J. Integr. Sci. Technol. 2023, 11(3), 524 Article

Journal of Integrated
SCIENCE & TECHNOLOGY

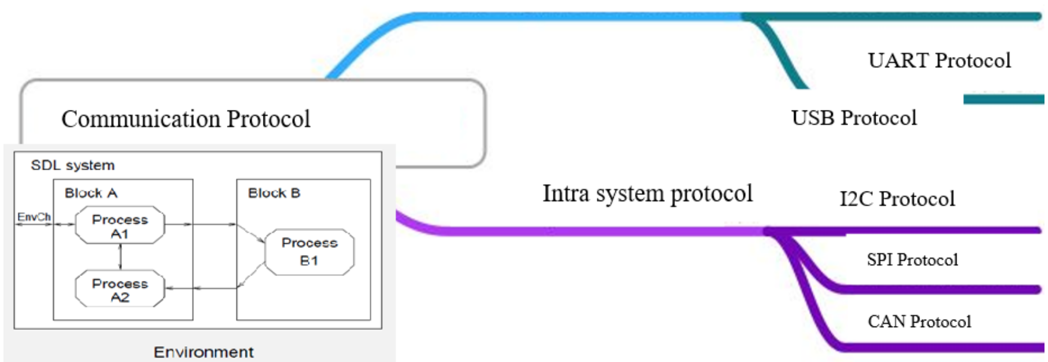
The Integrated SDL-based design approach to create and implement wireless communication protocol

Mhamane Sanjeev Chandrashekhar,[1] Amol Kumbhare[2*]

Electronics and Communication Engineering, Dr. APJ Abdul Kalam University, Indore, India

Received on: 12-Dec-2022, Accepted and Published on: 24-Feb-2023

ABSTRACT



Wireless sensor networks (WSN) and personal area networks (WPAN) are being increasingly used in data transfer or communication protocols. The High-end embedded systems sometimes contain a VME (Virtual Memory Environment) backplane or a comparable arrangement. The media access protocol of a local area network (LAN) arbitrates (selects the subsequent transmitter for) access to the shared network medium (typically a wire, fiber, or radio frequency). SDL (Specification and Description Language) is growing popular for the use in creation of communication protocols. In this study, the particular difficulties in networking real-time embedded devices have been discussed with details about a few protocols that show different methods for accessing the shared material. Connection-Oriented Protocols, Polling, Time Division Multiple Access (TDMA), Token Ring, Token Bus, Binary Countdown, and Token Bus are among the protocols addressed. In terms of validation and definition, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and SDLs are identical. The study has examined each of these protocols' benefits and drawbacks in light of the various criteria. Users can choose a protocol that suits demands with the help of the protocol tradeoff chart. One of the most flexible protocols for embedded systems is CSMA/CA, although only one protocol may be suitable for a given application.

Keywords: Model-based design, simulation, IEEE 802.15.3 protocol engineering, SDL, CSMA

INTRODUCTION

A collection of communication protocol guidelines enables data transmission between two or more communication systems across any physical media. The term "protocol" refers to the rules, guidelines, synchronization between communication systems, required syntax, and semantics. Protocols can be implemented using both hardware and software. Different communication protocols are commonly used in analogue and digital communication systems. Each protocol also has a specific set of applications. An electronic system or device that uses hardware and software is called an embedded system.[1] A processor or controller receives input from devices that are part of the physical world, such as sensors, actuators, etc., processes that information using the right software, and then produces the desired result. In this instance, for the components to deliver the desired result, they must interact. A protocol is needed for each communication entity to communicate information. Embedded systems can use various protocols implemented according to the application domain. Generally, the physical layer of a communication protocol describes the signals, their strength, how they are shaken, how buses are arbitrated, how devices are addressed, and whether they are wired or wireless. The application layer involves system configuration, baud rate choice, and data transmission and reception. Since embedded systems sometimes come with high costs which might impact their adaptability in real scenario and accordingly these must be very reasonably priced.[2]

*Corresponding Author: Mr. Mhamane Sanjeev Chandrashekhar
Electronics and Communication Engg. Dr. APJ Abdul Kalam University, Indore, India
Email: sanjeev.mhamane4@gmail.com

Cite as: *J. Integr. Sci. Technol., 2023, 11(3), 524. URN:NBN:sciencein.jist.2023.v11.524*

 ISSN: 2321-4635
http://pubs.thesciencein.org/jist

The cost of an integrated circuit is influenced by the die size, which uses less chip space. Due to this limitation, embedded systems' computing and storage capabilities are much lower than those of modern personal computers. Both WPAN and WSN are growing (WSN). A few examples of automation include residential and personal safety, personal healthcare, logistics, traffic monitoring, manufacturing operations, and agriculture. Communication standards regulate the connection between the endpoints of a communication system.[3–6] Consequently, protocols constitute the foundation of many networks, including computer and wireless sensor networks. It works either in software, hardware, or both. Implementation has an effect on system responsiveness, protocol expansion adaptability, and bug patch adaptability. There is no implementation plan in the conceptual design of protocols.

