



QUESTIONS

Contrôle du lundi 09/01/2023

MICROPROCESSORS 2

Dans ce document, vous trouverez d'abord les questions. In this document, you will first find the questions.

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse. Des points négatifs sont affectés aux mauvaises réponses. La pondération des mauvaises réponses est nulle pour les premières fausses réponses mais ensuite la pondération (négative) des mauvaises réponses augmente avec le nombre de mauvaises réponses.

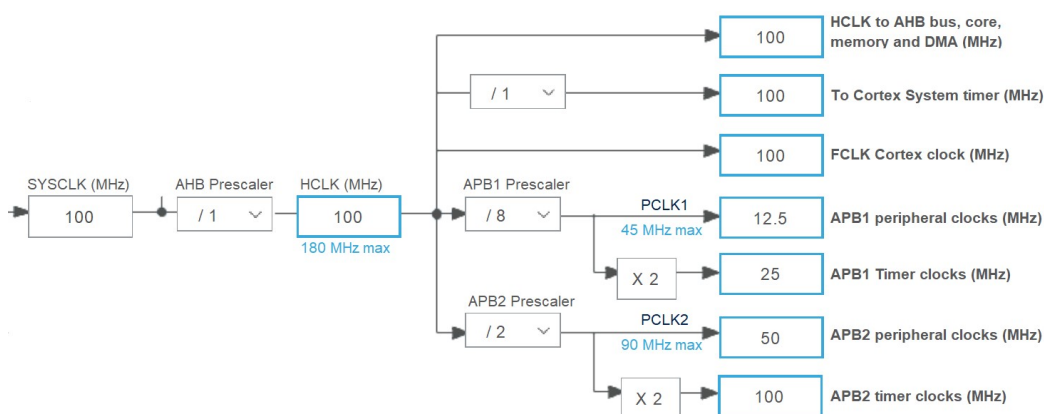
Questions with the symbol ♣ may have zero, one or more correct answers. The others have a single correct answer. Negative points are assigned for wrong answers. The weight of wrong answers is zero for the first few wrong answers, but then the (negative) weight of wrong answers increases with the number of wrong answers.

Un document ressource peut être distribué en plus de ce document. A resource document could be distributed in addition to this document.

10 points for the first 17th questions (Q1 to Q17).

10 points for the code handler (3 Handler codes to be written Q18 to Q20).

Question 1



Combien faut-il de temps pour exécuter 90 instructions en langage d'assemblage avec ce microcontrôleur si on considère que cela nécessite 100 cycles ? Si aucune des réponses ne vous paraît exacte, choisissez la valeur qui vous semble la plus proche.

How long does it take to execute 90 Assembly Language instructions with this microcontroller if it requires 100 cycles? If none of the answers seems correct choose the one which is nearest your result.

[A] 10ns

[B] 5ns

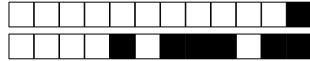
[C] 100 μ s

[D] 1ms

[E] 5ms

[F] 50ns

[G] 1 μ s

**ADC control register 2 (ADC_CR2)**

Address offset: 0x08

Reset value: 0x0000 0000

Bit 8 **DMA**: Direct memory access mode (for single ADC mode)

This bit is set and cleared by software. Refer to the DMA controller chapter for more details.

0: DMA mode disabled

1: DMA mode enabled

Bit 0 **ADON**: A/D Converter ON / OFF

This bit is set and cleared by software.

Note: 0: Disable ADC conversion and go to power down mode

1: Enable ADC

Question 2

Quelle est l'adresse du registre ADC1_CR2? What is the address of the ADC1_CR2 register ?

- ☐ A 0x40012008 ☐ B 0x40010200 ☐ C 0x40020208 ☐ D 0x00000008
☐ E 0x40002000 ☐ F 0x40020200 ☐ G 0x50000008

Question 3 Quel bit doit-on mettre à 1 pour lancer une conversion ? Donnez son numéro.
Which bit must be set to start a conversion? Give its number.

