International Journal of Research in Advent Technology, Vol.7, No.4, April 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Embedded Development Platforms To Design Prototypes Of Internet Of Things (Iot) Applications: A Study

Ayaskanta Mishra[1]
Assistant Professor, School of Electronics Engineering[1], Kalinga Institute of Industrial Technology (KIIT) Deemed to be University, Bhubaneswar, Odisha, India[1]
Email: ayaskanta.mishrafet@kiit.ac.in[1]

Abstract- In this paper a comprehensive study of different development platforms for Internet of Things (IoT) application are surveyed and a comparative analysis is presented. This paper reviews diverse spectrum of development platforms and a comparative analysis of different technical aspects is discussed. The major focus of this review paper is to give a quantitative analysis of embedded platforms with bare metal and Linux environment and their network connectivity to cloud analytics platforms. This review has considered some popular prototyping platforms like: Arduino UNO, Node MCU, Raspberry Pi3, Texas Instrument's Beaglebone Black, Intel Galileo, XBEE® as well as some industry grade platforms like cypress Semiconductor's PSoC4 BLE and microchip's SoC PIC32 and SAM series embedded platforms. Comparative analysis is based on technical specifications, computation power, cost, reliability, learning curve and time to market.

Keywords- IoT; prototyping; Arduino; NodeMCU; Raspberry Pi; Intel Galileo; XBee; PSoC; PIC32

1. INTRODUCTION

Internet of Things (IoT) is interconnection of cyber-physical objects for seamless data acquisition from sensors and at the same time controlling actuators and driving motors for any mechanical actions in physical space. The interconnections for such smart objects are done over a generic global network or "Internet". For enabling the decision making based on the data collected from cyber physical objects of human users the database and computing can be done in cloud or a server deployed in the internet. Communication aspect of IoT is taken care by the TCP/IP protocol stack, which is the backbone of Internet. As discussed in this context; smarter cyber physical objects like sensors, motors or actuators can be interfaced with intelligent devices or motes which are seamlessly connected through internet. In this paper, a comprehensive review has been done on various such popular development platforms prototyping of any IoT applications.

Developing an IoT application requires four major building blocks namely 1) Sensors and Actuators 2) MCU MPU Development boards 3) Network and protocols 4) Cloud and data analytics. These can be referred as four pillars of IoT. These four pillars of IoT can be described as given in figure 1. "Sensors" are Transducers, which converts physical parameters like Temperature, Pressure, Humidity, and Gas Concentration into Electrical parameters like Voltage/ Current. The physical movement is handled by a machine component called Actuator. Actuators requires two types inputs one is control input, which is relatively low power and the second one is the power input, provides the required energy to operate. The second one can be electric voltage or current, pneumatic or hydraulic pressure, or human power. A microprocessor is the computation engine - CPU (central processing unit) that is fabricated on a single chip. Microcontrollers are low power devices specially housing CPU, Memory and I/O in single chip especially designed for interfacing application or single chip computer. System on Chip (SoC) is modern approach to develop the whole computer system on a single chip including network adaptor except the power circuit and antenna outside the chip. Many modern prototyping platforms like Raspberry Pi, Intel Galileo, Cypress PSoC4 and Microchip platforms use above SoC technology.

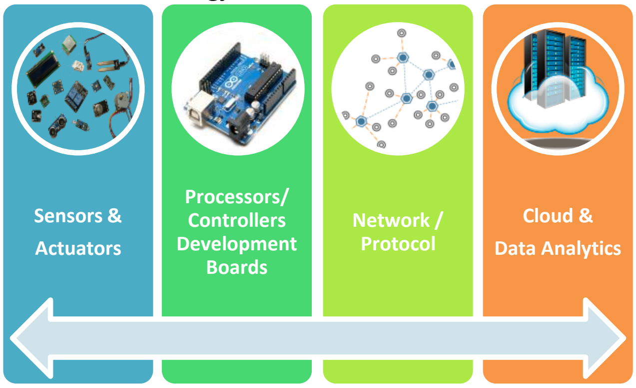


Fig.1: Four pillars of IoT

Network and protocol for IoT can be application specific it depends on communication requirement of the IoT application.WAN, LAN and PAN networks can be used for IoT depending upon range, data-rate and power constraints. For long range applications LoRaWAN is the latest and gaining a lot of popularity. Apart for this conventional WAN like LTE, GSM, GPRS can be used. In case of LAN both IEEE 802.3 Ethernet for wired applications and IEEE 802.11 WLAN for wireless application can be used. For power constraint applications with limited range WPAN technologies like Bluetooth, BLE and Zigbee can be used based on application requirements. The final building block is the cloud which is nothing but an Internet Server for database as well as analytics and visualization of data and application preferably on a web interface GUI. Various protocols are supported for such application layer servers like HTTP, MQTT and CoAP.

This paper is organized in five chapters. Section 1 describes the introduction to IoT prototyping aspects. Section 2 reviews various prototyping development platforms. Section 3 shows the comparative analysis. Section 4 gives a recommendation for selecting appropriate platforms for developing IoT applications based on specific scenarios. Finally Section 5 concludes the paper.

2. A REVIEW ON IOT DEVELOPMENT PLATFORMS

Developing an IoT application requires a smart motes or nodes. These smart devices have some computational intelligence. Such nodes can be programmed and interfaces with various sensors as well as actuators. These nodes can be interconnected using Internet enabled by TCP/IP protocol stack and connected to a centralized cloud server for various application specific scenarios. For developing and prototyping such smart embedded devices various industry standard development boards can be used. The following are some standard and popular embedded system prototyping platforms for development of IoT applications. A detailed comparison of Arduino [1], raspberry pi and ESP8266 is presented by authors [2].

2.1. Arduino UNO

Arduino UNO is a popular embedded platform with an added feature of being open source. Arduino is an ATmega328 8-bit 20 MHz micro controller at its core. Arduino is a very low end platform in terms of technical specifications for developing IoT applications prototypes. Figure 2 shows a Arduino UNO R3 Board.

