Embedded Networking Software Development
Lab 1 – Networking Layer 4 (Transport Layer)

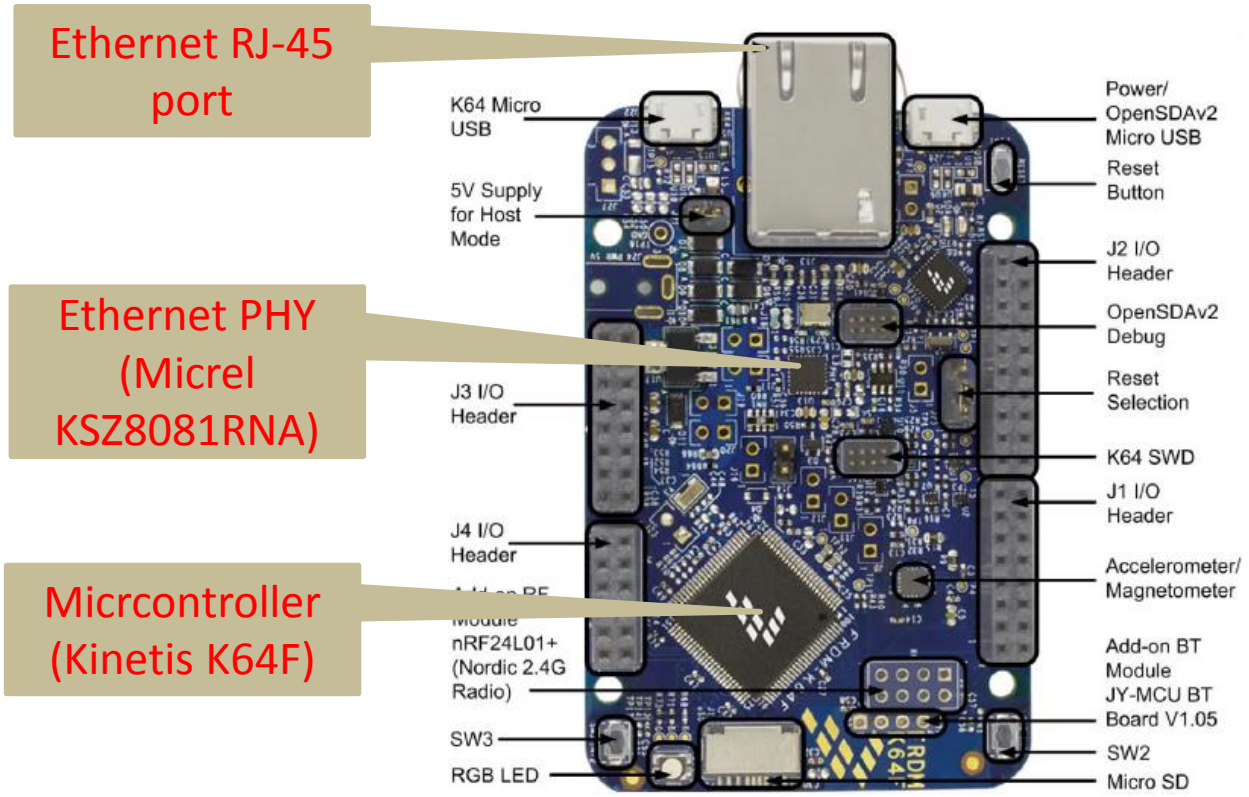
Version 0.1

Overview

- Purpose
  - Study an implementation of the UDP protocol on an embedded networking stack.
- Learning objectives
  - To become familiar with the internals of the UDP protocol
  - To learn how to use the Kinetis Design Studio (KDS) software development environment for the FRDM-K64F board.

