Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 1350929, 16 pages
https://doi.org/10.1155/2017/1350929

Research Article

An Application-Driven Modular IoT Architecture

Kumar Yelamarthi, Md Sayedul Aman, and Ahmed Abdelgawad

College of Science and Engineering, Central Michigan University, Mount Pleasant, MI 48859, USA

Correspondence should be addressed to Kumar Yelamarthi; yelam1k@cmich.edu

Received 2 February 2017; Accepted 29 March 2017; Published 14 May 2017

Academic Editor: Pierre-Martin Tardif

Building upon the advancements in the recent years, a new paradigm in technology has emerged in Internet of Things (IoT). IoT has allowed for communication with the surrounding environment through a multitude of sensors and actuators, yet operating on limited energy. Several researchers have presented IoT architectures for respective applications, often challenged by requiring major updates for adoption to a different application. Further, this comes with several uncertainties such as type of computational device required at the edge, mode of wireless connectivity required, methods to obtain power efficiency, and not ensuring rapid deployment. This paper starts with providing a horizontal overview of each layer in IoT architecture and options for different applications. Then it presents a broad application-driven modular architecture, which can be easily customized for rapid deployment. This paper presents the diverse hardware used in several IoT layers such as sensors, embedded processors, wireless transceivers, internet gateway, and application management cloud server. Later, this paper presents implementation results for diverse applications including healthcare, structural health monitoring, agriculture, and indoor tour guide systems. It is hoped that this research will assist the potential user to easily choose IoT hardware and software as it pertains to their respective needs.

1. Introduction

The "Internet of Things (IoT)" refers to a global network of objects, or "things," seamlessly connected to the internet which can fundamentally shift the way we interact with our surroundings. This IoT enables physical objects to see, listen, think, and perform tasks by sharing information and coordinating decisions. IoT transforms objects from being traditional to being smart by exploiting its underlying technologies such as sensor networks, embedded devices, communication technologies, and ubiquitous and pervasive computing. A growing number of physical objects are being connected to the internet at an unprecedented rate, leading to diverse range of applications including but not limited to smart cities [1, 2], smart homes [3–6], precision agriculture [7, 8], structural health monitoring, remote healthcare [6, 9], smart water and energy management [10], and smart cars [11–13].

This led to the development of complex and efficient applications and further technological challenges. Applications that have been developed on these new technologies must be tested, verified, and improved prior to real-time deployment. Simulation tools are useful as they offer a quick, flexible, and economical way to verify the behavior of an application. However, they lead to assumptions on several key factors in the test environment, leading to huge uncertainty. IoT applications and wireless sensor network are very much influenced by unpredictable events and physical characteristics that often result in inaccurate results during simulation.

Accordingly, there exists a strong demand to deploy real-time IoT systems, conduct hardware experiments, and benefit from appropriate tools for experimentation. Building upon this, several testbed platforms have been deployed by many researchers with various architectures, hardware, and topologies for a diverse range of applications. As IoT involves interconnecting a massive number of devices from many manufacturers and industries and performance strategy widely varies on different applications and user requirements, heterogeneity of devices and information is of paramount importance. Accordingly, architecture has been the backbone of IoT system, and traditional internet architecture needs to be revised to meet IoT challenges [14]. Furthermore, it is imperative to have a flexible layered architecture, especially at a time when the ever-increasing number of architectures has not converged to a reference model yet [15].

Numerous IoT architectures are presented in the literature [16–21] with each customized to address one specific application but not modular to be applicable for diverse applications. In this paper, we propose a modular IoT architecture that can be configured to serve the needs of different IoT applications including but not limited to (i) smart home, city, industry for environment control, optimizing energy usage, and automating appliances; (ii) agriculture for soil and temperature monitoring, optimal usage of irrigation systems, and early detection of plant diseases; (iii) healthcare for monitoring patient physiological parameters and remote motion tracking to name a few. This architecture is comprised of five modules: the physical sensor and actuator module that collects information and applies the needed feedback actions, a processing unit that captures and analyzes sensor data, wireless transceivers for sharing information, a gateway microcontroller/computer that manages the data from several microcontrollers and parses the data, and an application management cloud server in which the data storage and processing take place. Several prototypes have been built using the proposed IoT architecture, and extensive experiments have been conducted to illustrate how the proposed modular architecture can be customized with ease.

Primary contributions of this paper in comparison to related literature can be listed as follows:

(i) In comparison to related literature, this research proposes a modular IoT architecture that can be customized for a diverse range of applications.

(ii) The proposed architecture allows researchers to get up to speed and quickly deploy their IoT applications without having to dig through the details of standards specifications.

(iii) An overview of some key IoT implementations from the recent literature is presented.

(iv) Comparative analysis of different choices at each architectural level is presented.

The rest of this paper is organized as follows: Section 2 provides a summary of the related literature on different IoT architectures, applications, and their respective challenges. Section 3 presents the proposed modular IoT architecture along with a comparative analysis of choices at each architectural level. Implementation and evaluation of the proposed architecture for diverse applications are the focus of Section 4. Finally, Section 5 concludes this study.

2. Previous Work

This section presents an overview of existing literature in the IoT area, categorized based on the application and scalability. Of course, we do not pretend to be exhaustive, since the number of IoT applications in the world is huge.

2.1. Healthcare and Environment Monitoring Platforms. The first category represents the related IoT architectures that were proposed for healthcare and medical applications. The use of IoT in healthcare became of paramount importance in recent years, aiming at enabling preventive medicine and wellbeing preservation. IoT provides technology solutions to build networks of informed and connected e-patients, whereby communication among patients and healthcare and social care providers can take place in real time [22–24]. The main goal of such platforms is to provide low-cost, low-power, reliable, and wearable devices that collect the needed information for medical, active, and assisted living.

Research in [24] presents an IoT-based WSN for greenhouse environment monitoring. The authors claimed that the framework is scalable for other monitoring applications too and the edge nodes have the scope to be attached to additional sensors. Further, raw RSSI values are used for proximity sensing, with the major limitations being consistency and accuracy. Also, other applications have not been tested on this platform. Resolving these issues requires extensive mathematical algorithms [25–27], which are not feasible on the proposed architectures.

2.2. Structure Health Monitoring Platforms. Structure health monitoring (SHM) is a vital means to sustain the safety and maintainability of critical structures such as bridges, tunnels, and modern and historical buildings. SHM allows for nondestructive evaluations to detect the location and extent of structural damage, calculate the remaining life, and predict an upcoming accident. IoT-based SHM systems have recently increased due to its ease of deployment, low maintenance cost, and flexibility compared to traditional SHM technologies [28–32]. For instance, Lazo et al. proposed a solution for communicating devices that monitor the health of a bridge, whereby a 6LoWPAN was used [32]. Another example is the wireless platform featuring the TMS320F28335 DSP and the improved RF233 IEEE 802.15.4 wireless data transducer with up to 2 Mbps data rate.

2.3. Precision Agriculture IoT Platforms. Precision agriculture (PA) is another application area where IoT increases the efficiency, productivity, and profitability of many agricultural production systems. Real-time environmental information is remotely gathered from the agricultural environments and transferred to where it can be processed to discover problems, store data, or take necessary actions. Accordingly, several IoT systems have been developed for precision agriculture monitoring [7, 8]. IoT platforms have been recently developed to control the water consumption in irrigation. A smart irrigation system is illustrated in [8] where the authors designed and implemented a cloud-based field monitoring system that will guide the farmer about the soil condition of the field and has been implemented on a farm.

2.4. Localization and Object Tracking Platforms. One of the fundamental applications that can utilize IoT is indoor target localization and object tracking. Target localization and tracking systems play a crucial role in several context-aware applications through providing crucial information for positioning, tracking, and navigation, where the global positioning system (GPS) is typically infeasible indoors for poor satellite reception. This led to the development of a multitude of practical applications such as indoor localization in smart buildings [33, 34], navigation systems for the blind [35, 36], and autonomous navigational robots [37, 38].

In [16], the authors designed and implemented a data acquisition system for IoT. The system is simulated as a prototype for a monitoring platform but no performance analysis is conducted. An IoT-based framework is presented in [17] to facilitate the education of indoor localization employing Arduino and XBee wireless modules to perform indoor localization through RSSI-based triangulation. This is used to determine the position and dynamic map of nodes in the indoor environment but is limited in its accuracy and efficiency.

The above-mentioned works concentrated their work on some specific features or specific applications of IoT but did not focus on testing their platform in other applications, limiting the scalability and modularity of the architectures.

2.5. Scalability and Modularity. It is important to consider that any proposed system must utilize minimum resources and should be easy to implement. Kolios et al. [39] has proposed an event-triggering architecture, whereby the system consists of remote and local host and the event is triggered based on any unanticipated activities, detected, and monitored using the three phases of the event triggering and handling. This allows communication and computation to take place only when an event occurred, allowing less resource utilization and human intervention. The model is capable of contextualization, where the model can be attached to additional sensor node that can collect the context and sensor data from the mobile phone. The context can be stored in the data analytics and processed. However, one fundamental challenge in the proposed architecture is that it is not portable and also is not designed for low-power IoT applications. Balan et al. [40] have proposed a readily adaptable framework based on context-specific triggering, which collects data from user's smartphones such as location and movement patterns and provides suggestions such as directions. However, its implementation is challenged by expandability to include a diverse range of sensors and high-power intensive Wi-Fi based communication among devices.

Datta et al. [6] proposed a lightweight framework integrated with a mobile application that can connect heterogeneous devices. It supports only Android platform and uses sensor markup language for interacting with both actuators and sensors and uses a dedicated proxy for each communication channel. However, as semantic reasoning is used, the proposed model requires power and cost-intensive computational power, which is not always feasible in IoT applications.