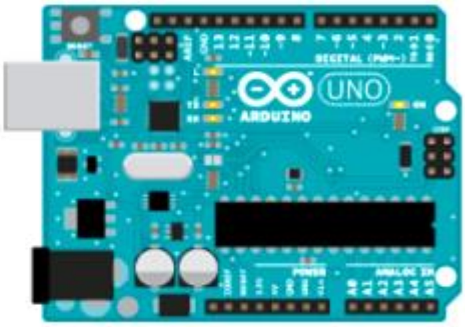


Fig.2: Arduino UNO R3 prototyping board

Arduino is at the top position in terms of popularity mostly because of it is easy to learn and use. In addition to that a lot of online resource and support forms and communities are available for Arduino. Arduino has its own software development platform called Arduino IDE and ‘C’ programming is used with void setup() and void loop() two main program structure.

2.2. Node MCU

Espressif Systems has developed a firmware of ESP family. Authors have presented a literature [3]. There are many Wireless LAN modules in this family namely ESP-01, ESP-12 etc. NodeMCU is based on the ESP8266 Wireless LAN (IEEE 802.11) firmware. Since its inception NodeMCU has gained a lot of popularity among developers because of two major reasons: firstly it is ESP-12 based device having very small form factor with inbuilt Wi-Fi and secondly it has good software support with LUA scripting language and Arduino IDE based c programming support. [3] LUA scripting language support for NodeMCU is from eLUA project and built on Espressif Non-OS SDK for ESP8266. Various open source projects like lua-cjson and spiffs are some added advantage of NodeMCU. Figure 3 shows a NodeMCU module. The technical specifications are given in Espressif literature [4].

Fig. 3: NodeMCU prototyping board with Espressif ESP8266 Wi-Fi SoC

2.3. Raspberry Pi

Raspberry Pi is a credit card size single board computer. Unlike above two discussed embedded platform Raspberry Pi is a full-fledged minicomputer with all the standard peripheral of a PC. A review has been presented by authors in their paper [5]. Latest Raspberry Pi has four USB ports which support all standard peripherals like keyboard, mouse, modems and dongles. In addition to that it has a HDMI port for connecting to display device like Monitor or TV. Raspberry Pi 3 Model B+ is powered by a Broadcom SoC with clock speed of 700 MHz to 1.4 GHz; on-board memory ranges from 256 MB to 1 GB RAM. Figure 4 shows a Raspberry Pi3 Single board Computer. A comparative analysis of technologies is presented in paper [6]. Power consumption of raspberry pi is presented in literature [7].

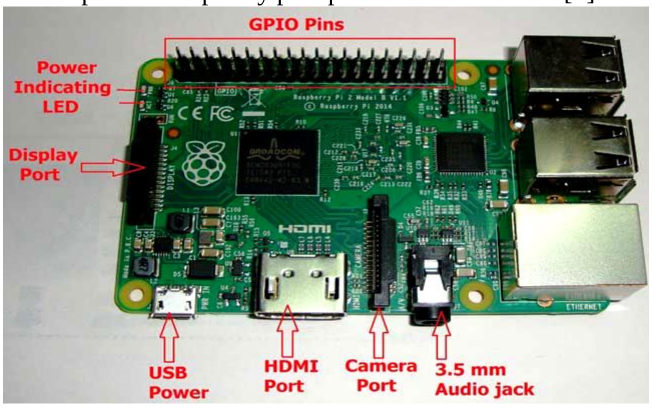


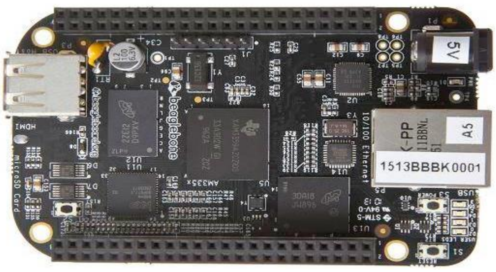
Fig. 4: Raspberry Pi 3 prototyping board with Broadcom SoC

The operating system and program memory are stored in Secure Digital (SD) card in either SDHC (early Raspberry Pi's) or MicroSDHC (Later Raspberry Pi's) sizes. The SoC is ARM based and house both CPU and GPU for processing. The latest Raspberry Pi has inbuilt Wi-Fi (IEEE 802.11) as well as Bluetooth 4.0 (IEEE 802.15.1) with Bluetooth Low Energy (BLE) support. Raspberry Pi also has extensive 40 pins General purpose input output (GPIO) with UART, SPI and I2C support. In addition to all these it has a 3.5 mm audio jack for audio and signal processing applications. Prices of Raspberry Pis range from US$5 to $35. Pi works on a Linux based OS platforms. Python is the most popular language for Raspberry pi developers, though Pi can support mostly all kind of programming language as it is a Linux based minicomputer. A case study of Restful framework in Raspberry Pi performance and energy overview is presented in the paper [8].

2.4. BeagleBone Black

Texas Instruments is a key player in embedded development platform. The latest version of BeagleBone series is BeagleBone Black. The BeagleBone is an open source platform for prototyping of IoT applications. The platform has a added advantage of being low-cost and low-power. The BeagleBone Series is a venture by joint collaboration between TI, Digi-Key and element14. The BeagleBone is a single-board computer. Texas Instrument's OMAP3530 system-on-a-chip is the core of BeagleBone boards. Figure 5 shows a Texas Instruments BeagleBone Black Single board Computer. The book has a excellent discussion of various aspect of BeagleBone Black [9].

]



The platform supports open source software development for scenario specific IoT application prototyping and development.

Texas Instrument’s ARM Cortex A8 based Sitara XAM3359AZCZ100 processor is a lower-cost, high-expansion SoC being used in BeagleBone boards. Sitara base BeagleBone boards are analogous to the BeagleBone Black however some features are tweaked, which are more relevant to modern age application development. A review on BeagleBone is presented in the paper [10].

