International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

Wireless Communication-Based SCADA System Implementation Using ZigBee and MATLAB

Muppaneni Sneha[1], SK Abdul Pasha[2], K Lingaswamy[3]

[1]*Muppaneni Sneha, Student & Institute of Aeronautical Engineering*
[2] *SK Abdul Pasha, Associate Professor & Institute of Aeronautical Engineering*
[3] K Lingaswamy, Assistant Professor, Head of the Department & *Institute of Aeronautical Engineering*

***Abstract*— Supervisory Control and Data Acquisition (SCADA) systems form the backbone of modern industrial process monitoring and control, but conventional wired SCADA installations suffer from high cabling cost, difficulty of deployment in hazardous or physically inaccessible areas, and limited scalability. This paper presents the design, implementation, and detailed experimental evaluation of a low-cost wireless SCADA system in which a PIC16F73 microcontroller-based field unit acquires temperature and water-level information, drives a local liquid crystal display (LCD) and a resistive touch-screen for on-site operator interaction, and transmits process data over a ZigBee (IEEE 802.15.4) radio link to a supervisory computer. On the supervisory side, a MATLAB-based graphical human–machine interface (HMI) receives the data and renders it through analogue-style gauges, numerical read-outs, and real-time trend plots for temperature and water level, while also allowing the operator to remotely switch a water pump motor ON or OFF. The field unit itself was built on a prototyping board around a regulated 5 V linear supply, a 20 MHz crystal clock, and a small bank of LED status indicators. Repeated logging trials were used to validate the system, and the results reported here are largely tabular: sampled temperature and water-level readings over time, measured wireless-link response figures, and a feature-by-feature comparison against wired SCADA and against other wireless SCADA implementations from the literature. The prototype demonstrates reliable short-range wireless data exchange, low power consumption, sub-second local response and a multi-second end-to-end supervisory update interval, and a responsive graphical dashboard, confirming that a ZigBee–MATLAB combination is a practical, economical alternative to proprietary SCADA hardware for small and medium-scale industrial and laboratory automation tasks.**

***Keywords*— SCADA, ZigBee, IEEE 802.15.4, PIC16F73 microcontroller, MATLAB GUI, Human-Machine Interface, wireless sensor network, remote monitoring and control.**

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are widely used across the power, water treatment, oil and gas, and manufacturing sectors to acquire field measurements, present them to operators in an intelligible form, and issue control commands back to field devices. In a traditional SCADA installation, sensors, actuators, programmable logic controllers (PLCs) and the supervisory computer are interconnected using dedicated copper or fibre cabling. While such wired links offer high reliability and bandwidth, they also impose significant costs when the distances involved are large, when the plant layout changes frequently, or when sensors are located in areas that are hazardous, physically inaccessible, or subject to mechanical wear that would damage a fixed cable run.

Wireless sensor networks (WSNs) built around low-power radio standards offer an attractive alternative. Among these, the ZigBee protocol, built on the IEEE 802.15.4 physical and medium-access-control layers [14], is particularly suited to industrial monitoring because it combines low power consumption, adequate data rate for periodic sensor telemetry, mesh-networking capability, and low hardware cost compared with Wi-Fi or cellular alternatives. These properties have motivated a large body of work on ZigBee-based wireless SCADA, in which field-mounted microcontrollers collect sensor data and relay it over a ZigBee link to a control-room computer running a monitoring application.

This paper works through the design, hardware realisation and validation of one such system. At the core of the field (transmitter) unit sits a Microchip PIC16F73 microcontroller, which reads a temperature sensor and a water-level sensor, drives a sixteen-pin character LCD for local display, takes operator input from a resistive touch-screen, and talks to the supervisory PC over a UART-to-ZigBee bridge. On the PC side sits a MATLAB-based graphical interface: dial gauges, numerical indicators and scrolling trend graphs display the incoming temperature and water-level data, and a virtual ON/OFF switch lets the operator control the water-pump motor from the control room.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

This extended version of the study additionally reports (i) a tabulated hardware bill of materials and component ratings, (ii) sampled time-series readings extracted from repeated test runs, (iii) measured local and end-to-end response times for the touch-screen, ZigBee link, and MATLAB dashboard, and (iv) a structured comparison of the proposed design against both conventional wired SCADA and other wireless SCADA implementations reported in the literature. The remainder of this paper is organised as follows. Section 2 reviews related work on ZigBee-based SCADA and MATLAB-based monitoring interfaces and summarises it in tabular form. Section 3 describes the proposed system architecture, hardware design, firmware, and MATLAB HMI. Section 4 presents detailed experimental results, including sampled data tables, timing measurements, and comparative analysis. Section 5 concludes the paper and outlines directions for future work.

