J. Electrical Systems 20-3 (2024): 7194-7202

[1]Rajesh Shankar Karvande

[2]Tatineni Madhavi

Design and Development of IoT-based secure wireless network for control and configuration of Remote Hardware-In-Loop Simulation Test-Setup

Journal of Electrical Systems

Abstract: Hardware-in-loop simulation (HILS) is a unique platform validating avionics software and subsystems. This process takes place in the HILS laboratory, where multiple avionics systems are interconnected, with additional avionics subsystems remotely integrated into the simulation. It is essential that these systems be controlled, monitored, and operated automatically through a single point of command, control, and configuration, managed remotely via a centralized computer. To achieve this, an IoT-based wireless network has been designed and implemented for the control and configuration of both local and remotely situated subsystems within the HILS environment, aligning with Industry 4.0 standards for control and automation. Notably, this design operates without internet connectivity, allowing for secure and reliable remote operation and data transmission of avionics systems like actuators, inertial navigation systems (INS), or propulsion systems, even in harsh environments with zero human intervention. Given the sensitivity and high value of avionics subsystems within HILS—many of which are flight-worthy—it is crucial to protect the network from unauthorized access for multifunctional. This novel control and configuration system leverages Bluetooth interfaces, an Arduino board, and the nRF24L01 transmitter and receiver module for long-distance wireless communication, integrated with a user application built on the QT framework on the desktop. This paper presents the development of the hardware architecture and software, along with suggestions for future enhancements.

Keywords: *Internet of the things; Hardware-In-Loop Simulation; Arduino-Uno board; Bluetooth interface; Wireless network; Remote Simulation;*

I. INTRODUCTION

The Internet of Things (IoT) and wireless networks are emerging fields crucial to the automation of industrial resources. These technologies are underpinned by optimized hardware architectures that require minimal energy to operate continuously, and around the clock. Essential IoT hardware, such as Arduino Uno, Raspberry Pi, and Node MCU, supports wireless command and control, with Arduino and Raspberry Pi being particularly popular in this domain. This research focuses on these two types of IoT hardware architectures, which were chosen based on the requirements of the command-and-control center. Custom software has been developed and deployed on-site, integrating with the hardware to meet the specific needs of the user environment. Hardware-In-Loop Simulation (HILS) is a unique facility employed to test and validate avionics subsystems, including mission-critical software [1][19]. HILS facilities consist of avionics hardware such as onboard computers (OBC), Inertial Navigation Systems (INS), actuator setups, and launch computers. These systems require stable power supplies, generally 230V AC for computers and a 28V DC programmable power supply for critical avionics subsystems. Given that HILS simulations are frequently run multiple times daily, there is a need for sequential switching of power units, often in remote or harsh environments where minimal human intervention is ideal. In HILS operations, the sequence begins with the activation of a 230V AC power supply for the lab's lighting, DC supply units, and simulation computers. This is followed by switching on the 28V DC for critical units, where current monitoring ensures proper subsystem functionality. The boot process for the simulation computer then begins, allowing real-time evaluation of the avionics software [2][18]. A specific configuration file, containing run type, aero-data, thrust data, and other parameters, is loaded after booting of simulation system[3]. The simulation commences once the six-degree-of-freedom (6DOF) module receives a lift-off command from the OBC post-initial health checks conducted by the launch computer. The closed-loop simulation is subsequently executed with control and guidance algorithms inside

[1] *Corresponding author: Scientist-F, RCI, Defence Research and Development Organization, Hyderabad, Telangana, India. and, Research Scholar, Department of Electrical, Electronics & Communication, GITAM School of Technology, GITAM University, Hyderabad Campus, Hyderabad, Telangana, India. Email: rajeshkarvande@rediffmail.com.

[2] Professor, Electrical, Electronics & Communication, GITAM School of Technology, GITAM University, Hyderabad Campus, Hyderabad, Telangana, India. Email: tatinen@gitam.edu.

Copyright © JES 2024 on-line: journal.esrgroups.org

the OBC, interacting with the simulation computer's plant model. The process in the HILS setup, as illustrated in Fig. 1, currently in the existing scenario, requires hands-on monitoring by numerous personnel due to multiple projects running concurrently. In line with advancements in industrial automation and control, the integration of IoT technology in defense operations can greatly enhance efficiency. A significant challenge, however, lies in the lack of internet access, making it essential to employ independent, secure wireless networks. The proposed automation solution is developed in a staged manner, prioritizing cost-efficiency and low energy consumption. Additionally, the system's security is paramount, ensuring that no unauthorized entities can access the network [4]. This paper provides an in-depth examination of HILS test-bed automation, covering command, control, and configuration processes. It is structured into sections on related work, problem definition, development methodology (including both hardware and software), testing, and analysis, and concludes with suggestions for future research directions.