- ☐ A 8 ☐ B 5 ☐ C 0 ☐ D 1 ☐ E 2

Question 4 ♣ Quel(s) code(s) permet(tent) le mode DMA sans modifier les autres modes de l'ADC1 ? Which code(s) allow the DMA mode without changing the other modes of the ADC1?

- ☐ A ADC1->CR2 | = 8; ☐ B ADC1->CR2 = 64; ☐ C ADC1->MODER = 1;
☐ D ADC1->CR2 = 1; ☐ E ADC1->CR2 & = 128; ☐ F ADC1->CR2 | = 256;
☐ G ADC1->CR2 | = 1<8; ☐ H ADC1->CR2 | = 32; ☐ I ADC1->CR2 & = 1<7;
☐ J ADC1->CR2 & = 8; ☐ K Aucune de ces réponses n'est correcte.

Question 5 A quelle fréquence est synchronisée l'USART1? At what frequency is the USART1 synchronized?

- ☐ A 115 200 Hz ☐ B 12,5 MHz ☐ C 32 kHz ☐ D 25 MHz ☐ E 50 MHz
☐ F 100 MHz

Question 6 Quelle fréquence est envoyée pour synchroniser l'USART3? What frequency is sent to synchronize the USART3?

- ☐ A 32 kHz ☐ B 12,5 MHz ☐ C 25 MHz ☐ D 115 200 Hz ☐ E 100 MHz
☐ F 50 MHz

Question 7 Quelle fréquence est envoyée pour synchroniser le TIMER 3? What frequency is sent to synchronize the TIMER 3?

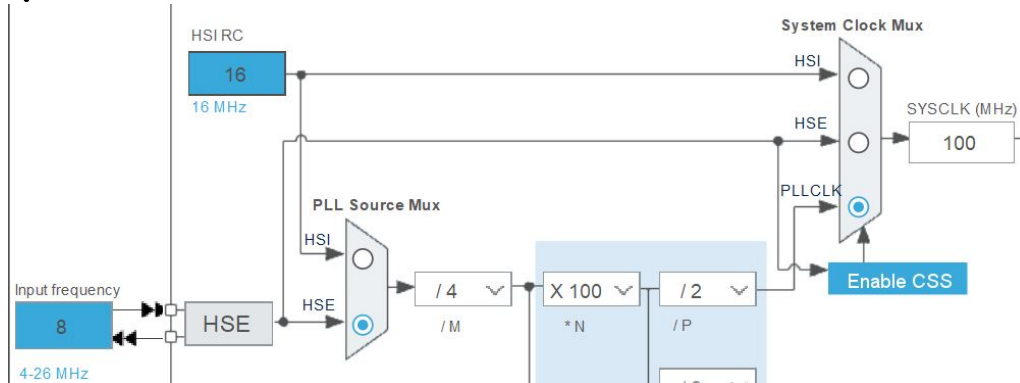
- ☐ A 25 MHz ☐ B 50 MHz ☐ C 12,5 MHz ☐ D 115 200 Hz ☐ E 32 kHz
☐ F 100 MHz

Question 8 Quelle fréquence est envoyée pour synchroniser le TIMER 8? What frequency is sent to synchronize the TIMER 8?

- ☐ A 25 MHz ☐ B 115 200 Hz ☐ C 50 MHz ☐ D 12,5 MHz ☐ E 32 kHz
☐ F 100 MHz



Question 9



Donnez les 8 LSB de PLL_CFGR (bit 0 à 7) correspondant à la configuration de la PLL. Give the 8 LSBs of PLL_CFGR (bit 0 to 7) corresponding to the configuration of the PLL.