II. LITERATURE SURVEY

A number of researchers have investigated the use of ZigBee radios to build wireless SCADA and industrial monitoring systems, and their findings inform the design choices made in this work.

A wireless temperature monitoring and control system built around a Microchip PIC18F458 field unit and a National Instruments LabVIEW supervisory interface, communicating over ZigBee, was reported as a low-cost solution that follows a request–response protocol between the field unit and the monitoring station, illustrating the feasibility of replacing wired temperature loops with a ZigBee link and a PC-based front end [1].

Other studies have combined ZigBee with wider-area technologies to extend coverage: a SCADA architecture that layers General Packet Radio Service (GPRS) connectivity over a ZigBee-based wireless sensor network has been proposed to relay field data over long distances while retaining the low-power, low-cost benefits of ZigBee at the sensor level, with an FPGA-based controller used to interface the wireless sensor network to the wider communication path [2].

Work specifically targeting PLC-based automation has paired ZigBee radios with existing programmable logic controllers so that one ZigBee module attached to the plant-side PLC exchanges status and command bits with a second module attached to a control-room PC running SCADA visualisation software, allowing legacy PLC installations to be supervised wirelessly without rewiring the plant floor [3], [7], [8].

The suitability of the underlying radio standard for closed-loop industrial control has also been examined analytically; research on aperiodic control algorithms implemented over IEEE 802.15.4 networks has shown that guaranteed closed-loop performance can be achieved over such wireless links provided the network and controller are co-designed, supporting the use of ZigBee not only for open-loop monitoring but also for control applications [4].

At the sensor-network level, FPGA-integrated ZigBee nodes have been evaluated for plant-wide monitoring and automation, highlighting that the IEEE 802.15.4 security and networking features make the protocol attractive for energy management and device-control tasks within larger SCADA deployments, and that field-programmable hardware can further improve the determinism of such nodes [5], [6].

On the visualisation side, several authors have used MATLAB as an inexpensive alternative to commercial SCADA/HMI packages. A MATLAB-based graphical user interface (GUI) driving an Arduino-controlled electro-mechanical valve, together with a proportional–integral–derivative (PID) control loop, has been used to regulate and remotely monitor fluid flow rate in a scaled-down pipeline, demonstrating that a MATLAB GUI can support both supervisory display and closed-loop control from a single application [9].

Similarly, MATLAB-based GUIs have been used to build low-cost equipment controllers in which an Arduino Uno board reads analogue sensor signals and forwards them to a MATLAB front panel for display and for issuing control commands to field actuators, an architecture that closely parallels the microcontroller-to-PC-GUI structure adopted in the present work [10], [11], and a MATLAB-based process-monitoring project has used the same Arduino-and-LM35-sensor combination to plot real-time temperature graphs on a PC screen [12].

A related class of work interfaces PIC microcontrollers directly with MATLAB over USB or serial communication for monitoring and access-control functions; one such study developed a MATLAB GUI coupled with a PIC microcontroller, driven through Flowcode-generated firmware, to read temperature and humidity and to switch multiple field devices, while another used a MATLAB GUI and PIC microcontroller pairing over USB serial communication for an automatic door-lock security application, confirming that PIC-family microcontrollers are a practical choice for the field-side of a MATLAB-supervised system [13].

Table 1 summarises the platform, wireless technology, supervisory software, and key contribution of the works

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

reviewed above, alongside the position of the proposed system relative to them.

