

Tutorial 7

In introducing C we've done quite a bit of theory. As such the tut is long.
However the prac is short. :-)

Question 1: (3)

- a) Explain why we may want to buffer the timer's ARR and PSC registers? (3)

Question 1: (7)

- a) C code is more abstract than assembly. What does this mean and why is it useful? (2)
b) C code is more portable than assembly. What does this mean and why is it useful? (2)
c) What does a C compiler do? (1)
d) Why are C and assembly inherently unsafe programming languages? (1)
e) Why is an unsafe language like C or assembly very suitable (even necessary) for microcontroller code development? (1)

Question 2: (5)

- a) Why do we still need (or at least prefer) to have some assembly code in our projects, in the form of a startup.s file? (2)
b) Why do we need to initialise the data segment before running C code? (1)
c) What does this process of initialising the data segment involve? (1)
d) Why might we prefer to name our data types as "int32_t" rather than "signed int"? (1)

Question 3: (7)

In the context of the line of C code:

```
*(uint16_t*)0x40001010 |= 0xAABB;
```

- a) What does the (uint16_t*) mean? (1)
b) What does the first * mean? (1)
c) What does the |= mean? (1)
d) Which individual memory addresses are modified by this line of code? What data is the contents of each of the memory address OR'd with? (4)

Question 4: (2)

Write a single line of C code which is the equivalent of:

| | |
|---------------------|---|
| LDR R0, =0x48001000 | @ cause R0 to point to 0x4800 1000 |
| LDR R1, =0x20000010 | @ cause R1 to point to 0x2000 0010 |
| LDR R2, [R1] | @ get the word which R1 points to into R2 |
| STR R2, [R0] | @ write word which R1 points to to address pointed to by R0 |

Question 5: (6)

- a) What are statically allocated variables? Where do they get allocated in memory? (2)
b) What are automatically allocated variables and where do they get allocated in memory? (2)

- c) Why would we prefer to minimize usage of statically allocated variables in favour of automatically allocated variables? (1)
- d) In our C code we should initialise automatic variables but it's not necessary to do so for static variables. Why is this? (1)

Bonus: (3)

Lines 84 to 102 of the startup file initialise two segment: data and bss

What is the difference between these segments?

What makes the initialisation of the bss segment a lot more efficient than the data segment in terms of flash memory footprint?

Marking:

Marked out of: 30

Available marks: 33