

GÉNIE ÉLECTRIQUE

MANUEL D'ACTIVITÉS

4^{ème} année de l'enseignement secondaire Sciences Techniques

Auteurs:

Hiyem FRIKHA DJEMEL
Professeur principal émérite HC

Fredj HMIDI
Professeur principal émérite HC

Hatem BEN HMIDA
Professeur principal émérite HC

Mohamed ELLEUCH
Professeur principal émérite HC

Ridha SAIBI
Inspecteur général de l'enseignement
préparatoire et secondaire

Zouhaier SBAI
Inspecteur de l'enseignement
préparatoire et secondaire

Évaluateurs:

Anis SELLAMI
Professeur universitaire

Mohamed LABIDI
Inspecteur général expert
en éducation

Rafik KASTOURI
Inspecteur général de l'enseignement
préparatoire et secondaire



ecole.edunet.tn



www.cnp.com.tn

© Tous droits réservés au Centre National Pédagogique

© Tous droits réservés au Centre National Pédagogique

AVANT-PROPOS

L'enseignement de la discipline technologie privilégie l'acquisition de connaissances globales par approche systémique et en promouvant l'utilisation des nouvelles technologies. Il se base sur des produits supports qui peuvent être de l'environnement quotidien de l'élève ou de l'environnement industriel. Il permet de :

- Faire découvrir à l'apprenant les divers constituants des systèmes techniques pour l'aider à mieux comprendre leurs fonctionnements et affirmer son choix personnel ;
- développer chez l'apprenant les compétences de raisonnement, de communication, d'expression, d'organisation de travail et de recherche méthodique ;
- développer chez l'apprenant les capacités d'auto apprentissage.

Cet ouvrage est conforme aux programmes d'étude de la 4^{ème} année de l'enseignement secondaire, génie électrique, section sciences techniques. Il est publié en deux volumes : un manuel de cours (numérique) et un manuel d'activités (papiers). Ces deux manuels sont destinés à la fois aux apprenants et aux enseignants. Ils sont composés de six thèmes. Ils gardent, cependant, l'avantage de proposer :

- A l'enseignant, des démarches et des stratégies pédagogiques favorisant l'expérimentation, la manipulation, la constatation, l'analyse et la synthèse et permettant la mobilisation et l'utilisation efficace de l'ensemble des ressources.
- A l'apprenant,
 - De développer ses compétences et de s'exercer en autonomie.
 - De développer sa capacité à exploiter, modifier et mettre en œuvre des objets et des systèmes techniques tout en assurant la sécurité de l'utilisateur et du matériel ainsi que la préservation de l'environnement.
 - D'acquérir une culture technologique basée sur la combinaison des connaissances et des compétences lui permettant de comprendre le fonctionnement des systèmes et objets techniques et de découvrir les solutions techniques envisagées.
 - D'ouvrir des horizons pour que l'apprenant puisse exploiter et utiliser les technologies émergentes.

Le manuel numérique est constitué par des ressources numériques. Il offre les éléments de cours, des aides pédagogiques, des activités de recherche, des animations ou des vidéos, etc... que les apprenants consultent en scannant un code QR imprimé sur les pages du manuel d'activités.

Un résumé et un code QR pour l'évaluation sont présentés à la fin de chaque séquence. Les aides pédagogiques sont en formats PDF, ou en vidéos.

Les réactions, commentaires et critiques constructives sont attendus à l'adresse suivante : equipeauteurs.4sctechge@gmail.com

Les Auteurs

PREFACE

A une époque où l'on assiste à un véritable développement croissant des supports d'apprentissage, qu'ils soient numériques, audiovisuels ou autres, le manuel scolaire reste encore de très loin le support d'apprentissage le plus efficace et le plus répandu.

Traditionnellement, le manuel servait principalement à transmettre des connaissances et à constituer un réservoir d'évaluations. Il avait aussi une fonction implicite de véhiculer des valeurs sociales et culturelles.

Aujourd'hui, ces fonctions sont encore d'actualité. Mais les manuels scolaires doivent également répondre à de nouveaux besoins : développer auprès des apprenants des habitudes de travail, proposer des méthodes d'apprentissages en phase avec le changement continu du monde dans de nombreuses technologies de pointe, intégrer les connaissances acquises à la vie de tous les jours.

La mission de l'école était d'inculquer aux apprenants des savoirs, savoir-faire, savoir-être et savoir-devenir. Notre école tunisienne a relativement réussi à développer les deux premières composantes, mais d'une façon non transférable et c'était le mal dont souffrait notre système éducatif. Les entrées pour développer ces savoirs et savoir-faire étaient à la fois par les contenus puis dans un deuxième temps par des objectifs morcelés suite au développement de la PPO (Pédagogie Par Objectifs).

Une autre approche a vu le jour à la fin des années 80 et a connu un développement important dans plusieurs pays c'est la pédagogie de l'intégration qui consistait à définir des compétences en partant des finalités de l'enseignement et des situations problèmes significatives. C'est un changement de paradigme qui s'est opéré et qui s'inscrit à la fois dans la rupture et la continuité :

- La continuité réside dans le fait que le savoir et le savoir-faire restent l'objet de l'apprentissage, l'élève est au centre de l'acte éducatif.
- La rupture réside dans le fait que le savoir et savoir-faire servent à résoudre des problèmes dans des situations significatives donnant un sens à l'apprentissage.

L'enseignement-apprentissage de la discipline technologie favorise l'émergence et le développement des valeurs sociétales et humanistes permettant la formation du citoyen. Des valeurs tels que la coopération, le respect d'autrui, la responsabilité, le respect de l'environnement... sont appréhendés à travers les activités pratiques et le projet commun encadré.

Les compétences de vie sont considérées comme un ensemble complet de compétences liant comportements, attitudes et connaissances que les élèves peuvent développer et conserver tout au long de leur vie.

Dans cet ouvrage, les activités proposées sont abordées par une approche globale et concrète des objets techniques et des systèmes pluri-technologiques didactisés. Elles permettent de consolider les acquis des apprenants et de développer leurs compétences à travers la réalisation d'opérations de mise en œuvre, de mesures, de vérifications et de simulations.

Dans la discipline de technologie, les compétences disciplinaires à développer tout au long du cursus de formation, visent la maîtrise des connaissances et leur mobilisation dans des situations d'apprentissage. Trois compétences disciplinaires sont retenues :

- **CD1**: s'approprier des connaissances et des habiletés pour résoudre des problèmes d'ordre scientifique et technologique.
- **CD2**: mobiliser ses connaissances et ses habiletés pour concevoir, créer et réaliser des produits technologiques.
- **CD3**: communiquer à l'aide de textes, codes et langages scientifiques et technologiques.

Conscients du rôle essentiel des activités pratiques dans l'acquisition des compétences disciplinaires et le développement des compétences de vie, les auteurs proposent une multitude d'activités pour chaque thème dont l'enseignant pourrait sélectionner celles qui s'adaptent le mieux à l'environnement matériel disponible aux laboratoires de technologie.

Toutes les activités sont conçues conformément à l'esprit des nouveaux programmes et aux démarches privilégiées dans le programme d'étude de technologie, particulièrement :



- La démarche d'investigation ;
- la démarche de résolution de problèmes ;
- la démarche par projet.

Dans la mesure du possible, la majorité des apprentissages traités est puisée dans le domaine technologique et dans la vie courante. Elles font appel parfois à l'utilisation de l'outil informatique, logiciels ou autres pour permettre à l'apprenant d'expérimenter, de réfléchir et de prévoir.

LOGOS

Les logos utilisés dans ce manuel sont donnés ci-dessous :

	Compétences disciplinaires
	Compétences de vie
	Contenu du thème
	Lien vers cours numérique
	Code QR
	Situation déclenchante
	Problématique
	Idée
	En petits groupes
	Mise en œuvre
	Résumé
	Grille d'auto-évaluation

	Fiche candidature
	Prévention : Danger électrique

	J'observe l'objet d'apprentissage
	J'analyse la situation
	Je réfléchis
	Je réponds
	Je récapitule
	Je réalise mon Projet
	Je synthétise
	Je conclus

SOMMAIRE

Thème 1		Microcontrôleurs	Pages
Sources d'interruption	Séquence 1	Activité 1 : Situation déclenchante	12
		Activité 2 : Mise en oeuvre d'une interruption par RB0	14
		Activité 3 : Fonction interruption	17
		Activité 4 : Configuration des bits du registre INTCON	18
		Activité 5 : Mise en oeuvre d'une interruption par RBI	19
		Résumé	22
Timers	Séquence 2	Activité 1 : Situation déclenchante	24
		Activité 2 : Détermination de la durée d'un cycle de comptage	25
		Activité 3 : Programmation d'une temporisation de 0,5s	28
		Activité 4 : Programmation d'une temporisation de 30s	29
		Résumé	32
Convertisseur Analogique Numérique	Séquence 3	Activité 1 : Situation déclenchante	34
		Activité 2 : Détermination des tensions issues des axes X et Y	35
		Activité 3 : Conversion Analogique/Numérique des tensions Vx et Vy	36
		Activité 4 : Programmation d'une icône tactile	38
		Résumé	42
		Je réalise mon projet	44
		Grille d'auto-évaluation	47
Thème 2		Logique combinatoire	Pages
Circuits combinatoires	Séquence 1	Activité 1 : Situation déclenchante	50
		Activité 2 : Etude de la fonction FP1	53
		Activité 3 : Etude de la fonction FP2	58
		Activité 4 : Etude de la fonction FP3	61
		Activité 5 : Etude de la fonction multiplexage	65
		Activité 6 : Mise en oeuvre du circuit intégré 74151	67
		Activité 7 : Mise en oeuvre du circuit intégré 74138	68
		Activité 8 : Mise en oeuvre du circuit intégré 74381	70
		Activité 9 : Mise en oeuvre d'une UAL à base de microcontrôleur	72
		Résumé	74
		Je réalise mon projet	78
		Grille d'auto-évaluation	79
Thème 3		Logique séquentielle	Pages
Compteurs/décompteurs intégrés synchrones	Séquence 1	Activité 1 : Situation déclenchante	82
		Activité 2 : Etude du circuit intégré 4510	83
		Activité 3 : Etude du circuit intégré 4516	88
		Activité 4 : Etude du circuit intégré 74193	89
		Activité 5 : Etude comparative entre les circuits intégrés 74192 et 74193	91
		Activité 6 : Etude du circuit intégré 74160	92
		Activité 7 : Etude du circuit intégré 74163	94
		Activité 8 : TIMER0 en mode compteur (Compteur modulo 5)	95
		Activité 9 : TIMER0 en mode compteur et affichage multiplexé	96
		Activité 10 : TIMER0 en mode compteur (Décompteur modulo 30)	98
		Activité 11 : Décompteur modulo 30 à partir d'une procédure d'interruption	100
		Résumé	102

GRAFCE
synchronisé

Séquence 2	Activité 1 : Situation déclenchante (Tri)	104
	Activité 2 : Situation déclenchante (Robot)	110
	Activité 3 : Mode de sécurité du robot	116
	Activité 4 : Programmation d'un Grafcet synchronisé	118
	Résumé :	120
	Je réalise mon projet	122
	Grille d'auto-évaluation	123

Thème 4

Amplificateurs Linéaires Intégrés

Pages

Amplificateur
linéaire
intégré

Séquence 1	Activité 1 : Situation déclenchante	126
	Activité 2 : Etude de la fonction amplification	127
	Activité 3 : Les calculateurs et les opérateurs analogiques	130
	Activité 4 : Etude du montage 1	134
	Activité 5 : Etude du montage 2	136
	Activité 6 : Etude du montage 3	137
	Activité 7 : Etude du montage 4	139
	Activité 8 : Etude du montage 5	140
	Activité 9 : Etude du montage 7	142
	Activité 10 : Etude du montage 8	144
	Activité 11 : Etude du montage 9	146
	Activité 12 : Etude du montage astable	148
	Activité 13 : Solution programmée (Comparateur analogique)	150
	Résumé	152
Je réalise mon projet	156	
Grille d'auto-évaluation	157	

Thème 5

Systemes triphasés équilibrés

Pages

Réseau
triphasé

Séquence 1	Activité 1 : Situation déclenchante	160
	Activité 2 : Etude du réseau triphasé	161
	Activité 3 : Caractéristiques des tensions triphasées	162
	Activité 4 : Carte mondiale des tensions simples	165
	Résumé	166

Moteurs
asynchrones
triphasés

Séquence 2	Activité 1 : Situation déclenchante	168
	Activité 2 : Constitution et principe de fonctionnement d'un moteur asynchrone triphasé	169
	Activité 3 : Bilan de puissances et caractéristiques (LEROY SOMER)	173
	Activité 4 : Bilan de puissances et caractéristiques (BOYUAN811)	177
	Activité 5 : Bilan de puissances et caractéristiques (Logiciel)	181
	Activité 6 : Démarrage des moteurs asynchrones triphasés	186
	Activité 7 : Démarrage direct	187
	Activité 8 : Démarrage étoile-triangle à un seul sens de marche	192
	Activité 9 : Critères de choix	196
Résumé	298	
Grille d'auto-évaluation	203	

Thème 6

Moteur à courant continu

Pages

Moteur à
courant continu
à excitation
indépendante

Séquence 1	Activité 1 : Situation déclenchante	206
	Activité 2 : Bilan énergétique (Banc d'essais)	207
	Activité 3 : Bilan énergétique (LEROY SOMER)	213
	Activité 4 : Bilan énergétique (Logiciel)	217
	Activité 5 : Banc d'essais des moteurs électriques (BOYUAN811)	222
	Activité 6 : Simulation d'un hacheur série	226
	Activité 7 : Variateur de vitesse	230
Résumé	234	
Grille d'auto-évaluation	236	

Liens des ressources numériques

238

THÈME 1 MICROCONTRÔLEURS

Le thème «Microcontrôleurs» permettra aux apprenants d'approfondir leurs savoirs et savoir-faire relatifs à la programmation en langage évolué, d'enrichir leurs habiletés relatives à l'implantation d'un programme dans un microcontrôleur et sa mise en œuvre pour commander un système technique.

COMPOSANTES DES COMPÉTENCES DISCIPLINAIRES ATTENDUES:

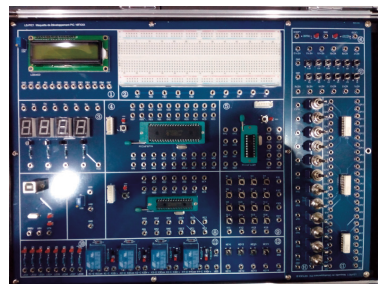
CD 1.7 Programmer un microcontrôleur.

CD 2.7 Mettre en oeuvre un système technique et une maquette didactique pilotés par une carte électronique à base de microcontrôleur.

CD 3.7 Traduire un cahier des charges de fonctionnement d'un système en algorithme et/ou en programme et en rendre compte.

COMPÉTENCES DE VIE VISÉES:

-  Participation.
-  Prise de décision.
-  Communication.



MICROCONTRÔLEURS:

1 PRÉREQUIS

- Présentation.
- Critères de choix.
- Langage de programmation.

3 CONDITIONS MATÉRIELLES NÉCESSAIRES

- Logiciels.
- Microcontrôleurs.
- Maquette didactique.

