

# **Network-Controlled LED System Using DE1-SoC**

## **Overview**

This project aims to establish a network-based control system for the LEDs on a DE1-SoC board. It leverages the functionalities of both the built-in Field Programmable Gate Array (FPGA) and the ARM processor (HPS) on the DE1-SoC. This project demonstrates a fundamental concept in computer networks, remote device control. It showcases how network communication protocols and hardware components can work together to achieve a specific action. The project highlights real-world applications like Networked Automation Systems, where similar principles are used in industrial settings where machines or devices are controlled remotely via network commands. This project combines software and hardware to achieve remote control functionalities of LEDs on the DE1-SoC board, providing hands-on experience with network programming, hardware-software integration, and practical applications of these concepts.

## **Introduction**

The DE1-SoC board, developed by Terasic, integrates an Altera Cyclone V FPGA and a dual-core ARM Cortex-A9 processor, making it a versatile platform for a wide range of applications, from embedded systems to advanced computing tasks. In this project, the DE1-SoC board is used to demonstrate the control of onboard LEDs through network commands sent over TCP/IP.

Remote control systems are crucial in various industrial and automation applications. This project explores the implementation of a network-based control system for LEDs on the DE1-SoC board, showcasing the practical application of network communication for remotely controlling hardware devices. By sending commands from a client computer to the DE1-SoC board, we can toggle the state of the LEDs, providing a clear example of remote hardware control.

## **Design**

The network-controlled LED system utilizes a client-server architecture for communication. The system architecture includes:

- **Client Application:** This application, running on a separate computer, allows users to send control commands for the LEDs. It establishes a TCP/IP connection with the server program on the DE1-SoC board, transmits commands over the network, and receives confirmation messages from the server when a command is accepted.
- **HPS Server Program:** This program runs on the DE1-SoC's ARM processor, it acts as the server. It listens for incoming connections on a specific port, parses received commands from the client and interacts with the FPGA logic to control the LEDs. The server program also sends confirmation messages back to the client for successful command execution.

The block diagram below shows how commands sent over the network from the computer are received by the server program on the HPS. The program parses the commands and sends control signals to the FPGA logic, which activates or deactivates LEDs on the board.



Figure 1. High-Level Design

The system leverages the TCP/IP protocol suite for reliable network communication. TCP, Transmission Control Protocol, establishes a connection between the client and server, ensuring data arrives in the correct order and without errors. Here is a simple breakdown of the TCP communication flow:

- Connection Establishment: The client initiates a connection request to the server on a specific TCP port. The server acknowledges the request, and a connection is established.
- Data Transfer: The client sends control commands as data packets over the established connection. The server receives these packets and acknowledges their receipt.
- Connection Termination: Once communication is complete, the client and the server initiate a connection termination sequence, closing the connection.

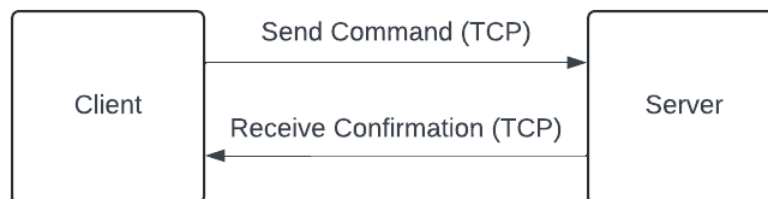
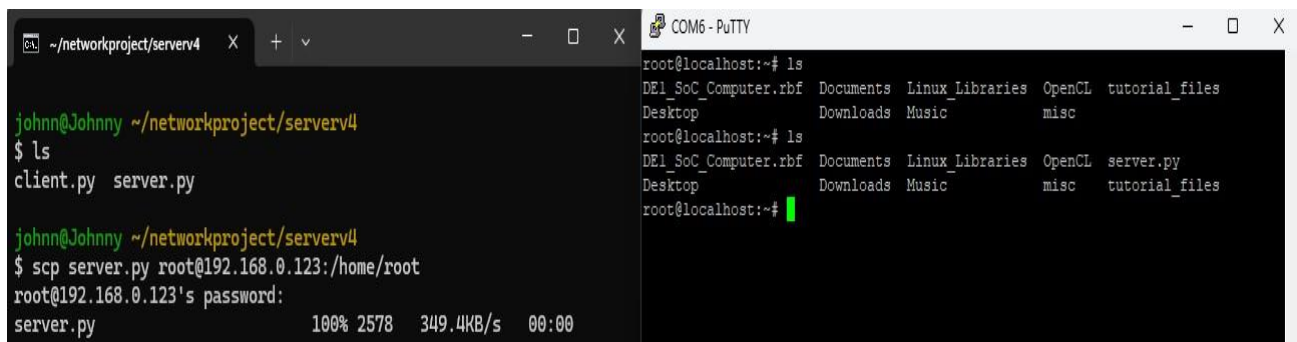


Figure 2. Network Communication Flow

In this diagram we see, the client sends commands to the server using TCP and the server processes these commands and sends back a confirmation to the client.

## Implementation

To get the server program onto the DE1-SoC I had to connect the SoC board to my home router. By doing this it connected the board to my LAN (Local Area Network). Once connected to the LAN, I used Secure Copy Protocol (SCP) to transfer the server program from my desktop computer to the board. SCP is a network protocol that allows file transfer between hosts on a network. It is based on the Secure Shell (SSH) protocol, which allows for the operation of network services over an insecure network. OpenSSH developers recommend using more modern protocols like SFTP (Secure File Transfer Protocol) however, for our problem SCP works.



```
johnn@Johnny ~/networkproject/serverv4
$ ls
client.py  server.py

johnn@Johnny ~/networkproject/serverv4
$ scp server.py root@192.168.0.123:/home/root
root@192.168.0.123's password:
server.py                                100% 2578    349.4KB/s   00:00
```

Figure 3. SCP to De1SoC

In Figure 3, we see SCP being used to transfer server.py over to the De1SoC. Now that sever.py is successfully transferred to the DE1SoC we can host the server on the board.

