Shardeum IoT Relay Control Project

□ World's First IoT Relay Control System on Shardeum **Secure • Transparent • Real-time ESP32 Automation via Blockchain** --- **Author:** BALAJI N **Date:** August 14, 2025 **Status:** First Shardeum IoT Chain Project

□ Table of Contents

1. □ Project Overview

This groundbreaking project seamlessly integrates **IoT hardware (ESP32 + Relay Module)** with **Shardeum blockchain smart contracts**, enabling **secure, decentralized control** of physical devices. Every command sent to the relay is recorded on-chain, ensuring complete transparency, immutability, and reliability.

□ Innovation Highlights

- **First of its kind**: Pioneer Shardeum IoT integration project
- **Real-world impact**: Demonstrates blockchain potential in device automation
- **Scalable foundation**: Blueprint for future industrial IoT applications
- **Zero trust architecture**: No central servers or single points of failure

2. ✨ Key Features

2.1 ▯ Blockchain-Powered IoT

- **Smart Contract Integration**: Deployed on Shardeum for relay state management - **Immutable Audit Trail**: Every relay toggle permanently recorded on blockchain - **Authorized Access Control**: Only designated wallet addresses can control the relay - **Decentralized Architecture**: Eliminates single points of failure

2.2 ⚡ Real-Time Device Control

- **Continuous Monitoring**: ESP32 actively polls blockchain for state changes - **Instant Response**: Relay switches ON/OFF automatically based on blockchain commands - **Versatile Applications**: Perfect for home automation, industrial processes, or smart farms - **Low Latency**: Optimized polling intervals for near real-time control

2.3 ▯ Security & Transparency

- **Cryptographic Security**: All transactions signed with private keys - **No Credential Storage**: Zero centralized servers storing sensitive data - **Tamper-Proof Commands**: Blockchain immutability prevents unauthorized modifications - **Public Verifiability**: All actions transparently viewable on blockchain explorer

2.4 ▯ Cross-Platform Integration

- **Python Backend**: Comprehensive scripts for contract deployment and interaction - **Web Interface**: Browser-based relay control with intuitive HTML + JavaScript UI - **Arduino Firmware**: Optimized ESP32 code for reliable hardware polling - **Multi-Platform Support**: Compatible with Windows, macOS, and Linux

3. ▯ Technical Architecture

▯ System Components

Component	Technology	Purpose
Smart Contract	Solidity	Relay state management and access control
Blockchain	Shardeum (EVM-compatible)	Decentralized transaction processing
IoT Device	ESP32 + Relay Module	Physical hardware control interface
Backend	Python (Web3.py)	Contract interaction and deployment
Firmware	Arduino C++	ESP32 blockchain polling and relay control

Component	Technology	Purpose
Frontend	HTML/CSS/JavaScript	User interface for relay control

System Workflow

```
graph TD
    A[User/Web Interface] -->|Send Transaction| B[Shardeum Blockchain]
    C[Python Scripts] -->|Deploy/Interact| B
    B -->|Smart Contract| D[Relay State Storage]
    E[ESP32 Device] -->|Poll State| B
    B -->|Return State| E
    E -->|Control Signal| F[Physical Relay]
    F -->|Device ON/OFF| G[Connected Appliance]
```

Detailed Architecture Flow

1. **Contract Deployment**: Smart contract deployed to Shardeum testnet/mainnet
2. **User Interaction**: Commands sent via Python scripts or web interface
3. **Blockchain Processing**: Transactions validated and state updated on-chain
4. **Device Synchronization**: ESP32 polls blockchain at regular intervals
5. **Physical Control**: Relay toggles based on current blockchain state
6. **Feedback Loop**: System maintains continuous synchronization

4. Installation & Usage

4.1 Repository Setup

```
# Clone the project repository
git clone https://github.com/your-username/shardeum-iot-relay-control.git
cd shardeum-iot-relay-control
```

4.2 Python Dependencies

```
# Install required Python packages
pip install -r requirements.txt

# Key dependencies include:
# - web3.py (Blockchain interaction)
# - python-dotenv (Environment management)
# - requests (HTTP communication)
```

4.3 ⚙ Environment Configuration

Create a `.env` file in the project root:

```
# Shardeum Network Configuration
SHARDEUM_RPC_URL=https://dapps.shardeum.org/
PRIVATE_KEY=your_wallet_private_key_here
WALLET_ADDRESS=your_wallet_address_here

# Contract Configuration
CONTRACT_ADDRESS=deployed_contract_address_here
GAS_LIMIT=100000
GAS_PRICE=1000000000

# ESP32 Configuration
DEVICE_ID=ESP32_RELAY_01
POLL_INTERVAL=5000
```

