple thread breakage detection - Single +5V power voltage - Low power consumption - Working temperature: -20~+85'C - 2000V ESD protection |
| 3 | Analog Mic Module | 1 | Microphone sensor module for Arduino with Analog output. |
| 4 | Piezo Vibration Sensor | 1 | Piezo vibration disk with analog output |
| 5 | Smoke & Gas Sensor MQ2 | 1 | Sensitive for Methane, Butane, LPG, smoke. This sensor is sensitive for flammable and combustible gasses. The heater uses 5V. |
| 6 | Non-Invasive Current  Sensor SCT013 | 1 | AC Current Metering 0-100A |
| 7 | Relay Module for Arduino | 1 | 240V 10 Amp On/Off switching 3.3/5V Operating voltage. |
| 8 | Microphone male jack | 7 | For connecting sensors |
| 9 | Microphone Female connector | 7 | For connecting sensors |
| 10 | Piezo Active Buzzer | 1 | Works as an alarm |
| 11 | 9X15 Test board | 1 | For soldering components |
| 12 | 1 MΩ Resistor | 1 | For Piezo Vibration sensor |
| 13 | 10 KΩ Resistors | 2 | For Current Sensor circuit |
| 14 | 15 Ω Resistor | 1 | As a load Resistor for Current Sensor Circuit |
| 15 | 10 μF 16V Capacitor | 1 | For Current Sensor Circuit |

**CHAPTER 4:**

**(Programming)**

## Introduction to The Chapter

In this chapter are the practical procedures conducted by the team for programming the *“IoT Machine-Guard”* device.

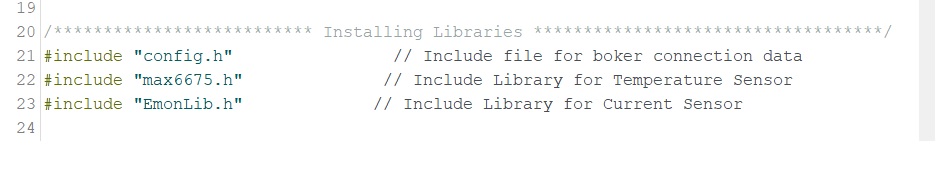
It lists the vital code blocks with a brief explanation for each block accompanied with screen shots of the Arduino IDE program for each code block.

## Explaining Vital Code Parts:

Note: The ESP 32 development board has been programmed using the Arduino IDE environment since it is easier and more familiar for the team members to work with.

1. Referring to figure (4.1) at the very beginning of the code two libraries have been included adding to a configuration file as follows:

**Figure 4.1: Code libraries setup**



1.A. The “*config.h”* file holds the WIFI setup commands; (WIFI SSID, WIFI Password), together with the Adafruit.io broker IO\_USERNAME and IO\_KEY that are needed to connect to Adafruit.io broker GUI.

Each user can edit this connection information to be able to connect to his own WIFI network and to his Adafruit.io private account.

1.B. The “max6675.h” is the specific library file of the Temperature Sensor.

1.C. The “EmonLib.h” is the specific library file of the Current Sensor.

1. Referring to figure (4.2) and at the beginning of the code following to the libraries setup section, a section holding the normal machine-type-related ratios for sound, vibration, smoke, temperature, and current.

\*Depending on the type of machine, those values have to be altered for both the triggers and machine auto-off functions to run properly.

**Figure 4.2: Machine Normal Output Values setup**



1. Referring to figure (4.3) before the Void Setup () function, five analog feeds have to be setup for the ESP 32-Web broker interaction to occur. In this part, each sensor is connected to a special feed holding the same name of the function it measures.

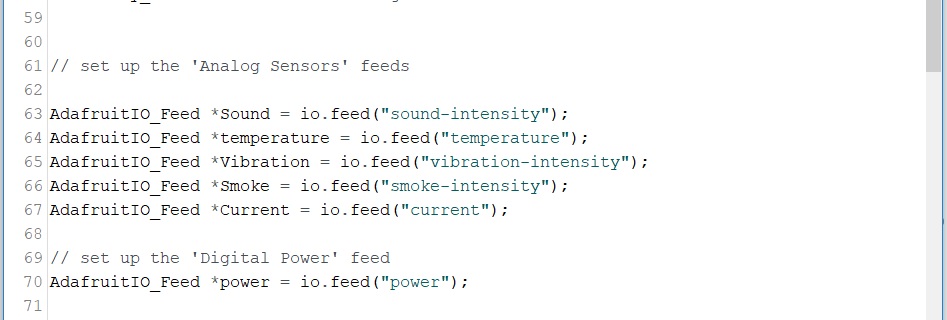
Those feeds will be used by the function (Feed save-> io.feed name) that will be included in the Void Loop () function for storing incoming data of each sensor and sending the new values to the same feed at Adafruit.io GUI.

An interactive, bidirectional, Digital “power” feed has been setup, too. This feed is responsible for the power of machine;

It can turn on or turn off the machine wirelessly, anywhere in the world, in case of emergency, and it displays the current power state of machine at the same time!

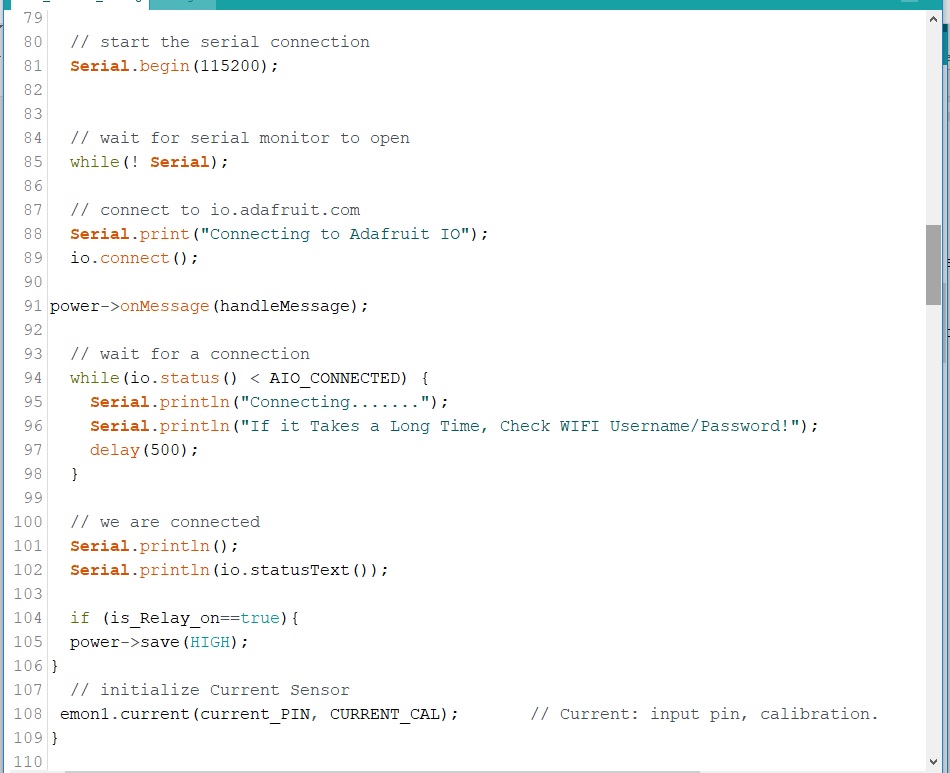
When machine is turned on, and after it connects to Adafruit.io successfully, this online toggle switch will toggle to the ON/1 position automatically, and if any sensor output-data exceeds the predefined level, this toggle switch will automatically toggle to the OFF or 0 state automatically.