2.5. Intel® Galileo Gen2

Intel® also has a product in IoT prototyping market called Intel® Galileo Family namely Gen1 and Gen2. The latest one being Galileo Gen2. Authors of [11] have presented a review on Intel Galileo. The Galileo is based on Intel x86 architecture and it is first in line of Arduino-certified development boards. This board is developed by Intel keeping the research & academia in mind. The Galileo boards are popularly referred in the community as “Breakout boards”. The main features of Intel Galileo boards are low-power and small-core. The Galileo boards are powered by Intel® Quark SoC X1000 the first product from Intel Quark family. This product is developed in Ireland to compete within Internet of Things market segment especially edge nodes of clouds to facilitate applications like smart wearable computing devices. Some key technical specifications of Quark SoC X1000 are 32-bit, Single-core, single-thread, Pentium (P54C/i586) architecture. Quark is an instruction set architecture (ISA) compatible CPU. The operating clock frequency is up to 400 MHz. Quark SoC is a counterpart from Intel® to the ARM based platform which is typically very popular in smart phones and other single-board computers. Author has presented an experimental analysis on Galileo for IoT [12]. The author of [13] has presented Galileo for beginners. Figure 6 shows an Intel® Galileo Gen2 development board.

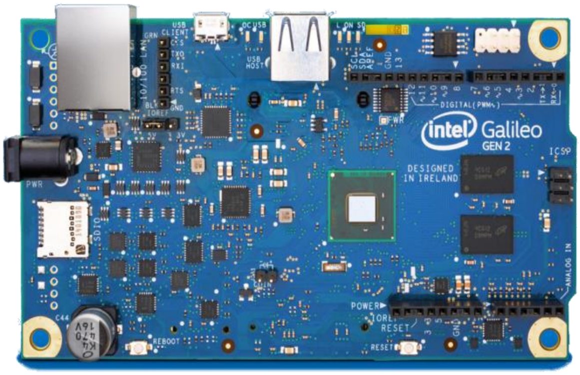


Fig.6: Intel Galileo Gen2 development board with Intel Quark X1000 SoC

The Galileo is more powerful when it comes to system specifications. A clock speed of 400MHz equipped with 256 MB of DDR3 RAM and 8 MB flash memory. The Galileo can outperform any Arduino device when it comes to computing. The Arduino platform are quite low end platform for an example, Arduino Mega is mere 16 MHz, 8 KB RAM and 256 KB flash memory. Rather Intel® Galileo boards are more comparable with other Single board computer platforms like Raspberry Pi Family or BeagleBone family. The author of [14] has given the technical details on Galileo. Raspberry Pi based devices are more powerful than the Galileo Series however RPi lacks in one front only that it does not have any flash memory rather has a SD Card/ MicroSD card slot for OS and Program memory requirements. Intel® Galileo support Linux OS based yocto distributions. There are some useful resources available by various authors. The author of [15] has given a starter’s guide for Galileo, [16] has written on programming aspects of Galileo and [17] has presented on essentials of Galileo.

2.6. Digi XBee®

Digi XBee radio modules are for LR-WPAN applications mostly used for low power Wireless Sensor Networks. XBee got its name from Zigbee (IEEE 802.15.4). The authors of [18] have presented Wireless Sensing using XBee. First lunched in 2005 by MaxStream brand is based on IEEE 802.15.4-2003 standard. XBee modules are designed as point-to-point and works on star topology with baud rates of 250 kbps. There are two types of models for XBee. Authors of [19] have presented a review on XBee. First one is cheap 1mW low power modules and second one is 100 mW XBee-PRO modules. As the number of XBee deployment grows new ecosystem of XBee gateways, adaptors and software has evolved. XBee radio modules are low power and used mostly in a Mesh scenario for low data rate sensor applications. Authors have presented a research article on WSN using ARM [20]. The common features of XBee modules are UART, Power management, flow control, Digital I/O and Analog to Digital Converter (ADC) built in. Field performance analysis IEEE 802.15.4 XBee is presented in [21]. Performance analysis of XBee based WSN is presented in [22]. Wireless Mesh Networking with XBee is presented by authors in [23].

The programmable XBee have an additional on-board processor for user's code. The programmable XBee and a surface-mount version of the XBee radios were both introduced in 2010. XBee support XCTU software framework and Zigbee packet can be captured through UART. Figure 7 shows XBee radio module. Application of

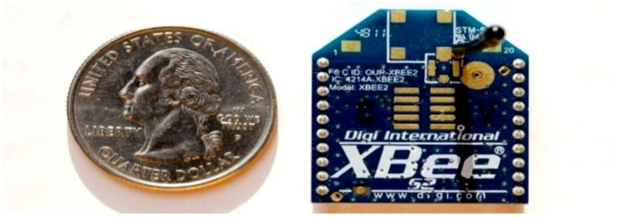


Fig.7: Digi XBee IEEE 802.15.4 form factor compatible radio modules

Zigbee compatible XBee and XBeePro is presented in paper [24]. Design of WSN based on Zigbee is presented by author in [25].

2.7. Cypress PsoC4 BLE

Cypress CY8CKIT-143A PSOC® 4 Bluetooth Low Energy Pioneer Kit 256K Kit provides designers certified 45x27x2mm, easy-to-use solution for creating a complete Bluetooth® Low Energy (BLE) system. The Kit consists of PsoC 4 Bluetooth Low Energy device with internal flash of 256KB, 24MHz and 32.768 kHz crystals, a PCB antenna, and other passives. The CY8CKIT-143A provides easy access to all GPIOs on the device. This Io Development Kit is very easy to use; designers can either plug the module into the CY8CKIT-042-BLE Kit or use the module with CY8CKIT-002 MiniProg3 (an external programmer not included in Kit). PSoC creator is the software to design embedded application for Cypress SoCs. Figure 8 shows the CY8CKIT-143A PSOC® 4 BLE Module. Power consumption analysis of Bluetooth Low Energy product is presented in the research work in [26]. A security system design using BLE is presented in [27]. Authors have presented experimental characterization of low power device for IoT applications in [28].