- ☐ A 0000 0100 ☐ B 0110 0100 ☐ C 0000 0010 ☐ D 0001 0000

Question 10 Les 8 bits (bit 8 à 15) de PLL_CFGR correspondant à la configuration de la PLL sont : The 8 bits (bit 8 to 15) of PLL_CFGR corresponding to the configuration of the PLL are:

- ☐ A 0011 0010 ☐ B 0001 1001 ☐ C 0110 0100 ☐ D 0001 0000

Question 11 Les 8 bits (bit 16 à 23) de PLL_CFGR correspondant à la configuration de la PLL sont : The 8 bits (bit 16 to 23) of PLL_CFGR corresponding to the configuration of the PLL are:

- ☐ A 0100 0001 ☐ B 0010 0011 ☐ C 0100 0010 ☐ D 0010 0000
☐ E 100 0000

Question 12 Voici un code : Here is a code :

```
void TIM2_IRQHandler(void){
```

```
    gn++;
    SERIAL_SendVal(gn/10.0); //to display a value
    SERIAL_SendCharBuf(" sec");
    HAL_TIM_IRQHandler(&htim2); //Acquitement TIM 2 (erase request flag)
```

```
}
```

“gn” is defined in global variable as an “int”.

TIM2_PSC is 10 000 -1.

Quelle valeur doit-on donner à TIM2_ARR pour afficher le bon temps? What value should be given to TIM2_ARR to display the correct time?

- ☐ A 124 ☐ B 999 ☐ C 2499 ☐ D 24 ☐ E 4294967295 ☐ F 499
☐ G 9999 ☐ H 249

Question 13 On règle la vitesse de conversion du canal 5 de l’ADC1 à la vitesse minimale (480 cycles). Les autres canaux sont à la vitesse maximale. Quelle valeur doit-on lire après initialisation de l’ADC1 dans ADC1_SMPR2?

The conversion speed of the ADC1 on channel 5 is set to the minimum speed (480 cycles). Other channel use maximum speed. What value should be read after initialization of the ADC1 in ADC1_SMPR2?

- ☐ A 0x1E0 ☐ B 0x38000 ☐ C 0x7 ☐ D 0x1110000000 ☐ E 0xF00000
☐ F 0x0 ☐ G 0x5 ☐ H 0x20



Question 14 Où se situe en mémoire le vecteur de l'IT USART2? Where is located the vector of USART2 IT ?





- ☐ A 0x36 ☐ B 0x2 ☐ C 0x20 ☐ D 0xD8 ☐ E 0x54 ☐ F 0x6C
☐ G 0x26 ☐ H 0x98 ☐ I 0x4C

Question 15 L'USART2 possède une priorité de valeur 3. Une demande d'interruption (autorisée) provient de TIM6 avec une valeur de priorité de 2. Le traitement en interruption de l'USART2 peut-il être préempté?

USART2 has a priority value of 3. An interrupt request (allowed) comes from TIM6 with a priority value of 2. Can USART2 handler be preempted?

- ☐ A NO ☐ B YES

Question 16 ♣

Register	Address	Value
▼  NVIC		
>  NVIC_ISER0	0xe000e100	0x12040040
>  NVIC_ISER1	0xe000e104	0x400000
>  NVIC_ISER2	0xe000e108	0x0

D'après le contenu de NVIC dans la capture d'écran, quelles sont les interruptions qui sont autorisées? Based on the contents of NVIC in the screenshot, which interrupts are allowed?

- ☐ A SPI3 ☐ B TIM1_UP_TIM10 ☐ C TIM4 ☐ D TIM2 ☐ E TIM5
☐ F ADC ☐ G EXTI0 ☐ H EXTI9 ☐ I TIM6_DAC ☐ J EXTI2
☐ K Aucune de ces réponses n'est correcte.