The purpose of an IoT architecture is to collect and utilize information from the environment to monitor, control, optimize, and automate applications. However, the process of performing these operations is often not straightforward. First, the physical entity needs to be sensed by a transducer, and the change in the electrical property must be converted into a numerical value, done by the physical sensors and actuators layer. This raw data needs to be filtered, processed at an abstract level at first, and done by the low-power embedded processor layer. As the data is obtained from different remote locations in the environment, wireless transceivers are used to send the data from multiple low-power embedded processors to the local internet gateway. This internet gateway further analyzes and relays the information to application management cloud server for storage, extensive data analysis, and user feedback. As such, different tasks are to be performed at each layer in the architecture.

There are several architectures that consist of emulators, simulators, and physical environment with providing security and availability. Large scale projects can be easily deployed using multisite or federation deployment. Testbeds such as NET Eye and IoT-LAB are good scheduling systems and can easily configure the selected nodes [41]. Also, these existing models consist of very few sensor nodes and data produced is manually managed by the system administrators. Some researches [42–44] have been proposed to address this requirement. In [42], the authors described a three-layer architecture to include perception, network, and application layer. However, an architecture with only three layers includes many elements that cross multiple layers and make the implementation with off-the-shelf components a complex task. Similarly, Aazam et al. [43] presented a five-layer architecture for business applications, and Pacheco et al. [44] presented a four-layer architecture, with element crossing multiple layers, making real-time implementation a complex task. In contrast, the proposed architecture is built upon five layers, where each layer is responsible for some simple and easy to define tasks as presented in Figures 1 and 2. This enables modularity at each layer and makes real-time implementation a simple task.

3. Proposed Modular IoT Architecture

This section presents a modular IoT architecture for diverse applications including but not limited to healthcare, wearable assistive services, infrastructure health monitoring, and precision agriculture. The modular design in the proposed research comprises an architecture that is subdivided into different systems (layers) with several choices for each system and can be replaced or combined without affecting the functionality of the overall system. Broadly, the different layers in the proposed system include sensors, microprocessor, a wireless transceiver, gateway, and application management cloud server, as presented in Figures 1 and 2. Based on the application and design specifications requested, a low-power microcontroller with a few input lines could be replaced by a high-performance microcontroller with onboard analog to digital converter and multiple input lines. Similarly, the digital temperature sensor could be replaced by a simple thermistor when accuracy is not of high significance.

As all subsystem choices used in this research are off-the-shelf components with open-source software libraries available, minimal effort is required when the system is implemented for a different application. The input-output relation

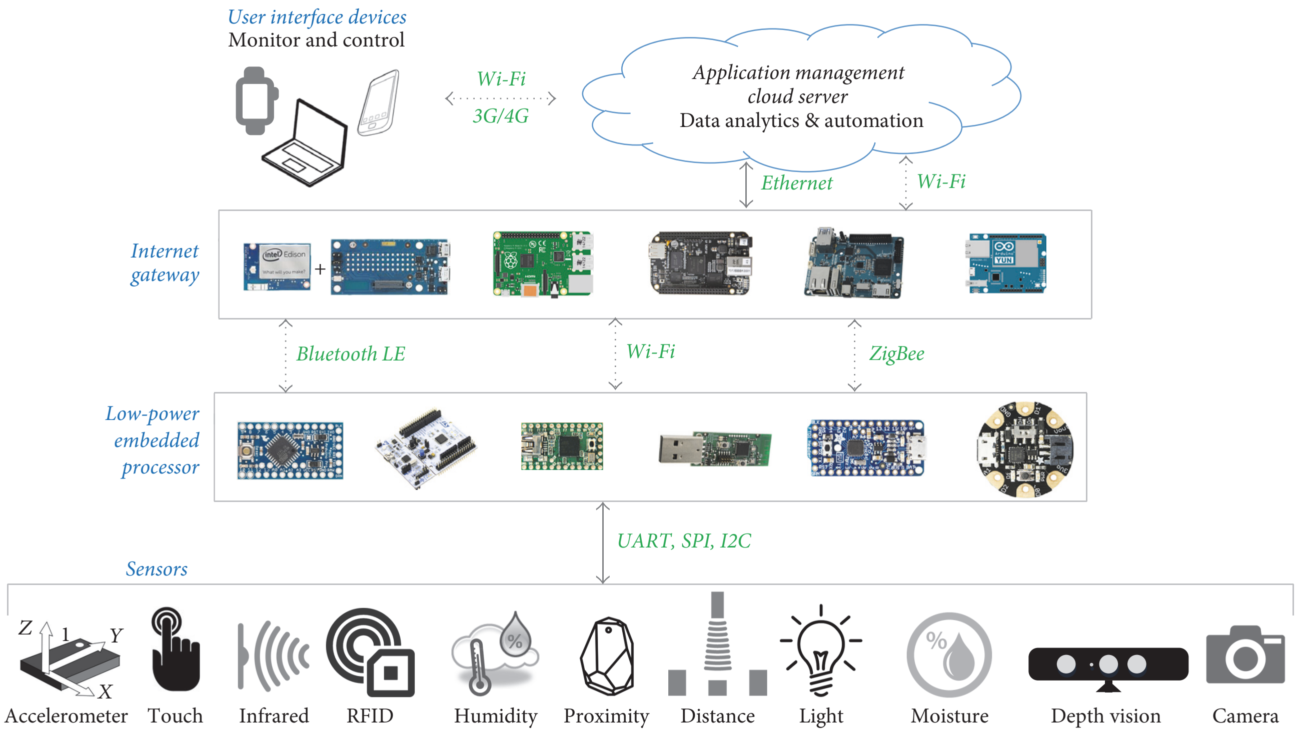


Figure 1: Layers in proposed modular IoT architecture.

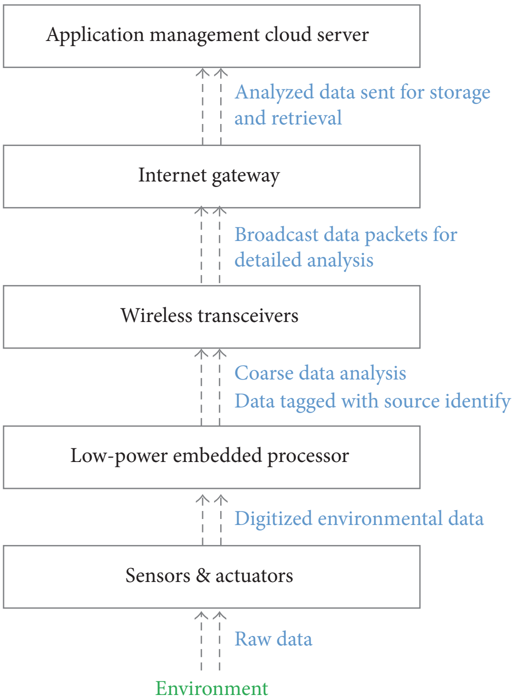

Application management cloud server
Analyzed data sent for storage
and retrieval
Internet gateway
Broadcast data packets for
detailed analysis
Wireless transceivers
Coarse data analysis
Data tagged with source identify
Low-power embedded processor
Digitized environmental data
Sensors & actuators
Raw data
Environment

Figure 2: Information flow with inputs and outputs at each layer.

between each of the layers of the architecture is shown in Figure 2, with modularity present at each layer individually. Cross-layer modularity has not yet been achieved in this work. As stated in Section 1, a primary contribution of this paper is to allow for rapid deployment without having to dig through details of standards specifications. Accordingly, each subsection herewith provides a comprehensive overview of choices at each architectural level, whereby the researcher can quickly decide the element of choice based on their respective application.

3.1. Physical Sensors and Actuators. The physical layer of the proposed modular architecture consists of the different application-specific sensors and actuators, as used to measure the attributes of different target applications. These sensors include healthcare sensors (electrocardiogram, pulse oximeter, and airflow), building ambience sensors (temperature, humidity, light intensity, presence, and indoor positioning), structural health sensors (piezoelectric and acoustic), and agricultural sensors (soil moisture, soil temperature, soil volumetric water content, wind speed, wind direction, rain meter, solar radiation, and leaf wetness) as presented in Table 1, all of which have been tested for feasible operation on the proposed modular IoT architecture.

At the fundamental layer, these sensors are interfaced to a low-power embedded processor to capture and parse raw data. As each sensor provides data in a unique format and the context of application might be different, the sensors first convert the sensed phenomena (e.g., temperature) into an equivalent electric voltage or current. Further, an onboard data converter is used to decode the information for faster processing and compatibility to the embedded processor. One such data converter to be used is a 6-bit CA3306 CMOS parallel ADC designed for low-power applications [62]. On the other hand, sensors that produce output in digital format

Table 1: Sensors operational with modular IoT architecture.

Sensor	Model	Reference
Accelerometer	ADXL345	[45]
Air humidity	HTU21D	[46]
Air temperature	SHT11	[47]
Camera	PixyCam, Raspberry Pi	[48]
Capacitive touch	MPR 121	[49]
Depth	Microsoft Kinect 1 & 2	[50]
Force	Flexiforce A20-25	[51]
Indoor localization	iBeacon	[52]
Light	GL5528	[53]
Proximity	HC-SR501	[54]
RGB color	TCS34725	[55]
Soil moisture	SEN0114	[56]
Sound	SEN-12642	[57]
Distance	HC-SR04	[58]
Pulse oximeter	SP02	[59]
Body temperature		[60]
Piezoelectric	PZT	[61]

(e.g., proximity) are connected directly to the embedded processor.

The majority of sensors in Table 1 require a battery and low computational power. As such, a low-power embedded processor presented in Section 3.2 could be utilized for portability and power efficiency. Certain application, on the other hand, requires significant processing power (e.g., depth sensor and seismic monitoring sensor). Accordingly, the internet gateway or single-board computers presented in Section 3.4 could be utilized to realize information and perform real-time data analytics.