2 SAVOIRS ET SAVOIR-FAIRE NOUVEAUX

- Sources d'interruption.
- Timers.
- Convertisseur Analogique Numérique.

4 CRITÈRES D'ÉVALUATION

- Choix correct et argumenté d'un microcontrôleur.
- Mise en oeuvre réussie d'une application à base de microcontrôleur programmé en langage évolué.
- Participation active.
- Respect total des règles de communication.



CONTENU DU THÈME 1:

Situation déclenchante de chaque séquence : Activité N°1	Présenter l'objet d'apprentissage :			
<ul style="list-style-type: none"> • Apprentissage : <ul style="list-style-type: none"> - Je développe. - J'applique. - Je consolide mes acquis. • Évaluation : <ul style="list-style-type: none"> - J'évalue mes acquis. 	ACTIVITÉS :		CDi,j	SUPPORTS
	SÉQUENCE 1: Sources d'interruption	Activité N° 2	CD1.7	<ul style="list-style-type: none"> - Ressources numériques. - Logiciels. - Microcontrôleurs. - Maquettes didactiques. - Ordinateurs. - Vidéoprojecteur. - Etc...
		Activité N° 3		
		Activité N° 4		
		Activité N° 5		
	SÉQUENCE 2: Timers	Activité N° 2	CD2.7	
		Activité N° 3		
		Activité N° 4		
	SÉQUENCE 3: C.A.N	Activité N°2	CD3.7	
		Activité N°3		
Activité N° 4				
<ul style="list-style-type: none"> • Intégration : <ul style="list-style-type: none"> - Je réinvestis. 	J'intègre mes acquis et je réalise mon P.C.E (Projet Commun Encadré)			



LIENS DES RESSOURCES NUMÉRIQUES:



Evaluation
TH1_seq1



Evaluation
TH1_seq2



Evaluation
TH1_seq3



Annexe
MikroC



Projet
n°01



Cours
TH1

D'autres solutions programmées, comme il est indiqué dans le programme de technologie, seront intégrées dans d'autres séquences d'apprentissage lors de l'étude des différents thèmes tels que :

- **THÈME2** : Logique combinatoire ;
- **THÈME3** : Logique séquentielle ;
- **THÈME4** : Amplificateur linéaire intégré ;
- **THÈME6** : Moteur à courant continu.

Séquence 1

SOURCES D'INTERRUPTION



Cours TH1



Vidéo1 TH1_seq1

ACTIVITÉ 1 : Situation déclenchante

J'observe l'objet d'apprentissage

Des statistiques faites dans le monde entier montrent que le nombre de cambriolages (vols) augmente de manière considérable d'une année à une autre.

L'une des solutions possibles pour assurer la sécurité d'un logement, de ses occupants et de tout ce qu'il contient et réduire le risque d'être victime d'un vol, est l'installation d'un système d'alarme. Ce dernier permet de prévenir des intrusions. Il assure une surveillance permanente soit en présence ou non du propriétaire.


Problématique : Comment éviter les vols dans des locaux ?

Idée n° 1 : Présentation de l'idée de départ.

Le rôle du système d'alarme, objet d'étude, est de déceler une intrusion. Il joue un rôle dissuasif (décourageant) en déclenchant une sirène, ce qui peut paniquer (trembler) l'intrus.


Idée n° 2 : Constituants du système d'alarme (figure 1).

Les constituants essentiels du système d'alarme sont :

- Les capteurs permettant de déceler une intrusion ;
- l'unité centrale qui gère les informations issues des capteurs et de la console ;
- la console (clavier tactile) permettant à l'utilisateur de communiquer avec l'unité centrale ;
- la sirène de dissuasion ;
- les voyants de signalisation : mode_veille et voyant_flash.


Figure 1

Idée n° 3 : Fonctionnement du système d'alarme.

L'opérateur définit le mode de fonctionnement par l'intermédiaire de la console :

- Mode veille (système actif).
- Mode hors service.

En mode veille (icône «Armer» sélectionnée (**icone_armer = 1**)):

- * Les capteurs surveillent constamment les locaux.
- * Le système exécute l'algorithme 1 ci-dessous :



```

Déclaration des variables
Fonction principale du programme
Début
Configuration des registres
Initialisation des sorties
Boucle infinie
  Début
    Voyant_flash est désactivé
    Si icone_armer est sélectionnée alors
      voyant mode_veille s'allume
    Sinon voyant mode_veille est désactivé
  Fin boucle
Fin fonction principale
  
```

Algorithme 1

Lors de la détection d'une intrusion, le capteur informe l'unité centrale. Celle-ci enclenche la **sirène** et active le clignotement du **voyant_flash**. Le bruit déclenché par la sirène dissuade l'intrus qui prend alors la fuite. L'intervention de l'utilisateur sur les lieux permet éventuellement de dissuader l'intrus.

Lors d'une détection d'intrusion, le système exécute l'algorithme 2 ci-dessous :

```

Déclaration d'une fonction d'interruption
Début
Tant que (icone_armer est actionnée) faire
  Début
    Allumer voyant mode_veille
    Clignoter le voyant_flash
    Activer sirene
  Fin tant que
  Eteindre voyant mode_veille
Fin fonction d'interruption
  
```

Algorithme 2

Après la détection d'une intrusion, l'opérateur peut, par l'intermédiaire de la console, arrêter le fonctionnement de la sirène et désactiver le voyant (**voyant_flash=0**) (deuxième action sur l'icône armer (**icone_armer =0**)) et mettre le système en mode hors service.

Remarques:

- Lors de l'activation du mode veille, l'opérateur dispose d'un laps de temps prédéfini (30 secondes) pour quitter le local sans déclencher la sirène.
- Lorsque l'opérateur retourne sur les lieux sans qu'il n'y ait eu d'intrusion étrangère, celui-ci intervient sur la console (en un laps de temps défini) pour mettre le système **hors service**.

Ces deux remarques ne sont pas tenues en considération dans les programmes à envisager dans les activités de l'interruption. La programmation d'une attente de durée prédéfinie fera l'objet d'une étude ultérieure (lors de l'étude du TIMERO en mode temporisateur).

- Dans cette étude, le système d'alarme est géré par une carte électronique à base du microcontrôleur **16F876A**. Ce dernier intègre un convertisseur analogique numérique et présente 22 broches configurables en entrées/sorties. Ce qui couvre nos besoins.
 - Une action sur l'icône «Armer» met le système d'alarme en mode veille.
 - Une deuxième action sur l'icône «Armer» met le système en mode hors service.



J'analyse la situation



En petits groupes :

1

Compléter les questions suivantes par les termes:



interruption ; rôle ; différence ; déroule.

- Quelle est la entre une fonction principale et celle d'une?
- Quel est le de la fonction d'interruption?
- Comment se une fonction d'interruption?

2

Compléter les hypothèses (H1, H2 et H3) suivantes par les termes :
déroule ; principal ; fonction ; secondaire ; l'appel.

H1	L'algorithme 1 décrit le fonctionnement du système.
H2	L'algorithme 2 décrit le fonctionnement du système.
H3	Le programme se normalement. Lors d'une intrusion, l'appel de la secondaire se fait à partir de la fonction principale par de son nom.

3

Proposer d'autres hypothèses si elles vous paraîtront nécessaires.



.....

4

Le plan d'action, ci-dessous, aide à la recherche des réponses argumentées aux questions posées et à la vérification des hypothèses.

- Réalisation des activités, recherches d'informations, cours, ... ;
- vérification des hypothèses ;
- formulation des nouveaux savoirs et évaluations.

ACTIVITÉ

2

: Mise en œuvre d'une interruption par RB0 (INT)



On donne, à la figure 2, le schéma de simulation relatif au système d'alarme étudié.



Aide à l'activité 2



Annexe MikroC

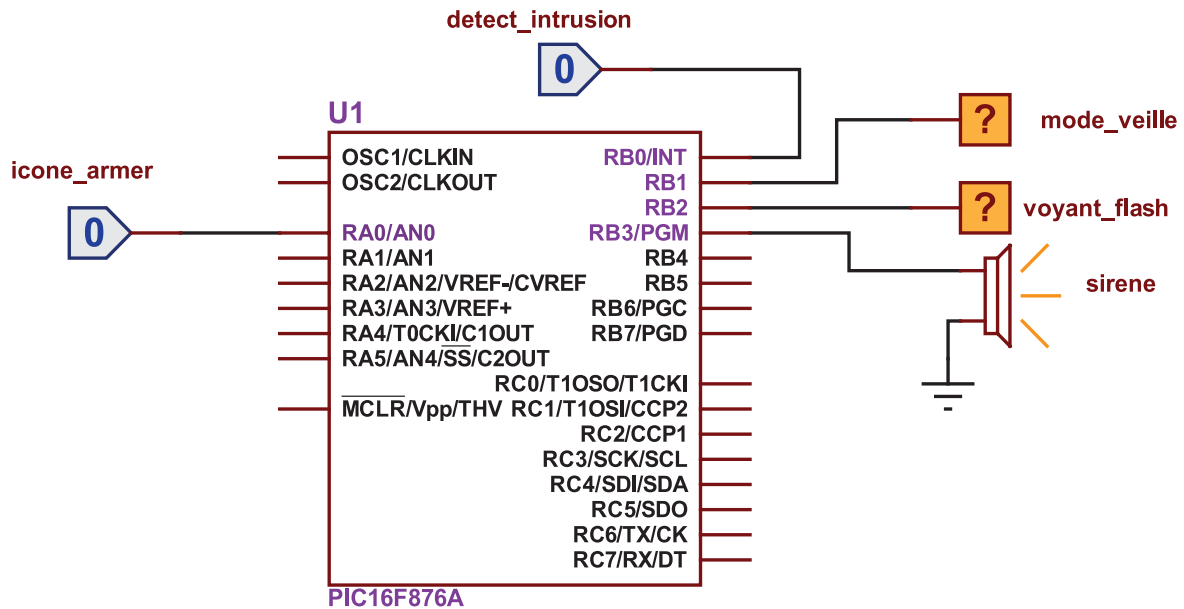


Figure 2



En petits groupes :

1. En se référant à la description du fonctionnement du système, à la figure 2 et à l'algorithme 1 :
- 1.1. Compléter le programme ci-dessous par les instructions convenables afin d'assurer le fonctionnement du système d'alarme en mode veille.

■ Programme :

```

// Fonctionnement du système en mode veille
sbit icone_armer at.....; // Déclarer variables
sbit mode_veille at .....;
.....at PortB.B2;

void main() // Fonction principale
{ // Début fonction principale
trisA=0x.....; trisB=0x.....; ADCON1=0x86; // Configurer et initialiser
PortB=.....;
Sound_Init(&PORTB, 3);

// Corps du programme
while (1) {
    voyant_flash =.....;
    if (.....) mode_veille = ... ;
    ..... mode_veille =.....;
}

// Fin fonction principale
}

```

- 1.2. Compiler ce programme et tester son fonctionnement à l'aide d'un logiciel de simulation.

2. En se référant à la description du fonctionnement du système et à l'algorithme 2, compléter la fonction ci-après par les instructions convenables afin d'assurer le fonctionnement du système d'alarme lors de la détection d'une intrusion.

■ Fonction :

```
// Fonctionnement du système d'alarme lors de la détection d'une intrusion

void interrupt() { // Déclarer fonction d'interruption
                  // Corps de la fonction
    while (.....==.....) {
        mode_veille = .....;
        voyant_flash = ! .....;
        Sound_Play( 2000, 1000);
    }
    mode_veille = .....;
}

```

3. On désire commander le système d'alarme en tenant compte du fonctionnement en mode veille et en cas de détection d'une intrusion.

3.1. Proposer une solution et tester son fonctionnement.



3.2. Obtient-on le fonctionnement désiré?



3.3. On donne ci-dessous le programme 1 complet :

■ Programme 1:

```
// Fonctionnement du système en mode veille

sbit icone_arter at PortA.B0;
sbit mode_veille at PortB.B1; // Déclarer variables
sbit voyant_flash at PortB.B2;

void main() { // Début du programme
    trisA = 0xff; trisB = 0x01; /* Configurer et initialiser
    ADCON1 = 0x86;
    INTCON = 0x90 ; Instruction ①
    PortB = 0;
    Sound_Init(&PORTB, 3);
    while (1) { /* Corps du programme
        PORTC = INTCON ; Instruction ②
        voyant_flash = 0;
        if (icone_arter == 1)
            mode_veille = 1;
        else mode_veille = 0;
    }
}

// Fonctionnement du système d'alarme lors de la détection d'une intrusion
void interrupt() { /* Fonction interrupt
    while (icone_arter == 1)
    {
        PORTC=INTCON ; Instruction④
        mode_veille = 1;
        voyant_flash = ! voyant_flash;
    }
}

```

* Tous droits réservés au Centre National Pédagogique

```

Sound_Play( 2000, 1000);
}
mode_veille = 0;
INTCON = 0x90 ;
}

```

Instruction ③ **/*

Etape 1:

Saisir le programme 1 ci-dessus à l'aide du compilateur MikroC (sans tenir compte des instructions ②, ③ et ④) et le compiler.

- Simuler son fonctionnement à l'aide d'un logiciel de simulation.
- Compléter les phrases suivantes par les termes :
déroule ; exécute ; ne revient pas ; principal ; saute.

Le programme principal se normalement.

Une interruption survient (intrusion), le microcontrôleur vers la routine d'interruption et les instructions mais il au programme

- Conclure sur le fonctionnement du système.

**Etape 2:**

- Compiler le programme 1 à l'aide du compilateur MikroC (sans tenir compte des instructions ② et ④).
- Simuler sous logiciel de simulation son fonctionnement.
- Sur maquette d'expérimentation ZITOPIC ou autres, vérifier le fonctionnement du système.
- Compléter les phrases suivantes par les termes:
exécute ; déroule ; revient ; principal ; saute.

Le programme principal se normalement.

Une interruption survient (intrusion), le microcontrôleur vers la routine d'interruption et ses instructions puis il au programme

- Conclure sur le fonctionnement du système.

**ACTIVITÉ 3 : Fonction interruption**

1. Compléter les phrases suivantes par les termes:
exécuter ; routine d'interruption ; interruption ; INTCON ; l'arrêt.

Aide à l'activité 3

Une est un évènement qui provoque d'un programme en cours d'exécution pour aller un autre programme appelé

Le registre permet de contrôler les interruptions.

2. Cocher les réponses correctes.

La détection d'un front actif sur l'entrée d'interruption (**RB0/INT**) d'un microcontrôleur provoque :

- l'arrêt définitif du programme en cours ;
- une pause dans l'exécution du programme principal ;
- l'appel d'une fonction spécifique qui gère les évènements extérieurs.

3. Préciser le type de la source d'interruption utilisée dans l'activité 2. (interne ou externe).

.....

