ISSN No:-2456-2165

Application of Object Oriented Programming Paradigm in Embedded Systems Development

[1]Kingsley U. Eze, [2]Godswill Okeke. and [3]Fergus U. Onu
Department of Computer Science, Ebonyi State University Abakaliki

Abstract:- This paper is to review the application of object oriented programming paradigm in embedded system development. An embedded system is a computer hardware system with program that is composed of microprocessor/microcontroller, designed to perform dedicated functions, either as a standalone system or as a part of a large system. The paper first reviewed the various sub-systems that make up the embedded systems and explained the classes of embedded systems. The sub-systems include microprocessor or microcontroller, ROM, RAM and program. Also the paper reviewed the trend in programs, starting from low level language (Assembly Language) to high level language which has been integrated with embedded system as it evolved from the early stage to the current stage. The paper went further to review the benefits of object oriented programming languages and its application to embedded system.

Keywords:- *Object Oriented Programming, Embedded Systems, microcontroller, real time operations.*

I. INTRODUCTION

An embedded system is a computer hardware system with program that is composed of microprocessor/microcontroller, designed to perform dedicated functions, either as a standalone system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations. Complexities range from a single microcontroller to a suite of processors with connected peripherals and networks; from no user interface to complex graphical user interfaces. The complexity of an embedded system varies significantly depending on the task for which it is designed [1].

Embedded systems development was earlier carried out using low level programming languages like Assembly language and later C language. Assembly language has good features which made it suitable for embedded system development at that early stage. Assembly Language is fast in operation because its codes are closer to the machine language than the high level language such as the C language. C language as a high level programming language is coded in human readable language which makes programming a lot easier than the machine language and assembly language. The high level languages have also evolved from the early stage of the procedural programming languages to the era of object oriented programming languages.

Object-oriented programming (**OOP**) is a programming paradigm that deals with the concept of "objects". These objects can contain data and codes. The data are in the form of fields and the codes are in the form of procedures. According to [2] object oriented programming models objects and their interaction in the problem space and the development of a system based on these objects and their relationships. The object oriented approach to programming makes software development processes simple and easy. It came with the concepts of object, class, encapsulation, inheritance, abstraction, and polymorphism. Prominent among the Object-oriented programming languages are C++, Smalltalk, Java, Ruby, Visual Basic .NET, and Python.

A system is formed by combining different units or subsystems to carry out a unique function. An embedded system is the combination of software and hardware components to accomplish a certain task within a certain period of time and sometimes additional mechanical or (and) electronic parts are required. In [3] any computing system that is not in the form of the regular desktop pc, mainframe, laptop or other handheld devices are defined as embedded systems. They are always part of different electronic gadgets, automobiles, home electronics, industrial machines, office equipments and even toys.

II. LITERATURE REVIEW

Embedded systems are specialized computer systems that are embedded in application environments or other computer systems and offer specialized support. According to [4] embedded systems are small sized independent systems built on microcomputers, which control different equipments. These equipments are of a relatively wide range and play an important role in everyday life. We meet them in cellular phones, in microwave ovens, in cars, in video recorders, watches, toys and in several other equipments. From the onset till now, the microprocessors have different limitations which include memory size and size of data bus. The use low level programming language was another limitation because of the tediousness of the processes. Embedded systems development has gone through series of evolution both on the hardware and software/firmware sub-systems.

According to [5] an embedded system is an electronic or computer system which is designed to control, access the data in electronics based systems. This system includes a single chip microcontroller such as cortex, ARM and also microprocessors, FPGAs, DSPs, and ASICs. If we take any engineering product that needs control and if a computer is incorporated within that product to undertake the control, then we have an embedded system.

Embedded systems development started with single microprocessor and machine language programming (low-

ISSN No:-2456-2165

level programming language). According to [6], Intel released the first commercially available microprocessor unit, the Intel 4004, an early microprocessor that still required support chips and external memory. The 4004 had a 4-bit processor. The instructions were stored in ROM (read-only memory), while the data was stored in an external register[7]. Figure 1 shows the basic structure of an embedded system.

