IoT Toolkit Manual

AWS Cloud Integration with SD Card Data Storage

Version: 1.0

Date: August 2025

Document Type: Technical Manual

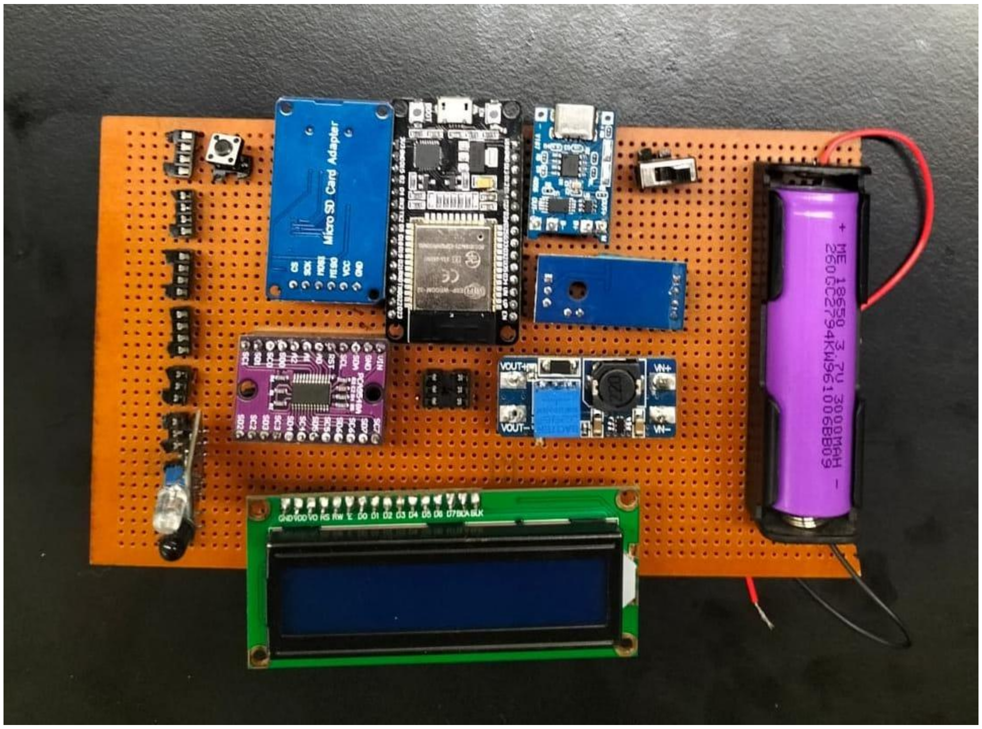
Table of Contents

---

1. Introduction

Welcome to the comprehensive manual for our advanced IoT Toolkit designed for seamless AWS cloud integration. This cutting-edge solution bridges the gap between local sensor data collection and cloudbased analytics, providing researchers, engineers, and developers with a robust platform for Internet of Things applications.

Our IoT Toolkit represents a sophisticated approach to modern data acquisition and cloud connectivity. The system is engineered to operate in both online and offline modes, ensuring continuous data collection regardless of network connectivity status. This dual-mode operation makes it particularly suitable for research environments, industrial monitoring, and remote sensing applications where network reliability may vary.



The toolkit's versatility stems from its ability to interface with multiple sensor types while maintaining high-performance data processing and storage capabilities. Whether you're conducting environmental monitoring, industrial process control, or academic research, this toolkit provides the flexibility and reliability required for professional-grade IoT deployments.

Key advantages of our system include real-time data visualization, automatic cloud synchronization, local data redundancy through SD card storage, and an intuitive web-based configuration interface. The system's architecture ensures that data integrity is maintained throughout the collection, processing, and transmission pipeline.