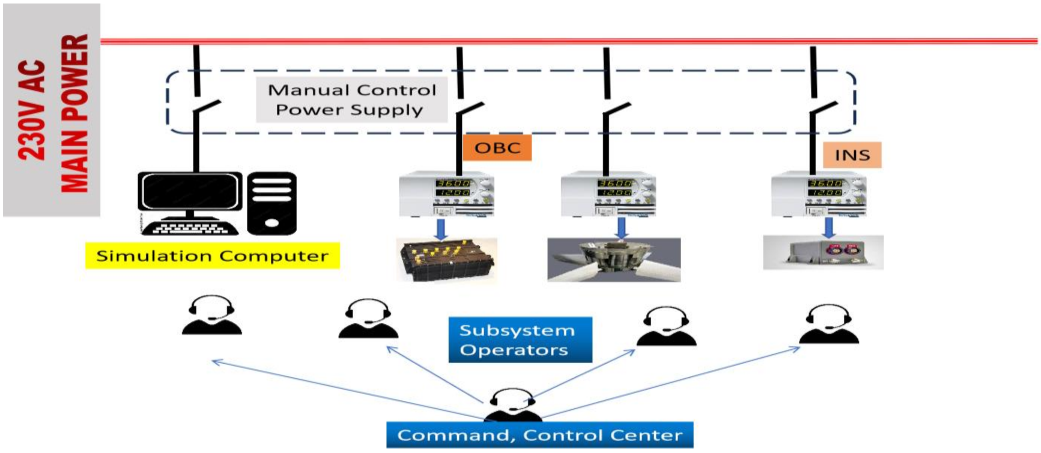

Fig. 1. Typical HILS Test-bed with Manual Control of Devices.

II. RELATED WORK

An extensive literature survey has been conducted to review existing work in home automation, control, and AI. Numerous researchers have contributed to these fields. In the paper titled "An Open Platform for Hardware-in-the-Loop Virtual Simulation via the Internet of Things," by Ke Dong, Tianle Yu, et al., the authors explore an FPGA-based IoT hardware architecture that leverages cloud computing for transmitting HILS data. Their network design, however, lacks internet accessibility, necessitating a wireless local network. Another relevant study by Severin Nowak, Nima Tehrani, et al., titled "Cloud-based DRERMS Test Platform Using Real-Time Power Simulation," aligns closely with our research on power control but is limited by its dependency on the internet. Similarly, the paper "Closing the Loop: A High-Performance Connectivity Solution for Realizing Wireless Closed-Loop Control in Industrial IoT Applications" focuses on IoT applications for industrial control. The paper "Wireless Control of Unmanned Vehicle for Surveillance Using Virtual Reality Technology" discusses the video processing capabilities of UAVs over wireless networks. A considerable body of literature supports the use of IoT-based hardware in various applications. The unique contribution of this paper is in enabling the remote operation of a HILS test-bed through an IoT-like wireless network without internet accessibility, incorporating enhanced security features.

III. PROBLEM DEFINITION

Multiple HILS test setups operate in parallel for avionics validation and testing, with numerous HILS runs conducted to test software. Each HILS simulation follows a systematic operational process: first, the setup's power supply is switched on, and the simulation computer loads the configuration data file. Next, the avionics subsystem initiates the simulation, and upon completion, data is sent for analysis. This process currently requires manual monitoring and control by HILS personnel, resulting in sequential rather than parallel operation of setups. Efficiency could be significantly improved through intelligent planning, and automating routine activities without human intervention. This can be achieved by implementing modern network monitoring and control technologies [4]. With the industry moving towards greater automation, we propose automating the HILS setup using an advanced IoT-based wireless network. This approach involves designing and developing custom hardware and software to meet user and HILS process requirements. Given that avionics units connected to HILS are flight-worthy, network security is paramount to prevent unauthorized access that could compromise operations or pose risks to the avionics units under test. To address this, we plan to implement an RFID-based secure network access system with dual-layer security. The first layer is RFID-based, and the second is a 4x4 keypad for PIN entry, enhancing security and safeguarding critical avionics hardware from unauthorized access. There are three steps to developing a fully automated HILS Network. In the first step, the power supply for the subsystems remotely situated has been controlled. In the second step, the data is to be exchanged between the simulation computer and the centralized located computer. For this another IoT-based device Rasberry-Pi could be used with the independent wireless network [7]. The third and final step is to send the commands in real-time and acquire the data in real-time on the wireless network. This is planned for actuator or INS subsystems located at distance from HILS Test setup. This research has been carried out to achieve the first objective. The following sections detail the hardware design and software developed for this application, with a focus on cost-effectiveness and ease of deployment across multiple HILS setups and locations [5]. Various microcontrollers and microprocessors are utilized in the IoT network, with Arduino being one of the simplest architectures. The proposed hardware architecture incorporates an Arduino Uno board paired with a Bluetooth interface (HC-06). On the user side, a transmitter communicates with a receiver on the HILS test bed, which is connected to a relay interface for controlling different power supply units [6]. Since the power supply operates at 230 V AC, an AC relay has been selected to manage the 230 V power supply to the simulation computer, launch computer, and DC power supply linked to critical flight hardware. The following section calculates the total load for the 230 V line and 28 V DC power supply.

