

# CSE530: EOSI - ASSIGNMENT 3 SETUP INSTRUCTIONS

## Part 1

**Sushant Trivedi (1213366971) and Ravi Bhushan (1214347783)**

### SPI DEVICE DRIVER FOR LED STRIP (WS2812)

#### Files Included:

1. spi\_device.c
2. spi\_driver.c
3. led\_test.c (This is the user level test program)
4. Makefile

#### Steps to Setup

**1. Connect the LED Strip to its respective pins as mentioned below:**

IO11 (SPI MOSI) → DI

5V → 5V

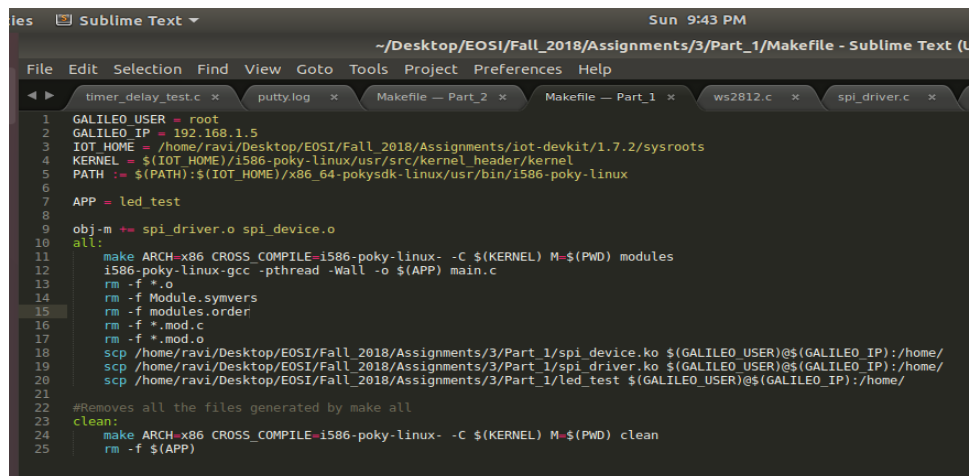
GND → GND

**2. Update the makefile as per the following instructions**

**GALILEO\_USER → Enter the user name. Default “root”**

**GALILEO\_IP → Enter Galileo IO. Default “192.168.1.5”**

**IOT\_HOME → Enter sysroots address**



```
1 GALILEO_USER = root
2 GALILEO_IP = 192.168.1.5
3 IOT_HOME = /home/ravi/Desktop/EOSI/Fall_2018/Assignments/iot-devkit/1.7.2/sysroots
4 KERNEL = $(IOT_HOME)/i586-poky-linux/usr/src/kernel_header/kernel
5 PATH = $(PATH):$(IOT_HOME)/x86_64-pokysdk-linux/usr/bin/i586-poky-linux
6
7 APP = led_test
8
9 obj-m += spi_driver.o spi_device.o
10
11 all:
12     make ARCH=x86 CROSS_COMPILE=i586-poky-linux- -C $(KERNEL) M=$(PWD) modules
13     rm -f *.o
14     rm -f Module.symvers
15     rm -f modules.order
16     rm -f *.mod.c
17     rm -f *.mod.o
18     scp /home/ravi/Desktop/EOSI/Fall_2018/Assignments/3/Part_1/spi_device.ko $(GALILEO_USER)@$(GALILEO_IP):/home/
19     scp /home/ravi/Desktop/EOSI/Fall_2018/Assignments/3/Part_1/spi_driver.ko $(GALILEO_USER)@$(GALILEO_IP):/home/
20     scp /home/ravi/Desktop/EOSI/Fall_2018/Assignments/3/Part_1/led_test $(GALILEO_USER)@$(GALILEO_IP):/home/
21
22 #Removes all the files generated by make all
23 clean:
24     make ARCH=x86 CROSS_COMPILE=i586-poky-linux- -C $(KERNEL) M=$(PWD) clean
25     rm -f $(APP)
```

### 3. Insert the modules loaded on the /home directory onto kernel using the following commands in any order. Please see the image for the expected output

```
cd /home
```

```
insmod spi_device.ko
```

```
insmod spi_driver.ko
```