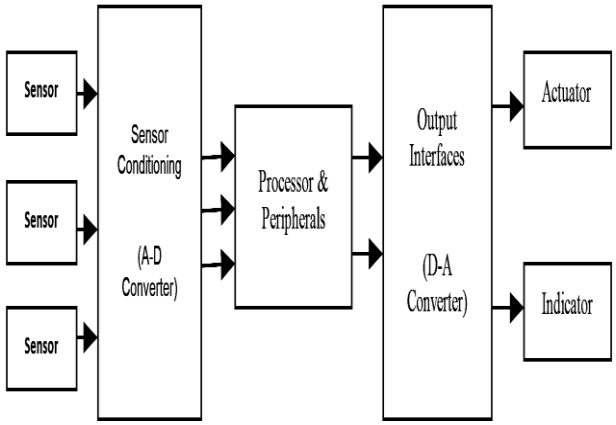

Fig. 1: The Basic Structure of an Embedded System [1].

The basic structure of an embedded system includes the following components:

- Sensor: The sensor measures and converts the physical quantity to an electrical signal
- A-D Converter: An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- Processor & ASICs: Processors assess the data to measure the output and store it to the memory.
- D-A Converter: A digital-to-analog converter changes the digital data fed by the processor to analog data
- Actuator: An actuator compares the output given by the D-A Converter to the actual output stored and stores the approved output.

III. MICROPROCESSORS/MICROCONTROLLERS

The basic hardware component of the embedded system is the microprocessor or microcontroller. Microcontroller and Microprocessor appears to mean the same thing but there is much difference between these two ICs. Microprocessor only have CPU in the chip like most of the Intel Processors but Microcontroller also have RAM, ROM and other peripherals along with the CPU or processor

Microprocessor essentially is an integrated circuit that has all the features of a central processing unit of a computer. A microprocessor is a multipurpose, programmable logic device that operates on binary instructions stored in the memory which accepts binary data as input and processes data based on the stored instruction to give an output. Figure 2 is a basic block diagram of a microprocessor.

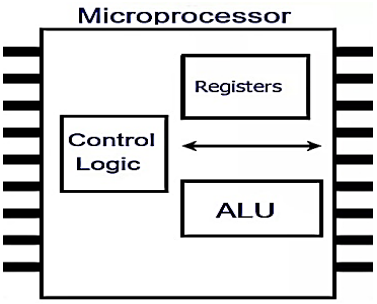

Fig. 2: Basic Block Diagram of a Microprocessor [8].

The first generation of microprocessor was introduced by Intel Cooperation between 1971and 1972. It was known as Intel 4004. It was a 4-bit processor implemented on a single chip. This processor was able to perform basic arithmetic and logical operations such as addition, subtraction, Boolean OR and Boolean AND. The control unit was capable of performing control functions such as fetching instructions from the memory, decoding it, and sending control pulses to execute it. This was followed by the second generation in 1973. Intel introduced an 8- bit microprocessor with the name Intel 8008. It could perform arithmetic and logic operations just like the first generation. It has 8-bit word length. An improved version of the 8-bit processor known as 8088 was introduced as well. In 1978 Intel's 8086, Zilog Z800 and 80286 were among the third generation of microprocessors introduced. They were 16 bit processors with a performance like minicomputers. Subsequently the fourth generation microprocessors with 32-bit were introduced by several different companies. The most prominent among them is the Intel 80386. The fifth generation started from 1995 to date. After 80856, Intel produced Pentium processor followed by Pentium Pro CPU, This allows multiple CPUs in a single system multiprocessing. They are 64-bit processors. There are other improved versions of the 64-bit processors like the Dual, Quad, Octa Core processors

Microprocessors have some shortcomings which includes

- The microprocessor has a limitation on the size of data.
- Floating point operations are not supported by most microprocessors.
- Microprocessor heats up easily.
- ROM, RAM and other internal peripherals are not part of Microprocessors.

Some of the shortcomings of microprocessors led to the emergence of microcontrollers. A microcontroller is an all in one system with processor, memory and peripherals which can be used to carry out specific task of embedded system. In embedded system computational hardware are often built around a microcontroller, MCU rather than a microprocessor, MPU. Both the microprocessor and microcontroller are used but typically the microcontroller is used in applications where size, power requirement and cost are major constraints. Microcontroller is implemented in a way that the entire processing engine is within one chip. This gives the microcontroller the advantage of small size and low cost. Figure 3 is a basic block diagram of a microcontroller.

