Analysis of Biomedical Applications in Embedded Systems Devices

[1]Yue Dong and [2]Charlie Siu
[1,2]School of Information Science and Engineering, Southeast University, China, 211189.
[1]yuedongsu@hotmail.com

Article Info
Journal of Biomedical and Sustainable Healthcare Applications (http://anapub.co.ke/journals/jbsha/jbsha.html)
Doi: https://doi.org/10.53759/0088/JBSHA202101015
Received 12 November 2020; Revised form 10 January 2021; Accepted 25 April 2021.
Available online 05 July 2021.

Abstract – Embedded systems are rapidly being used in clinical and biological applications, as well as commercial, telecommunications, government, and other business applications. Embedded system solutions are growing in popularity, not only with types of technologies, garments, industries, healthcare and military hardware, and mobile computers, but with software solutions like' electronic worlds' and 'mobile worlds,' deep learning, and internet of things, which allow for the creation of a wide range of application. With the growth of viral illnesses like the Covid-19 virus, tele-health technologies for diagnostics, prognostic, and patient treatment have become more important in recent decades. In medical technologies, embedded device techniques have taken a significant role. Developing techniques to improve the security of medical practitioners in the case of pandemic contagious diseases, such as epidemics, is particularly important. Patients released from clinics home-based or in treatment wards that are non-intensive during the quarantine period, or segregated in their residences, outpatients' departments, and moderately ailing individuals are progressively being monitored remotely, instantaneously, safely, and rapidly for this reason. The applications biomedical applications in embedded systems will be discussed in this paper.

Keywords - Embedded Systems, Biomedical Applications, Medical Devices, Internet of Things.

I. INTRODUCTION

The utilization of mobile gadgets occurs at a moment when the COVID-19 epidemic is hastening telehealth implementation. During the epidemic, more than 42% of Americans utilized telehealth. In the summertime of 2020, claims increased by 6,000 percent as a result of this. These gadgets have enabled physicians to check on individuals remotely, diagnose them, and prescribe therapy during curfews. They've also aided in the alleviation of a dearth of health in rural regions. Here are five ways that clinical and IoT instruments facilitate the development of new innovations:

- Attached bronchodilators: Smart inhalers can monitor medicine and provide notifications when it's time to take a dosage. Apps that are connected to the inhaler may instruct patients on how to use it and provide information such as allergy predictions. Sensors can monitor a patient's respiration and notify their doctor if anything is wrong. The doctor may then make any required adjustments to their therapy.
- Glucose sensors and compressors: Smart sensors in spots or implantation can test blood glucose without causing discomfort. To test their sugar concentration, individuals no longer need to scratch their wrist or have blood extracted. Insulin delivery systems use an automated pump to administer a constant supply of insulin. This allows patients to better manage their condition at home and gain control over their health. They may reduce the strain on the health system by avoiding excessive glucose levels highs and lows.
- Smart watches: Smart watches are not only for counting steps anymore. They can track anything from pulse rate to calories consumed and reduction, water usage, and sleep habits, among other physiologic data. Sensors that convey microcurrents across the body are used in certain technologies to assess a patient's condition.
- Smart hospital beds: Smart beds with embedded sensor applications can continually monitor a patient's symptoms and warn healthcare personnel if anything changes. This will enable them to react quickly to situations and achieve greater results. Smart beds may also adapt to the patient's most comfortable position.
- Medical devices for chronic illnesses: Doctors are shifting their attention away from emergency treatment to address the demands of aging populations. Medical gadgets that monitor an individual's vital signals may detect early indicators of sickness. Doctors may intervene early in the course of a disease to help prevent it from becoming chronic. As a result, hospital admissions are reduced, and resources are better used.

It is critical for software and hardware in integrated systems to function together. To handle and evaluate the data collected by medical devices, special software is required. A rising number of gadgets, monitors, and other biomedical applications must be tracked and controlled by the software. Hardware must be durable and dependable. The gadgets must provide data that is easy to understand for clinicians and patients. They must also be examined on a regular basis to verify that the sensors are functioning properly.

