

Performance Evaluation of UART-Based Bluetooth Home Automation System using LPC2138 ARM7 Microcontroller

Vrushabh M. Ukhalkar, Prof. Sunanda K. Kapde
Department of Electronics and Telecommunication Engineering
Deogiri Institute of Engineering and Management Studies,
Chhatrapati Sambhajanagar, Maharashtra, India.

Abstract - This paper evaluates the response performance of a UART-based Bluetooth home automation prototype developed using the LPC2138 ARM7 microcontroller and HC-05 Bluetooth module. The system receives ASCII commands from an Android Bluetooth terminal application and transfers them to the LPC2138 through UART1 at 9600 bps with 8 data bits, no parity, and 1 stop bit. The received commands are decoded by custom Embedded C firmware to control onboard LED indicators representing home appliances, while a 16×2 LCD provides real-time status feedback. Unlike basic Bluetooth switching demonstrations, this work focuses on practical performance parameters such as command response delay, command reliability, distance behavior, protocol suitability, and command-filtering robustness. The firmware includes carriage-return and newline filtering to avoid invalid-command execution caused by extra characters transmitted by mobile terminal applications. Experimental testing was carried out for six commands representing bulb ON/OFF, fan ON/OFF, all outputs ON, and all outputs OFF. The observed response delay varied from 52 ms to 60 ms, with an average response delay of approximately 56 ms. Command reliability was observed between 95% and 100% over repeated trials. Distance testing showed stable operation at 1 m, 3 m, and 5 m, minor delay at 8 m, and variable response near 10 m depending on environmental conditions. The results indicate that UART-based HC-05 communication is suitable for short-range, low-cost, prototype-level home automation using LPC2138.

Index Terms— Bluetooth communication, command reliability, HC-05, home automation, LPC2138, response latency, UART.

I. INTRODUCTION

Home automation systems are widely used to reduce manual switching effort and improve user convenience in residential, laboratory, and academic prototype environments. In a basic embedded automation system, a microcontroller receives user

commands, processes them through firmware, and controls output devices such as lamps, fans, indicators, or relay-based loads. Wireless control further improves flexibility because the user can operate the system without direct contact with physical switches.

Bluetooth is a suitable communication option for short-range prototype-level automation because it is inexpensive, easy to configure, and supported by most Android smartphones. The HC-05 Bluetooth module is commonly used in embedded applications because it converts Bluetooth Serial Port Profile data into UART serial data. This allows direct communication between a smartphone application and a microcontroller through simple serial command transfer.

The LPC2138 ARM7TDMI-S microcontroller provides GPIO ports, UART channels, timers, ADC, DAC, SPI, I2C, PWM, and on-chip flash memory. These features make it suitable for embedded control and communication-based applications. In the developed prototype, UART1 is used for HC-05 Bluetooth communication, while UART0 is reserved for programming and debugging through the CP2102 USB-UART interface. The board operates with a 12 MHz crystal oscillator and the firmware is developed using Embedded C.

Many Bluetooth home automation projects demonstrate basic ON/OFF switching, but practical performance factors such as command response delay, repeated-command reliability, distance behavior, command filtering, and protocol suitability are also important for evaluating such systems. Mobile Bluetooth terminal applications may transmit extra carriage-return and newline characters along with the actual command. If these extra characters are not filtered, the firmware may interpret them as invalid commands.

This paper focuses on the performance evaluation of a UART-based Bluetooth home automation prototype using LPC2138 and HC-05. The main evaluation parameters include response latency, command reliability, distance behavior, UART suitability, and command-decoder robustness. The prototype uses custom Embedded C drivers for UART, GPIO, switch handling, command decoding, output control, and LCD feedback.

II. RELATED WORK

Home automation has been widely explored using microcontrollers, wireless modules, smartphone applications, and IoT-based platforms. In general, such systems use a controller to receive input commands and operate output devices such as lights, fans, indicators, or relay-connected loads. Recent smart-home studies show that embedded devices are important for low-cost automation, local control, and user convenience in residential applications [7]. Bluetooth-based automation is commonly implemented using Android phones and HC-05 Bluetooth modules because the HC-05 provides simple serial communication support through UART [4]. Arduino-based Bluetooth automation systems have been reported in earlier works, where the main focus is appliance switching through mobile commands [8]. Voice-controlled and Android-controlled systems also show that Bluetooth Serial Port Profile is useful for short-range command transmission between a smartphone and embedded hardware [9].