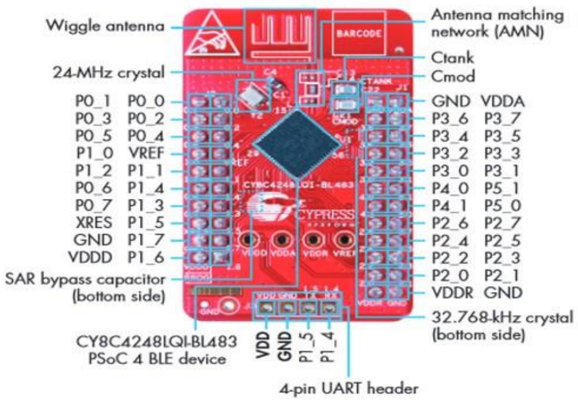


Fig.8: Cypress CY8CKIT-143A PSOC® 4 BLE Module

2.8. Microchip PIC32 curiosity IoT development board

Microchip PIC32 curiosity IoT Development board from Microchip Technology is for developing Amazon FreeRTOS-based applications. The Amazon FreeRTOS Curiosity PIC32MZ EF Bundle DM320104-BNDL has a Curiosity PIC32MZ EF IoT development board, a Wi-Fi board and an USB UART click board that is used for developing an AWS cloud-connected application. An ultra low power digital architecture for IoT is presented by authors in their research [29]. This bundle includes a LAN8720A PHY daughter board to create Ethernet-connected applications. Authors have presented a programming aspect of PIC32 in [30].

Fig.9: Microchip PIC32 curiosity IoT development board with 32-bit MCU

Amazon FreeRTOS is a microcontroller operating system that makes small, low-powered edge devices. This bundle makes it easy for designers to develop, deploy, secure, and maintain IoT Applications. Amazon FreeRTOS is popular open source operating system based on FreeRTOS, for microcontrollers, and includes inbuilt software libraries to securely connect devices locally to AWS Greengrass, and directly to the cloud, and update remotely. The high-performance PIC32MZ EF MCUs host the Amazon FreeRTOS and run at up to 415 DMIPs with industry-leading connectivity options including 10/100 Ethernet MAC, Dual CAN and Hi-Speed USB, ample Flash memory of up to 2 MB, rich peripherals, and a robust tool chain which empower embedded designers to build complex applications rapidly. Figure 9 shows the Microchip PIC32 curiosity development board. The authors have presented embedded computing and mechatronics using PIC32 in [31]. Further a design of secure and energy efficient embedded system for future Internet applications is presented in [32].

2.9. Microchip SAM Series IoT Dev Kit

Microchip Zero Touch Secure Provisioning Kit is an IoT development Dev Kit that helps designer to develop quick and secure IoT devices those are complaint with the AWS security regulations. AWS has a strict mutual authentication requirement between device and the remote servers of AWS cloud. A robust authentication must be in place to ensure a complete safe guarding of system credentials such as private keys from the application core to avoid leaving backdoors opened to software loop holes. In addition, the software is as secure as the user's skill set is in security. Human users and software can often be one of the easiest targets for a hacker as they are the least reliable elements. Incorporating Microchip pre-configured ATECC508-MAHAW or ATECC508ASSHAW Crypto Authentication devices into a system is a very secure method to connect to the AWS IoT service. It leaves the whole handling of certificate and private key manipulation to Microchip secure provisioning factories in addition to keeping credentials away from software and users. The ATECC508A and ATECC608A devices in the kit are generic devices. Starting with the upgraded Zero Touch Provisioning Kit for AWS IoT Version B and benefit from the new configuration and provisioning scripts (Python based) and AWS IoT account configuration scripts (using Cloud formation). Figure 10 shows Microchip IoT development kit with hardware cryptography and AWS cloud support. Microchip product specifications as given in company web resource [33].

Fig.10: Microchip IoT Development kit with hardware cryptography and AWS support

This version B of the kit comes with an easier on boarding process to generate certificates and provision them into the Crypto Authentication device using Python scripts. In addition, the user will have access to a Cloud Formation script to generate a web UI reflecting the I/O of the kit and utilize it as a foundation to develop virtually any sensor based use cases. In addition to the ATECC508A and ATECC608A devices, the kit includes a Cortex-M4 ATSAMG55 and Wi-Fi ATWINC1500 using FreeRTOS and the ATWINC1500 integrated TLS stack. The IoT development kit can be programmed using c language using the ATMEL studio. This is the latest SAM series product from ATMEL after the acquisition by Microchip. The authors have presented a study on efficient power consumption wireless communication techniques for IoT application in [34].

3. COMPARATIVE ANALYSIS

In chapter 2, we have seen nine popular IoT prototyping development platforms in brief. In this chapter a comparative analysis is done based on seven aspects of IoT application prototyping. First one being the comparison of technical specifications of all the above discussed development platforms. In the later part of the chapter we can see the comparison of these nine development platforms based on computational power, cost, reliability, learning curve, OS support and programmability and time to market.

3.1. Technical Specification

The comparison of technical specifications is done based on CPU, RAM, Memory, I/O, Network Connectivity, OS and programming aspects of the all the above mentioned nine IoT application development platforms.

3.2. Computational power

The following Figure.11 shows the comparative analysis of the computational power of the mentioned IoT development platforms in scale of relative grading based on CPU, RAM.