With embedded systems are characterized as specialized hardware with a Central Processing Unit (CPU) and communication systems that is tailored to execute a single task. Through the use of a microcontroller or microprocessor, platforms comprising of a mixture of hardware and software are purposed to permit the approach integrating the electric and mechanical elements to perform particular purposes. Embedded systems seldom interface with the end customer directly. They function with restricted resources in real time for a specific purpose, and they may undertake very vital activities in their utilization regions on occasion. Mistakes here may result in significant losses of properties and life. It is critical from this perspective that these systems be "reliable" and "tolerant of faults." It is an automated (or electro - mechanical) framework that normally includes components such as memory, communication interface modules, detectors, and physiological output. It is generated by encoding computer on microcontrollers, microcomputers and Digital Signal Processors (DSPs), or the Field Programmable Gate Array (FPGA). It is not to be conflated with the specialized system, which is a separate event. Embedded devices are often smaller in size and have lower CPU power. Furthermore, a devoted system is one that has been designated to perform particular tasks [1]. For instance, the library's web host and the web servers can be hosted on various computing systems, and this is considered as dedicated systems if the objective of these systems is to completely host the web servers or offer the information.

Healthcare embedded device implementations are selected since they are genuine and extremely quick. Integrated schemes are the computing schemes, and have to be comprehended and manage their functions and features. The processors, memory for data storage, transforming micro-controllers or the DSP (Digital Signal Processor), actuators and sensors, including interconnections make up an intelligent system. Section I present an introduction of embedded systems and outline the five ways that clinical and internet of things instruments facilitate the development of new innovations. Section II below presents a discussion of biomedical applications of embedded systems. In this section embedded system programming tools will be discussed; and a discussion of the hardware for embedded systems will be provided as well. Lastly Section III concludes the paper.

II. BIOMEDICAL APPLICATIONS OF EMBEDDED SYSTEMS

Embedded System Programming Tools

The following is a list of Embedded System Programming Tools:

Integrated Development Environment (IDE)

An Integrated Development Environment (IDE) is a piece of program that gives developers all the tools they need to create software [2]. A sources code editor, build automated tools, and a debugger are all standard components of an IDE. Some IDEs, like NetBeans and Eclipse, come with the required compilers, interpreters, or both; others, like SharpDevelop and Lazarus, don't. The line between an IDE and the rest of the computer developing ecosystem isn't always clear; for example, a change control systems or other tools to make building a graphical user interface (GUI) easier are occasionally incorporated. For usage in object-oriented computer programming, many current IDEs include a class explorer, an object search engine, and a type structure graph.

Compiler

A converter is computer software that converts compiled code in one computer program (the source dialect) into code written in other programming languages (the target language) (the target language). The term "compiler" refers to software that converts software code from such a high-level computer program to a lower-level language (such as machine code, binary data, or machine code) in order to build an executable program: Compilers come in a variety of shapes and sizes, and they all provide valuable results. A cross-compiler generates code for CPU or system software other than the one it is running on. A bootstrapping converter is implemented in the target language. A decompiler is software that converts one language into another. A source-to-source compiler, sometimes known as a transpiler, is software that converts between high-level languages. A linguistic rewriter is software that converts the version of expression into another language without changing the language. A compiler-compiler is one that generates another translator (or part of one).

Software and Hardware Defects (Debugger)

A debugger, often known as the debugging instrument is a computing application, which is utilized for testing and troubleshooting the target programs (applicants).

The major aim of a breakpoint is to permit developers to follow targeted programs' activities in the process and track the variation in computing resources (typically the memory location used by the programs codes and machine system application), which may display faulty codes. The capacity to halt or manage processes, which are linked to certain points, exhibit the components such as storage systems (e.g. disk drivers), registers, CPU memory and adjust the recollections or record component to activate the chosen test data, which might be the cause of faulty application are all the typical debugging features.

Emulators

An emulator is a piece of equipment or application that helps system software (referred to as the host) to act like another (called the guest). In most cases, an emulator allows the hypervisor to execute software or utilise external devices that are developed for the visitor system. Emulation refers to a computer software's capacity to replicate (or simulate) another application or device on an electrical gadget.