Informal criteria for the protocol system

The range of applications for our methodology encompasses specification through implementation. We start by taking a look at the general system architecture of the wireless communication system known as InfoPad to give some background and inspiration for the data connection and media access protocols. The reader should be able to understand this method because it needs to address a variety of issues. Graphs of state transition systems were used to record data connection protocols.[7] It would have been simpler to understand what needed to be done and resulting in a significantly more efficient protocol implementation[8] if an abstract state machine language with clear message-passing semantics and language constructs intended to aid protocol construction had been employed.

Media access protocols

Relationship-focused protocols were widely used to link remote terminals to mainframes before LANs gained popularity. These protocols, which are generally connected using a modem and serial lines, only support two nodes per physical transmission media. A four-processor network utilizing this protocol is seen in Figure 1. Multiple transmissions through intermediary nodes are necessary for communication between nodes that are not physically linked. Between nodes that are directly linked, these protocols are deterministic. Indirect connections between nodes might result in significant latency. A lot of pass-through traffic-handling nodes may get inundated for demanding applications, making it impossible to utilize low-cost nodes in a big system.[9]

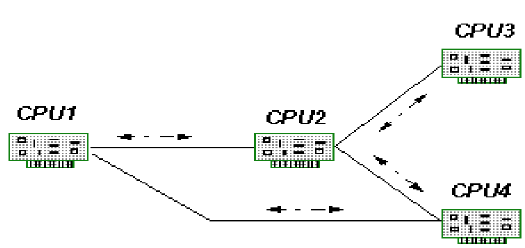


Figure 1 A network utilizing connection-oriented protocols as an example.

Time Division Multiple Access (TDMA)

TDMA, which is frequently employed in satellite communications, can be used in local area networks. In this protocol, a network master issues a frame sync signal before to each cycle of messages to synchronise the clocks of all nodes. As illustrated in Figure 2, each node broadcasts after the sync within the time slice that was given to it precisely. Since individual polling messages are no longer sent, performance is similar to polling but more resilient to heavy loads. The limitations of TDMA are exacerbated by the need for fixed-length messages to fit within time slices.[10]

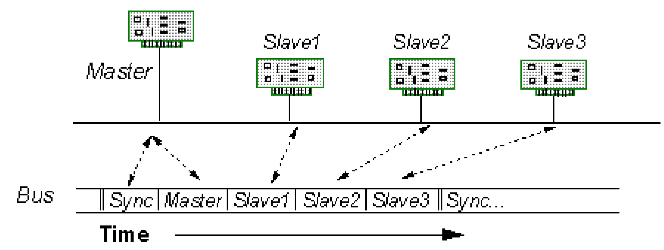


Figure 2: Time Slices of TDMA protocols

Types of Communication Protocols in Embedded Systems

Communication protocols are the skeleton linkages for an embedded system in the electronic world, thus it is crucial to comprehend what they are, why they are employed, and how they differ. The "protocol" defines the syntax, semantics, synchronization, and possibly error recovery techniques across communication systems.

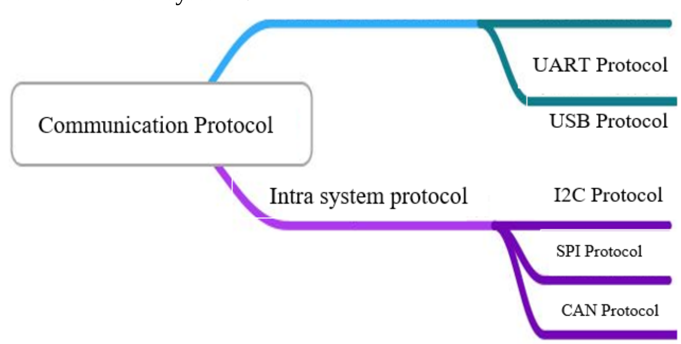


Figure 3. Inter system protocols facilitate communication between two devices.

Related Work

The current protocol engineering practices and Codesign of hardware and software for embedded systems forms the frontline of recent studies. The challenge of integrating current protocol engineering techniques with embedded system design methodologies is the next recurrent event.[11] The objective implementation provides a way to effectively implement communication protocols on small target systems with few resources. Designing effective, accurate protocols, putting them into practice, and integrating them into the entire system are the goals of creating embedded communication systems. All phases of the design flow should be supported by an integrated design process. TAU SDL Suite is a development tool for developing, simulating, validating, and implementing software and test protocols.[11]

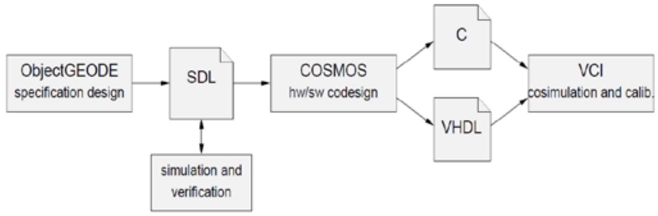


Figure 4. Hardware and software code sign flow is based on the COSMOS tool.

