# **AMP on PolarFire ICICLE ES Kit**

For getting started information on the Icicle Kit please see the [PolarFire SoC Icicle Kit quick start guide](https://www.microsemi.com/products/fpga-soc/polarfire-soc-icicle-quick-start-guide).

Community support on github and other [forums](https://forum.microchip.com/s/?&page=1&offset=0&filters=false&selectedlist=section1&followingtopics=false&myforums=false&myactivity=false) is also helpful.

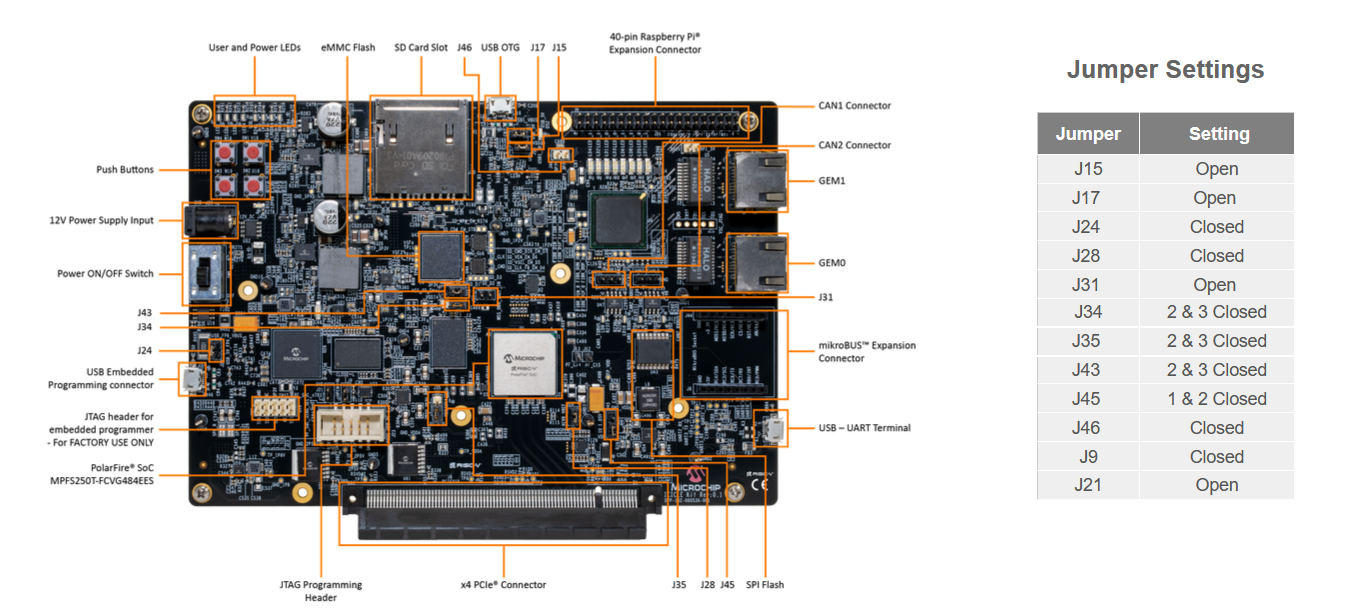


Figure 1: Hardware Kit and Jumper setting

**Refer:** <https://github.com/polarfire-soc/polarfire-soc-documentation/blob/master/reference-designs-fpga-and-development-kits/updating-mpfs-kit.md>

Update the Icicle kit to the latest reference design and Hart Software System (HSS)

**For updating the latest MSS (Microcontroller Subsystem) Configurator design and HSS (Hart-Software System):**

* Download and install the Libero SoC Development Suite from [here](https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/fpga/libero-software-later-versions). (Install it in Windows machine)
  + To verify the software downloads:

Enter the following command on terminal to validate the checksum:

md5sum <path\_to\_installer>

sha256sum <path\_to\_installer>

* Flash a latest Libero MSS component for FPGA design and HSS using Libero SoC. ([Link](https://github.com/polarfire-soc/polarfire-soc-documentation/tree/master/boards/mpfs-icicle-kit-es/updating-icicle-kit))
* Install FlashPro Express and USBImager