**Figure 4.3: Analog & Digital Feeds Setup**



1. Referring to figure (4.4) this part of code is responsible for (A) establishing the connection to Adafruit.io web broker and to start data transfer. It is also responsible for (B) changing the power toggle switch state to ON/1 when machine connects successfully to the broker, and (C) initializes the current sensor.

**Figure 4.4: Setup Connection to Adafruit.io web broker**



1. Referring to figure (4.5) this part of code is responsible for (A) keeping the system connected to Adafruit.io web broker whenever the WIFI connection is available, and (B) controls the power of machine by testing the current state of function (is Relay on).

**Figure 4.5: Keep Connected to Adafruit.io web broker & Power Control Setup**

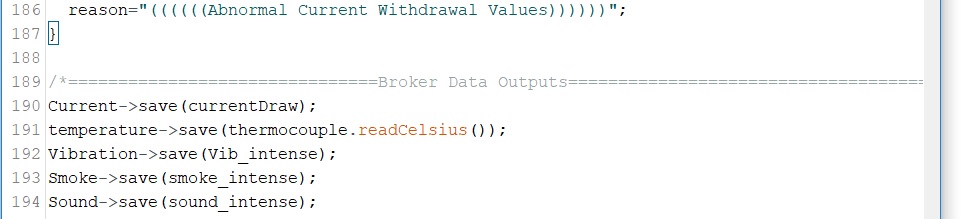


**CHAPTER 4:**

**(Research Results)**

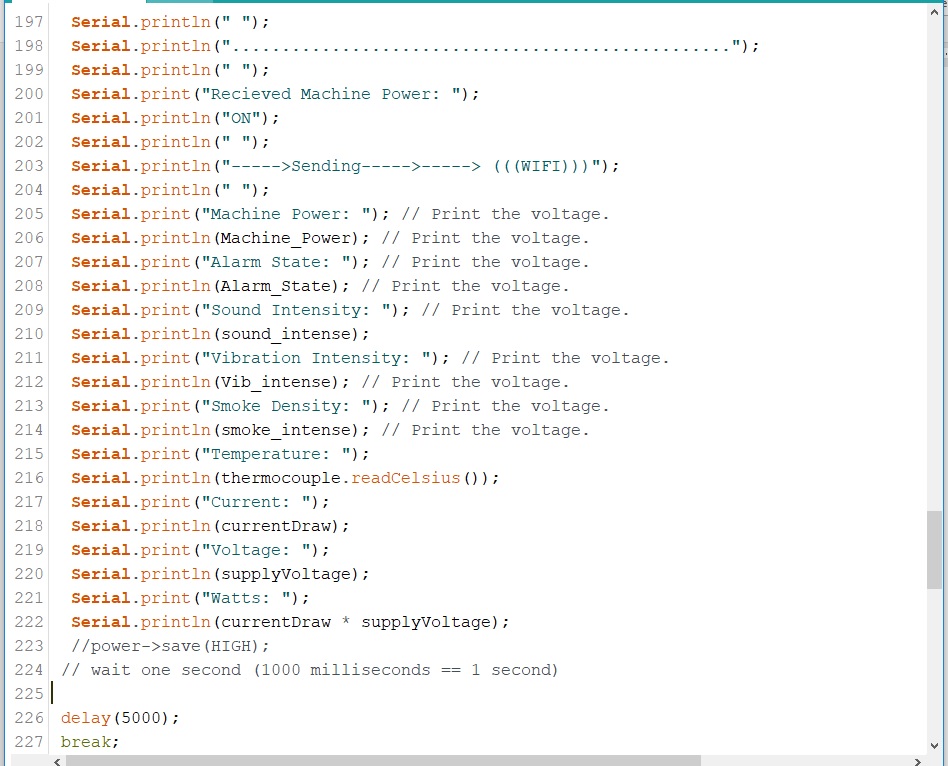
1. Referring to figure (4.6) this part of code is responsible for sending sensors’ data output to Adafruit.io web broker to be displayed in a GUI containing gauges and line charts, and to be saved in a data base specific to each sensor.

**Figure 4.6: Send Sensors’ data outputs to Adafruit.io web broker**



1. Referring to figure (4.7) this part of code is responsible for tasks to be done when machine’s power is turned ON. It (A) prints the Power State of machine to the Serial Monitor, and (B) displays the data outputs read by all sensors. It also (c) sets a delay of (5000) the period of time needed for uploading the data outputs to Adafruit.io web broker.

**Figure 4.7: Print Current Power State of machine (ON State) and Show Sensors’ data**



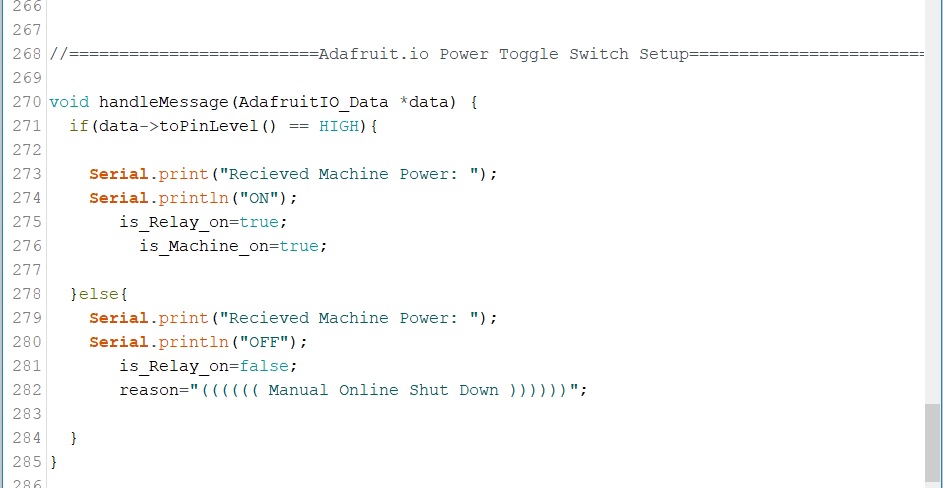
1. Referring to figure (4.8) this part of code is responsible for tasks to be done when machine’s power is turned OFF. It (A) prints the Power State of machine to the Serial Monitor, and (B) Shows the reason for the shutdown (whether the machine was turned OFF manually through the Adafruit.io broker’s power toggle switch, or due to a sensor that has triggered an alarm and turned the machine OFF for exceeding a pre-defined output level). It also (C) changes the state of the power toggle switch on Adafruit.io broker to OFF/0 state automatically.

**Figure 4.8: Print Current Power State of machine (OFF State) and Show the Reason**



1. Referring to figure (4.9) and at the end of the code, this part of code is responsible for receiving power messages coming from Adafruit.io power toggle switch, those commands are used for switching the power of machine ON/OFF.

**Figure 4.9: Handle Power Messages from Adafruit.io**



**CHAPTER 5:**

**(Experimenting)**

## 5.1. Introduction to The Chapter

In this chapter are the research results recorded after building the *“IoT Machine-Guard”* device. It lists the different experiments and tests conducted using the *“IoT Machine-Guard”* device.