There are recent work advances that develops an integrated simulation environment with additional physical channel models, protocol layers, or the popular Unified Modeling Language (UML). On the other hand, the advantages of an SDL-based design cycle include a high level of abstraction,[12] independence from implementation, and formal semantics that allow protocols to be tested using SDL specifications. The TIMA laboratories developed the COSMOS tool, a codesign tool for hardware and software. SDL, an input language used by COSMOS, is based on the original work. Hardware/software partitioning is done manually by the user at the SDL process level.[13] Interface components are also constructed to link the hardware and programmed processes.

Designing systems with SDL

All necessary support components outlined in the fast prototype design cycle, including timers, inter-process communication, and environment interfaces, must be available in run-time systems. The system's real-time constraints are described using a preliminary SDL specification and a set of timing annotations. In the suggested solution, RTEMS, an open-source real-time operating system, is utilized. There are various approaches to design hardware implementations based on SDL needs. The primary tool utilized to create the software component is Telelogic's Advanced code generator.[14]

SDL-based implementations of communication protocols

The SDL run-time system used in Telelogic's light integration paradigm does not permit preemption, accordingly a tight integration strategy may be adopted. Using the tight integration technique, each SDL process is paired with an OS job. Therefore, jobs that are higher on the priority list can take precedence over tasks that are lower on the list. A real Telelogic does not already have an integration library for the Virtuoso OS, thus one had to be made. This adaption was an important outcome for systems because other operating systems lacked a universal template [DZM01]. The design specifications for the tight integration library were never publicly available.[15]

Partitioning of Hardware and Software in Embedded Systems

In the design of software-centric systems, the V-cycle is frequently employed, with the construction phases (requirements, analysis, design, and implementation) being followed by the verification stages (such as tests and formal proofs). The V-cycle can obviously only begin for embedded systems when functions have been divided into hardware and software. The Y-chart method is typically used for system partitioning [BWH+03]. The end result of this process is an ideal hardware/software architecture.[16]

There is a need to ensure that there are no deadlocks or "live locks" in the protocol, so the informal requirements of the system need to be incorporated into a formal specification towards verification of the existence of these qualities. The protocol need to be evaluated based on a variety of environmental statistics models to be able to figure out its effectiveness. It is best to consider the models utilised in the design process as well as in exploration, verification, and performance estimationstages as complementary viewpoints of the same system rather than as distinct perspectives.[17]

Developing an SDL-based integrated design flow:

• examining some of the more common problems that can arise when attempting to connect an instruction set simulator with an SDL simulation;

• Using the real-time operating system Reflex for devices with exceptionally little resource allocation and putting into practise the theoretical concepts for a successful run-time environment.

• Investigating different possibilities for a useful and closely integrated layer - often referred to as an SDL run-time environment - for the Telelogic C Advanced code generator. Using [Nol09] as an example, two of these accomplishments include successfully designing and implementing the IEEE 802.15.3 MAC protocol using recently developed tools and methodologies, as well as implementing a simulation framework that combines the Telelogic SDL simulator and the LEON2 instruction set simulator TSIM.

Problem statement

New techniques for testing and verifying the enhanced system are necessary, according to the HiPEAC embedded systems roadmap [V+06], because of the system's growing complexity. Up until 2009, it is anticipated that the combined annual growth rate for design automation tools and test automation tools would be around 20%. Complicated SoCs and embedded systems can take a long time to validate, which is a widespread practice in the semiconductor industry and accounts for 80% or more of the entire design time and cost [MED03]. The existing embedded systems that are being developed could not function as a result. As a result, one of the key problems in the field of electronic design automation (EDA) is the creation of new methods that can speed up the validation of embedded systems, which in turn results in devices that are inexpensive [V+06].[18]

Embedded Communication System Design

The definition of messages and the order in which they can be conveyed is a need for computer communication systems. This is similar to how people communicate with one another through the use of etiquette and accents. Protocols have been devised to facilitate seamless communication between devices made by different manufacturers. The standards don't specify a specific implementation approach; they rather provide abstract syntactical and functional descriptions.[19] Academic institutions and the electronic design automation (EDA) industry have shown a great deal of interest in the development of methods for the design of embedded systems and system-on-chip (SoC) architectures. Similar goals to those of protocol engineering were pursued in the development of system specification languages and design tools: to improve overall design quality and shorten the length of the product development cycle. The creation of tools that will simplify computer-aided design exploration and translate high-level descriptions of behavioral systems into low-level physical representations are currently receiving a lot of interest from academics.[20]

Protocols for wireless media access

When multiple people are simultaneously using the wireless channel then to control the sequential access of all devices to the channel, a multiple access mechanism must be in place. It also include the management of shared wired media. Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), and Frequency Division Multiple Access (FDMA) (CDMA), and space division multiple access (SDMA), as well as their combinations, are examples of multiple access systems that are often used. The transmitters in FDMA, CDMA, and SDMA are isolated from one another and can each employ a specific frequency, code sequence, or area. With TDMA, a single transmitter is given temporary access to the channel throughout the whole frequency range and space (Figure 5). Due to its straightforward approach, minimal signal processing requirements and suitability for wireless sensor networks topologies.