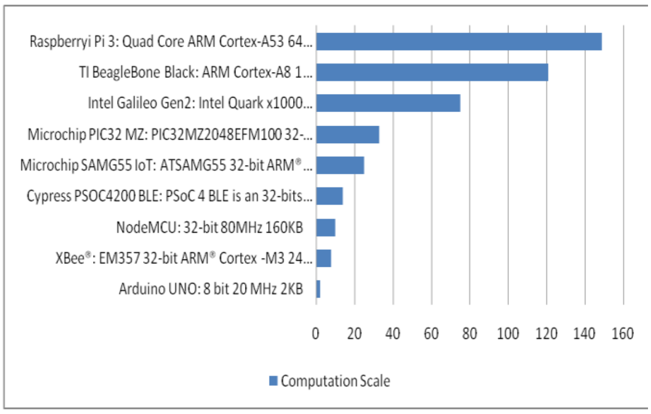


Fig. 11 Computational Power of IoT development platforms

Table 1. Technical Specification

Platform	CPU/MCU/SoC	RAM	Memory	I/O	Network Connectivity	OS/ Programming
Arduino	ATmega328 8-bit 20 MHz	2KB	Flash Memory 32KB, EEPROM 1KB	DIO: 14 with 6 PWM, Analog: 6, I2C, SPI,UART	External Shield WLAN, BLE, WAN	Bootstrap/ Arduino IDE
NodeMCU	Xtensa® 32-bit 80 MHz	160KB	Flash Memory 16MB	GPIO: 17, 8 PWM, I2C, ADC:10 bits, SPI, UART	WLAN 802.11 b/g/n HT ESP8266 inbuilt	XTOS/ Arduino IDE C++, Lua, lua-cjson, and spiffs
Raspberry Pi 3*	Broadcom BCM2837. 4× ARM Cortex-A53, 1.2GHz. GPU: BroadcomVideoCoreIV	1GB LPDDR2 (900 MHz)	MicroSD	40-pin header, populated, I2C, SPI,UART, USB	10/100Ethernet, WLAN802.11n, Bluetooth 4.1, BLE	Linux, Raspbian/ C++, Python, Java…
BeagleBone Black	TI DM3730 Processor - 1 GHz ARM Cortex-A8 core. PowerVR SGX 2D/3D graphics processor	512 MB LPDDR RAM	MicroSD	4xUART, 8× PWM, LCD, GPMC, MMC1, 2× SPI, 2× I²C, A/D Converter, 2× CAN Bus, 4× Timers	10/100 Ethernet IEEE 802.11n (WLAN Version)	Linux, Android, Windows, Embedded/ C, C++, Python, Perl, Ruby, Java, Shell Script
Intel Galileo Gen2	Intel Quark SoC x1000 400 MHz	256 MB DDR3 RAM	Flash Memory 8M, EEPROM 8 kb, Micro SD card slot up to 32GB	DIO: 20 with 6 PWM, Analog: 6, I2C, SPI,UART, USB host & client	10/100 Ethernet, External: Intel Centurion Wi-Fi 802.11 b/g/n	Yocto 1.4 Poky Linux/ Arduino IDE C++, python, Nodejs, js
XBee®	EM357 32-bit ARM® Cortex -M3 processor Operation at 6, 12, or 24 MHz	12 KB RAM	128 or 192 kB flash Memory	DIO 0-8 PWM 0-1 ADC AD 0-5 UART, SPI	IEEE 802.15.4 6LoWPAN	Bootstrap/AT Mode, API Mode UART
CypressPSoC4 BLE	PSoC 4 BLE is an 32-bits ARM® Cortex®-M0 48MHz PSoC 4200	32 KB/16 KB SRAM	256 KB/128 KB Flash	GPIO 3 Ports up to 98 I/O, ADC 12bits, 4 UART, 8 PWM	Bluetooth 4.1	Bootstrap/ PSoC Creator Embedded C
Microchip PIC32 MZ curiosity	PIC32MZ2048EFM100 32-bit MCU: 200MHz	512KB SRAM	2MB Flash	Integrated FPU & Crypto accelerator,Hi-Speed USB	LAN8720A: 10/100 Ethernet Transceiver & CAN ATWINC1500 IoT network controller: IEEE 802.11 b/g/n Single-band 2.4GHz	FreeRTOS/ MPLAB Embedded C
Platform	**CPU/MCU/SoC**	**RAM**	**Memory**	**I/O**	**Network Connectivity**	**OS/ Programming**
Microchip	ATSAMG55 32-bit	176 KB	512 KB	Two PIO	ATWINC1500	Bare metal/

SAM IoT Dev kit	ARM® Cortex®-M4 120MHz (Complete development and prototyping platform on AWS IoT service Includes three CryptoAuthed Xplained Pro Rev B (ATCRYPTOAUTH-XPRO-B) add-on boards, with ATECC508A and ATECC608A.)	SRAM	Flash	Controllers provide control of up to 48 I/O lines, One 8-channel ADC, resolution up to 12 bits, SPI, USART, USB	IoT network controller: IEEE 802.11 b/g/n Single-band 2.4GHz	Atmel Studio 7 Embedded C

Raspberry Pi 3 is the latest hardware # All the specification collected from official source

3.3. Cost

The following Figure.12 shows the comparative analysis of the cost in $ USD of the mentioned IoT development platforms.

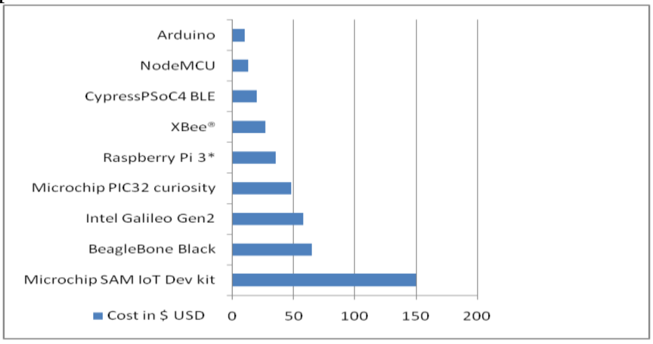


Fig. 12 Cost in $ USD of IoT development platforms

3.4. Reliability

The following Figure.13 shows the comparative analysis of the reliability of the mentioned IoT development platforms. Reliability index is the measure of stability of system over an IoT deployment and service provisioning. This factor depends up on the device built quality as well as the type of programming or stack used in the device.