Question 17 Si le NVIC autorise la demande d'interruption provenant de l'USART1 on lit alors en 0xE000E104: If the NVIC enables the interrupt request from the USART1, it is read at 0xE000E104:

- ☐ A 0x00400000 ☐ B 0x00400040 ☐ C 0x00400020 ☐ D 0x00100000
☐ E 0x00410000



Question 18 On déclare en variable globale un tableau de 10 entiers : We declare in global variable an array of 10 integers: `int g_TAB[10]={1,1,2,3,5,8,13,21,34,55};`

On a ouvert une liaison série sur UART2 afin d'envoyer ces entiers avec "`void SERIAL_SendVal(int)`". On possède aussi une variable globale `g_NTOTAL` qui indique le nombre d'éléments du tableau à envoyer sur la liaison série : "`int g_NTOTAL=5;`". We have opened a serial link on UART2 in order to send these integers with "`void SERIAL_SendVal(int)`". We also have a global variable `g_NTOTAL` which indicates the number of elements of the array to send on the serial link : "`int g_NTOTAL=5;`".

On a aussi déclaré une variable pour gérer un "état" selon vos besoins : We have also declared a variable to manage a "state" according to your needs:

```
int g_STATE=INIT;
#define INIT 0
#define SENDING 1
```

Si vous gérez d'autres états donnez-leur un nom explicite en ajoutant vos propres définitions. If you manage other states, give them an explicit name by adding your own definitions.

On veut envoyer `g_NTOTAL` éléments de ce tableau après l'appui sur un bouton (connecté à PC1). La détection du front montant s'effectue par le module EXTI et le traitement en interruption est lancé à chaque appui sur le bouton.

Attention : l'envoi des `g_NTOTAL` valeurs du tableau doit s'effectuer à raison d'une seule valeur par seconde à l'aide de la fonction "`void SERIAL_SendVal(int)`". Un nouvel appui ne sera pris en considération que si les `g_NTOTAL` éléments ont tous été envoyés. Si on appuie avant l'envoi total du tableau, l'appui est ignoré. We want to send `g_NTOTAL` elements of this array after pressing a button (connected to PC1). The rising edge is detected by the EXTI module and interrupt processing is started each time the button is pressed. Attention: the `g_NTOTAL` values of the array must be sent at a rate of one value per second using the "`void SERIAL_SendVal(int)`" function. A new press will only be taken into account if all the `g_NTOTAL` elements have been sent. If you press the button before the whole array has been sent, the press is ignored.

Le Timer 3 et le NVIC ont été initialisés de manière à déclencher `TIM3_IRQHandler` une fois par seconde. Timer 3 and NVIC have been configured in order to trigger `TIM3_IRQHandler` once a second.

La structure de votre code est la suivante (tout est dans le même fichier pour faciliter l'écriture): The global structure of your code is the following one (all the code is here in the same file to ease writing):

```
// Declaration of global variables
int g_TAB[10]={1,1,2,3,5,8,13,21,34,55};
unsigned char g_rec;
int g_NTOTAL=5, g_N=0;
...
int main(){
    STM32_Init();
    HAL_UART_Receive_IT(&huart2, &g_rec, 1);
    while(1){};
}
void EXTI1_IRQHandler(void){
    ... Code to be written
    HAL_GPIO_EXTI_IRQHandler(BUTTON_Pin);
}
void USART2_IRQHandler(void){
    ... Code to be written
    HAL_UART_IRQHandler(&huart2);
}
void TIM3_IRQHandler(void){
    ... Code to be written
    HAL_TIM_IRQHandler(&htim6);
}
```

Ecrivez le code de `TIM3_IRQHandler`. Write the code of `TIM3_IRQHandler`.



Question 19 Le Port PC1 et le NVIC ont été initialisés de manière à déclencher EXTI1_IRQHandler à chaque appui sur le bouton PC1.

PC1 GPIO Port and NVIC have been configured in order to trigger EXTI1_IRQHandler at each push on PC1 button.

Ecrivez le code de EXTI1_IRQHandler. Write the code of EXTI1_IRQHandler.

Question 20 On veut ajouter la fonctionnalité qui permet de modifier le nombre total de variables à envoyer (g_NTOTAL) de la manière suivante.