In addition to capturing data, the fundamental IoT layer should also be able to provide feedback in visual or physical format. Accordingly, the proposed modular architecture is capable of providing this feedback. Per the visual feedback devices, the system can readily provide feedback through seven-segment, LED, and LCD displays while following UART, SPI, or I^2C communication protocols. When a physical feedback is necessary, it can be provided through mechanical controllers such as haptic actuators, DC motor, servo motor, and stepper motors, all of which are powered by an external battery. As the actuators and mechanical controllers use analog control signals, the embedded processors have to use an external digital to analog converter (DAC). One such data converter used in the proposed architecture is an MCP4725 DAC [63].

3.2. Low-Power Embedded Processor. The second layer in the proposed modular IoT architecture is the low-power embedded processor. The primary role of this processor is to capture sensor information, perform fundamental data analysis, provide feedback to the user as necessary, and relay information to the internet gateway. Depending on the application, this embedded processor can be powered from a battery or an energy scavenging module. Due to the scope of the portable IoT application, this section is limited to battery powered embedded processor. Also, it has to be considered that an exhaustive list of embedded processors is not feasible, and accordingly only the most promising systems in terms of rapid deployment and low-power usage are presented in this study. A comparative list of these embedded processors is presented in Table 2.

Arduino based microcontrollers (MCU) are one of the most popular prototyping platforms among researchers and are readily available for quick implementation. While the most popular MCU from Arduino is the Uno [64], it is not considered here as it consumes high power and has a large form factor compared to its alternatives. The Arduino Pro Mini [65] is a low-power MCU with a small form factor and 32 KB of flash memory and operates at 20 MHz. The built-in ADC and input-output connectivity via UART, SPI, and I2C make it a popular choice. On the flip side, it does not comprise an onboard USB for faster interface with a host computer.

The NUCLEO-F401RE is one MCU considered here due to its high-performance capability and open-source software [66]. Unlike traditional MCUs, this has a 32-bit MCU operating at a relatively high frequency of 84 MHz for implementation of computationally intensive applications. The readily available compatibility with ARM mbed OS (free IoT-based operating system) combined with onboard storage capacity makes this MCU a good choice. However, one fundamental challenge in this MCU is the high power consumption.

ATmega32U4 is a low-power Atmel 8-bit AVR RISC-based MCU with 32 KB self-programming flash program memory, 2.5 KB SRAM, 1 KB EEPROM, and 12-channel 10-bit ADC. The device achieves up to 16 MIPS throughput at 16 MHz. This MCU is very close to the Arduino Pro Mini MCU but has a smaller collection of sensors that could be readily interfaced. ATtiny10 [67] is another 8-bit MCU operating at a speed of 20 MHz and also has a very small form factor [68]. However, the small (1 KB) onboard flash memory and limited I/O connectivity options limit the applicability of this MCU for diverse applications.

MSP430F2410 is a member of the Texas Instruments MSP430 family of ultra-low-power MCUs optimized to achieve extended battery life in portable IoT applications [69]. This is based on a powerful 16-bit RISC CPU, with 16-bit registers, four universal serial communication interface (USCI) modules, and up to 48 I/O pins. MSP430F2410 features 256 B flash memory and 4 KB of RAM. This MCU serves the needs of many IoT systems, industrial control applications, and hand-held meters. However, one fundamental challenge is the lack of open-source or free software available to implement the MCU in diverse applications. Furthermore, the complexity and time to deploy an IoT application using this MCU are much higher in comparison to other alternatives.

The Texas Instruments CC2640 is a SimpleLink wireless MCU with a built-in powerful ARM Cortex M3 processor operating at a speed of 48 MHz [68]. The low-power utilization and the different I/O connectivity modes allow for interfacing a broad range of sensors while operating on a coin cell battery. Another primary advantage of the

Table 2: Comparison of low-power microcontrollers.

Characteristic	Arduino Pro Mini	NUCLEO-F401RE	ATmega32U4	ATtiny10	CC2640	MSP430-F2410	Adafruit Pro Trinket	Adafruit FLORA
CPU	ATmega328	STM32-Cortex M4	AVR	ATtiny10	ARM Cortex M3	MSP430	ATmega328	ATtiny85
Built-in transceiver	✘	✘	✘	✘	BLE or ZigBee	✘	✘	✘
ADC: size (bits), samples/s	10, –	16, 2400	10, 15 k	8, 15	12, 200 k	12, –	10, –	10, –
Clock speed (MHz)	20	84	16	12	48	16	12	8
Bus width (bits)	8	32	8	8	16	16	8	
Memory (kB)	32	512	32	1	128	56	28	8
I/O connectivity	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	ISP	UART, SPI, I2C, GPIO	UART, IrDA, I2C, GPIO, SPI	UART, I2C, SPI	UART
Real-time clock	✘	√	✘	✘	√	✘	✘	✘
Coin cell battery operation	√	✘	√	√	√	√	√	√

CC2640 microcontroller is its built-in wireless transceiver (BLE or ZigBee). However, similar to MSP430F2410, this MCU requires expensive software and the complexity and time to deploy an IoT application are much higher in comparison.

Adafruit Pro Trinket 3 V [70] is an alternative to other MCUs presented with the advantages of small form factor, readily available support for a diverse range of sensors, low-power consumption, I/O connectivity, and decent computational efficiency to collect and analyze data for a broad range of IoT applications. Furthermore, onboard USB for connectivity with host computer allows for rapid prototyping and implementation, and operation of a coin cell battery allows for operation in extended time periods.

Lastly, Adafruit FLORA is a miniature MCU board that is an ideal choice for wearable IoT applications [71]. Adafruit FLORA is powered by ATtiny85 CPU and is programmable using Arduino IDE. ATtiny85 onboard chip runs at 8 MHz and has 8 K of flash, 512 bytes of SRAM, and 512 bytes of EEPROM [72]. This microcontroller can be considered to have ultralow power as it only draws 9 mA current while running, with one limitation being the number of simultaneous connections available for sensors and actuators.

The presented MCUs have been programmed for a simple IoT application, whereby they had to capture data from a light sensor at a frequency of 10 Hz and transmit it to the gateway. Current consumption from these MCUs is obtained and presented in Table 3. When operating in low-power mode, the MCUs were capturing data from sensor once every 15 seconds and going to sleep the remaining time, with a total duration of 5 minutes. It is clear that NUCLEO-F401RE is not an ideal choice due to its highest current consumption. It is worth stating that while Arduino Pro Mini has a small form factor compared to an Arduino Uno, it consumes higher current in both normal and low-power modes. While both Adafruit Pro Trinket and TI CC2640 consume the same amount of current, Pro Trinket would be a good choice purely based on the availability of open-source software.

Table 3: Current consumption of MCUs in different modes.

MCU	Low-power (mA)	Normal (mA)
NUCLEO-F401RE	n/a	120.0
Arduino UNO	18.9	25.4
Arduino Pro Mini	23.3	37.4
Adafruit Pro Trinket	3.8	8.4
CC2640	n/a	8.2

3.3. *Wireless Transceiver Technologies.*

The wireless transceiver provides connectivity to transmit and receive data between the different components of the architecture. However, wireless communication comes at a cost of increased power usage due to the high-energy consumption during transmission [73]. Addressing this challenge, numerous low-power wireless technologies have been proposed. This section compares the relevant wireless technologies as presented in Table 4.

Bluetooth Low Energy (BLE) is a short-range standard developed by the Bluetooth Special Interest Group (SIG) [74], based on the IEEE 802.15.1 standard, and operates in the 2.4 GHz spectrum. With a peak current of 12–16 mA and a throughput of 120 bytes/second, a BLE unit would draw an average current of 49 μA and last approximately 191 days [75]. Furthermore, it provides a throughput of 305 kbps and a transmission range of 50+ meters at latency of approximately 2.5 ms. It adopts an adaptive frequency hopping technique and works seamlessly to reduce interference. Being the widely adopted choice and readily built into numerous devices, BLE is the ideal candidate in portable IoT applications.

ZigBee was developed by a group of 16 companies and operates based on the IEEE 802.15.4 standard [57]. However, the absence of frequency hopping techniques limits the ability to reduce interference, and the wide channel separation of 5 MHz does not effectively utilize the frequency band. With a peak current of 30–40 mA, it could theoretically operate on a coin cell battery [75]. However, the complexity of

Table 4: Comparison of wireless technologies.

Characteristic	BLE	ZigBee	Wi-Fi	IrDA	NFC	ANT+	nRF24
Peak current (mA)	12–16	30–40	116	10	50	17	18
Max throughput (kbps)	305	100–250	6000	10^6	424	20	1000
Range (m)	50	100–300	50–150	0.10	0.20	10	10
Latency (ms)	2.5	20	1.5	25	1000	0	1000
Power efficiency (μW/bit)	0.153	185.9	0.00525	11.7	–	0.71	2.48
Coin cell battery operation	√	√	✘	✘	✘	√	√
Battery time (approximate days)	191	✘	✘	✘	✘	52	170

ZigBee devices and transmission protocols limits the ability to operate on a coin cell battery, unless it is integrated with the system-on-chip on the MCU [76].

Wireless Fidelity (Wi-Fi) follows the IEEE 802.11 standard and operates in the 2.4 GHz frequency spectrum. It has been designed for large data transfer with a high throughput and hence consumes relatively higher electric current. With a peak current of 116 mA, it is not suitable for coin cell battery operation and hence for wearable IoT systems. However, Wi-Fi is a good candidate for communications between the microcontrollers and the Internet due to its high data rates (up to 150 Mbps).

The Infrared Data Association (IrDA) SIG developed the IrDA technology for a high throughput of 1 Gbps for near-field communication [77]. However, it requires a clear line of sight and has a range of 10 cm, limiting its implementation. The near-field communication (NFC) is a paradigm that allows the communication with passive NFC tags without the need for a power source [75]. However, NFC applicability is limited to communications over a range of 20 cm.