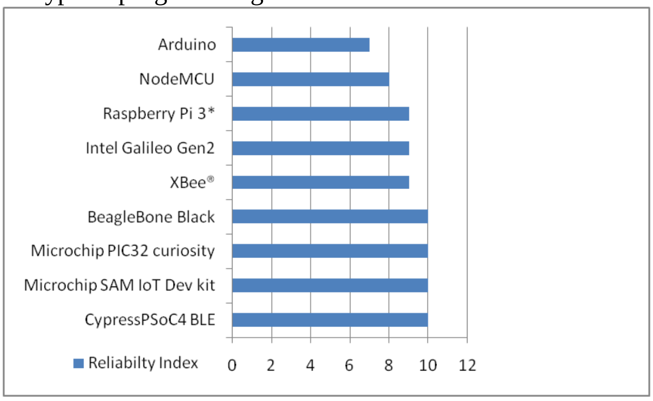


Fig.13: Reliability Index of IoT development platforms

3.5. Learning Curve

The following Figure.14 shows the comparative analysis of the learning curve of the mentioned IoT development platforms. Learning curve is the difficult level in getting technical expertise to develop IoT products in the platform.

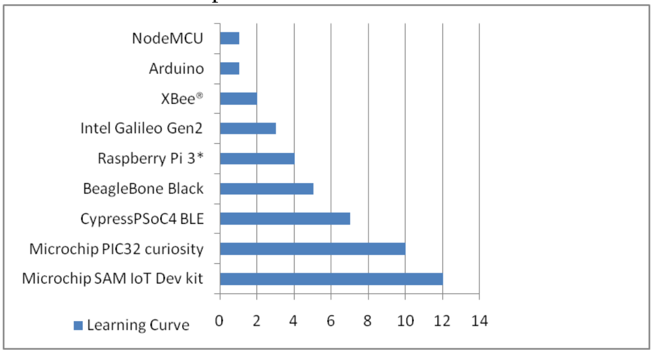


Fig.14: Learning curve of IoT development platforms

3.6. OS support & Programmability

The following Figure.15 shows the comparative analysis of the OS support & Programmability of the mentioned IoT development platforms. This gives an idea that how well supported the particular platform is. Some platform support only firmware level programming at the same time some platform support bare metal as well as Linux or RTOS.

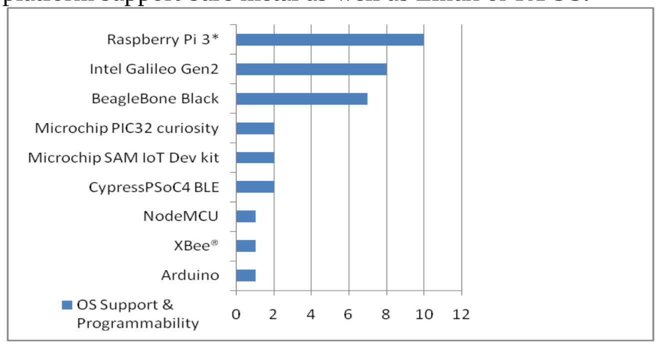


Fig.15: OS Support and programmability of IoT development platforms

This OS support also affect the overall system programmability factor as a full-fledged Linux OS can support a wide range of programming language.

3.7. Time to market

The following Figure.16 shows the comparative analysis of the Time to market of the mentioned IoT development platforms. Time to market is the factor which is crucial for developing a IoT application. The stiffer will be the learning curve and more complex would be the development platform more time it will take to develop an application on that platform.

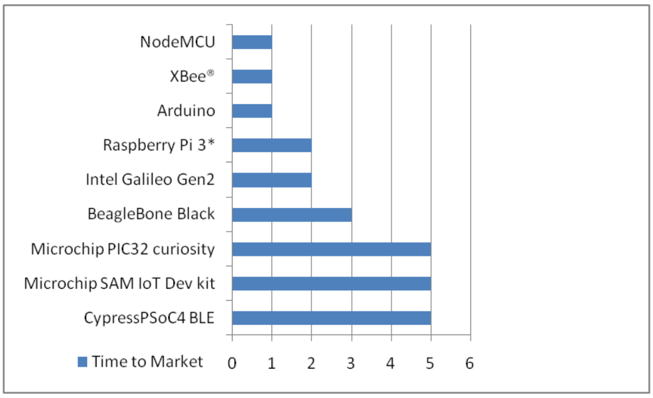


Fig.16: Time to market of IoT development platforms

4. RECOMMENDATIONS FOR IOT APPLICATION PROTOTYPING

Based on the comparative analysis given in Chapter III, we can draw some inference and a recommendation can be drafted. In case the IoT application is a basic one which does not require any specific industrial demand Arduino or NodeMCU can be used both are very less learning curve and time to market. A very easy prototyping can be done using Arduino using required shield based on the application requirements. In case the application requires a very high computational requirement like deployment of a cloud server of MQTT broker to be deployed then Raspberry Pi 3 is the best option. Other development boards like TI's BeagleBone Black or Intel® Galileo Gen2 can also be used for such complex applications. All these platforms give a very good Linux OS support and very rich set of programming library support for complex and computationally power hungry applications. If the IoT prototyping requirements involves a very high duty cycle or high reliability ideally for everyday deployment scenario then industry grade MMRP based embedded platforms like Cypress PSoC4 BLE, Microchip PIC32 or SAMG55 based IoT dev kit can be used for best service.

IoT is not a technology by itself rather a technology framework for interconnected smart objects and devices over internet enabled by various modern days technologies. Toda's IoT industry takes the technological advantages of multi-disciplinary research in the fields of industrial instrumentation (sensors and actuators), VLSI, Embedded systems, networking and Internet technologies. Further IoT industry gets fuelled by the latest advancement in the field of cloud-computing platforms. The extensive use of tools likes Machine learning and Artificial Intelligence is aiding the IoT industry multi folds in recent times. Figure 17 depicts a generic IoT architecture for modern applications. Point to be noted here the architecture remain unchanged in most of the scenario only the building blocks can be customized based of scenario specific requirements. A heterogeneous IoT application framework is presented in the paper [35].

