Nom et prénom : ..... N° : .....

Classe : 4e Sciences Techniques ..... Date : .....

Note

..... / 20

A. PARTIE CIRCUITS COMBINATOIRES INTÉGRÉS (8 points)

Se référer au dossier technique, pages 1/3 et 3/3.

A.1. Étude du générateur de tarif (circuits U2 et U3)

A.1.1. En se référant au tableau 2, compléter le tableau ci-dessous par le tarif T écrit en binaire naturel. **(0,5 pt)**

Classe	K1	K0	Tarif (D)	T3	T2	T1	T0
aucun véhicule	0	0	0	**0**	**0**	**0**	**0**
1	0	1	2	**0**	**0**	**1**	**0**
2	1	0	3	**0**	**0**	**1**	**1**
3	1	1	5	**0**	**1**	**0**	**1**

Barème : 0,125 par ligne juste (4 lignes).

A.1.2. Sachant que A est reliée à K0 et B à K1, compléter le tableau des niveaux logiques à appliquer aux entrées de données des quatre multiplexeurs (cadres en pointillés de la figure 3). **(1 pt)**

Multiplexeur	entrée C0 (K1K0 = 00)	entrée C1 (K1K0 = 01)	entrée C2 (K1K0 = 10)	entrée C3 (K1K0 = 11)
U2 : 1C0…1C3 → 1Y = T0	**0**	**0**	**1**	**1**
U2 : 2C0…2C3 → 2Y = T1	**0**	**1**	**1**	**0**
U3 : 1C0…1C3 → 1Y = T2	**0**	**0**	**0**	**1**
U3 : 2C0…2C3 → 2Y = T3	**0**	**0**	**0**	**0**

Barème : 0,25 par ligne juste (4 lignes).

A.1.3. Donner le niveau logique des entrées $\overline{1G}$ et $\overline{2G}$ de U2 et U3. Justifier pourquoi A doit être reliée à K0 et non à K1. **(0,5 pt)**

$\overline{1G} = \overline{2G} =$ **0** (validation active à l'état bas) pour U2 et U3. A est l'entrée de sélection de **poids faible** : elle reçoit le bit de poids faible K0 du code de classe.

Barème : niveaux 0,25 ; justification 0,25.

A.1.4. Déduire du tableau de la question A.1.1 les équations logiques simplifiées de T0, T1, T2 et T3 en fonction de K1 et K0. **(1 pt)**

$T0 = K1$; $T1 = K1 \oplus K0$; $T2 = K1 \cdot K0$; $T3 = 0$

Barème : 0,25 par équation juste.

A.2. Étude du calcul de la monnaie à rendre (circuit U5)

A.2.1. En se référant au document constructeur du 74LS381, répondre par « Vrai » ou « Faux ». **(1 pt)**

Proposition	Vrai ou Faux
Le 74LS381 peut réaliser huit fonctions différentes.	**Vrai**
Pour S2S1S0 = 001 et C_N = 1, il réalise F = A − B.	**Faux**
Pour S2S1S0 = 000, toutes les sorties F sont au niveau logique 1.	**Faux**
Pour S2S1S0 = 011 et C_N = 0, il réalise l'addition F = A + B.	**Vrai**

Barème : 0,25 par proposition.

A.2.2. Donner les niveaux logiques à appliquer aux entrées de sélection et à C_N de U5 pour obtenir R = P − T. **(0,5 pt)**