Overview (2)

- The FRDM-K64F board



Background Information

- UDP Protocol
- UDP Header

Table 45-112. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 45-113. UDP header fields

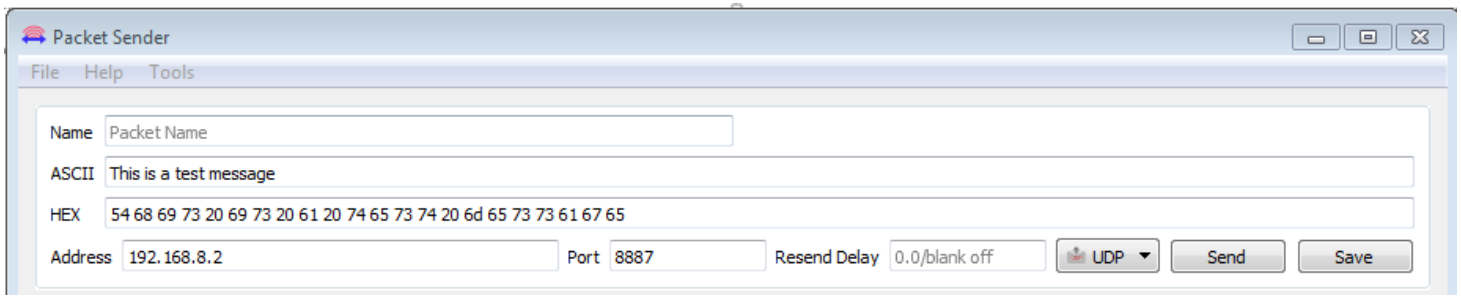
Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

Software Requirements

- Complete the code that implements the support for the UDP protocol in the embedded networking stack for the FRDM-K64F board:
  - Complete the layer-4 function that binds a UDP end-point to a given UDP port number
  - Complete the layer-4 function that sends a UDP datagram over the network
  - Complete the layer-4 function that looks up the UDP end point corresponding to the destination UDP port of an incoming UDP datagram.

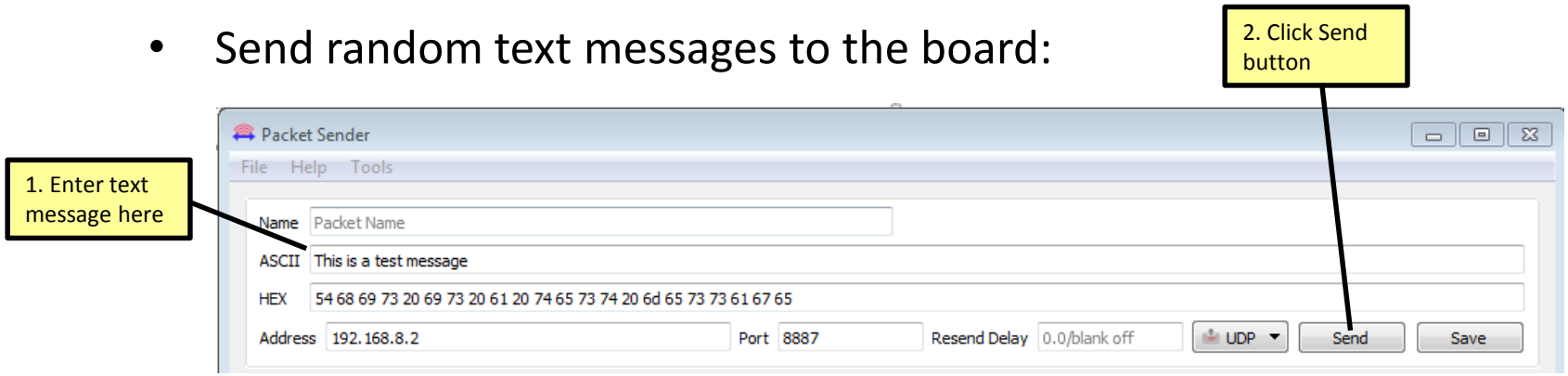
Test Plan

- Test your completed embedded networking stack on the FRDM-K64F board, by using the “Packet Sender” tool from a PC, as follows:
  - Connect the Ethernet port of the board to the Ethernet port of your PC
  - Configure a static IPv4 address for the PC’s wired Ethernet interface that is in the 192.168.8.0/24 subnet (e.g., 192.168.8.1)
  - Configure “Packet Sender” on the PC to send UDP datagrams to port 8887, to the IPv4 address of the board (which by default is 192.168.8.2)



Test Plan (2)

- Test case 1: Random messages
  - Send random text messages to the board:



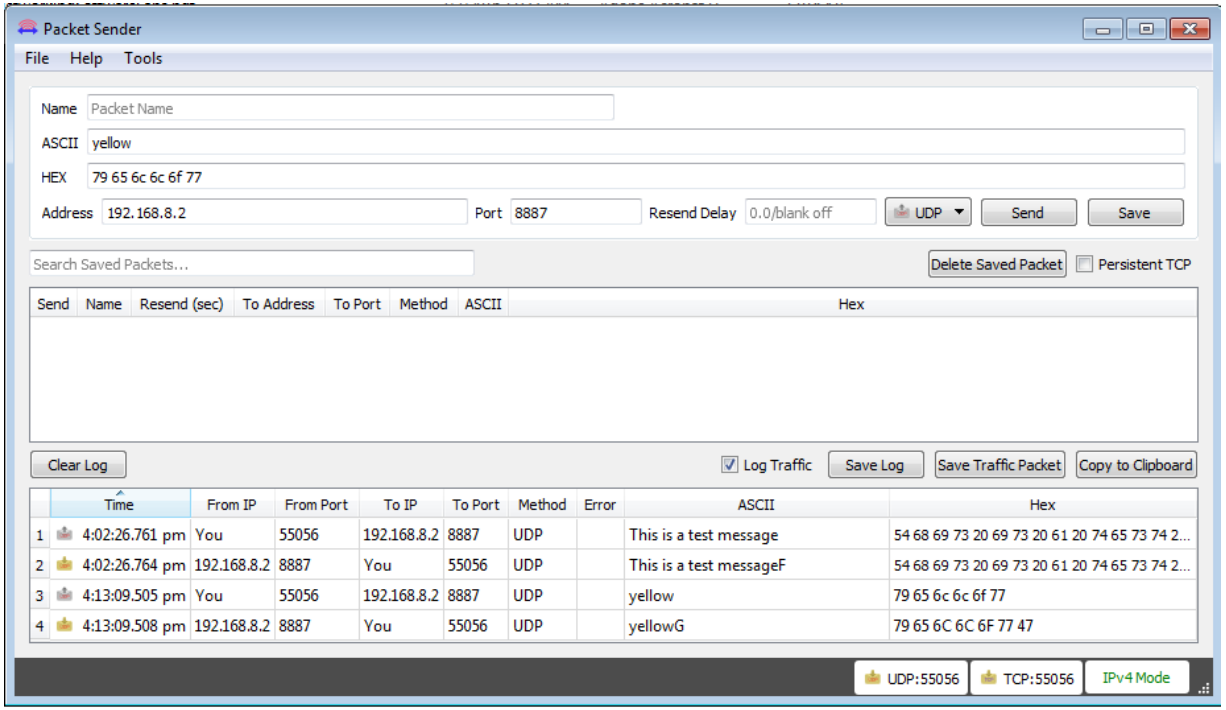
  - The board should respond echoing your message, appending a capital letter ('A' .. 'Z') at the end of the original message:

	Time	From IP	From Port	To IP	To Port	Method	Error	ASCII	Hex
1	4:02:26.761 pm	You	55056	192.168.8.2	8887	UDP		This is a test message	54 68 69 73 20 69 73 20 61 20 74 65 73 74 2...
2	4:02:26.764 pm	192.168.8.2	8887	You	55056	UDP		This is a test messageF	54 68 69 73 20 69 73 20 61 20 74 65 73 74 2...