Now, we can connect to the server on any device on the network using the Client.py file. The Client.py script allows users to establish a connection with the server running on the DE1-SoC board, send commands to control the LEDs remotely, and receive confirmation from the server about the status of the commands sent.



```
COM6 - PuTTY
root@localhost:~# python server.py
Server listening on 0.0.0.0:8080
Connection from 192.168.0.33:65310
Received command: 1
Toggling LEDR1
Received command: 3
Toggling LEDR3
[ ]

Connected to server at 192.168.0.123:8080
Enter a command (or 'exit' to quit): 1
Server response: Command received
Enter a command (or 'exit' to quit): 3
Server response: Command received
Enter a command (or 'exit' to quit):
```

Figure 4. Client-Sever Interaction

In Figure 4, we see the client and server interacting with each other. We see the established TCP connection where the client initiates a connection request to the server on a specific TCP port.

The server acknowledges the request, and a connection is established. The client then sends control commands as data packets over the established connection. The server receives these packets and acknowledges their receipt. Once communication is complete, the client and the server initiate a connection termination sequence, closing the connection.

Figure 4 shows us the client sending the command 1 and 3 which turns LED1 and LED3 on the DE1-SoC. Below is the result of these commands on the board.

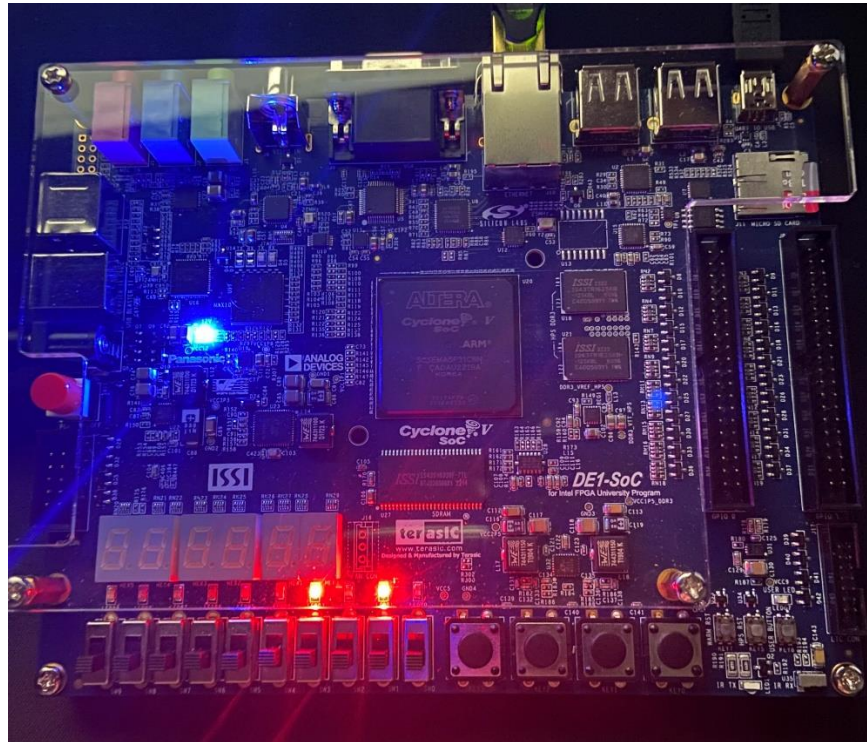


Figure 5. De1-SoC Results

Figure 5 shows us the results on the board. We see that once receiving the commands from the client, the server correctly tells the board to turn LED1 and LED3 on. Through this implementation, the project successfully demonstrates the remote control of hardware components via network commands, providing hands-on experience with network programming, hardware-software integration, and the practical application of these concepts in real-world scenarios.

## Conclusion

This project successfully demonstrated the implementation of a network-controlled LED system using the DE1-SoC board, showcasing the integration of network communication protocols with hardware control. By leveraging both the FPGA and ARM processor on the DE1-SoC, we achieved a seamless client-server interaction that allowed remote control of the board's LEDs via TCP/IP commands. Overall, the project illustrates the potential of network-based control systems in industrial and automation applications, where remote device control is critical. The successful execution of this project underscores the importance of combining theoretical knowledge with practical implementation to solve real-world engineering challenges.

## Citations

[1] “DE1-SoC User Manual”, Altera Corporation, 2013 <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=836&PartNo=4>

[Date accessed: 06/23/2024]

[2] “My\_First\_Fpga.pdf”, Altera Corporation, 2013 <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=836&PartNo=4>

[Date accessed: 06/23/2024]

[3] “My\_First\_Hps.pdf”, Altera Corporation, 2013 <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=836&PartNo=4>

[Date accessed: 06/26/2024]

[4] “SSH File Transfer Protocol” [https://en.wikipedia.org/wiki/SSH\\_File\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol)

[Date accessed: 06/29/2024]

[5] “Socket Programming in Python”, RealPython <https://realpython.com/python-sockets/>

[Date accessed: 06/29/2024]

## Project Repository

The complete source code, documentation, and additional resources for this project are available on GitHub. You can access the repository at <https://github.com/JuanCantu1/Network-Controlled-LED-System>. Additionally, the repository contains a video demonstration of the project. This can be found at the bottom of the README.