Some characteristics distinguish Integration Services Development from other programming platforms. The amount of program memory used by embedded software should be kept to a minimum. Application systems should not execute any faster than the system demands [3]. It should be simple to interact with embedded systems and straightforward to read. Image Instruments are the most common medical embedded device implementations. Imaging devices e.g. Positron Emission Tomography (PET), Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) are all embedded systems, despite the fact that their operating principles vary. Defibrillators are another illustration of embedded platform uses in medicine. Defibrillators, a machine that monitors a patients' pulse for irregular situations and typically returns it to usual state whenever the aberrant pulse has been identified, are the best samples of biomedical embedded system application. The computerized flow sensors are used to keep track of a patient's respiratory system. The human body's systolic and diastolic pressures may be detected with a blood pressure monitor and a glucose test kit. The designs for medical devices, which allow users to effectively track data such as heart rate, glucose and hypertension, are examples of integrated devices. Fatal heart monitoring machines are used to measure the pulse of new-borns throughout pregnancy, delivery, and birthing. It is fair to claim that computers are present in all electronic gadgets. There were a lot of "embedded" electronics.

Fig 1. Solutions of the NVIDIA Jetson

With a decrease in the development and size of the computing power, micro computing devices with integrated devices like Raspberry Pi and Ardunio [4] can gather participant information and perform data analysis with rewritten software. Embedded device in similar tiny gadgets can make management choices that assist patients get effective treatments and drugs. Embedded systems, detectors, motor drivers, and detectors are all becoming more prevalent in small designs. It offers cost-effective and practical solutions for any emergency situation. Studies like "the camera application, which measure the overall social distances" has aided in better grasping the essential social distances that have to remain between individuals since infectious illnesses like Covid19 and others are spread via touch. The development that has increased with vaccine research all around the globe has now resulted in the conclusion of the quarantine phase and the restoration to community interaction. Nevertheless, there will still be a danger of infection if the sociological boredom is not preserved for contagious illnesses, which are increased every year. As a result, keeping social distance in densely populated locations is critical. The initial items utilized to solve the issue were realistic embedded device alternatives, like smart phone program apps and smart wristbands.

A software application through framework, such as a web API, is another option for this issue. However, their high cost is a drawback. Another embedded device solution to the issue is the application rewrite utilizing solutions like as NVIDIA, merging low-cost Understanding Machine Learning, Computer existing workplace cameras, and giving real-time surveillance of sociological distance. There will be no additional product costs since this will be a component that can be put on current cameras. Other fields benefit from embedded software solutions since they are quick, inexpensive, and simple to use (Fig 1).

Hardware for Embedded Systems

The term "embedded system" has already outlived its usefulness. Everything is indeed an automatic machine, from smart televisions to smartphones, washing machines to set-top box. They are no longer constrained to performing particular assignments; otherwise, they are multi-task. Microcontrollers are utilized for computer systems in the application with lower processing duration (the process, which contributes in the changes from conception to death with minor delays). Temperatures, gas flow, humid, ultrasonic and other sensor devices are embedded systems. Low-cost and simple-to-use smart system for enthusiasts might be utilized to access this domain. Operating devices have low-power smart processors inside high-frequency computing devices, have computing platform or more direct actual-time administration infrastructures, and operate sensors and trainers [5]. Despite the fact that transaction volumes are still smaller than those of computer networks, the gap in today's science and technology has begun to decrease. At the point when operations-based tasks like robotic system rationalization, intelligent systems, and deep learning are performed, embedded device approach takes up much space. Software drivers, static analysis instruments, configuration management tools, Linux tools, software libraries, logic analysers, IDE, debugger, and oscilloscope, and others are among the most significant software/hardware tools, according to surveys. It is essential to have a thorough understanding of the hardware used in digital systems. Every topic of electrical applications, every fundamental circuit, is found on mobile processors and digital devices, and the devices examined were developed using logic gates. Because the gadgets are not simple, understanding them with just half of the knowledge is impossible. Integrated systems are information systems, and all of the elements and characteristics of this computer network must be understood and controlled. Most are accredited to hardware and illustrate embedded schemes through that specific hardware, as we can see. The following is a list of hardware components:

