SPI (Serial Peripheral Interface) Guide for BeagleY-AI

by Brian Fraser
Last update: Mar 31, 2025

Guide has been tested on

      BeagleY-AI (Target):   <mark>**Debian 12.8**</mark> (bit it's version independent)
      PC OS (host):   <mark>**Debian 12.8**</mark> (or higher)

This document guides the user through

1. Setting up the hardware SPI

Table of Contents

Formatting

1. Commands for the host Linux's console are show as:
   ```
   (host)$ echo "Hello PC world!"
   ```
2. Commands for the target (BeagleY-AI) Linux's console are shown as:
   ```
   (byai)$ echo "Hello embedded world!"
   ```

Revision History

- Mar 31, 2025: Target the Beagle-Y AI

1. SPI

The LCD on the Zen Hat is connected via the Serial Peripheral Interface (SPI). It is a four wire protocol:

- Serial Clock (`SCLK`): This is the clock that controls when data is transmitted.

- Master Out, Slave In (`MOSI`): Transmits data from the CPU to the LCD

- Master In, Slave out (`MISO`): Transmits data from the device back to the CPU. This is not used for the LCD because it sends no data to the CPU.

- Chip Enable (`CE`): Also called chip select, or device enable. This allows multiple devices to share the same SPI bus. In this case, `CE0` is wired to the LCD. `CE1` is connected to the `PYMNL` headre on the Zen Hat and can be connected to a different SPI device so that it shares the same `SCLK`, `MOSI`, and `MISO` wires (also on the `PYMNL` header).

This guide is heavily based on the [a guide produced by William Schmidt](#) (student in CMPT 433, Spring 2025). Thanks William!

1. On the target, copy the device tree source files to your home directory:

    - Check your installed kernel version
      ```
      (byai)$ uname -r
      6.1.83-ti-arm64-r63
      ```

    - List the Device Tree source versions installed on the target:
      ```
      (byai)$ ls -d1 /opt/source/dtb*
      /opt/source/dtb-5.10-ti
      /opt/source/dtb-6.1-Beagle
      /opt/source/dtb-6.6-Beagle
      /opt/source/dtb-6.9
      ```

    - Copy the device tree source folder which matches your kernel to your home directory so we can modify it. In my case, it's 6.1:
      ```
      (byai)$ cp -R /opt/source/dtb-6.1-Beagle ~
      ```

2. Clone William Schmidt's SPI device tree file which supports the MCU SPI (hardware):
   ```
   (byai)$ cd ~
   (byai)$ git clone https://github.com/wcs3/BYAI-mcu_spi0.git
   ```

    - You will need internet access to do this.

3. Copy the new device tree into the device tree source folder:
   ```
   (byai)$ cd ~/dtb-6.1-Beagle/src/arm64/overlays/
   (byai)$ cp ~/BYAI-mcu_spi0/k3-am67a-beagley-ai-spidev0-mcu.dts .
   ```

4. Change the default `spi0` device to use this new device tree that supports the hardware SPI:

    - Edit the base device tree:
      ```
      nano ~/dtb-6.1-Beagle/src/arm64/ti/k3-am67a-beagley-ai.dts
      ```

    - Find the `alias` section and change the `spi0` line as highlighted below:

```
        aliases {
                serial0 = &wkup_uart0;
                serial2 = &main_uart0;
                serial3 = &main_uart1;
                serial6 = &main_uart6;
                mmc1 = &sdhci1;
                mmc2 = &sdhci2;
                rtc0 = &rtc;
                spi0 = &mcu_spi0;
                usb0 = &usb0;
                usb1 = &usb1;
                i2c1 = &mcu_i2c0;
        };
```

- Save and close the file: CTRL+X, then Y.

5. Build and install the device tree from our modified sources:
```
(byai)$ cd ~/dtb-6.1-Beagle/
(byai)$ make
(byai)$ sudo make install_arm64
```

6. Enable the SPI overlay and configure buffer size:
```
(byai)$ nano /boot/firmware/extlinux/extlinux.conf
```

   - In the `(default)` section at the bottom of the file, find the line starting with '`fdtoverlays`' (may be commented out with a #). Change this line to be:
     ```
     fdtoverlays /overlays/k3-am67a-beagley-ai-spidev0-mcu.dtbo
     ```

     - If you already have overlays listed, append this new DTBO file to your existing line, space separated.

   - For better performance, configure the SPI buffer size by appending the following to the end of the line "`append console=...`":
     spidev.bufsiz=131072

     - Our screen is 240x240 pixels at 16 bit colour depth = 115,200 bytes; so rounding up to nearest power of 2 (2^{17})

   - It should now look like (changes highlighted; long line omits some content with ...)
```
label microSD (default)
    kernel /Image
    append console=ttyS2,115200n8 ... spidev.bufsiz=131072
    fdtdir /
    fdt /ti/k3-am67a-beagley-ai.dtb
    fdtoverlays /overlays/k3-am67a-beagley-ai-spidev0-mcu.dtbo
```

7. Reboot:
```
(byai)$ sudo reboot -f
```

8. Verify SPI shows up:
```
(byai)$ ls /dev/spidev*
/dev/spidev0.0  /dev/spidev0.2
```

   - As noted by William Schmidt, spidev0.0 is the first device (our LCD), and if you want to connect a second device to the same SPI bus (using CE1) it will be accessed via spidev0.2.