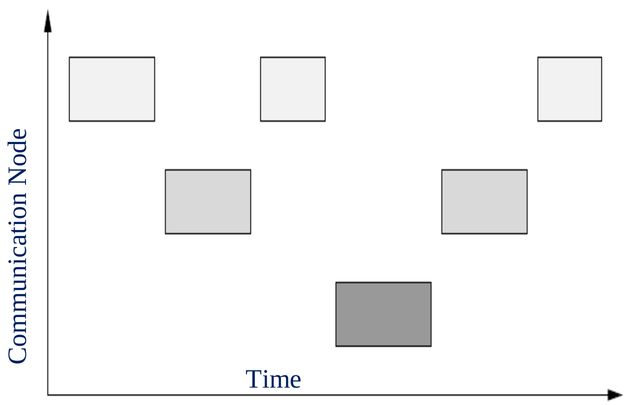


Figure 5. Communication nodes must alternate using the same wireless channel when employing the TDMA method.

The SDL specifications' structure

A document known as an SDL specification can be used to explicitly describe the behavior and structure of a system. The executable specifications are arranged in a hierarchical order. The object with the highest level is the system. Blocks are lower-level structural components that can include additional blocks or processes and are present in the system diagram. Processes themselves can be made up of these building components. Blocks are used to both give a specification structure and to initialize it; SDL processes are used to record a specification's behavior. Channels are connections that are required to enable communication between various blocks and actions. Packages could also be a part of the SDL standards. Packages are collections of reusable components that may include definitions of methods, data and signal types, process types, and block types, among others. The system definition permits instantiation of references to various sorts.[21,22]

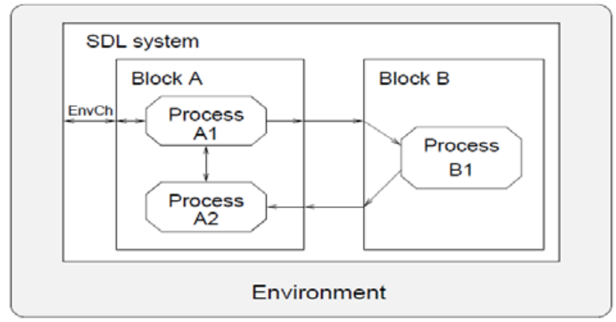


Figure 6. SDL system integrated into its surroundings.

Protocol embeds system development

There are several parallels between the protocol development process and software development. The methodical exposition of the development activities require the understanding of a few extra features that are unique to protocol engineering. In order to create networks of possibly disparate devices, communication protocols are utilized. This indicates that the same protocol is implemented and working on a variety of hardware and software platforms, i.e., on various processors and perhaps even utilizing various operating systems.[21,22]

Requirements analysis

The creation of a communication service and the protocol that delivers it begins, like any other technical endeavor, with a study of the criteria it must satisfy. Use cases or application scenarios are frequently the driving forces behind this study, which generally results in imprecise representations of system requirements. The Unified Modeling Language (UML) has been shown to be helpful in the formal specification of requirements during the software development process.[23] In the [dV02], it has been suggested to specify real-time system needs using UML.The normal range of the granularity of time is between 6 and 8 orders of magnitude. The media access protocol's low-level operations normally take place on the order of a symbol period, which for our systems is typically between 0.1 and 10 microseconds. Tight timing control at the modem interface is necessary for effective use of these modems. Due to the fact that it must manage a portion of the modem, the media access protocol interacts considerably more closely with the hardware. There is little to no data flow and virtually exclusively control-oriented state machines that determine this behaviour. SDL-like languages are beneficial for modelling the system at a higher level of abstraction but are not appropriate for constructing a cycle-accurate model of the implementation.

Implementations of communication protocols based on SDL

In the mapping procedure, processing units from the target architecture are manually mapped to SDL processes. The partitioning is chosen based on their "timing needs and computational complexity," according to [Mut02]. The CAdvanced code generator from Telelogic is the main tool used to create the software component. The server models produced by this C code generator are then used to implement the servers. All support functions that are not specifically mentioned in the SDL standard must be performed by a run-time system. Inter-process