ACTIVITÉ 4 : Configuration des bits du registre INTCON



Aide à l'activité 4

1. A l'aide d'un logiciel de simulation, saisir le schéma de la figure 3.

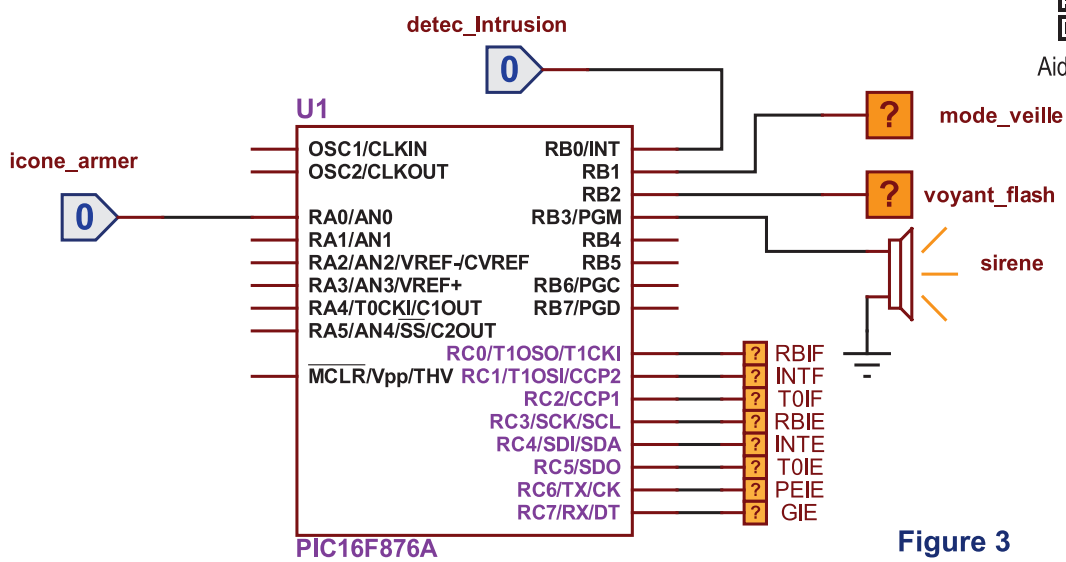


Figure 3

2. Reprendre le programme 1 et le compiler en tenant compte des instructions ①, ②, ③ et ④ et en configurant le registre TRISC. Simuler le programme puis compléter les tableaux suivants afin de déterminer la valeur du registre INTCON.

Avant l'interruption :

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	INTCON
...	(.....) ₁₆

Au cours de l'interruption :

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	INTCON
...	(.....) ₁₆

Après l'interruption :

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	INTCON
...	(.....) ₁₆

3. Identifier les bits du registre INTCON qui ont subi un changement d'état au cours de l'interruption.

.....

4. Par pair, récapituler les nouveaux savoirs



Compléter le paragraphe ci-dessous par l'un des mots suivants :

0x90 ; l'interruption INT; INTCON.B7 ; revenir ; INTF; INTCON.B1; entrée; GIE ; INTCON.GIE ; autre; 0x90 ; INTCON.INTF ; descendant.

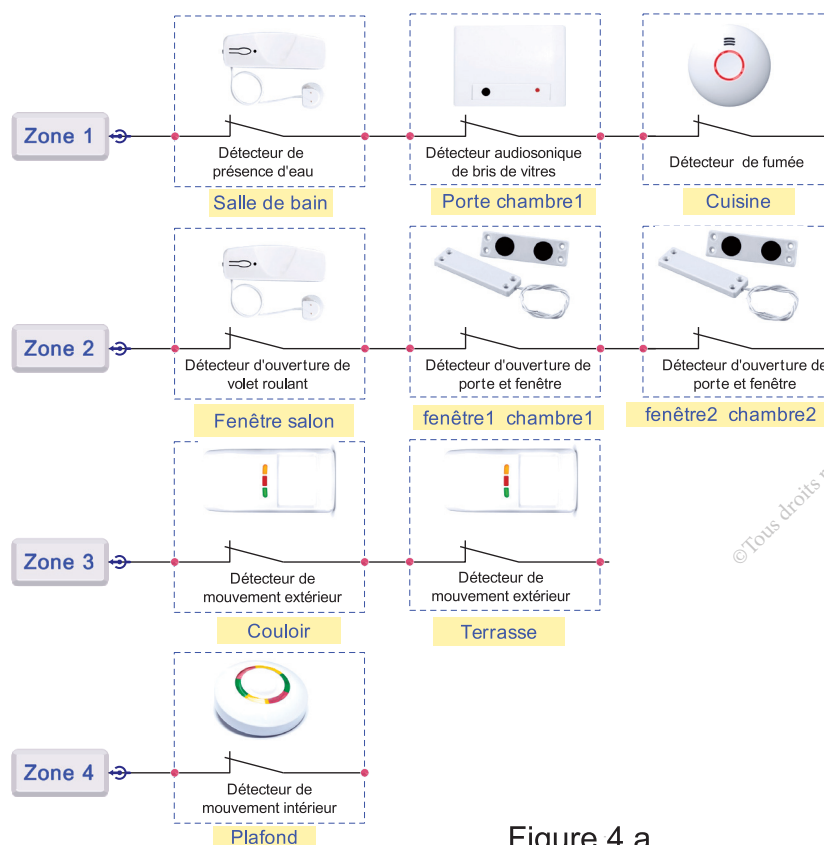
Pour pouvoir exploiter la source d'interruption externe INT (par RB0):

- ✎ La broche RB0 doit être configurée en
- ✎ le registre INTCON doit être configuré à au début du programme principal (pour activer
- ✎ à la fin de la routine d'interruption, on doit réactiver le bit du registre INTCON (pour autoriser une interruption) et désactiver l'indicateur (pour au programme principal). Cela revient à écrire à la fin de la routine d'interruption les deux instructions suivantes :
 (.....=1 ;=0) ou bien
 (.....= 1 ;=0) ;
 ou tout simplement écrire l'instruction INTCON=
- ✎ L'interruption INT se produit sur un front de la broche RB0 si on configure le bit 6 du registre option_reg à 0: OPTION_REG.INTEDG = 0 ;

ACTIVITÉ 5 : Mise en œuvre d'une interruption par RBI



■ Le système d'alarme couvre en réalité quatre zones qui peuvent être contrôlées séparément. La figure 4.a représente les quatre zones à contrôler.



Aide à l'activité 5

© Tous droits réservés au Centre National Pédagogique

Figure 4.a

■ On donne à la figure 4.b les sources d'interruption du microcontrôleur 16F87XA .

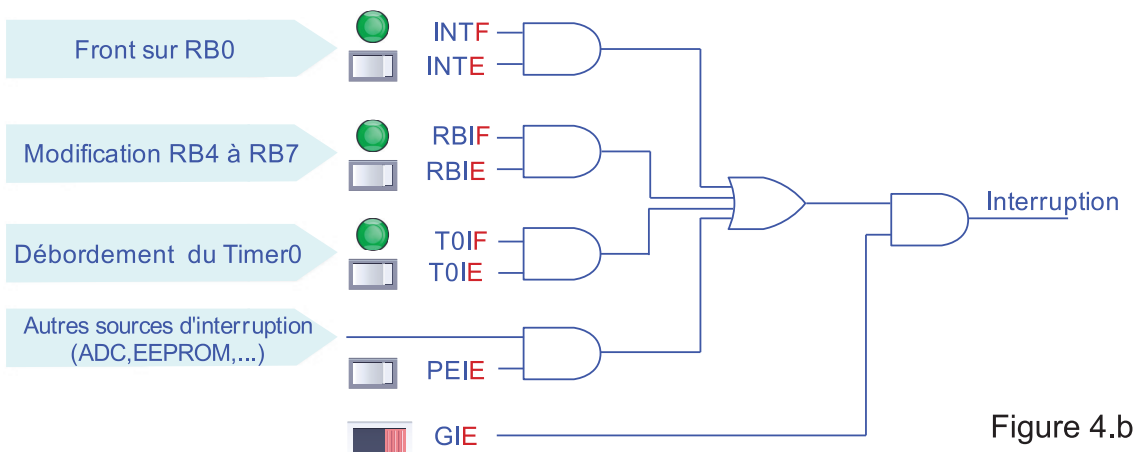


Figure 4.b

1. En se référant à la figure 4.b, citer :

1.1. Quelques sources d'interruptions internes.



1.2. Les sources d'interruptions externes.



2. Le système d'alarme couvre quatre zones pouvant être contrôlées séparément.

2.1. Préciser la source d'interruption qui pourrait être utilisée.



2.2. Sur la figure 5, apporter les modifications nécessaires au schéma de simulation de la figure 3 pour valider la solution proposée.

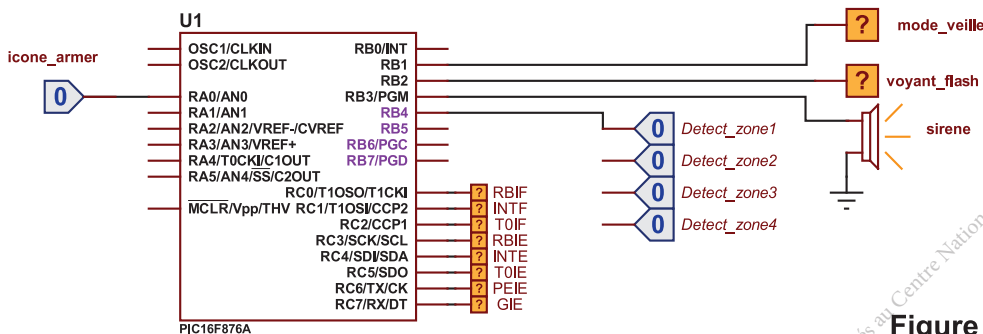


Figure 5

2.3. Compléter le tableau suivant afin de déterminer la valeur du registre INTCON relative à la solution proposée.

Avant l'interruption :

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	INTCON
....	(.....) ₁₆

Au cours de l'interruption :

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	INTCON
....	(.....) ₁₆

© Tous droits réservés au Centre National Pédagogique

✎ Après l'interruption :

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	INTCON
....	(.....) ₁₆

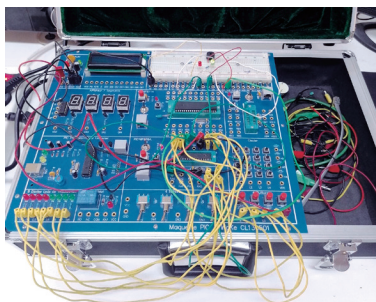
2.4. Apporter les modifications nécessaires au programme 1 pour valider la solution envisagée en complétant les instructions ⑤ et ⑥ dans le programme 2 suivant.

■ **Programme 2:**

```
// Fonctionnement du système en mode veille
sbit icone_arter at PortA.B0;
sbit mode_veille at PortB.B1;
sbit voyant_flash at PortB.B2;
void main() {
  TRISA=0xFF; TRISB=0xF0; TRISC=0x00; ADCON1=0x86;
  INTCON=..... ; // Instruction ⑤
  PORTB=0;
  Sound_Init(&PORTB, 3);
  while (1) {
    PORTC=INTCON ; // Affichage de la valeur du registre INTCON sur le PortC
    voyant_flash =0;
    if (icone_arter==1) mode_veille =1;
    else mode_veille =0;}
  }

// Fonctionnement du système d'alarme lors de la détection d'une intrusion
void interrupt() {
  while (icone_arter==1) {
    PORTC=INTCON ;
    mode_veille =1;
    voyant_flash=! voyant_flash;
    Sound_Play( 2000, 1000);
  }mode_veille =0;
  INTCON=..... ; // Instruction ⑥
}
```

2.5. Compiler le programme 2 et le simuler à l'aide d'un logiciel de simulation puis tester son fonctionnement sur maquette d'expérimentation.



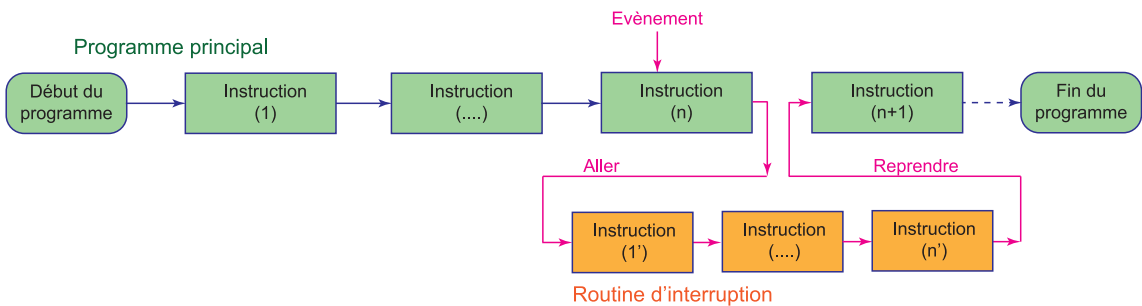


1 De quoi s'agit-il ?

Une interruption est un **évènement** qui provoque **l'arrêt** d'un programme en cours d'exécution pour aller exécuter un **autre programme** appelé programme (ou routine) d'interruption.

A la fin du programme d'interruption, le microcontrôleur **reprend** le programme principal à l'endroit où il s'est arrêté.

2 Déroulement d'une interruption



3 Quel est le registre à configurer ?

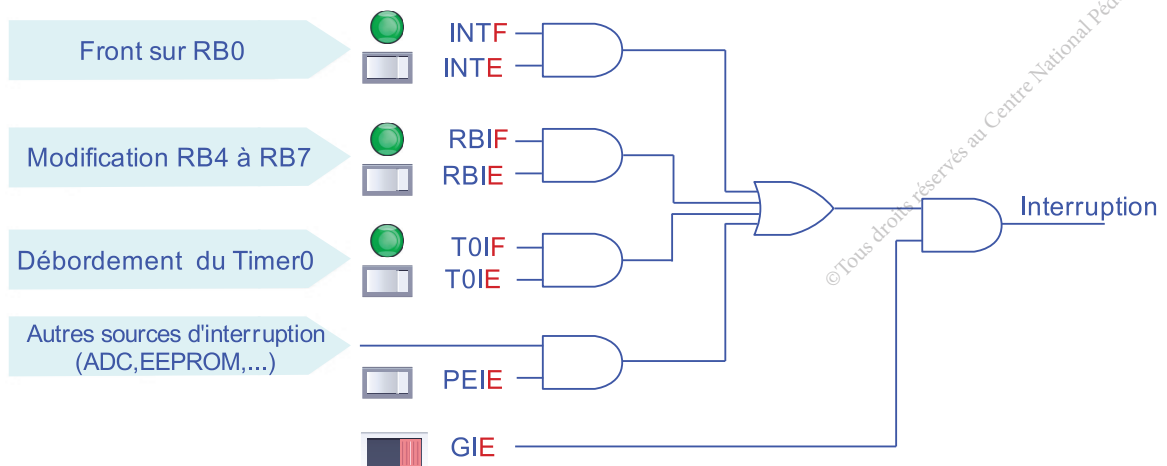
Registre de contrôle d'une interruption : INTCON

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

// Source d'interruption RB0
Intcon =0x90;

// Source d'interruption RBI
Intcon =0x88;

// Source d'interruption (RB0 et RBI)
Intcon =0x98;



Les sources d'interruption des microcontrôleurs de types PIC 16F87XA

© Tous droits réservés au Centre National Pédagogique



4 De quelle manière ?

Pour exploiter la source d'interruption externe **INT** :

- la broche **RB0** doit être configurée en **entrée** ;
- le registre **INTCON** doit être configuré à **0x90** au début du programme principal (pour activer l'interruption **INT**) et on doit remettre l'indicateur **INTF** à **zéro** et le bit **GIE** à **un** à la fin de la routine d'interruption pour revenir au programme principal et autoriser une autre interruption par **RB0**.