Si on reçoit le caractère '+' alors on enverra une valeur de plus du tableau (en limitant à 10) au prochain envoi.

Si on reçoit le caractère '-' alors on enverra une valeur de moins du tableau (en limitant à 10).

We want to add the functionality which allows us to modify the total number of variables to be sent (g_NTOTAL) in the following way:

If the character '+' is received, then one more value will be sent from the array (limited to 10) the next time it is sent.

If the character '-' is received, then one less value will be sent from the array (limited to 10).

On veut pouvoir arrêter (Stop!) de manière prioritaire l'envoi des données par l'envoi du caractère 'S' ou 's'. On devra pouvoir recommencer un nouvel envoi par l'appui sur le bouton.

On a initialisé la liaison série de manière à déclencher USART2_IRQHandler chaque fois qu'un nouveau caractère est reçu. Vous pouvez donc utiliser dans USART2_IRQHandler: HAL_UART_Receive_IT(&huart2, &g_rec, 1); pour lancer l'écoute du prochain caractère. Le caractère reçu est dans g_rec (attention vous ne devez lire qu'un seul caractère à la fois dans le Handler).

Il faut déclarer pour cela g_rec en variable globale : unsigned char g_rec; We want to be able to stop (Stop!) the sending of data as a priority by sending the character 'S' or 's'. It should be possible to start sending again by pressing the button.

The serial link has been initialized in such a way as to trigger USART2_IRQHandler each time a new character is received. You can therefore use in USART2_IRQHandler: HAL_UART_Receive_IT(&huart2, &g_rec, 1); to launch again the handler for the next character. The received character is available in g_rec (be careful you can only read one character at a time in the Handler).

unsigned char g_rec; is a global variable.

La modification du nombre de valeurs à envoyer ne peuvent pas intervenir pendant un envoi du tableau. The modification of the number of values to be sent cannot be done while the array is being sent.

La fonction STM32_Init contient toutes les initialisations nécessaires aux déclenchements des interruptions comme indiqué ci-dessus.

The STM32_Init function contains all the initializations necessary for interrupt triggers as described above.

Ecrivez le code de UART_IRQHandler. Write the code of UART2_IRQHandler.



MICROPROCESSEURS REPONSES 09/01/2023
MICROPROCESSORS ANSWERS MONCHAL

NOM/NAME Prénom/First name :

.....

Merci de coder votre numéro d'étudiant à 5 chiffres en noircissant bien les cases:

Please code your 5-digit student number by filling in the boxes in black:

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

2 feuilles à détacher : seuls documents à rendre pour la partie Microprocesseur de cet examen. 2
detachable sheet: only documents to be returned for the Microprocessor exam.

QUESTION 1 :

QUESTION 2 :

QUESTION 3 :

QUESTION 4 :

QUESTION 5 :

QUESTION 6 :

QUESTION 7 :

QUESTION 8 :

QUESTION 9 :

QUESTION 10 :

QUESTION 11 :

QUESTION 12 :

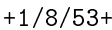
QUESTION 13 :

QUESTION 14 :

QUESTION 15 :

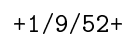
QUESTION 16 :

QUESTION 17 :



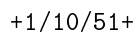
TIM3_IRQH	0	1	2	3	4	5	6
-----------	---	---	---	---	---	---	---

This image shows a full page of primary-ruled paper. It features a series of horizontal dotted lines spaced evenly down the page, designed for handwriting practice. A single solid line runs horizontally across the top of the page, serving as a baseline or margin. The rest of the page is white, providing space for writing.



EXTI_IRQH 0 1 2 3

[illegible]



UART2_IRQH	0	1	2	3	4	5	6	7	8	9	10
------------	---	---	---	---	---	---	---	---	---	---	----

This image shows a full page of primary-ruled paper. It features a solid vertical line on the left side, creating a margin. The rest of the page is filled with horizontal dotted lines, providing a guide for handwriting practice. There are no other markings or text on the page.