When it is 230 V AC to lower DC voltage conversion, the transformer ratio can be calculated as,

$$\text{Turns Ratio} = (Primary\ Voltage)/(Secondary\ Voltage) \tag{1}$$

For 20 V AC conversion from 230 V AC,

115:1 transformer ratio is required.

$$V_{DC} = V_{AC} \times 1.414 \tag{2}$$

so for 20 AC RMS AC secondary voltage, 28 V DC is calculated.

Avionics Systems Total Current Calculation (28 V DC):

OBC = 0.9 A, INS = 1.6 A, MIU= 0.5A, EMA = 2.5 A and other subsystems = 0.5 A.

So, the total current required by 28 V P.S.

Total Current Required by 28 V Power Supply = 0.9 + 1.6 + 0.5 + 2.5 + 0.5 = **6 A**

The Relay should withstand the above DC 28V, 6 A current and should work smoothly with the avionics load.

Output Power Calculation:

The output power P_{out} is given by:

$$P_{out} = V_{out} \times I_{out} \tag{3}$$

Where, $V_{out} = 28\ V$ and $I_{out} = 6\ A$

$\boldsymbol{P_{out}} = 28 \times 6 = \boldsymbol{168\ W}$.

Input Power Calculation:

To calculate the required input power, we need to account for the efficiency η of the conversion.

Assuming a typical efficiency of around 85% for a well-designed AC-DC converter:

$$P_{in} = Pout/\eta$$

(4)

Where, $\eta = 0.85$

$$P_{in} = 168/0.85 \approx 197.65\ W$$

Input Current Calculation:

The input current

$$V_{DC} \approx V_{AC}$$

$$I_{in} = P_{in} / V_{in}$$

(5)

Where, V_{in} = 230 V,

$$I_{in} = 197.65/230 = \mathbf{0.86\ A.}$$

Output Power = 168 W, Input Power = 197.65 W and Input current is 0.86 A at 230 V AC.

Accordingly, Relay is selected and the hardware is controlled. It is shown in the fig. 3 with Aurdino board. The user control is executed via a general laptop equipped with a built-in Bluetooth interface. This interface facilitates command exchanges with the Arduino Uno board, located in a corner of the room. Bluetooth technology operates within the 2.4 GHz ISM spectrum band (2400 to 2483.5 MHz), providing an optimal balance between range and throughput. The 2.4 GHz band is globally accessible, establishing it as a standard for low-power wireless connectivity. Bluetooth supports transmit power ranging from -20 dBm to +20 dBm. A basic diagram of the full setup is illustrated in Fig. 2.