Sample reply from the board

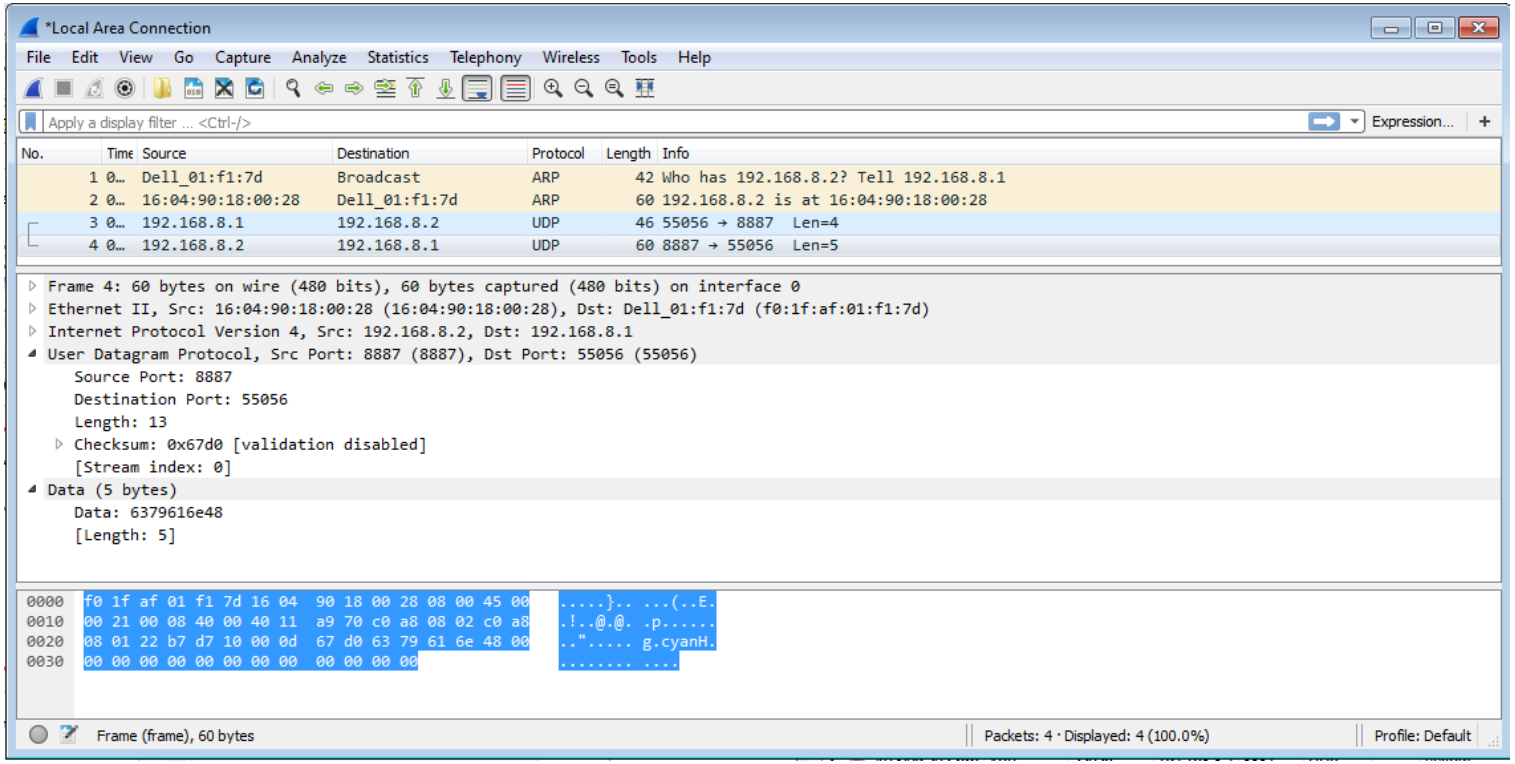
Test Plan (3)

- Test Case 2: Command messages
  - Send “command” messages to the board to change the color of the blinking LED. Valid commands are the strings: “red”, “green”, “blue”, “yellow”, “cyan”, “magenta” and “white”.