- MCU (Microcontroller Unit) [6]
- FPGA (Field Programmable Gate Array) [7]
- ASIC (Application Specific Integrated Circuit) [8]
- GPIO (General-Purpose Input/Output) [9]

A single chip processor is what embedded systems, one of the most fundamental components of integrated devices, are. A CPU, memory, digital input-outputs, and other devices is all included (timer, interrupt, ADC etc.). RE R&C, Synergy, RA, STM8, RH850, V850, H8 Family, 78 K, RL78, and RX Family are only a few examples of well-known mobile processors. There is no requirement for analog information systems in sophisticated integrated devices, as it may seem. Electrical knowledge that is both basic and practical may be enough. Real-time operating systems (RTOS), programing library for third party and application platforms are the three primary categories in which the software components are grouped. Below is a table from the Aspencore group's 2017 survey on the forums used in integrated framework architecture (Table 1).

Table 1. Embedded design with the board of development

Developments Board Began with	N = 356.00	%
Nordic/nRF52-DK	3	0.8%
MSP430 – TI	3	0.8%
ESP32 3 0.8%	3	0.8%
Digi	3	0.8%
Silicon Labs	4	1.1%
Intel Edison	5	1.4%
Avnet	5	1.4%
Altera StratixVDSP Kit	5	1.5%
Renasas	6	1.7%
Cypress kits	6	1.7%
Freescale (NXP)	10	2.8%
Atmel	10	2.8%
Beagle Board Bone Black	12	3.4%
Rasberry Pi	15	4.2%
Arduino	20	5.6%
Microchip	21	5.9%
NXP	26	7.3%
Xilinx		
TI (LaunchPad = 5)		
ST Microelectronics		

Measuring Equipment is required in embedded device labs in both biomedicine and non-biomedical sectors. Digital oscilloscopes generators, table top digital millimetres, power supplies, computers, and projectors are just a few examples. The amount of assistance tools and measurement devices required vary based on design work and research. Working with Arduino, for instance, which is a reduced structure created for designers and artists who aren't particularly concerned in electrical or computer engineering, the measuring devices required and those operating with detectors may change (Fig 2).

The computer component is the primary reason for Arduino's widespread adoption. The website provides a free installation of the Integrated Development Environment (IDE) obligatory to construct an application for Arduino. It may be built quickly using a developing ecosystem that runs on Linux, Mac OS X, and Windows OS. Arduino IDE is founded on the Manufacturing computer language and the interlinking project. It is easier to integrate Arduino initiatives (named drawing) and present them on cards using the programming environment, which is very user-friendly. Programming is done in a grammatical structure that is similar to C++. Many transactions and perivascular interaction can be conducted with ease thanks to the operating libraries. The Arduino [10] includes an expandable library system, which is one of its most advanced features. As a result, new external libraries can be easily implemented. One of the most appealing aspects of the Arduino is the ability to add new hardware functionality using "shield" cards. Many different projects can be realized using these extra documents that are consistent with the Arduino board. Modules like Bluetooth, wifi, motor driver, and LCD screen are instances of these adapters. Detectors, which are additional kind of appliances in embedded devices, are developed to fulfill a variety of functions as technology advances. This is cutting-edge technology. MEMS/NEMS embedded sensor systems, as well as optical technologies, will be utilised in Internet-based object applications. Sensors that can withstand harsh environments Sensors having novel characteristics (such as high resolution and sensitivity, low cost, high reliability (robustness), self-calibrating, and resistance to mistakes, losses, and degradation). Sensors technologies may be applied in the manufacturing process.