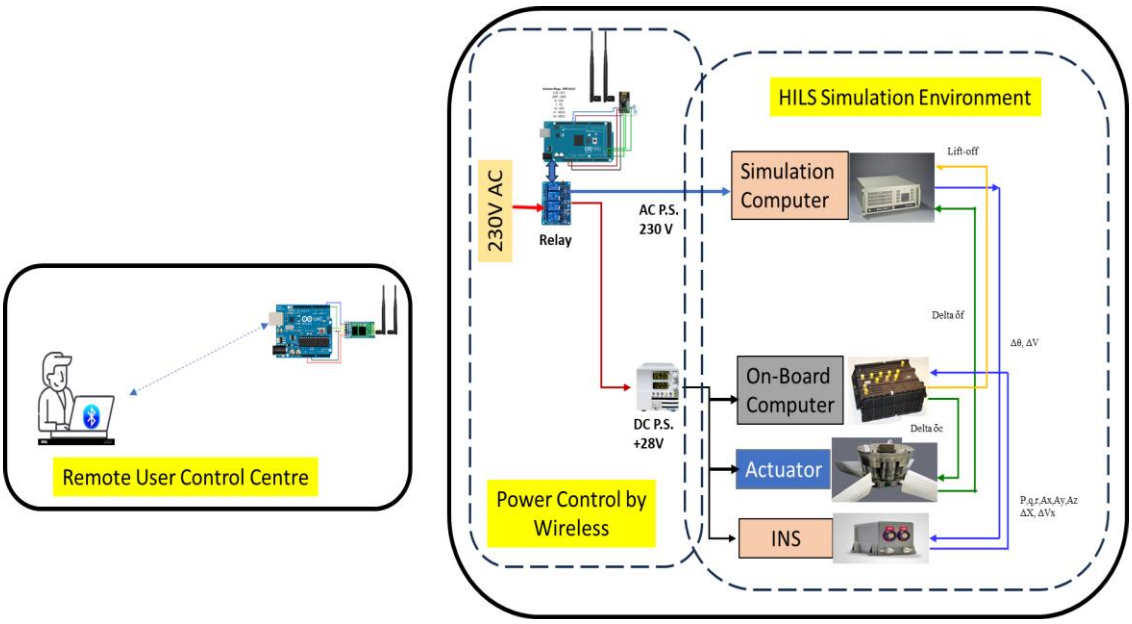

Fig. 2. User Control Room Bluetooth Architecture

The Bluetooth interface integrated into the laptop or desktop can be programmed using the Graphical User Interface software (GUI) that is QT Bluetooth API, enabling data transmission and reception via Bluetooth with the Arduino board. The user interface is developed with QT, utilizing the Bluetooth device library for back-end support. Commands such as “SWITCH-ON”, “SWITCH-OFF” and “SEND-CONFIG” are programmed to function through this Bluetooth interface. On the Arduino Uno side, the HC-06 wireless Bluetooth transceiver receives the data, which is then transmitted across the wireless network.

The Arduino Uno board is based on the ATmega328P microprocessor operating at 16 MHz, with 2 KB SRAM, 32 KB FLASH, and 1 KB EEPROM for various applications. The board supports communication interfaces such as UART, I2C, and SPI, and features 14 digital I/O pins, 6 analog input pins, and 6 PWM pins for interfacing with external modules. It operates on a 5V power supply with a current draw of 20 mA, utilizing a 9V battery connector. The relay-controlled power supply interfaces with the actuators, allowing precise control at the HILS setup [9][10].

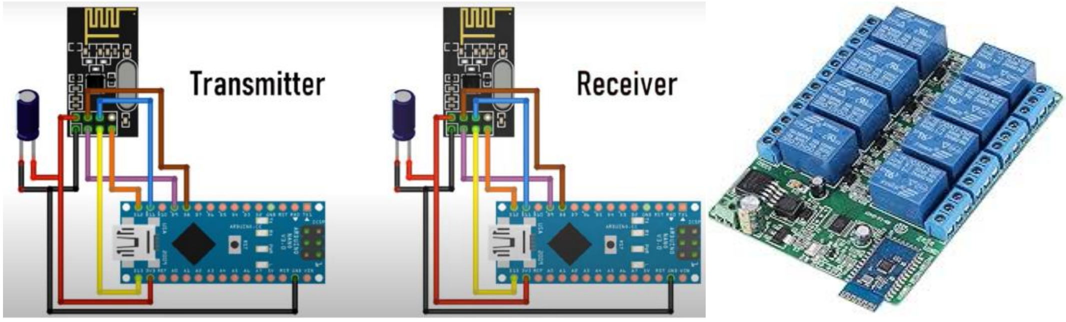

Fig. 3. Arduino Board Interface with Transceiver Module and relay module.

The transceiver module employed is the nRF24L01, which operates in the 2.4 GHz band and supports baud rates from 250 kbps up to 2 Mbps. In open spaces and at lower baud rates, the module can achieve a range of up to 100 meters. The maximum current draw during transmission is 12.3 mA. This module facilitates communication between the user's desktop/laptop and the HILS laboratory. If the line of sight is obstructed by walls or doors, the operational range can be significantly reduced due to interference. The circuit is installed in the corner of the user's remote operation room. An experimental setup is established in the HILS laboratory, featuring antenna and receiver circuitry for monitoring voltage and current parameters [8]. In the third stage of development as explained in the above section with the wireless module, it is planned to send deflection commands are sent to the actuators, and feedback commands are monitored via the wireless network to the user computer. These feedback signals are then transmitted to the 6-DOF plant model for rate and acceleration computations [15]. Simulation data is sent to the On-Board Computer (OBC) for control and guidance algorithm execution.