---

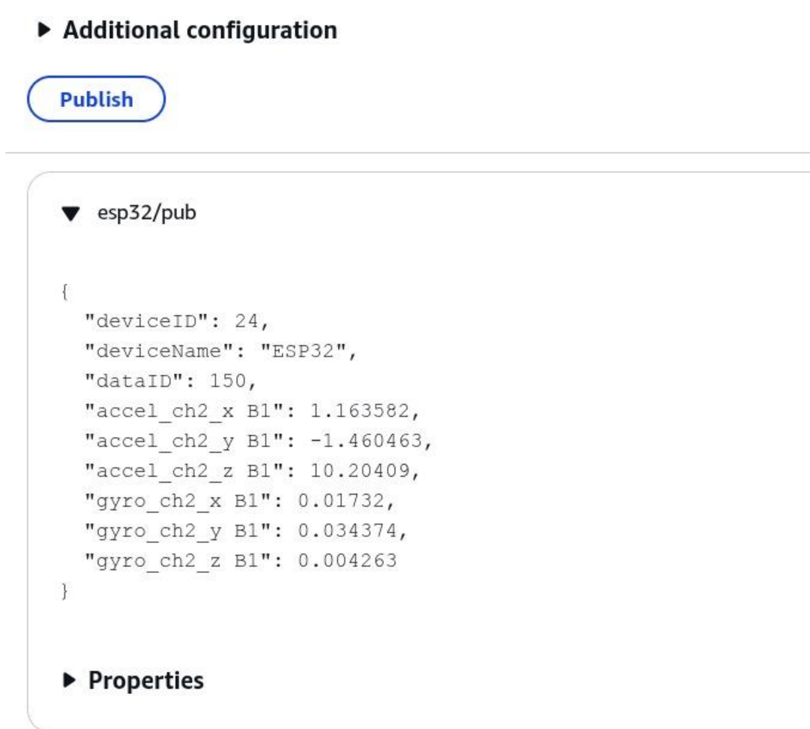
2. Use of Our Kit

2.1 Dual Mode Operation

Our IoT toolkit operates in two distinct modes, each designed to serve specific use cases and operational requirements:

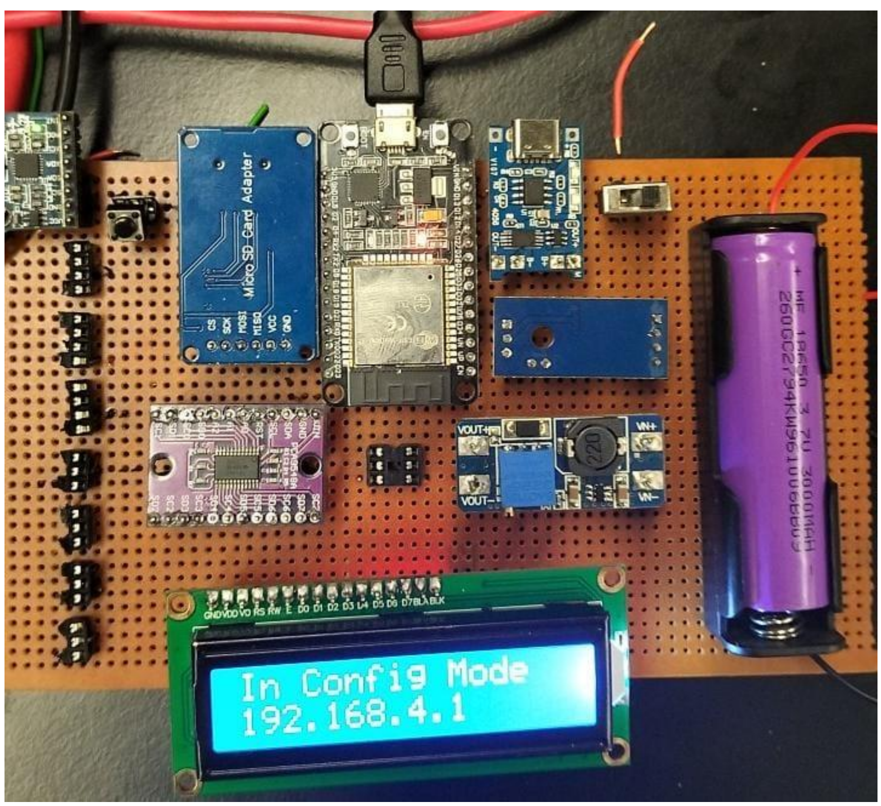
Online Mode (Cloud-Connected)

In online mode, the toolkit establishes a continuous connection to your configured Wi-Fi network and transmits sensor data directly to AWS cloud services. This mode enables real-time monitoring and analysis through cloud-based dashboards and analytics platforms. The system automatically handles data formatting, compression, and secure transmission protocols to ensure reliable cloud delivery.