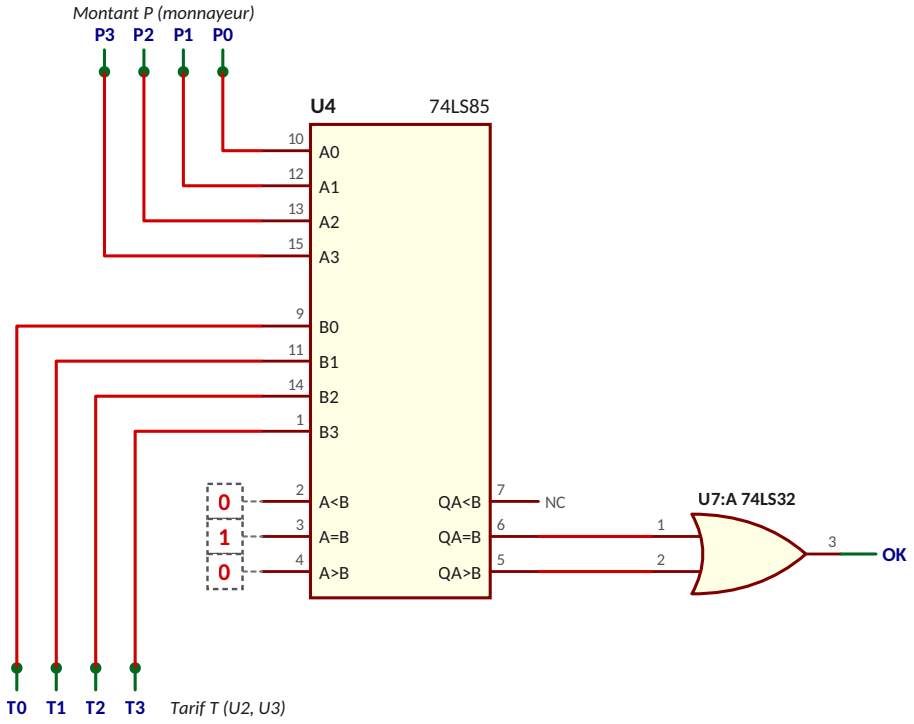
S2	S1	S0	C_N
0	**1**	**0**	**1**

Barème : S2S1S0 = 010 : 0,25 ; C_N = 1 : 0,25.

© Dr Hassène Bedoui — genieelectrique.tn — Devoir corrigé en téléchargement gratuit — Reproduction commerciale interdite

A.3. Étude du comparateur (circuits U4 et U7)

A.3.1. Compléter le schéma ci-dessous : écrire dans les cadres en pointillés le niveau logique des entrées de mise en cascade et tracer les liaisons assurant : – l'application du montant P sur les entrées A et du tarif T sur les entrées B de U4 ; – l'obtention du signal OK = 1 lorsque P ≥ T à l'aide de la porte U7:A. **(1,5 pt)**



Barème : cascade A>B = 0, A<B = 0, A=B = 1 : 0,5 ; liaisons P → A et T → B (poids respectés) : 0,5 ; QA>B et QA=B vers U7:A : 0,5.

A.3.2. Écrire l'équation logique du signal OK en fonction des sorties de U4. **(0,5 pt)**

OK = QA>B + QA=B (P > T ou P = T)

Barème : 0,5.

A.3.3. En l'absence de véhicule, quelle est la valeur de OK ? Justifier. En déduire pourquoi le programme teste aussi le code de classe (instruction *if (OK == 1 && PORTC != 0)*). **(0,5 pt)**

Sans véhicule, K1K0 = 00 ⇒ T = 0 ⇒ P ≥ 0 est toujours vrai ⇒ **OK = 1**. Sans le test de la classe, la barrière se lèverait sans véhicule ni paiement.

Barème : OK = 1 justifié 0,25 ; conséquence 0,25.

A.3.4. Compléter le tableau de fonctionnement du module de tarification. **(1 pt)**

Classe	P (D)	T (binaire)	OK	F = R (binaire)	Barrière (levée / baissée)
1	2	**0010**	**1**	**0000**	**levée**
2	5	**0011**	**1**	**0010**	**levée**
3	4	**0101**	**0**	**1111 (non significatif)**	**baissée**
3	9	**0101**	**1**	**0100**	**levée**

Barème : 0,25 par ligne juste.

© Dr Hassène Bedoui — genieelectrique.tn — Devoir corrigé en téléchargement gratuit — Reproduction commerciale interdite

Ne rien écrire ici

B. PARTIE MICROCONTRÔLEUR (12 points)

Se référer au dossier technique, pages 1/3 et 2/3.

B.1. Configuration des ports

B.1.1. Compléter le tableau ci-dessous puis les valeurs à charger dans les registres de direction. **(1,5 pt)**

Broches	Entrée / Sortie	Analogique / Logique
RA0/AN0	**Entrée**	**Analogique**
RB0/INT, RB1, RB4	**Entrée**	**Logique**
RB2, RB5, RB6, RB7	**Sortie**	**Logique**
RC0, RC1	**Sortie**	**Logique**

Registre	Binaire	Hexadécimal
TRISA	**0000 0001**	**0x01**
TRISB	**0001 0011**	**0x13**
TRISC	**0000 0000**	**0x00**

1 : entrée ; 0 : sortie.

Barème : tableau 0,5 (0,125 par ligne) ; TRISB 0,5 ; TRISA 0,25 ; TRISC 0,25.

B.2. Conversion analogique-numérique

B.2.1. Compléter le registre ADCON1 : seule AN0 analogique, résultat justifié à droite. **(0,75 pt)**

ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	Hexadécimal
Valeur	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0x8E**

Barème : ADFM 0,25 ; PCFG3…0 = 1110 : 0,25 ; 0x8E : 0,25.