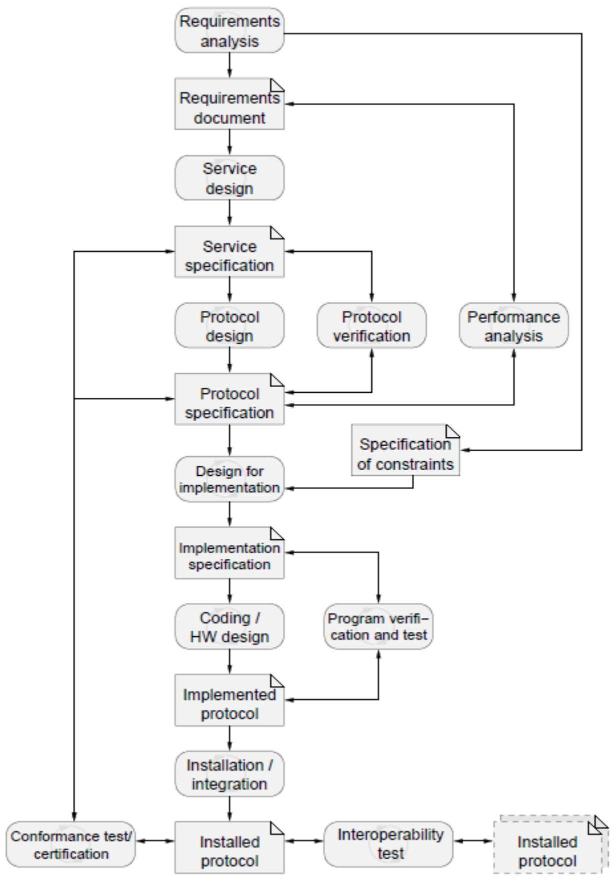


Figure 7. Phases of protocol development and their outcomes based on [K on 03].

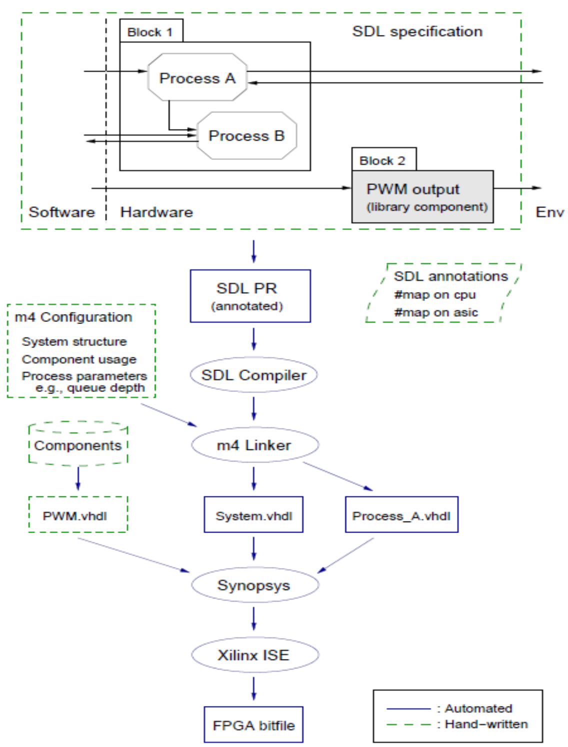


Figure 8. Muth's [Mut02] hardware synthesis method for a rapid prototyping system.

communication, timers, and environment interface are some of these features. The run-time system is implemented using the method outlined by Ahmed et.al.[24] using the RTEMS open-source real-time operating system.

The design strategy we advise for the embedded systems makes use of SDL, a high-level system description language. Two of the various approaches that have been mentioned in this discussion and that can be used in conjunction with our plan are formal verification and performance analysis. Our main goal has been to provide effective run-time environments for embedded devices. In order to enable hardware/software partitioning during the design space exploration stage, we also created a simulation methodology employing an instruction set simulator.[25]

SDL-based integrated design flow

Our approach shortens the time required for development and, at the very least, enables the automatic conversion of the model into an implementation that just requires software. Unlike other methods, which simply employ SDL as a tool for protocol specification and begin a brand-new software and hardware implementation effort without using the SDL model for synthesis, our technique uses SDL as a tool for synthesis. Unlike previous approaches, which merely employ SDL as a tool for protocol specification, this one uses it to specify protocols. The implementation accomplishes this in a way that allows it to maintain the model's formally validated properties. Deeply embedded systems' effective run-time environments enable them to operate with lower memory and processor overheads. Not to mention, extremely accurate profiling data are produced by modelling an implementation model using an instruction set simulator. By utilising the basic protocol model and integrating the instruction set simulator with a functional SDL simulation of a communication network, we are able to reduce the amount of time required to create test benches for the instruction set simulator.[26]

Implementation results

The amount of memory and processing power that the resultant code will demand on the target platform was calculated using straight forward SDL models. An arrangement of various processes that resembles a chain makes up one set of SDL models. The signal that the initial process, known as Ping request, receives from the environment and delivers to the next process in the chain is called another signal of the same type. Up to the moment at which the last link in the chain broadcasts a Ping. confirm signal, the receiver will carry on as usual. Then, this signal will be sent up the chain in reverse. As soon as the input signal that initiated the activity is received, the signals are transmitted without any processing lag. Finally, the environment receives the Ping.confirm signal.