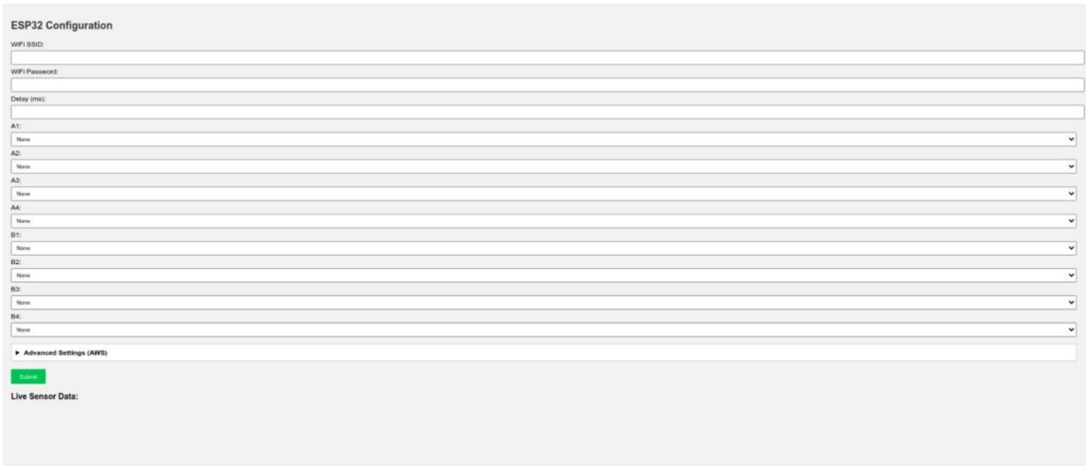


Offline Mode (Local Operation)

Offline mode ensures uninterrupted data collection even when network connectivity is unavailable. In this configuration, all sensor data is stored locally on the integrated SD card while simultaneously being available for viewing through the toolkit's built-in web interface. This mode is particularly valuable for remote deployments or environments with unreliable internet connectivity.



- **2.2 Data Visualization**
- The toolkit provides comprehensive data visualization through interactive graphs and charts powered by Chart.js library. Users can monitor sensor readings in real-time, analyze historical trends, and identify patterns in their data. The visualization system supports multiple chart types including line graphs, bar charts, and scatter plots, making it suitable for various data analysis requirements.



The system provides comprehensive data visualization capabilities for monitoring sensor readings in real-time and analyzing historical trends through the integrated web interface.

2.3 Port Configuration and Sensor Connectivity

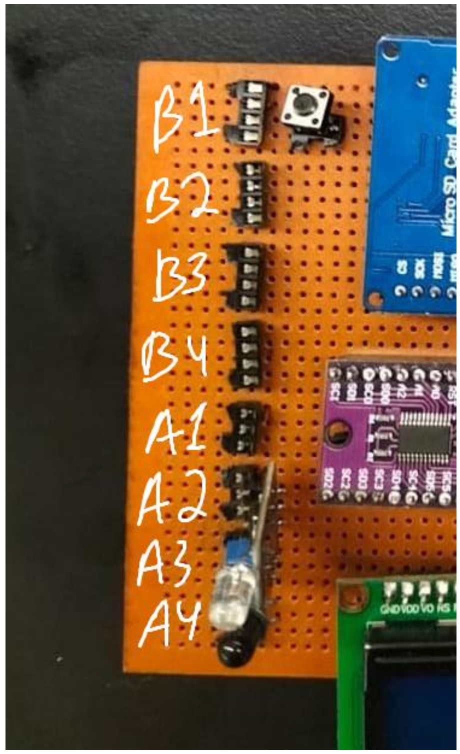
Our toolkit features 8 general-purpose ports strategically divided into two groups to accommodate different sensor types:

Group A: Analog Sensor Ports (A1-A4)

- **Port Count:** 4 dedicated analog ports
- **Connection Type:** Direct ADC GPIO connection
- **Pin Configuration:** 3-pin connector per port (VCC, GND, Signal)
- **Voltage Range:** 0-3.3V
- **Resolution:** 12-bit ADC (4096 discrete values)

Group B: I2C Sensor Ports (B1-B4)

- **Port Count:** 4 dedicated I2C ports
- **Connection Type:** I2C multiplexer module
- **Pin Configuration:** 4-pin connector per port (VCC, GND, SDA, SCL)
- **Operating Voltage:** 3.3V/5V selectable
- **Bus Speed:** Fixed at 500 kHz for optimal performance
- **Address Range:** 7-bit addressing (0x08 to 0x77)