Pour exploiter la source d'interruption externe **RBI** :

- les broches **RB4**, **RB5**, **RB6** et **RB7** doivent être configurées en **entrées**.
- le registre **INTCON** doit être configuré à **0x88** au début du programme principal (pour activer l'interruption **RBI**) et on doit remettre l'indicateur **RBIF** à **zéro** et le bit **GIE** à **un** à la fin de la routine d'interruption pour revenir au programme principal et autoriser une autre interruption par **RBI**.

5 Quelle est la structure d'un programme contenant une interruption INT ?

```

//Déclaration des variables
// Déclaration d'une fonction d'interruption
void interrupt() {
    ..... // Instructions
    intcon = 0x90; }

// Fonction principale
void main() {
    .....// Configurations des registres
    trisB=0xFF; intcon = 0x90;
    ..... // Instructions
}

```

6 Où cela se produit ?



Système d'alarme anti-intrusion



Appel téléphonique lors d'un jeu vidéo



Détecteur optique de flamme



Evaluation TH1_seq1

Séquence 2

TIMERS



Cours TH1

ACTIVITÉ 1 : Situation déclenchante



Je cherche une solution

Lors de l'activation du mode veille du système d'alarme, l'opérateur dispose d'un laps de temps prédéfini (30s) pour quitter le local sans déclencher la sirène.

Lorsque l'opérateur retourne à son local sans qu'il n'y ait eu d'intrusion étrangère, il intervient sur la console (en un laps de temps défini) pour mettre le système hors service.



Problématique : Existe-t-il un module intégré dans le microcontrôleur qui permet d'obtenir une temporisation prédéfinie ?



J'analyse la situation



En petits groupes :

- Compléter les questions suivantes par les termes : la fonction ; temporisation ; besoin ; programmer.
 - Comment programmer une.....?
 - De quoi a-t-on.....pourune temporisation?
 - Comment réalisertemporisation?



- Compléter les hypothèses (H1 et H2) suivantes par les termes : temporisation; TIMER0; réaliser; OPTION_REG.

H1	Le peut être utilisé pour une temporisation.
H2	La configuration du registre..... est nécessaire pour programmer une

- Proposer d'autres hypothèses si elles vous paraîtront nécessaires.

.....

.....

.....

.....

© Tous droits réservés au Centre National Pédagogique

- 4 Le plan d'action, ci-dessous, aide à la recherche des réponses argumentées aux questions posées et à la vérification des hypothèses.
- Réalisation des activités, recherches d'informations, cours,... ;
 - vérification des hypothèses ;
 - formulation des nouveaux savoirs et évaluations.

ACTIVITÉ 2 : Détermination de la durée d'un cycle de comptage

La durée d'un cycle de comptage du TIMER0 en mode temporisateur est donnée par la formule 1 ou par l'application PIC Timer Calculator :

Formule 1:
$$T_c = \frac{(256 * DIV) * 4}{Fosc}$$

Avec : **DIV** : valeur de prédiviseur
Fosc : fréquence de l'oscillateur
Tc : temps écoulé pour un cycle



Aide à l'activité 2

PIC Timer Calculator (PTC):

Cette application permet de calculer la durée d'un cycle de comptage du TIMER0, la fréquence et la configuration du registre OPTION_REG.

On donne ci-dessous, le schéma de simulation (figure 1), l'algorithme 1 relatif au calcul et à l'affichage de la durée d'un cycle de comptage et le programme en MikroC incomplet.

■ Schéma de simulation :

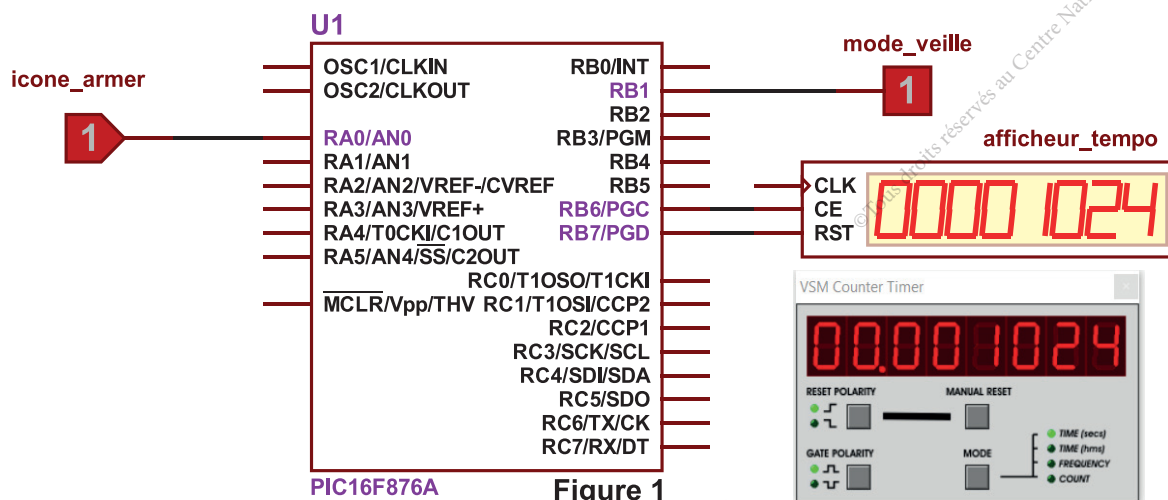


Figure 1

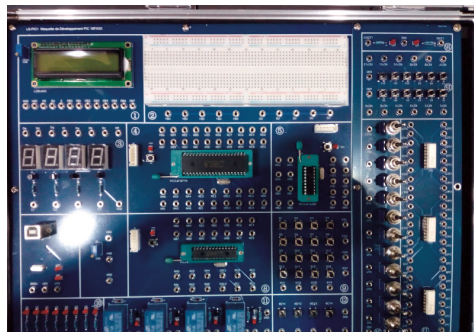
■ **Algorithme 1 :**

Algorithme 1

```

Déclaration des variables
Déclaration d'une fonction principale
Début
Configuration des registres
Initialisation des variables et des registres
Boucle infinie
  Début
    Si (icone_arder=0)
      Début
        Initialisation du module VSM counter TIMER
        Initialisation du TMR0 à 0
        Mise à zéro de l'indicateur T0IF
        Désactivation du voyant mode_veille
        Mise à zéro de la variable interne tempo
      Fin si
    Si ((icone_arder=1) et (tempo=0))
      Début
        Démarrage du module VSM counter TIMER
    Si (l'indicateur T0IF=1) //Détection de la fin de cycle de comptage
      Début
        Arrêt du module VSM counter TIMER
        Initialisation du TMR0 à 0
        Mise à zéro de l'indicateur T0IF
        Activation du voyant mode_veille
        Inhibition du comptage d'un nouveau cycle
      Fin si
    Fin si icone_arder
  Fin boucle infinie
Fin fonction principale
  
```

1. Traduire l'algorithme 1 en langage MikroC (programme 1) en complétant les instructions manquantes puis le simuler.



© Tous droits réservés au Centre National Pédagogique

■ Programme 1 :

```
# define icone_armer .....
# define mode_veille.....
bit tempo;
void main() {
  adcon1=0x87; trisb=0x00; trisA =0x01;
  icone_armer=0; tempo=0;
  Portb=0;
  TMR0=0;
  option_reg=0x08;
  while(1) {
    if (icone_armer==0) {
      rb6_bit=0; rb7_bit=1;
      TMR0=.....; INTCON.T0IF=.....;
      Mode_veille=.....; tempo=.....; }

    if ((icone_armer==1).....(tempo==0)) {
      rb6_bit=1; rb7_bit=0;
      if(INTCON.T0IF==.....) {
        rb6_bit=0;
        tmr0=.....; INTCON.T0IF=.....;
        mode_veille =.....; Tempo=1; } }
  }
}
```

2. Pour différentes pré-divisions (DIV) et pour une fréquence d'oscillateur $f_{osc} = 4\text{MHz}$, compléter le tableau ci-dessous par :

- La valeur du registre option_reg ;
- la valeur de la durée T_c (**Tc_theo**) en appliquant la formule 1 ou l'application (PTC);
- la valeur de la durée T_c (**Tc_prat**) obtenue par simulation après avoir modifié à chaque fois la valeur du registre option_reg dans le programme 1.

DIV	1	2	4	8	16	32	64	128	256
Option_reg	0x08	0x01	0x03	0x05	0x06
Tc_theo(μs)	256	1024	4096	16384	32768
Tc_prat(μs)	259	1024	4099	16384	32764

3. Dédurre l'intérêt du programme ainsi simulé.



4. Par pair, récapituler les nouveaux savoirs



- ✎ Le TIMER0 compte de à (compteur modulo) puis recommence à 0, on parle dans ce cas de
- ✎ Tout du TMR0 (passage de à 0) entraîne le positionnement (mise à 1) de l'indicateur T0IF du registre
- ✎ L'effacement (mise à 0) du bit se fait par programmation.

ACTIVITÉ 3 : Programmation d'une temporisation de 0,5s

La période de clignotement du voyant « mode_veille » est d'une seconde (1s) : le voyant s'allume pendant 0,5s et s'éteint pendant 0,5s. On prend une pré-division DIV = 256 du TIMER0 et une fréquence $f_{osc} = 4\text{MHz}$.



Aide à l'activité 3

Le TIMER0 est initialisé à zéro : TMR0 = 0.

1. Sachant qu'un cycle de comptage dure $65536\mu\text{s}$, calculer le nombre d'itérations « i » nécessaires pour obtenir une temporisation de 0,5s.



.....

.....

.....

.....

2. On donne ci-dessous, le schéma de simulation (figure 2) et l'algorithme partiel 2 relatif à la détermination et à l'affichage de la durée d'une temporisation de 0,5s.

■ Schéma de simulation :

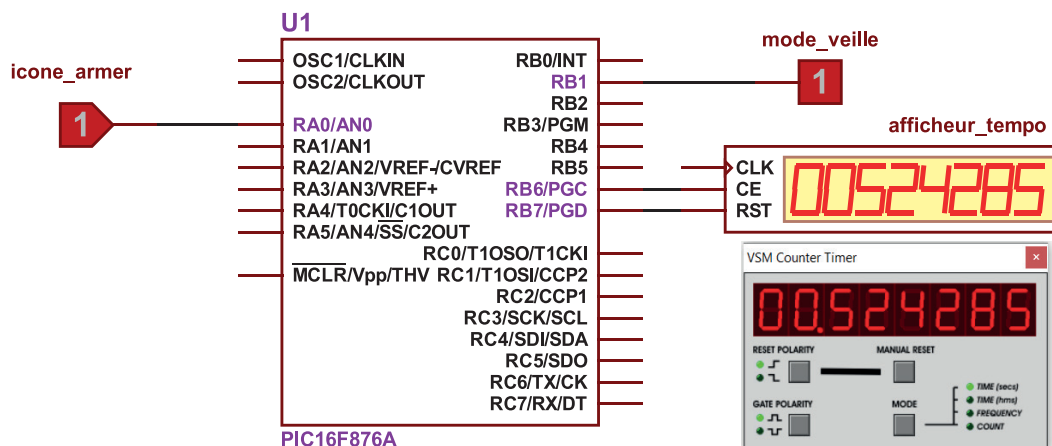


Figure 2

■ Algorithme partiel 2 : Algorithme relatif au comptage du nombre d'itérations « i ».

```

Si (l'indicateur T0IF=1)
  Début
  Incrémentation de la variable i
  Remise à zéro le l'indicateur T0IF
  Fin si
  
```

```

Si (i=8) // Nombre d'itérations nécessaires correspondant à 0.5s
  Début
  Mise à 1 de la variable Tempo
  Arrêt du module VSM
  Activation du voyant mode_veille
  Initialisation de la variable i à 0
  Fin si
  
```

Algorithme partiel 2 :

© Tous droits réservés au Centre National Pédagogique

2.1. Compléter dans le programme 2 ci-dessous, les instructions manquantes relatives à l'algorithme partiel 2 afin d'allumer le voyant mode_veille après 0,5s de l'activation de l'entrée icone_armer.

■ **Programme 2:**

```
# define icone_armer PortA.B0
# define mode_veille PortB.B1
char i;
bit tempo;
void main() {
    adcon1=0x87; trisb=0x00; trisa=0x01;
    option_reg=0x07;
    icone_armer=0; tempo=0; i=0;
    Portb=0;
    TMR0=0;
    while(1==1) {
        if (icone_armer==0) {
            rb6_bit=0; rb7_bit=1; // Initialiser le module VSM
            TMR0=0; INTCON.TOIF=0;
            mode_veille=0; tempo=0; }
        if((icone_armer==1)&&(tempo==0)) {
            rb6_bit=1; rb7_bit=0; // Démarrer le module VSM
            if(INTCON.TOIF==1) {
                i=.....;
                .....; }
            if (i ==.....) {
                Tempo=1;
                rb6_bit=0;
                mode_veille =.....;
                .....; }
            } } }
```

2.2. Compiler ce programme, simuler son fonctionnement à l'aide d'un logiciel de simulation puis le tester sur maquette d'expérimentation.

ACTIVITÉ 4 : Programmation d'une temporisation de 30s

1. Calculer le nombre d'itérations « j » pour obtenir une temporisation de 30s nécessaire pour que le propriétaire quitte le local sans déclencher l'alarme.

La programmation de la temporisation est assurée par le TIMER0 en mode temporisation.

$T_j = 30s$ pour $F_{osc} = 4MHz$ et $DIV = 256$.

On suppose qu'initialement $TMR0 = 0$.



Aide à l'activité 4



2. On donne ci-dessous, le schéma de simulation (figure 3) et l'algorithme partiel 3 relatif à la détermination et à l'affichage de la durée d'une temporisation de 30s.