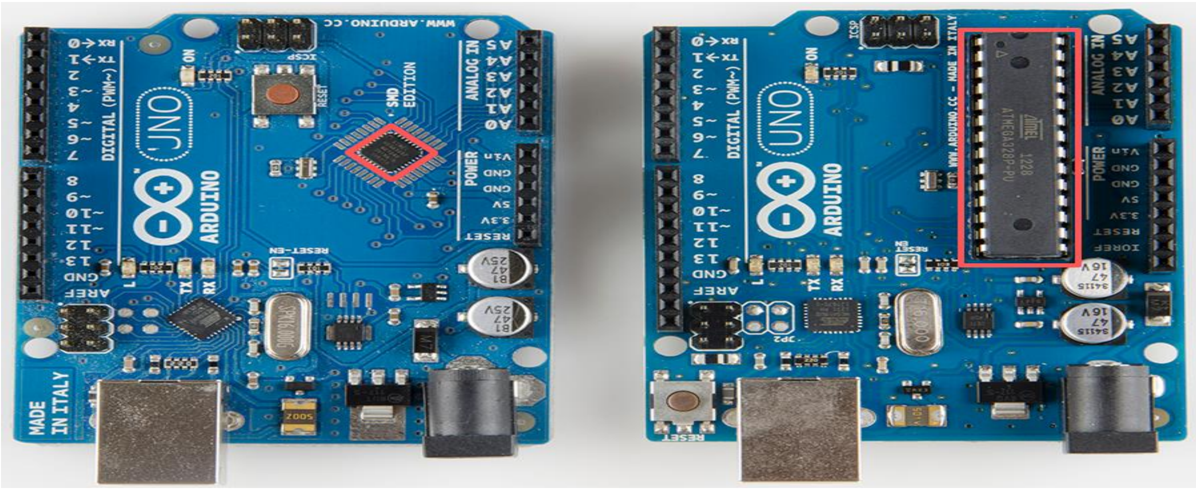

Fig 2. Arduino Uno SMDR3

Software for Embedded Systems

Except you know a software engineering genius, gadgets about to get a huge amount of data, even to produce eye-catching interior wiring devices, will not be excellent embedded software. In fact, the program is now used in integrated system management programs for a variety of components. Embedded gadgets now have access to image analysis, server software, user interactions, and operating systems. You'll be using the C computing languages a lot while working on embedded devices. Even though other programs like Java and C ++ are used in various industries, and Meeting is used in electronics systems that need speed or don't have access to a C compiler, C is just the most widely used language. The results of the Aspencore group's integrated software market analysis in 2017, that included a survey of businesses and scientists operating in the field of integrated systems, found that the most commonly used platforms are C, C++, and Java (Table 2).

Table 2. Embedded scheme programmed in the Aspencore 2017 survey

LANGUAGE	2017 N=880	2015 N=1217
C	56%	66%
C++	22%	19%
Assembly Language	4%	3%
Python	3%	2%
Java	2%	2%
LabVIEW	2%	1%
C#	2%	2%
MATLAB	2%	1%
JavaScript	1%	0%

The C programming language is certainly the most prevailing aspect since it integrates both the high and moderate capacities, generates less codes, is widely utilized in embedded systems library, is known as the translator for various embedded devices, and has easier accessibility to resources. FreeRTOS, QNX, Pharos, SeL4, SCIOPTA, OSE, Honesty, Micrium (uC and OS-III), Windows 7, Texas equipment BIOS and DSP, Texas Component RTOS, Windows Embedded 7, Ubuntu, In-house/custom and Integrated Linux are the most popular working systems in integrated devices. The evolution of embedded systems is likewise moving at a quick pace. While previously dealing with embedded systems with limited resources, it is now feasible to discuss devices that operate at very fast speeds. Internet applications and encryption techniques, for example, have recently emerged as new research fields. Using ready-made packages on the networks we deal with has become a must. Platforms that help us are becoming more important as programming grows more sophisticated. The Internet of Things (IOT) framework includes "gateways," which link additional components and sensors to the internet. By improving machinery that can operate with newly developed procedure constructions and also adjust to proper procedures, it is assured that higher additional value, user-friendly assessment, service, surveillance, management and decision-making application are established with the establishment of innovative generation communications network and standards.

At the same time, internet of things constructions have complied and executed the principles required to process more containable, controllable and safeguard system architecture, to communicate with the sensor system, to have quality services meant to avoid production disruptions, to endorse regional and international guidelines, to work publicly with various systems when required, to have computer program circuits for various applications, and to have a modular and discoverable system design (mobile, etc.). Coding is another appealing aspect of embedded system software. Considering the idea of greatest advantage, the smallest space, shortest time, and authoring is the most suitable. It is best to compose with the principles of least amount of space, least amount of time, and most benefit in mind.