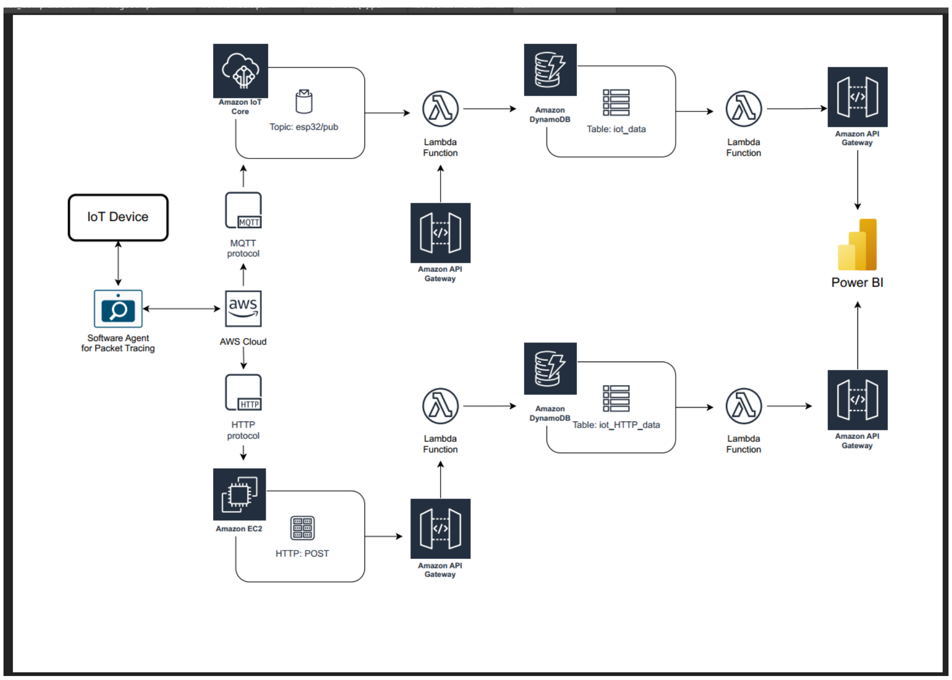


The I2C multiplexer system operates through the ESP32's dedicated SDCL pins, enabling simultaneous communication with multiple I2C sensors without address conflicts. This architecture supports complex sensor arrays and multi-parameter monitoring applications.

2.4 Cloud Integration and AWS Connectivity

The toolkit seamlessly integrates with Amazon Web Services (AWS) cloud infrastructure, providing scalable and secure data storage and processing capabilities. The system supports multiple AWS services including:

- **AWS IoT Core:** For device management and secure communication
- **AWS Lambda:** For serverless data processing and analytics
- **Amazon CloudWatch:** For monitoring and alerting
- **AWS DynamoDB:** For real-time database operations



Data transmission to AWS utilizes industry-standard security protocols including TLS 1.2 encryption and X.509 certificate-based authentication. The system automatically handles retry logic and data buffering to ensure reliable delivery even under challenging network conditions.

---

3. Microcontroller Specifications

3.1 ESP32 WROOM Development Kit Overview

At the heart of our IoT toolkit lies the ESP32 WROOM development kit, a powerful and versatile microcontroller platform that serves as the central processing unit for all data collection, manipulation, storage, and transmission operations. The ESP32 was specifically selected for this application due to its exceptional balance of processing power, connectivity options, and energy efficiency.

The ESP32's dual-core architecture provides the computational capability necessary for concurrent sensor data processing and network communications. This parallel processing capability ensures that sensor sampling rates remain consistent even during intensive network operations or data processing tasks.

3.2 Key Technical Advantages

Processing Capabilities

- **CPU Architecture:** Dual-core 32-bit LX6 microprocessor
- **Operating Frequency:** 240 MHz

Connectivity Features

- **Wi-Fi Standards:** 802.11 b/g/n (2.4 GHz band)
- **Bluetooth:** Classic Bluetooth and BLE 4.2
- **Network Security:** WPA/WPA2/WPA3 support

Sensor Compatibility

The ESP32's extensive GPIO and communication interface support makes it compatible with a vast array of sensors including:

- Temperature and humidity sensors
- Pressure and altitude sensors
- Motion and vibration sensors
- Light and UV sensors
- Gas and air quality sensors
- Proximity and distance sensors

3.3 Memory Architecture and Optimization

Our firmware implementation has been carefully optimized to maximize the ESP32's memory utilization while maintaining robust functionality:

Flash Memory Utilization

The ESP32 WROOM module provides 4MB of flash memory, with our current firmware utilizing approximately 89% of the available program storage space:

- **Total Available Flash:** 1,310,720 bytes (1.31 MB allocated for sketch)
- **Current Firmware Size:** 1,167,851 bytes
- **Remaining Capacity:** 142,869 bytes
- **Utilization Efficiency:** 89%

HTML Rendering Optimization

A significant optimization achievement in our firmware involves the strategic relocation of HTML content from ESP32 flash memory to SD card storage:

- **HTML in Flash Memory:** 1,167,851 bytes
- **HTML on SD Card:** 1,163,891 bytes
- **Memory Savings:** 3,960 bytes
- **Performance Impact:** Negligible rendering delay

This optimization preserves valuable flash memory space for core functionality while maintaining responsive web interface performance.