The BASUMA, which stands for Body Area System for Ubiquitous Multimedia Applications, was chosen because its main goal was to create a body-area wireless communication system that would allow users to access multimedia applications while utilising very little power. The study has chosen the IEEE 802.15.3 standard as a result of a number of criteria. One of the initial requirements for the body area system was the exchange of MPEG-1 video streams. The supported data speeds of 11 to 55 Mbit/s enable this.

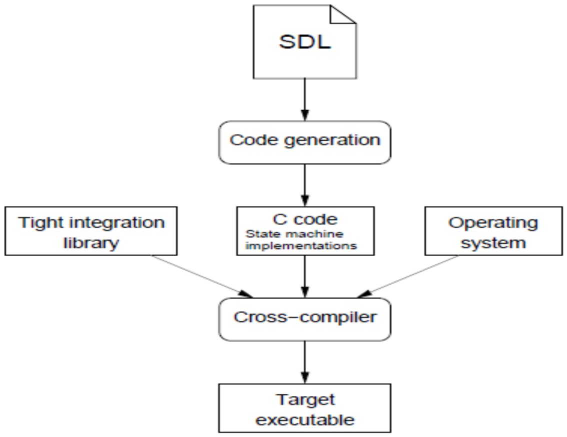


Figure 9. Integration of newly created software with a target operating system

Table 1. The light integration approach was used to implement the four SDL systems' necessary memory space and processing speed.

Required memory space	Light integration approach			
	Ping2	Ping4	Ping8	PingSave4
text	**12541**	**12416**	**14284**	**16114**
bss	**328**	**2578**	**1318**	**864**
data	**2021**	**232**	**562**	**242**
total	**13678**	**10316**	**14754**	**13620**
Execution time (5000 signals)				
ticks	**5,48,364**	**12,12,656**	**24,50,632**	**22,91,606**
seconds	**4.28**	**9.47**	**19.1**	**17.9**

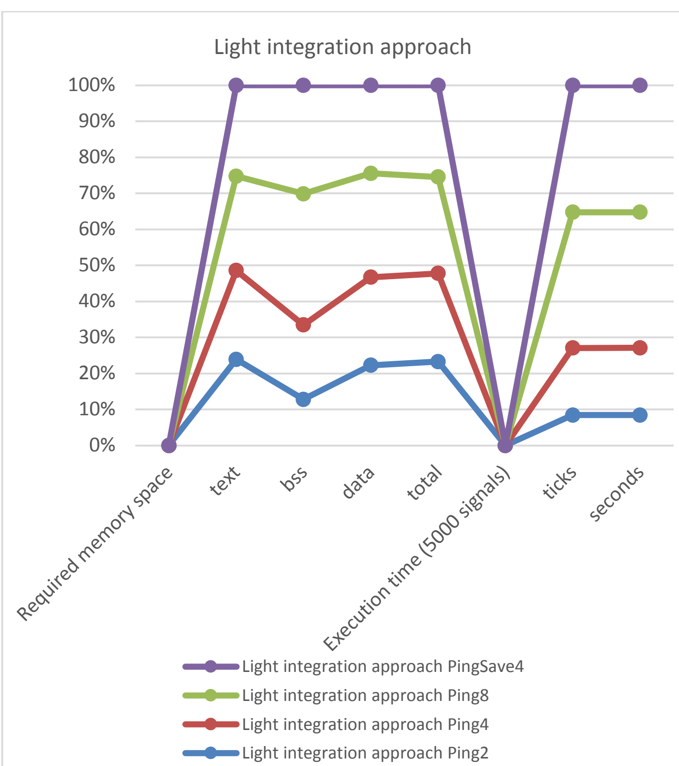


Table 2. Four SDL systems' necessary memory usage and processing speed were implemented using a tight integration strategy

Ping2 app.	**Light integration**			**Tight integration**		
	text	**data**	**bss**	**text**	**data**	**Bss**
SDL model	**1240**	**4521**	**3259**	**45**	**5601**	**452**
Environment	**15671**	**534**	**5640**	**630**	**570**	**365**
Reflex	**5423**	**843**	**365**	**1287**	**568**	**325**
Run-time system /	**8543**	**1257**	**9541**	**3210**	**1085**	**651**
integration library	**1524**	**12365**	**9541**	**10235**	**1562**	**4587**
libgcc and libc	**324**	**3654**	**128**	**745**	**985**	**624**
Total (Ping2.elf)	**158745**	**13524**	**986541**	**1235**	**547**	**3254**

