PROJECT STAGE - I REPORT

A report submitted in partial fulfilment of the requirements for the

Award of Degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

Anushka Pote (5)

Aditya Suryawanshi (12)

Advait Jog (13)

Anshu Jamare (17)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT

(An Autonomous Institute)
Approved by AICTE, affiliated to RTMNU, City Nagpur, accredited by NBA (AICTE)
& NAAC (UGC), Recognized by UGC u/s 2(f) & 12(B), New Delhi.
NAGPUR, MAHARASHTRA

SHRI RAMDEOBABA COLLEGE OF ENGINEERING AND MANAGEMENT, NAGPUR

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project report titled

“Self-Recovering Network Infrastructure Using Automated Router Diagnosis”

is a bonafide work of

1. Anushka Pote (Roll No.-05, Sec-A)
2. Aditya Suryawanshi (Roll No.-12, Sec-A)
3. Advait Jog (Roll No.-13, Sec-A)
4. Anshu Jamare (Roll No.-17, Sec-A)

Submitted as a part of term work in 7th semester in Project Stage – 1 (ECP-455) during the academic session 2025-2026.

Date: 10th November 2025

Place: RCOEM, Nagpur

Dr. P.R. Selokar

(Project Guide)

CONTENTS

Sr. No.	Title	Pg. No.
1	Abstract	1
2	Introduction	1
3	System Architecture and Component Rationale	2
4	Hardware Implementation and Safety Protocols	4
5	Software Logic and Robustness	4
6	Project Outcomes and Future Scope	5
7	References	9

1. Abstract:

This project, "Self-Recovering Network Infrastructure using Automated Router Diagnosis," tackles the important problem of network downtime caused by ineffective manual router intervention. The main goal is to develop a reliable, cost-effective embedded system that constantly checks network health, identifies critical failures like router hangs or ISP drops, and automatically restarts (reboots) to restore connectivity[1]. This enhances system uptime and removes the need for human involvement. The system is designed with a distributed microcontroller setup, using the single-plug method for commercial use.

2. Introduction:

1) Problem Statement:

The industry standard for fixing consumer-level network device failures involves a manual power-cycle, which is time-consuming and often occurs outside business hours. This latency between failure detection and manual recovery leads to significant service disruption and productivity loss. Furthermore, relying on an Operating System (OS) based solution like Raspberry Pi for this critical task introduces its own failure points (OS corruption, long boot time). Our project proposes an embedded, hardware-based solution focused on deterministic, sub-minute recovery and advanced diagnostic accuracy to counter these challenges.

2) Project Objective:

The primary objectives that determine the success of this project are:

- Autonomous Fail-Safe Diagnosis:
  Implement a robust multi-target ping mechanism utilising the global Anycast Network architecture. This method reliably verifies authentic network failures across various global DNS servers, thereby reducing false positives and confirming the necessity of the reboot.
- Reliable Power Actuation:
  The integration and control of the high-current 9V DC router power line via a 5V DC relay necessitates the preservation of galvanic isolation between the low-voltage control circuitry operating at 3.3V and the high-current switched circuit at 9V.

- Advanced self-diagnosis and feedback mechanisms are proposed, utilising closed-loop feedback systems (ACK) and visual status indicators (Red/Green LEDs) to verify the functionality of the control unit and confirm the receipt of reboot commands.
- Power Efficiency and Commercial Scalability:
  The objective is to design a power architecture that leverages the router's existing 9V supply through an LM2596 Buck Converter to power all controllers, thereby creating a streamlined Single-Plug Solution appropriate for market application.

3. System Architecture and Component Rationale:

1) Component Rationale:

Components	Role	Technical Rationale
ESP-32	Network Monitor	Chosen for its integrated Wi-Fi capabilities and powerful Tensilica LX6 processor. Its primary function is to handle the resource - intensive, asynchronous task of maintaining a network connection and executing time-sensitive Ping tests, thereby ensuring fast detection[2].
Raspberry Pi Pico-H	Control Unit	Chosen for its RP2040 dual-core architecture, providing deterministic, real-time control over the relay. The MicroPython environment allows for rapid prototyping and low-level GPIO control essential for reliable switching.
LM2596 Buck Converter	Power Manager	This DC-DC SMPS module is essential for achieving the Single-Plug Solution. It efficiently converts the 9V DC input (tapped from the router line) into a clean, stable 5V DC output with excellent line and load regulation for the MCUs.
5V 1-channel optocoupler Relay module	Actuator	Provides high-current switching capacity up to 10A for the 9V router line. It uses an internal optocoupler for basic isolation, ensuring the microcontroller's 3.3V logic is protected from the 9V circuit.