IV. SECURITY FEATURE

Critical avionics hardware that is flight-worthy is integrated into the HILS test setup. Vulnerabilities in the network can potentially damage the avionics systems. This risk can be mitigated by implementing authorization mechanisms such as keys, PINs, or RFID solutions, thereby protecting against unauthorized network usage. For this application, the RFID module RC522 has been utilized, comprising two main components: the reader and the tag. Each tag is assigned to an authorized individual. At the beginning of each day, the authorized person activates the board for use within the HILS setup [16]. An alternative option for authorized access is a PIN-based system using a 4x4 keyboard. This interface allows users to generate a password for network access. Upon key presses, the selected row and column connect to the Arduino board [17]. Either interface can be used for authorization. Once validated, an "AUTHORIZED" message is displayed on an LCD screen, as shown in Fig. 4. Both the RFID and keyboard systems have been tested, but the final configuration favors the RFID option due to its portability and the absence of complex passwords to memorize. This ensures network security and prohibits unauthorized access.

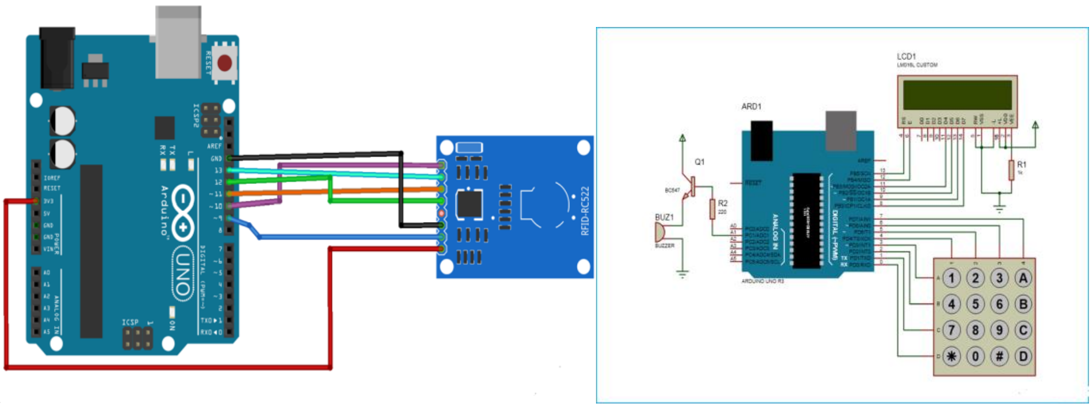

Fig. 4. RFID Interface and Code for Security Access to Arduino

V. Testing and analysis

The integrated circuit testing was conducted over variable distances, and performance measurements were taken. Due to walls between the user desktop and the HILS setup, the transmitter operates effectively within a range of 40 to 50 meters. The latency in operations remains within acceptable limits, factoring in both data acquisition and wireless transmission delays [11]. This is acceptable since the circuit is designed primarily for initial phases, such as switching the power supply on and off, rather than participating in real-time operations.

The RFID interface has been tested with tags, successfully authorizing secure access to the network. Unauthorized access is declined by the transmitter board, safeguarding critical flight hardware from spurious signals. This circuit is effective for distances less than 100 meters; for greater distances, a transmitter with enhanced capabilities should be selected [12]. Additionally, the circuit is energy-efficient and offers a cost-effective method for automating the HILS test setup.

The software was developed in the Arduino IDE 2 (Integrated Development Environment), which offers improved features over previous versions, including auto-completion, a built-in debugger, and synchronization of sketches with the Arduino Cloud [13]. The software development is divided into three parts:

1. Bluetooth Software on the User Laptop.
2. Arduino Board Software on the User Side.
3. Arduino Board Software at the HILS Setup.

The Bluetooth software initiates commands to control the HILS setup using the QT-based Bluetooth library. The graphical user interface (GUI) is designed in QT to enable programming and data transmission to the Bluetooth-paired device. The flowchart for the transmitter as well as the receiver relay side is depicted in Fig. 5. It is a continuous loop waiting for input from the user and execution of the command.