ANT+ is a proprietary technology operating in 2.4 GHz spectrum [78] and is commonly used in sports and fitness equipment. Utilizing an efficient chipset and a throughput of 120 bytes/second, when switched on continuously with a 225 mAh battery, would draw an average current of 175.5 μA and last 52 days [75]. Also, a throughput of 20 kbps, a transmission range of 10 m, and latency of almost 0 ms make it an ideal candidate for real-time monitoring and tracking applications.

Another proprietary wireless technology is the nRF24 single-chip ANT™ ultra-low-power wireless solution designed to work with Nike and Apple devices. This technology is developed by Nordic Semiconductors and operates in the unlicensed 2.4 GHz band. With a peak current of 12.3 mA, it can operate on a single coin cell battery. For example, a Nike+ unit using this technology would last approximately 42 days with a constant data rate of 34 bytes/s [75]. However, the proprietary nature of the ANT+ and wireless technology limits its widespread adaptability in IoT systems.

3.4. Internet Gateway. The different front-end nodes collect sensor data and relay it to an internet gateway. This gateway further relays the information (possibly after analyzing it) to the application management cloud server for storage and extensive data analysis. The gateway also forwards requests from the server to the embedded processors for actuators and feedback devices on the front-end nodes.

An internet gateway in the proposed modular IoT architecture is mainly composed of four modules: (1) a low-power wireless interface module, which serves as a means to receive data wirelessly from the several nodes in its vicinity, (2) a data collection module, which offers the memory space needed to collect all the data received from the nodes, (3) an MCU or single-board computer that performs intermediate data analysis, (4) and a high-rate wireless/wired interface that passes output data to the application management server for storage and detailed analysis.

In addition to parsing data between the nodes and cloud, this gateway has the ability to serve as a supernode with significant performance capability. Furthermore, it serves as the cluster head or a network coordinator that is in charge of overlooking a few networking functionalities such as Medium Access Control (MAC) and multihop data routing decision. For instance, it can compute the time division multiple access (TDMA) schedules through which nodes can access the gateway and share it with its connected nodes. Moreover, it can serve as the root node of an IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) tree, which is then used for multihop data forwarding. The gateway node is also in charge of network-wide security services such as data authentication and encryption, since the low-power MCUs do not have enough resource to handle such energy-hungry security services. Table 5 compares the different implementation alternatives used in the proposed architecture.

Intel Edison is a single-board computer, used in low-power sensor network applications [79]. With an onboard 32-bit Intel Quark processor operating at a speed of 100 MHz, it allows for extensive onboard computations. A large onboard memory of 4 GB allows for storage and fast retrieval of historic data. The built-in BLE transceiver chip and the diverse I/O connectivity methods allow for easier interface with a broad range of sensors.

The Raspberry Pi 3 single-board computer is a widely used candidate for implementing gateway nodes. This can be powered by a 5 V Li-Ion battery and features a Linux operating system with a wide set of programming and connectivity options [80]. Equipped with a 1.2 GHz 64-bit quad-core ARMv8 CPU and 1 GB RAM, this gateway single-board computer has the power to process data from computationally intensive sensors such as depth or seismic monitoring

Table 5: Comparison of Internet gateway.

Characteristic	Intel Edison	Raspberry Pi 3	BeagleBone Black	ODROID-XU4	Arduino Yún
CPU	Intel Atom + Intel Quark	ARM Cortex-A53	ARM Cortex-A8 + Dual PRU	Samsung Exynos 5422 Cortex™-A15 and Cortex-A7	ATmega32U4 + Atheros AR9331
GPU	✘	Broadcom VideoCore-IV	PowerVR SGX530 GPU	ARM® Mali™-T628 MP6	✘
Built-in wireless transceiver	Wi-Fi, BLE	Wi-Fi, BLE	✘	Wi-Fi	Wi-Fi
ADC: channels, bits	✘	✘	8, 12	✘	6, 12
Clock speed (MHz)	500, 100	1200	1000, 200	2000	16, 400
Bus width (bits)	32	32	32	32	8
RAM (MB)	1024	1024	512	2048	0.025, 64
Flash storage (GB)	4	✘	4	✘	0.000032, 0.016
Low-level peripherals	UART, PWM, SPI, I^2C, I^2S	UART, SPI, I^2C, I^2S	UART, PWM, SPI, I^2C, CAN	UART, SPI, I^2C, I^2S	UART, SPI, I^2C
Power ratings	100 mA @ 3.3 V	800 mA @ 5 V	460 mA @ 5 V	4 A @ 5 V	50 mA @ 3.3 V, ✘

sensors or handle simultaneous data from numerous sensors listed in Table 1. The built-in Wi-Fi and BLE transceivers allow for swift wireless connectivity and provide the ability to deploy in remote applications. Furthermore, the four USB ports and 40 general-purpose input-output pins allow for interfacing a wide range of sensors and actuators listed in Table 1 with ease.

BeagleBone Black is another single-board computer with a low-cost platform for IoT applications running Linux operating system [81]. It operates with AM335x 1 GHz ARM Cortex-A8 processor, 512 MB RAM, 4 GB 8-bit eMMC onboard flash storage, and a 3D graphics accelerator. Depending on the target application, this MCU can configure up to 65 general-purpose digital IO pins, eight PWMs, four timers, seven 1.8 V analog inputs, some serial UARTs, two I2C, and two SPI along with access to 3.3 V and 5 V. The additional advantage provided by this unit is the built-in ADC for acquiring real-world analog data without much hassle and the CAN bus interface for ready deployment in automobile applications.

ODROID-XU4 is another device with more powerful hardware and a smaller form factor [82]. This gateway computer is equipped with the Samsung Exynos 5422 Cortex-A15 2 GHz and Cortex-A7 octa-core CPUs with 2 GB of RAM and eMMC5.0 HS400 flash storage. This gateway single-board computer supports different versions of Linux. Using USB 3.0 and Gigabit Ethernet interfaces, ODROID-XU4 offers high data rates which are increasingly required to support advanced processing power on ARM devices. This allows for rapid streaming and analysis of data in time-sensitive applications.

Finally, Arduino Yún is another MCU based on ATmega32U4 and Atheros AR9331 [83]. The Atheros processor supports a Linux distribution based on OpenWrt named OpenWrt-Yun. The Arduino Yún board has built-in Ethernet and Wi-Fi support, a USB-A port, micro-SD card slot, 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and 3 reset buttons.

Table 6: Current consumption of gateway controllers with wireless transceivers.

Gateway MCU	Average current consumption (mA)
Intel Edison with BLE	285.30
Raspberry Pi 2 with BLE	294.83
Raspberry Pi 3 with BLE	328.69
Raspberry Pi 3 with nRF24L01	347.40
ODROID-XU4 with BLE	606.89
BeagleBone Black with BLE	320.24
Arduino Yún with BLE	283.01

ATmega32U4 has 32 KB (with 4 KB used for the bootloader).

As the input-output connectivity, form factor, and applicability with a diverse range of sensors are almost similar in all these gateway controllers, one simple test that can be performed is evaluating the current consumption. Accordingly, these gateway controllers have been programmed with a simple IoT application, whereby they had to capture data from an MCU at a frequency of 10 Hz and transmit it to the application cloud server. Current consumption from these gateway controllers is obtained and presented in Table 6. It is evident that while ODROID-XU4 has superior performance ability, the current consumption is very high in comparison with others. So, it is ideally suitable for applications where high computation performance is the primary criteria such as scanning the environment with a depth sensor on a drone or in the case of an indoor-based navigation system for the blind [36]. While BeagleBone Black consumes low current

Table 7: Comparison of application management systems.

Platform	E1	E2	E3	E4	E5	E6	E7
TinyREST [84]	√	✘	√	n/a	✘	✘	√
Evrythng [86]	✘	√	√	√	√	√	√
ThingWorx [87]	✘	✘	√	√	√	√	✘
Sensorpedia [88]	✘	✘	√	√	√	√	√
Paraimpu [89]	✘	✘	✘	√	√	√	√
Proposed	√	√	✘	√	√	✘	√

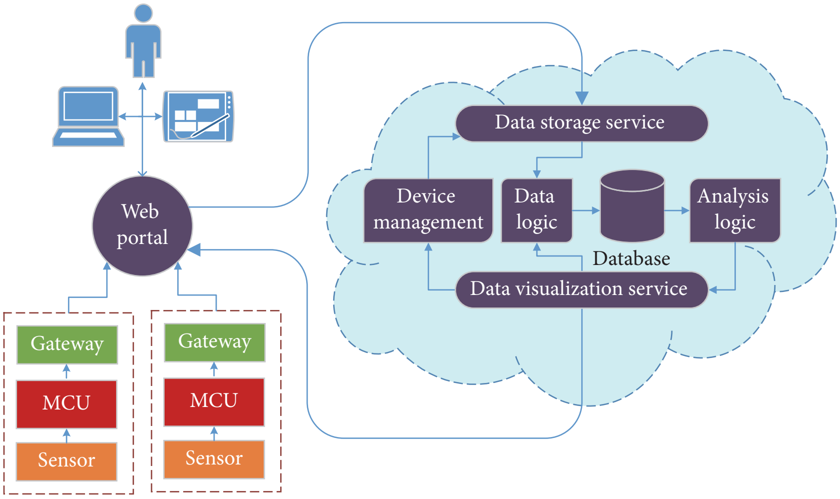


Figure 3: Architecture of application management cloud server.

in comparison, its implementation is limited due to the challenges in support for a diverse range of sensors and actuators. On the other hand, the built-in CAN-bus and broad adoption of such protocol in automobile applications make this a popular gateway controller. Lastly, Raspberry Pi 2 and Raspberry Pi 3 have low current consumption in comparisons with other gateway controllers. Further, support for a diverse range of sensors and actuators and readily available support for interface with all of the MCUs listed in Table 2 make Raspberry Pi 2 with BLE transceivers the ideal choice of gateway MCU.

