**CG2271 Realtime Operating Systems**

**Lab 6 – Analog to Digital Conversion**

**Answer Book**

|  |  |
| --- | --- |
| **Name** | **Student ID (Axxxxx)** |
| John Woo Yi Kai | A0272561E |
| Rishi Moorthy | A0273618X |
| Wang Yanjie | A0276202M |
| Kenneth Wong Cun Wi | A0303203A |
| Kuek Yeau Hao Jonathan | A0258485M |

**Question 1.** (4 MARKS)

Given ref voltage and ADC resolution bits, ADC accuracy is given by

(5d.p.)

Similarly for ADC resolution bits and the same

(5d.p.)

**Question 2.** (2 MARKS)

Advantage: Higher precision where 12-bit can detect much smaller voltage changes (0.00081V compared to 0.01289V), enabling better measurement detail. Specifically, with 12 bits, the ADC can divide the input voltage range into quantisation levels, compared to only levels for an 8-bit ADC. This means each digital code represents a smaller voltage step, giving higher measurement accuracy and finer encoding of the analog signal.

Disadvantage: From Table 23-7, the base conversion time for a 12-bit single-ended conversion is 20 ADCK cycles, compared to 17 ADCK cycles for 8-bit mode. Therefore, the 12-bit ADC takes longer per conversion, resulting in slower sampling rates.

**Question 3.** (4 MARKS)

According to the ADCx\_SC3 field descriptions on page 370 under AVGS, the ADC can average over 4, 8, 16, or 32 samples if hardware averaging is enabled.

Advantage of averaging is to reduce random noises by smoothing those small fluctuations in the input signal. This provides a more stable and accurate reading.

Disadvantage of averaging is increasing conversion time, since multiple samples need to be taken before averaging, reducing the overall sampling rate.

**Question 4.** (3 MARKS)

The best place to turn the continuous conversion back on is inside the ADC interrupt handler after reading the previous result. When continuous mode is OFF (ADCO = 0), each ADC conversion produces one result. When the conversion completes, the COCO flag is set, and the ADC interrupt handler (ADC0\_IRQHandler()) runs. After reading the result, the ADC stops until software starts the next conversion by writing to SC1[0] again.

Thus, if we want to continue sampling as quickly as possible, it would be inside the ADC0\_IRQHandler() function after reading the ADC result. As soon as one conversion finishes and the interrupt fires, the handler reads the result and triggers the next conversion, achieving continuous conversion of samples as quickly as possible.

**Question 5.** (2 MARKS)

PTE20: ADC0\_SE0

PTE21: ADC0\_SE4a

**A close up of a document

AI-generated content may be incorrect.**

**Question 6.** (2 MARKS)

turn is declared as static so that it retains its value between function calls to ADC0\_IRQHandler(). With static, the variable lives for the lifetime of the program but is only accessible inside the function. This allows the interrupt handler to remember which ADC channel to convert next.

**Report Total: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 17**

**Demo Total: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ 3**

**Total: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 20**