However, many existing works mainly demonstrate successful ON/OFF control and do not discuss practical response behavior in detail. Parameters such as response delay, command reliability, distance variation, invalid command handling, and terminal-generated carriage-return/newline characters are often not analyzed. These factors are important because a wireless switching system should not only receive commands but should also respond consistently under repeated operation.

ARM7-based implementation using LPC2138 gives better exposure to register-level embedded programming compared with high-level development boards. The LPC2138 provides UART, GPIO, timers, ADC, DAC, SPI, I2C, PWM, and on-chip flash memory, making it suitable for embedded communication and control applications [1], [2]. The ARM7TDMI-S core also supports efficient embedded control execution for such microcontroller-based systems [3].

The 16×2 HD44780-compatible LCD is commonly used as a local feedback display in embedded systems because it can show system status, output condition, and debugging information [5]. In the present work, LCD feedback is used along with Bluetooth command execution to observe mode status, load status, and invalid command conditions. The HC-05 module directly supports UART communication, whereas I2C is more suitable for short-distance chip-to-chip peripherals such as RTC, EEPROM, sensors, and I/O expanders [4], [6]. Therefore, UART is more appropriate for Bluetooth command transfer in the present system.

In this work, the focus is shifted from only implementing Bluetooth-based switching to evaluating the response performance of a UART-based LPC2138 home automation prototype. The study considers response latency, repeated command reliability, distance behavior, CR/LF filtering, and UART protocol suitability.

III. PROBLEM STATEMENT AND OBJECTIVES

Many low-cost Bluetooth-based home automation systems mainly demonstrate wireless ON/OFF switching of appliances. However, for practical evaluation, it is also important to study how quickly the system responds to a command, how consistently the command is executed during repeated trials, and how the system behaves at different operating distances. If these parameters are not analyzed, the system remains only a functional demonstration and does not clearly show its performance capability.

Another practical issue is command handling in Bluetooth terminal applications. Some mobile terminal applications transmit additional carriage-return and newline characters along with the actual command. If the firmware directly processes every received character, the system may display invalid command messages or may behave incorrectly. Therefore, reliable command filtering is required in the UART command decoder.

The objective of this work is to evaluate the response performance of a UART-based Bluetooth home automation prototype using LPC2138 and HC-05. The specific objectives are:

1. To interface the HC-05 Bluetooth module with LPC2138 using UART1 communication.
2. To implement command-based load control using custom Embedded C firmware.

3. To evaluate command response delay for Bluetooth commands A, B, C, D, E, and F.
4. To analyze command reliability over repeated trials.
5. To observe Bluetooth communication behavior at different distances.
6. To justify UART as a suitable protocol for HC-05-based command transfer.
7. To handle carriage-return and newline characters for robust command decoding.

IV. EXPERIMENTAL SYSTEM ARCHITECTURE

The experimental system is developed using the LPC2138 ARM7 microcontroller, HC-05 Bluetooth module, 16×2 LCD display, onboard LEDs, and manual switches. The LPC2138 acts as the main control unit and performs UART reception, command decoding, GPIO output control, switch monitoring, and LCD status updating. The HC-05 Bluetooth module provides wireless serial communication between the Android smartphone and the LPC2138 controller [4].

The Android Bluetooth terminal application is used to transmit ASCII commands. Commands A, B, C, D, E, and F are assigned to different load-control operations. After receiving a command through Bluetooth, the HC-05 module converts the wireless data into UART serial data. This data is received by UART1 of LPC2138. The firmware then filters unwanted carriage-return and newline characters, validates the command, updates the corresponding GPIO output, and refreshes the LCD status message. The basic command path of the experimental system is:

Smartphone → HC-05 Bluetooth Module → UART1 → LPC2138 → Command Decoder → GPIO Output Manager → LEDs and LCD Feedback

The onboard LEDs are used as prototype load indicators. LED1 represents bulb/load status, LED2 represents fan/load status, and LED3 is used as a mode or status indicator. The 16×2 LCD works as a feedback display and shows system readiness, command status, output state, and invalid command messages. The LCD is connected in 4-bit mode to reduce GPIO usage while maintaining sufficient display functionality [5].