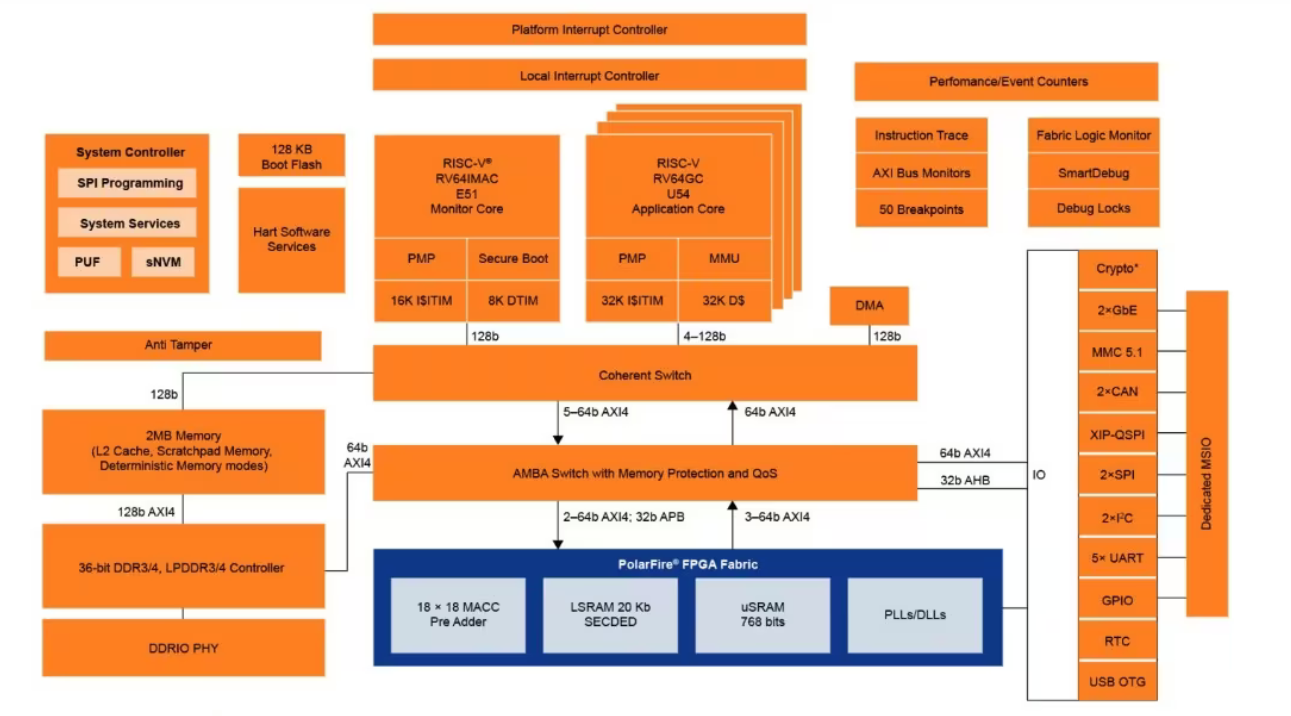
****

Figure 2: Block diagram: We can enable peripherals as needed.

**For Custom Linux using Yocto:** (Pre-built binary images are available [here](https://github.com/polarfire-soc/meta-polarfire-soc-yocto-bsp/releases/))

