Errata of

Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C

Third Edition

First Printing (July 2017)

ISBN-10: 0982692668

Yifeng Zhu

Correction Date: June 17, 2020

Thank you all for providing me feedbacks and corrections!

Chapter 1. See a Program Running

Chapter 2. Data Representation

- Page 31, paragraph beneath heading "2.4 Signed Integers", One common characteristic of these numeral systems is that the most significant bit (also called the ~~rightmost~~ leftmost bit)…

Chapter 3. ARM Instruction Set Architecture

Chapter 4. Arithmetic and Logic

- Page 77,

```
      ADDS ~~r2~~, r2, r3   ; r1 = r2 + r3, and update N, Z, C, and V flags
```

should be

```
      ADDS r1, r2, r3   ; r1 = r2 + r3, and update N, Z, C, and V flags
```

- Page 86, the title of Example 4-2, "The mask selects bit ~~0,~~ 2, 4, and 5." should be "The mask selects bit 2, 4, and 5."

Chapter 5. Load and Store

Chapter 6. Branch and Conditional Execution

- Page 112, Table 6-2, The logic implementation of GT should be

$$\bar{Z}(NV + \bar{N}\bar{V})$$

Chapter 7. Structured Programming

Chapter 8. Subroutines

- Page 162, "set LR ~~is set~~ to PC_1 + 4"

Chapter 9. 64-bit Data Processing

Chapter 10. Mixing C and Assembly

Chapter 11. Interrupt

- Page 253, "all interrupts with a priority value ~~lower~~ larger than or equal to BASEPRI are disabled."

- Page 253

CPSIE i	Enable interrupts and configurable fault handlers	MOVS r0, ~~#1~~ #0 MSR PRIMASK, r0
CPSIE f	Enable interrupts and fault handlers	MOVS r0, ~~#0~~ #1 MSR FAULTMASK, r0

- Page 264 and 265, Example 11-13

```
EXTI->RTSR |= EXTI_RTSR_TR3;  should be EXTI->RTSR |= EXTI_RTSR_RT3
EXTI->FTSR &= ~EXTI_FTSR_TR3; should be EXTI->FTSR &= ~EXTI_FTSR_RT3;
```

Chapter 12. Fixed-point and Floating-point Arithmetic

- Page 292, Example 12-5, The software-based multiplication example should preserve LR.

```
area_of_rectangle PROC

  PUSH {LR}
  ; area = length * width
  ; call software library

  BL __aeabi_fmul

  POP {PC}  ; return area in r0

  ENDP
```

Chapter 13. Instruction Encoding and Decoding

Chapter 14. Generic-purpose I/O

- Page 342, the pictures in Figure 14.1 and Figure 14.2 should be swamped.
- Page 358, Example 14-1,

```
GPIOB->ODR |= 1UL<<~~6~~;
```

should be

```
GPIOB->ODR |= 1UL<<2;
```

Chapter 15. General-purpose Timers

- Page 379, at the bottom, removing "driving the timer is 2.097 MHz."
- Page 381, in the code of Example 15-1:

```
TIM1->CCMR1 &= ~TIM_CCER_CC1NP; // select active high
```

should be

```
TIM1->CCER &= ~TIM_CCER_CC1NP; // select active high
```

- Page 396, Example 15-4

```
TIM1->CCMR1 &= ~TIM_CCER_CC1NP; // select active high
```

should be

```
TIM1->CCER &= ~TIM_CCER_CC1NP; // select active high
```

- Page 391, 4th paragraph, the prescaler factor is set as ~~63~~ 39
- Page 393, flow chart, step 5 of configure Timer
  "Enable TIM**2**_ARR" should be "Enable TIM**1**_ARR".
- Page 403,

```
RCC->APB1ENR |= RCC_APB1ENR_TIM4EN;
```

should be

```
RCC->APB1ENR1 |= RCC_APB1ENR1_TIM4EN;
```

- On page 406, "Figure ~~15-17~~ **15-27** shows an example time diagram of measuring the pulse width. It is assumed that no filtering has been applied to the input signal (TI1)".

Chapter 16. Stepper Motor Control

Chapter 17. Liquid-crystal Display (LCD)

Chapter 18. Real-time Clock (RTC)

- Page 459, Example 18-2

```
RTC->DR = 1U<20 | 6U<<16 | 0U<<12 | 5U<<8 | 2U<<4 | 7U;
should be
RTC->DR = 1U<<20 | 6U<<16 | 0U<<12 | 5U<<8 | 2U<<4 | 7U;
```

Chapter 19. Direct Memory Access (DMA)

Chapter 20. Analog-to-Digital Converter

- Page 489, Second to last line of text reading "ADC interrupt handler or the ~~DAM~~ DMA controller..."

- Page 486:

$$V = \frac{Digital\ Value}{2^n - 1} \times V_{REF}$$

should be

$$V = \frac{Digital\ Value}{2^n} \times V_{REF}$$

- Page 496

$$ADC\ Result = \frac{V_{input}}{V_{REF}} \times 4095$$

should be

$$ADC\ Result = \frac{V_{input}}{V_{REF}} \times 4096$$

- Page 496

$$V_{input} = \frac{ADC\ Result}{4095} \times V_{REF}$$

should be

$$V_{input} = \frac{ADC\ Result}{4096} \times V_{REF}$$

- Page 498, in the Initialization ADC 1 section there is a typo in step 3:
  3. Enable I/O analog switch booster (SYSCFG_CFGR1_BOOSTEN) in register ~~ADC123_COMMON->CCR~~ SYSCFG_CFGR1.
- Page 498, "Therefore, software needs to ~~wait~~ wake up ADC"

- Page 508

$$DAC_{output} = V_{ref} \times \frac{DOR}{4095}$$

should be

$$DAC_{output} = V_{ref} \times \frac{DOR}{4096}$$

Chapter 21. Digital-to-Analog Converter

Chapter 22. Serial Communication Protocols

Chapter 23. Multitasking

Chapter 24. Digital Signal Processing