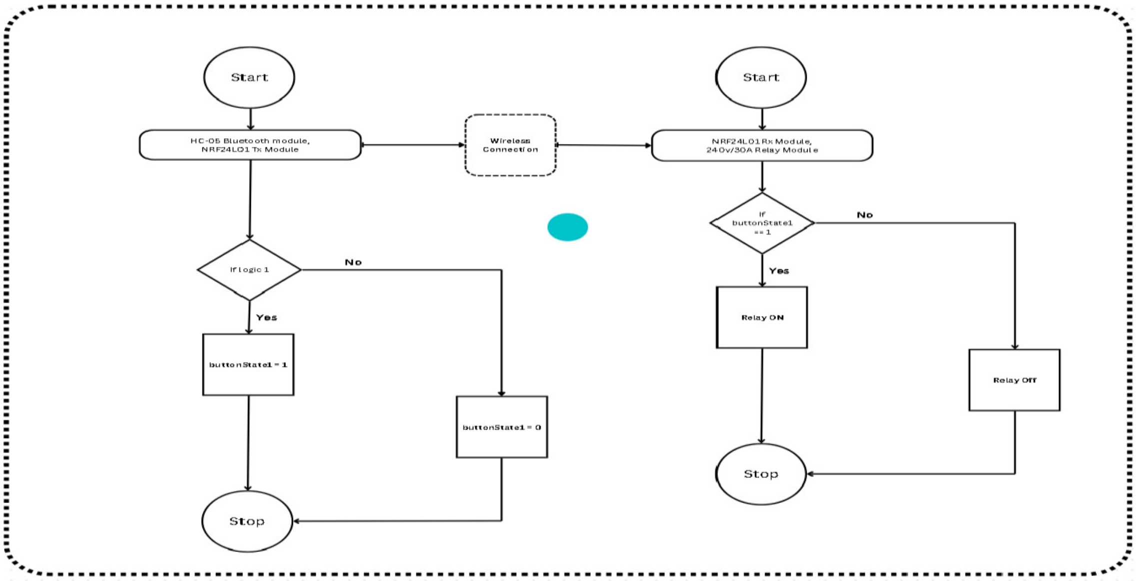

Fig. 5. Software Flowchart for Relay Operation.

The Arduino board at the user site is responsible for receiving data via Bluetooth and sending "ON" or "OFF" commands to the transmitter device. The third component involves the Arduino board software at the HILS setup, which is crucial as it operates the relay. The software code integrated into the Arduino board is illustrated in Fig. 6 for both the user and HILS setups [14].

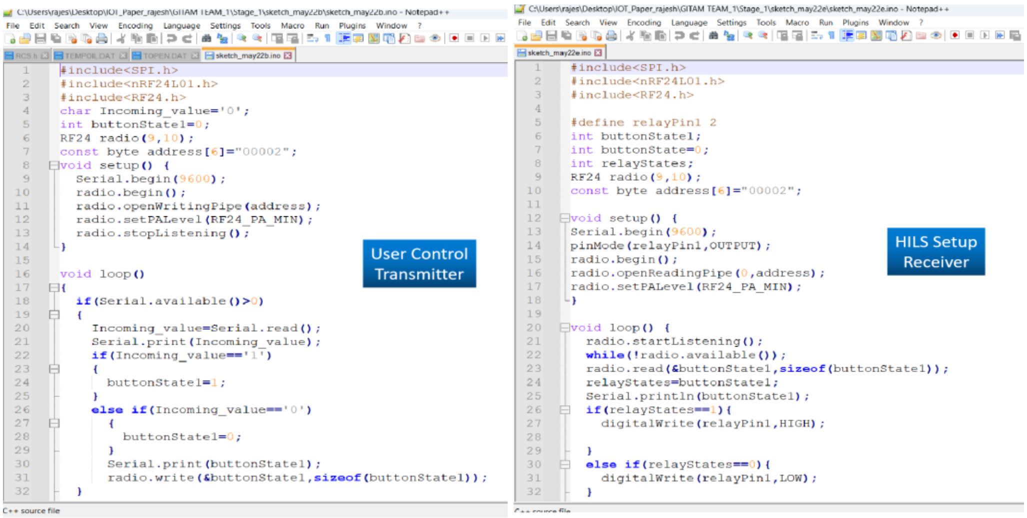

Fig. 6. Software code for Transmitter and Receiver.

The relay operates based on inputs received from the `radio. Start listening ()` and `radio. Read ()` library functions. The relay status is assigned to the button state, and the relay is controlled using the `digitalWrite()` function, which activates the appropriate digital I/O (DIO) based on user selection. DIO-1 controls the power supply, DIO-2 manages the actuator power supply, and DIO-3 is connected to the embedded system and inertial navigation system (INS) dedicated electronics, which draw less current. Thus, all electronics are controlled from the user's desktop, allowing HILS runs to be performed in a stepwise manner. The pictorial view of the experimental setup is shown in Fig. 7. All the components have been assembled in the HILS Laboratory for an integrated solution. Number of times the circuit is operated, tested, and then deployed for HILS setup.