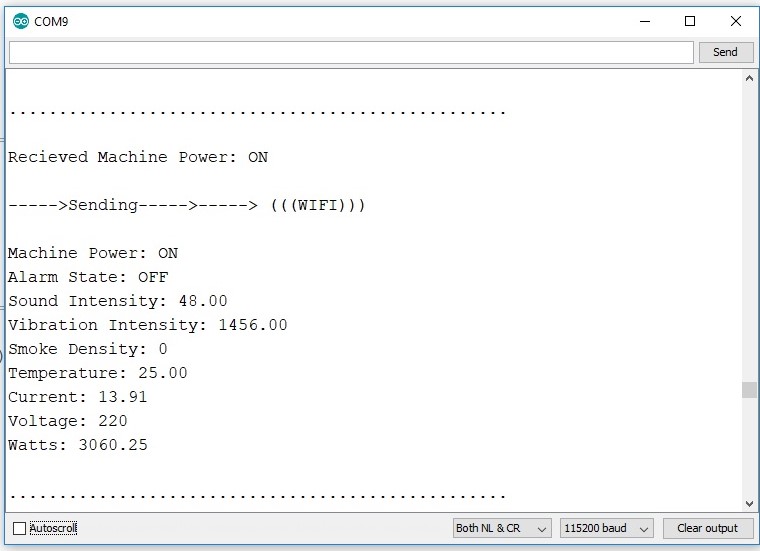
## 5.2. (Experiment 1)

\* The first experiment was applied on a window air conditioner.

A- In figure (5.1) are the serial data outputs of a working window air conditioner:

**Figure 5.1: *IoT Machine-Guard* Serial Monitor Data Output**

**(A.C “ON” State)**

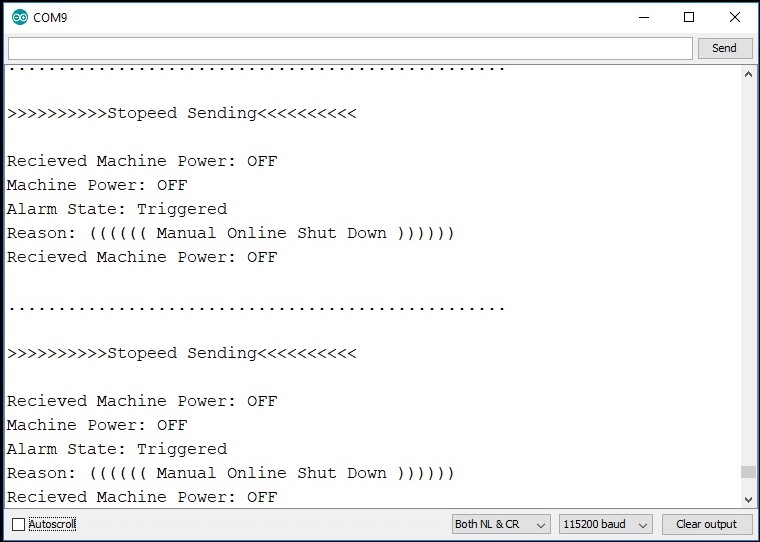


B- In figure (5.2) are the serial data outputs of a non-working window air conditioner after online shut down:

It shows the reason for shut down “Manual Online Shut Down”

**Figure 5.2: IoT Machine-Guard Serial Monitor Data Output**

**(Online A.C “OFF” State)**

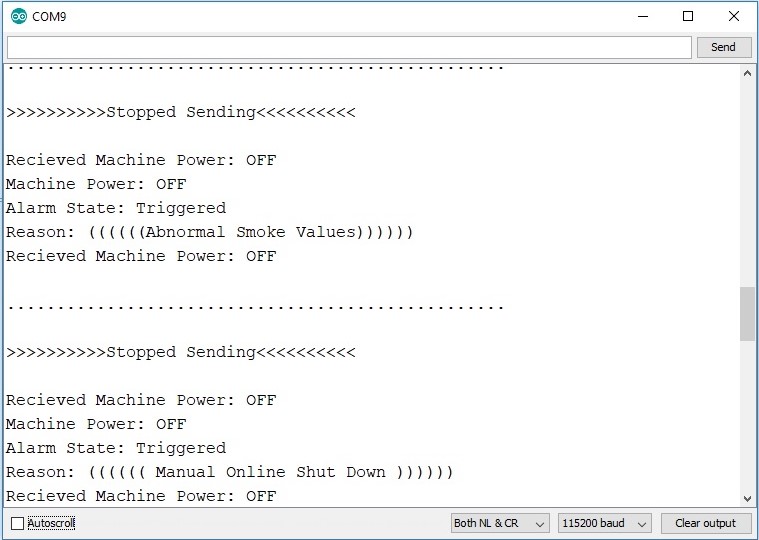


C- In figure (5.3) are the serial data outputs of a non-working window air conditioner after stimulating the Smoke Sensor:

It shows the reason for shut down “Abnormal Smoke Values”. This data output way is also applicable for all other sensors; i.e. each sensor displays it’s own “Reason” message if the incoming data exceeds its pre-defined level.

**Figure 5.3: IoT Machine-Guard Serial Monitor Data Output**

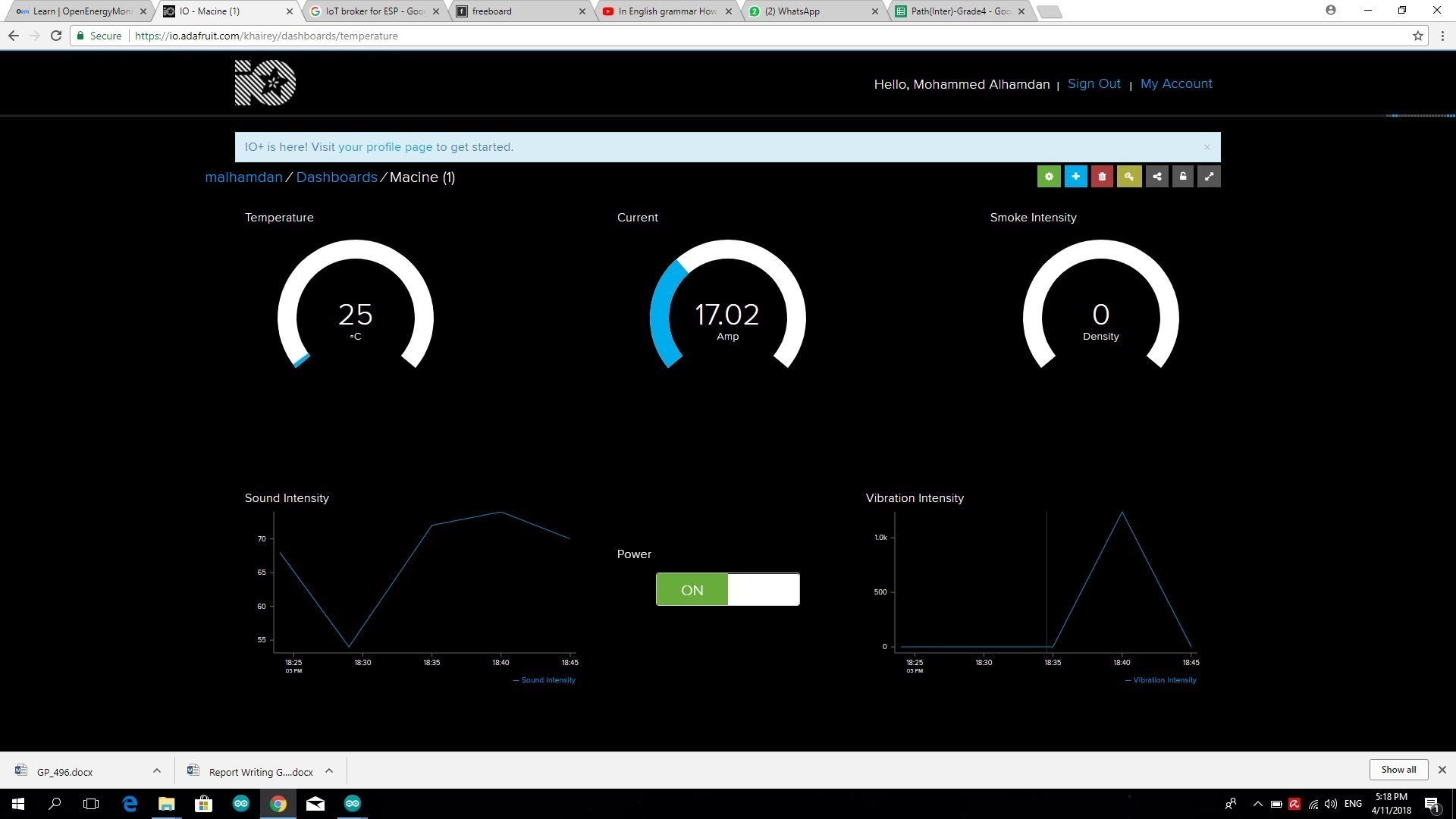
**(Smoke Sensor Stimulated, Machine OFF State)**



