Leonardo Fusser (1946995) December 4th, 2020

Embedded Linux 247-305VA
Fall 2020
Analog Output Assignment
Manijeh Khataie

1. What is PWM and how does it work?

   - When used for motor control, PWM (pulse width modulation) works by driving a motor with a series of "ON – OFF" pulses and varying the duty cycle (the fraction of time that the output voltage is "ON" compared to when it is "OFF") of the pulses while keeping the same frequency throughout.
     -https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html#:~:text=As%20its%20name%20suggests%2C%20pulse,while%20keeping%20the%20frequency%20constant.

2. What are the pin names and numbers on the BeagleBoneBlack for analog output?

   - eHRPWM0 (pwmchip0): *P9_22/P9_31*, *P9_21/P9_29* | eHRPWM1 (pwmchip2): *P9_14/P8_36*, *P9_16/P8_34* | eHRPWM2 (pwmchip5): *P8_19/P8_45*, *P8_13/P8_46* | eCAP0 (pwmchipX): *P9_42* | eCAP2 (pwmchipX): *P9_28*

3. How do you enable a PWM on an output pin?

   - To enable a PWM on one of the output pins, use the "config-pin" tool and choose the "pwm" setting. Example (below is assuming logged in as debian and working out of home directory):

     config-pin -a pin_number (like p1.08) pwm
     config-pin -q pin_number (like p1.08) -> shows mode set to

4. For an analog input pin, how can we see if it is available? If it isn't, how can we make it available?

   - If the analog input pin isn't available, then you must free it (un-export the pin) in order to make it available again.

5. How can you control a GPIO pin?

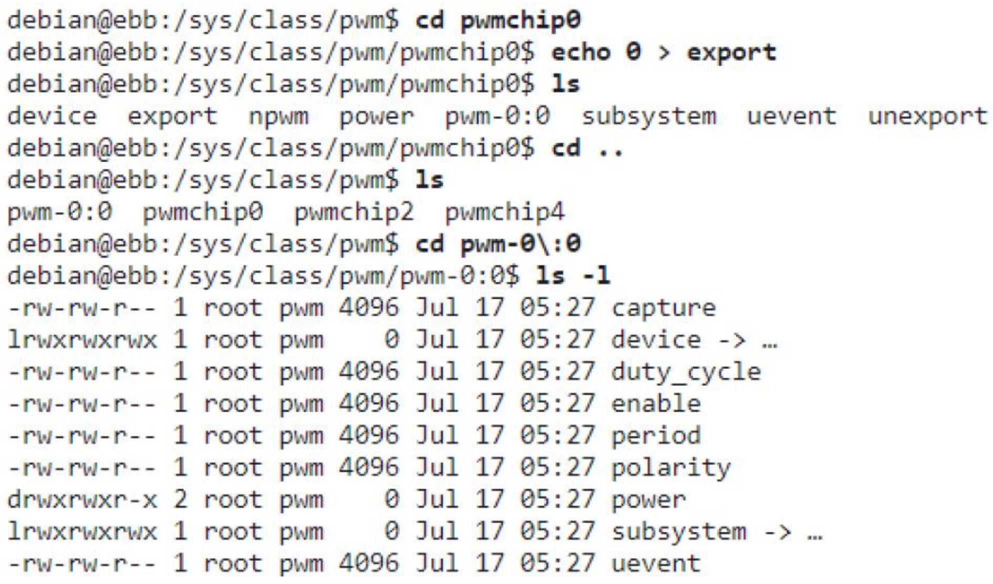
   - To control one of the GPIO pins, first navigate to the "/sys/class" folder on the BBB. Then make sure you reserve a GPIO (export), set a direction for the GPIO (input or output), read or write a value to the GPIO and free the GPIO when finished (un-export). You must have super-user privileges in order to these commands.

6. What is the name of the device for PWM and where is it located on the file system?

   - The names for the PWM can be found under the "/sys/class/pwm" folder on the BBB. The names that can be found on the BBB for PWM are pwmchip0, pwmchip2, pwmchip3, pwmchip5 and pwmchip6.

7. How can you control a PWM pin?

   - To control one of the PWM pins, you must configure and use the correct "pwmchip" device. Here is an example on how to do so:



```
debian@ebb:/sys/class/pwm$ cd pwmchip0
debian@ebb:/sys/class/pwm/pwmchip0$ echo 0 > export
debian@ebb:/sys/class/pwm/pwmchip0$ ls
device  export  npwm  power  pwm-0:0  subsystem  uevent  unexport
debian@ebb:/sys/class/pwm/pwmchip0$ cd ..
debian@ebb:/sys/class/pwm$ ls
pwm-0:0  pwmchip0  pwmchip2  pwmchip4
debian@ebb:/sys/class/pwm$ cd pwm-0\:0
debian@ebb:/sys/class/pwm/pwm-0:0$ ls -l
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 capture
lrwxrwxrwx 1 root pwm    0 Jul 17 05:27 device -> …
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 duty_cycle
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 enable
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 period
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 polarity
drwxrwxr-x 2 root pwm    0 Jul 17 05:27 power
lrwxrwxrwx 1 root pwm    0 Jul 17 05:27 subsystem -> …
-rw-rw-r-- 1 root pwm 4096 Jul 17 05:27 uevent
```

8. Write the commands to make a 1uS pulse with 10% duty-cycle on pwmchip0. Explain.

   - First, we must find the period. To do so, we use this formula: DC = (PW / T) * 100%… 10% = (1uS /T) * 100%… T = 10uS. After finding the period, we have to find the amount of time the signal will remain "ON" (this is also the pulse width). We can figure out the duration it will be "ON" by using this formula: (10uS/100) * 10% = 1uS will be "ON". We do this because we cannot enter the % duty cycle on the BBB. After figuring out these two values, we can use these following commands in order to create our signal (below is assuming that you already configured the "pwmchip0" device like in Q7, working out of the directory /sys/class/pwm/pwm-0:0 and logged in as debian):
     **Values below are in nanoseconds*

     sudo sh -c "echo 10000 > period"
     sudo sh -c "echo 1000 > duty_cycle"
     sudo sh -c "echo 1 > enable"

9. If we use a DMM (digital multimeter) between pin PWM0 and ground, what is the value that DC voltmeter shows (3.3V Vref)? *10% duty-cycle like in Q8.

    - The value the voltmeter should read is around 331.4 mV. -> (3.314V/100) * 10%.

10. Change the duty-cycle to 50%. Now what will be the value?

    - The value the voltmeter should read is around 1.657V. -> (3.314V/100) * 50%.

11. Explain how a servo works and write a command so that a servo goes to a +45-degree position.

    - A servo motor consists of a DC motor that is attached to a potentiometer and a control circuit. The position of the motor shaft can be controlled by sending a PWM signal to the servo. The HS-422 servo motor expects a pulse every 20 mS (period – T). To make the servo motor go to a +45-degree position, the following commands must be used (below is assuming that you already configured the "pwmchip0" device like in Q7, working out of the directory /sys/class/pwm/pwm-0:0 and logged in as debian):
      **Values below are in nanoseconds*

      ```
      sudo sh -c "echo normal > polarity"
      sudo sh -c "echo 20000000 > period"
      sudo sh -c "echo 1 > enable
      sudo sh -c "echo 1900000 > duty_cycle"
      sudo sh -c "echo 0 > enable"
      ```