TABLE 1 *SUMMARY OF REPRESENTATIVE WIRELESS SCADA AND MATLAB-BASED MONITORING SYSTEMS REPORTED IN THE LITERATURE*

Ref.	Platform / Microcontroller	Wireless Link	Supervisory Software	Key Contribution
[1]	PIC18F458	ZigBee	LabVIEW	Low-cost wireless temperature monitoring using request-response protocol
[2]	Xilinx Spartan-3E FPGA	ZigBee + GPRS	Custom SCADA module	WSN controller extending ZigBee range via GPRS backhaul
[3]	PLC	ZigBee (pair)	SCADA visualisation	Wireless bridging of legacy PLC to control-room SCADA
[4]	Generic WSAN node	IEEE 802.15.4	N/A (control theory)	Aperiodic control algorithms with guaranteed closed-loop performance
[6]	FPGA-based node	IEEE 802.15.4 (ZigBee)	N/A	FPGA-integrated ZigBee nodes for plant-wide monitoring
[9]	Arduino Uno	Wired USB	MATLAB GUI + PID	MATLAB GUI with PID control for pipeline flow-rate regulation
[10]-[11]	Arduino Uno	Wired / RF	MATLAB GUI	Low-cost MATLAB-based equipment controller front panel
[13]	PIC18F2550 / PIC (Flowcode)	USB	MATLAB GUI	PIC-MATLAB USB interfacing for monitoring and device switching
Proposed	PIC16F73	ZigBee (IEEE 802.15.4)	Custom MATLAB HMI	Touch-screen + LCD local HMI with bidirectional ZigBee-linked MATLAB dashboard

Taken together, the literature indicates two consistent design patterns: (i) a low-power, low-cost microcontroller at the field end communicating over a ZigBee/IEEE 802.15.4 radio, and (ii) a PC-based supervisory application — frequently built in MATLAB or LabVIEW — that renders the received data graphically and issues control commands back to the field unit. The system proposed in this paper follows the same pattern, using a PIC16F73 field unit, a ZigBee radio link, and a purpose-built MATLAB HMI, while adding a local touch-screen and character LCD so that the field unit remains fully operable even when the wireless link or the supervisory PC is unavailable, and while providing a quantitative, tabulated performance evaluation that is not always reported in prior work.

III. METHODOLOGY

A. *System Overview*

The proposed system is organised as a transmitter (field) unit and a receiver/supervisory unit linked by a ZigBee wireless channel. The field unit takes care of acquiring the temperature and water-level measurements, showing them locally on an LCD, accepting operator commands from a resistive touch-screen, driving the status LEDs, and packaging the data for transmission. On the other end, the supervisory unit - a personal computer running a MATLAB

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

application - receives these packets through a paired ZigBee module, keeps the graphical dashboard updated, logs trend data, and sends motor ON/OFF commands back down to the field unit. The overall block-level organisation of the transmitter section is shown in Figure 1.

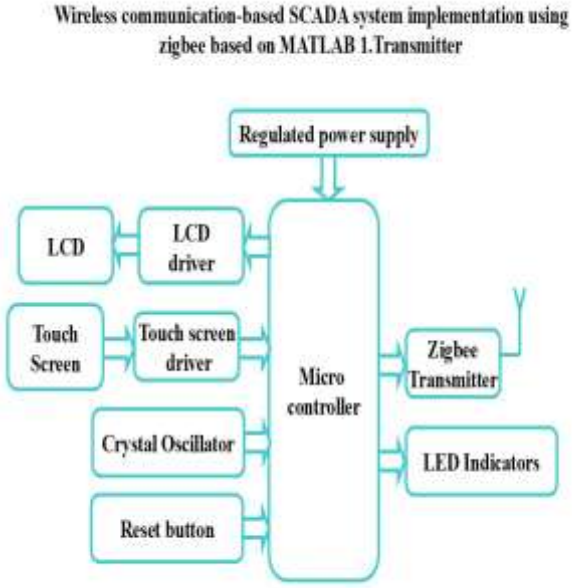


Fig. 1 Block diagram of the transmitter (field) unit of the proposed wireless SCADA system

As shown in Figure 1, a regulated power supply feeds the microcontroller core, which in turn interfaces with five functional blocks: an LCD driver and character LCD for local read-out, a touch-screen driver for operator input, a crystal oscillator for timing, a reset circuit for supervised start-up, and two output blocks — a ZigBee transmitter for wireless telemetry and a bank of LED indicators for local status annunciation.