D- In figure (5.4) are the online data outputs of a working window air conditioner before stimulating the Smoke Sensor:

A GUI of three gauges and two line-charts are used for data display.

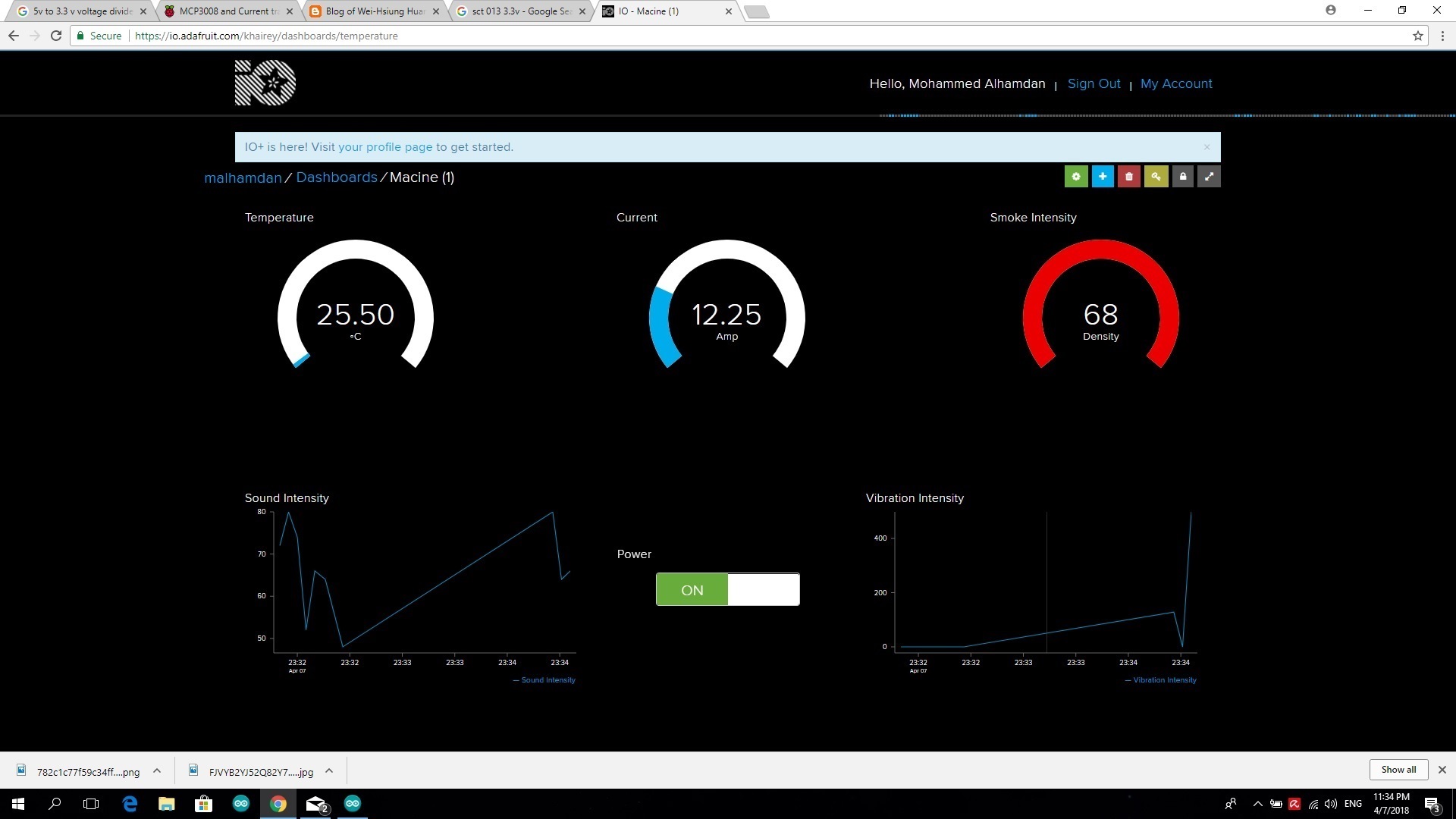
**Figure 5.4: IoT Machine-Guard Adafruit.io Data GUI Output (trial 1.1)**



E- In figure (5.5) are the online data outputs of a working window air conditioner after stimulating the Smoke Sensor:

A GUI of three gauges and two line-charts are used for data display.

