Device Driver\_양영식

NXSOL-OJT-2024

Exported on 2024-02-20 13:27:09

Table of Contents

1 Introduction 3

1.1 Implemented Features 3

2 Target GPIO pins 4

3 Used existing APIs in drivers/gpio/gpio-uclass.c 5

4 Memory location of GPIOa values 6

5 Add the new source 7

6 Manual 8

7 Test 9

7.1 Test result 13

7.2 LEDs  successfully turned on and off before and after each GPIO output command 14

# Introduction

* Final assignment of u-boot training.
* Training period: January 30, 2024, to February 19, 2024.
* Board: BTC08(s5p6818)

Utilizing the GPIO device driver implemented in gpio-uclass.c, I have developed features as specified in the assignment requirements and tested them using u-boot commands.

## Implemented Features

* GPIO input/output (GPIOA, GPIOB, GPIOC, GPIOD, GPIOE)
* LED control (AP\_LED0, AP\_LED1)
* Button detection (AP\_KEY0)
* Test code

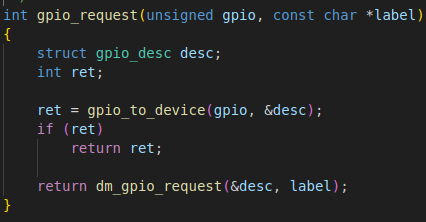
# Target GPIO pins

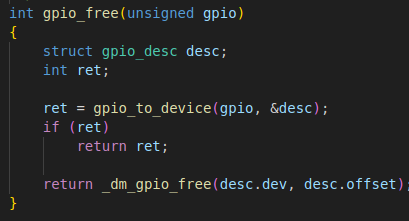
1 button, 2 LEDs

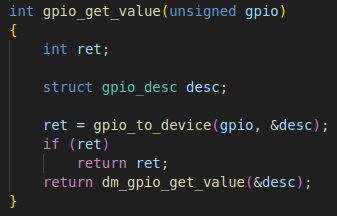


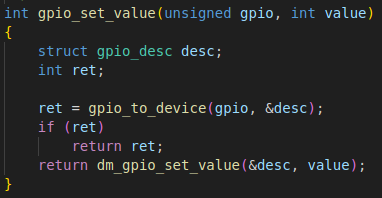


# Used existing APIs in drivers/gpio/gpio-uclass.c



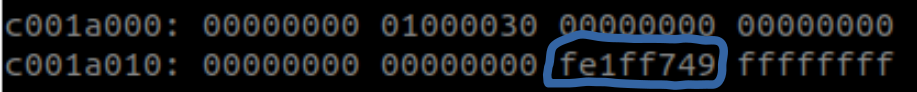






# Memory location of GPIOa values

|  |
| --- |
| md 0xc001a000 |



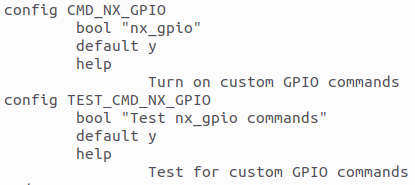
0xc001a018

# Add the new source

common/Makefile

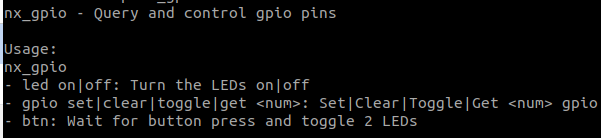


common/kconfig



And then build

# Manual



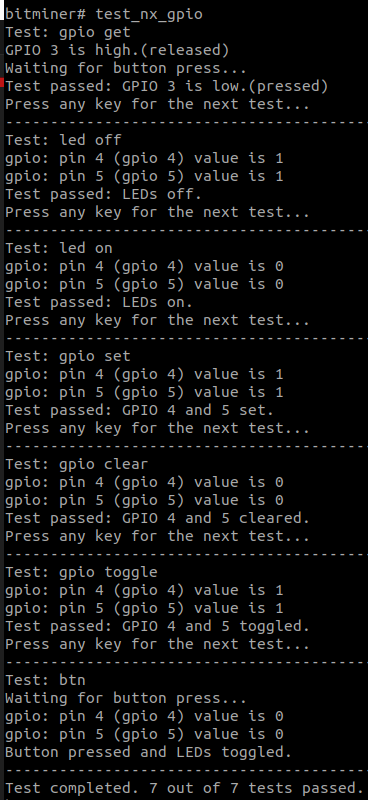
# Test

|  |
| --- |
| #include <common.h>  #include <command.h>    **static** **void** gpio\_get(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: gpio get\n");  \*total\_tests += 1;  /\* Given: button connected to GPIO 3 is released \*/  /\* When: check the value of GPIO 3 \*/  gpio\_request(3, "cmd\_nx\_gpio");  char gpio\_value = gpio\_get\_value(3);  /\* Then: GPIO 3 is high \*/  **if** (gpio\_value == 1)  printf("GPIO 3 is high.(released)\n");  **else** {  printf("Test failed: GPIO 3 is not high.\n");  **return**;  }  printf("Waiting for button press...\n");  /\* Given: button connected to GPIO 3 is released \*/  /\* When: button is pressed \*/  **while** (1) {  gpio\_value = gpio\_get\_value(3);  **if** (gpio\_value == 0)  **break**;  }  /\* Then: GPIO 3 is low \*/  **if** (gpio\_get\_value(3) == 0) {  printf("Test passed: GPIO 3 is low.(pressed)\n");  ++\*passed\_tests;  } **else**  printf("Test failed: GPIO 3 is not low.\n");  gpio\_free(3);  }    **static** **void** led\_off(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: led off\n");  \*total\_tests += 1;  /\* Given: LEDs are on \*/  /\* When: turn off the LEDs \*/  run\_command("nx\_gpio led off", 0);  /\* Then: GPIO 4 and 5 are high \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio\_get\_value(4) == 1 && gpio\_get\_value(5) == 1) {  printf("Test passed: LEDs off.\n");  ++\*passed\_tests;  } **else**  printf("Test failed: LEDs are not off.\n");  gpios\_free(4, 5);  }    **static** **void** led\_on(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: led on\n");  \*total\_tests += 1;  /\* Given: LEDs are off \*/  /\* When: turn on the LEDs \*/  run\_command("nx\_gpio led on", 0);  /\* Then: GPIO 4 and 5 are low \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio\_get\_value(4) == 0 && gpio\_get\_value(5) == 0) {  printf("Test passed: LEDs on.\n");  ++\*passed\_tests;  } **else**  printf("Test failed: LEDs are not on.\n");  gpios\_free(4, 5);  }    **static** **void** gpio\_set(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: gpio set\n");  \*total\_tests += 1;  /\* Given: LEDs are on \*/  /\* When: setting GPIO 4 and 5 \*/  run\_command("nx\_gpio gpio set 4", 0);  run\_command("nx\_gpio gpio set 5", 0);  /\* Then: LEDs are off \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio\_get\_value(4) == 1 && gpio\_get\_value(5) == 1) {  printf("Test passed: GPIO 4 and 5 set.