Designing Medical Devices

One of most useful fields of embedded software and hardware is medical equipment development. Embedded software implementations are also used to create biomedicine powertrains, medical and biogenetic integrators, portable diagnostic approaches, mobile patients' tracking system, doctor call structure, patient queue system and their structures, in addition to the abovementioned therapeutic visualization schemes [11]. In the biomedical discipline, there is a major commercial sector in addition to academic study. These healthcare embedded device, embedded computers, and client project demands should all be SBC (Single Board Computer) ready. Investors, medical personnel, and consumers in the medical industry may benefit from their high-performance, low-cost solutions. Portable embedded systems with low power and excellent performance provide a wide range of diagnostic tasks, save time, and lower total diagnostic expenses. Raspberry Pi is also another device that is often used in electrical, robotic, and biological projects, similar to Arduino. There is no operating system on the Arduino. It can only execute Arduino-compliant applications, which are typically written in C++. The Raspberry Pi is commonly equipped with a Linux-based operating system. With this functionality, it might be referred to as a small computer. The Raspberry Pi is now the cheapest computer available. At first sight, they seem to be quite similar. Pins, connections, and screw holes, to name a few. In truth, both cards are diametrically opposed. In terms of software, the Raspberry Pi stands out due to its greater memory and embedded processors. It is a personal computer that may be used for image analysis in clinical research (Fig 3)

Fig 3. Raspbery Pi4 Model-B

Programs like opencv simplecv are used to create healthcare instruments that can conduct image processing and a variety of other tasks. Via embedded device remedies, for example, you may work on a picture captured using a medical modality. You may use an embedded device instrument such as the raspberry pi, nvidia, arduino or complex microprocessors and cards to run the image assessment program you generate on regular computing systems. With the relevant application you create on Raspberry Pi and the display linked to it, for instance, you may execute image analysis such as corner detection over the picture.

Constraints on Embedded Systems

One of the difficulties with embedded devices is that they are often run on hardware, which is not a full-fledged computer. This is usually a low-cost device that has been purpose-built or is off the shelf, such as the Raspberry Pi. As a result, this design has a number of limitations. As this case shows, there may be a restricted amount of features accessible. There might be some interface on this gadget that may not display on your personal computer. This is also difficult to modify, since the system software may restrict the capacity of the device's storage. And, as you can see, upgrading the architecture of this integrated platform would be quite tough. There might possibly be a communication constraint. Although this gadget has connected ports, the embedded system utilized elsewhere would only be capable of interacting wirelessly and with certain wireless protocols. Embedded systems, on the other hand, are generally created to accomplish a particular job.

There's no use in adding more memory or telecommunications features to these gadgets if they're never going to be used. We can maintain our expenses low since we designed this gadget to execute a specific task, but this come at the cost of having these extra capabilities.

The power of these microcontrollers is a typical restriction. We often deploy these gadgets in the field, where there would be no direct control supply. During the day, we are often hooking into a power interconnection or replenishing the battery with renewable energy. It's possible that you'll have to manually service the intelligent system only to remove the battery since there's no other way to power it. These low-cost wearable sensors are often bought off-the-shelf. As a result, the integrated device's CPU and computational capability may be limited. This isn't always a terrible thing. Lower-speed CPUs generate less heat, which might be beneficial in these compute nodes. From a connectivity standpoint, the placement of these embedded systems may restrict what we should launch. If the sensors are on an oil filter inside the middle of the jungle, we would not be capable of linking it to a network switch or a tornado access network [12]. The sort of connectivity that the embedded system may utilize may be determined by its geographical position. As a result, there could be a trade-off in terms of the kind of connectivity and the communication rates accessible to that integrated platform.

The low-cost embedded systems also lack several cryptography features. There is a Microprocessor, but it is restricted. And unless it has been built that way from the start, there's generally no extra cryptographic hardware on that device. You won't be able to add or update cryptographic features using the equipment on that machine. Most of these purpose-built machines will not have a real mouse or keyboard, and others don't even have a video interface, making them difficult to upgrade. It's possible that you won't be able to link to this gadget over the networks to update the software. You may also need to go straight to the device to put in a USB disk or do another sort of software change. Security is frequently an afterthought on such microcontrollers. It's possible that no authorization is needed to unlock the system's software, or that the authorization is relatively restricted. With our portable devices and personal computers, we're accustomed to features like multi-factor verification and linking to file systems. However, these are features that may not be available on an integrated platform.