SRAM Management

The ESP32's SRAM is utilized for dynamic operations including buffers, arrays, and object instances:

- **Total SRAM Capacity:** 327,680 bytes (320 KB)
- **Current SRAM Usage:** 48,624 bytes (14.8%)
- **Available for Local Variables:** 279,056 bytes
- **Buffer Management:** Optimized for sensor data and network operations

3.4 Data Processing and JSON Payload Management

JSON Payload Structure and Optimization

Our system generates structured JSON payloads for efficient data transmission to AWS cloud services. The payload structure adapts dynamically based on active sensors and configuration settings:

Base Metadata Payload:

```json
{
  "deviceID": 24,
  "deviceName": "ESP32",
  "dataID": 21
}
```

- **Payload Size:** 48 bytes
- **Transmission Frequency:** Once per session
- **Content:** Device identification and session metadata

Full Sensor Data Payload:

```json
{
  "deviceID": 24,
  "deviceName": "ESP32",
  "dataID": 41,
  "Vibration A1": 125,
  "Tilt A2": 0,
  "Vibration A3": 0,
  "Vibration A4": 0
}
```

- **Payload Size:** 113 bytes
- **Transmission Frequency:** Based on configured interval
- **Content:** Complete sensor readings with timestamps

High-Performance Data Processing

JSON payload generation is optimized for minimal processing overhead:

- **Average Processing Time:** ~37 microseconds
- **Measurement Method:** ESP32 high-resolution timer
- **Performance Impact:** Negligible on overall system operation
- **Concurrent Processing:** Maintains sensor sampling rates

System Uptime and Reliability Considerations

The toolkit implements robust uptime monitoring with considerations for long-term deployment:

Standard Uptime Tracking:

- **Method:** millis() function
- **Precision:** Millisecond resolution
- **Maximum Duration:** 49 days before overflow
- **Overflow Behavior:** Automatic reset to zero

Extended Uptime Tracking:

- **Method:** esp_timer_get_time() function
- **Precision:** Microsecond resolution
- **Maximum Duration:** >200 years (64-bit integer)
- **Reliability:** No overflow concerns for practical deployments

---

4. How to Operate

4.1 Initial Setup and Power Management

Power System Activation

The initial setup process begins with proper power system management to ensure safe operation and optimal performance:

1. **Power Switch Operation**
    - Locate the main power switch on the IoT kit housing
    - Toggle the switch to the "ON" position to connect the internal battery to the system
    - Observe the power indicator LED which should illuminate to confirm power connectivity
    - Allow 10-15 seconds for system initialization and component stabilization
2. **Battery Charging Procedures IMPORTANT SAFETY WARNING:** Always turn OFF the IoT kit power switch before initiating the charging process to prevent potential damage to internal components.
    - Connect the USB Type-C charging cable to the designated charging port
    - The charging indicator will display a RED light during active charging
    - Monitor the charging indicator which will change to GREEN when charging is complete
    - Typical charging time ranges from 2-4 hours depending on current battery level
    - Disconnect the charging cable once the GREEN indicator is displayed

System Error Diagnostics and Initial Boot Sequence

Upon power activation, the IoT kit performs a comprehensive system check and initialization sequence:

1. **Error Detection and Display**

- The LCD screen will display any detected connectivity or hardware error
- Common error types include sensor communication failures, SD card mount errors, or Wi-Fi module initialization issues
- If errors are displayed, capture the LCD and contact your supplier for technical support

2. **Successful Boot Sequence**

- If no errors are detected, the system will automatically enter "Config Mode"
- The LCD screen will display "CONFIG MODE" along with the device's IP address
- This IP address is critical for accessing the web configuration interface
- The system is now ready for initial configuration

-

4.2 Network Configuration and Wi-Fi Setup

Connecting to the IoT Kit's Hosted Network

The IoT kit creates its own Wi-Fi access point for initial configuration:

1. **Network Connection Steps**
    - On your smartphone, tablet, or computer, access Wi-Fi settings
    - Look for the network named "nIoTo" in available networks
    - Select "nIoTo" and enter the password: **12345678**
    - Wait for successful connection confirmation
    - Ensure your device shows "Connected" status to the nIoTo network
2. **Accessing the Configuration Interface**
    - Open any web browser on the connected device
    - Navigate to the IP address displayed on the IoT kit's LCD screen
    - The browser should load the configuration webpage
    - If the page doesn't load, verify the IP address and network connection