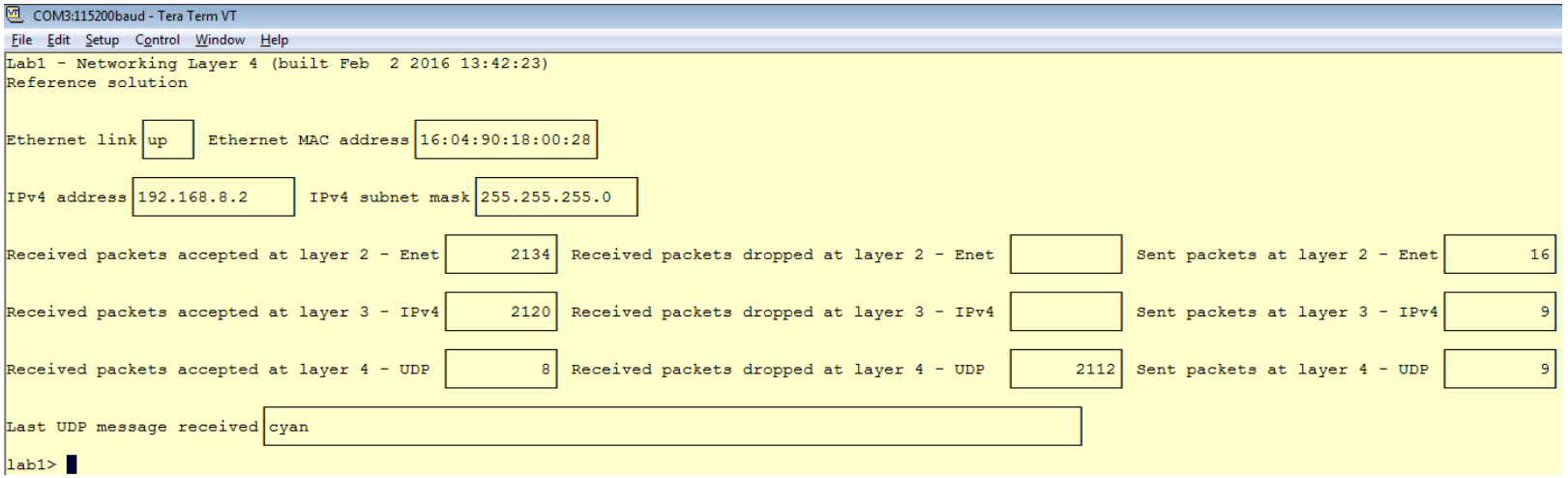
Test Plan (4)

- Test Case 3: Wireshark Analysis
  - Use Wireshark to verify that UDP packets sent by the FRDM-K64F board are well-formed.



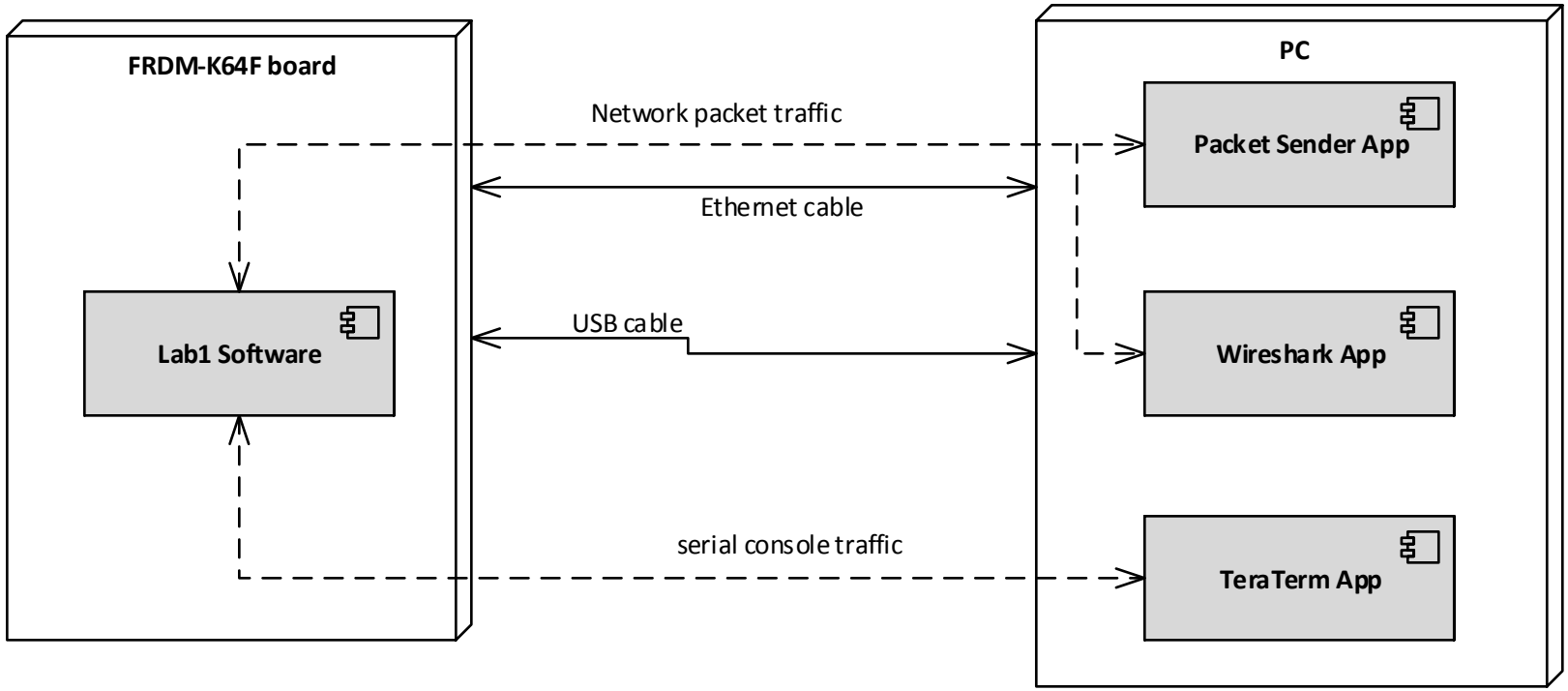
Test Plan (5)

