Homework IV **Benjamin Streit**
Professor Gary J. Minden EECS 388
Date Assigned: 10/18/18 Date Due: 10/25/18

1. Which subroutine is used to define a task?

> The following subroutine is used to define a task:
>
> ```
> xTaskCreate();
> ```
>
> In addition, one can call the following to define a task:
>
> ```
> xTaskCreateStatic();
> ```
>
> The only difference between the two subroutines is RAM being automatically allocated from the FreeRTOS heap or being statically allocated at compile time.
>
> Note: The parameters for each subroutine call have been omitted.

2. Explain the difference between vTaskDelay and vTaskDelayUntil.

> `vTaskDelay` causes the calling task to enter the `Blocked` state and remain so for the specified number of ticks from the time of calling. Meaning the time at which a given calling task exists the `Blocked` state is relative to the time that `vTaskDelay` was called.
>
> `vTaskDelayUntil`, on the other hand, causes the calling task to enter the `Blocked` state and remain so until an *absolute* time has been reached. Meaning the time at which a given calling task exists the `Blocked` state is exactly the given time, not a time relative to the time that `vTaskDelayUntil` was called.
>
> With these explanations in mind, it is clear that the difference between `vTaskDelay` and `vTaskDelayUntil` is the former's exiting of the `Blocked` state is relative to the time at which the call was made, while the latter's is not.

3. Which subroutine is used to start the FreeRTOS scheduler?

> The following subroutine is used to start the FreeRTOS scheduler:
>
> ```
> vTaskStartScheduler();
> ```

4. When you create a task, you assign it a priority. Let's say from 0 to 7. Which value is the highest priority?

> When creating a task, priorities can be assigned values between `0`, being the lowest, and `configMAX_PRIORITIES - 1`, being the highest. Assuming `configMAX_PRIORITIES` is 8, the highest priority would then be 7.

5. How many bits does the TM4C1294 ADC generate for a conversion?

> The TM4C1294 ADC generates 12 bits for an analog to digital conversion.

6. How long does it take for the TM4C1294 ADC to convert an analog voltage to a digital number? (Assume a 16 MHz FADC clock.)

> With a 16 MHz FADC clock, the TM4C1294 ADC requires 1 microsecond to convert an analog voltage to a digital number.
>
> Note: This value includes the sampling time for the ADC.

7. If the input range of a 12-bit ADC is from 0.0 VDC to 1.0 VDC and after conversion you read the hexadecimal value 0x300, what voltage does this represent?

With 12-bits and a maximum of 1.0 VDC, we compute the number of steps:

$$\texttt{steps} = 2^{12}$$
$$\texttt{steps} = 4096$$

Next, we compute the voltage for each step:

$$\texttt{step\_voltage} = \frac{1.0 \text{ VDC}}{\texttt{steps}}$$
$$\texttt{step\_voltage} = 0.000244140625 \text{ V}$$

Now, we compute the voltage for hexadecimal value 0x300, or 768 in decimal:

$$\texttt{voltage} = \texttt{step\_voltage} * 768$$
$$\texttt{voltage} = 0.1875 \text{ V}$$

It is clear that the converted value of 0x300 represents a voltage of 0.1875 V.

8. What is the base address of the TM4C1294 ADC0?

The base address of the TM4C1294 ADC0 is 0x4003.8000.

9. Which DriverLib subroutine would you use to read a value from the TM4C1294 ADC0?

The following subroutine is used to read a value from the TM4C1294 ADC0:

```
ADCSequenceDataGet(ADC0_BASE, 0, &ulValue);
```

10. A sensor measures a physical temperature. The relationship between temperature and sensor voltage is:

$$\text{Vs(T)} := 0.03 * \text{T} + 0.25$$

Where T is the temperature in degrees Celsius and Vs is the sensor voltage. If the temperature is 50 °C, what is the sensor voltage?

Using the supplied relationship between temperature and sensor voltage:

$$\text{Vs(T)} := 0.03 * 50^\circ \text{ C} + 0.25$$
$$\text{Vs(T)} := 1.75 \text{ V}$$

It is clear that with the given relationship and a temperature of 50° C, the sensor voltage is 1.75 V.

11. The TM4C1294 ADC has Vlow = 0.0 V and a Vhigh =3.3 V. The ADC is 12-bits. If Vin = 2.75 V, and an ADC conversion completed, what is the value read from the ADC?

With 12-bits and a Vhigh of 3.3 V, we compute the number of steps:

$$\texttt{steps} = 2^{12}$$
$$\texttt{steps} = 4096$$

Next, we compute the voltage for each step:

$$\texttt{step\_voltage} = \frac{3.3 \text{ VDC}}{\texttt{steps}}$$
$$\texttt{step\_voltage} = 0.0008056640625 \text{ V}$$

Now, we compute the hexadecimal value for a voltage of 2.75 V:

$$value_{10} = \frac{\texttt{Vin}}{\texttt{step\_voltage}}$$
$$value_{10} = 3413$$

$$value_{16} = D55$$

It is clear that the value read from the ADC will be 0xD55.

12. Given the temperature/voltage relationship in problem 10 and the ADC parameters in problem 11, if the value read from the ADC is 0x0735, what temperature does this represent?

With 12-bits and a maximum of 3.3 VDC, we compute the number of steps:

$$\texttt{steps} = 2^{12}$$
$$\texttt{steps} = 4096$$

Next, we compute the voltage for each step:

$$\texttt{step\_voltage} = \frac{3.3 \text{ VDC}}{\texttt{steps}}$$
$$\texttt{step\_voltage} = 0.0008056640625 \text{ V}$$

Now, we compute the voltage for hexadecimal value 0x0735, or 1845 in decimal:

$$\texttt{voltage} = \texttt{step\_voltage} * 1845$$
$$\texttt{voltage} = 1.4864501953 \text{ V}$$

Finally, we manipulate the equation from problem 10 to solve for temperature:

$$\text{V} := 0.03 * \text{T} + 0.25$$

$$\text{T} = \frac{\text{V-0.25}}{0.03}$$

$$\text{T} = 41.33^\circ \text{ C}$$

Thus, for the given values and expressions, a value read of 0x0735 from the ADC would represent a temperature of 41.33° C.