Status LEDs	Indicator	Red and Green LEDs provide immediate, non-invasive visual feedback to the end-user regarding the current state of the network and the system's ongoing actions.

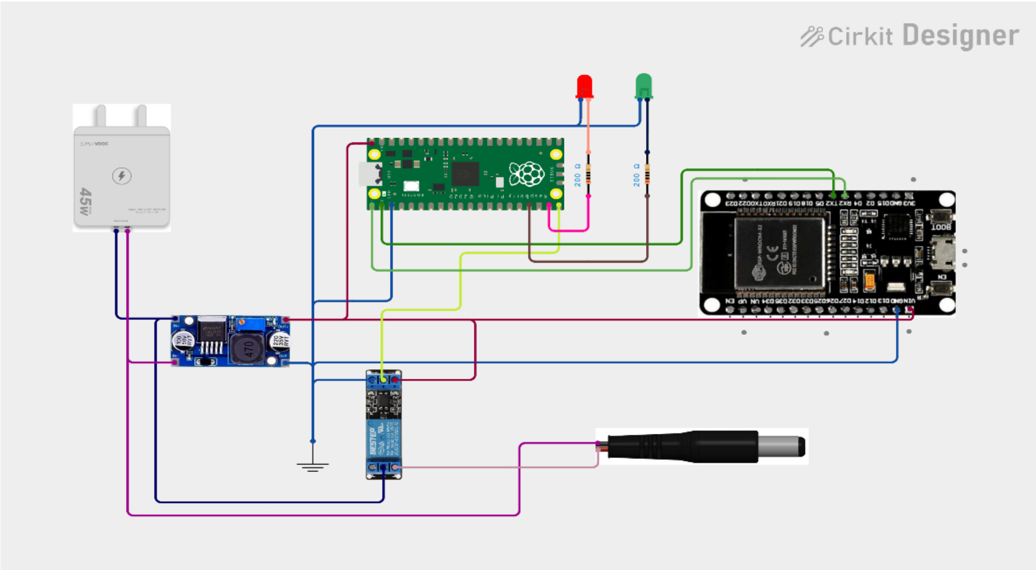

Figure 1: Circuit Diagram

2) Power Management: A Single Plug Approach

The design addresses the commercial challenge of the Dual-Plug problem by utilizing the LM2596 Buck Converter. The 9V DC router power line is T-Tapped (Parallel connection) before the relay switch.

- Tapping: The 9V power is tapped to provide input to the Buck Converter.
- Conversion: The Buck Converter steps the voltage down to a precisely calibrated 5.02V DC.
- Distribution: This stable 5V output is then distributed across the VBUS/VIN pins of both ESP32 and RPi Pico and the Relay Coil (VCC).
- Safety: This method ensures that the controllers remain powered and operational even when the main 9V line to the router is physically cut by the relay during the reboot cycle.

4. Hardware Implementation and Safety Protocols:

Wiring and Grounding Protocols

1. Common Ground Reference:
   All components ESP32, Pico, Buck, Relay are connected to a single GND Rail Buck OUT-, establishing a stable 0V reference, which is critical for accurate UART communication.
2. Actuation Wiring (COM/NC Logic):
   Based on the requirement for the router to be ON by default, the switching terminals were wired for Default ON state - The 9V DC source is connected to COM, and the Router Plug is connected to NC (Normally Closed). This necessitates an inverted software logic to break the connection during reboot.

UART Communication Setup

Communication is established via a simple 3-wire link TX, RX, GND:

- Command Link: ESP32 GPIO 17 (TX) connects to Pico's GP1(RX).
- Feedback Link: Pico's GP0 (TX) connects to ESP32's GPIO 16 (RX).

5. Software Logic and Robustness:

The software is designed for reliability, minimizing CPU cycles and implementing strict failure criteria.