Configuration Form Completion

The web-based configuration form allows you to customize the IoT kit's operation according to your specific requirements:

1. **Wi-Fi Credentials Configuration**
    - **Network Name (SSID):** Enter your target Wi-Fi network name
    - **Password:** Input the Wi-Fi network password
    - **Security Type:** System auto-detects (WPA2/WPA3)
    - **Connection Test:** Optional test connection button to verify credentials
2. **Data Transmission Settings**
    - **Upload Interval:** Specify the delay between data transmissions to AWS cloud
    - **Recommended Range:** 30 seconds to 10 minutes depending on application
    - **Battery Optimization:** Longer intervals conserve battery life
    - **Real-time Requirements:** Shorter intervals for time-critical monitoring
3. **Sensor Configuration Analog Sensor Setup (Group A: A1-A4):**

- Select sensor types from dropdown menus for each port
- Configure measurement ranges and calibration factors
- Enable/disable individual sensor channels as needed

I2C Sensor Setup (Group B: B1-B4):

- Specify I2C sensor models and addresses
- Configure measurement parameters and units
- Set communication timing and retry parameters
- Enable automatic sensor discovery if supported

4. **Form Submission and Confirmation**
    - Review all configuration settings for accuracy
    - Click the "Submit" button to save configuration
    - The system will display "Successful credentials saved" confirmation page

      Configuration is automatically saved to both ESP32 flash memory and SD card

4.3 Operational Mode Management

Dual Mode Operation Control

The IoT kit supports two distinct operational modes, each serving different use cases:

1. **Config Mode (Offline Operation)**
    - **Purpose:** Local data monitoring and configuration access
    - **Network:** Creates "nIoTo" Wi-Fi access point
    - **Data Storage:** All data saved to SD card
    - **Web Access:** Full configuration interface available
    - **Power Consumption:** Optimized for extended battery operation

-

2. **Active Mode (Online Operation)**
   - **Purpose:** Cloud data transmission and remote monitoring
   - **Network:** Connects to configured Wi-Fi network
   - **Data Storage:** Dual storage (SD card + AWS cloud)
   - **Cloud Services:** Full AWS integration active
   - **Power Consumption:** Higher due to constant connectivity

Mode Switching Mechanism

The system provides intuitive mode switching through a physical push button interface:

1. **Mode Switching Procedure**
   - Locate the push button on the IoT kit housing
   - Perform a **double-tap** action (two quick presses within 1 second)
   - Observe the LCD screen for mode change confirmation
   - Allow 5-10 seconds for mode transition to complete

2. **Mode Transition Indicators**
   - **Audio Feedback:** Brief beep confirms button recognition
   - **LCD Update:** Mode display changes immediately
   - **LED Indicators:** Status LEDs reflect current operational mode
   - **Network Activity:** Wi-Fi connection establishment in Active mode

4.4 Data Monitoring and Access

Local Data Visualization (Config Mode)

When operating in Config Mode, users can access comprehensive data visualization through the local web interface:

1. **Accessing Local Interface**
   - Ensure connection to "nIoTo" network
   - Navigate to the IP address shown on LCD
   - Browse to the data visualization section
   - Real-time graphs update automatically every few seconds

2. **Graph Interaction Features**
    - **Graph Display Features:** Real-time data visualization with automatic updates
    - **Time Range Selection:** Choose from preset ranges (1 hour, 6 hours, 24 hours)
    - **Data Export Options:** Download raw data files for further analysis

Cloud Data Access (Active Mode)

In Active Mode, data is transmitted to AWS cloud services for advanced analytics and remote access:

1. **AWS Dashboard Access**
    - Data becomes available in your configured AWS account
    - Access through AWS IoT Console or custom dashboards
    - Real-time and historical data analysis capabilities
    - Integration with AWS analytics and machine learning services

2. **Data Redundancy and Backup**
    - All cloud-transmitted data is simultaneously stored on SD card

      Local backup ensures data preservation during connectivity issues
    - Automatic retry mechanism for failed cloud transmissions
    - Data integrity verification for both local and cloud storage

---

5. Technical Specifications

5.1 Hardware Specifications

Main Processing Unit

-

- **Microcontroller:** ESP32 WROOM Development Kit
- **CPU Architecture:** Dual-core 32-bit LX6
- **Operating Frequency:** 240 MHz (adjustable)
- **Flash Memory:** 4 MB
- **SRAM:** 320 KB
- **PSRAM:** Not applicable
- **Operating Voltage:** 3.3V
- **Input Voltage Range:** 3.0V - 3.6V

