

الصفحة	1	الامتحان الوطني الموحد للبكالوريا المسالك المهنية الدورة العادية 2020 - الموضوع -	الجمهورية المغربية وزارة التربية الوطنية والتكوين المهني والتعليم العالي والبحث العلمي المركز الوطني للتقويم والامتحانات	
16	PPPPPPPPPPPPPPPPPPPP			NS 216A

4	مدة الإنجاز	الاختبار التوليقي في المواد المهنية - الجزء 1	المادة
10	المعامل	شعبة الهندسة الكهربائية مسلك النظم الإلكترونية والرقمية	الشعبة أو المسلك

☞ *Le sujet comporte au total 16 pages.*

☞ *Le sujet comporte 3 types de documents :*

- *Pages 02 à 06 : Socle du sujet (Couleur Verte).*
- *Pages 07 à 10 : Documents ressources portant la mention DRES XX (Couleur Rose).*
- *Pages 11 à 16 : Documents réponses portant la mention DREP XX (Couleur Blanche).*

Sujet :

SYSTEME D'IRRIGATION AUTOMATISE.....(70 points)

La numérotation des questions est continue : de la question 1 (Q1) à la question 19 (Q19).

☞ *Toutes les réponses doivent être rédigées sur les documents réponses : DREP XX.*

☞ *Les pages portant en haut la mention DREP XX (Couleur Blanche) doivent être obligatoirement jointes à la copie du candidat même si elles ne comportent aucune réponse.*

☞ *Le sujet est noté sur 70 points.*

☞ *Aucun document n'est autorisé.*

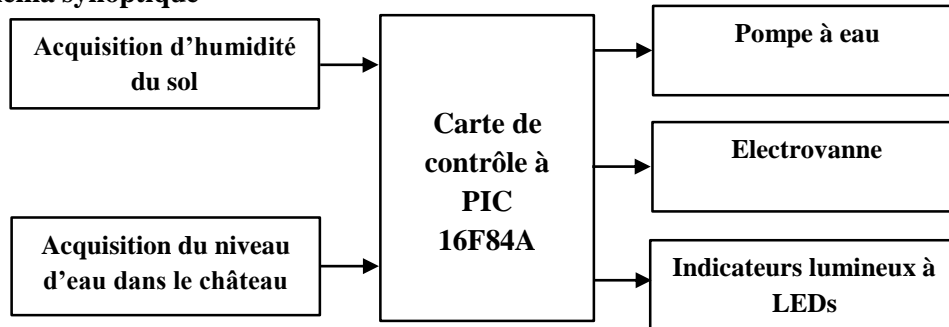
☞ *Sont autorisées les calculatrices non programmables.*

Systeme d'irrigation automatisé

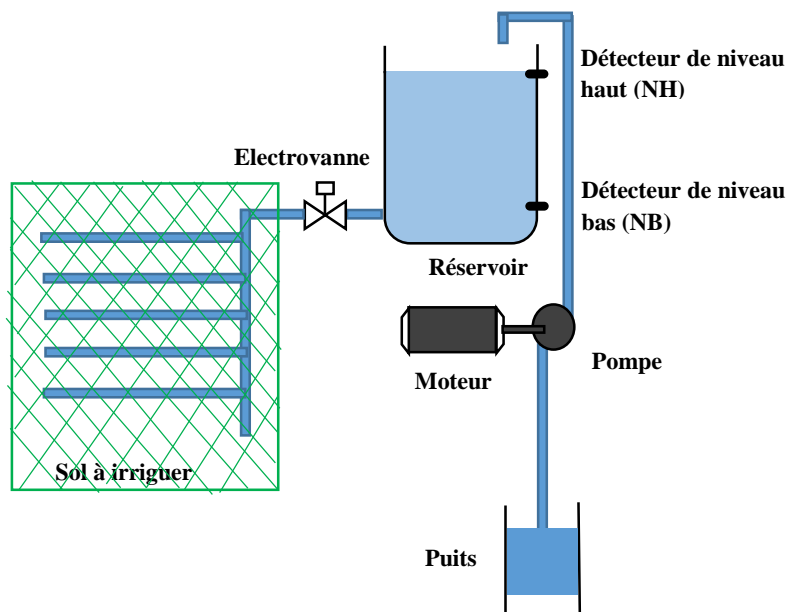
1- Mise en situation

L'**irrigation** est l'opération consistant à apporter artificiellement de l'eau à des végétaux cultivés pour augmenter la production et permettre leur développement normal, en cas de manque d'eau pluviale. Le système d'irrigation automatisé, dont le schéma synoptique est donné ci-dessous, est un système contrôlé par une carte à Microcontrôleur à base du PIC 16F84A.