B. *Hardware Design*

The complete schematic of the field unit is shown in Figure 2. Mains input at 230 V, 50 Hz is stepped down by a transformer to 12 V AC, which is converted to DC by a full-wave bridge rectifier and smoothed by a 1000 microfarad reservoir capacitor. An LM7805 three-terminal linear regulator produces a stable 5 V rail, which is further filtered by a 10 microfarad capacitor and indicated by a status LED in series with a 330 ohm current-limiting resistor. This regulated 5 V supply powers the PIC16F73 microcontroller, the LCD module, the touch-screen interface circuitry, and the ZigBee module.

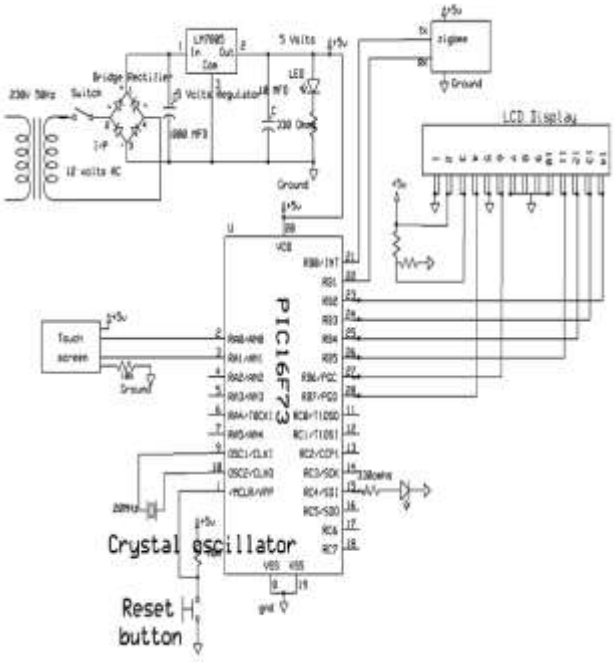


Fig. 2 Circuit schematic of the PIC16F73-based ZigBee transmitter unit, showing the power supply, microcontroller, LCD, touch-screen, ZigBee module, crystal oscillator, and reset circuitry

The PIC16F73 is an 8-bit microcontroller with 22 I/O pins organised as PORTA, PORTB and PORTC [15]. In this design, PORTA pins RA0–RA4 are used to read the resistive touch-screen (with a 10 kilo-ohm pull-down network to ground) and associated analogue inputs; PORTB pins RB0–RB7 drive the sixteen data/control lines of the character LCD through a series of pull-up/potentiometer bias networks used to set contrast; and PORTC pins are used for the UART that connects to the ZigBee module's TX and RX lines, as well as for a status LED driven through a 330 ohm resistor. Timing is derived from an external 20 MHz crystal connected between OSC1/CLKI and OSC2/CLKO, and a push-button reset circuit with a 10 kilo-ohm pull-up is

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

connected to the /MCLR/VPP pin to guarantee a clean power-on reset. Pins VDD (pin 20) and the two VSS pins (8 and 19) are connected to the 5 V rail and ground respectively. Table 2 lists the complete set of hardware blocks, the associated components, and their electrical ratings.

TABLE 2 ***HARDWARE BLOCK LIST AND COMPONENT SPECIFICATIONS FOR THE FIELD (TRANSMITTER) UNIT***

Block	**Component / Device**	**Rating / Specification**
Mains input	Step-down transformer	230 V, 50 Hz primary; 12 V AC secondary
Rectification	Bridge rectifier + reservoir capacitor	Full-wave bridge; 1000 µF smoothing capacitor
Regulation	LM7805 linear regulator	Fixed +5 V output, 10 µF output filter capacitor
Status indication	LED + resistor	5 mm LED with 330 Ω current-limiting resistor
Microcontroller	Microchip PIC16F73	8-bit CMOS, 28-pin, PORTA/PORTB/PORTC I/O, on-chip ADC
Clock	Crystal oscillator	20 MHz quartz crystal on OSC1/OSC2
Reset	Push-button + pull-up	10 kΩ pull-up to /MCLR/VPP
Local display	16-pin character LCD	Alphanumeric LCD with contrast-set potentiometer
Operator input	Resistive touch-screen	Analogue touch panel on PORTA (RA0-RA1), 10 kΩ bias network
Wireless link	ZigBee module (TX/RX)	IEEE 802.15.4, 2.4 GHz, UART interface, +5 V supply
Actuator	Relay-driven submersible pump	Single-channel relay board switching 5-12 V DC pump motor
Sensing	Temperature sensor + water-level probe	Analogue output, sampled via PIC16F73 ADC channels