■ **Schéma de simulation :**

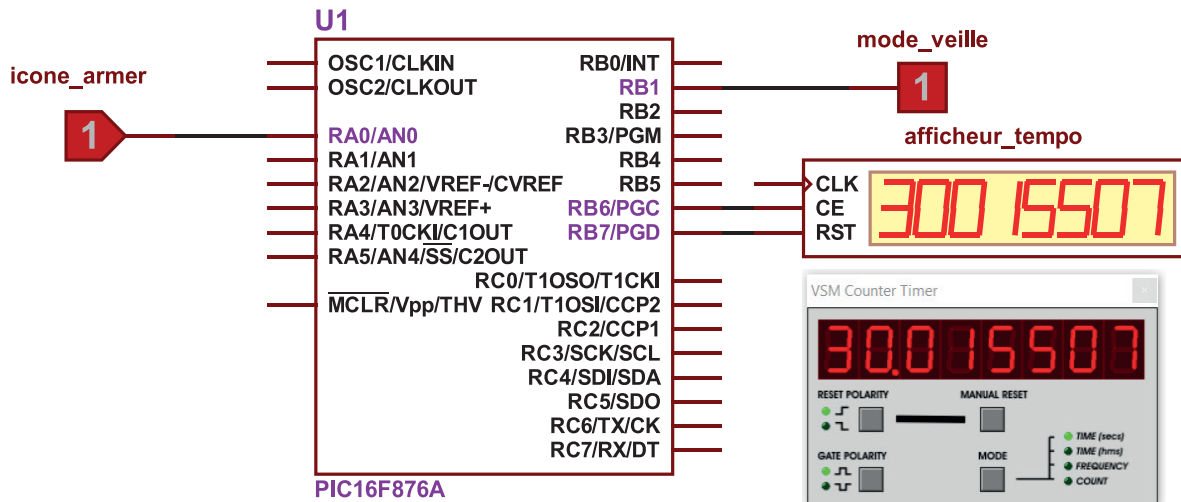


Figure 3

■ **Algorithme partiel 3: Algorithme relatif au comptage du nombre d'itérations «j».**

Si (l'indicateur T0IF=1)

 Début

 Incrémentation de la variable j

 Remise à zero le l'indicateur T0IF

 Fin si

Si (j =458) // Nombre d'itérations nécessaires correspondant à 30s

 Début

 Mise à 1 de la variable Tempo

 Arrêt du module VSM

 Activation du voyant mode_veille

 Initialisation à 0 de la variable j

 Fin si

Algorithme partiel 3

© Tous droits réservés au Centre National Pédagogique

2.1. En se référant à l'algorithme partiel 3 et aux commentaires proposés, compléter le programme 3 en MikroC : si l'entrée icone_armer est actionnée, le voyant mode_veille clignote à une période de 1s pendant une durée de 30s. Après la temporisation de 30s, le voyant mode_veille reste allumé.

■ Programme 3 :

```

# define icone_arder PortA.B0
# define mode_veille PortB.B1
char i;
.....; // Déclaration d'une variable j de type entier non signé
bit tempo;
void main() {
  icone_arder=0; tempo=0;
  trisa =0x01; Portb=0; TMR0=0; adcon1=0x87; trisb=0x00; option_reg=0x07;
  while(1) {
    if (icone_arder==0) { rb6_bit=0; rb7_bit=1; // Initialiser le module VSM
      tempo=0; TMR0=0; mode_veille=0; }
    if ((icone_arder==1)&&(tempo==0))
      { rb6_bit=1; rb7_bit=0; // Démarrer le module VSM
    if(INTCON.T0IF==1) { i=i+1; j=.....; // Incréments j
      INTCON.T0IF=0; }
    if(i==8) {
      mode_veille=.....; // Clignoter le voyant mode_veille
      i=0; }
    if(j ==.....) { // Nombre d'itérations nécessaires correspondant à 30s
      Tempo=1; rb6_bit=0; // Arrêter le module VSM
      mode_veille =.....; // Activer le voyant mode_veille
      .....; // Initialiser j à 0
    }
  }
}

```

2.2. Compiler ce programme, simuler son fonctionnement à l'aide d'un logiciel de simulation puis le tester sur maquette d'expérimentation.

3. Par pair, récapituler les nouveaux savoirs



Compléter les phrases suivantes:

* Pour programmer une temporisation, il faut:

☞ Configurer le registre en choisissant le fonctionnement du TIMER0 en mode et en fixant une pré-division.

☞ Déterminer le..... des itérations pour obtenir la temporisation désirée.

☞ Contrôler la fin du comptage en testant l'indicateur..... du registre INTCON.

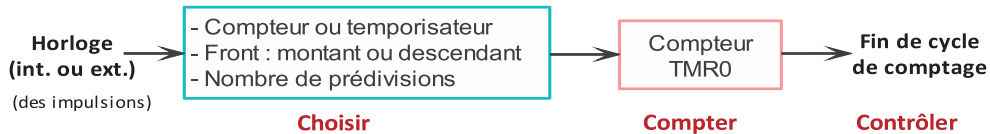
* Le choix d'une égale à 255 permet de minimiser le nombre des itérations.



1 De quoi s'agit-il ?

Le **TIMERO** est un module intégré dans les microcontrôleurs de type PIC. C'est un compteur à 8 bits assurant la fonction **comptage** ou la fonction **temporisation**.

2 Déroulement d'un cycle de comptage :



3 De quelle manière ?

Pour exploiter le module **TIMERO**, il faut:

- choisir le mode de fonctionnement (compteur ou temporisateur) :

* En mode compteur, le **TIMERO** **compte** le nombre d'impulsions survenues sur la broche **RA4/T0CKI** (le **TIMERO** utilise un signal d'horloge externe en provenance d'un capteur par exemple). Le nombre d'impulsions ainsi comptées représente le **nombre** d'évènements survenus.

* En mode temporisateur, le **TIMERO** utilise l'**horloge interne** du microcontrôleur en provenance d'un oscillateur relié à la broche **OSC1/OSC2** (oscillateur à quartz par exemple). Le nombre d'impulsions ainsi comptées représente **une durée** bien déterminée.

- Choisir un **front** montant ou descendant pour le fonctionnement en mode compteur;
- choisir une **pré-division**;
- contrôler le comptage à **chaque fin de cycle**.

4 Quel est le registre à configurer ?

Registre de configuration du **TIMERO** : **Option_reg**

Le registre **Option_reg** permet de choisir le **mode de fonctionnement**, le **front** et le **prédiviseur**. Ce choix se fait par programmation.

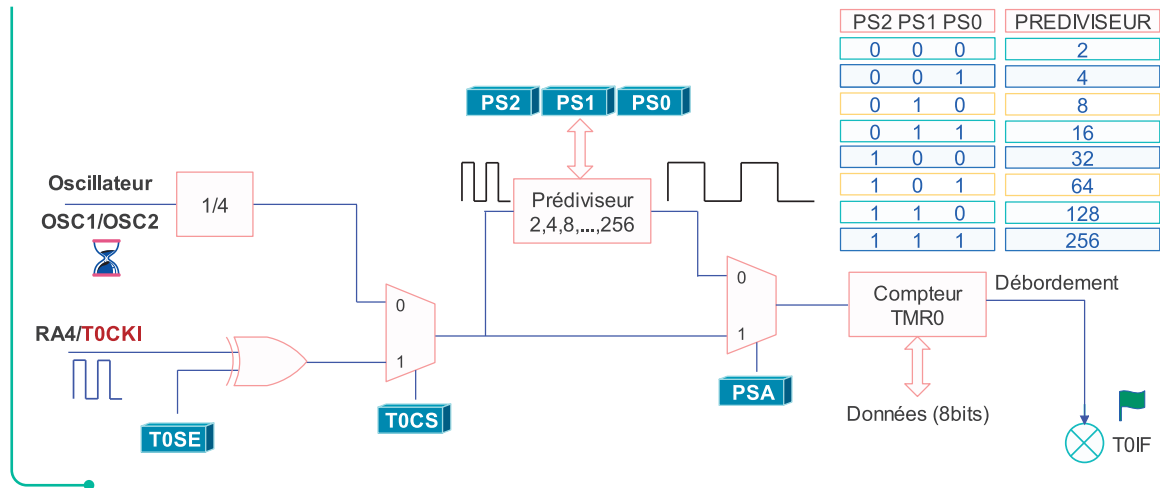
Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Choisir l'horloge :
0 : oscillateur
1 : RA4/T0CKI

Choisir un front :
0 : front montant
1 : front descendant

Choisir un prédiviseur :
PSA = 0 : avec prédivision
PSA = 1 : sans prédivision

Tous droits réservés au Centre National Pédagogique



5 Comment programmer une temporisation ?

Pour programmer une temporisation il suffit de :

- Configurer le registre *Option_reg* dans la partie configuration du programme principal ;
- déterminer le nombre d'itérations pour obtenir la temporisation désirée ;
- contrôler le comptage par l'exploitation du bit *TOIF* du registre *INTCON*.

6 Quelle est la structure d'un programme contenant une temporisation ?

```

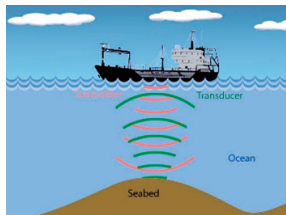
// Déclaration
Char i ; // Déclarer une variable i
// Fonction principale
void main {
// Configuration
Option_reg = 0x07 ; // Mode temporisateur avec une pré-division par 256
// Initialisation
i=0 ; // Initialiser le compteur des itérations
TMR0 =0 ; // Initialiser TMR0
while(1) {
if ( intcon.TOIF==1) { // Contrôler le comptage
intcon.TOIF=0 ; // Mettre l'indicateur TOIF à 0
i= i+1 ; // Incrémenter le compteur des itérations
}
if (i==.....) i=0 ; // Si on obtient le nombre d'itérations voulues on initialise i
}

```

7 Où cela se produit ?



Détecter la présence et déterminer la position ou la vitesse d'objets.



Mesurer la profondeur de l'océan.



Evaluation TH1_seq2

© Tous droits réservés au Centre National Pédagogique

Séquence 3

CONVERTISSEUR ANALOGIQUE NUMÉRIQUE (C.A.N)



Cours TH1

ACTIVITÉ 1 : Situation déclenchante



Je prends en compte l'une des structures d'accueil:

Lorsque l'utilisateur appuie sur une icône de l'écran tactile avec la pointe d'un stylet ou avec le doigt, deux tensions V_x et V_y se produisent. Le microcontrôleur identifie l'icône sélectionnée en déterminant ses coordonnées (X,Y) suite à la conversion des signaux V_x et V_y en grandeurs numériques. Une fois identifiée, le microcontrôleur tient en considération l'icône sélectionnée pour commander le système d'alarme.



Figure 1 : synoptique partiel de l'écran tactile

Principe de fonctionnement d'une dalle tactile : Dalle Tactile

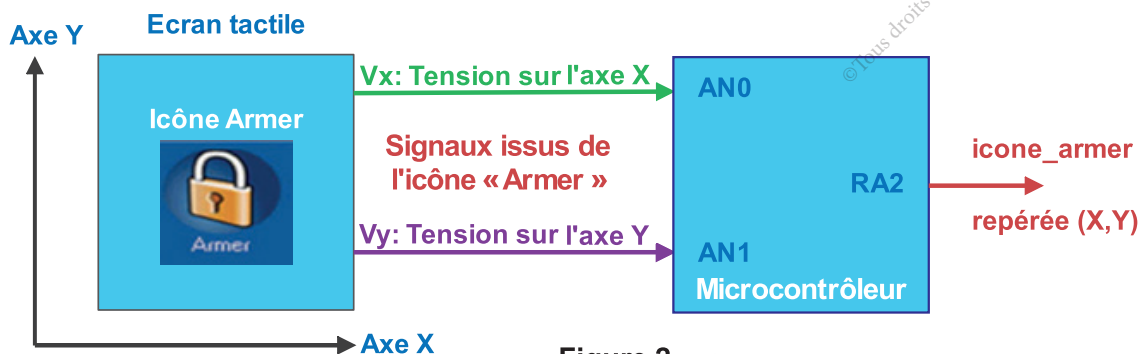


Figure 2



Problématique : Comment fonctionne le module Convertisseur Analogique Numérique ?



J'analyse la situation



En petits groupes :

1

Compléter les questions suivantes par les termes :
la conversion ; différence ; registre ; multiplexés.



- Quelle est la entre une entrée analogique et une entrée numérique?
- Comment s'effectueanalogique/numérique?
- Quels sont les portsdans un microcontrôleur?
- Y a-t-il un à configurer lors de la conversion analogique numérique?

2

Compléter les hypothèses (H1, H2 et H3) suivantes par les termes :
une bibliothèque ; conversion ; ADCON1.

H1	La A/N nécessite la configuration d'un registre approprié.
H2	Le registre..... permet la configuration des entrées analogiques et numériques de certains ports du microcontrôleur.
H3	le compilateur MikroC PRO offre prédéfinie pour effectuer la conversion.

3

Proposer d'autres hypothèses si elles vous paraîtront nécessaires.



4

Le plan d'action, ci-dessous, aide à la recherche des réponses argumentées aux questions posées et à la vérification des hypothèses.

- Réalisation des activités, recherches d'informations, cours,... ;
- vérification des hypothèses ;
- formulation des nouveaux savoirs et évaluations.

ACTIVITÉ 2 : Détermination des tensions issues des axes X et Y

Se référer à "Aide à l'activité 2" :



Aide à l'activité 2

1. En se référant au schéma structurel (figure 3), exprimer puis calculer la valeur de la tension V_x .



.....

.....

.....

.....

.....

.....

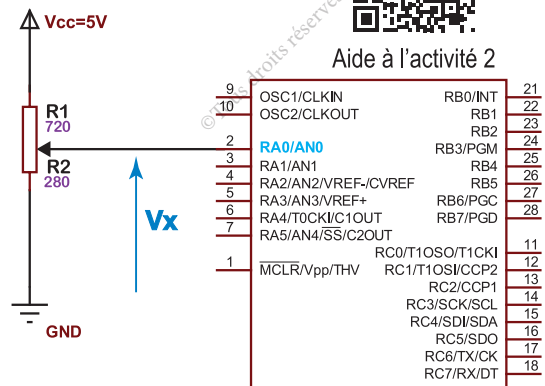


Figure 3

PIC16F876A

2. En se référant au schéma structurel (figure 4), exprimer puis calculer la valeur de la tension V_y .

.....

.....

.....

.....

.....

.....

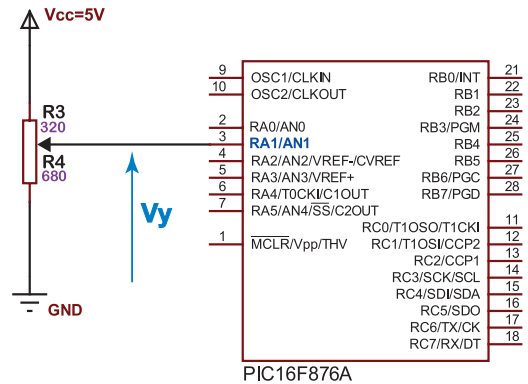


Figure 4

ACTIVITÉ 3 : Conversion Analogique/Numérique des tensions V_x et V_y



Aide à l'activité 3



En petits groupes :

Les tensions V_x et V_y , obtenues à la suite d'un appui sur l'icône « Armer », sont converties en valeurs numériques par le module Convertisseur Analogique Numérique (C.A.N) intégré dans le microcontrôleur. Ce convertisseur possède une résolution de 10 bits.

1. Etude du convertisseur analogique numérique (C.A.N)

On donne l'algorithme 1 et le schéma de simulation (figure 5) assurant la conversion d'une tension U_c issue d'un potentiomètre en signal numérique. Le résultat de la conversion est affiché en binaire sur les ports B et C.