2- Schéma synoptique



3- Description et fonctionnement



- ✓ L'acquisition d'humidité du sol génère un signal TOR **CH** :
 - **CH = 1**, indique que le sol est sec ;
 - **CH = 0**, indique le sol est humide.
- ✓ L'acquisition du niveau d'eau génère deux signaux logiques (**TOR**) **NB** et **NH** indiquant respectivement la détection du niveau bas et du niveau haut du réservoir d'eau ;
- ✓ L'électrovanne est actionné lorsque le sol est sec ;
- ✓ La pompe d'eau est actionnée lorsque le niveau bas **NB** est détecté et s'arrête lorsque le niveau **NH** est détecté ;
- ✓ Les indicateurs lumineux à LEDs :
 - **L1** pour indiquer si le niveau de l'eau dans le réservoir est au niveau bas ;

- L2 pour indiquer si le niveau de l'eau dans le réservoir est au niveau haut ;
- L3 pour indiquer l'état de marche de la pompe ;
- L4 pour indiquer l'état de marche de l'électrovanne ;
- L5 pour indiquer l'état du sol (sec ou humide).

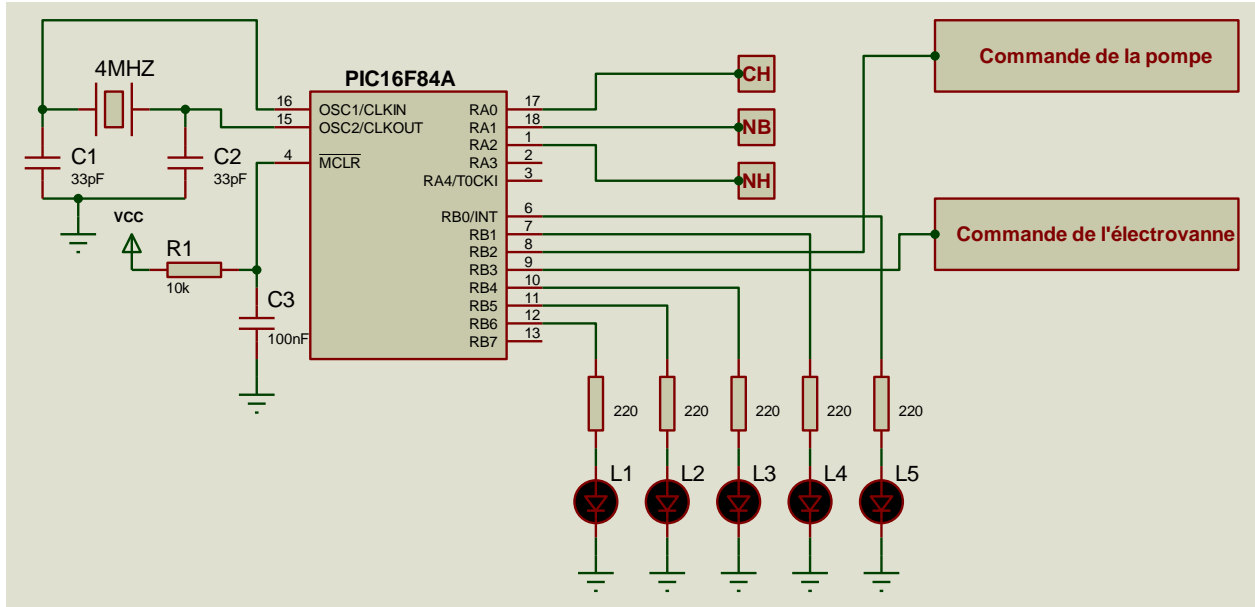


Schéma électronique de la carte de commande du système d'irrigation

4- Situations d'évaluation (voir documents DRES 01 à DRES 04)

- Q1:** Quelle architecture utilisée pour le microcontrôleur PIC ? **3 pts**
- Q2:** Quelle est la référence du microcontrôleur utilisé dans la carte de commande ? **3 pts**
- Q3:** Quel est le type d'horloge utilisé ? **4 pts**
- Q4:** Déterminer la fréquence **F** d'un cycle machine. **3 pts**
- Q5:** En s'aidant de la fiche technique du microcontrôleur, quel est le rôle du bit **RP0** dans le registre **STATUS** ? **3 pts**
- Q6:** Compléter les lignes de configuration en assembleur en respectant les états suivants : **4 pts**