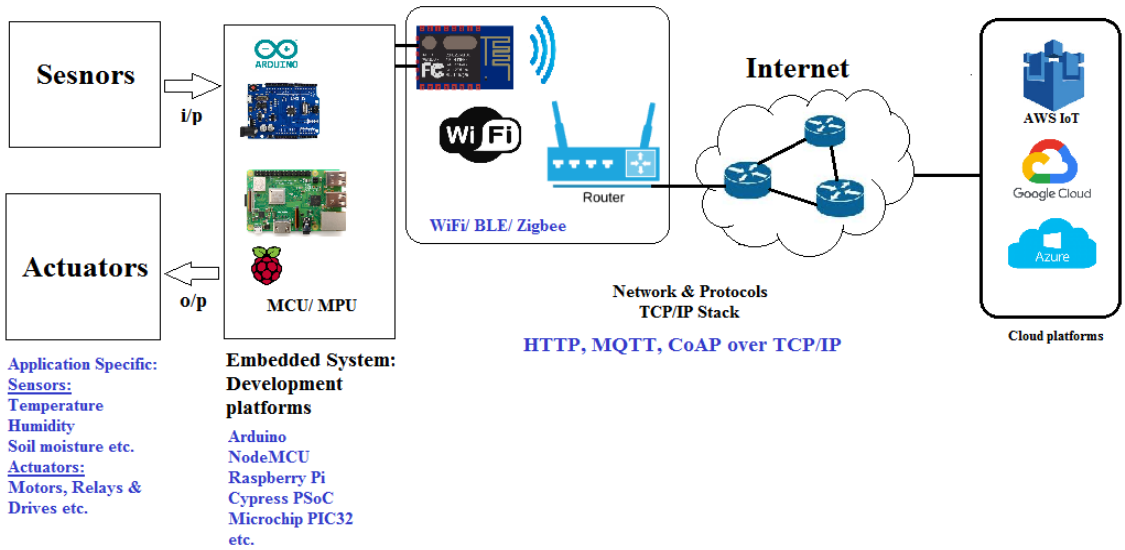


Fig.17: Generic Architecture of Internet of Things for modern day's applications

5. CONCLUSION

The review can be concluded with a endnote that, the selection of IoT prototyping development platform can be completely based on application specific requirement. No particular embedded development platform is best or worst. It is the application, which makes it best suitable. The previous section has given some recommendations based on which a particular development platform can be selected and an IoT application prototyping can be provisioned to the end user.

The Arduino family is best suitable for low duty cycle applications with not much demand on computational resources (8-bit 20 MHz MCU) having very easy to learn and user friendly development environment with lot of online resource available. Arduino UNO is low cost under $10 with lower reliability score of 7 in this review. Arduino UNO lacks in one aspect that, it requires external shield or Wireless Modules to be connected to cloud for IoT applications. NodeMCU is close competitor to Arduino UNO with better programming options as it supports both Arduino IDE and LUA scripting. In this review having a 32-bit 80MHz MCU is having a computational power almost four times of Arduino UNO. However XBEE® is a whole new platform especially for Wireless Sensor Network deployment with IEEE 802.15.4 Zigbee technology. XBEE® is best suitable for low power sensor network application with complete support of mesh networking. XBEE® is based on a 32-bit ARM® Cortex -M3 MCU up to 24MHz. XBEE® is having a better reliability index of 9, which is more than Arduino as well as NodeMCU in this review.

Unlike the above three development boards the Cypress PSoC4 BLE is based on a 32-bits ARM® Cortex®-M0 48MHz MCU support industry grade product with support for Memory Mapped Register Programming (MMRP). Microchip SAM IoT kit is powered by a 32-bit ARM® Cortex®-M4 120MHz MCU where as Microchip PIC32 is powered by a 32-bit MCU 200MHz. Cypress and Microchip development boards support the industry grade deployment of IoT product and solution with higher duty cycle applications. The key is the reliablity and extensive support for programmablity. In this review a higest reliability index of 10 has been given to above three Cypress and Microchip platfroms.As these boards offer more reliabilty and higher grade product development support hence the cost of these devices are significatly higher. One more key point to be highlighted here that all the above discussed development platfrom works on bare metal environment.

Finally in this review we have discussed in section 2 about three MPU based boards, which is of a another segment altogether. These boards have support for a full fledged Linux based OS and are having a extensive features like rich application support. The three boards we have discussed in this category are Intel® Galileo Gen2, Texas Instruments Beaglebone Black and Raspberry Pi 3. Intel® Galileo Gen2 is powered by an Intel® Quark SoC x1000 400 MHz 256 MB DDR3 RAM. TI Beaglebone Black is having a 1 GHz ARM Cortex-A8 core with PowerVR SGX 2D/3D graphics processor with 512 MB LPDDR RAM. Raspberry Pi 3 is powered by a Broadcom BCM2837 quad core ARM Cortex-A53, 1.2GHz with a GPU of VideoCoreIV and 1 GB LPDDR2 RAM. All these platform support Linux based OS. As far as the technical specification is concerned Raspberry Pi 3 is a sure winner and surprisingly the cost is lowest around $35 however Intel® Galileo Gen2 costing around $58 and TI board is near about $65. This is a clear indicator why the Raspberry Pi is so popular among product and application developers around the globe.

ACKNOWLEDGMENTS

All the development boards are evaluated and tested in Wireless Communication and Networking Lab, School of Electronics Engineering, KIIT Deemed to be University.

REFERENCES

[1] Nayyar and V. Puri. "A review of Arduino board's, Lilypad's & Arduino shields". In 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, pages 1485-1492, 2016.

[2] D. R. Patnaik. “A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board”. International Journal of Advanced Research in Computer Science, Volume 8, No. 5, pages 2350-2352, May-June 2017.

[3] M. Mehta. “ESP8266: A Breakthrough in wireless sensor networks and internet of things”. International Journal of Electronics and Communication Engineering & Technology(IJECET)Volume 6, Issue 8, pages 07-11, Aug 2015.