Déclaration des variables

Fonction principale

Début

Configuration des registres

Initialisation du convertisseur

Boucle infinie

Début

// Lecture du résultat de la conversion de la tension U_c

N ← valeur lue par le C.A.N sur le canal 2

// Affichage du résultat de la conversion en binaire

PortB ← ADRESL

PortC ← ADRESH

Fin boucle

Fin fonction principale

Algorithme 1

© Tous droits réservés au Centre National Pédagogique

1.1. Traduire l'algorithme 1 ci-dessus en langage MikroC (programme1).

■ Programme 1:

```

unsigned int N;
void main() {
    TRISA = .....;
    TRISC = .....;
    TRISB = .....;
    ADC_init();
    ADCON1 = 0x80;
    while(1) {
        N = ADC_Get_Sample(.....);
        PORTB = .....;
        PORTC = .....;
    }
}
    
```

■ Schéma de simulation :

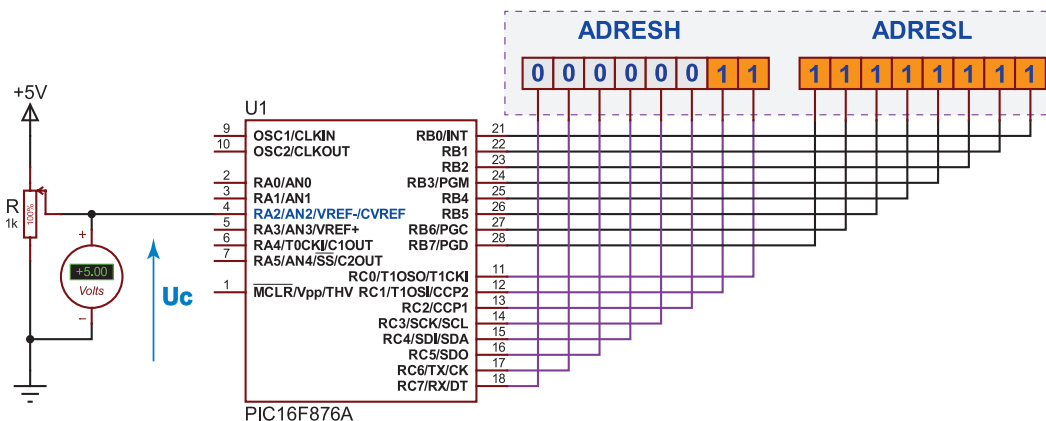


Figure 5

1.2. Saisir le programme sur le compilateur MikroC.

1.3. Simuler le montage de la figure 5 à l'aide d'un logiciel de simulation.

1.4. Agir sur le potentiomètre pour faire varier la tension U_c . Pour différentes valeurs de la tension U_c , relever le nombre N en binaire puis le convertir en décimal en complétant le tableau ci-dessous.

$U_c(V)$	0	1,40	2,50	3,40	5
$N_{(2)}$
N	1023

1.5. Exprimer N en fonction de U_c .

.....

.....

1.6. Fixer la tension U_c à 3,4V.

- Remplacer dans le programme 1, l'instruction $ADCON1=0x80$ par $ADCON1=0x00$.
- Compiler le programme et simuler le montage de la figure 5.
- Relever la valeur binaire du nombre N en complétant le tableau suivant.

ADCON1	ADRESH								ADRESL							
$(80)_{16} = (10000000)_2$	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0
$(00)_{16} = (00000000)_2$

2. Par pair, récapituler les nouveaux savoirs

Compléter les phrases ci-dessous par l'un des termes suivants :
analogique ; dix ; à droite ; convertir ; binaire ; à gauche ; ADRESL.



- ✎ Le Convertisseur Analogique Numérique (C.A.N) : Analogic Digital Converter (A.D.C) permet de la tension comprise entre 0 et 5V en un nombre binaire sur bits compris entre 0 et 1023.
- ✎ Le résultat de la conversion est enregistré dans deux registres ADRESH et de 8 bits chacun.
- ✎ Si ADFM = 1 (bit 7 du registre ADCON1= 1) alors le résultat de la conversion est justifié
- ✎ Si ADFM = 0 (bit 7 du registre ADCON1= 0) alors le résultat de la conversion est justifié

ACTIVITÉ 4 : Programmation d'une icône tactile

Lorsque l'icône « Armer » est actionnée, le système d'alarme est activé.
On donne:

- L'algorithme associé à ce mode de fonctionnement :

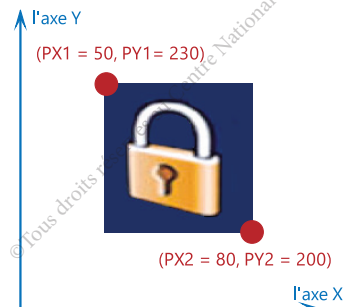
Algorithme

```
/* Si (icône « Armer » est actionnée) alors activer le mode veille
du système d'alarme sinon le système d'alarme est en mode hors service */
Si (icone_armer =1) alors mode_veille ← 1
sinon mode_veille ← 0
```

- La relation entre une tension V et son résultat de conversion N est :

$$N = 1023 \cdot V/5$$

- Les coordonnées "angle supérieur gauche" et "angle inférieur droit" de l'icône « Armer » de l'écran tactile.
- L'équation liant la valeur de la position du pixel Px et la valeur numérique Nx de l'axe X : $Px = Nx \cdot (240/1023)$
- L'équation liant la valeur de la position du pixel Py et la valeur numérique Ny de l'axe Y : $Py = Ny \cdot (320/1023)$



En petits groupes :

1. Dédurre de ce qui précède, les valeurs de Ny, Vy, Nx et Vx en complétant les tableaux suivants :

Px (pixel)	50	80
Nx
Vx (V)

Py (pixel)	200	230
Ny
Vy (V)

2. Compléter l'algorithme 2 permettant de détecter un appui sur icône «Armer».

■ **Algorithme 2 :**

Algorithme 2

```

Déclaration des variables
Fonction principale
Début
Configuration des registres
Initialisation des ports
Boucle infinie
  Début
    // Lecture du résultat de la conversion de la tension Vx
    Nx ← valeur lue par le C.A.N sur le canal 0
    // Lecture du résultat de la conversion de la tension Vy
    Ny ← valeur lue par le C.A.N sur le canal 1
    Px ← ..... // Calcul de Px
    Py ← ..... // Calcul de Py
    // Condition d'activation de l'icône « Armer »
    Si ((Px>.....) et (Px<.....)) et ((Py>.....) et (Py<.....)) alors icone_ armer ←.....
    sinon icone_ armer ←.....
  Fin si
  Fin boucle
Fin fonction principale
  
```

3. Traduire l'algorithme 2 en langage MikroC (programme 2) puis le saisir en exploitant le compilateur MikroC.

■ **Programme 2 :**

```

unsigned int Px,Py,Nx,Ny;
sbit icone_ armer at PortB.B0;
void main()
{
  TRISA = 0xFF; ADCON1 = 0x80;
  TRISB = 0;
  PortB=0;
  while(1) {
    Nx = .....;    Ny = .....;
    Px = (240.0/1023.0)*Nx;    Py = .....;
    if (( Px>50 ..... Px<80) ..... ( Py>200 .....Py<230)) icone_ armer =.....;
    else .....;
  }
}
  
```

© Tous droits réservés au Centre National Pédagogique

4. On donne le schéma de simulation (figure 6) relatif à cette application. Vérifier le fonctionnement du programme à l'aide d'un logiciel de simulation puis sur une maquette d'expérimentation.

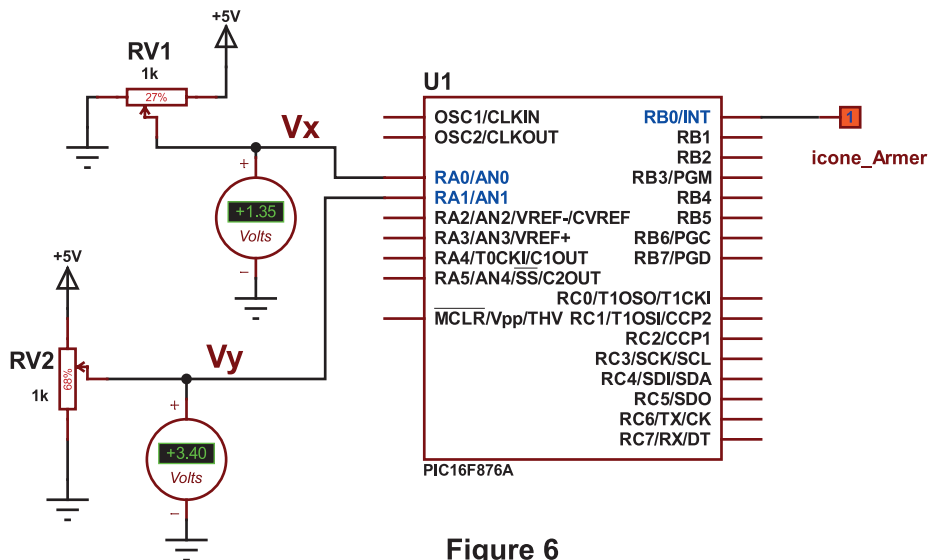
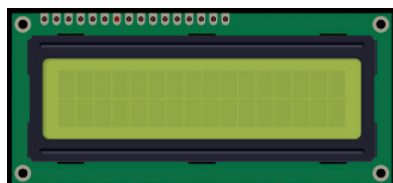


Figure 6

5. Les coordonnées de l'icône « Armer » de l'écran tactile peuvent être affichées par un afficheur LCD. On donne le schéma de simulation (figure 7) relatif à cette application. Compléter le programme 3 ci-après afin d'afficher les coordonnées de l'icône « armer » et de commander la sortie « icone_armer ».



LCD 16x2



Aide à l'activité 4

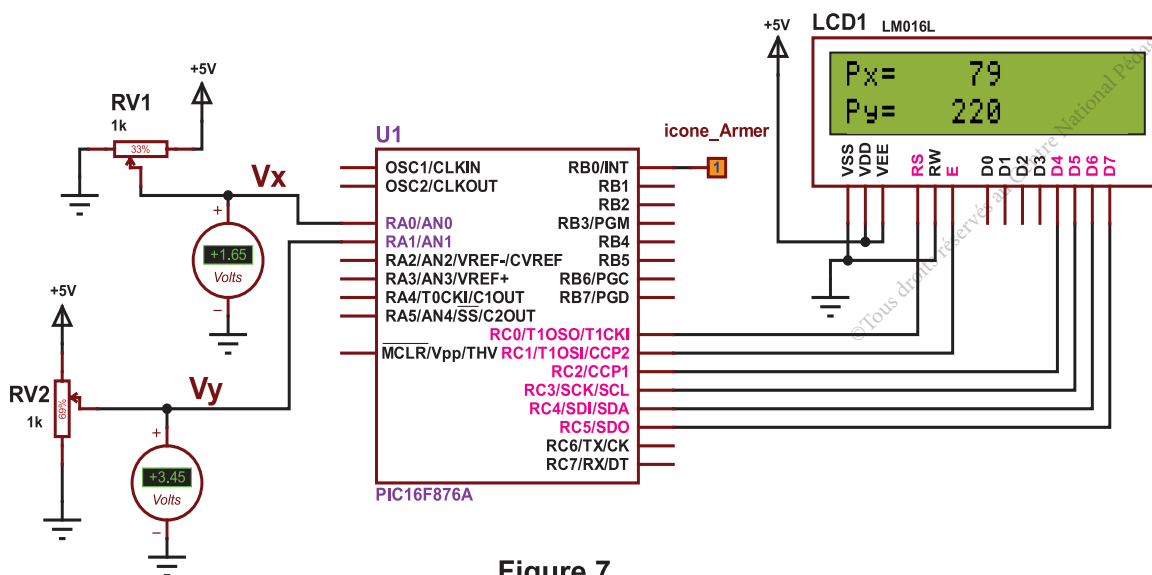


Figure 7

■ **Programme 3 :**

```

int Px,Py,Nx,Ny;
char Pxtxt[7];
char Pytxt[7];
sbit icone_Armer at PortB.B0;
// Connexion des broches de Lcd
sbit LCD_RS at RC0_bit;
sbit LCD_EN at RC1_bit;
sbit LCD_D4 at RC2_bit;
sbit LCD_D5 at RC3_bit;
sbit LCD_D6 at RC4_bit;
sbit LCD_D7 at RC5_bit;
// Configuration du registre TRISC
sbit LCD_RS_Direction at TRISC0_bit;
sbit LCD_EN_Direction at TRISC1_bit;
sbit LCD_D4_Direction at TRISC2_bit;
sbit LCD_D5_Direction at TRISC3_bit;
sbit LCD_D6_Direction at TRISC4_bit;
sbit LCD_D7_Direction at TRISC5_bit;

void main()
{
    TRISA=0xFF; ADCON1 = 0x80; TRISB=0;
    PortB=0;
    lcd_init(); lcd_cmd(_lcd_cursor_off); lcd_cmd(_lcd_clear);
    while(1)
    {
        Nx = ADC_Read(0);
        Ny = ADC_Read(1);
        Px = (240.0/1023.0)*Nx;
        Py = (320.0/1023.0)*Ny;
        .....(Px,Pxtxt); // Conversion en texte de Px
        .....(1,1,"Px="); // Afficher "Px=" à la ligne 1 et colonne 1
        lcd_out(1,4,Pxtxt); // Afficher la valeur de Px à la ligne 1 et colonne 4
        intTostr(.....); // Conversion en texte de Py
        lcd_out(.....,"Py="); // Afficher "Py=" à la ligne 2 et colonne 1
        lcd_out(.....); // Afficher la valeur de Py à la ligne 2 et colonne 4
        if (( Px>50 && Px<80) &&( Py>200 &&Py<230)) icone_Armer =1;
        else icone_Armer =0;
    }
}

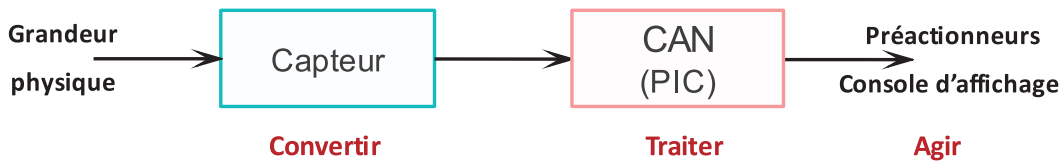
```



1 De quoi s'agit-il ?

- Le Convertisseur Analogique Numérique (C.A.N ou A.D.C) permet de **convertir** une grandeur **analogique** (tension ou image d'une grandeur physique : température, vitesse, ...) en une valeur **numérique** (nombre N) afin d'être traitée et/ou affichée.
- Certains microcontrôleurs intègrent un C.A.N. Ceux de type PIC de la famille Mid-Range (milieu de gamme) possède une résolution sur **10 bits**.

2 Déroulement d'un cycle de conversion :



3 De quelle manière ?

- Pour exploiter la fonction conversion, il faut :
 - choisir un microcontrôleur qui **intègre** un module C.A.N;
 - brancher le signal analogique issu d'un capteur ou d'un potentiomètre de consigne sur une broche multiplexée en modes **analogique** et numérique.
- Dans le programme principal il faut :
 - **Configurer** le registre ADCON1 ;
 - **initialiser** le module C.A.N ;
 - **démarrer** la conversion sur le canal adéquat et lire le résultat.

4 Quel est le registre à configurer ?

Registre de configuration d'un convertisseur : ADCON1

Le registre ADCON1 permet le contrôle de la **configuration** du PORTA et/ou PORTE en mode analogique ou numérique et de **justifier** le résultat de la conversion à gauche ou à droite. Ce choix se fait par programmation pour les PIC16F87X.

Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

=0 : Justification à gauche
=1 : Justification à droite

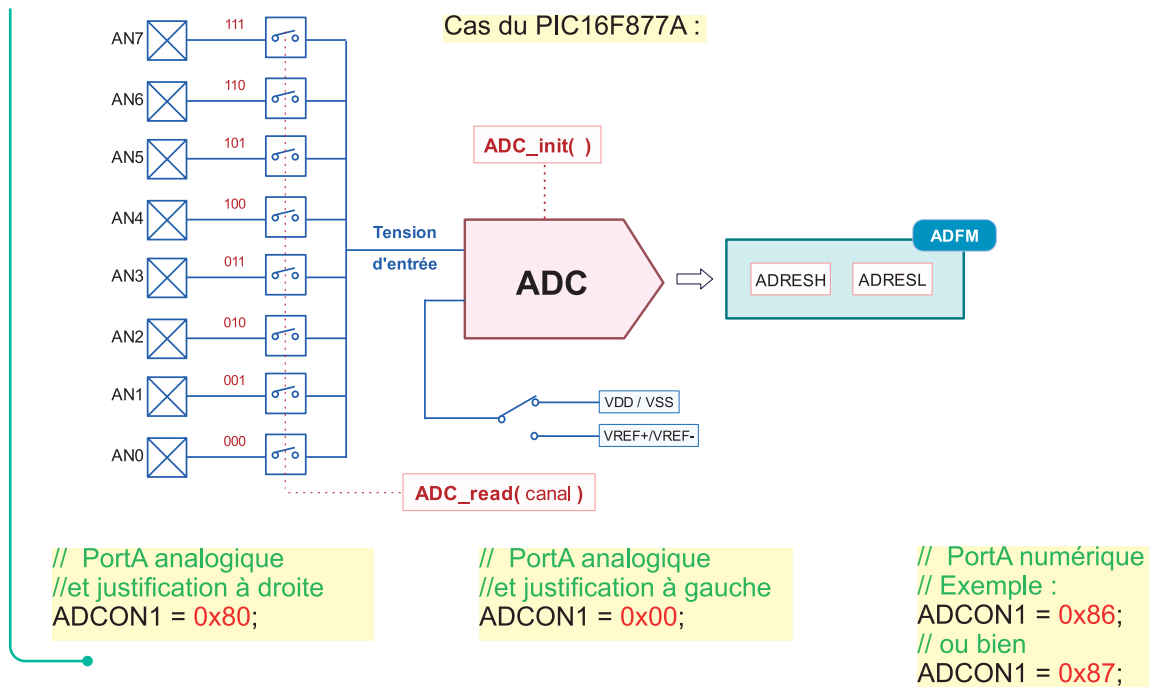
Bits non implantés

=0000 : PortA analogique
=011x : PortA numérique

5 Comment programmer une conversion ?

Pour programmer une conversion analogique numérique, il suffit de (d'):

- **Configurer** le registre **ADCON1** dans la partie configuration du programme principal ;
- **initialiser** le module C.A.N ;
- **démarrer** la conversion et **lire** le résultat sur le canal adéquat.



6 Quelle est la structure d'un programme contenant une conversion A/N ?

```
// Déclaration
unsigned int N ;
void main () { // Fonction principale
// Initialisation
ADC_init();
// Configuration
ADCON1 = 0x..... ;
while (1) { // Boucle infinie
// Démarrer la conversion
N=ADC_read(canal);
..... }
}
```

7 Où cela se produit ?



Station météo connectée

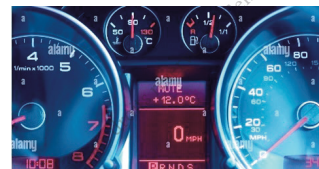


Tableau de bord d'une voiture



Evaluation TH1_seq3

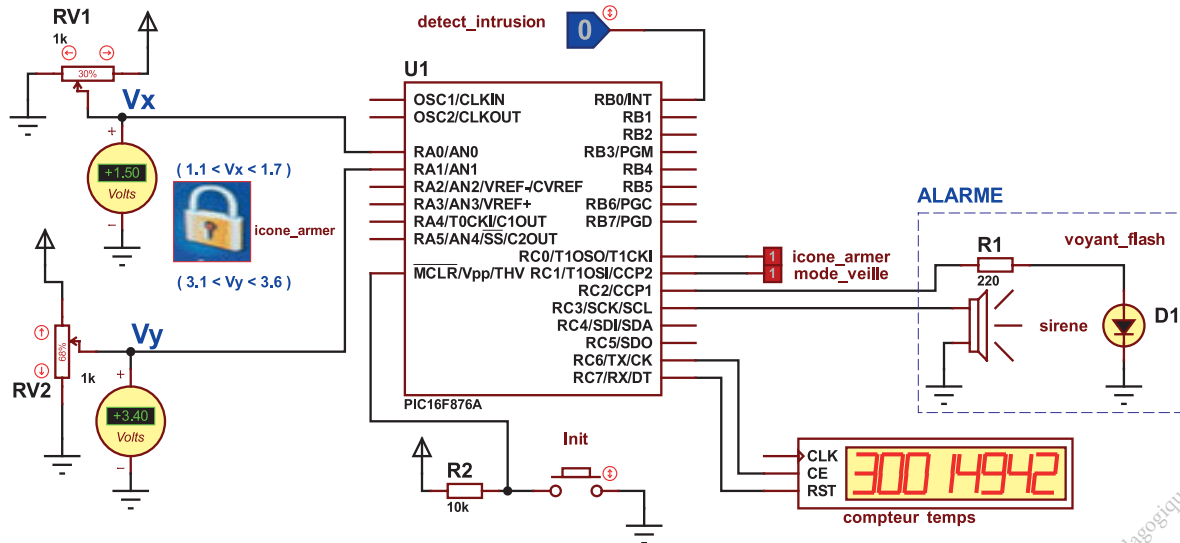


JE REALISE MON PROJET

Remarque :

Les projets proposés sont à titre indicatif. Les élèves peuvent choisir leurs propres projets ou bien l'enseignant peut fournir des projets.

Systeme d'alarme :



Phase d'intégration



Vidéo1 TH1_seq3

© Tous droits réservés au Centre National Pédagogique



Un projet commun encadré (P.C.E) devrait répondre à plusieurs questions, telles que :

1. Quel est le but du projet?
2. Qui sont les membres de l'équipe et quelles sont leurs responsabilités?
3. Quels sont les délais et les échéances à respecter?
4. Quelles sont les ressources disponibles et comment seront-elles allouées?
5. Comment le projet sera-t-il évalué et mesuré?
6. Comment les conflits au sein de l'équipe seront-ils gérés?
7. Comment les décisions seront-elles prises et communiquées?
8. Comment les membres de l'équipe communiqueront-ils entre eux?
9. Comment les obstacles et les imprévus seront-ils gérés?
10. Comment le projet sera-t-il clôturé et quels sont les critères de réussite ?

Répondre à ces questions permettra de clarifier les objectifs, les responsabilités, les délais et les attentes pour tous les membres de l'équipe, et permettra de mettre en place une structure solide pour le projet. Cela contribuera également à minimiser les risques et à maximiser les chances de réussite du projet.



Savoir plus



Fiche de candidature à un projet



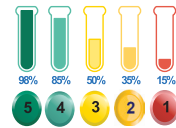
La fiche de la candidature, ci-dessous, est à remplir par chaque membre d'un même groupe d'apprenants. Une copie doit être rendue à l'enseignant.

Exemple	Fiche de candidature
1- Nom & Prénom (responsable) 2- Nom & Prénom (rapporteur) 3- Nom & Prénom (membre)	Noms et Prénoms des membres du groupe : ✍ - ✍ - ✍ - ✍ -
Exemple : projet n°01 Concevoir et réaliser une maquette à base de circuits intégrés permettant de mettre en œuvre le fonctionnement d'un système d'alarme.	Projet :
Étapes : - Recherche documentaire; - choix technologiques ; - conception: dessins, schémas, programmes ; - réalisations.	Présentation des étapes de réalisation : ✍ - ✍ - ✍ - ✍ -
Les travaux devraient être partagés d'une manière équitable ou effectués en collaboration.	Partage des tâches : ✍ - ✍ - ✍ -
Indiquer sur le portfolio l'objet de chaque rencontre, les avancements acquis, les difficultés rencontrées, les solutions proposées, l'entente entre le groupe, etc.	Bilan des actions : Dates et objets des rencontres présentiels : ✍ ✍ Dates des réunions à distance : ✍ ✍ ✍
Le choix de l'outil de communication doit être réfléchi et argumenté.	Outils de communication : Power point, Posters, Vidéo, autre.....

Le [portfolio](#) est manuscrit et individuel. Il contient : la fiche de candidature, les activités, les difficultés rencontrées, les solutions, les recommandations de l'enseignant ainsi que les composantes du projet (dessins, schémas, programmes, etc...). Chaque élève doit tenir une copie de la candidature dans son [portfolio](#).



Grille d'auto-évaluation



Thème 1	Activités				
Critères d'évaluation	Degrés d'appréciation				
	5 Excellent	4 Très bien	3 Passable	2 Insuffisant	1 Faible
Compétences disciplinaires attendues					
CD1.7: Programmer un microcontrôleur.					
J'ai réussi à programmer un microcontrôleur.					
J'ai réussi à faire un choix correct et argumenté d'un microcontrôleur.					
CD 2.7: Mettre en œuvre un système technique et une maquette didactique pilotés par une carte électronique à base de microcontrôleur.					
J'ai réussi à mettre en œuvre un système technique piloté par une carte électronique à base de microcontrôleur.					
J'ai réussi à mettre en œuvre une maquette didactique pilotée par une carte électronique à base de microcontrôleur.					
CD 3.7: Traduire un cahier des charges de fonctionnement d'un système en algorithme et/ou un programme et en rendre compte.					
J'ai réussi à traduire un cahier des charges de fonctionnement d'un système en un algorithme.					
J'ai réussi à traduire un cahier des charges de fonctionnement d'un système en un programme.					
Les compétences de vie et les éducations à...					
J'ai appliqué l'esprit critique pour un choix argumenté du microcontrôleur.					
J'ai communiqué clairement au sein du groupe et avec l'ensemble de ma classe.					
J'ai appris à prendre les décisions d'une manière efficace.					
J'ai exprimé mes idées d'une manière claire, courte et dans un langage adapté à mes différents interlocuteurs.					

© Tous droits réservés au Centre National Pédagogique

Le thème "Logique combinatoire" permettra aux apprenants d'approfondir leurs savoirs et savoir-faire relatifs à la résolution de problèmes de logique combinatoire, d'enrichir, davantage, leurs habiletés relatives à la conception, la réalisation et l'exploitation de systèmes combinatoires.

COMPOSANTES DES COMPÉTENCES DISCIPLINAIRES ATTENDUES:

CD 1.5 Résoudre un problème de logique combinatoire.

CD 2.5 Réaliser un montage à base de circuits combinatoires et analyser le fonctionnement.

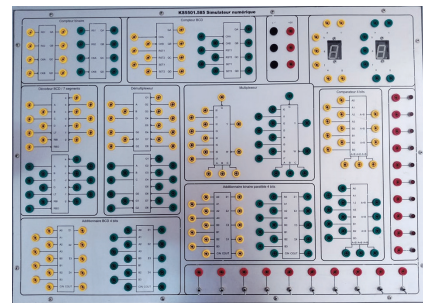
CD 3.5 Lire et décoder le schéma du circuit de commande d'un système combinatoire et en rendre compte.

COMPÉTENCES DE VIE VISÉES:

 Résolution de problèmes.

 Prise de décision.

 Communication.



LOGIQUE COMBINATOIRE:

1 PRÉREQUIS

- Fonction combinatoire.
- Mise en oeuvre d'une fonction combinatoire.
- Système combinatoire.

3 CONDITIONS MATÉRIELLES NÉCESSAIRES

- Logiciels.
- Systèmes techniques.
- Maquettes didactiques.

2 SAVOIRS ET SAVOIR-FAIRE NOUVEUX

- * **Circuits combinatoires:**
 - Additionneur.
 - Comparateur.
 - Multiplexeur/Démultiplexeur.
 - U.A.L.
- * **Applications:**
 - Solution câblée.
 - Solution programmée.

4 CRITÈRES D'ÉVALUATION

- Analyse réussie d'un circuit combinatoire.
- Identification et mise en oeuvre réussies d'un circuit combinatoire intégré.
- Mise en oeuvre réussie d'une solution programmée.
- Participation active.
- Respect total des règles de communication.



CONTENU DU THÈME 2:

Situation déclenchante de chaque séquence : Activité N°1	Présenter l'objet d'apprentissage : J'observe, j'analyse la situation et je fixe un plan d'action et les étapes à suivre.				
	ACTIVITÉS :		CDi.j	SUPPORTS	
<ul style="list-style-type: none"> Apprentissage : <ul style="list-style-type: none"> - Je développe. - J'applique. - Je consolide mes acquis. Évaluation : <ul style="list-style-type: none"> - J'évalue mes acquis. 	SÉQUENCE 1: Circuits combinatoires	Additionneur	Activité N° 2	CD1.5 CD2.5 CD3.5	<ul style="list-style-type: none"> - Ressources numériques. - Logiciels. - Microcontrôleurs. - Maquettes didactiques. - Ordinateurs. - Vidéoprojecteur. - Etc...
			Activité N° 3		
		Comparateur	Activité N° 4		
			Multiplexeur/ Démultiplexeur		
		Activité N° 6			
		Activité N° 7			
		U.A.L	Activité N° 8		
			Activité N° 9		

- Intégration :
 - Je réinvestis.

J'intègre mes acquis et je réalise mon P.C.E
(Projet Commun Encadré)



LIENS DES RESSOURCES NUMÉRIQUES:



Evaluation TH2



Projet n°01



Projet n°02



Projet n°03



Projet n°04



DATASHEETdes C.I

Remarque :

Les circuits combinatoires ne feront pas l'objet d'un cours. Ils seront utilisés selon l'application.

Séquence 1

CIRCUITS COMBINATOIRES

ACTIVITÉ 1 : Situation déclenchante



Vidéo1 TH2_seq1



Aide à l'activité 1

J'observe l'objet d'apprentissage

L'ordinateur de bord de la voiture (figure 1), enregistre et calcule les valeurs telles que la distance, la consommation du carburant et la vitesse moyenne durant la conduite. Il indique la distance approximative qui peut être parcourue avec ce qu'il reste du carburant dans le réservoir. On donne ci-dessous (figure 2) le schéma synoptique simplifié de la gestion d'autonomie de distance.



Figure 1

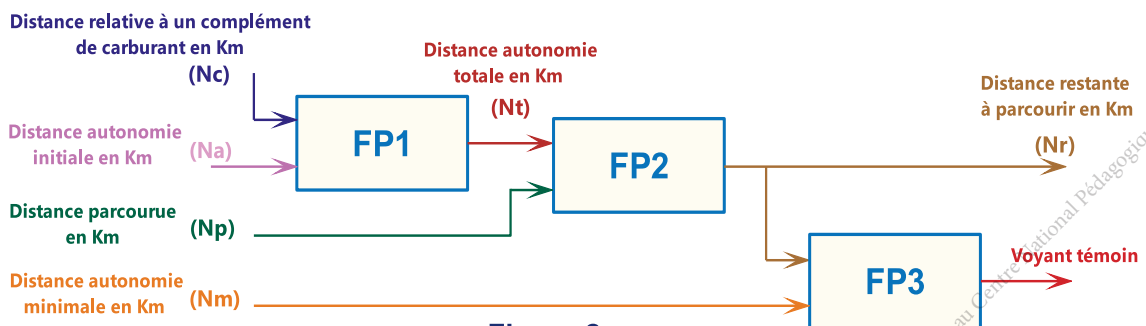


Figure 2

Nr (Distance restant à parcourir) : c'est le nombre de kilomètres pouvant être parcourus avec le reste du carburant détecté dans le réservoir.