Bit de configuration	Etat choisi
CP	Désactivé
WDT	Désactivé
PWRTE	Activé
OSC	HS

الصفحة	NS 216A	الامتحان الوطني الموحد للبكالوريا - الدورة العادية 2020 - الموضوع
4		- مادة: الاختبار التوليقي في المواد المهنية - الجزء 1 - شعبة الهندسة الكهربائية مسلك النظم الإلكترونية والرقمية
16		

Q7: Identifier les entrées et les sorties du système. **5 pts**

Q8: Compléter les lignes de configuration des entrées/sorties en **assembleur**. **6 pts**

Q9: Compléter les lignes de configuration des ports en **assembleur**. **3 pts**

Q10: Compléter les lignes de configuration des entrées/sorties en **langage C du MikroC**. **4,5 pts**

Q11: Compléter les lignes de configuration des ports en **langage C du MikroC**. **2 pts**

Lorsque la pompe est en marche la **LED L3** clignote (elle s'allume pendant **500 ms** et s'éteint pendant **500 ms**).

Q12: Calculer la fréquence **Fc** de clignotement de la **LED L3**. **2 pts**

Pour clignoter la **LED L3**, on met en œuvre le Timer **TMR0** en **mode Timer**. On rappelle que la fréquence de l'oscillateur **Fosc = 4 MHz**.

Q13: Calculer la temporisation **T1 (en µs)** générée par **TMR0** sans pré-division (sans présalaire), sachant que le **TMR0** est initialisé à une valeur égale à **100**. **3 pts**

Q14: Calculer la temporisation **T2 (en ms)** générée par **TMR0** avec une pré-division de **128**. **3 pts**

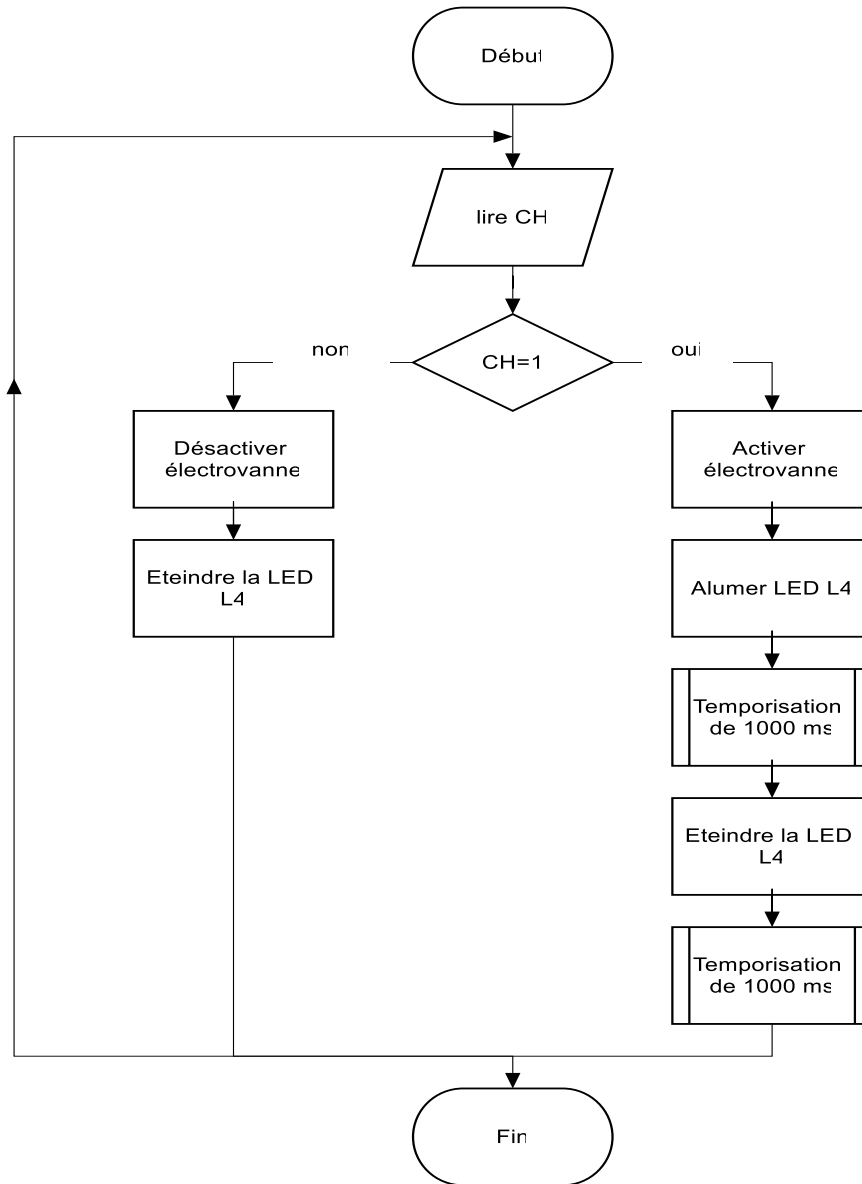
Q15: Par quelle valeur entière **N** faut-il multiplier **T2** pour avoir une temporisation **T** proche de **500 ms**. **2,5 pts**