3.5. Application Management Cloud Server. The application management cloud server is responsible for facilitating the end-users' ability to access the sensed data. This is achieved by implementing several services including, but not limited to, data storage, data analytics, and data visualization in addition to providing an appropriate application program interface (API) and software tools through which the end-user can access the data as in Figures 1 and 3. This section provides an overview of several relevant projects and initiatives in application management cloud servers and compares each of platforms most common metrics, including device discovery (E1), integration of things to the web (E2), RESTful interaction with the platform (E3), data formats (E4), security (E5), physical mashups (E6), and web messaging (E7), as presented in Table 7.

Device discovery is one fundamental feature necessary in IoT applications, but the majority of the existing systems with the exception of TinyREST [84] do not support it. A recent survey [85] presents numerous approaches for discovering elements and their respective data on the web, but none of the presented systems have them yet. The requirement of integrating things to the web through either gateway controllers or web servers is supported by only a few platforms such as Evrythng [86] and the custom solution presented in this work. Meanwhile other systems employ embedded HTTP servers and 6LoWPAN protocols to integrate things to the web directly. RESTful interaction is a requirement satisfied by the majority of the systems presented as they provide APIs for their users to interact with their respective devices, services, or data collected. Also, most systems support well-known and widely accepted data formats such as TXT, XML, JSON, CSV, and HTML, all of which are employed based on the device and application in a test.

Security and privacy to data are two of the most common attributes to be considered in IoT application, and, accordingly, the most common platforms account for the same. Platforms such as Evrythng, ThingWorx [87], Paraimpu [88], and custom proposed system require an API key in the HTTP header, and an open standard for authorization is adopted by Sensorpedia [89]. One appealing aspect of majority of systems is that web mashups can be created in any programming language supporting HTTP communication. Overall, each application management system has its own unique

8 bits	4 bits	12 bits	10 bits	10 bits	10 bits	10 bits
Destination ID	Source ID	Sensor types	Sensor 1 data	Sensor 2 data	Sensor 3 data	Sensor 4 data

Figure 4: Configuration of data packet between MCU and gateway controller.

advantages and could be chosen based on the application of interest.

4. Implementation and Evaluation

The proposed system architecture has been implemented in hardware and subjected to several tests for a variety of applications, including but not limited to healthcare, smart home, and structural health monitoring. As low-power consumption is the key factor in portable IoT application, a sampling rate of 1 Hz was set for all sensors, except in case of accelerometer, where a higher rate was necessary. Data from the individual or combination of sensors are collected and packaged into 8 bytes with information of source location and sensor data as in Figure 4. This information is transmitted through the low-power transceiver to the gateway controller for analysis. Further, information from all MCUs in the vicinity is collected for comparative analysis as necessary and transmitted to the application management cloud server for detailed data analysis and storage.

4.1. Range and Current Consumption. One of the fundamental tests in assessing the efficiency of the proposed system is identifying current consumption and also reliable communication distance. Accordingly, during the first test, the power amplifier level is set at minimum (−18 dBm) and the data rate has been changed to find the current consumption and maximum communication distance. Results obtained as presented in Table 8 show that the data rate has minimal influence on the power consumption but does vary the communication distance significantly. In the second test, the data rate has been set at 20 Kbps, and the power amplifier level has been changed to find the power consumption and maximum communication distance. Results obtained as presented in Table 9 show that the data rate has low influence on both power consumption and communication distance.

Table 8: Power consumption of sensor nodes with varied data rate.

Data rate (Mbps)	0.25	1	2
Power consumption (mW)	0.47	10.48	10.55
Reliable communication distance (m)	>100	56.41	39.75

Table 9: Power consumption of sensor nodes with varied power levels.

Power amplifier level (dBm)	−18	−12	−6	0
Power consumption (mW)	10.47	10.48	10.48	10.50
Maximum communication distance (m)	>100	>100	>100	>100

4.2. Healthcare: Posture and Physical Activity. Healthcare applications are one area of interest where the proposed architecture could be implemented. With posture and physical activity recognition being two of the fundamental and yet complex tasks, the proposed system has been customized to perform these tasks. First, the sensors utilized in this test are an accelerometer for physical activity recognition, body temperature sensor for monitoring user health, and force sensor to detect sedentary activity. Second, with portability and low-power consumption being a major deciding factor in wearables, Adafruit FLORA has been chosen as the MCU and BLE transceiver has been chosen to relay information to the gateway. As this gateway had to collect information from multiple users simultaneously and relay it to the application cloud server, a system with decent computational power and wired connectivity to the network would suffice. Accordingly, Raspberry Pi 2 has been chosen. Data collected from the sensors required the application of complex mathematical algorithms to detect physical activity. Accordingly, the custom-designed cloud server analyzed this data in MATLAB and provided feedback to the user via a simple text message as presented in Figure 5.

Data obtained from the application management cloud server for activities such as walking, jogging, and running are presented in Figure 6. Through a simple FFT analysis with 512-sample window, the application management cloud server was able to recognize the data patterns to classify walking, jogging, and running, respectively, as presented in Figure 7. In addition, this module can be used in assisted living community to detect sudden fall of the user as in Figure 8 and trigger an alert to the appropriate personnel via the gateway controller and application management cloud server.

4.3. Infrastructure Health Monitoring: Damage Detection. The deteriorating health of complex and intricate infrastructure has challenged the engineers to find efficient solutions in infrastructure health monitoring. Addressing this challenge, the proposed framework has been customized to design, build, and test an IoT-based SHM system as presented in Figure 9. First, sensors utilized in this test are piezoelectric transducers placed at strategic locations to transmit and receive acoustic signals. Second, generating the high-frequency acoustic signals required a computationally intensive MCU. Moreover, this application requires deployment in remote areas with limited connectivity. Addressing these two requirements, Raspberry Pi 3 that served as both MCU and gateway was used. As transmission of huge amounts of data to the cloud server requires extensive power, real-time data analysis had to be performed on the device. Accordingly,

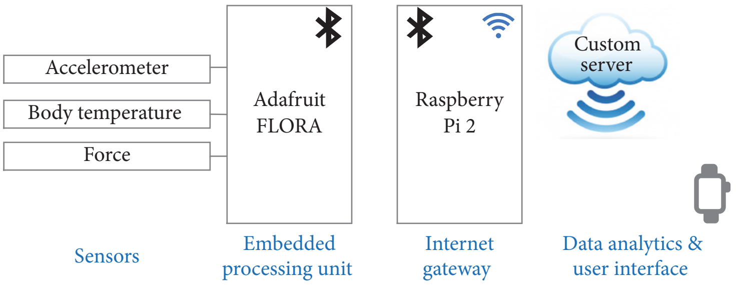


FIGURE 5: Architectural components of the prototype for healthcare applications.

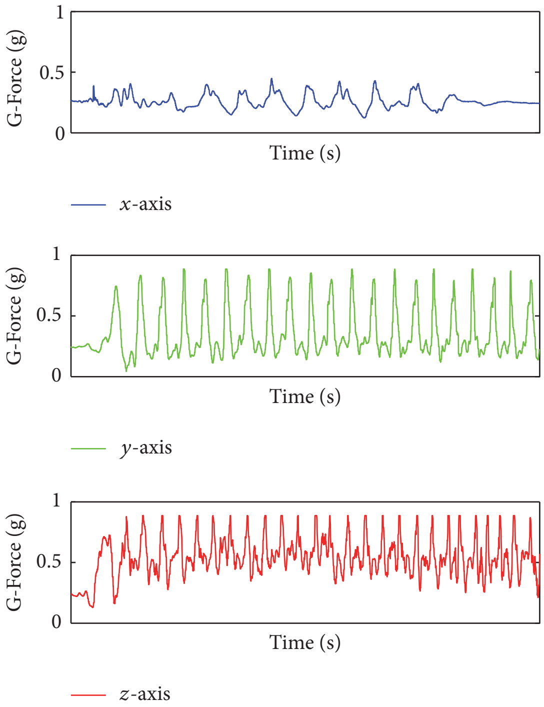


FIGURE 6: Movement patterns information as recorded on the cloud server.

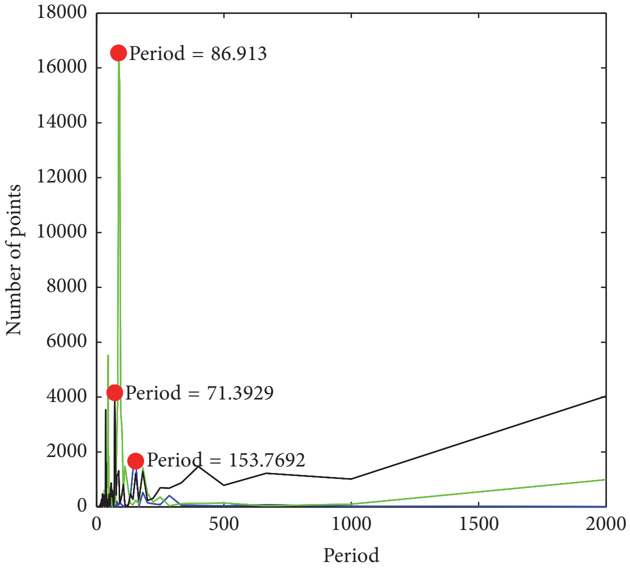


FIGURE 7: Classification of user activity on the application management server.

the system has been extended with DAC, ADC, and a high-speed buffer. Once the data was analyzed, the findings were sent to the ThingWorx application management cloud server through simple 3G connectivity and were available to the users via a website interface as in Figure 10.

This data stored in the cloud server can be monitored remotely from any mobile device, and an alert could be triggered by the user whenever the values from a location pass a threshold. Overall, the presented framework and model utilized has been demonstrated to obtain a maximum of 1.03% error for the damage location and a maximum of 8.43% error for the damage width.