Power Management System

- **Battery Type:** Rechargeable Li-ion
- **Battery Capacity:** 3000 mAh
- **Charging Interface:** USB Type-C
- **Charging Current:** 1A maximum
- **Power Consumption (Active Mode):** 150-200 mA
- **Power Consumption (Config Mode):** 80-120 mA
- **Sleep Mode Current:** <10 µA
- **Operating Time:** 15-25 hours (depending on mode and sensor configuration)

Communication Interfaces

- **Wi-Fi Standard:** 802.11 b/g/n
- **Frequency Range:** 2.4 GHz
- **Wi-Fi Security:** WPA/WPA2/WPA3
- **Bluetooth:** Classic + BLE 4.2
- **Antenna:** Integrated PCB antenna
- **Range:** Up to 100 meters (line of sight)

Input/Output Capabilities

- **Analog Inputs:** 4 channels (A1-A4)
- **Digital I/O:** 8 channels (configurable)
- **I2C Interfaces:** 4 channels (B1-B4) via multiplexer
- **SPI Interface:** 1 channel (for SD card)
- **UART Interface:** 2 channels (1 for programming, 1 available)
- **PWM Channels:** 16 channels available

Physical Dimensions

- **Main Unit Dimensions:** 120mm × 80mm × 35mm
- **Weight:** 250 grams (including battery)
- **Enclosure Material:** ABS plastic with UV resistance
- **Protection Rating:** IP54 (dust and splash resistant)
- **Mounting Options:** Wall mount, DIN rail, magnetic base

5.3 Data Storage

SD Card Specifications

- **Card Type:** MicroSD/SDHC/SDXC
- **Supported Formats:** FAT16, FAT32, exFAT
- **Maximum Capacity:** 128 GB
- **Speed Class:** Class 10 or higher recommended
- **Data Logging Format:** CSV, JSON
- **File Organization:** Date-based folders with hourly files

5.4 Sensor Interface Specifications

Analog Input Specifications (Group A)

- **Input Voltage Range:** 0V to 3.3V

- **Resolution:** 12-bit (4096 levels)
- **Accuracy:** ±1% (with calibration)
- **Sampling Rate:** Fixed rate optimized for system performance
- **Input Impedance:** >100 kΩ
- **Common Mode Rejection:** >60 dB
- **Temperature Drift:** <50 ppm/°C

I2C Interface Specifications (Group B)

- **Operating Voltage:** 3.3V or 5V (selectable)
- **Clock Speed:** Fixed at 500 kHz for optimal performance
- **Address Range:** 0x08 to 0x77 (7-bit addressing)
- **Maximum Cable Length:** 5 meters (with proper cables)
- **Pull-up Resistors:** 4.7 kΩ (built-in)
- **Multiplexer Channels:** 4 independent channels
- **Hot-swap Capability:** Supported with proper procedures

---

6. Troubleshooting Guide

6.1 Power and Battery Issues

Problem: Device Won't Power On

Symptoms: No LED indicators, blank LCD screen, no response to button presses **Troubleshooting Steps:**

1. **Battery Check:**
    - Ensure the power switch is in the ON position
    - Check battery charge level by connecting USB-C charger
    - If RED charging light doesn't appear, battery may be completely depleted
    - Allow 30 minutes of charging before attempting to power on
2. **Connection Verification:**
    - Inspect power switch for proper connection
    - Check for loose internal connections (contact supplier if suspected)
    -

Verify USB-C charging port is clean and free of debris

Problem: Short Battery Life

Symptoms: Device powers off unexpectedly, frequent charging required **Possible**

Causes and Solutions:

1. **High Power Consumption Mode:**
    - Switch to Config Mode for extended battery operation
    - Reduce data transmission frequency in Active Mode
    - Disable unused sensors to minimize power draw
2. **Battery Degradation:**
    - Check charging behavior (GREEN light should appear when full)
    - If battery doesn't reach full charge, replacement may be needed
    - Contact supplier for battery replacement procedure
3. **Environmental Factors:**
    - Extreme temperatures reduce battery performance
    - Ensure operation within specified temperature range (-10°C to +60°C)

6.2 Connectivity Issues

Problem: Cannot Connect to "nIoTo" Network

Symptoms: Network not visible in Wi-Fi settings, connection fails **Troubleshooting Steps:**

1. **Network Visibility:**
    - Ensure device is in Config Mode (check LCD display)
    - Move closer to the IoT kit (within 10 meters)
    - Restart your connecting device's Wi-Fi
    - Reset the IoT kit by power cycling
2. **Password Issues:**
    - Verify password is entered exactly: **12345678**
    -
    -