Q16: Compléter la configuration du registre **OPTION_REG** et donner sa valeur en **Hexadécimal**. **3 pts**

Lorsque l'électrovanne est actionné la **LED L4** clignote (elle s'allume pendant **1000 ms** et s'éteint pendant **1000 ms**).

Q17: Ecrire le programme en langage **C** du **MikroC** permettant de réaliser le clignotement de la **LED L4**. **4 pts**

La gestion de l'irrigation du sol est donnée par l'organigramme suivant :



Q18: Compléter le programme en **langage C** du **MikroC** correspondant à l'organigramme.

6 pts

الصفحة	NS 216A	الامتحان الوطني الموحد للبكالوريا - الدورة العادية 2020 - الموضوع
6		- مادة: الاختبار التوليقي في المواد المهنية - الجزء 1 - شعبة الهندسة الكهربائية مسلك النظم الإلكترونية والرقمية
16		

La gestion du pompage de l'eau dans le réservoir est donnée par le programme suivant :

```

if (PORTA.F1 == 1) // On vérifie si le niveau de l'eau est bas
{
do // on entre dans une boucle conditionnelle
{
PORTB.F2 = 1 ; // On met en marche le moteur de pompage
PORTB.F3 = 1 ; // On allume la led L3
delay_ms(500) ; // Temporisation de 500 ms
PORTB.F3 = 0 ; // On éteint la led L3
delay_ms(500) ; // Temporisation de 500 ms
} while (PORTA.F2 == 0); // On teste si le niveau de l'eau est haut
PORTB.F2 = 0; // On arrête le moteur de pompage
}

```

Q19: Compléter l'organigramme correspondant au programme en langage C ci-dessus : **6 pts**

DRES 01

PIC16F84A

2.2 Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device.

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 2-2 shows the data memory map organization.

Instructions *MOVWF* and *MOVF* can move values from the W register to any location in the register file ("F"), and vice-versa.

The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR) (Section 2.5). Indirect addressing uses the present value of the RP0 bit for access into the banked areas of data memory.

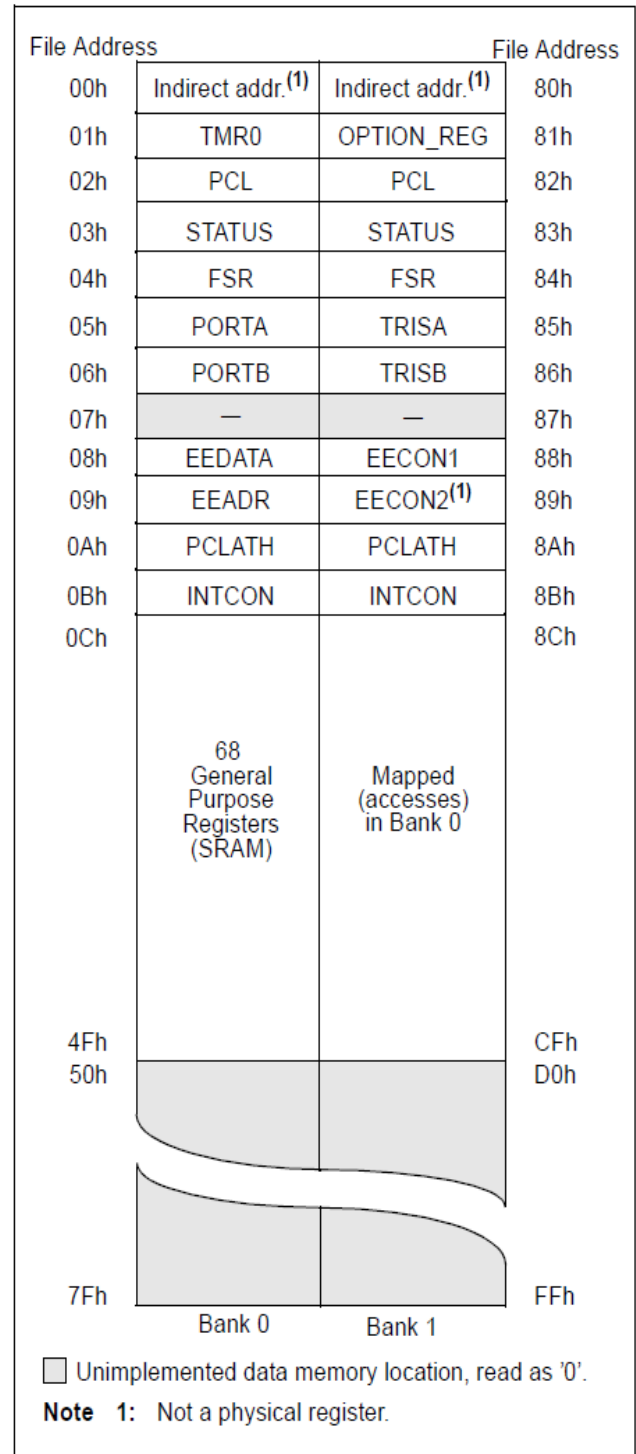
Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers. The remainder are General Purpose Registers, implemented as static RAM.