ISSN No:-2456-2165

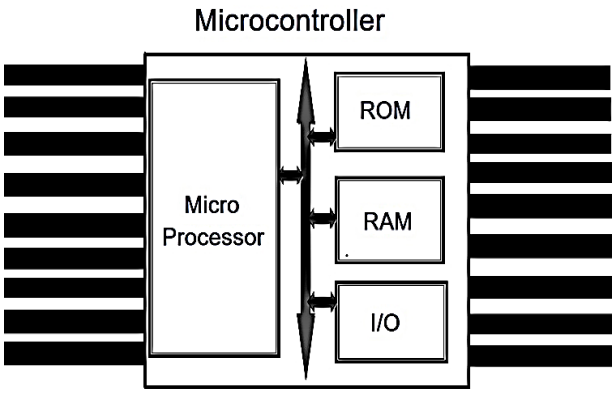

Fig. 3: Basic Block Diagram of a Microcontroller [8]

IV. PROGRAMS/SOFTWARE

At the early stage of the embedded system development, the programming was based on low level/machine language. Machine language is the language that the computer understands. It is written in binary and can be represented in hexadecimal for it to be a bit easy to ready. Still at the early stage, assembly language which is also a low level language was used for embedded system programming before the advent of the high level programming language.

A. Low Level Language(LLL)- Assembly Language

An assembly language is a type of low-level programming language which is closer to the machine language and can communicate with the computer hardware after assembling process. Unlike machine language, which consists of binary and hexadecimal characters, assembly languages are designed to be readable by humans. All programs must be converted to machine language for the computer to be able to understand the codes. The difference is that the instructions, variables and addresses have names instead of binary/hex numbers. Programs written in Assembly language must be converted (assembled) for computer to understand them. Figure 4 describes the conversion of assembly language code to machine code.

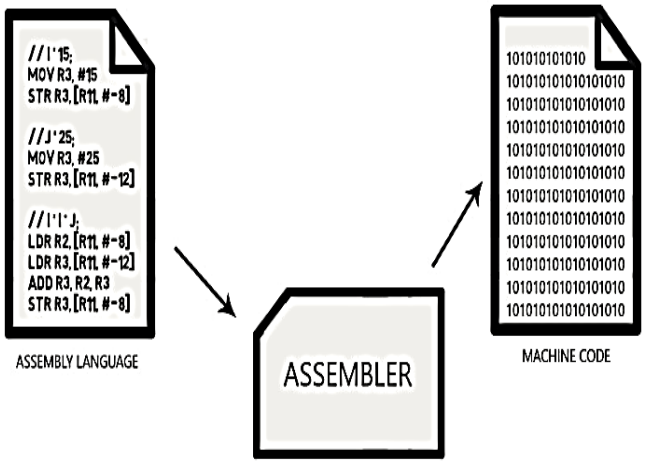

Fig. 4: Conversion of Assembly Language Code to Machine Code. [9]

Shortcomings of the low level languages as follows:

- Program written for one processor type in assembly language will not run in another type of processor.
- Assembly language program code is difficult to understand and difficult to debug than that of a higher level language.
- Low level languages use symbols and lacks variables and functions which can work directly with the processor

B. High Level Programming Language -Object Oriented Programming Languages

A **High Level Language** (**HLL**) is a programming language such as Java, C++ or Pascal which can be used to write programs that can be implemented on different types of computer. The program is written in human readable format that will require translation using a compiler or an interpreter for it to be in computer understandable format.HLL is easy to write, read and maintain.

The early stage of the HLL was hinged on the procedural language paradigm. Pascal, BASIC and C language are good examples of the procedural language. As the procedural language became widely used by programmers, the Object Oriented Programming paradigm was introduced. It is important to state that many of the most widely used programming languages (such as C++, Java, Python, etc.) are multi-paradigm.

Object-oriented programming is based on hierarchy of classes, and well-defined cooperating objects [10]. Object-oriented programming adopts a technique where there is an initial plan for the functions and operations that will be applied to data structures before writing any code. Object-oriented programming follows the approach by which real world processes work. It is easy to map a real-world problem to a solution in Object-oriented programming. Programs can be built from the working modules which interact with one another, rather than starting the code again from scratch [10]. Programming task can easily be broken into parts based on objects. Code reusability is a common practice in OOP. Productivity is increased and costs are reduced. These, are some of the benefits of Object-oriented programming over other programming techniques like Procedural. Object-oriented languages are used in the development of most of the software and applications that are in use today. Some of these languages are Java C++, Python Ruby etc..