4.4. Portable Tour Guide Module: Location-Based Service. The advancement in BLE technology revolutionized indoor contextual awareness and allowed for advancement in a diverse range of applications. One such application is location-aware systems to serve as tour guides in the museums, historical sites, and academic environments. The proposed architecture was validated for this location-based service through the design and implementation of a portable tour guide module presented in Figure 11 and system architecture in Figure 12.

The two technologies often utilized in indoor localization are passive RFID and BLE. Additionally, one popular feature in tour module is the ability to take pictures and/or record videos while in the tour. Accordingly, first, the presented system has been equipped with RFID and BLE transceivers and a 5-Megapixel camera to take pictures and videos by the user as appropriate. Second, in addition to recording videos, one method often used to relay information to users while in a tour is showcasing video as related to the object or location of interest. Accordingly, the presented module is enabled with a small LCD screen to display and play videos while in the tour. Further, performing these graphics-intensive applications require an embedded processor with onboard GPU, and, as such, a Raspberry Pi 3 has been used. Information from this module was relayed to the cloud server via Wi-Fi. One particular feature required in this tour module is presenting the location of the user on the campus map in

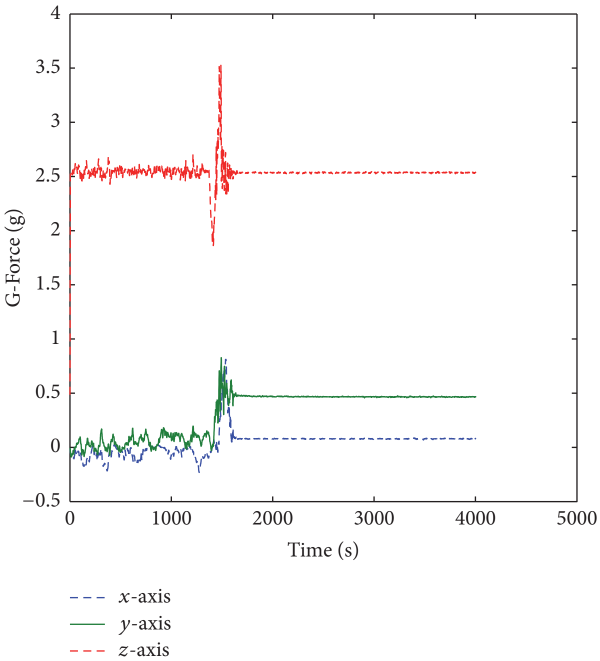


Figure 8: Detection of sudden falls by the user.

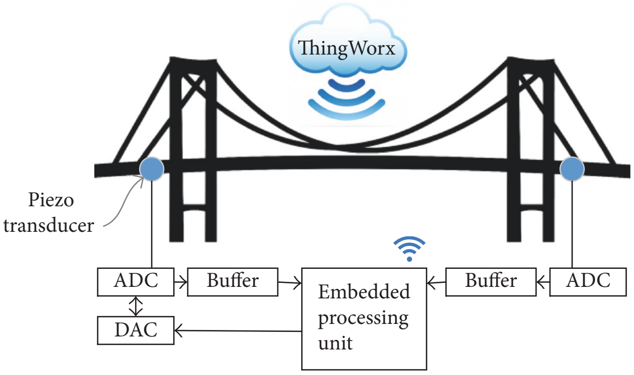
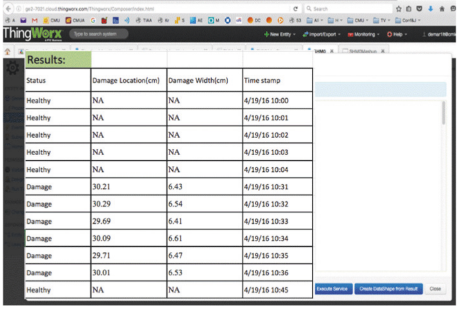


Figure 9: Architectural components of the prototype for SHM.

real time. Accordingly, ThingWorx application management cloud server was used to track the user location and presented on Google Maps as in Figure 13.

The passive RFID reader and built-in BLE transceivers collected and sent information on all tags and respective RSSI values through the embedded processor and gateway controller to ThingWorx application management server. This information was processed on the cloud server to identify the location, and an appropriate website address was sent back to the tour module. Accordingly, the tour module would open the website to play a video and provide the user with up to date information on the respective laboratory or classroom. Implemented in an academic environment, the presented system was able to localize with an accuracy of 0.73 m, play videos of respective labs, allow users to take pictures and videos, and helped them learn about the university.



Results:

Status	Damage Location(cm)	Damage Width(cm)	Time stamp
Healthy	NA	NA	4/19/16 10:00
Healthy	NA	NA	4/19/16 10:01
Healthy	NA	NA	4/19/16 10:02
Healthy	NA	NA	4/19/16 10:03
Healthy	NA	NA	4/19/16 10:04
Damage	30.21	6.43	4/19/16 10:31
Damage	30.29	6.54	4/19/16 10:32
Damage	29.69	6.41	4/19/16 10:33
Damage	30.09	6.61	4/19/16 10:34
Damage	29.71	6.47	4/19/16 10:35
Damage	30.01	6.53	4/19/16 10:36
Healthy	NA	NA	4/19/16 10:45

Figure 10: Damage detection results as available on cloud server.

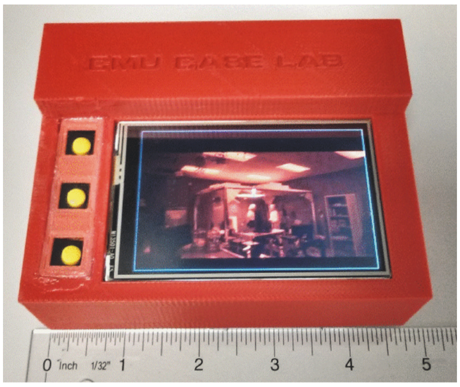


Figure 11: Handheld tour module.

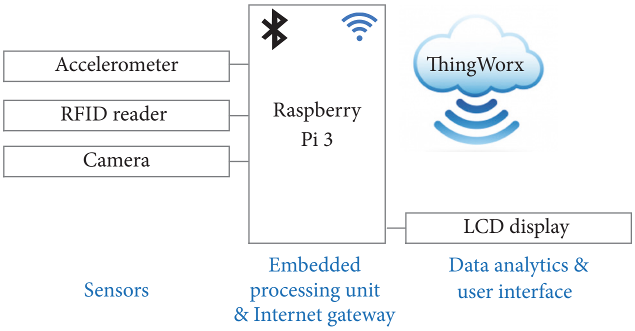
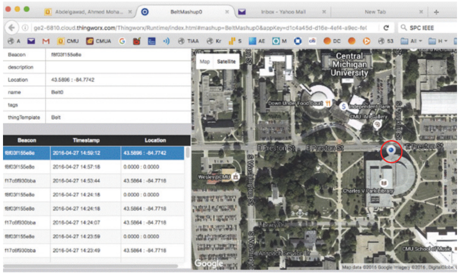

Accelerometer
RFID reader
Camera
Raspberry Pi 3
ThingWorx
LCD display
Sensors
Embedded processing unit & Internet gateway
Data analytics & user interface

Figure 12: Architectural components of the prototype for portable tour module.



Beacon	f8f03f155e8e
description	
Location	43.5896 : -84.7742
name	Belt0
tags	
thingTemplate	Belt

Beacon	Timestamp	Location
f8f03f155e8e	2016-04-27 14:59:12	43.5896 : -84.7742
f8f03f155e8e	2016-04-27 14:57:18	0.0000 : 0.0000
f17c6f930bba	2016-04-27 14:53:44	43.5864 : -84.7718
f8f03f155e8e	2016-04-27 14:24:18	0.0000 : 0.0000
f8f03f155e8e	2016-04-27 14:24:18	43.5864 : -84.7718
f17c6f930bba	2016-04-27 14:24:07	43.5864 : -84.7718
f8f03f155e8e	2016-04-27 14:23:59	0.0000 : 0.0000
f17c6f930bba	2016-04-27 14:23:49	43.5864 : -84.7718

Figure 13: Localization of user tagged on Google Maps through cloud server.

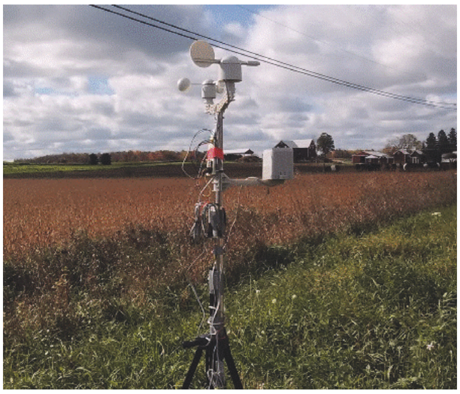


Figure 14: Weather sensing IoT system as utilized in a farm.

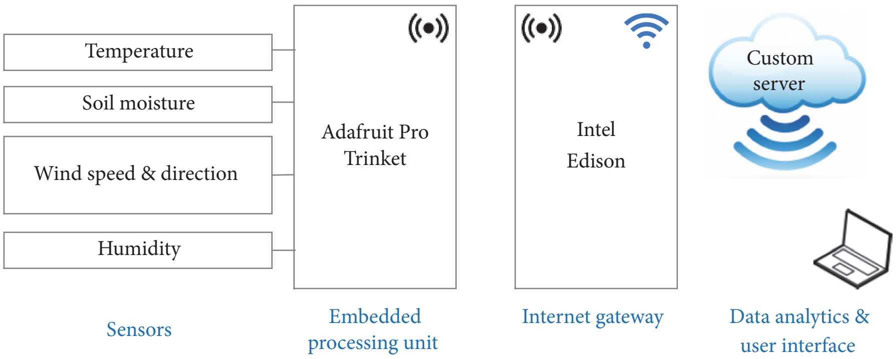


Figure 15: Architectural components utilized in agriculture system.