2.2.1 GENERAL PURPOSE REGISTER FILE

Each General Purpose Register (GPR) is 8-bits wide and is accessed either directly or indirectly through the FSR (Section 2.5).

The GPR addresses in Bank 1 are mapped to addresses in Bank 0. As an example, addressing location 0Ch or 8Ch will access the same GPR.

FIGURE 2-2: REGISTER FILE MAP - PIC16F84A



DRES 02

PIC16F84A

2.3.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bit for data memory.

As with any register, the STATUS register can be the destination for any instruction. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

Only the `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions should be used to alter the STATUS register (Table 7-2), because these instructions do not affect any status bit.

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16F84A and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

2: The C and DC bits operate as a borrow and digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

3: When the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. The specified bit(s) will be updated according to device logic

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	TO	PD	Z	DC	C
bit 7								bit 0

bit 7-6 **Unimplemented:** Maintain as '0'

bit 5 **RP0:** Register Bank Select bits (used for direct addressing)
 01 = Bank 1 (80h - FFh)
 00 = Bank 0 (00h - 7Fh)

bit 4 **TO:** Time-out bit
 1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
 0 = A WDT time-out occurred

bit 3 **PD:** Power-down bit
 1 = After power-up or by the `CLRWDT` instruction
 0 = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
 1 = The result of an arithmetic or logic operation is zero
 0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for borrow, the polarity is reversed)
 1 = A carry-out from the 4th low order bit of the result occurred
 0 = No carry-out from the 4th low order bit of the result

bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for borrow, the polarity is reversed)
 1 = A carry-out from the Most Significant bit of the result occurred
 0 = No carry-out from the Most Significant bit of the result occurred

Note: A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

DRES 03

PIC16F84A

2.2.2.2 OPTION_REG REGISTER

The OPTION_REG register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: When the prescaler is assigned to the WDT (PSA = '1'), TMR0 has a 1:1 prescaler assignment.

FIGURE 2-1: OPTION_REG REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 - n = Value at POR reset

bit 7: **RBP**: PORTB Pull-up Enable bit
 1 = PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled (by individual port latch values)

bit 6: **INTEDG**: Interrupt Edge Select bit
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin

bit 5: **T0CS**: TMR0 Clock Source Select bit
 1 = Transition on RA4/T0CKI pin
 0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: TMR0 Source Edge Select bit
 1 = Increment on high-to-low transition on RA4/T0CKI pin
 0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3: **PSA**: Prescaler Assignment bit
 1 = Prescaler assigned to the WDT
 0 = Prescaler assigned to TMR0