The Multi-Target Ping Algorithm

The ESP32 runs a precise ESPPing algorithm that simultaneously targets three different global endpoints (e.g., 8.8.8.8, 1.1.1.1) along with the local router gateway IP[4] [5].

- Logic: The network is only declared DOWN if the logical AND of all three ping tests fails. This utilizes the stability of the Anycast Network of global DNS providers to filter out transient packet loss or localized ISP issues, ensuring the reboot trigger is truly a last resort.
- Threshold: A failure must be confirmed across 3 consecutive checks MAX_FAIL_COUNT = 3 before the reboot command is issued.

Critical Control and Inverted Logic (RPi Pico)
The Pico executes the critical reboot sequence using the logic dictated by the hardware's NC wiring[3]:

1. Initial State: The function turn_router_on() sets the RELAY PIN to HIGH 1, which de-energizes the coil, maintaining the NC link and keeping the router ON.
2. Action: The function turn_router_off() sets the RELAY_PIN to LOW 0, which energizes the coil, breaking the NC link and cutting power to the router.
3. Downtime Duration: The REBOOT_DURATION is set to 10 seconds to ensure a guaranteed, clean power-cycle and visual confirmation for the user.

Feedback Mechanism (ACK)
The ACK protocol serves as a Control Unit Health Check. Upon receiving the 'R' command, the Pico immediately sends 'A' back.

- Success: If the ESP32 receives 'A' within 500ms, it confirms the Pico is functional and the command was processed.
- Fault Detection: If the ESP32 times out, it logs a Pico Fault, resets its own counter, and prevents futile command sending, thereby safeguarding its own operational stability.

6. Project Outcomes and Future Scope:

Outcomes Achieved (Stage 1 - Prototype)

- Successful Full Cycle:
  Autonomous network failure detection and power-cycle demonstrated using the inverted COM/NC logic.
- Reliability:
  Multi-Target Ping and ACK feedback loops are successfully implemented and verified, proving robust communication integrity.
- Power Solution:
  The commercial viability challenge was resolved via LM2596 integration, enabling operation via a single router adapter.