The system architecture is organized into three layers. The hardware layer includes LPC2138, HC-05, LCD, LEDs, switches, and the power/reset arrangement. The driver layer includes UART, GPIO, LCD, and switch-handling routines. The application layer includes command filtering,

command decoding, mode selection, output control, and response observation. This layered arrangement makes the firmware easier to test and debug because UART reception, LCD display, switch reading, and output control can be verified separately.

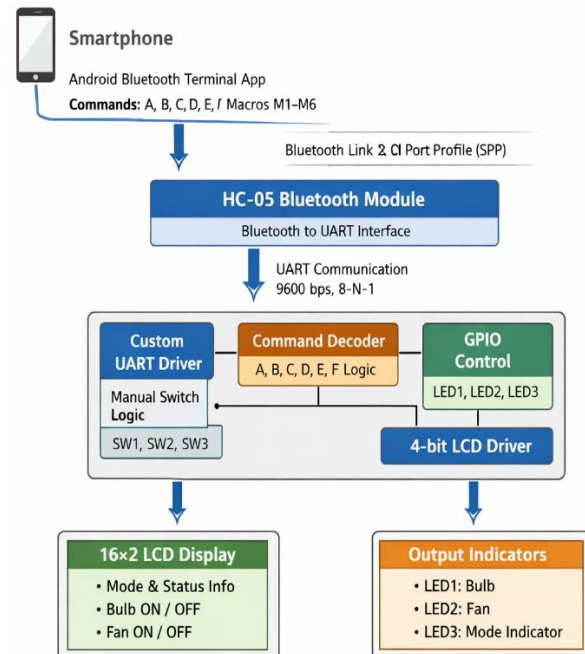


Fig. 1. Main system architecture of UART-based Bluetooth home automation using LPC2138 and HC-05.

V. HARDWARE AND SOFTWARE IMPLEMENTATION

The experimental prototype is implemented using an LPC2138 RhydoLABZ development board, HC-05 Bluetooth module, 16×2 LCD display, onboard LEDs, and onboard switches. The LPC2138 ARM7TDMI-S microcontroller is used as the main processing unit for UART communication, command decoding, GPIO control, switch monitoring, and LCD feedback.

The HC-05 Bluetooth module is connected to UART1 of LPC2138. The transmitter pin of HC-05 is connected to P0.9/RXD1 of LPC2138, and the receiver pin of HC-05 is connected to P0.8/TXD1 of LPC2138. UART1 is configured at 9600 bps with 8 data bits, no parity, and 1 stop bit. UART0 is used separately for programming and debugging through the CP2102 USB-UART interface.

The onboard LEDs are used as output indicators. LED1 connected to P0.17 represents bulb or load-1 status, LED2 connected to P0.18 represents fan or load-2 status, and LED3 connected to P0.19 is used as a mode/status indicator. Manual switches are

connected to P0.14, P0.15, and P0.16. SW1 and SW2 are used for manual load control, while SW3 is used for switching between Bluetooth mode and manual mode.

The 16×2 LCD display is interfaced in 4-bit mode to reduce the number of GPIO pins required for display operation. The LCD control pins RS, RW, and EN are connected to P1.17, P1.18, and P1.19 respectively, while data pins D4, D5, D6, and D7 are connected to P1.20, P1.21, P1.22, and P1.23 respectively. The LCD displays system status, selected mode, load condition, and invalid command messages.

The software is developed in Embedded C using Keil µVision 5 [10]. The program is flashed into the LPC2138 using Flash Magic. The firmware consists of initialization routines, UART driver functions, LCD driver functions, GPIO configuration, switch scanning, command filtering, command decoding, and output control logic. During Bluetooth operation, commands received from the Android Bluetooth terminal application are processed by the UART command decoder. During manual operation, switch inputs are read and the corresponding output states are updated.