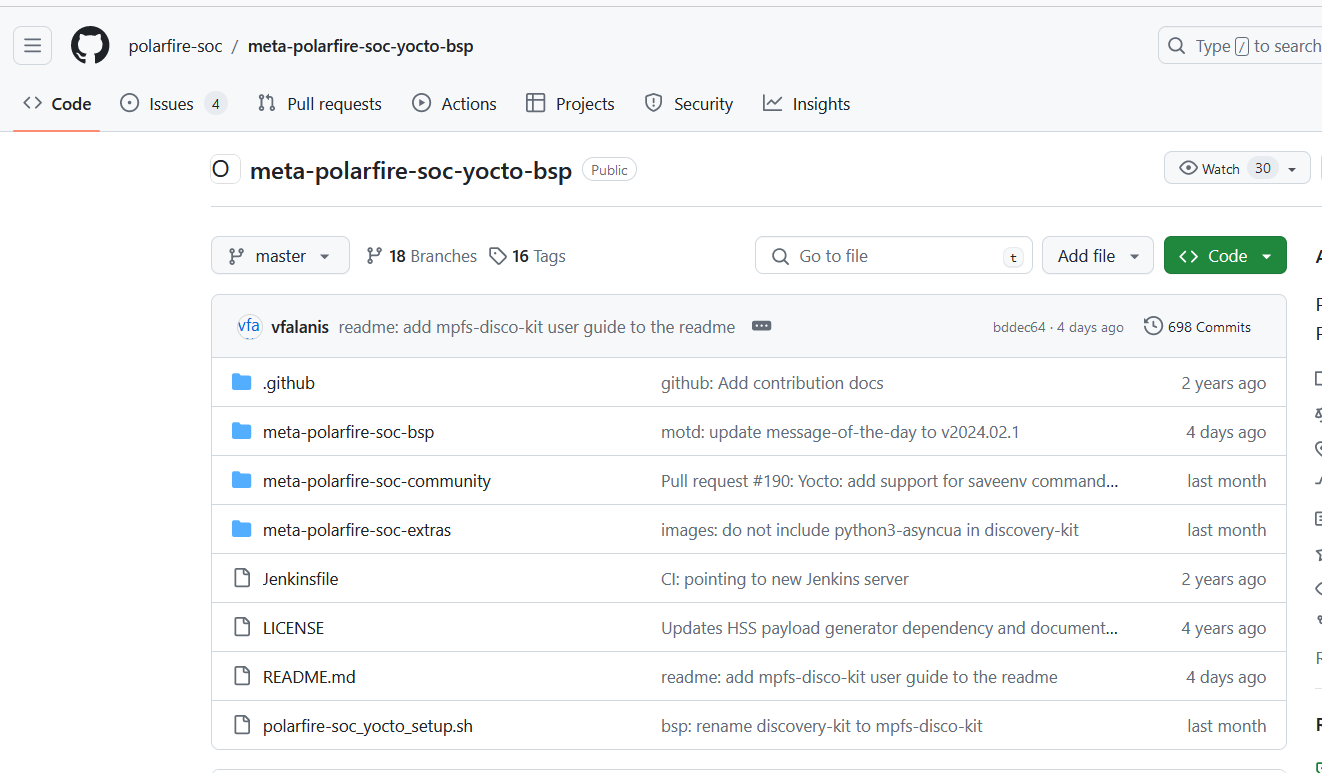
****

Figure 3: git-repo for custom Linux using Yocto

* Create the workspace.
  + $ mkdir yocto-dev && cd yocto-dev
  + $ repo init -u https://github.com/polarfire-soc/polarfire-soc-yocto-manifests.git -b main -m default.xml
* Setup BitBake environment.
  + $. ./meta-polarfire-soc-yocto-bsp/polarfire-soc\_yocto\_setup.sh
* Building a Linux image.
  + MACHINE=icicle-kit-es-amp bitbake mpfs-dev-cli
* Copy the created Disk Image to flash device (USB mmc flash)
  + cd yocto-dev/build
  + bmaptool copy tmp-glibc/deploy/images/icicle-kit-es-amp/mpfs-dev-cli-icicle-kit-es-amp.wic /dev/sdX
* Target machine Linux login.
  + Login with root account, there is no password set.

**Programming a Linux Image:**

Before proceeding with the steps shown below, prepare the Linux image that you want to program to the kit. This could be:

* A pre-built image downloaded from the PolarFire SoC Yocto BSP releases
* An sdcard.img image created using Microchip Buildroot External

**Note:** If using an image generated with the PolarFire SoC Yocto BSP, please extract the image so it has the ".wic" extension. Compressed images, i.e with the ".wic.gz" extension should not be programmed as this can cause issues at boot time.

1. Connect the USB-UART connector to your host PC. This connection will give you access to the PolarFire SoC UARTs
2. Open a terminal application to interact with the HSS through MMUART0. Settings are 115200 baud, 8 data bits, 1 stop bit, no parity, and no flow control.
3. Power on the board and the Microchip logo will be displayed on MMUART0 as the HSS boots.
4. Press a key in the terminal application to stop the HSS from booting. This will give you access to the HSS command line interface and a ">>" for input will be displayed in the terminal.
5. Type mmc to select mmc as a boot source and then usbdmsc in the HSS command line interface. If successful, a message saying "Waiting for USB Host to connect" will be displayed.
6. Connect the USB-OTG connector to your host PC. The eMMC content will be transferred to the kit through this connection.
7. The eMMC should now appear as mass storage device/drive on your host PC.
8. Launch USBImager
9. Select the Image file you would like to program to the eMMC. Note: Linux images are generated with a time stamp; assets from different releases will have different names.
10. Select the target Device to program the image to.
11. Click Write.
12. Once writing has completed, unmount/eject the drive from the host PC and press CTRL+C in the HSS command line interface. Disconnect the micro-USB cable from the USB-OTG connector.
13. Type boot to boot the newly copied Linux image.
14. HSS boot messages will appear on MMUART0 and the Linux boot will appear on MMUART1.

**Download Putty:** Serial-port terminal

Assuming you opted for the freertos build, the default if no override is selected, booting the Icicle Kit with the AMP image installed will at first seem immediately familiar: Connect to the first of the four USB UART buses, which appear when you plug a micro-USB cable into the correct port on the Icicle Kit and you'll be greeted with Hart Software Services' boot sequence; switch to the second UART bus and you'll see a Linux login prompt.