**Figure 5.5: IoT Machine-Guard Adafruit.io Data GUI Output (trial 1.2)**



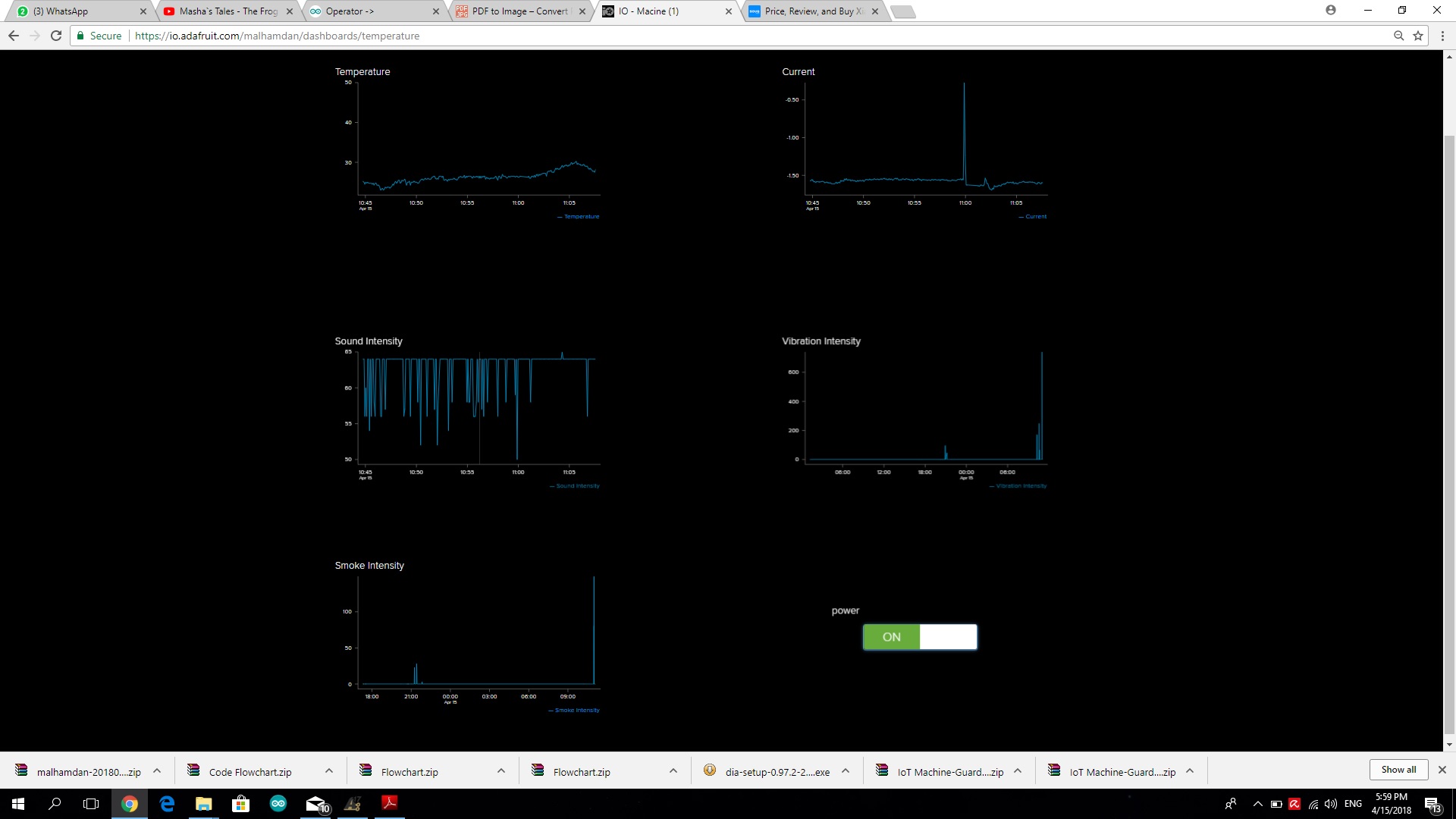
## 5.3. (Experiment 2)

\* The second experiment was applied on a CNC machine as follows:

1. In figure (5.6) are the online data outputs of a working CNC machine:

* A GUI of five line-charts are used for data display.
* Current sensor was not connected for machine wire inadequacy.
* Smoke Sensor stimulated.

**Figure 5.6: Online Data Output of a Working CNC Machine (trial 1)**



1. In figure (5.7) are the online data outputs of a working CNC machine:

* A GUI of five line-charts are used for data display.
* Four Data feeds are displayed.
* Current sensor was connected and stimulated.
* Temperature Sensor is stimulated.

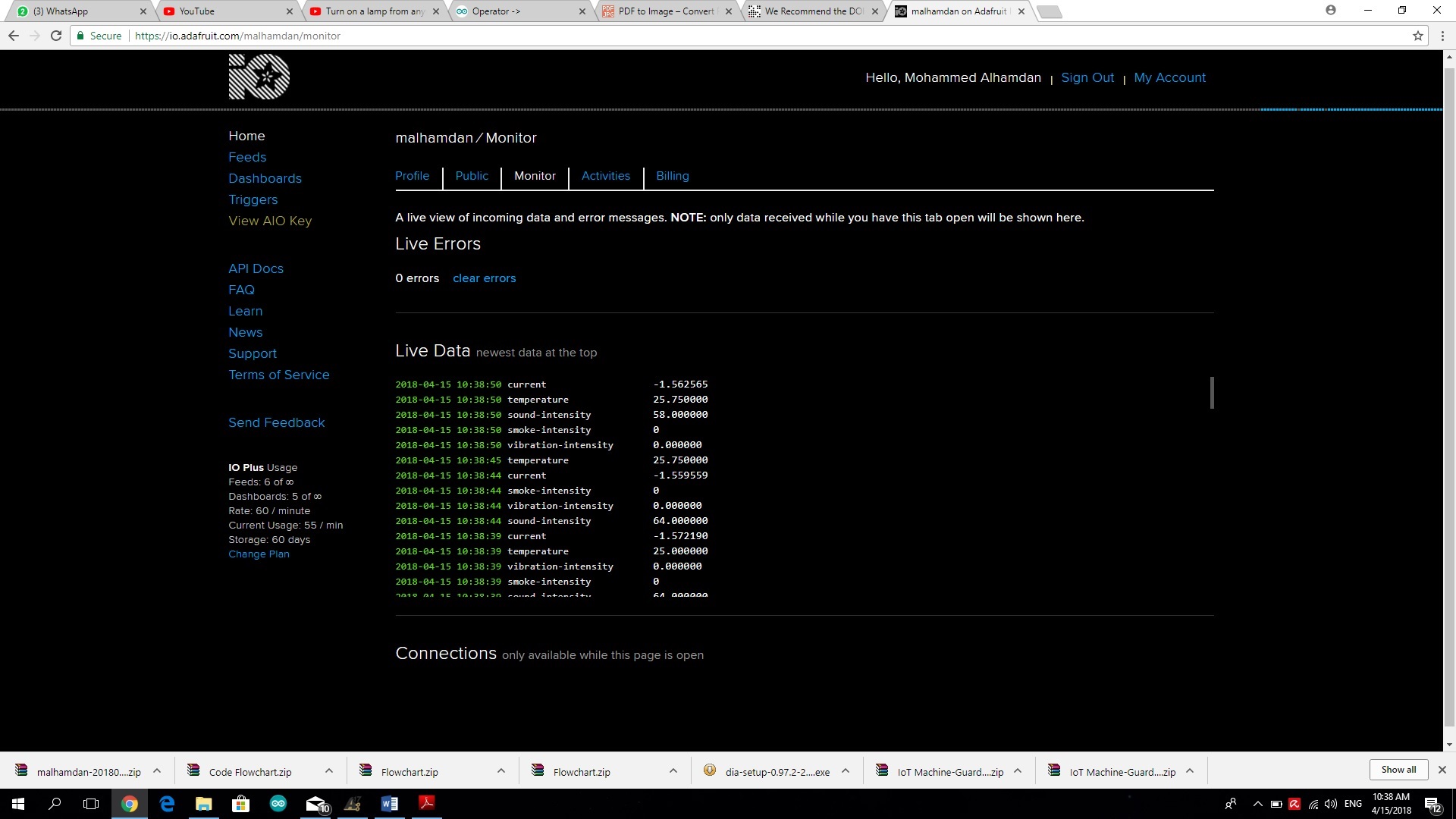
**Figure 5.7: Online Data Output of a Working CNC (trial 2)**