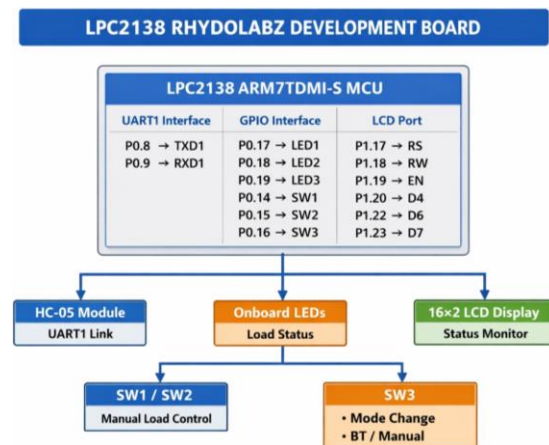


Fig. 2. Hardware interface mapping of LPC2138 RhydoLABZ development board.

Table I. Hardware and communication configuration

Parameter	Specification
Microcontroller	LPC2138 ARM7TDMI-S
Development board	RhydoLABZ LPC2138 development board
Bluetooth module	HC-05

Communication protocol	UART
UART channel used	UART1
Baud rate	9600 bps
UART frame format	8 data bits, no parity, 1 stop bit
Crystal frequency	12 MHz
Display	16×2 LCD in 4-bit mode
Output indicators	Onboard LEDs
Software tool	Keil µVision 5
Programming tool	Flash Magic

The hardware configuration shown in Table I defines the basic communication and control environment used for performance evaluation. The HC-05 module communicates with the LPC2138 through UART1 at 9600 bps, which is suitable for simple ASCII command transmission in short-range Bluetooth control applications [4]. The LPC2138 development board operates with a 12 MHz crystal oscillator and provides sufficient GPIO resources for interfacing LEDs, switches, and LCD display modules [1], [2].

Table II. Pin mapping of LPC2138 prototype

Component	LPC2138 Pin	Function
HC-05 TXD	P0.9	RXD1
HC-05 RXD	P0.8	TXD1
LED1	P0.17	Bulb / Load-1 indicator
LED2	P0.18	Fan / Load-2 indicator
LED3	P0.19	Mode / Status indicator
SW1	P0.14	Manual load-1 control
SW2	P0.15	Manual load-2 control
SW3	P0.16	Bluetooth / Manual mode selection
LCD RS	P1.17	LCD register select
LCD RW	P1.18	LCD read/write
LCD EN	P1.19	LCD enable
LCD D4	P1.20	LCD data line
LCD D5	P1.21	LCD data line
LCD D6	P1.22	LCD data line
LCD D7	P1.23	LCD data line

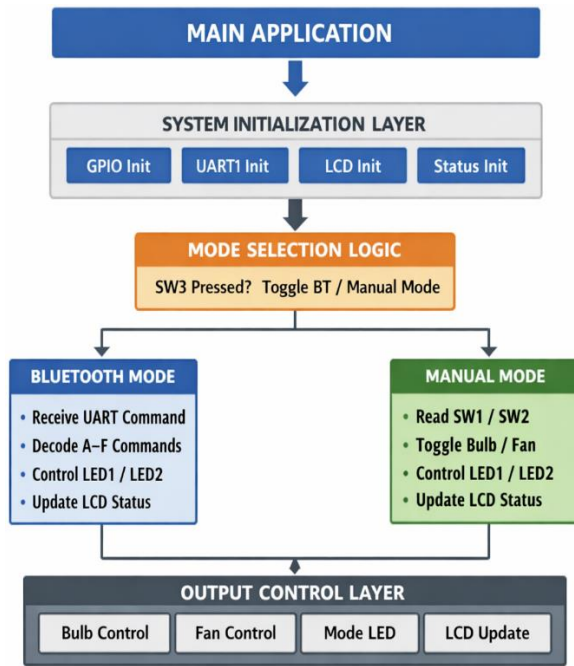


Fig. 3. Main firmware flow for Bluetooth and manual control modes.

