Homework 3 Linux I/O

B06901160 翁挺瑋

1. What is Linux IIO subsystem?

A: The Industrial I/O subsystem is intended to provide support for devices that in some sense are analog to digital or digital to analog converters (ADCs, DACs).

Ex. ADCs, Gyros, Pressure sensors.

1. How is the efficiency compared between interrupt I/O and polling I/O?

A:Interrupt is inefficient when devices constantly interrupt the CPU. Polling is inefficient when CPU does not find a device for pairing.

1. pi\_2\_mmio.h

The pointer operation was to set array offset by seven and ten respectively

1. #ifndef PI\_2\_MMIO\_H
2. #define PI\_2\_MMIO\_H
4. #include <stdint.h>
6. #define MMIO\_SUCCESS 0
7. #define MMIO\_ERROR\_DEVMEM -1
8. #define MMIO\_ERROR\_MMAP -2
9. #define MMIO\_ERROR\_OFFSET -3
11. **extern** **volatile** uint32\_t\* pi\_2\_mmio\_gpio;
13. **int** pi\_2\_mmio\_init(**void**);
15. **static** **inline** **void** pi\_2\_mmio\_set\_input(**const** **int** gpio\_number) {
16. // Set GPIO register to 000 for specified GPIO number.
17. \*(pi\_2\_mmio\_gpio+((gpio\_number)/10)) &= ~(7<<(((gpio\_number)%10)\*3));
18. }
20. **static** **inline** **void** pi\_2\_mmio\_set\_output(**const** **int** gpio\_number) {
21. // First set to 000 using input function.
22. pi\_2\_mmio\_set\_input(gpio\_number);
23. // Next set bit 0 to 1 to set output.
24. \*(pi\_2\_mmio\_gpio+((gpio\_number)/10)) |=  (1<<(((gpio\_number)%10)\*3));
25. }
27. **static** **inline** **void** pi\_2\_mmio\_set\_high(**const** **int** gpio\_number) {
28. \*(pi\_2\_mmio\_gpio+7) = 1 << gpio\_number;
29. }
31. **static** **inline** **void** pi\_2\_mmio\_set\_low(**const** **int** gpio\_number) {
32. \*(pi\_2\_mmio\_gpio+10) = 1 << gpio\_number;
33. }
35. **static** **inline** uint32\_t pi\_2\_mmio\_input(**const** **int** gpio\_number) {
36. **return** \*(pi\_2\_mmio\_gpio+13) & (1 << gpio\_number);
37. }
39. #endif

4.