High Level Languages have numerous benefits. They include but not limited to the following:

- They are programmer friendly.
- It provides higher level of abstraction from machine languages.
- It is not a machine dependent language.
- Easy to learn. It is written in human readable format
- Less error prone, easy to find and debug errors.
- Programming in High Level Language is more productive.

ISSN No:-2456-2165

V. CLASSIFICATION OF EMBEDDED SYSTEMS

Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller. Figure 5 shows details of this classification.

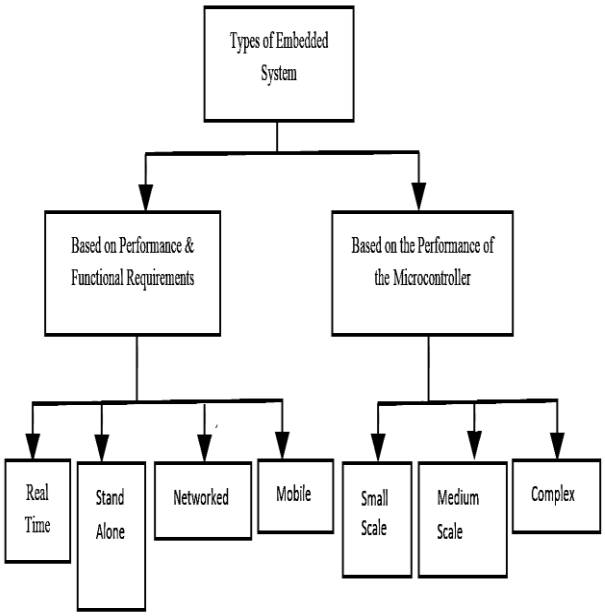

Fig. 5: Classification of Embedded Systems [11]

Embedded systems are further classified into four categories based on their performance and functional requirements:

- Stand alone embedded systems
- Real time embedded systems
- Networked embedded systems
- Mobile embedded systems

Embedded Systems are classified into three types based on the performance of the microcontroller such as

- Small scale embedded systems
- Medium scale embedded systems
- Sophisticated embedded systems

VI. SMALL SCALE EMBEDDED SYSTEMS

This is a low performance embedded system which is relatively cheap to produce. Microcontroller of 4-bit or 8-bit is used to design Small Scale Embedded Systems. The clock frequency ranges from 0,1 MHz to 4-MHz.The power source can be battery and other system resources such as the memory are low. The memory is within the range one hundred bytes to two Kbytes and the Input/output ports are generally digital. This type of embedded system is programmed using assembly language. The main tools required for the program development are an editor, assembler, cross assembler and integrated development environment (IDE). They usually do not act as independent system but as component of the system and they are used for implementation of specific task. . A good example of this type is the MicroChip16f84

A. Medium Scale Embedded Systems :

This is a mid-range performance embedded system that is based on microcontroller of 16-bit or 32-bit design a good price/performance ratio. The processor is RISC architecture with clock frequency that ranges from 4-8 MHz to 40 MHz. The RAM size ranges from 2 to 8 Kbytes and the Input/output ports are both analogue and digital. Hardware and software are getting complex. Programming is carried out in high level programming languages such as C, C++ and Java. Software programming tools used in medium scale embedded system are compilers, IDE, editors, debuggers, simulators, RTOS, source code engineering tools etc. Good examples of processor types in use for this implementation are the older Intel 8051, MicroChip 16f87x, ATMEL ATmega type and Motorola 68.

B. Complex Embedded Systems:

This is a high performance embedded system. Microcontrollers of 32-bit or 64-bit are used to design Complex Embedded Systems and may involve the use of ASIPs, IPs, PLAs, scalable or configurable processors. In this implementation, no one-chip application is possible. The systems are close to the lower end of the PC categories. This type of embedded systems has more hardware and software complexities than the medium scale embedded systems. They are developed to perform large scale complex functions. Both hardware and software components are used to design final systems or hardware products. PC104 is a typical application example of complex embedded system. It is a lower performance PC mother board with104 expansion connection points. This is compatible with the PC world; it has capacity to use software developed for PCs.

VII. DISCUSSION

Embedded systems were initially implemented with processors of 4bit to 8bit architecture and with limited memory. At this stage majorly Assembly Language (low level language) was used to program the embedded system. As technology evolves there was a change from the use of low level. The use of High level programming language like C programming became prevalent over time. C language from inception was implemented as a procedural language. Being high level languages it defers from the Assembly language in the following ways:

Program written in C language in not machine dependent. It could be compiled and ran in different architectures as against the Assembly Language which is machine dependent.