[4] Espressif official documentation: ESP-32 datasheet V3.0, https://www.espressif.com/en/support/download/documents

[5] H. Chaudhari. “Raspberry Pi Technology: A Review”. International Journal of Innovative and Emerging Research in Engineering, Vol. 2, Issue 3, pages 83-87, 2015.

[6] M. John. “Comparative study on various system based on Raspberry-Pi Technology”. International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 01, pages 1486-1488, Jan 2018.

[7] G. Bekaroo and A. Santokhee. "Power consumption of the Raspberry Pi: A comparative analysis". In IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech), Balaclava, pages 361-366, 2016.

[8] L. H. Nunes et al. "A Study Case of Restful Frameworks in Raspberry Pi: A Performance and Energy Overview". In IEEE International Conference on Web Services, Anchorage, AK, pages 722-724, 2014.

[9] S. Barrett; J. Kridner. "Bad to the Bone: Crafting Electronic Systems with BeagleBone Black ". Bad to the Bone: Crafting Electronic Systems with BeagleBone Black, Second Edition, Morgan & Claypool, 2015.

[10] A. Nayyar and V. Puri. "A Review of Beaglebone Smart Board's-A Linux/Android Powered Low Cost Development Platform Based on ARM Technology". In 9th International Conference on Future Generation Communication and Networking (FGCN), Jeju, pages 55-63, 2015.

[11] A. Nayyar and V. Puri, “A review of Intel Galileo development boards Technology”. International Journal of Engineering Research and Applications, Vol. 6, Issue 3, (Part -4), pages 34-39, March 2016.

[12] P. Cocchi. "Analyzing and Experimenting the Intel Galileo Board for the Internet-Of-Things". DIAG Technical Reports 2015-12, Department of Computer, Control and Management Engineering, Universita' degli Studi di Roma "La Sapienza", 2015.

[13] M. Ramon. “Intel Galileo Gen 2 and Intel Edison for Beginners: A Hands-on Introduction”. Apress; 1st ed. edition December 2016.

[14] M. Schwartz. “Intel Galileo Blueprints”. Packt Publishing Limited, June 2015.

[15] M. Richardson, “Getting Started with Intel Galileo”, O′Reilly; 1 edition, March 2014.

[16] C. Rush. “Programming the Intel Galileo: Getting Started with the Arduino -Compatible Development Board”, McGraw-Hill Education TAB; 1 edition (16 December 2016).

[17] R. Grimmett. “Intel Galileo Essentials”, Packt Publishing Limited (24 February 2015).

[18] A. H. Kioumars and L. Tang. "ATmega and XBee-based wireless sensing". The 5th International Conference on Automation, Robotics and Applications, Wellington, 2011, pp. 351-356.

[19] V. Khedekar, S. Mahajan, A. Karangalel, “A Review on XBEE Technology”, International Journal of Emerging Technologies in Engineering Research (IJETER), Volume 4, Issue 4, pages 99-101, April 2016.

[20] Doraipandian, Manivannan and P. Neelamegam. "Wireless Sensor Network Using ARM Processors: A Review in Hardware Perspective". IJERTCS 4.4: 48-59, Apr. 2019.

[21] P. Rycerski, L. M. Candanedo Ibarra, F. Galatoulas, K. N. Genikomsakis, A. Bagheri, C. S. Ioakimidis. "Field performance analysis of IEEE 802.15.4 XBee for open field and urban environment applications in Smart Districts', Energy Procedia, Volume 122, pages 673-678, ISSN 1876-6102, 2017.

[22] R. Piyare, Seong-ro Lee. "Performance Analysis of XBee ZB Module Based Wireless Sensor Networks". International Journal of Scientific & Engineering Research, Volume 4, Issue 4, pages 1615-1621, April-2013.

[23] Mayalarp, Vachirapol & Limpaswadpaisarn, Narisorn & Poombansao, Thanachai & Kittipiyakul, Somsak. "Wireless mesh networking with XBee". In 2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Pattaya, Chonburi, Thailand, 2010.

[24] Jingxia, W. A. N. G. "Application of zigbee/ieee 802.15. 4 protocol compatible rf module xbee/xbee pro [j]". Electronic Engineer 3: 008, 2017.

[25] Yuming, Wu Yongsheng Wang Wei Shen. "Design of wireless sensor networks based on ZigBee [J]". Electronic Measurement Technology 11, 2009.

[26] Garcia-Espinosa, Eduardo, et al. "Power Consumption Analysis of Bluetooth Low Energy Commercial Products and their Implications for IoT Applications". Electronics 7.12: 386, 2018.

[27] Prakash, Y. W., et al. "Smart Bluetooth low energy security system". In International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, 2017.

[28] Bazzi, Alessandro, et al. "Experimental Characterization of a Low Power Device for IoT Applications: Micro. sp©". In IEEE Conference on Standards for Communications and Networking (CSCN), IEEE, 2018.

[29] Rossi, Davide, et al. "Ultra-low-power digital architectures for the Internet of Things". Enabling the internet of things, Springer, Cham, pages 69-93, 2017.

[30] Di Jasio, Lucio. "Programming 32-bit Microcontrollers in C: Exploring the PIC32". Elsevier, 2011.

[31] Lynch, Kevin, N. Marchuk, and M. Elwin. Embedded computing and mechatronics with the PIC32 microcontroller. Newnes, 2015.

[32] Kerényi, Kristóf, and P. Szabó. "Design of Secure and energy-efficient embedded systems for Future Internet applications". framework 1, 2011.

[33]https://www.microchipdirect.com/product/search/all/AT88CKECC-AWS-XSTK-B Microchip Product descriptions.

[34] Mahmoud S., and A. AH Mohamad. "A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications". 2016.

[35] A. Mishra. "Design and Deployment of MQTT Based HeTNeT Using IEEE 802.15.4 and IEEE 802.11 for Internet of Things". International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5, Issue XI, pages 1616-1625, 2017.