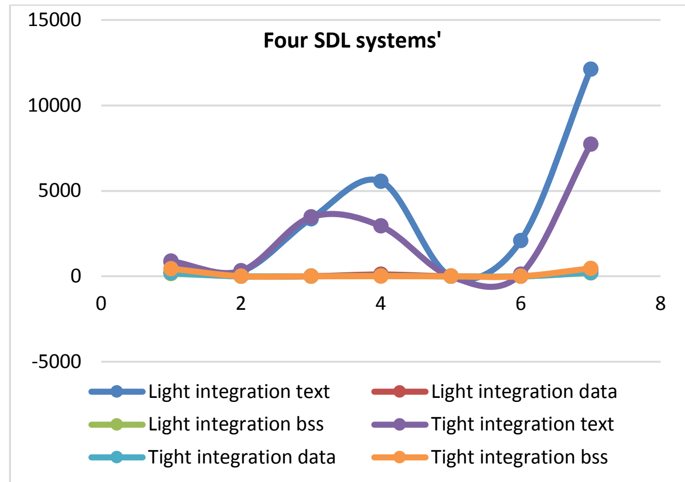
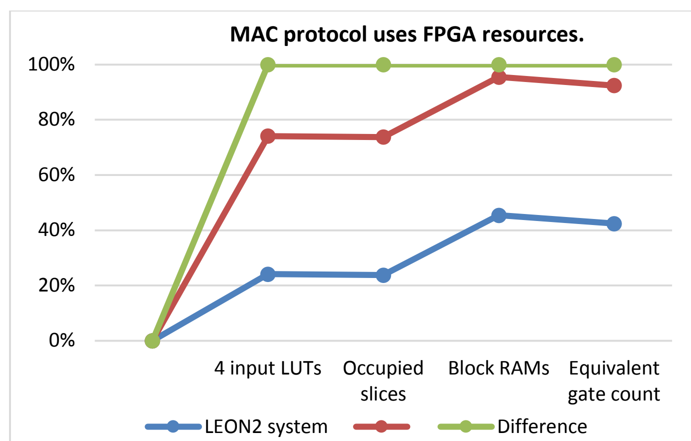


Table 3. System for the MAC protocol uses FPGA resources.

Resources	LEON2 system		Difference
	Original	**With prot. acc.**	
4 input LUTs	**15142**	**24513**	**12530**
Occupied slices	**48572**	**35647**	**3254**
Block RAMs	**52**	**54**	**5**
Equivalent gate count	**13254**	**45875**	**35421**



This embeded systems board, along with an FPGA-based baseband processor and an RF board, can be used to showcase an entire IEEE 802.15.3-compliant wireless system. All of its parts can be combined onto a single chip, enabling it to execute applications like real-time and multimedia that demand effective power management, rapid data transfer rates, and guaranteed service quality.

CONCLUSIONS

An embedded system is something like a wireless sensor node. The embedded microcontroller is yet another example. These systems feature limited power supplies, memory, and processing capabilities. They also need to be able to communicate with other electronic devices and work reliably for a time period that might be anywhere between months and years without needing any kind of maintenance. As a consequence, the software designers and communication protocols created for such platforms have a responsibility to use the energy that is readily accessible as efficiently as possible and must be free of design flaws that cause malfunctions or other unexpected behavior. We outlined a method for designing embedded systems that takes these kinds of needs into consideration in our thesis. SDL was chosen because of the high degree of abstraction it offers to imitate system behavior. SDL is often used in the design of protocols, hence this choice was picked. An automated tool has the ability to simulate SDL models, formally validate them, and convert them to C code. independently experiencing a transformation. As a result, locating delays in the implementation process is significantly easier. During the first phase, the functional SDL paradigm was originally put into practise. used as a testing environment for the implementation model simultaneously with the design of the implementation itself. There are many approaches to overcome timing constraints, including the creation of better software algorithms, an increase in clock frequency, or the creation of the most effective hardware solution. This automated the synthesis of VHDL from a high-level specification. In the end, we used our integrated SDL-based design approach to create and implement the challenging IEEE 802.15.3 wireless communication protocol. the MAC's preferred protocol. The LEON2 processor's target processor was chosen to be created utilizing this data. The software created by SDL model of this protocol can be executed on the LEON2 CPU. We were able to do a hardware/software split on the protocol and identify the required protocol accelerator capabilities with the aid of the simulation framework. The productive outcome validates the applicability of current strategy and demonstrates its efficacy.

CONFLICT OF INTEREST STATEMENT

The Authors have no conflict of interest for this work.

