Peripheral configuration

Please write the program in ANSI C99 programming language.

In this exercise you are supposed to configure an Analog-Digital converter of a microcontroller based on the requirements and the given source code. All the information can be found in the attached reference manual of the microcontroller's ADC. Do not expect that the MCU just had a reset, the ADC might be already configured.

Requirements

1. Configure the Analog-Digital converter of the given microcontroller
   - 1.1. Extend the given source code to fulfill the requirements
2. Set the registers of the given ADC structure in the **`void ADCConfig(ADC_TypeDef *ADC1);`** function
   - 2.1. All the non-mentioned registers shall be left unset
     - 2.1.1. The mentioned registers' non-used bits shall be set to the reset value
   - 2.2. The ADC shall be switched on
   - 2.3. The ADC resolution shall be 10 bit
   - 2.4. The ADC shall be set to single conversion mode
   - 2.5. The ADC_CR2 configuration register's value shall not be overwritten completely
     - 2.5.1. The ADC_CR2 configuration register value shall not be changed except the sampling time selection bits
       - 2.5.1.1. The sampling time selection for the first 24 channels shall be 96 ADC clock cycles
       - 2.5.1.2. The rest of the bits shall be kept on their current value
   - 2.6. The ADC_CR3 register shall be set to reset value
   - 2.7. The ADC's DMA option for single conversion shall be disabled
   - 2.8. Only ADC channel 6 shall be used
     - 2.8.1. Other channels shall not be used
3. After the registers are set, initialize the peripheral with the **`extern void ADC_Init(ADC_TypeDef* ADCx);`** function
4. The **`void ADC_Task(ADC_TypeDef* ADC1)`** function contains the peripheral's task
   - 4.1. The AD conversion shall be started with:
     - 4.1.1. Setting the corresponding bit
     - 4.1.2. Invoking the **`extern void ADC_Start(ADC_TypeDef* ADCx);`** function
   - 4.2. The results are in the proper ADC registers and shall be read by the task
   - 4.3. The task shall provide the value of the signal with the correct properties
     - 4.3.1. The value of the signal shall be float32 type
       - 4.3.1.1. Conversion shall be made to get the float32 type from the 8-bit registers
       - 4.3.1.2. The *value* parameter of the *signal* shall be converted to voltage:
         - 4.3.1.2.1. The maximum range of the ADC is 3.6 Volt
         - 4.3.1.2.2. The maximum value of the ADC in 10 bits is $2^{10} - 1$

4.3.2. The availability of the signal depends on the value, according the following conditions:

- 4.3.2.1. If the value is below the threshold, the availability of the signal shall be TRUE
  - 4.3.2.1.1. The threshold is pre-defined in the header file
- 4.3.2.2. If the value is equal or above the threshold
  - 4.3.2.2.1. The availability of the signal shall be FALSE
  - 4.3.2.2.2. The value shall be saturated to the threshold value

Design constraints

- The function can only use the following APIs:
  - `ADC_Init(ADC_TypeDef* ADCx);`
  - `ADC_Start(ADC_TypeDef* ADCx);`
  - `ProvideMeasuredSignal(dtSignalParameters);`