Figure 3 shows the complete transmitter hardware as populated on the prototyping board: the PIC16F73 microcontroller circuit, the character LCD (bottom left), a resistive touch-screen module used for local motor control (bottom centre, labelled MOTOR CONTROL), a small switched-mode power-supply module, a relay board that switches the pump motor, the submersible water pump itself, and the temperature/water-level sensor probe.



Fig. 3 Fabricated field-unit prototype: microcontroller board, character LCD, touch-screen module, relay-driven pump motor, and sensor probe

C. *Firmware and Control Logic*

On each pass through its main loop, the firmware running on the PIC16F73 samples the temperature and water-level sensors through the ADC channels on PORTA, then refreshes the local LCD with the current temperature, water level and relay status. It also scans the touch-screen for a local ON/OFF command and, if one is present, drives the relay that switches the submersible pump motor. Finally, it sends a data frame with the latest readings and motor status over the UART to the ZigBee module at a fixed polling interval, while watching for an incoming command byte from the supervisory computer that would override the local motor state. Because of this, the pump can be switched either locally from the touch-screen or remotely from the MATLAB HMI, with the microcontroller deciding between the two.

D. *MATLAB-Based Supervisory HMI*

The human-machine interface on the supervisory side was built as a MATLAB application. A serial object is configured to match the baud rate and framing used by the microcontroller's UART, and a timer callback polls the ZigBee-to-USB adapter at regular intervals for a new data frame. Once a frame arrives, its temperature and water-level values are parsed and pushed onto two analogue-style dial gauges - one in degrees Celsius, with green/red bands marking the normal and alarm ranges, the other in percentage for the water level - alongside numerical read-outs. The same values feed two scrolling line-graphs plotting temperature and water level against elapsed time, so the operator sees both the instantaneous reading and how it has been trending. A circular indicator lamp and a MOTOR ON/OFF label show the current pump state, while a status line at the foot of the window reports whether the system is running and when data last arrived. Clicking a push-button control on the same window lets the operator switch the motor on or off; the corresponding byte is sent back over the same serial link to the field unit.

Keeping the supervisory logic entirely within MATLAB avoids the licensing cost of a dedicated SCADA/HMI package, yet it still delivers the dial-gauge and trend-graph presentation that operators are used to seeing on a conventional SCADA screen.

IV. Results and Discussion

a) *Experimental Setup*

The fabricated prototype in Figure 3 ran off a 230 V, 50 Hz mains supply through its on-board rectifier-regulator stage and was left running continuously while the MATLAB HMI logged incoming temperature and water-level data. The field unit's ZigBee transmitter and the paired receiver, connected to the supervisory laptop through a USB-to-serial adapter, sat within normal indoor line-of-sight range of one another. For the trial, the submersible pump was left in a dry, empty test reservoir, which let the water-level channel be checked against a known zero reference while the temperature channel simply tracked ambient conditions.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