B.2.2. Calculer le quantum q du convertisseur puis exprimer le résultat N de la conversion en fonction de la hauteur h. **(0,5 pt)**

q = 5 / 1023 ≈ **4,89 mV** ; N = U_h / q = **N = 1023·h / 5 ≈ 204,6·h**

Barème : q 0,25 (accepter 5/1024) ; expression de N 0,25.

B.2.3. Compléter le tableau des seuils de classification (N arrondi à l'entier le plus proche). **(1 pt)**

Seuil	h (m)	U_h (V)	N (décimal)	N (hexadécimal)
classe 1 / classe 2	2	**2**	**409**	**0x199**
classe 2 / classe 3	3	**3**	**614**	**0x266**

Barème : U_h 2 × 0,125 ; N décimal 2 × 0,25 ; N hexadécimal 2 × 0,125.

B.2.4. La conversion donne N = 450. Calculer la hauteur du véhicule ; en déduire sa classe, le code K1K0 envoyé et le tarif affiché. **(0,75 pt)**

U_h = 450 × 5 / 1023 ≈ 2,20 V ⇒ **h ≈ 2,20 m** ⇒ 2 m ≤ h < 3 m : **classe 2**, **K1K0 = 10**, tarif **3 D**.

Barème : h 0,25 ; classe et code 0,25 ; tarif 0,25.

B.3. Interruptions et Timer0

B.3.1. Compléter les registres OPTION_REG (front descendant sur RB0/INT, tirages internes désactivés, horloge interne, T0SE = 0, prédiviseur 1:64 affecté au Timer0) et INTCON (seule l'interruption INT autorisée). **(1,5 pt)**

OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	Hexa
Valeur	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0x85**
INTCON	**GIE**	**PEIE**	**T0IE**	**INTE**	**RBIE**	**T0IF**	**INTF**	**RBIF**	**Hexa**
Valeur	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0x90**

Barème : OPTION_REG : RBPU = 1 et INTEDG = 0 : 0,25 ; T0CS = T0SE = PSA = 0, PS = 101 : 0,25 ; 0x85 : 0,25. INTCON : GIE = INTE = 1 : 0,5 ; 0x90 : 0,25.

B.3.2. La temporisation est réalisée par scrutation du drapeau T0IF (voir le programme B.4). Quel événement met T0IF à 1 ? Pourquoi le programme doit-il le remettre à 0 après chaque détection ? **(0,5 pt)**

T0IF passe à 1 au **débordement du Timer0** (TMR0 passe de 255 à 0). Il ne s'efface pas seul : sans remise à 0, le même débordement serait **compté plusieurs fois** et la temporisation serait fausse.

Barème : événement 0,25 ; remise à 0 justifiée 0,25.

© Dr Hassène Bedoui — genieelectrique.tn — Devoir corrigé en téléchargement gratuit — Reproduction commerciale interdite

B.3.3. Calculer la durée T_D séparant deux débordements du Timer0 (quartz 4 MHz). **(0,5 pt)**

$T_{cy} = 4 / f_{osc} = 1\ \mu s$; $T_D = 64 \times 256 \times 1\ \mu s =$ **16,384 ms**

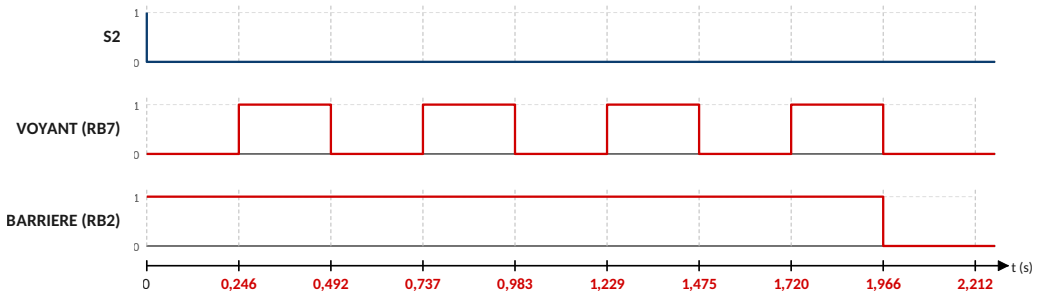
Barème : T_{cy} 0,25 ; T_D 0,25.