REFERENCES

1. M. Bharath, J. Indra, S. Kirubakaran. Design and Development of Virtual Lab Environment for Embedded System and IoT Applications. In *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*; IEEE, **2022**; pp 1–6.
2. J. Zidar, T. Matić, I. Aleksi, F. Sušac. Low Power Embedded System Sensor Selection for Environmental Condition Monitoring in Supply Chain. *Int. J. Electr. Comput. Eng. Syst.* **2022**, 13 (2), 99–107.
3. S.D. Sanap, V. More. Design of efficient S-box for Advanced Encryption Standard. *J. Integr. Sci. Technol.* **2022**, 10 (1), 39–43.
4. N. Debabhuti, S. Mukherjee, P. Sharma, et al. Development of a portable gas sensing platform with QCM sensors for volatiles of agro products. *J. Mater. Nanosci.* **2022**, 9 (2), 115–119.
5. H. Rajoriya, R. Sadiwala. Deep compressive sensing and reconstruction algorithm in wireless Internet of Things. *J. Integr. Sci. Technol.* **2023**, 11 (2), 487.
6. M.K. Pote, P. Mukherji, A. Sonawane. Design of 5G Microstrip patch array antenna for gain enhancement. *J. Integr. Sci. Technol.* **2022**, 10 (3), 198–203.
7. S.K. Apat, J. Mishra, K.S. Raju, N. Padhy. The robust and efficient Machine learning model for smart farming decisions and allied intelligent agriculture decisions. *J. Integr. Sci. Technol.* **2022**, 10 (2), 139–155.
8. C. Ünsalan, H.D. Gürhan, M.E. Yücel. Embedded System Design with ARM Cortex-M Microcontrollers. *Embed. Syst. Des. with ARM Cortex-M Microcontrollers* **2022**.
9. O. Vermesan, P. Friess, P. Guillemin, et al. Internet of Things Strategic Research Roadmap. In *Internet of Things - Global Technological and Societal Trends from Smart Environments and Spaces to Green Ict*; River Publishers, **2022**; pp 9–52.
10. M. Ashjaei, L. Lo Bello, M. Daneshtalab, et al. Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities. *J. Syst. Archit.* **2021**, 117, 102137.
11. A. V. Jha, B. Appasani, A.N. Ghazali, et al. Smart grid cyber-physical systems: communication technologies, standards and challenges. *Wirel. Networks* **2021**, 27 (4), 2595–2613.
12. K. Kolomvatsos, M. Tsiroukis, S. Hadjiefthymiades. An experiment description language for supporting mobile IoT applications. *Build. Futur. Internet through FIRE 2016 FIRE B. a Res. Exp. based Approach English* **2017**, 461–490.
13. P. J. Patil, R. V. Zalke, K. R. Tumasare, et al. IoT Protocol for Accident Spotting with Medical Facility. *J. Artif. Intell. Capsul. Networks* **2021**, 3 (2), 140–150.
14. M. Krishna, S.M.B. Chowdary, P. Nancy, V. Arulkumar. A Survey on Multimedia Analytics in Security Systems of Cyber Physical Systems and IoT. In *Proceedings - 2nd International Conference on Smart Electronics and Communication, ICOSEC 2021*; IEEE, **2021**; pp 1–7.
15. D. Jayakumar, D.U. Palani, D. Raghuraman, et al. Certain investigation on monitoring the load of short distance orienteering sports on campus based on embedded system acceleration sensor. *Eur. J. Mol. Clin. Med.* 7 (9).
16. C.K. Lo, C.H. Chen, R.Y. Zhong. A review of digital twin in product design and development. *Adv. Eng. Informatics* **2021**, 48, 101297.
17. Z. Yuan, J. Jin, L. Sun, K.W. Chin, G.M. Muntean. Ultra-Reliable IoT Communications with UAVs: A Swarm Use Case. *IEEE Commun. Mag.* **2018**, 56 (12), 90–96.
18. P. Marwedel. Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems; Springer Nature, **2010**; Vol. 16.
19. A. Triantafyllou, P. Sarigiannidis, T.D. Lagkas. Network protocols, schemes, and mechanisms for internet of things (IoT): Features, open challenges, and trends. *Wirel. Commun. Mob. Comput.* **2018**, 2018.
20. Köksal, B. Tekinerdogan. Architecture design approach for IoT-based farm management information systems. *Precis. Agric.* **2019**, 20 (5), 926–958.
21. K. Li, L. Cheng, Z. Lyu, N. Xiang. Reliability assessment based on time waveform characteristics with small sample: A practice inspired by few-shot learnings in metric space. *Appl. Soft Comput.* **2022**, 115.
22. K. Bhosle, B. Ahirwadkar. Deep learning Convolutional Neural Network (CNN) for Cotton, Mulberry and Sugarcane Classification using Hyperspectral Remote Sensing Data. *J. Integr. Sci. Technol.* **2021**, 9 (2), 70–74.
23. C.R. Kumar, A. Almasarani, M.A. Majid. 5G-Wireless sensor networks for smart grid- A ccelerating technology's progress and innovation in the Kingdom of Saudi Arabia. *Procedia Comput. Sci.* **2021**, 182, 46–55.
24. I. Ahmed, P.K. Yadav. An Automated System for Early Identification of Diseases in Plant Through Machine Learning. In *Lecture Notes in Networks and Systems*; Springer, Singapore, **2022**; Vol. 425, pp 803–814.
25. N. Moustafa. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, 72, 102994.
26. I. Ahmed, P.K. Yadav. Plant disease detection using machine learning approaches. *Expert Syst.* **2022**, e13136.