The pin mapping in Table II was selected to separate communication, output control, manual input, and LCD feedback functions. UART1 pins P0.8 and P0.9 are dedicated to Bluetooth communication, while P0.17 to P0.19 are used for output indication. Switch inputs on P0.14, P0.15, and P0.16 allow local manual control and mode selection. The LCD is connected to Port 1 in 4-bit mode, which reduces the number of required data pins while still providing real-time status display [5].

VI. RESPONSE-LATENCY MEASUREMENT METHOD

The response-latency test was performed to observe the time required by the prototype to execute a Bluetooth command after it was transmitted from the Android Bluetooth terminal application. In this work, response delay is considered as the time interval between pressing a command button on the smartphone and observing the corresponding change in LED output or LCD status on the prototype.

The measured delay includes Bluetooth transmission from the smartphone, wireless reception by the HC-05 module, UART data transfer to LPC2138, command filtering, command decoding, GPIO output updating, and LCD status refreshing. Therefore, the measured value represents the complete practical response of the prototype instead of only the microcontroller execution time.

The Android Bluetooth terminal application was configured with command buttons for A, B, C, D, E, and F. Each command was tested repeatedly, and the output response was observed through the onboard LEDs and LCD display. The firmware was also designed to ignore carriage-return and newline characters because some Bluetooth terminal applications transmit these extra characters after the command.

Table III. Bluetooth command mapping

Command	Operation	Output Action
A	Bulb ON	LED1 ON
B	Bulb OFF	LED1 OFF
C	Fan ON	LED2 ON
D	Fan OFF	LED2 OFF
E	All ON	LED1 and LED2 ON
F	All OFF	LED1 and LED2 OFF
CR / LF	Ignored	No output change
Other character	Invalid command	LCD shows invalid command

The response delay was observed for each valid Bluetooth command. A total of 20 trials were considered for each command to check command reliability. A command was treated as successful when the correct LED output and LCD status were updated after transmitting the command. If the command was missed or not executed correctly during observation, it was treated as an unsuccessful trial.

Because the experiment was performed on a hardware prototype, the measured delay may vary slightly due to smartphone Bluetooth behavior, terminal application settings, power supply stability, environmental interference, and observation method. However, the values are useful for comparing the practical response behavior of the developed UART-based Bluetooth control system.

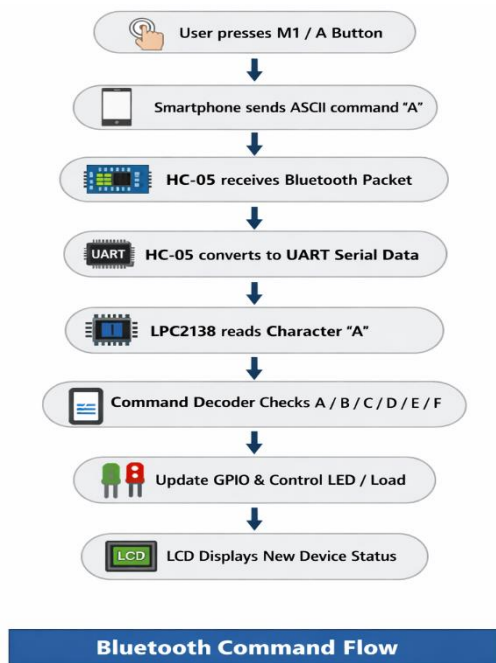


Fig. 4. Bluetooth command execution path used for response-delay observation.

VII. RESULTS AND DISCUSSION

The developed prototype was tested for Bluetooth command execution, response delay, command reliability, distance-wise behavior, and protocol suitability. The system successfully received commands from the Android Bluetooth terminal application through the HC-05 module and executed the corresponding output action using LPC2138 GPIO pins. The LCD display provided real-time feedback for valid commands, mode status, output condition, and invalid command conditions.

The response delay was measured for six Bluetooth commands: A, B, C, D, E, and F. Each command was tested for 20 trials. The observed delay values were in the range of 52 ms to 60 ms. The average response delay of the prototype was approximately 56 ms, which is suitable for short-range home automation control.