These microcontrollers also serve a very specialized purpose, and it's uncommon for them to have any other capabilities. Because it was explicitly designed and programmed to manage that particular function, this is not typically an instrument that allows users to add on enhanced capabilities after the essence. Supplementary functionality was not planned at the outset. Having an embedded processor that completes a specific task makes it possible to design it particularly for that component and nothing else. Designers can control costs by constructing this way, which would be incredibly significant if we're making a lot of such microcontrollers. We must also consider how using lower-cost elements may impact the device's quality. We may be restricting the lifetime of that specific arduino microcontroller by utilizing lower cost elements. You will not have immediate access to the control system or application on these microcontrollers. It may also be important to have entry to that device's computer system. We would like to be apt to evaluate that equipment from a safety perspective, assuring the system software is updated or that there are no privacy concerns. However, if you will not have direct exposure to the necessary hardware, completing the task will be incredibly challenging.

When writing code for an embedded system, it is critical to focus on the crucial points of readability and optimization. A code that is readable can be considered above optimization in case there is sufficient memory. For the freshly designed product, several verification steps are necessary (IEC 61508, ISO 26262, EN 50657, EN 50128, and IEC 62304 among others). The International Electro-technical Commission (IEC) 62304 is a key worldwide standard, particularly for medical equipment. A basic PCBs certificate, on the other hand, may cost hundreds of euros [13]. The program must talk with the analogue detector through SPI and I2C in order to adapt it. The process will not be accomplished if the digitized detector is not regulated by software. Operating systems, Timeline, engineering group average skill, software itself, microcontroller, development tools, interfaces, and other hardware are among the issues that create difficulties throughout the integrated system design process. Cost, processing power, memory, and energy usage are the main constraints. Hardware vulnerabilities e.g. Specter and Meltdown that are caused by processor hardware deficiencies are a major problem for embedded systems remedies.

The majority of illnesses and medical issues are caused by our lives, and embedded systems may assist us in reshaping our everyday behaviours and overhauling our lifestyles, especially because most people don't have time to see their physicians for frequent check-ups. As a result, smart embedded technology-oriented clinical devices, most of which follow the format of wearables, have been created to help individuals become more aware of their own health by allowing people to analyse their pulse rate, hypertension, glucose, weight, and a variety of other indicators. FitBit is an example of a product that has swept the fitness industry. It is among the most sophisticated embedded software examples available. It may be worn as a bracelet and records all of our health indicators, such like hypertension, weight, and sleep, to help us achieve our fitness and health objectives. Future health issues are less probable for such consumers. Micro devices with embedded systems are presently capable of collecting patient data and making control decisions that may aid in providing patients with effective treatment and drugs as their size and processing capacity decreases. In a nutshell, this technique eliminates the need for a doctor to come to the patient's house and check his signs. With the introduction of these designed automated technologies, not only is the standard of healthcare improved, but also the expense of medical therapy is reduced. Embedded systems, on the other hand, offer a plethora of uses in a wide range of fields, especially medicine and health. With artificial intelligence and internet of things on the horizon, embedded systems' future prospects are greater than ever, and we can only anticipate additional uses of embedded systems, making medical services easier and accessible than ever before. Embedded systems and medical applications are the subject of this article.

III. CONCLUSIONS

Present and emerging characteristics indicate which the embedded software will proceed to be applied in clinical applications. Commercial endpoint electronics with diverse sensors are self-calibrated and fault-tolerant platforms, which strive to be able to affect the external environment and provide vital datasets, have a production process whenever effective, and have a virtualized and conceptual sensor approach. The wide-range businesses establishing embedded systems will incredibly increase, and the marketplace will expand. It is envisaged that new and improved editions of the graphic cards will be released. Because of their mobility, embedded systems will become more popular in health care due to the rise in software packages and cheap cost. With networks connected with embedding developments and advanced software, it is also expected that virtual learning implications in healthcare will continue to grow. Any other requirements; recordings, wireless transmission, and electronic transmission (phone, tablet, computer, etc.); monitoring of key health indicators (sleep, epilepsy, heart, etc.); Devices that can be worn - Flexible electronic gadgets with energy collecting and storage capabilities, innovations that can transmit information to the internet and receive instructions, and wearable Internet of Things compatibility; Safeguarding individual-real-time in vivo measuring schemes; Sensors and gadgets that improve portability tele-rehabilitation, tele-radiology; frameworks for developing preventative and public health programs and individualized clinical monitoring equipment; computer-cloud interface; bio-signal collection and analysis; digital realities mechanisms and three-dimensional treatment and learning simulators.