1. In figure (5.8) are the online live data outputs of a working CNC machine:

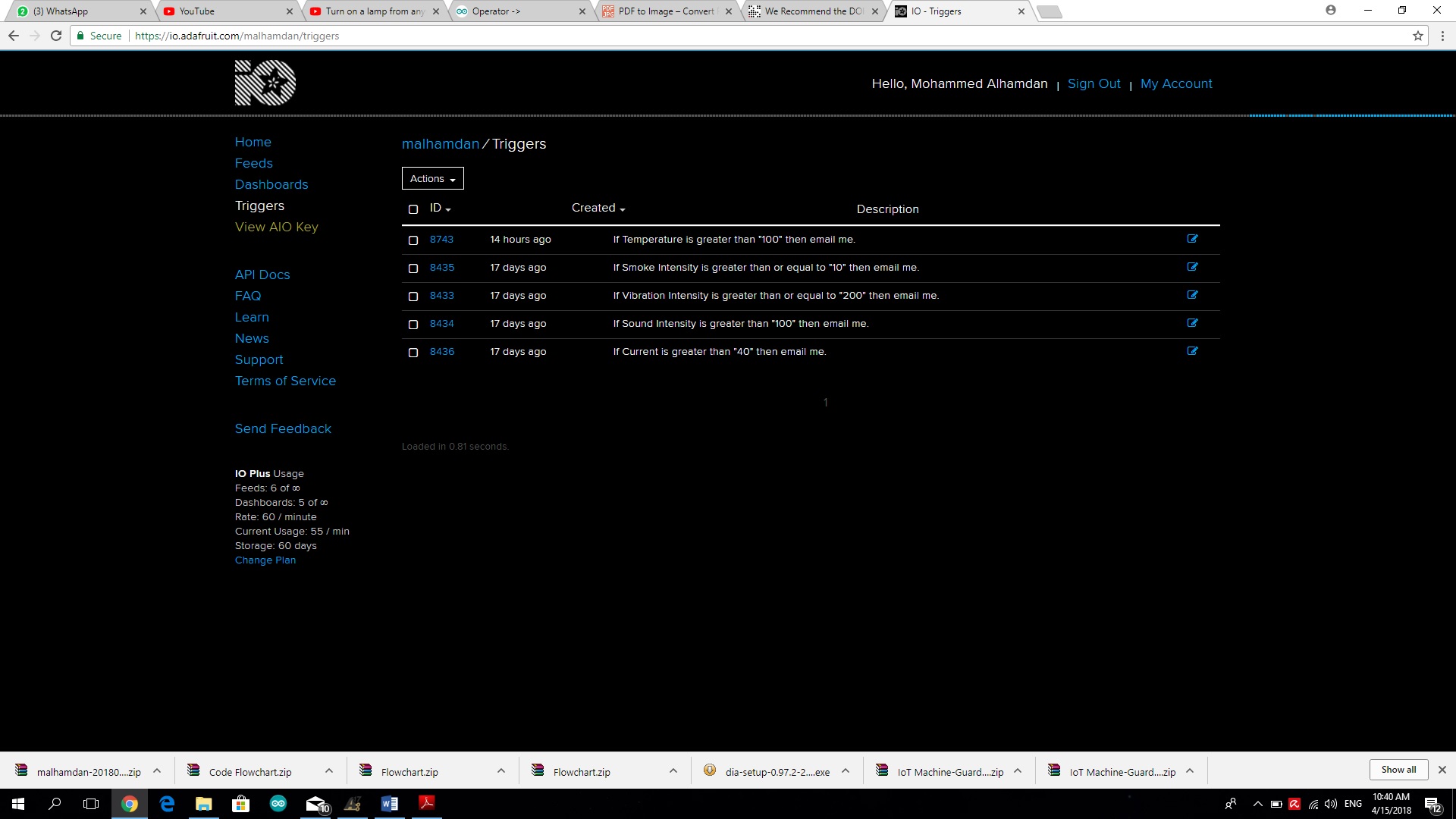
* Current sensor was not connected.

**Figure 5.8: Online Live Data Output of a Working CNC (trial 1)**



1. In figure (5.9) are the online triggers set to send emails in case any sensor data output exceeds a pre-defined level:

**Figure 5.9: Online Emergency Triggers**



## 5.4. (Experiment 3)

\* The third experiment was applied-again- on a window air conditioner at home:

1. In figure (5.10) are the online data outputs of a working air conditioner for a duration of 1 hour:

* A GUI of five line-charts are used for data display. They display the data output of the air conditioner from 18.45 PM, to 19.45 PM.
* Current sensor was connected. It shows Compressor-on / compressor-off periods clearly.
* Sound Sensor was not connected due to a website issue; when all five feeds are connected, the power feed doesn’t work properly. So, we had to eliminate the sound data temporarily.
* Smoke Sensor stimulated at 19.00 PM.

**Figure 5.10: Online Data Output of a Working A.C 1 Hour Duration**



1. In figure (5.11) are the online data outputs of a working air conditioner for a duration of 2.5 hours:

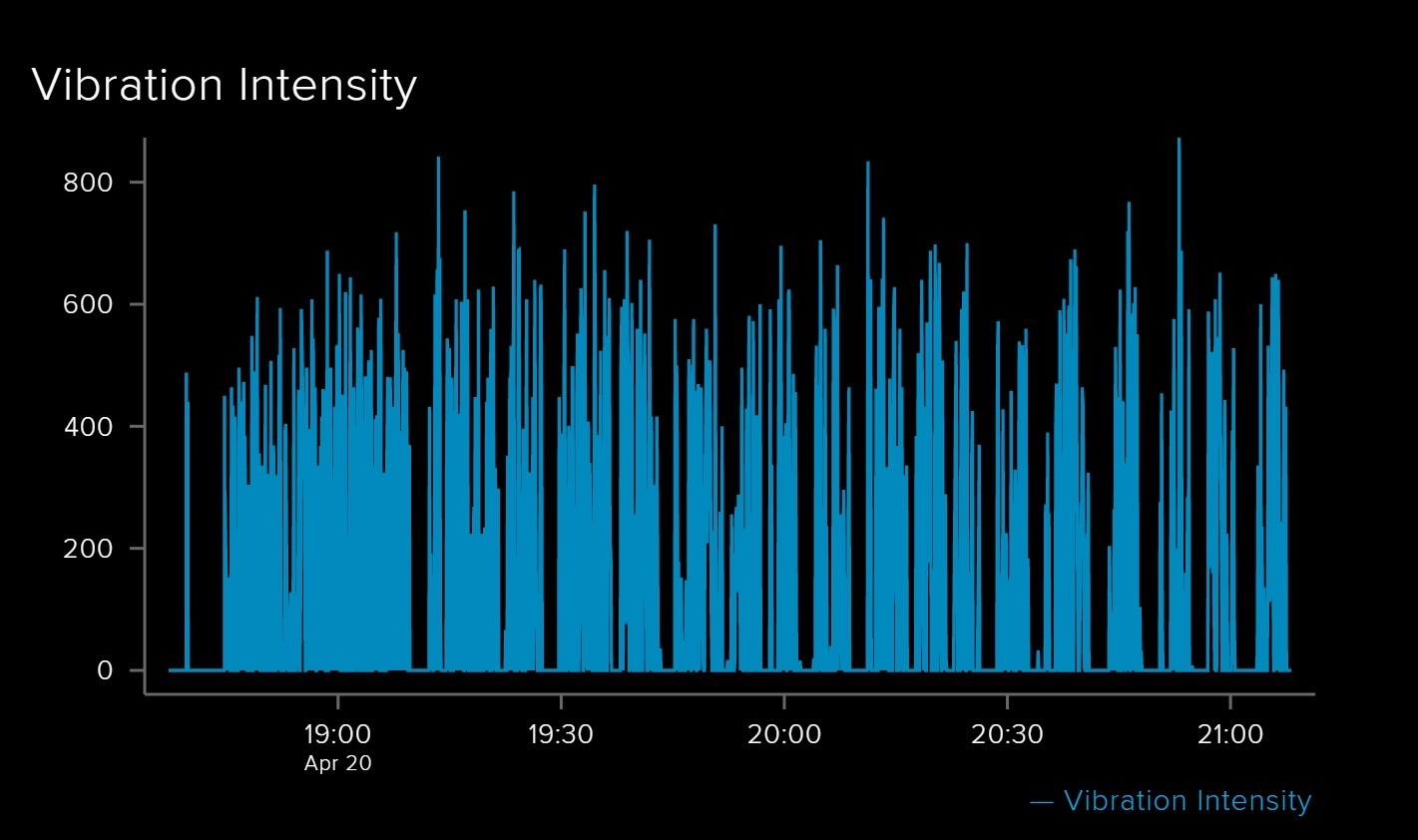
* A GUI of five line-charts are used for data display. They display the data output of the air conditioner from 19.00 PM, to 21.30 PM.
* Current sensor was connected. It shows Compressor-on / compressor-off periods clearly.
* Sound Sensor was not connected due to a website issue; when all five feeds are connected, the power feed doesn’t work properly. So, we had to eliminate the sound data temporarily.
* Smoke Sensor stimulated at 19.00 PM.
* Temperature sensor stimulated between 21.00 and 21.30 PM.