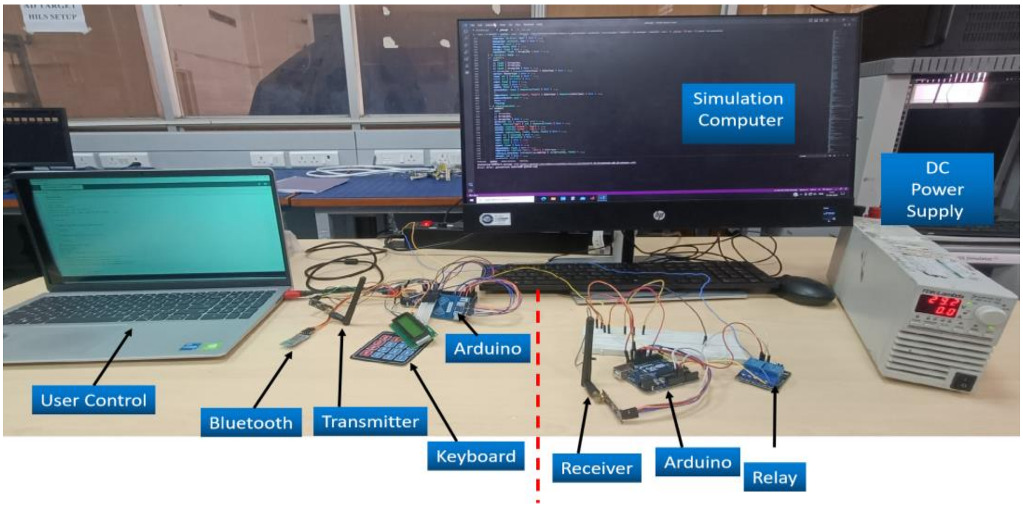

Fig. 7. Experimental Setup at HILS Lab

This scheme is useful for the automated testing process that enables the power supply switching. Mostly the embedded system and the simulation computer are also tuned at the kernel level (RT-Linux) to boot and run the simulation application that resides at a particular directory location. In this experiment, sequential operations have been carried out, that is relay for the simulation computer is switched on first, and after 120 seconds that is given to boot the simulation computer and get ready with the application, the relay of avionics subsystems operates. The INS switched ON, then actuator, and finally, the OBC switched ON. OBC sends the simulation start command. The simulation takes place in real time and according to the trajectory profile the HILS time is allocated. After about 300 seconds the simulation stops. The user sends to Switch OFF the power supply of OBC through Laptop-Bluetooh-Aurdino-RFTx-RFRx-Aurdino-Realy-Power Supply. After all, the avionics have been switched OFF the simulation computer reboots to Power off the process start.

VI. CONCLUSION AND SUGGESTIONS

This paper presents the design and development of automation for the HILS laboratory. The system allows for wireless control, command, and configuration from a user computer located up to 50-60 meters away, utilizing Bluetooth technology and an Arduino board with a transceiver module to manage power sources for the avionics systems. The software development was executed in the Arduino IDE environment, with a QT GUI deployed on the user desktop for command transmission via Bluetooth libraries. This architecture has been tested and successfully deployed in the HILS laboratory, demonstrating a cost-effective design with low power consumption. Network security is ensured through the implementation of RFID and PIN-based access mechanisms to prevent unauthorized access, safeguarding the integrated critical avionics systems. Thus, a wireless network incorporating RFID and PIN-based algorithms has been developed. The research effectively utilizes the popular Arduino board, maximizing its resources. Future development is proposed to expand this setup into a single control center that can manage multiple HILS setups. This initial step toward automating the HILS configuration addresses the challenge of managing multiple setups simultaneously. The implementation of such wireless architectures has successfully automated the HILS setup. Moreover, due to limitations in data transmission with the Arduino board, future work will consider using more advanced platforms, such as Raspberry Pi or other microprocessor boards, alongside the development of a Wi-Fi network (without internet) to facilitate data transmission over the wireless network. The road map for future development is to develop the data transmission using a wireless network that is before and after simulation. In the final stage, real time wireless communication has to be introduced to send real-time commands to subsystems and acquire real time data for distance communication.

REFERENCES

[1] A. Singh, N. Singh, T. Agrawal and Y. K. Chauhan, "Wireless Control of Unmanned Vehicle for Surveillance using Virtual Reality Technology," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida India, 2018, pp. 725-731, doi: 10.1109/GUCON.2018.8674903

[2] J. Kölsch, C. Heinz, S. Schumb and C. Grimm, "Hardware-in-the-loop simulation for Internet of Things scenarios," 2018 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), Porto, Portugal, 2018, pp. 1-6, doi: 10.1109/MSCPES.2018.8405399

[3] M.Anniken Semb Kvalsund, Dietmar Winkler, "Development of an Arduino-based, open-control interface for hardware in the applications", J. HardwareX, 2023, Vol. 16, e00488, ISSN 2468-0672, https://doi.org/10.1016/j.ohx.2023.e00488.

