Stage 1: Linux Configured NXP Switch

- setup linux build environment
    - run $source setup.env
    - run $bld -c linux -m ls1028ardb
    - if with docker
        - run $bld docker
    - if on mac
        - run the dockerfile manually with $make in flexbuild/docker/debian/Makefile
        - use dockerfile provided to build debian/ubuntu container with required libraries
        - build the linux image with $bld -c linux -m ls1028ardb
- customize the linux kernel
    - after kernel build is complete run $bld linux:menuconfig to configure kernel drivers
    - to enable tsn feature navigate to :
        - Networking support —-> Networking options —-> QoS and/or fair queueing —->
        - set or ensure the following are set as M (modules):
            - Credit Based Shaper (CBS)
            - Earliest TxTime First (ETF)
            - Time Aware Priority (taprio) Scheduler
        - verify if PTP clock support is enabled, Device Drivers —-> PTP clock support
    - build the kernel again if not already set, otherwise skip to next step
    - check device tree file for verification but all tsn feature already configured with ethernet port using enetc and ptp clock; mscc_felix switch configured with internal (felixport4:enetcport2 and felixport5:enetcport3) and external ethernet connection using enetc
- prepare the final sd card flashable boot firmware
    - run $bld bsp -b sd
    - flash it onto sd card
- boot it onto switch
    - put the sd card into switch
    - Set DIP switches (small physical switches) to SD card (check manual)

    - power on switch
    - connect to the switch using UART (Serial) port (USB-to-Serial cable/adapter)
    - Use a terminal emulator progra (Eg: PuTTY) on your PC to access the switch
    - console will display the boot log, After the kernel finishes booting, you will see the login prompt for the Debian OS, where you can log in (default credentials are often root/root or user/user)
    - set static IP manually using the Serial Console first.

- access over ssh

    - Connect one of the NXP switch's Ethernet ports to your PC (or a local network switch then can also use DHCP server)
    - SSH client on your PC to connect to switch's IP address

Stage 2: DS TT port and configured on Switch and tested with existing 5g network

- PTP Synchronization

    - synchronize the PTP hardware clock to a PTP master, run $ptp4l -i eth0 -m
    - synchronize the system clock to the hardware clock, run $phc2sys -s eth0 -w

- test the kernel

    - After booting, use the tc command to confirm that the taprio scheduler is available. run $tc qdisc replace dev eth0 parent root taprio num_tc 8 ...
    - If this command doesn't return an error like "Unknown qdisc", your kernel build was successful.
    - Verify PTP Hardware Clock (high-precision hardware clock from the ENETC). run $ls /sys/class/ptp
    - You should see ptp0, ptp1, etc.

- port DS TT

    - copy DS-TT code into flexbuild/src/apps/my-5g-tsn-tool
    - create makefile if not already created, which will tell FlexBuild how to cross-compile your application
    - Cross-Compile the Application
    - Integrate into the Root Filesystem
    - add rules to copy the compiled TT application binary into your root filesystem during the build process. by adding your package to the Buildroot configuration (bld buildroot:menuconfig)

- test it with existing 5g network test environment by replacing the previous DS TT running on old switch

Stage 3: UPF NW TT port and configured on Switch and tested with existing 5g network

- port UPF NW TT
    - Choose a UPF, select an open-source UPF to start with (e.g., from Open5GS or Free5GC)
    - Port, add the UPF source code as a new package in FlexBuild and cross-compile it
    - modify the source code forwarding rules as required to specific ethernet ports using ioctl/netlink
    - modify the UPF's networking code to offload packet processing to dedicated hardware acceleration engines (e.g., the DPAA2) for faster performance (later after testing without modification)
    - cross compile it
- test it with existing 5g network test environment by removing the UPF NW TT running on pc, now both UPF NW TT and DS TT should run on the NXP Switch

Stage 4: Implementing additional TSN Functionalities

- explore following features:
    - Frame Preemption (IEEE 802.1Qbu & 802.3br)
    - Stream Reservation Protocol (SRP) (IEEE 802.1Qat)
    - Redundancy (IEEE 802.1CB)