B.3.4. Déterminer le nombre N_1 de débordements correspondant à ≈ 0,25 s, la durée réelle t_1 entre deux inversions du voyant, puis la durée réelle t_s de la temporisation de sécurité. **(1 pt)**

0,25 / 16,384 ms ≈ 15,26 ⇒ **$N_1 = 15$** ; $t_1 = 15 \times 16{,}384$ ms ≈ **0,246 s** ; $t_s = 8 \times t_1 \approx$ **1,966 s**.

Barème : N_1 0,5 ; t_1 0,25 ; t_s 0,25.

B.3.5. Le véhicule dégage S2 à t = 0. Compléter la graduation de l'axe des temps (une division = t_1) puis tracer l'allure des signaux VOYANT et BARRIERE. **(1 pt)**



Barème : graduation 0,25 ; VOYANT (8 inversions, 4 périodes) 0,5 ; BARRIERE baissée à t_s 0,25.

B.4. Programme

B.4.1. Compléter le programme ci-dessous en s'aidant des commentaires et des réponses précédentes. **(12 × 0,25 pt)**

Instructions (mikroC PRO for PIC)	Commentaires
`// Déclaration des variables et des broches`	
`sbit S2 at RB4_bit; sbit OK at RB1_bit; sbit BARRIERE at RB2_bit;` `sbit FEU_V at RB5_bit; sbit FEU_R at RB6_bit; sbit VOYANT at RB7_bit;`	*Noms des broches utilisées*
`unsigned int N;` `unsigned short arrivee = 0, cpt = 0, ninv = 0;`	*N : résultat de la conversion ; arrivee = 1 : véhicule détecté ; cpt : nombre de débordements ; ninv : nombre d'inversions du voyant*
`// Procédure d'interruption`	

`void interrupt() {`
`  if (INTCON.INTF == 1) { arrivee = 1;  INTCON.INTF = 0; }`
`}`
	Front descendant sur RB0/INT (S1) : mémoriser l'arrivée du véhicule, effacer le drapeau INTF

`// Programme principal`	

`void main() {`
`  TRISA = 0x01;  TRISB = 0x13;  TRISC = 0x00;`
`  PORTB = 0x40;  PORTC = 0;`
	Configuration des ports (tableau 1 du DT) ; feu rouge allumé, tout le reste éteint (barrière baissée) ; aucune classe

`  ADCON1 = 0x8E;`
`  OPTION_REG = 0x85;`
`  INTCON = 0x90;`
	AN0 analogique, justifié à droite ; front descendant sur INT, Timer0 interne 1:64 ; seule INT autorisée

`  while (1) {`
`  if (arrivee == 1) {`
`  N = ADC_Read(0);`
`  if (N < 409) PORTC = 0x01;`
`  else if (N < 614) PORTC = 0x02;`
`  else PORTC = 0x03;`
`  arrivee = 0;`
`  }`
	Boucle infinie. Conversion de U_h puis envoi du code de classe K1K0 sur RC1 RC0 (tableau 2 du DT)

`  if (OK == 1 && PORTC != 0) {`
`  BARRIERE = 1;  FEU_V = 1;  FEU_R = 0;`
`  while (S2 == 0);`
`  while (S2 == 1);`
	Montant suffisant et véhicule classé : lever la barrière, feu vert ; attendre que le véhicule ait dégagé S2

`  TMR0 = 0;  INTCON.T0IF = 0;  cpt = 0;  ninv = 0;`
`  while (ninv < 8) {`
`  if (INTCON.T0IF == 1) {`
`  INTCON.T0IF = 0;`
`  cpt++;`
`  if (cpt == 15) { cpt = 0;  VOYANT = !VOYANT;  ninv++; }`
`  }`
`  }`
	Temporisation de sécurité (≈ 2 s) par scrutation du drapeau T0IF : à chaque débordement, effacer T0IF et le compter ; tous les 15 débordements (≈ 0,25 s), inverser le voyant orange ; arrêter après 8 inversions

`  BARRIERE = 0;  FEU_V = 0;  FEU_R = 1;  PORTC = 0;` `  }  }  }`	*Baisser la barrière, rallumer le feu rouge, effacer la classe*

© Dr Hassène Bedoui — genieelectrique.tn — Devoir corrigé en téléchargement gratuit — Reproduction commerciale interdite

GRILLE DE NOTATION DÉTAILLÉE (20 points)