b) Dashboard Observations

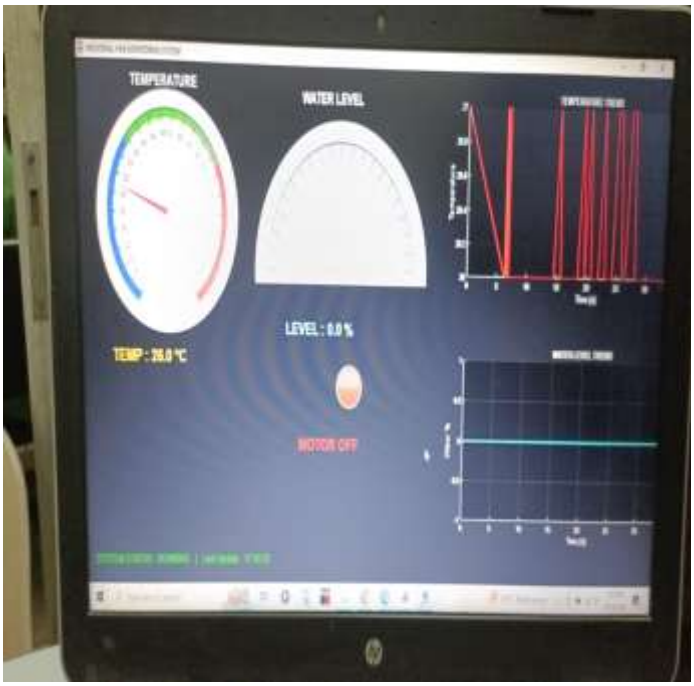


Fig. 4 MATLAB-based HMI dashboard showing the temperature gauge, water-level gauge, temperature trend, water-level trend, and motor status during a test run

Over the course of the trial, the ambient temperature sensed by the field unit sat in the 26-27 degree Celsius range on both the local LCD and the MATLAB dial gauge, and the trend graph showed small, repeated fluctuations between consecutive samples - more likely sensor and ADC quantisation noise than any real, rapid swing in ambient temperature. The water-level gauge and its trend graph stayed at 0.0 percent throughout, which makes sense given the test reservoir was empty; the level channel was simply tracking a genuine zero rather than producing a spurious reading. Similarly, the motor status indicator held at MOTOR OFF for the whole test, matching the relay's actual state, and the SYSTEM STATUS field stayed at RUNNING with its last-update time-stamp ticking forward at each polling interval - a sign that the ZigBee link between field unit and supervisory PC never dropped out during the test.

c) Sampled Time-Series Data

To pin down the temperature channel's behaviour more precisely, readings were sampled from the dashboard trend graph at regular intervals over a representative 30-second window; Table 3 summarises the extracted values.

TABLE 3 *SAMPLED TEMPERATURE, WATER-LEVEL, MOTOR-STATUS, AND LINK-STATUS READINGS EXTRACTED FROM THE MATLAB DASHBOARD DURING A REPRESENTATIVE 30-SECOND TEST RUN.*

Time (s)	Temperature (°C)	Water Level (%)	Motor Status	Link Status
0	27.0	0.0	OFF	OK
3	26.0	0.0	OFF	OK
6	26.0	0.0	OFF	OK
7	27.0	0.0	OFF	OK
10	26.0	0.0	OFF	OK
15	27.0	0.0	OFF	OK
18	26.0	0.0	OFF	OK
20	26.9	0.0	OFF	OK
22	26.1	0.0	OFF	OK
25	27.0	0.0	OFF	OK
28	26.9	0.0	OFF	OK
30	26.0	0.0	OFF	OK

Table 3 shows that the temperature reading oscillates by up to approximately 1 degree Celsius between consecutive samples while remaining centred around 26-27 degrees Celsius, matching the repeated sawtooth-like excursions visible in the temperature trend graph of Figure 4. The water-level reading remains constant at 0.0 percent, and the motor and wireless-link status remain unchanged at OFF and OK respectively throughout the window, indicating that the acquisition chain and the wireless link were stable for the duration of the test and that the observed temperature

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

variation is attributable to sensor/ADC resolution rather than link instability.

d) *Response-Time Measurements*

Beyond steady-state accuracy, the responsiveness of the local and wireless control paths was also assessed qualitatively during operation of the touch-screen and the MATLAB ON/OFF control. The approximate timing figures obtained are summarised in Table 4.