**NOTE:** Please remove any earlier inserted modules which may be requesting the same GPIOs as this will cause our module to fail.

```
[ 24.119226] IPv6: ADDRCONF(NETDEV_UP): enp0s20f6: link is not ready
[ OK ] Started MPD daemon.
[ OK ] Started Hostname Service.
[ OK ] Started Set hostname to galileo + emp MAC.
[ 25.020466] stmmaceth 0000:00:14.6 enp0s20f6: Link is Up - 100Mbps/Full - flow control rx/tx
[ 25.028344] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s20f6: link becomes ready

quark login: root
root@quark:~# ls
ip_set.sh
root@quark:~# ./ip_set.sh
root@quark:~# [ 389.165618] random: nonblocking pool is initialized

root@quark:~#
root@quark:~#
root@quark:~# cd /home
root@quark:/home# insmod spi_driver.ko
[ 1468.802913] SPI_DRIVER: INIT SUCCESS
root@quark:/home# insmod spi_device.ko
[ 1474.000000] spi_driver: Probe Function Called
Putty SSH Client E: Device Created WS2812
[ 1474.000000] spi_driver: INIT COMPLETED
root@quark:/home# cd /sys/class/WSRing/WS2812
root@quark:/sys/class/WSRing/WS2812# ls
dev power subsystem uevent
root@quark:/sys/class/WSRing/WS2812# cd /dev/WS2812
-sh: cd: /dev/WS2812: Not a directory
root@quark:/sys/class/WSRing/WS2812# cd /dev/
root@quark:/dev# ls
-sh: ls: command not found
root@quark:/dev# ls
WS2812
autofs      log          ttyq4        spidev1.0    tty21        tty39        tty56        tty4         ttyq6        vcs3
autofs      loop-control ttyq5         stderr        tty22        tty4         ttyq7        ttyq5        ttyq7        vcs4
block       loop         ttyq6         stdin         tty23        tty40        ttyq8        ttyq6        ttyq8        vcs5
bus         loop         ttyq7         stdout        tty24        tty41        ttyq9        ttyq7        ttyq9        vcs6
char        mem          ttyq8         tty           tty25        tty42        ttyq         ttyq8        ttyqa        vcsa
console     memory_bandwidth ttyq9         tty0          tty26        tty43        ttyq0        ttyq9        ttyqb        vcsa1
core        mcbk0        ttyq          tty1          tty27        tty44        ttyq1        ttyqa        ttyqc        vcsa2
cpu_dma_latency mcbk0p1      ttyq0         tty10         tty28        tty45        ttyq2        ttyqb        ttyqd        vcsa3
disk        mqeue        ttyq1         tty11         tty29        tty46        ttyq3        ttyqc        ttyge        vcsa4
fd          net          ttyq2         tty12         tty3         tty47        ttyq4        ttyqd        ttygf        vcsa5
full        network_latency ttyqb         tty13         tty30        tty48        ttyq5        ttyge        udev_network_queue vcsa6
fuse        network_throughput ttyqc         tty14         tty31        tty49        ttyq6        ttygf        vga_arbiter
ipset       null         ttyqd         tty15         tty32        tty50        ttyq7        ttygrintk    uinput        vici
i2c=0       port         ttyqe         tty16         tty33        tty50        ttyq8        ttyg0        ui0           watchdog
iio:device0 ppp          ttyqf         rFkill        tty34        tty51        ttyq1        ttyg1        uiol          watchdog0
i2c=0       ptax         ttyq0         rtc           tty18        tty52        ttyq2        ttyg2        urandom       zero
initctl     ptg0         ttyq1         rtc0          tty19        tty53        ttyq3        ttyg3        vcs1
input       pts          ttyq2         shm           tty2         tty37        ttyq4        ttyg4        vcs2
kmsg        ttyq0        ttyq3         snapshot      tty20        tty38        ttyq5        ttyg5
root@quark:/dev#
root@quark:/dev#
root@quark:/dev#
root@quark:/dev#
root@quark:/dev#
```

As can be seen, in the above image the WS2812 device has been created in the /sys/class/WSRing/ and /dev/ directory

### 4. User Level Test Code: Run the ./led\_test command from /home directory. This would ask you to input the following entries:

- No of LEDs to switch on currently
- No of times you want the pattern to run in circles
- Led color for each led that you want to light up

```

2863,077862] WS_WRITE: Write Completed
2863,084614] WS_WRITE: Write Completed
2863,089047] WS_WRITE: Write Completed
2863,089790] WS_WRITE: Write Completed
2863,101470] WS_WRITE: Write Completed
2863,105910] WS_WRITE: Write Completed
2863,112757] WS_WRITE: Write Completed
2863,117293] WS_WRITE: Write Completed
2863,123957] WS_WRITE: Write Completed
2863,128390] WS_WRITE: Write Completed
2863,133225] WS_WRITE: Write Completed
2863,133657] WS_WRITE: Write Completed
2863,146396] WS_WRITE: Write Completed
2863,152087] WS_WRITE: Write Completed
2863,155520] WS_WRITE: Write Completed
2863,163285] WS_WRITE: Write Completed
[C 2864,034402] WS_RELEASE: Driver Released Successfully

root@quark:/home#
root@quark:/home#
root@quark:/home#
root@quark:/home#
root@quark:/home#
root@quark:/home#
root@quark:/home#
root@quark:/home# ./led_tester
sh: ./led_tester: No such file or directory
root@quark:/home# ./led_test
sh: ./led_test: No such file or directory
root@quark:/home# ./led_test
[ 2893,603924] WS_OPEN: Device opened
[ 2893,607423] WS_IOCTL: RESET Command Received
How many LEDs do you want to light up?(1-16)16
How many times do you want the circular display to repeat?(0=infinite)1
Color Panel
1 - Red      2 - Green  3 - Blue   4 - Yellow
5 - White   6 - Violet  7 - Cyan   8 - Magenta
9 - Orange  10 - Brown  11 - Maroon 12 - Pink
13 - Custom Color(Need to enter RGB value)
LED 1 color[1-13]1
LED 2 color[1-13]2
LED 3 color[1-13]3
LED 4 color[1-13]4
LED 5 color[1-13]5
LED 6 color[1-13]6
LED 7 color[1-13]7
LED 8 color[1-13]8
LED 9 color[1-13]9
LED 10 color[1-13]10
LED 11 color[1-13]11
LED 12 color[1-13]12
LED 13 color[1-13]1
LED 14 color[1-13]2
LED 15 color[1-13]3
LED 16 color[1-13]4

```

As can be seen in the image above, we were asked for the choices as explained earlier.

Upon successful user input, it successfully calls WS\_WRITE function from the driver file\_operations datastructures. This is evident by the printing as shown below.

```

3063,172697] WS_WRITE: Write Completed
3063,183345] WS_WRITE: Write Completed
3063,193074] WS_WRITE: Write Completed
3063,195544] WS_WRITE: Write Completed
3063,203705] WS_WRITE: Write Completed
3063,208130] WS_WRITE: Write Completed
3063,213671] WS_WRITE: Write Completed
3063,219389] WS_WRITE: Write Completed
3063,225232] WS_WRITE: Write Completed
3063,231580] WS_WRITE: Write Completed
3063,238396] WS_WRITE: Write Completed
3063,243193] WS_WRITE: Write Completed
3063,247625] WS_WRITE: Write Completed
3063,254459] WS_WRITE: Write Completed
3063,259540] WS_WRITE: Write Completed
3063,265717] WS_WRITE: Write Completed
3063,271436] WS_WRITE: Write Completed
3063,275678] WS_WRITE: Write Completed
3063,282797] WS_WRITE: Write Completed
3063,287198] WS_WRITE: Write Completed
3063,294023] WS_WRITE: Write Completed
3063,298460] WS_WRITE: Write Completed
3063,305311] WS_WRITE: Write Completed
3063,309745] WS_WRITE: Write Completed
3064,316798] WS_WRITE: Write Completed
3064,321530] WS_WRITE: Write Completed
3064,325769] WS_WRITE: Write Completed
3064,333613] WS_WRITE: Write Completed
3064,339052] WS_WRITE: Write Completed
3064,344685] WS_WRITE: Write Completed
3064,349298] WS_WRITE: Write Completed
3064,356163] WS_WRITE: Write Completed
3064,361807] WS_WRITE: Write Completed
3064,366326] WS_WRITE: Write Completed
3064,372124] WS_WRITE: Write Completed
3064,377655] WS_WRITE: Write Completed
3064,384367] WS_WRITE: Write Completed
3064,388788] WS_WRITE: Write Completed
3064,395612] WS_WRITE: Write Completed
3064,401309] WS_WRITE: Write Completed
3064,405763] WS_WRITE: Write Completed
3064,412545] WS_WRITE: Write Completed
3064,418984] WS_WRITE: Write Completed
3064,423788] WS_WRITE: Write Completed
3064,428231] WS_WRITE: Write Completed
3064,435124] WS_WRITE: Write Completed
3064,439575] WS_WRITE: Write Completed
3064,446388] WS_WRITE: Write Completed
3064,452124] WS_WRITE: Write Completed
3064,456571] WS_WRITE: Write Completed
3064,463448] WS_WRITE: Write Completed
3064,467881] WS_WRITE: Write Completed
3064,474793] WS_WRITE: Write Completed
3064,479142] WS_WRITE: Write Completed
3064,485978] WS_WRITE: Write Completed
3064,491697] WS_WRITE: Write Completed

```

The LED Ring looks as follows for the color inputs chosen by me.