Partie	Question	Réponse attendue	Détail du barème	Points	Note
A	**A.1.1**	T = 0000 ; 0010 ; 0011 ; 0101	4 × 0,125	**0,5**	
A	**A.1.2**	T0 : 0011 ; T1 : 0110 ; T2 : 0001 ; T3 : 0000	4 × 0,25	**1**	
A	**A.1.3**	$\overline{G}$ = 0 ; A = poids faible ⇒ K0	0,25 + 0,25	**0,5**	
A	**A.1.4**	T0 = K1 ; T1 = K1 ⊕ K0 ; T2 = K1·K0 ; T3 = 0	4 × 0,25	**1**	
A	**A.2.1**	V ; F ; F ; V	4 × 0,25	**1**	
A	**A.2.2**	S2S1S0 = 010 ; C_N = 1	0,25 + 0,25	**0,5**	
A	**A.3.1**	Cascade 0 / 0 / 1 ; P → A, T → B ; QA>B, QA=B → OU	3 × 0,5	**1,5**	
A	**A.3.2**	OK = QA>B + QA=B	0,5	**0,5**	
A	**A.3.3**	T = 0 ⇒ OK = 1 ; test de la classe obligatoire	0,25 + 0,25	**0,5**	
A	**A.3.4**	Tableau de fonctionnement (4 lignes)	4 × 0,25	**1**	
Total partie A — circuits combinatoires intégrés				**8**	
B	**B.1.1**	Tableau E/S ; TRISA = 0x01 ; TRISB = 0x13 ; TRISC = 0x00	0,5 + 0,25 + 0,5 + 0,25	**1,5**	
B	**B.2.1**	ADCON1 = 1000 1110 = 0x8E	3 × 0,25	**0,75**	
B	**B.2.2**	q ≈ 4,89 mV ; N = 1023·h/5	0,25 + 0,25	**0,5**	
B	**B.2.3**	2 V / 409 / 0x199 ; 3 V / 614 / 0x266	0,25 + 0,5 + 0,25	**1**	
B	**B.2.4**	h ≈ 2,20 m ; classe 2, K1K0 = 10 ; 3 D	3 × 0,25	**0,75**	
B	**B.3.1**	OPTION_REG = 0x85 ; INTCON = 0x90	0,75 + 0,75	**1,5**	
B	**B.3.2**	Débordement de TMR0 (255 → 0) ; effacement logiciel de T0IF	0,25 + 0,25	**0,5**	
B	**B.3.3**	T_{cy} = 1 µs ; T_D = 16,384 ms	0,25 + 0,25	**0,5**	
B	**B.3.4**	N_1 = 15 ; t_1 ≈ 0,246 s ; t_s ≈ 1,966 s	0,5 + 0,25 + 0,25	**1**	
B	**B.3.5**	Graduation ; VOYANT ; BARRIERE	0,25 + 0,5 + 0,25	**1**	
B	**B.4.1**	12 instructions à compléter	12 × 0,25	**3**	
Total partie B — microcontrôleur				**12**	
TOTAL				**20**	

Corrigé du programme (dans l'ordre d'apparition)

1	2	3	4	5	6	7	8	9	10	11	12
0x01	**0x13**	**0x40**	**0x8E**	**0x85**	**0x90**	**ADC_Read(0)**	**614**	**8**	**0**	**15**	**!VOYANT**

Consignes de correction

- Une erreur n'est pénalisée qu'une fois : une valeur fausse en B.1, B.2 ou B.3 recopiée de façon cohérente dans le programme obtient les points du programme.
- Valeurs acceptées en binaire (0b…) ou en décimal si elles sont exactes. TRISA = 0xFF est faux : le DT impose les broches non utilisées en sorties ; VOYANT = ~VOYANT accepté.
- B.2.3 : accepter 613 (partie entière) à condition que le programme soit cohérent. B.2.2 : accepter q = 5/1024.
- B.3.4 : accepter N_1 = 16 si l'élève arrondit par excès et en tire t_1 = 0,262 s et t_s = 2,097 s (0,5 sur 1).
- A.3.1 : toute liaison qui inverse les poids (P0 sur A3…) est comptée fausse pour le groupe concerné. Accepter A>B et A<B « indifférents » si A=B = 1.
- A.3.4, ligne 3 : accepter « résultat négatif / non significatif » sans la valeur 1111.

Compétences évaluées — Partie A : exploiter un document constructeur ; utiliser un multiplexeur comme générateur de fonctions ; compléter un câblage (comparateur + porte) ; exploiter une UAL. Partie B : configurer les ports et les registres (TRISx, ADCON1, OPTION_REG, INTCON) ; exploiter le CAN ; calculer une temporisation par Timer0 (scrutation de T0IF) ; tracer un chronogramme ; compléter un programme utilisant l'interruption INT.

© Dr Hassène Bedoui — genieelectrique.tn — Devoir corrigé en téléchargement gratuit — Reproduction commerciale interdite