4.4 ▯ Smart Contract Deployment

```
# Deploy the relay control contract
python scripts/deploy_contract.py

# Verify deployment
python scripts/verify_contract.py
```

4.5 ▯ Contract Interaction

```
# Turn relay ON
python scripts/relay_on.py

# Turn relay OFF
python scripts/relay_off.py

# Check current relay state
python scripts/check_state.py

# View transaction history
python scripts/transaction_history.py
```

4.6 ▯ ESP32 Hardware Setup

Hardware Requirements:

- ESP32 Development Board
- 5V Relay Module
- Jumper Wires
- Power Supply (if needed)

Wiring Diagram:

```
ESP32 Pin     →    Relay Module Pin
GPIO 2        →    IN (Control)
3.3V          →    VCC
GND           →    GND
```

Firmware Upload:

1. Open `esp32_relay.ino` in Arduino IDE
2. Update WiFi credentials:
   ```
   const char* ssid = "your_wifi_ssid";
   const char* password = "your_wifi_password";
   ```
3. Configure blockchain settings:
   ```
   const char* rpcUrl = "https://dapps.shardeum.org/";
   const char* contractAddress = "your_contract_address";
   ```
4. Upload to ESP32 board
5. Monitor serial output for connection status

5. □ Project Impact

□ Immediate Benefits

□ Decentralized Automation - Eliminates single points of failure - Reduces dependency on centralized services - Increases system reliability and uptime - Provides true peer-to-peer device control

□ Enhanced Security - Cryptographically secured commands - Immutable audit trails - Transparent operation logs - Reduced attack surface

</div>

Long-term Vision

- **Industrial IoT Revolution**: Foundation for large-scale blockchain-based automation
- **Smart City Integration**: Scalable infrastructure for municipal IoT networks
- **Energy Management**: Decentralized control of power grids and renewable systems
- **Supply Chain Automation**: Transparent and secure logistics control systems

Market Potential

- **Growing IoT Market**: $1.1 trillion projected by 2028
- **Blockchain Adoption**: Increasing enterprise blockchain implementations
- **Security Concerns**: Rising demand for decentralized security solutions
- **Automation Trends**: Industrial 4.0 driving automation demand

6. Future Enhancements

Multi-Device Support

- **Parallel Control**: Manage multiple relays simultaneously
- **Device Groups**: Organize devices into logical groups
- **Batch Operations**: Execute commands across multiple devices
- **Load Balancing**: Distribute commands across device networks

Cross-Chain Integration

- **Multi-Blockchain Support**: Extend beyond Shardeum to Ethereum, Solana, Polygon
- **Chain Bridging**: Enable cross-chain device control
- **Interoperability**: Seamless operation across different blockchain networks
- **Cost Optimization**: Automatic selection of most cost-effective chains

Advanced Dashboard

- **Real-time Monitoring**: Live device status and performance metrics
- **Historical Analytics**: Detailed usage patterns and trends
- **Alert Systems**: Automated notifications for device issues
- **Mobile App**: Native iOS and Android applications

AI Integration

- **Predictive Control**: AI-driven device automation based on usage patterns
- **Anomaly Detection**: Automatic identification of unusual device behavior
- **Energy Optimization**: AI-powered energy consumption optimization
- **Maintenance Scheduling**: Predictive maintenance alerts and scheduling

Enterprise Features

- **Role-Based Access**: Granular permission management
- **API Gateway**: RESTful APIs for enterprise integration
- **Compliance Reporting**: Automated regulatory compliance reports
- **Audit Trails**: Comprehensive audit logging and reporting

7. References

Official Documentation

- Shardeum Developer Documentation
- Web3.py Complete Guide
- ESP32 Arduino Framework

Technical Resources

- Solidity Programming Language
- Ethereum Virtual Machine Specification
- IoT Security Best Practices

Research Papers

- "Blockchain in IoT: A Survey" - IEEE Internet of Things Journal
- "Decentralized IoT Security Architecture" - ACM Computing Surveys
- "Smart Contracts for IoT Applications" - Blockchain Research Institute

Ready to Get Started? This project represents the cutting edge of blockchain-IoT integration. Join us in building the future of decentralized device automation! ** Connect With Us:** - GitHub: [Repository Link] - Email: balaji@example.com - Twitter:

Quick Start Checklist

- [ ] Clone repository
- [ ] Install Python dependencies
- [ ] Configure environment variables
- [ ] Deploy smart contract
- [ ] Set up ESP32 hardware
- [ ] Upload firmware
- [ ] Test relay control
- [ ] Monitor blockchain transactions

Need Help? Check our troubleshooting guide or open an issue on GitHub!