4.5. Precision Agriculture. Precision agriculture is one of the paradigms which can use the IoT advantages to optimize the production efficiency and uniformity across the agriculture fields, optimize the quality of the crops, and minimize the negative environmental impact. Accordingly, the proposed modular architecture has been implemented as presented in Figure 14, with its respective architectural components in Figure 15.

The sensors utilized in this application test are a temperature sensor, soil moisture sensor, wind speed and direction sensor, and a humidity sensor to enable for economical usage of natural resources. As rapid deployment of the application system is an important factor, Adafruit Pro Trinket has been chosen as the MCU, and nRF transceiver has been chosen to relay information to the gateway. As the gateway in this application had to collect information from multiple weather modules simultaneously and relay it to the application cloud server, a system with decent computational power and yet consumes low power was required. Accordingly, Intel Edison has been chosen to serve as the internet gateway. Data collected from the sensors required the application of complex custom algorithms to activate agricultural systems and often require user intuition. Accordingly, the cloud server analyzed this data in MATLAB and provided visual feedback on weather patterns to the user as in Figures 16 and 17.

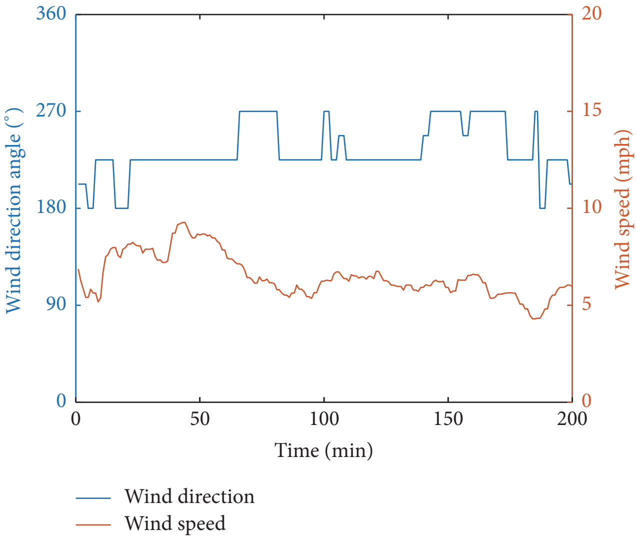


Figure 16: Tracking wind direction and speed via the custom application management cloud server.

5. Conclusions

IoT applications are now rapidly evolving. The primary purpose of IoT is to communicate with the surrounding

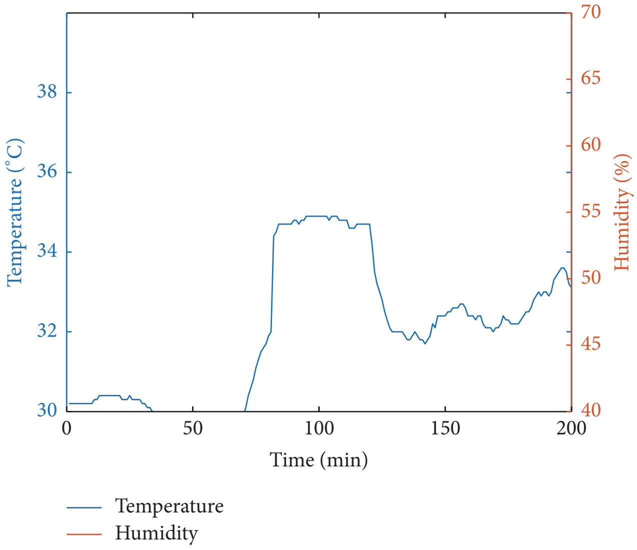


Figure 17: Tracking temperature and humidity via the custom application management cloud server.

environment through a multitude of sensors and actuators and facilitate the user to make constructive decisions. Much of the published work in IoT architecture is theoretical and is tied to a specific application. Addressing the challenges in the selection of computational devices, wireless connectivity, internet gateway, and application cloud server, this research presented a real-time, low-power, low-cost, and reliable modular IoT architecture to facilitate implementation in diverse applications.

The proposed work addressed the existing limitation by proposing a modular IoT architecture with a wide choice of subsystems that are readily available in the market, along with the advantages and limitations in each. Based on the application and its respective specification in consideration, the designer can review the architectural layer choices presented in Section 3, identify the respective module, and combine them to obtain a reliable IoT architecture. As all modules presented in this work are off-the-shelf components with open-source software available, the developer can build the software base and implement the IoT architecture seamlessly for practical development.

The proposed framework has been verified throughout real-time hardware implementation for different applications such as healthcare, wearables, structural health monitoring, object tracking, and connected vehicles. Further, research in this area is anticipated in the near future to address more applications.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been sponsored by the National Science Foundation under Grant no. 1542368.

References

[1] E. Theodoridis, G. Mylonas, and I. Chatzigiannakis, "Developing an IoT Smart City framework," in *Proceedings of the 4th IEEE International Conference on Information, Intelligence, Systems and Applications (IISA '13)*, pp. 1–6, IEEE, Piraeus, Greece, July 2013.

[2] M. Nakamura and L. Du Bousquet, "Constructing execution and life-cycle models for smart city services with self-aware IoT," in *proceedings of the 12th IEEE International Conference on Autonomic Computing (ICAC '15)*, pp. 289–294, July 2015.

[3] M. Li and H.-J. Lin, "Design and implementation of smart home control systems based on wireless sensor networks and power line communications," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4430–4442, 2015.

[4] B. Jiang and Y. Fei, "Smart home in smart microgrid: a cost-effective energy ecosystem with intelligent hierarchical agents," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 3–13, Jan. 2015.

[5] A. Anvari-Moghaddam, H. Monsef, and A. Rahimi-Kian, "Optimal smart home energy management considering energy saving and a comfortable lifestyle," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 324–332, 2015.

[6] S. K. Datta, C. Bonnet, A. Gyrard, R. P. Ferreira Da Costa, and K. Boudaoud, "Applying Internet of Things for personalized healthcare in smart homes," in *Proceedings of the 24th Wireless and Optical Communication Conference (WOCC '15)*, pp. 164–169, Taipei, Taiwan, October 2015.

[7] J. Li, W. Gu, and H. Yuan, "Research on IOT technology applied to intelligent agriculture," in *Proceedings of the 5th International Conference on Electrical Engineering and Automatic Control*, vol. 367, pp. 1217–1224, July 2016.

[8] N. Sales, O. Remedios, and A. Arsenio, "Wireless sensor and actuator system for smart irrigation on the cloud," in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT '15)*, pp. 693–698, Milan, Italy, December 2015.

[9] M. Maksimović, V. Vujović, and B. Perišić, "A custom Internet of Things healthcare system," in *Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI '15)*, Aveiro, Portugal, June 2015.

[10] S. Anjana, M. N. Sahana, S. Ankith, K. Natarajan, K. R. Shobha, and A. Paventhan, "An IoT based 6LoWPAN enabled experiment for water management," in *Proceedings of the 9th IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS '15)*, Kolkata, India, December 2015.

[11] C. Jiacheng, Z. Haibo, Z. Ning, Y. Peng, G. Lin, and S. Xuemin, "Software defined Internet of vehicles: architecture, challenges and solutions," *Journal of Communications and Information Networks*, vol. 1, no. 1, pp. 14–26, 2016.

[12] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in internet of vehicles: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339–2352, 2015.

[13] K. M. Alam, M. Saini, and A. El Saddik, "Toward social internet of vehicles: concept, architecture, and applications," *IEEE Access*, vol. 3, pp. 343–357, 2015.

[14] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[15] S. Krco, B. Pokric, and F. Carrez, "Designing IoT architecture(s): a European perspective," in *Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 79–84, Seoul, South Korea, March 2014.

[16] M. F. Al Faisal, S. Bakar, and P. S. Rudati, "The development of a data acqusition system based on internet of things framework," in *Proceedings of the 2014 International Conference on ICT for Smart Society (ICISS '14)*, pp. 211–216, Bandung, Indonesia, September 2014.

[17] A. Atabekov, J. He, and P. O. Bobbie, "Internet of things-based framework to facilitate indoor localization education," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC '16)*, pp. 269–274, Atlanta, Ga, USA, June 2016.

[18] L. Hu, R. Sun, F. Wang, X. Fei, and K. Zhao, "A Stream processing system for multisource heterogeneous sensor data," *Journal of Sensors*, vol. 2016, Article ID 4287834, 8 pages, 2016.

[19] J. Granados, A.-M. Rahmani, P. Nikander, P. Liljeberg, and H. Tenhunen, "Towards energy-efficient HealthCare: an Internet-of-Things architecture using intelligent gateways," in *4th International Conference on Wireless Mobile Communication and Healthcare (MOBIHEALTH '14)*, pp. 279–282, November 2014.

[20] F. Zhang, M. Liu, Z. Zhou, and W. Shen, "An IoT-based online monitoring system for continuous steel casting," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1355–1363, 2016.

[21] L. Chen, K. Yang, and X. Wang, "Robust cooperative Wi-Fi fingerprint-based indoor localization," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1406–1417, 2016.

[22] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in IoT-based information system for emergency medical services," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.

[23] N. B. Krishnan, S. S. S. Sai, and S. B. Mohanthy, "Real time internet application with distributed flow environment for medical IoT," in *Proceedings of the 1st International Conference on Green Computing and Internet of Things (ICGCIoT '15)*, pp. 832–837, Noida, India, October 2015.

[24] T. Gomes, J. Brito, H. Abreu, H. Gomes, and J. Cabral, "GreenMon: an efficient wireless sensor network monitoring solution for greenhouses," in *2015 IEEE International Conference on Industrial Technology, ICIT 2015*, pp. 2192–2197, esp, March 2015.