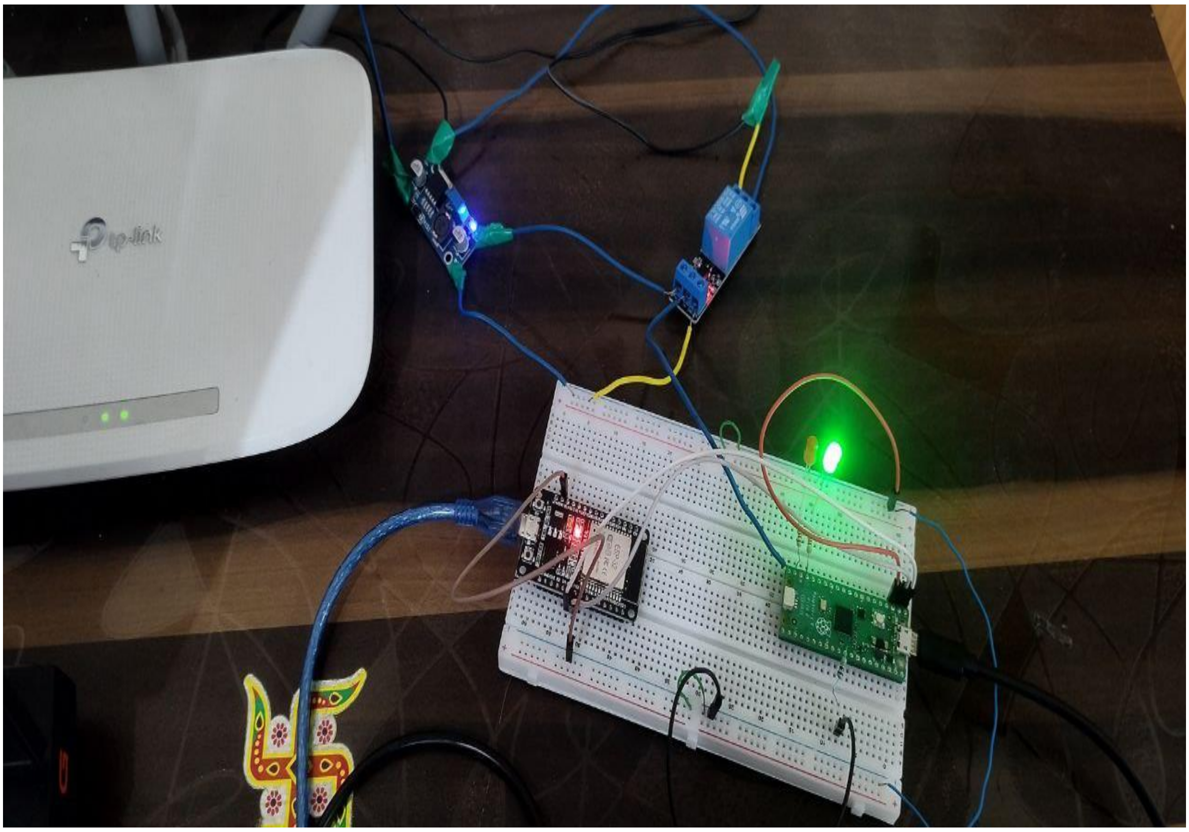

Figure 2: Router OK

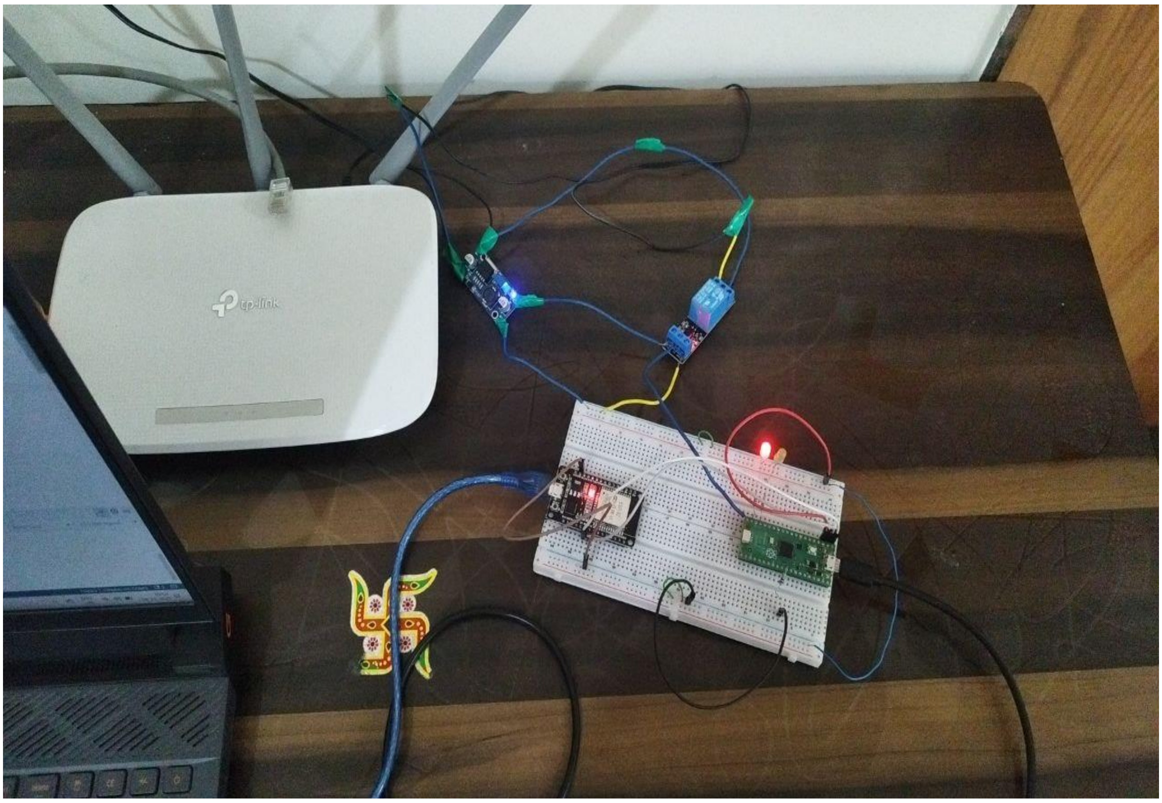

Figure 3: Reboot Process

```
Output    Serial Monitor ×
Message (Enter to send message to 'LilyGo T-Display' on 'COM5')
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
```