* Open four Putty terminal.
* For the configuration:
  + Connection - Serial
  + Device name - /dev/ttyUSB0, /dev/ttyUSB1, /dev/ttyUSB2, /dev/ttyUSB3
  + Baud rate - 115200

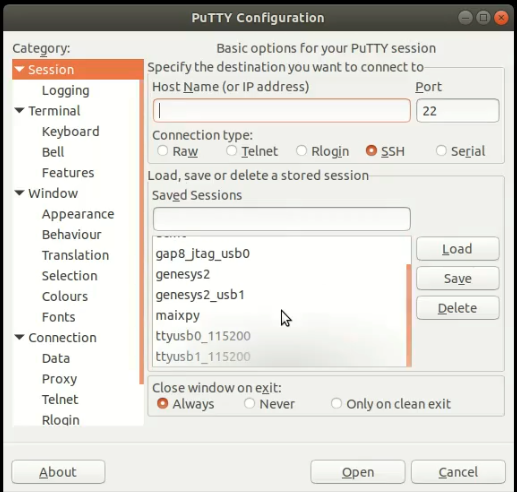


Figure 4: Putty Configuration



Figure 5: Dependencies to get various SW

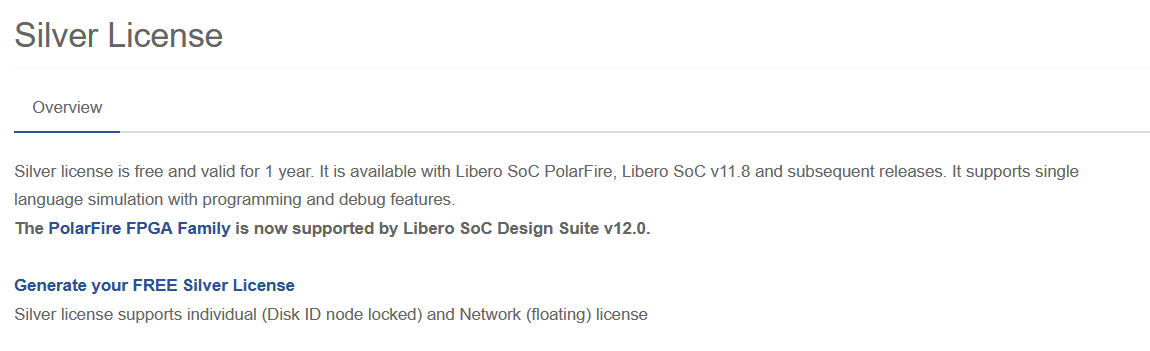


Figure 6: License needed for getting started

**Steps to run the code on PolarFire SoC ICICLE ES Kit**

* Connect the power cable.
* Connect the USB-UART cables.
* Power on the board. Board will boot the Linux and asks for login password.

Password – **root**

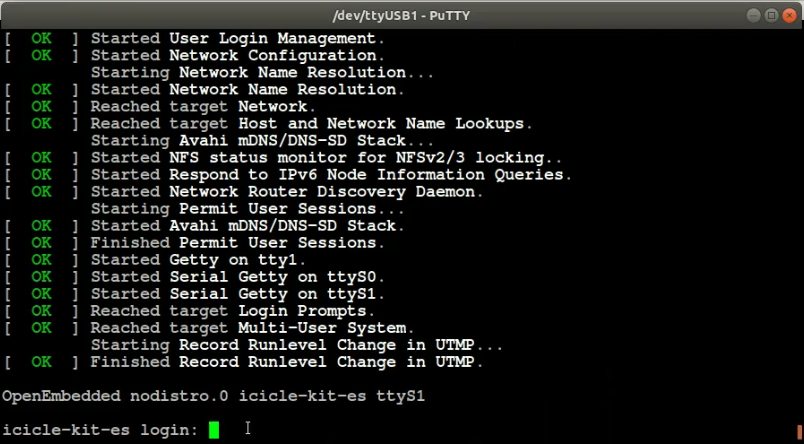


Figure 7: Kit login password

Microprocessor subsystem (MSS):

MSS configurator is a software tool to configure the PolarFire SoC microprocessor subsystem.

The MSS Configurator is used to configure:

* MSS clocks
* Fabric interfaces
* I/O banks
* DDR memories
* Debug features

The MSS configurator software is available in two options:

* Stand-alone software
* Installed with the Libero SoC Design Suite v12.5 (or later) as PFSoC MSS Configurator.

Hart Software Services (HSS):

HSS is a collection of services that run on the E51 monitor core.

HSS is used for the following:

* Program memory using USB mass storage or MODEM transfer.
* Copy a program (Linux or Bare Metal) from a non-volatile storage (for example, eMMC or SD card) to the LIM or DDR.
* Create a payload containing multiple applications to be booted and run.
* Pass messages between cores in the MSS.

The HSS uses Bare Metal drivers to initialize the system, which are found in the PolarFire SoC Bare Metal Library. It also relies on XML generated by the PolarFire SoC MSS Configurator to configure the system on boot. The HSS comprises of the following:

* A superloop monitor running on the E51 processor, which receives requests from the individual U54 application processors to perform certain services on their behalf.
* A Machine-Mode software interrupt trap handler, which allows the E51 to send messages to the U54s, and requests them to perform certain functions for it related to rebooting the U54.