Table IV. Response delay and command reliability

Command	Function	Average Response Delay	Successful Trials	Reliability
A	Bulb ON	52 ms	20/20	100%
B	Bulb OFF	55 ms	20/20	100%
C	Fan ON	58 ms	19/20	95%
D	Fan OFF	54 ms	20/20	100%

E	All ON	60 ms	20/20	100%
F	All OFF	57 ms	20/20	100%
Average	—	≈56 ms	—	≈99.16%

From Table IV, it is observed that all commands except command C achieved 100% reliability during repeated trials. Command C achieved 95% reliability due to one unsuccessful or missed response during testing. This variation may occur because of Bluetooth terminal behavior, wireless communication disturbance, timing of command transmission, or observation limitations. Overall, the prototype showed reliable command execution for short-range Bluetooth-based automation.

Distance-wise testing was also performed to observe the practical Bluetooth response behavior. The system showed stable response at 1 m, 3 m, and 5 m. A slight delay was observed at 8 m, while response near 10 m became variable depending on obstacles, smartphone Bluetooth strength, and surrounding interference.

Table V. Distance-wise Bluetooth response observation

Distance	Response Observation	Remarks
1 m	Stable response	No noticeable delay
3 m	Stable response	Commands executed properly
5 m	Stable response	Reliable operation
8 m	Slight delay observed	Minor variation
10 m	Variable response	Depends on obstacle, phone strength, and environment

The results indicate that the system is suitable for room-level and short-range home automation applications. The HC-05 module provides sufficient wireless range for indoor prototype operation, while UART communication offers simple and direct serial data transfer between the Bluetooth module and LPC2138.

UART was selected instead of I2C because the HC-05 module directly supports serial UART communication through TXD and RXD pins [4]. I2C is mainly useful for short-distance chip-to-chip communication with peripherals such as RTC, EEPROM, sensors, and I/O expanders [6].

Therefore, UART is more suitable for Bluetooth command transmission in the present system.

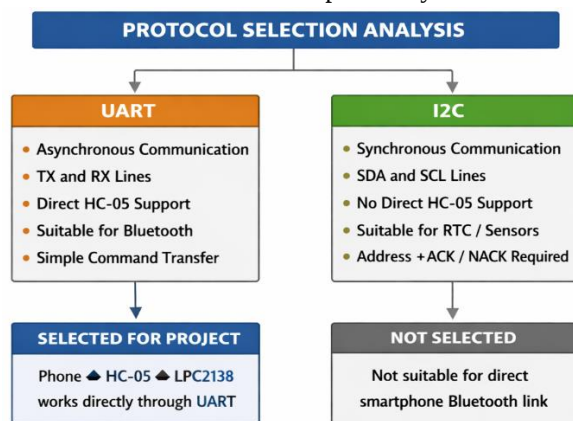


Fig. 5. Protocol selection analysis for HC-05-based Bluetooth communication.

The protocol selection analysis confirms that UART is the most appropriate communication method for this prototype because it requires only transmit, receive, and ground connections, and it does not require clock synchronization or device addressing. This reduces hardware complexity and firmware overhead for simple command-based Bluetooth automation. The CR/LF filtering logic further improves decoder stability by ignoring extra line-ending characters transmitted by Bluetooth terminal applications.

VIII. ADVANTAGES AND APPLICATIONS

The developed UART-based Bluetooth home automation prototype has several advantages. The system uses low-cost and easily available hardware components such as the LPC2138 development board, HC-05 Bluetooth module, LEDs, switches, and 16×2 LCD. Since the HC-05 directly supports UART communication, the hardware connection is simple and does not require additional protocol conversion circuitry.

Another advantage of the system is the use of custom Embedded C firmware. UART reception, command filtering, GPIO control, switch handling, LCD display, and output control are implemented at firmware level. This improves understanding of register-level embedded system design compared with ready-made high-level platforms. The use of LCD feedback also makes the system easier to test because command status, output condition, mode status, and invalid command information can be observed directly.