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

DRES 04
Jeu d'instructions

Mnemonic, operands	Description	Cycles	14-bit opcode				Status affected	
			MSB		LSB			
BYTE ORIENTED FILE REGISTER OPERATIONS								
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z
DECFSZ	f, d	Decrement f, skip if 0	1(2)	00	1011	dfff	ffff	
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z
INCFSZ	f, d	Increment f, skip if 0	1(2)	00	1111	dfff	ffff	
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	
NOP	-	No operation	1	00	0000	0xx0	0000	
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z
BIT ORIENTED FILE REGISTER OPERATIONS								
BCF	f, b	Bit clear f	1	01	00bb	bfff	ffff	
BSF	f, b	Bit set f	1	01	01bb	bfff	ffff	
BTFSC	f, b	Bit test f, skip if clear	1(2)	01	10bb	bfff	ffff	
BTFSS	f, b	Bit test f, skip if set	1(2)	01	11bb	bfff	ffff	
LITERAL AND CONTROL OPERATIONS								
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z
ANDLW	k	AND literal With W	1	11	1001	kkkk	kkkk	Z
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk	
CLRWDT	-	Clear watchdog Timer	1	00	0000	0101	0100	TO, PD
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	
IORLW	k	Inclusive OR literal With W	1	11	1000	kkkk	kkkk	Z
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	
RETLW	k	Return with literal to W	2	11	01xx	kkkk	kkkk	
RETURN	-	Return from subroutine	2	00	0000	0000	1000	
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	TO, PD
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z
XORLW	k	Exclusive OR literal With W	1	11	1010	kkkk	kkkk	Z

DREP 01

Q1: Architecture utilisée pour le microcontrôleur PIC:

Cocher la bonne réponse :

- A- Architecture Von Neumann
- B- Architecture Harvard
- C- Architecture Moore
- D- Architecture Murphy

Q2: Référence :

Q3: Type d'horloge :

Cocher les bonnes réponses :

- A- RC
- B- HS
- C- LP
- D- XT

Q4: Fréquence du cycle :

Q5: Rôle du bit RP0 :

Q6: Lignes de configuration du PIC16F84A :

list -----
#include< ----- >
__config -----

DREP 02

Q7: Entrées et sorties du système :

Entrées		Sorties	
Nom de l'entrée	La broche correspondante	Nom de la sortie	La broche correspondante

Q8: Configuration des entrées/sorties en langage assembleur :

Programme en assembleur	Description
ORG 00	Début du programme principal
	Aller à bank 1
	configurer RA0
	configurer RA1
	configurer RA2
	configurer RB0
	configurer RB1
	configurer RB2
	configurer RB3
	configurer RB4
	configurer RB5
	configurer RB6
	Retour au bank 0

DREP 03

Q9: Configuration des ports en langage assembleur :

Programme en assembleur	Description
ORG 00	Début du programme principal
_____	Aller à bank 1
_____	configurer PortA (les bits non utilisés sont mis à 0)
_____	configurer PortB (les bits non utilisés sont mis à 0)
_____	Retour au bank 0

Q10: Configuration des entrées/sorties en langage C :

Programme en langage C	Description
void main() {	Début du programme principal
TRISA.F0=1;	configurer RA0
_____	configurer RA1
_____	configurer RA2
_____	configurer RB0
_____	configurer RB1
_____	configurer RB2
_____	configurer RB3
_____	configurer RB4
_____	configurer RB5
_____	configurer RB6

Q11: Configuration des ports en langage C :

Programme en langage C	Description
void main() {	Début du programme principal
_____	configurer PortA (les bits non utilisés sont mis à 0)
_____	Configurer PortB (les bits non utilisés sont mis à 0)

Q12: Fréquence Fc:

DREP 04

Q13: Temporisation T1 (en μ s) :

Q14: Temporisation T2 (en ms) :

Q15: Valeur entière N :

Q16: Registre OPTION_REG :

Registre OPTION_REG	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	0	-----	0	-----	-----	-----	-----

OPTION_REG = -----

Q17: Programme en langage C :