- Test Case 4: Board Network Stats Analysis
  - Start a TeraTerm session and connect it to the board's serial console
  - While sending messages from "Packet Sender", look at the Network stats dashboard on the board's serial console. Verify the consistency of the stats (e.g., # of packets received at layer 4 cannot be greater than #r of packets received at layer 3, and # of packets received at layer 3 cannot be greater than # of packets received at layer 2)



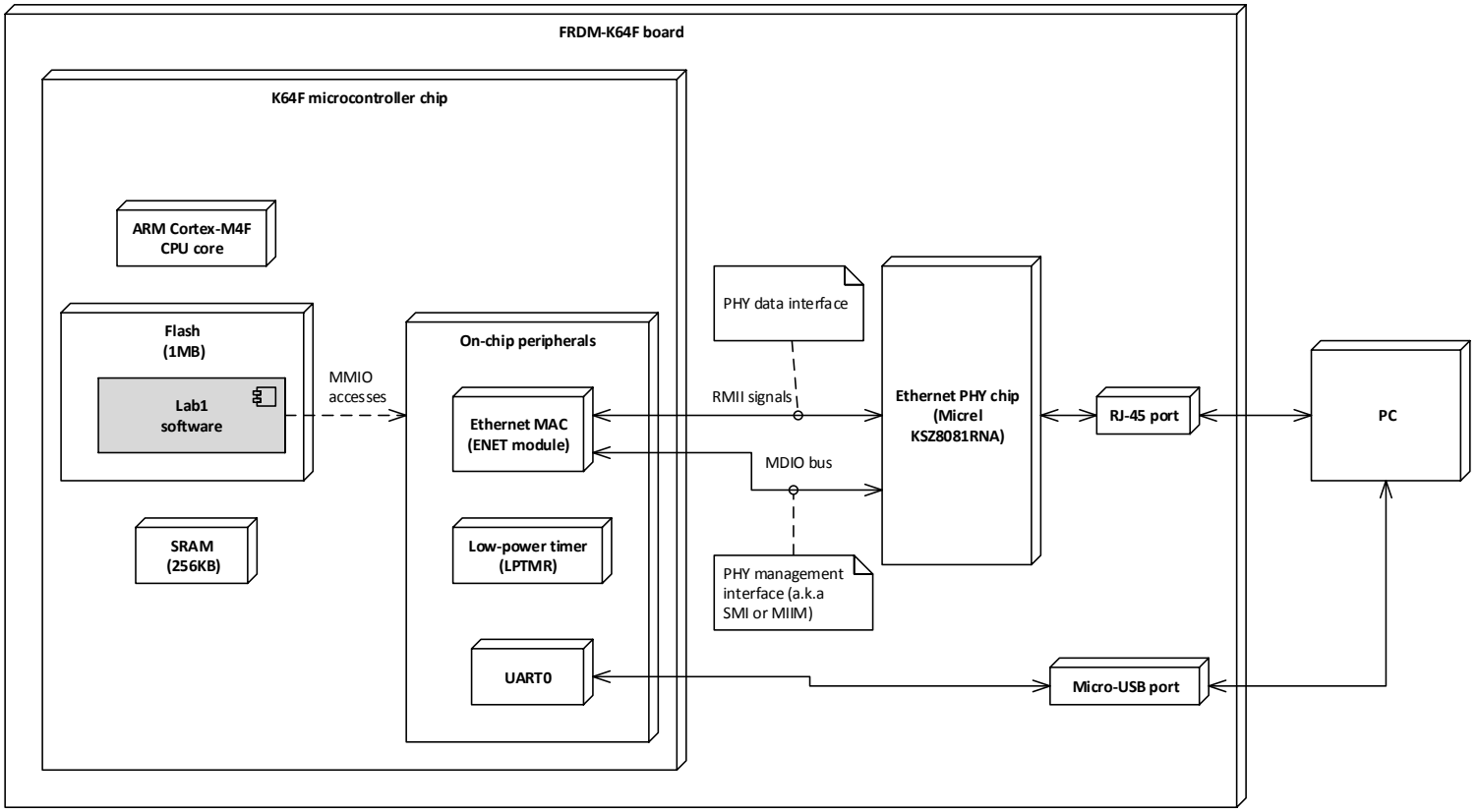
Software Architecture Diagrams

- Hardware/Software Topology



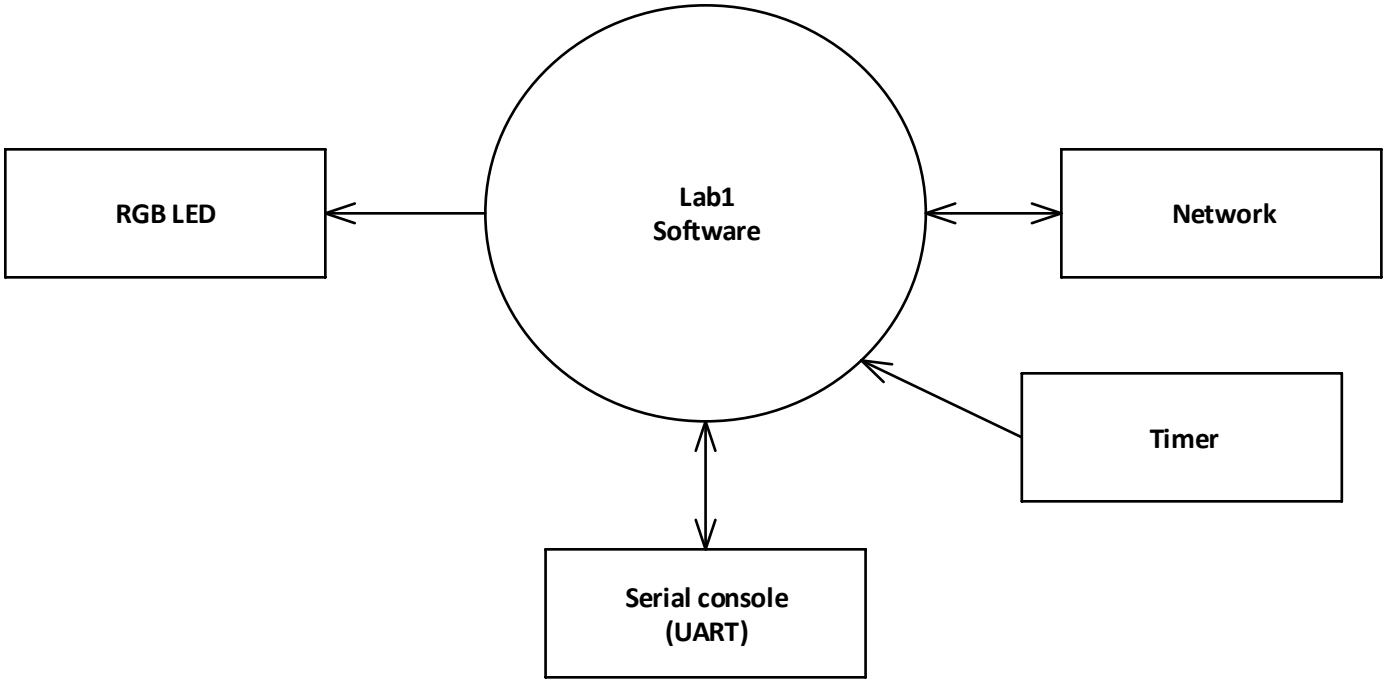
Software Architecture Diagrams (2)