\n");  ++\*passed\_tests;  } **else**  printf("Test failed: GPIO 4 and 5 are not set.\n");  gpios\_free(4, 5);  }    **static** **void** gpio\_clear(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: gpio clear\n");  \*total\_tests += 1;  /\* Given: LEDs are off \*/  /\* When: clear GPIO 4 and 5 \*/  run\_command("nx\_gpio gpio clear 4", 0);  run\_command("nx\_gpio gpio clear 5", 0);  /\* Then: LEDs are on \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio\_get\_value(4) == 0 && gpio\_get\_value(5) == 0) {  printf("Test passed: GPIO 4 and 5 cleared.\n");  ++\*passed\_tests;  } **else**  printf("Test failed: GPIO 4 and 5 are not cleared.\n");  gpios\_free(4, 5);  }    **static** **void** gpio\_toggle(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: gpio toggle\n");  \*total\_tests += 1;  /\* Given: initial state of GPIO 4 and 5 \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  char gpio4\_value = gpio\_get\_value(4);  char gpio5\_value = gpio\_get\_value(5);  /\* When: toggle GPIO 4 and 5 \*/  run\_command("nx\_gpio gpio toggle 4", 0);  run\_command("nx\_gpio gpio toggle 5", 0);  /\* Then: GPIO 4 and 5 are toggled \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio4\_value != gpio\_get\_value(4) && gpio5\_value != gpio\_get\_value(5)) {  printf("Test passed: GPIO 4 and 5 toggled.\n");  ++\*passed\_tests;  } **else**  printf("Test failed: GPIO 4 and 5 are not toggled.\n");  gpios\_free(4, 5);  }    **static** **void** btn(unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  printf("Test: btn\n");  \*total\_tests += 1;  /\* Given: initial state of GPIO 4 and 5 / the button connected to GPIO 3 is released \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  char gpio4\_value = gpio\_get\_value(4);  char gpio5\_value = gpio\_get\_value(5);  /\* When: wait for button to be pressed \*/  run\_command("nx\_gpio btn", 0);  /\* Then: LEDs are toggled \*/  gpio\_request(4, "cmd\_nx\_gpio");  gpio\_request(5, "cmd\_nx\_gpio");  **if** (gpio4\_value != gpio\_get\_value(4) && gpio5\_value != gpio\_get\_value(5)) {  printf("Button pressed and LEDs toggled.\n");  ++\*passed\_tests;  } **else**  printf("Button press not recognized or LEDs not toggled.\n");  gpios\_free(4, 5);  }    **static** **void** run\_test(**void** (\*test)(unsigned int \*, unsigned int \*), unsigned int \*total\_tests, unsigned int \*passed\_tests)  {  test(total\_tests, passed\_tests);  printf("Press any key for the next test...\n------------------------------------------------\n");  **while** (!tstc());  getc();  }    **void** do\_test\_nx\_gpio(**void**)  {  unsigned int total\_tests = 0;  unsigned int passed\_tests = 0;  run\_test(gpio\_get, &total\_tests, &passed\_tests);  **if** (passed\_tests == 0) /\* Other tests depend on this one \*/  **return**;  run\_test(led\_off, &total\_tests, &passed\_tests);  run\_test(led\_on, &total\_tests, &passed\_tests);  run\_test(gpio\_set, &total\_tests, &passed\_tests);  run\_test(gpio\_clear, &total\_tests, &passed\_tests);  run\_test(gpio\_toggle, &total\_tests, &passed\_tests);  btn(&total\_tests, &passed\_tests);  printf("------------------------------------------------\nTest completed. %u out of %u tests passed.\n", passed\_tests, total\_tests);  }    U\_BOOT\_CMD(test\_nx\_gpio, 1, 0, do\_test\_nx\_gpio,  "Test cmd\_nx\_gpio",  "\nRun tests for cmd\_nx\_gpio\n"  ", which include gpio get, led on|off, gpio set|clear|toggle, and btn\n"  ); |

Code Block 1 test code

## Test result

|  |
| --- |
| $ test\_nx\_gpio |



## LEDs  successfully turned on and off before and after each GPIO output command