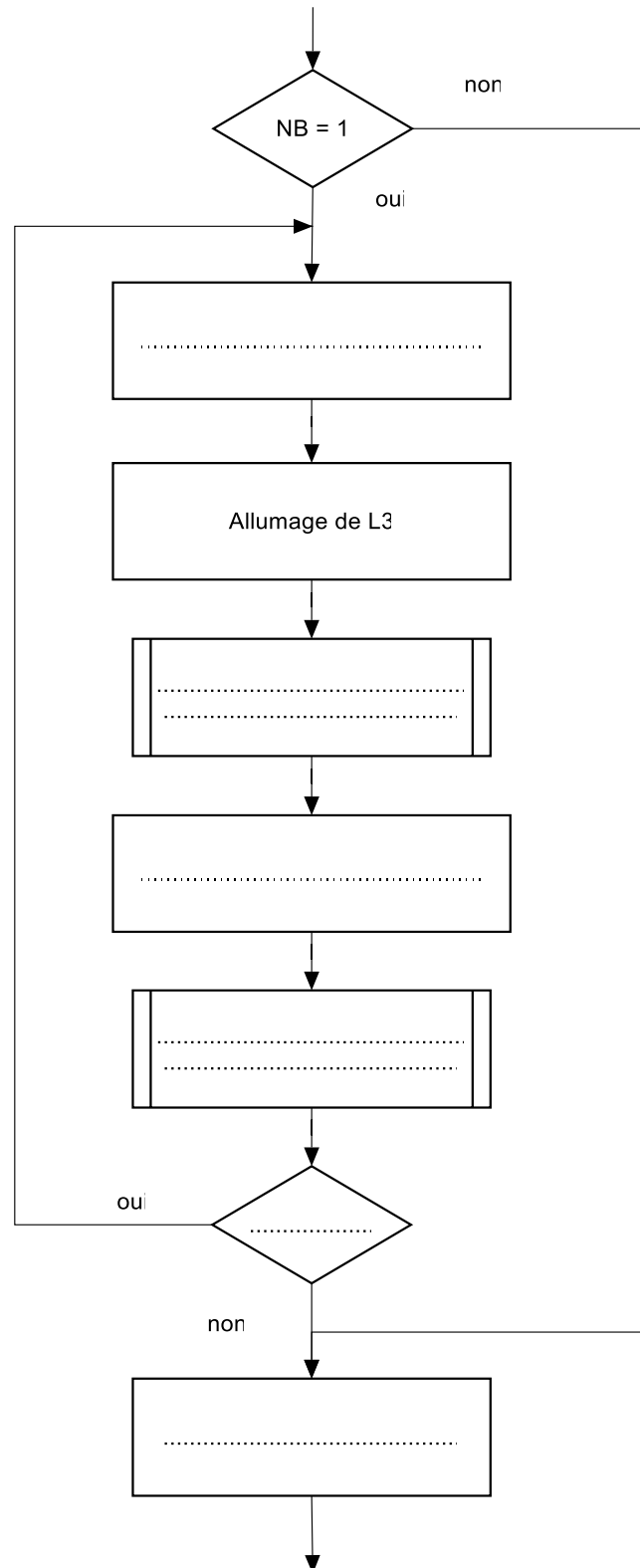
DREP 05

Q18: Le programme en langage C correspondant à l'organigramme :

```
while (1)           // on entre dans une boucle infinie
{
    if (PORTA.F0 == 1)           // On vérifie si l'entrée CH =1
    {
        PORTB.F3 = 1 ;           // On met en marche l'électrovanne
        ..... // On allume la led L4
        ..... // Temporisation de 1000 ms
        ..... // On éteint la led L4
        ..... // Temporisation de 1000 ms
    }
    else {
        ..... // On désactive l'électrovanne
        ..... // On éteint la led L4
    }
}
```

DREP 06

Q19: L'organigramme correspondant au programme en langage C :



الصفحة	<p style="text-align: center;">الامتحان الوطني الموحد للبكالوريا المسالك المهنية الدورة العادية 2020 - عناصر الإجابة -</p>		<p style="text-align: center;">المملكة المغربية وزارة التربية الوطنية والتكوين المهني والتعليم العالي والبحث العلمي المركز الوطني للتقويم والامتحانات</p>	
1				
5				
***	PPPPPPPPPPPPPPPPPPPP	NR 216A		
4	مدة الإنجاز	الاختبار التوليقي في المواد المهنية - الجزء 1		المادة
10	المعامل	شعبة الهندسة الكهربائية مسلك النظم الإلكترونية والرقمية		الشعبة أو المسلك

Eléments de corrigé

Q1. Architecture utilisée pour le microcontrôleur PIC: /3 points

Cocher la bonne réponse :

- A- Architecture Von Neumann
- B- Architecture **Harvard**
- C- Architecture Moore
- D- Architecture Murphy

Q2. Référence : /3 points

PIC 16F84A

Q3. Type d'horloge : 2 x2 points = 4 points

Cocher les bonnes réponses :

- A- RC
- B- HS
- C- LP
- D- XT

Q4. Fréquence du cycle : /3 points

$F = 4 / 4 = 1 \text{ MHz}$

Q5. Rôle du bit RP0 : /3 points

Elle permet le choix entre Bank 0 et Bank 1

Q6. Lignes de configuration du PIC16F84A :