Check for autocorrect interference on mobile devices Try connecting from a different device to isolate the problem

3. **Network Conflicts:**
   - Check for other devices using the same network name
   - Restart the IoT kit to refresh the Wi-Fi access point
   - Wait 2-3 minutes after mode changes for network stabilization

Problem: Cannot Access Web Interface

Symptoms: Browser shows "page not found" or connection timeout **Resolution**

Steps:

1. **IP Address Verification:**
   - Double-check the IP address shown on LCD screen
   - Ensure you're connected to the "nIoTo" network
   - Try accessing from a different browser
2. **Network Configuration:**
   - Disable mobile data on smartphones to force Wi-Fi usage
   - Clear browser cache and cookies Try
   - incognito/private browsing mode
3. **Firewall and Security:**
   - Temporarily disable firewall on connecting device
   - Check if corporate networks block local IP access
   - Verify no VPN is interfering with local connections

6.3 Sensor and Data Issues

Problem: Sensor Readings Show Zero or Invalid Values

Symptoms: All sensor channels show 0, extremely high values, or no data

Diagnostic Approach:

1. **Physical Connections:**

- Verify all sensor connections are secure
- Check pin alignment for proper insertion
- Inspect cables for damage or wear

2. **Sensor Configuration:**
    - Review sensor selections in configuration form
    - Ensure sensor types match physical connections
    - Verify I2C addresses for Group B sensors
3. **Power Supply Verification:**
    - Check if sensors require 5V instead of 3.3V
    - Verify power selection jumpers if present
    - Test sensors individually to isolate issues

Problem: Data Not Uploading to AWS Cloud

Symptoms: Local data available but no cloud data, upload errors on LCD **Resolution**

Process:

1. **Network Connectivity:**
    - Verify Wi-Fi credentials are correct in configuration
    - Check internet connectivity of the target Wi-Fi network
    - Ensure network allows HTTPS outbound connections
2. **AWS Configuration:**
    - Verify AWS account credentials and permissions
    - Check AWS IoT Core device certificates
    - Review AWS service quotas and limits
3. **Data Format Issues:**
    - Monitor JSON payload formatting in serial output
    - Check for special characters in sensor data
    - Verify timestamp formatting compatibility

6.4 SD Card and Storage Issues

Problem: SD Card Not Recognized

Symptoms: "SD Card Error" on LCD, no local data storage **Troubleshooting:**

1. **Card Compatibility:**
    - Use Class 10 or higher MicroSD card
    - Format as FAT32 for best compatibility
    - Maximum tested capacity: 128 GB
2. **Physical Issues:**
    - Ensure card is fully inserted
    - Check for dust or corrosion on contacts
    - Try a different known-good SD card
3. **File System Problems:**
    - Reformat the SD card on a computer
    - Check for file system errors using disk utility
    - Avoid removing card while device is powered on

7. Credits

7.1 Development Team

This IoT Toolkit represents the collaborative effort of a dedicated team of engineers and researchers committed to advancing IoT technology and cloud integration solutions.

Firmware Development

Lead Developer: Basil Khowaja **Role:**
Senior Firmware Engineer
Contributions:

- ESP32 firmware architecture and development
- AWS cloud integration protocols
- Sensor interface optimization
- Web interface programming
- Data processing algorithms
- Real-time operating system implementation
- Memory management optimization
- JSON payload structure design

Basil's expertise in embedded systems programming and cloud technologies has been instrumental in creating the robust and efficient firmware that powers this IoT toolkit. His innovative approach to memory optimization and dual-mode operation has resulted in a system that maximizes functionality while maintaining optimal performance characteristics.

Hardware Design and Development

Lead Designer: Vishal Raj Sundrani
Role: Senior Hardware Engineer **Contributions:**

- PCB design and layout optimization
- Power management system architecture
- Sensor interface circuit design
- Enclosure design and mechanical engineering
- Component selection and sourcing

Vishal's comprehensive understanding of hardware design principles and manufacturing processes has resulted in a toolkit that combines reliability, functionality, and cost-effectiveness. His attention to detail in power management and thermal design ensures optimal performance across diverse operating conditions.

Academic Supervision and Project Guidance

Project Supervisor: Sir Tariq Mumtaz

Position: Faculty Member, Electrical and Computer Engineering Department

Institution: Habib University **Academic**

Contributions:

- Project conceptualization and scope definition
- Technical review and validation
- Industry standards compliance guidance
- Research methodology and documentation
- Student mentorship and career development
- Academic integrity and quality assurance