The prototype also supports both Bluetooth-based control and manual switch-based control. This

makes the system useful even when Bluetooth communication is not available. The CR/LF filtering logic improves command-decoder reliability by ignoring extra newline characters transmitted by Bluetooth terminal applications. The measured average response delay of approximately 56 ms shows that the system is suitable for short-range switching applications.

The proposed system can be used in small-scale home automation demonstrations, academic embedded system laboratories, Bluetooth control experiments, ARM7 microcontroller training, and prototype development of wireless appliance-control systems. With suitable relay driver circuits and electrical isolation, the same concept can be extended for controlling real AC appliances such as lamps, fans, and other household loads.

IX. CONCLUSION AND FUTURE SCOPE

This paper presented the performance evaluation of a UART-based Bluetooth home automation prototype using the LPC2138 ARM7 microcontroller and HC-05 Bluetooth module. The system successfully received ASCII commands from an Android Bluetooth terminal application and executed corresponding output actions through LPC2138 GPIO pins. A 16×2 LCD was used to provide real-time feedback for command status, output condition, selected mode, and invalid command handling.

The experimental results show that the prototype achieved an average response delay of approximately 56 ms for six Bluetooth commands. Command reliability was observed between 95% and 100% during repeated trials. The system showed stable response at short distances up to 5 m, minor delay at 8 m, and variable behavior near 10 m depending on obstacles, smartphone Bluetooth strength, and surrounding interference. These results indicate that UART-based HC-05 communication is suitable for short-range, low-cost, prototype-level home automation.

The use of UART communication simplified the interface between the HC-05 module and LPC2138 because only transmit, receive, and ground connections were required. Compared with I2C, UART is more suitable for HC-05-based Bluetooth command transfer, while I2C is better suited for peripheral devices such as RTC, EEPROM, and sensors. The CR/LF filtering logic improved command-decoder stability by ignoring extra line-

ending characters transmitted by Bluetooth terminal applications.

In future work, the prototype can be extended by adding relay-driver circuits with proper isolation for controlling real AC appliances. The system can also be upgraded with password-based Bluetooth access, EEPROM-based state memory, sensor feedback, power monitoring, mobile application development, and IoT connectivity using Wi-Fi or cloud-based monitoring. Additional testing using digital timing instruments can further improve the accuracy of response-latency measurement.

REFERENCES

- [1] NXP Semiconductors, "LPC2131/32/34/36/38 single-chip 16/32-bit microcontrollers; 32/64/128/256/512 kB ISP/IAP flash with 10-bit ADC and DAC," Datasheet, Rev. 05, 2011.
- [2] NXP Semiconductors, "UM10120: LPC2131/2/4/6/8 user manual," User Manual, 2002.
- [3] ARM Ltd., "ARM7TDMI technical reference manual," Technical Reference Manual, Revision 3.
- [4] Guangzhou HC Information Technology Co., "HC-05 Bluetooth to serial port module," Datasheet / AT command documentation.
- [5] Hitachi, "HD44780U dot matrix liquid crystal display controller/driver," Datasheet, 1999.
- [6] NXP Semiconductors, "UM10204: I2C-bus specification and user manual," User Manual, Rev. 7, 2021.
- [7] A. Al Tareq, M. R. Mostofa, M. J. Rana, and M. S. Rahman, "A comprehensive review of intelligent home automation systems using embedded devices and IoT," Control Systems and Optimization Letters, vol. 2, no. 2, pp. 198-203, 2024.
- [8] P. Khare, Gangacharan, A. Singh, S. Mishra, and A. Kumar, "Research paper on Bluetooth-controlled home automation using Arduino," International Journal of Novel Research and Development, vol. 9, no. 4, pp. i276-i280, Apr. 2024.
- [9] V. Pal, S. Kumar, V. Kumar, G. Jha, and P. Pant, "Voice controlled home automation system using Arduino," IITM Journal of Management and IT, vol. 12, no. 1, pp. 1-7, 2021.
- [10] Arm Keil, "Keil μ Vision IDE and MDK-Arm documentation," Technical Documentation.