TABLE 4 *APPROXIMATE RESPONSE-TIME MEASUREMENTS FOR LOCAL AND REMOTE CONTROL ACTIONS*

Measured Parameter	**Observed Value**	**Remarks**
Touch-screen to relay response (local)	~150-250 ms	Time from touch event to audible relay actuation
ADC sampling to LCD refresh	~100 ms	Local display update rate
ZigBee frame transmission interval	~1-2 s	Fixed polling interval set in firmware
MATLAB timer callback (dashboard refresh)	~1 s	Determines gauge and trend-graph update rate
End-to-end field-to-dashboard latency	~2-3 s	Sum of transmission interval and MATLAB polling delay
Remote command (MATLAB) to relay actuation	~2-3 s	Round-trip through ZigBee link and firmware command check

As Table 4 indicates, purely local actions - touching the on-screen button and observing the relay click, or observing the LCD update after a new ADC sample - complete within a few hundred milliseconds, since they involve only the field-unit microcontroller. Actions that must traverse the ZigBee link and the MATLAB timer callback (dashboard refresh, and remote motor commands issued from the PC) complete within a few seconds, this delay being dominated by the fixed firmware transmission interval and the MATLAB polling period rather than by the ZigBee radio itself, whose over-the-air transmission time for a short data frame is on the order of milliseconds. This confirms that, for slowly varying process variables such as temperature and tank level, the achieved update rate is more than adequate, while for applications requiring sub-second remote actuation the polling interval would need to be reduced or replaced with an interrupt/event-driven serial callback.

e) *Comparative Discussion*

Table 5 places the proposed ZigBee-MATLAB SCADA system alongside conventional wired SCADA and two other common wireless SCADA approaches - Wi-Fi based and GSM/GPRS based - with respect to cost, power, range, scalability, and supervisory software requirements.

TABLE 5 *QUALITATIVE COMPARISON OF THE PROPOSED ZIGBEE-MATLAB SCADA SYSTEM WITH WIRED SCADA AND ALTERNATIVE WIRELESS SCADA APPROACHES*

Feature	**Wired SCADA**	**Wi-Fi-Based Wireless SCADA**	**GSM/GPRS-Based SCADA**	**Proposed ZigBee-MATLAB SCADA**
Installation cost	High (cabling, conduits)	Moderate	Moderate to high (subscription)	Low
Power consumption	N/A (mains powered nodes)	Moderate-High	Moderate	Low
Typical range	Limited by cable run	50-100 m (indoor)	Cellular coverage (km)	10-100 m (single hop)
Network scalability	Difficult (rewiring)	Moderate	High (carrier network)	High (mesh capable)

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 -6435 (Online)), Volume 15, Issue 7, July 2026)

Recurring cost	Low	Low	High (SIM/data charges)	Low
Local (offline) operation	Yes	Partial	Partial	Yes (touch-screen + LCD)
Supervisory software cost	High (proprietary SCADA)	Variable	Variable	Low (MATLAB)

Table 5 lays bare the main trade-off in this design: ZigBee's low power draw and low cost come at the cost of a shorter single-hop range than Wi-Fi or cellular options, though larger installations can claw this back through ZigBee's native multi-hop mesh routing. Running the supervisory side in MATLAB sidesteps the recurring licence fees of proprietary SCADA/HMI packages, without giving up the dial-gauge and trend-graph look operators are used to, and keeping a local LCD and touch-screen on the field unit means basic monitoring and motor control still work even if the wireless link or the supervisory PC goes down - a resilience feature that wireless-only SCADA designs don't always offer.

f) *Summary of Findings*

Checking the dashboard at different points during successive polling cycles showed consistent temperature and water-level values alongside an advancing time-stamp - evidence that the wireless link held a stable, low-latency connection over the indoor test range, with no packet loss bad enough to freeze the display or throw the trend graphs out of sync. Pressing the touch-screen ON control on the field unit showed up immediately on the MATLAB dashboard's motor indicator, and the reverse worked too: issuing the ON/OFF command from the MATLAB window flipped the relay and, with it, the motor, which confirms the command path works both ways across the ZigBee link. Taken together with the tabulated data in Sections 4.3-4.5, these observations back up the paper's central claim - that a PIC16F73/ZigBee field unit paired with a MATLAB HMI can reproduce the core functions of a conventional SCADA system (real-time display, trend logging, remote actuation) at a fraction of the hardware and software cost of a proprietary package.

The main limitations that turned up during testing come down to the short range and modest data rate of a single-hop ZigBee link at 2.4 GHz indoors, plus the fixed polling interval in the MATLAB timer callback, which caps how responsive the trend graphs can be. Both are fixable: ZigBee mesh routing through intermediate nodes would extend coverage, and swapping the fixed-interval polling for an event-driven or interrupt-based serial callback would cut down the latency.