Figure 4: Network OK

```
Output    Serial Monitor ×
Message (Enter to send message to 'LilyGo T-Display' on 'COM5')
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
Ping SUCCESS. Network OK.
All Pings FAILED. Count: 1
All Pings FAILED. Count: 2
All Pings FAILED. Count: 3
--- CRITICAL FAILURE DETECTED ---
Command Sent: R
ACK received: Pico alive, rebooting router!
```

Figure 5: Ping Failure

```
Shell
MPY: soft reboot
--- RPi Pico Control System READY ---

--- COMMAND RECEIVED: REBOOT (R) ---
Router Power OFF for 10 seconds...
Reboot successful. Monitoring resumed.
```

Figure 6: Router power down using Pico and Relay

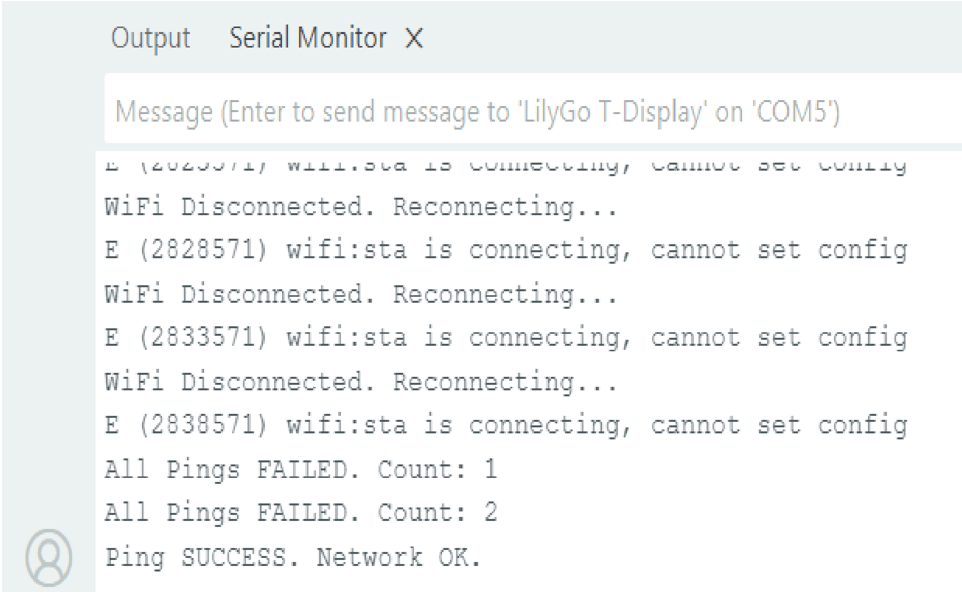

Figure 7: Reconnection to gain normalcy

Future Enhancements (Stage 2 - Commercialization)

The project foundation is ready for the following advanced commercial features:

1. Remote Data Analytics and Predictive Maintenance:
   Integrate the ESP32 with Firebase to Ping Latency and Reboot Events with timestamps. This data will be used to generate Predictive Maintenance Reports (e.g., in Power BI), flagging recurring ISP instability before a complete failure occurs.

2. User Interface and Enclosure:
   Design and 3D print a robust enclosure with integrated DC Barrel Jacks for a true Plug-and-Play solution, enhancing aesthetics and durability.

3. Web Control:
   Develop a secure, mobile-friendly web interface hosted by the ESP32 allowing users to check the network status and manually initiate a remote reboot command, providing essential user control over the autonomous system.

7. References:

1. “Reset your router the modern way” Jenny List, Hackaday, 2019.
2. “Automatic Router and Server Reset Using ESP8266,” Instructables, 2017.
3. P. Zabala et al., “Development of Programmable Relay Switch Using Microcontroller,” Asian Journal of Research in Science, 2018.
4. “ESP-32 Wi-Fi Network Smart Monitoring System” Aysar Thamer Naser Tuaimah, SRJECS journal, 2023
5. “A Generic IoT Quantum Safe - Watchdog Timer Protocol” ACM Digital Library, 2023