# Writeup 3

CS 444
November 10, 2018

Anton Synytsia, Eytan Brodsky, David Jansen

Contents

# I. Required Technologies

The following technologies are required to perform this lab:

1) Raspberry Pi, preferably model 3B+,
2) Micro SD card and card reader, with at least 4GB space.
3) 3.3V TTL UART to USB converter for establishing serial console.
4) Power adapter or charger for Raspberry Pi.
5) Access to `OS2` server.

# II. Setup

The following sections describe how to set up Raspbian Stretch Lite on Raspberry Pi. The first section describes how to setup Raspbian on SD card, the second section focuses on testing Raspbian with serial console, the third section describes how to build Raspberry Pi Linux kernel on `OS2`, and the fourth section describes how to setup a built Linux kernel image on SD card.

## *A. Downloading and Setting Up Raspbian*

Refer to the following steps for setting up Raspbian Stretch Lite on SD card:

1) Download and extract `2018-10-09-raspbian-stretch-lite.zip` from https://www.raspberrypi.org/downloads/raspbian/.
2) Download and install Etcher from https://www.balena.io/etcher/.
3) Mount the micro SD card to your laptop, via the SD card reader.
4) Start Etcher and do the following:
   a) Set image to `2018-10-09-raspbian-stretch-lite.img` (or alike).
   b) Set drive to SD card.
   c) Click textitFlash!
5) Once the setup is complete, navigate to the SD card drive and use text editor to append the following to `config.txt`:

```
kernel=kernel8.img
enable_uart=1
```

## *B. Running Raspbian with Serial Console*

The following steps, heavily based on Adafruit guide, describe how to initiate a Raspbian serial console session:

1) Install Prolific Chipset and SiLabs CP210X drivers for the TTL serial cable [1].
2) Connect black, white, and green wires to the outer pins 3, 4, and 5 respectively [2].
3) Leave red wire unpinned, as the a separate power adapter is used instead [2]. It is important that only one power source is used as the board can get damaged [2].
4) Insert the micro SD card into Raspberry Pi.
5) Insert the TTL serial cable USB into your laptop.

6) Start Putty and do the following:
   a) Set *Connection type* to serial mode.
   b) Set *Serial line* to `COM6`; COM6 here refers to the port of our TTL serial cable. To determine the port of your TTL serial cable, on Windows platform, access *Device Manager* and check for the available ports; for other platforms, refer to Adafruit guide [2].
   c) Set *Speed* to 115200 [2].
   d) Click *Open*.
7) Connect the power adapter to Raspberry Pi. It is important that this step is performed after initiating the serial console session.
8) (Optional) Within the serial console, press *RETURN* key to activate communications [2].
9) After loading, use `pi` as user name and `raspberry` as password.

### C. Raspberry Pi Linux Kernel

*1) Setting Up:* Perform the following steps for downloading and setting up version 4.14.y Raspberry Pi Linux kernel on `OS2`:

```
cd /scratch/fall2018/group1
git clone git@github.com:raspberrypi/linux.git
cd linux
# checkout v4.14.y
git checkout tags/raspberrypi-kernel_1.20180417-1
```

*2) Compiling:* Execute the following set of commands to build Raspberry Pi Linux Kernel on `OS2`:

```
cd linux
KERNEL=kernel8
make -j4 ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- bcmrpi3_defconfig
make -j4 ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- all
```

Once built, refer to the next section for setting up the built image on SD card.

*3) Uploading to SD Card:* Refer to the following steps for setting up `kernel8.img` on SD card:

1) Download `Image` from `arch/arm64/boot/Image` to your local file system, either using WinCP or another file transfer protocol.
2) Rename `Image` to `kernel8.img`.
3) Mount SD card to your laptop.
4) Copy `kernel8.img` to SD card, so that it is located at the same path as `kernel7.img`.

## III. Morse Code LED Trigger

The following sections describe our solution to Raspberry Pi Morse code LED trigger, as well as, instructions for compiling, setting up, and running the blinker.

### A. Solution

### B. Compiling

Perform the following steps to compile Raspberry Pi with our Morse code LED trigger:

1) Provided that Raspberry Pi Linux Kernel is cloned and checked out to the correct version at your local space, on `OS2`, copy `linux` folder, shipped with this repository, to your `linux` folder. This will overwrite and add the following files to `linux/drivers/leds/trigger/`:

   ledtrig-morse.c Our Morse code LED trigger.

   Kconfig Configures our Morse code LED trigger.

   Makefile Registers our Morse code LED trigger.

2) Executed the following set of commands to rebuild the kernel:

```
cd linux
make clean
KERNEL=kernel8
make -j4 ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- bcmrpi3_defconfig
make -j4 ARCH=arm64 CROSS_COMPILE=aarch64-linux-gnu- all
```

### C. Setting Up

Upload the new, built kernel image to SD card, described in II-C3 section. You may have to delete the original `kernel8.img` from the SD card first though.

### D. Running

Start a new serial console session, described in II-B section. Then run the following set of commands to activate the Morse code LED trigger:

```
sudo su
echo none > /sys/class/leds/led0/trigger # optional
echo morse > /sys/class/leds/led1/trigger
```

## References

[1] S. Monk, "Software installation," https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable/software-installation-mac, Nov 2018.

[2] ——, "Adafruit raspberry pi lesson 5. using a console cable," https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable/overvi Nov 2018.