References

[1]. S. Balandin and M. Gillet, "Embedded Networks in Mobile Devices", *International Journal of Embedded and Real-Time Communication Systems*, vol. 1, no. 1, pp. 22-36, 2010. Doi 10.4018/jertcs.2010103002.

[2]. O. Rübenkönig, Z. Liu and J. Korvink, "Integrated Engineering Development Environment", *The Mathematica Journal*, vol. 10, no. 3, 2007. Doi: 10.3888/tmj.10.3-8.

[3]. Φ. Γώττας, "Development and Integration of Teleconference services in Moodle", *Διεθνές Συνέδριο για την Ανοικτή & εξ Αποστάσεως Εκπαίδευση*, vol. 6, no. 2, 2016. Doi: 10.12681/icodl.665.

[4]. D. Balaso, "Automatic College Bell Using Raspberry-Pi", *International journal of Emerging Trends in Science and Technology*, 2016. Doi: 10.18535/ijetst/v3i06.06.

[5]. F. Hamady, A. Kayssi, A. Chehab and M. Mansour, "Evaluation of Low-Power Computing when Operating on Subsets of Multicore Processors", *Journal of Signal Processing Systems*, vol. 70, no. 2, pp. 193-208, 2012. Doi: 10.1007/s11265-012-0697-z.

[6]. P. Duangmanee and P. Uthansakul, "Clock-Frequency Switching Technique for Energy Saving of Microcontroller Unit (MCU)-Based Sensor Node", *Energies*, vol. 11, no. 5, p. 1194, 2018. Doi: 10.3390/en11051194.

[7]. A. AL-Marakeby, "FPGA on FPGA: Implementation of Fine-grained Parallel Genetic Algorithm on Field Programmable Gate Array", *International Journal of Computer Applications*, vol. 80, no. 6, pp. 29-32, 2013. Doi: 10.5120/13867-1725.

[8]. G.W.A.D., "Application specific integrated circuit (ASIC) technology", *Microelectronics Reliability*, vol. 32, no. 3, pp. 447-448, 1992. Doi: 10.1016/0026-2714(92)90076-w.

[9]. S. Akbar, B. Henryranu, M. Handono and A. Basuki, "Implementasi Purwarupa Perangkat Rumah Cerdas Pervasif Berbasis Protokol Universal Plug And Play (UPnP) Dan Raspberry Pi General Purpose Input/Output (GPIO)", *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, p. 116, 2015. Doi: 10.25126/jtiik.201522143.

[10]. W. Castro, "Residential design automation with Arduino use", *Revista Acta Científica*, vol. 8, 2017. Doi: 10.21745/ac08-15.

[11]. A. YUAN and M. FU, "Research of the method of software/hardware partitioning in the embedded system", *Journal of Computer Applications*, vol. 28, no. 9, pp. 2427-2429, 2009. Doi: 10.3724/sp.j.1087.2008.02427.

[12]. X. Li, R. Ni, J. Chen, Y. Lyu, Z. Rong and R. Du, "End-to-End Network Slicing in Radio Access Network, Transport Network and Core Network Domains", *IEEE Access*, vol. 8, pp. 29525-29537, 2020. Doi: 10.1109/access.2020.2972105.

[13]. S. Choi and Y. Kim, "Proposal for Improving Software Classification method in the 2nd Edition Revised Works of IEC 62304", *Journal of the Institute of Electronics and Information Engineers*, vol. 58, no. 5, pp. 47-58, 2021. Doi: 10.5573/ieie.2021.58.5.47.