[25] F. Zafari and I. Papapanagiotou, "Enhancing iBeacon based micro-location with particle filtering," in *58th IEEE Global Communications Conference, GLOBECOM 2015*, usa, December 2015.

[26] F. Subhan, H. Hasbullah, and K. Ashraf, "Kalman filter-based hybrid indoor position estimation technique in bluetooth networks," *International Journal of Navigation and Observation*, vol. 2013, Article ID 570964, 13 pages, 2013.

[27] M. S. Aman, H. Jiang, C. Quint, K. Yelamarthi, and A. Abdelgawad, "Reliability evaluation of iBeacon for micro-localization," in *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON '16)*, October 2016.

[28] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, "Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 63–72, 2014.

[29] H. Zhang, J. Guo, X. Xie, R. Bie, and Y. Sun, "Environmental effect removal based structural health monitoring in the internet of things," in *Proceedings of the 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '13)*, pp. 512–517, July 2013.

[30] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Deploying wireless sensor networks with fault-tolerance for structural health monitoring," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 382–395, 2015.

[31] A. Myers, M. A. Mahmud, A. Abdelgawad, and K. Yelamarthi, "Toward integrating Structural Health Monitoring with Internet of Things (IoT)," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, pp. 438–441, Grand Forks, ND, USA, May 2016.

[32] C. Lazo, P. Gallardo, and S. Céspedes, "A bridge structural health monitoring system supported by the internet of things," in *2015 IEEE Colombian Conference on Communications and Computing, COLCOM 2015*, col, May 2015.

[33] Y. Gu and F. Ren, "Energy-efficient indoor localization of smart hand-held devices using bluetooth," *IEEE Access*, vol. 3, pp. 1450–1461, 2015.

[34] Y. Gu, L. Quan, F. Ren, and J. Li, "Fast indoor localization of smart hand-held devices using bluetooth," in *10th IEEE International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2014*, pp. 186–194, usa, December 2014.

[35] K. Yelamarthi, D. Haas, D. Nielsen, and S. Mothersell, "RFID and GPS integrated navigation system for the visually impaired," in *53rd IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2010*, pp. 1149–1152, usa, August 2010.

[36] K. Yelamarthi and K. Laubhan, "Navigation assistive system for the blind using a portable depth sensor," in *IEEE International Conference on Electro/Information Technology, EIT 2015*, pp. 112–116, usa, May 2015.

[37] B. Olszewski, S. Fenton, B. Tworek, J. Liang, and K. Yelamarthi, "RFID positioning robot: an indoor navigation system," in *2013 IEEE International Conference on Electro/Information Technology, EIT 2013*, usa, May 2013.

[38] J. Beckwith, S. Sherbrook, R. Lefief, M. Williams, and K. Yelamarthi, "An RFID based autonomous indoor tour guide robot," in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems, MWSCAS 2012*, pp. 562–565, usa, August 2012.

[39] P. Kolios, C. Panayiotou, G. Ellinas, and M. Polycarpou, "Data-driven event triggering for IoT applications," *IEEE Internet of Things Journal*, 2016.

[40] R. K. Balan, A. Misra, and Y. Lee, "LiveLabs: Building an in-situ real-time mobile experimentation testbed," in *15th Workshop on Mobile Computing Systems and Applications, HotMobile 2014*, usa, February 2014.

[41] A. S. Tonneau, N. Mitton, and J. Vandaele, "How to choose an experimentation platform for wireless sensor networks? A survey on static and mobile wireless sensor network experimentation facilities," *Ad Hoc Networks*, vol. 30, pp. 115–127, 2015.

[42] M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani, "Architecting the internet of things: state of the art," in *Robots and Sensor Clouds*, pp. 55–75, Springer, 2016.

[43] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: integrating internet of things and cloud computing and the issues involved," in *2014 11th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2014*, pp. 414–419, pak, January 2014.

[44] J. Pacheco, S. Satam, S. Hariri, C. Grijalva, and H. Berkenbrock, "IoT security development framework for building trustworthy smart car services," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pp. 237–242, Tucson, AZ, USA, September 2016.

[45] Analog Devices Inc., "Digital Accelerometer," ADXL345 datasheet, 2009.

[46] Measurement Specialties, "HTU21D(F) sensor digital relative humidity sensor with temperature output," HTU21D datasheet, 2013.

[47] Sensirion, "Humidity and Temperature Sensor IC," SHT10, SHT11, SHT15 datasheet, 2011.

[48] Camera Module, "*Raspberry* Pi," https://www.raspberrypi.org/products/camera-module.

[49] MPR 121, "Proximity capacitive touch sensor controller," MPR 121: Proximity Capacitive Touch Sensor Controller, *Freescale Semiconductor Inc*, http://www.nxp.com/pages/proximity-capacitive-touch-sensor-controller:MPR121.

[50] Microsoft, "Meet kinect for windows," https://developer.microsoft.com/en-us/windows/kinect.

[51] Tekscan, "FlexiForce Standard Model A201," A201 datasheet.

[52] Apple, "iBeacon for Developers," https://developer.apple.com/ibeacon.

[53] Sparkfun, "CdS photoconductive cells," GL5528 datasheet.

[54] Unique-Bargains, "HC-SR501 PIR motion detector," HC-SR501 datasheet, 2016.

[55] TAOS, "Color light-to-digital converter with IR filter," TCS3472 datasheet, 2012.

[56] DFRobot, "Moisture Sensor (SKU:SEN0114)," https://www.dfrobot.com/wiki/index.php/Moisture_Sensor_(SKU:SEN0114).

[57] Sparkfun, "SparkFun Sound Detector," https://www.sparkfun.com/products/12642.

[58] Elec Freaks, "Ultrasonic ranging module HC - SR04," HC-SR04 datasheet, 2016.

[59] Pulse and Oxygen in Blood Sensor (SPO2) for e-Health Platform, "Cooking hacks," https://www.cooking-hacks.com/pulse-and-oxygen-in-blood-sensor-spo2-ehealth-medical.

[60] Body Temperature Sensor for e-Health Platform, "*Cooking Hacks*," https://www.cooking-hacks.com/body-temperature-sensor-ehealth-medical.

[61] Piezo Systems Inc., "Standard quick-mount bending generator," Piezoelectric sensor datasheet, 2011.

[62] Intersil, "6-Bit, 15 MSPS, Flash A/D Converters," CA3306, CA3306A, CA3306C datasheet, 2002.

[63] Microchip Technology Inc., "12-bit digital-to-analog converter with EEPROM memory in SOT-23-6," MCP4725 datasheet, 2007.

[64] Arduino Uno, *Arduino*, https://www.arduino.cc/en/Main/ArduinoBoardUno.

[65] Arduino Pro Mini, *Arduino*, https://www.arduino.cc/en/Main/ArduinoBoardProMini.

[66] Nucelo-F401RE Board, "ST microelectronics," http://www.st.com/en/evaluation-tools/nucleo-f401re.html.

[67] ATtiny10, Atmel Corporation, http://www.atmel.com/devices/attiny10.aspx.

[68] CC2640 SimpleLink™ Bluetooth® Smart Wireless MCU, *Texas Instruments*, http://www.ti.com/product/CC2640.

[69] "Texas Instruments MSP430F2410 16-bit Ultra-Low-Power Microcontroller," http://www.ti.com/product/MSP430F2410.

[70] Adafruit Pro Trinket-3V 12MHz, http://www.adafruit.com/products/2010.

[71] Adafruit Flora - Miniature wearable electronic platform, https://www.adafruit.com/products/659.

[72] Atmel, "Atmel 8-bit AVR microcontroller with 2/4/8K bytes in-system programmable flash," http://www.atmel.com/Images/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet-Summary.pdf.

[73] M. Ghamari, H. Arora, R. S. Sherratt, and W. Harwin, "Comparison of low-power wireless communication technologies for wearable health-monitoring applications," in *2nd International Conference on Computer, Communications, and Control Technology, I4CT 2015*, pp. 1–6, mys, April 2015.

[74] Bluetooth Technology Basics, Bluetooth, https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics.

[75] P. Smith, *Comparing Low-Power Wireless Technologies*, Digi-Key, Septemper 2011.

[76] CC2630 SimpleLink™ 6LoWPAN and ZigBee® Wireless MCU, "Texas Instruments," http://www.ti.com/product/cc2630.

[77] IrDA IrSimple™ Specifications, "Infrared Data Association," http://irdajp.info/irsimple.html.

[78] nRF24AP2 Single-Chip ANT™ ultra-low power wireless network solution Datasheet, *Nordic Semiconductor*, June 2010.

[79] Intel® Edison Development Platform, "Intel® Corporation," http://download.intel.com/support/edison/sb/edison_pb_331179002.pdf.

[80] Raspberry Pi 3 Model B, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[81] BeagleBoard, "BeagleBone black," https://beagleboard.org/black.

[82] AmeriDroid, "ODROID-XU4 [0007A]," http://ameridroid.com/products/odroid-xu4.

[83] Arduino, "Arduino Yun," https://www.arduino.cc/en/Main/ArduinoBoardYun.

[84] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim, "TinyREST—a protocol for integrating sensor networks into the internet," in *Proceedings of the Real-World Wireless Sensor Networks (REALWSN '05)*.

[85] Y. Zhou, S. De, W. Wang, and K. Moessner, "Search techniques for the web of things: a taxonomy and survey," *Sensors*, vol. 16, no. 5, article 600, 2016.

[86] Evrythng, IoT Smart Products Platform, 2016. https://evrythng.com.

[87] ThingWorx, 2016, https://www.thingworx.com/.

[88] Paraimpu, 2016, https://www.paraimpu.com/.

[89] Sensorpedia, Oak Ridge National Laboratory, 2016. http://www.sensorpedia.com.