**Figure 5.11: Online Data Output of a Working A.C 2.5 Hours Duration**



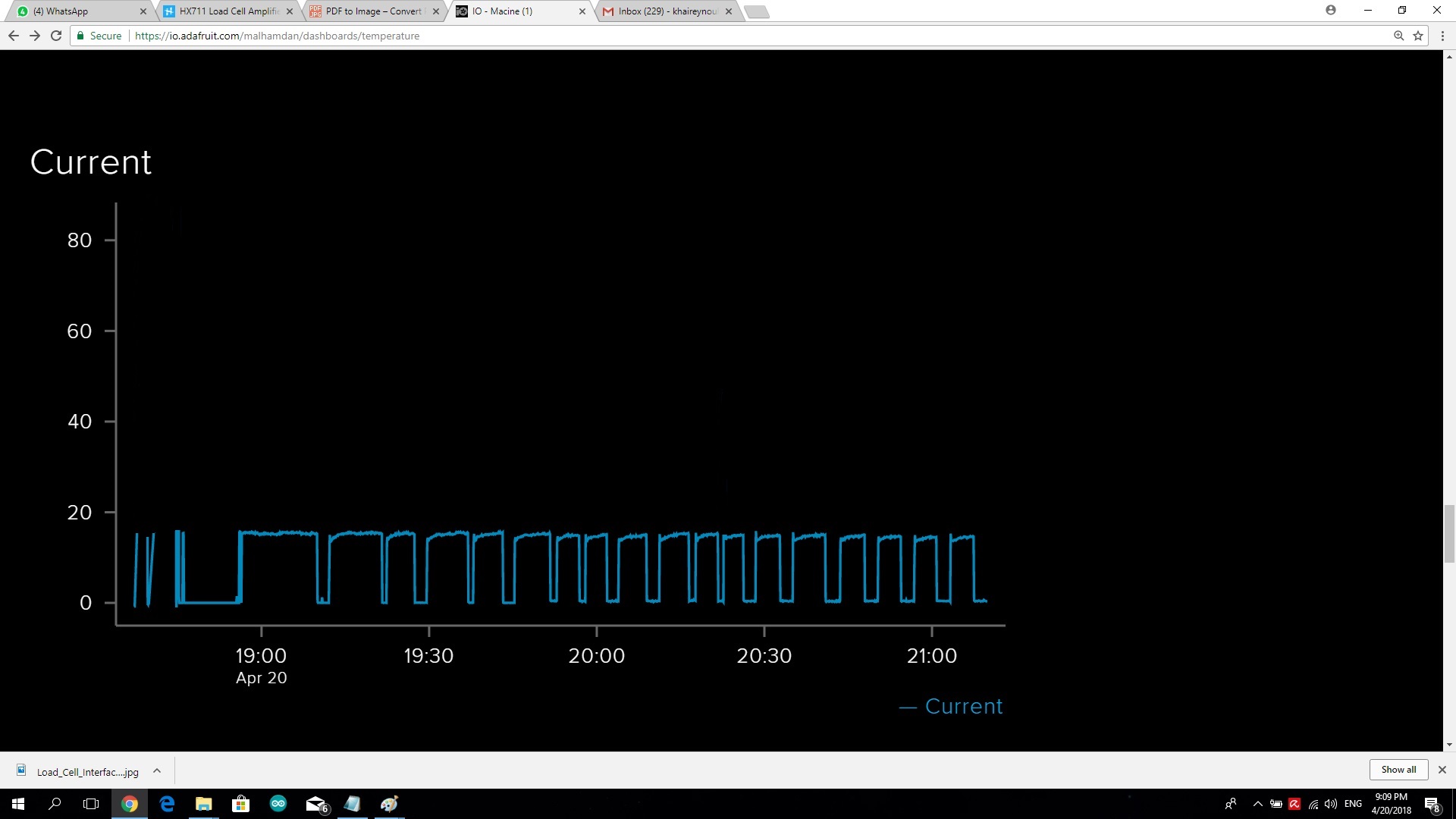
1. In figure (5.12) are the online data outputs of the vibration sensor connected to an old working air conditioner for a duration of 2 hours:

* It shows high values of vibration, specially at the times when the compressor starts (about 850) which if compared to a newly produced window A.C, it will be clear that this investigated A.C is defective, old, or of bad quality.
* **Figure 5.12: Online Vibration Data Output of a Working A.C 2 Hours Duration**