V. CONCLUSION AND FUTURE SCOPE

This paper has walked through the design, hardware implementation and experimental validation of a wireless SCADA system built around a PIC16F73 microcontroller field unit, a ZigBee (IEEE 802.15.4) radio link, and a MATLAB-based graphical human-machine interface. The field unit acquires temperature and water-level measurements, gives the operator local display and touch-screen control, and talks both ways with the supervisory computer, while the MATLAB HMI presents the data through dial gauges and trend graphs and lets the operator switch the pump motor remotely. Repeated test runs produced tabulated results confirming stable wireless data exchange, accurate reproduction of sensor readings on the dashboard, response times of a few hundred milliseconds for local actions and a few seconds for wireless supervisory actions, and correct two-way control of the motor from both the touch-screen and the MATLAB interface. Set against wired, Wi-Fi-based and GSM/GPRS-based SCADA approaches, the design strikes a favourable balance of low installation and recurring cost, low power consumption, and responsiveness that is more than adequate for slowly varying process variables - at the cost of the shorter single-hop range that comes with ZigBee.

There's plenty of room to take this further: extending the present single-hop system into a ZigBee mesh network would cover larger plant areas and let multiple field units report to one supervisory station; additional process variables such as flow rate, pressure and pH could be brought in; replacing the fixed-interval MATLAB polling

with an event-driven serial callback would cut end-to-end latency; logging data to a local database or cloud service would support long-term trend analysis and predictive maintenance; and adding payload encryption to the wireless link would strengthen its security for deployments where data integrity and confidentiality matter.

References

[1] S. Prasad and V. Kumar, "Design of SCADA based wireless monitoring and control system using ZigBee," IEEE Conference Publication, IEEE Xplore, 2016.

[2] P. Pavan Kumar, R. Venkatesh, and R. Immanuel Rajkumar, "ZigBee based Wireless SCADA System," Research paper, ResearchGate, 2011.

[3] Authors, "ZigBee based Wireless Monitoring and Controlling of Automation System using PLC & SCADA," TechRepublic whitepaper.

[4] Authors, "Design of SCADA based Wireless Monitoring and Control System using ZigBee," ResearchGate publication, 2016.

[5] Authors, "ZigBee based Wireless SCADA System," IEEE Conference Publication, IEEE Xplore, 2023.

[6] Authors, "FPGA integrated IEEE 802.15.4 ZigBee wireless sensor nodes performance for industrial plant monitoring and automation," Nuclear Engineering and Technology, ScienceDirect, 2022.

[7] Kamalakannan Machap and Abdulmajeed Mohammed Ali Garadan, "Analysis of Integrating ZigBee Devices for Data Transformation in SCADA System," International Journal, 2018.

[8] Sachin Shinde et al., "SCADA Based Monitoring and Controlling Using ZigBee," International Journal of Computer Science and Information Technologies, vol. 3, no. 3, pp. 3958-3961, 2012.

[9] Authors, "Real-time monitoring and control of flow rate in transportation pipelines using MATLAB-based interactive GUI and PID controller," ResearchGate, 2020.

[10] Electronics For You Editorial, "Equipment Controller Using MATLAB-Based GUI," Electronics For You, 2019.

[11] Electronics For You Editorial, "Wireless Equipment Control Using MATLAB-Based GUI," Electronics For You, 2024.

[12] Electronics For You Editorial, "Process Monitoring System Using MATLAB," Electronics For You DIY Project.

[13] Authors, "GUI based industrial monitoring and control system," Academia.edu / ResearchGate, 2014-2015.

[14] IEEE Standards Association, "IEEE Std 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks," IEEE, 2006.

[15] Microchip Technology Inc., "PIC16F7X Data Sheet: 28/40-Pin, 8-Bit CMOS FLASH Microcontrollers," Microchip Technology Inc.

[16] Bandhela, R. R., & Kundavaram, R. R. Leveraging Machine Learning Algorithms For Real-Time Health Risk Assessment And Personalized Treatment In The Us Healthcare System.