- Hardware Context Diagram



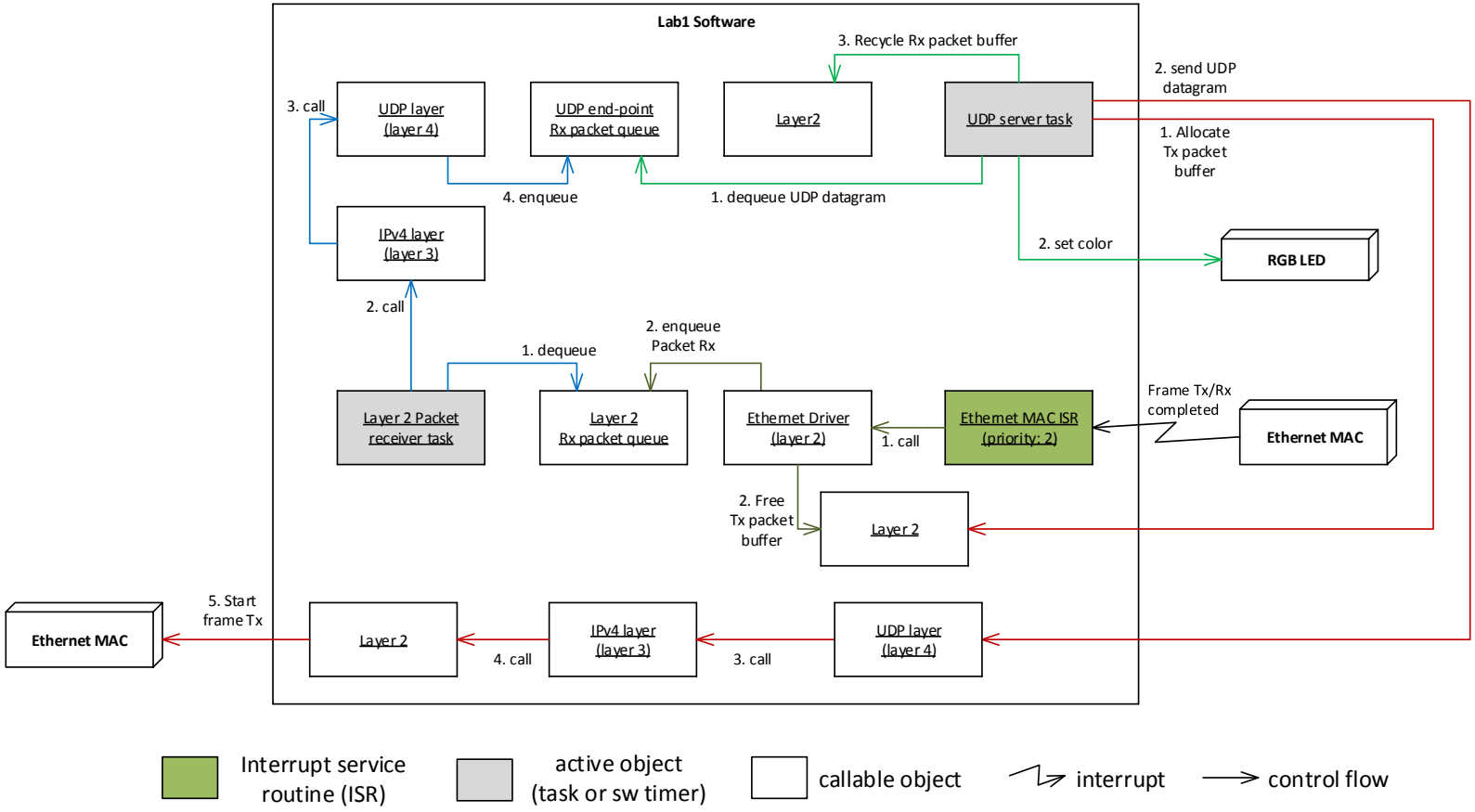
Software Architecture Diagrams (3)

- Software Context Diagram



Software Architecture Diagrams (4)

- Runtime Architecture



Software Architecture Diagrams (5)

- Runtime Architecture (cont.)