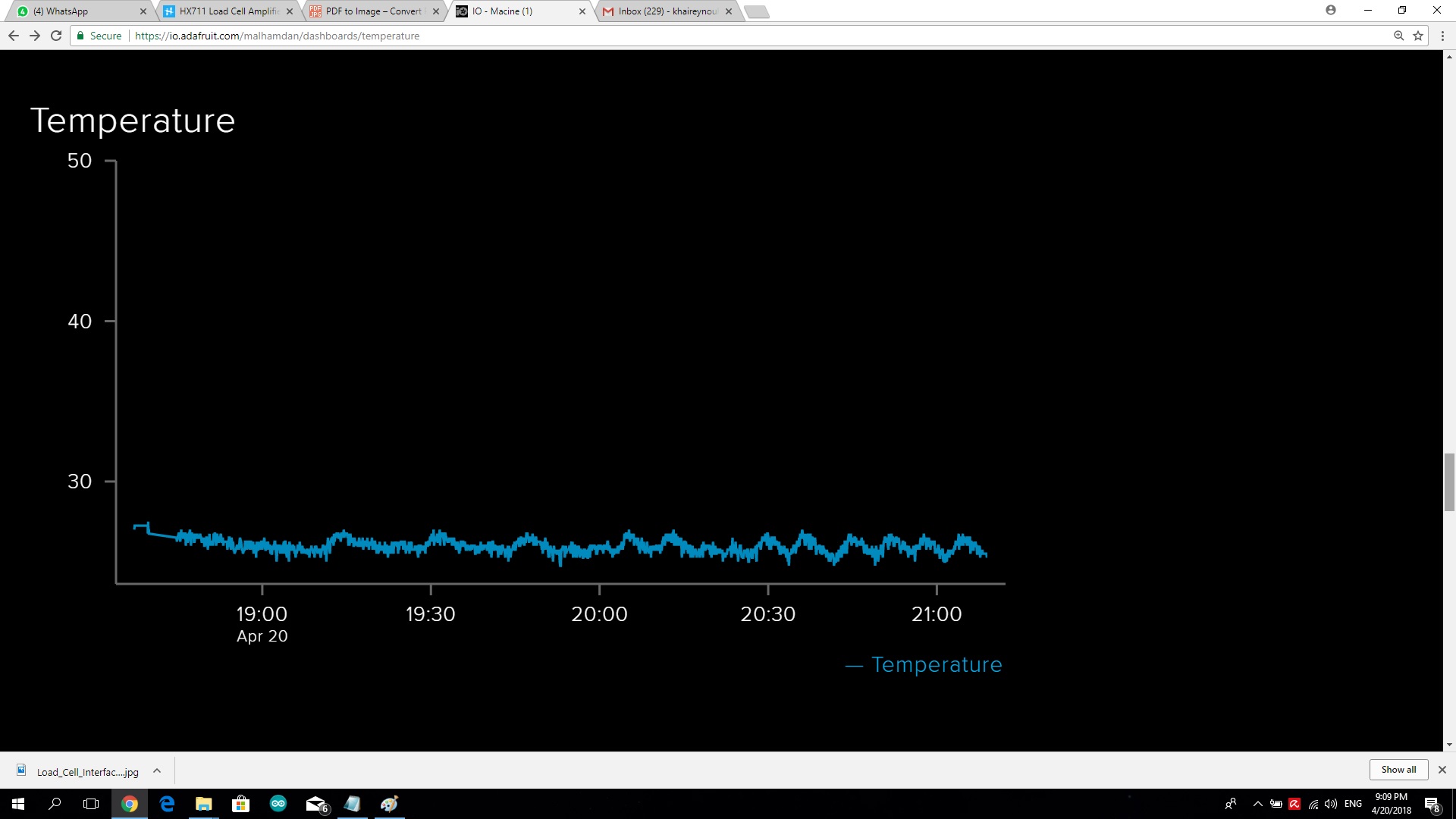
1. In figure (5.13) are the online data outputs of the current sensor connected to an old working air conditioner for a duration of 2 hours:

* It shows normal values of current withdrawal (about 15. A during compressor-on durations, and about 1.5. A to .75. mA during compressor-off durations).
* **Figure 5.13: Online Current Data Output of a Working A.C 2 Hours Duration**



1. In figure (5.14) are the online data outputs of the temperature sensor connected to an old working air conditioner for a duration of 2 hours:

* It shows normal values of temperature (about 24oc during compressor-on durations, and about 27oc during compressor-off durations).
* **Figure 5.14: Online Temperature Data Output of a Working A.C 2 Hours Duration**



1. In figure (5.15) are the online data outputs of the smoke sensor connected to an old working air conditioner for a duration of 2 hours:

* It shows normal values of smoke (0) at all times.
* The smoke sensor has been stimulated for proof nearly by 19.00 PM and it shows high values of smoke detected around the working A.C.
* **Figure 5.15: Online Smoke Data Output of a Working A.C 2 Hours Duration**



**CHAPTER 6:**

**(Research Results & Recommendations)**

## 6.1. Introduction to The Chapter

In this chapter are the research results recorded after building the *“IoT Machine-Guard”* device. It lists all issues encountered during experiments, research findings and recommendations based on these findings.

## 6.2. Research findings and Recommendations:

1. In the previous chapter, the IoT Machine-Guard device has been tested for proof.

Two experiments were made to test the efficiency and ability of the IoT Machine-Guard in reading, recording, and sending sensors’ data output wirelessly to a web broker.

1. The research revealed that the IoT Machine-Guard device is really efficient in accomplishing all the required tasks and appeared to be really useful.
2. The IoT Machine-Guard device could read all of the five sensors’ data outputs, compare them to pre-defined levels specific to each sensor, trigger an alarm, cut the power -immediately- off machine, send all these data outputs to an IoT web broker, and was able to send an email containing a message to the machine owner all at a time!
3. Those modern IoT devices have proved to be really useful in the field of industry in many ways. They can track and record machinery data, analyze these data outputs for quality measurements, and also for safety issues.
4. The researcher recommends using such modern IIoT devices in all factories and for any kind of electrical machines.

**References**

Alkhalil, A., & Ramadan, R. A. (2017). IoT Data Provenance Implementation Challenges. *Procedia Computer Science*, *109*(1), 1134-1139.

Baena, F., Guarin, A., Mora, J., Sauza, J., & Retat, S. (2017). Learning Factory: The Path to Industry 4.0. *Procedia Manufacturing*, *9*, 73-80.

Bryman, A., & Bell, E. (2015). *Business research methods*. Oxford University Press, USA.

Ehret, M., & Wirtz, J. (2017). Unlocking value from machines: business models and the industrial internet of things. *Journal of Marketing Management*, *33*(1), 111-130.

Hossain, M. S., & Muhammad, G. (2016). Cloud-assisted industrial internet of things (IIoT) - enabled framework for health monitoring. *Computer Networks*, *101*(1), 192-202.

Neuman, W. L., & Robson, K. (2014). *Basics of social research*. Pearson Canada.

Rio, R. (2017). The Intersection of Enterprise Asset Management and Industrial Internet of Things. *ARC Advisory Group*.

Savin-Baden, M., & Major, C. H. (2013). *Qualitative research: The essential guide to theory and practice*. Routledge.

Schweichhart, K. (2017). Reference Architectural Model Industrie 4.0 (RAMI 4.0) – An Introduction. *Plattform Industrie 4.0 and ZVEI, Administration Shell*. Retrieved from https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference\_architectural\_model\_industrie\_4.0\_rami\_4.0.pdf

Silicon Partners. (2017). Industry 4.0 and Industrial Internet of Things(IIoT). *The Silicon Partners*. Retrieved from http://www.thesiliconpartners.com/industry4-0-and-IIoT

Thanh, N. C., & Thanh, T. T. (2015). The interconnection between interpretivist paradigm and qualitative methods in Education. *American Journal of Educational Science, 1*(2), 24-27.

Upasani, K., Bakshi, M., Pandhare, V., & Lad, B. K. (2017). Distributed maintenance planning in manufacturing industries. *Computers & Industrial Engineering*, *108*(2), 1-14.

Wollschlaeger, M., Sauter, T., & Jasperneite, J. (2017). The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, *11*(1), 17-27.

Zezulka, F., Marcon, P., Vesely, I., & Sajdl, O. (2016). Industry 4.0–An Introduction in the phenomenon. *IFAC-PapersOnLine*, *49*(25), 8-12.