[4] Erfan Soltanmohammadi et. al, "A Survey of Traffic Issue in Machine to Machine communication over LTE", IEEE Internet of Things, Vol. 3, No: 16, 2016.

[5] Ke Dong et. al, "An Open platform for Hardware-In-Loop virtual Simulation via Internet of Things", 2nd Information Communication Technologies Conference, 2021.

[6] Rajesh Karvande and B. Ramesh Kumar, "Development of Hardware-In-Loop Simulation Test-bed for testing of Navigation System-INS", Proceeding of International conference on Machine Intelligence Research and Advancement, 2013, pp. 536-539, DOI: 10.1109/ICMIRA.2013.112.

[7] Ahmed Bilal Zia, Kshmata Chauhan, "A Research paper on Internet of Things based upon smart Homes with Security Risk assessment using OCTAVE allegro", International Journal of Research and Technology, ISSN: 2278-0181, Vol. 9 Issue 06, June-2020.

[8] Rajesh Karvande, "cPCI based Hardware-In-Loop Simulation System development under Real Time Operating System", of International conference on Advances in computing, Communications and informatics, 2014, pp. 736-739, DOI: 10.1109/ICACCI.2014.9698563.

[9] Adnan Aijaz, Aleksander Staneov, "Closing the Loop: A high performance connectivity solutions for realizing wireless close loop control in industrial IoT applications", IEEE Internet of Things Journal, Vol. 8, No:15, Aug. 2021, DOI: 10.1109/JIOT.2021.3073505

[10] Yao Zhao, Wei Zhang, De-bin Sang, Sang Yu, "On joint hardware-in-the-loop simulation of aircraft control system and propulsion system", 2017 36th Chinese Control Conference (CCC), pp.10387-10391, 2017.

[11] Emre Yigitoglu et. al, "Distributed Orchestration in Large-Scale IoT Systems", IEEE international Congress on Internet of Things, 2021, DOI: 10.1109/IEEE.ICIOT.2017.16.

[12] Rajesh S. Karvande, L. Sobhan Kumar, Maharshi Raman, "Multi-node MIL-STD-1553 Avionics Bus-Monitor Development Under Linux Platform", Advances in Decision Sciences, Image Processing, Security and Computer Vision, vol.3, pp.245, 2020.

[13] Hanghao Lv et al.,"Impact of Wireless on Robot Control: The network Hardware-In-Loop-Simulation FrameWork and Real Life Comparision", IEEE Transaction on Industrial Informatics, DOI: 10.1109/TII.2022.3227639

[14] Mastu Thorton, John Branigan, " Internet-of-Things Hardware-in-the-Loop Simulation Architecture for providing Frequency Regulation with Demand Response", IEEE Transaction on Industrial Informatics, Vol. 14, Issue:11,2017, DOI: 10.1109/TII.2017.2782885

[15] Abhisekh Singh, Neha Singh et. al., "Wireless Control of Unmanned Vehicle for Surveillance using Virtual Reality Technology", 2018 International Conference on Computing, Power and Communication Technologies (GUCON) DOI: 10.1109/GUCON.2018.8674903.

[16] Karvande, R. & Madhavi, T. High Performance Optimized and Unified Bus Monitor for Distributed Avionics Architecture Validation Under Hardware-In-Loop Simulation Framework. In International Conference on current Trends in Advanced Computing 2024, pp. 80-84. doi: 10.1109/ICCTAC61556.2024.10581142.

[17] Rajesh K. & Tatineni, M. Optimized digital Simulation Methodology for System Validation using Real-Time Hardware-In-Loop Simulation Platform. IEEE AERO EL SYS MAG, 2024, 39(8), pp. 18-27. doi: 10.1109/MAES.2024.3412034.

[18] Zhou Y.; K. -V. Ling, F. Ding and Y. Hu, "Online Network-Based Identification and its Application in Satellite Attitude Control Systems," in IEEE Trans. Aerosp. Electron. Syst., 2023. 59(3), pp. 2530-2543, doi: 10.1109/TAES.2022.3215946

[19] Qian, X.; Xiaodong, L.; Wenjing, Z.; Ya, B.; Le, M. & Penghzen, H. A Servo Simulation Test Calibration and Compensation Method for Hardware-In-Loop Simulation of Aerocraft Control System. *In* Proceeding of 34th Chinese Control and Decision Conference (CCDC) 2022. doi: 10.1109/CCDC55256.2022.10033579