Assembly language program code is difficult to understand and difficult to debug than that of a higher level language.

Programmer do not need to know in details about the hardware like registers in the processor when writing program in high level language as against such requirements when developing in assembly language.

ISSN No:-2456-2165

Low level languages use symbols and lacks variables and functions which can work directly with the processor

Today processors used in Embedded Systems are either in the form microcontrollers or microprocessors with design architecture as high as 64bit. The microprocessor does not have any internal peripheral like ROM, RAM and other I/O devices which are present in microcontrollers.

With this advancement in technologies which led to availability of the various system resources such as faster processor architecture with bigger memory capacity, embedded systems developer delights in C language because its memory management capability. It allows the user to manage program memory as it offers the feature of dynamic memory allocation which makes it much faster than any other language. The implementation of Object Oriented Programming is now on the increase because of the features inherent in them. The features of Object Oriented Programming paradigm includes but not limited to the following:

- Modularity which enhances easy troubleshooting
- Inheritance encourages reuse of codes
- Polymorphism promotes flexibility
- Problem solving is more effective
- Data hiding and abstraction mechanism enhances security
- High productivity is achieved

C++ being the object oriented version of C language became one of the most used object oriented programming languages deployed in embedded systems. By design, C++ is very useful to embedded development because the language has features that make it to sits in between higher-level software and hardware. This enables the programmer to access and control hardware components directly without losing the benefits of a high-level language. C++ has many features that can be used very effectively in embedded programming. They include RAII, namespaces, templates, const, constexpr, enum class etc.

Future holds a lot to the embedded systems application. It has been applied in the development of products like digital watches, microwave ovens, washing machine, hybrid vehicles, avionics, credit card readers, cell phones etc. Most microprocessors/microcontrollers manufactured are implemented in embedded systems. The industry is growing rapidly. It is being applied to the development of emerging technologies like Internet of Things (IoT), machine learning, Artificial Intelligence (AI), Virtual Reality (VR) and Augmented Reality (AR) and deep learning.

VIII. CONCLUSION

The application of object oriented programming paradigm in embedded system development has come to stay as long as we live in this era of technology where almost everything around us are intelligent and designed to interact with each other. For this to be achieved, the components that drive the intelligence must be embedded. The key components involved are the software and the microprocessor/microcontroller. Both the software and processor have evolved over time. At each level of the evolution, betters features are incorporated to the system. Object oriented programming is one of such new developments in programming which has contributed much to the embedded system development.

REFERENCES

[1.] Omni.Sci, Embedded Systems, https://www.omnisci.com/technical-glossary/embedded-systems, retrieved 24/10/2021
[2.] Poo, D. and et al (2008), Object-Oriented Programming and Java, Springer, London.
[3.] Frank Vahid and Tony Givargis (2001), Embedded System Design: A Unified Hardware/Software Introduction, John Wiley & Sons, USA.
[4.] Marwedel, P. (2005), Embedded System Design, Springer, Germany
[5.] Embedded Systems Role in Automobiles with Applications, www.watelectronics.com,https://www.watelectronics.com/importance-of-embedded-systems-in-automobiles-with-applications/, retrieved 27/10/2021
[6.] Intel's First Microprocessor, www.intel.com,https://www.intel.com/content/www/us/en/history/museum-story-of-intel-4004.html, Retrieved 25/10/2021
[7.] Michael Barr (1999),Programming Embedded Systems in C and C++, O'Reilly and Associates, USA
[8.] Components101, (October 2019). Difference between Microprocessor and Microcontroller. Retrieved from https://components101.com/articles/difference-between-microprocessor-and-microcontroller
[9.] EDUCBA. Machine Language vs Assembly Language. Retrieved from https://www.educba.com/assembly-language-vs-machine-language/. Retrieved 22/10/2021
[10.] UK Essays. (November 2018). Object Oriented Programming. Retrieved from https://www.ukessays.com/essays/information-technology/object-oriented-programming-what-is-object-information-technology-essay.php?vref=1
[11.] Watelectronics.com. Classification of Embedded Systems. Retrieved from https://www.watelectronics.com/classification-of-embedded-systems/. Retrieved 23/10/21