/4 points

list p=16F84A

/1 point

#include<p16F84A.inc>

/1 point

__config __CP_OFF & __WDT_OFF & __PWRTE_ON & __HS_OSC (ou __config H'3FF2) **/2 points**

Q7. Entrées et sorties du système :

20 x 0,25 point = 5 points

Entrées		Sorties	
Nom de l'entrée	La broche correspondante	Nom de la sortie	La broche correspondante
CH	RA0	Commande de la pompe	RB2
NB	RA1	Commande de l'électrovanne	RB3
NH	RA2	L1	RB6
		L2	RB5
		L3	RB4
		L4	RB1
		L5	RB0

Q8. Configuration des entrées/sorties en langage assembleur :

12 x 0,5 point = 6 points

Programme en assembleur	Description
ORG 00	Début du programme principal
BSF STATUS, RP0	Aller à bank 1
BSF TRISA, 0	configurer RA0
BSF TRISA, 1	configurer RA1
BSF TRISA, 2	configurer RA2
BCF TRISB, 0	configurer RB0
BCF TRISB, 1	configurer RB1
BCF TRISB, 2	configurer RB2
BCF TRISB, 3	configurer RB3
BCF TRISB, 4	configurer RB4
BCF TRISB, 5	configurer RB5
BCF TRISB, 6	configurer RB6
BCF STATUS, RP0	Retour au bank 0

Q9. Configuration des ports en langage assembleur :

6 x 0,5 point = 3 points

Programme en assembleur	Description
ORG 00	Début du programme principal
BSF STATUS, RP0	Aller à bank 1
MOVLW 0x07 MOVWF TRISA	configurer PortA (les bits non utilisés sont mis à 0)
MOVLW 0x00 MOVWF TRISB	configurer PortB (les bits non utilisés sont mis à 0)
BCF STATUS, RP0	Retour au bank 0

Q10. Configuration des entrées/sorties en langage C :

9 x 0,5 point = 4,5 points

Programme en langage C	Description
void main() {	Début du programme principal
TRISA.F0=1;	configurer RA0
TRISA.F1=1;	configurer RA1
TRISA.F2=1;	configurer RA2
TRISB.F0=0;	configurer RB0
TRISB.F1=0;	configurer RB1
TRISB.F2=0;	configurer RB2
TRISB.F3=0;	configurer RB3
TRISB.F4=0;	configurer RB4
TRISB.F5=0;	configurer RB5
TRISB.F6=0;	configurer RB6

Q11. Configuration des ports en langage C :

2 x 1 point = 2 points

Programme en langage C	Description
void main() {	Début du programme principal
TRISA = 0x07 ;	configurer PortA (les bits non utilisés sont mis à 0)
TRISB = 0x00 ;	Configurer PortB (les bits non utilisés sont mis à 0)

Q12. Fréquence :

/2 points

$T_c = (500+500) \text{ ms} = 1 \text{ s}$ donc $F_c = 1 \text{ Hz}$

Q13. Temporisation T1 (en μ s) : **/3 points**

$$T = (255 - 100) * 4 / 4 = 155 \mu s$$

Q14. Temporisation T2 (en ms) : **/3 points**

$$T2 = 128 * T1 = 128 * 155 = 19840 \mu s = 19,84 ms$$

Q15. Valeur entière N : **/2,5 points**

$$N = 500 / 19,84 = 25$$

Q16. Registre OPTION_REG : **6 x 0,5 point = 3 points**

Registre OPTION_REG	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	0	0	0	0	1	1	0

OPTION_REG = 0x86

Q17. Programme en langage C : **4 x 1 point = 4 points**

```
PORTB.F1 = 1 ;
delay_ms (1000) ;
PORTB.F1 = 0 ;
delay_ms(1000) ;
```

Q18. Le programme en langage C correspondant à l'organigramme : **6 x 1 point = 6 points**

```
while (1) // on entre dans une boucle infinie
{
    if (PORTA.F0 == 1) // On vérifie si l'entrée CH = 1
    {
        PORTB.F3 = 1 ; // On met en marche l'électrovanne
        PORTB.F1 = 1 ; // On allume la led L4
        delay_ms(1000) ; // Temporisation de 1000 ms
        PORTB.F1 = 0 ; // On éteint la led L4
        delay_ms(1000) ; // Temporisation de 1000 ms
    }
    else {
        PORTB.F3=0 ; // On désactive l'électrovanne
        PORTB.F1 = 0 ; // On éteint la led L4
    }
}
```

Q19. L'organigramme correspondant au programme en langage C :

6 x 1 point = 6 points