Na (Distance autonomie initiale) : c'est le nombre de kilomètres déjà parcourus.

Np (Distance parcourue) : c'est la distance parcourue par la voiture à partir de la dernière remise à zéro de l'ordinateur de bord.

Nt (Distance autonomie totale) : c'est la distance totale qui pourrait être parcourue par la voiture.

Nc (Distance relative à un complément de carburant) : c'est la distance correspondant à un complément de carburant.

Nm (Distance autonomie minimale) : dès que l'autonomie est inférieure à une certaine valeur minimale ($N_m=30$ km par exemple), un voyant témoin s'allume.



La figure 3 représente un exemple simplifié de la carte électronique assurant la gestion d'autonomie de distance.

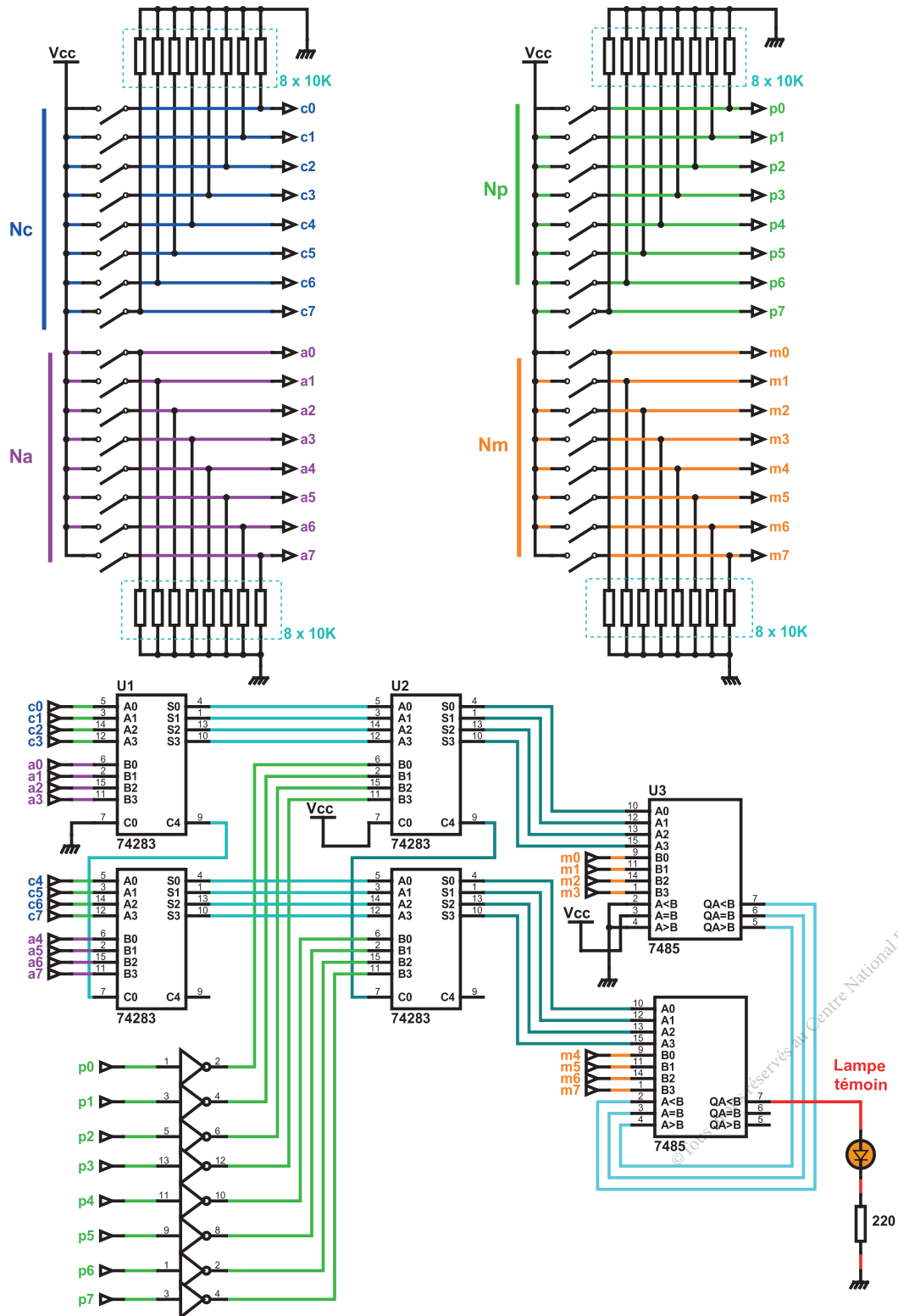


Figure 3



Problématique : Quelles sont les structures électroniques qui conduisent à la gestion d'autonomie de distance pour éviter les pannes sèches?



Idée n° 1 : Ecrire devant chaque fonction le terme qui convient :

Multiplexeur ; soustracteur ; comparateur ; transcodeur ; additionneur.

- FP1 : Cette fonction nécessite un circuit
- FP2 : Cette fonction nécessite un circuit
- FP3 : Cette fonction nécessite un circuit



Idée n° 2 : Compléter la phrase ci-dessous par les mots suivants :
combinatoire ; cahier des charges ; logique combinatoire ;
fonction combinatoire ; microcontrôleur.

- ✂ Pour résoudre un problème de, il faut utiliser un circuit intégré afin de répondre aux exigences déclarées dans un ou programmer un en vue d'illustrer une.....



J'analyse la situation



En petits groupes :

1

Compléter les questions suivantes par les termes:

logique combinatoire ; circuits ; opérations arithmétiques.

- ✂ Quelles sont les réalisées par les fonctions FP1, FP2 et FP3 ?
- ✂ Quels sont les qui peuvent réaliser ces opérations?
- ✂ Quelle est la différence entre et circuit combinatoire?

2

Compléter les hypothèses (H1 et H2) suivantes par les termes :
circuit logique ; logique combinatoire ; circuit combinatoire ; décomposition.
Ajouter d'autres hypothèses si elles vous paraîtront nécessaires.

H1	Le choix d'un est une étape nécessaire pour l'analyse d'un
H2	Le fonctionnement d'un système de est décrit par une en plusieurs blocs fonctionnels.
H3

3

Le plan d'action, ci-dessous, aide à la recherche des réponses argumentées aux questions posées et à la vérification des hypothèses.

- Réalisation des activités, recherches d'informations, cours,....;
- vérification des hypothèses ;
- formulation des nouveaux savoirs et évaluations.

I- GESTION DU NIVEAU DU CARBURANT DANS LE RESERVOIR

ACTIVITÉ 2 : Etude de la fonction FP1



Aide à l'activité 2

1. Mise en œuvre du circuit intégré 74283

La fonction **FP1** est assurée par le circuit intégré **74283**.
On donne ci-dessous le schéma de câblage correspondant.



En petits groupes :

1.1. A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage ci-dessous (figure 4) et vérifier le fonctionnement du circuit intégré **74283**.

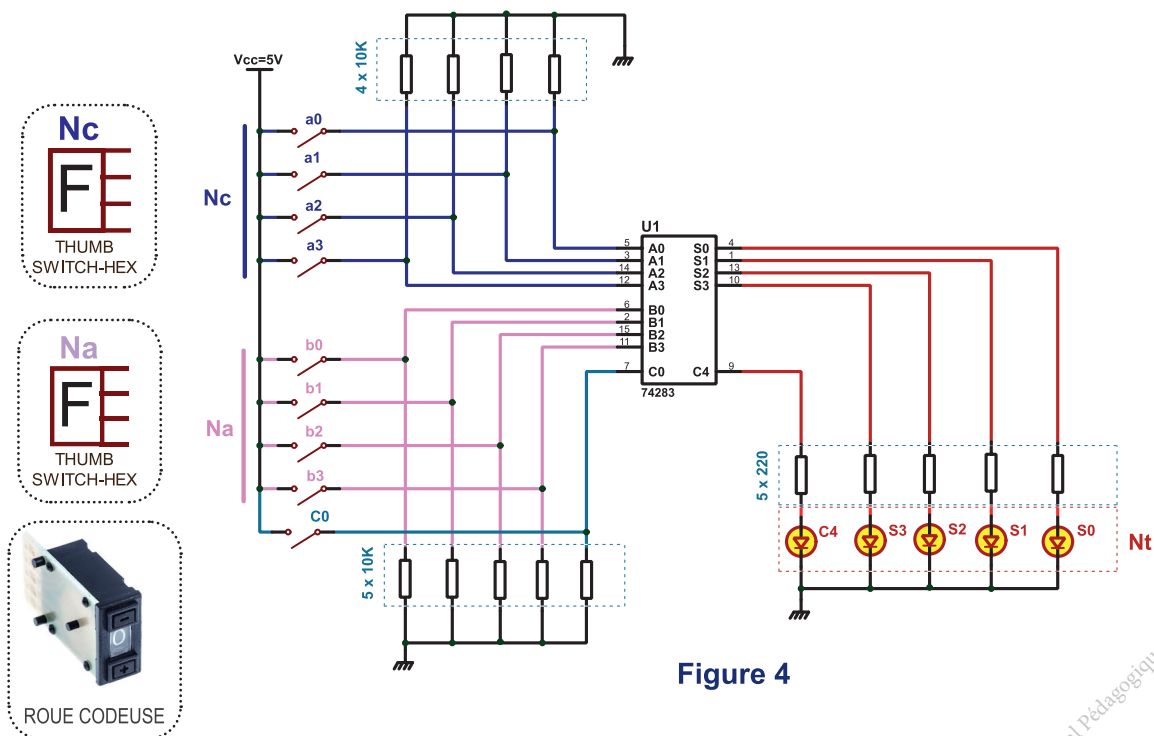


Figure 4

1.2. Compléter le tableau ci-dessous :

	Opérande A (Nc)				Opérande B (Na)				Opérande S (Nt)				
C ₀	a ₃	a ₂	a ₁	a ₀	b ₃	b ₂	b ₁	b ₀	C ₄	S ₃	S ₂	S ₁	S ₀
0	1	0	0	1	0	1	1	0
1	1	0	1	1	0	1	1	1

1.3. Déduire le rôle du circuit intégré **74283**.

1.4. Identifier les différentes broches du circuit intégré **74283** en complétant le tableau ci-après.

Désignation	N° de broche	Fonction
A(0...3)	Bits de l'opérande A
B(0...3)	6, 2, 15, 11
C ₀	Retenue à l'entrée
C ₄
S(0...3)	Bits du résultat de l'addition en binaire

1.5. Le circuit intégré **74283** fait partie de la famille TTL. Proposer une référence de circuit intégré de la famille C-MOS assurant la même fonction que le circuit **74283** puis préciser sa tension d'alimentation.



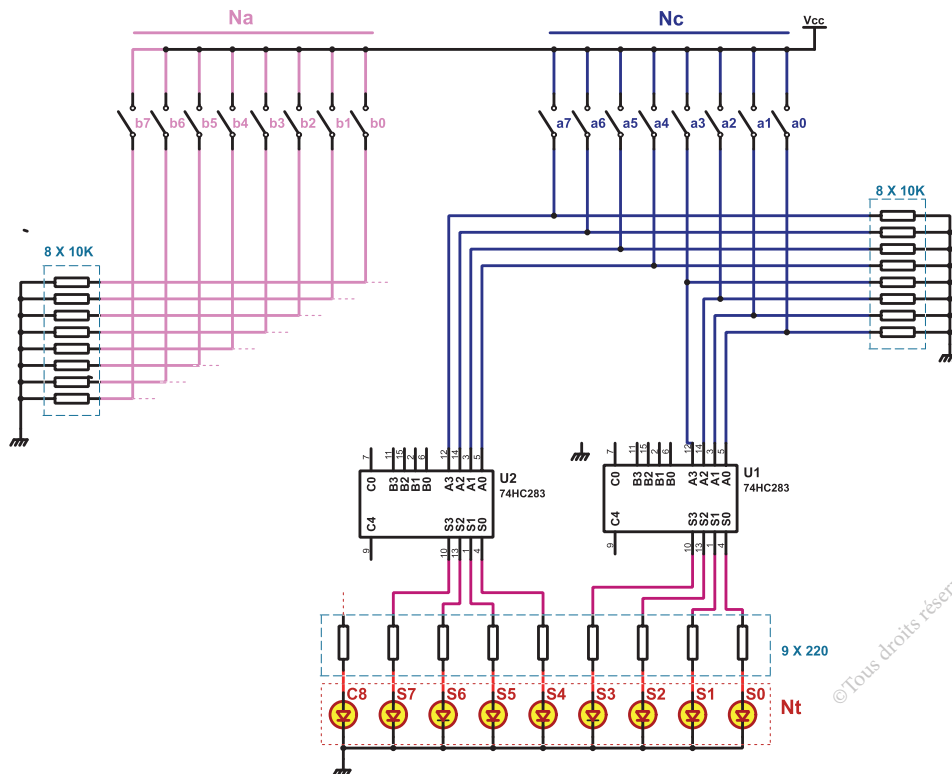
2. Mise en cascade des circuits intégrés 74283

On désire étendre les nombres **Nc** et **Na** à 8 bits lors de la réalisation de la fonction **FP1** avec le circuit intégré **74283**.

2-1- Donner le nombre de circuits intégrés **74283** nécessaires pour effectuer cette opération.



2-2- Compléter sur la figure 5 le schéma de câblage correspondant.



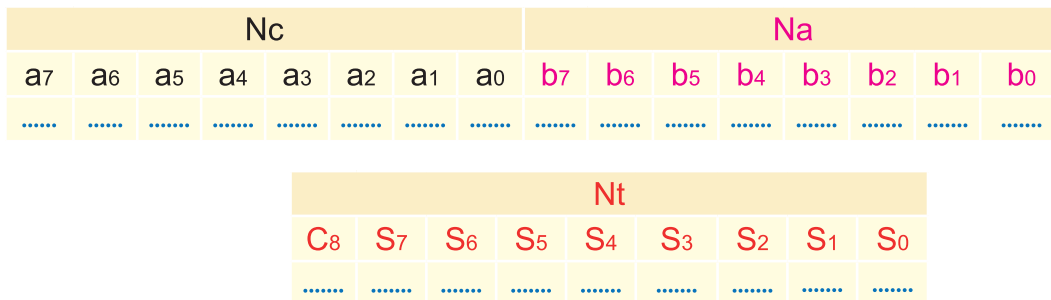
© Tous droits réservés au Centre National Pédagogique

Figure 5



En petits groupes :

2.3. A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage de la figure 5 et vérifier son fonctionnement en complétant les tableaux ci-après pour $Nc = 41_{(10)}$ et $Na = 66_{(10)}$.



3. Mise en œuvre d'une solution à base de microcontrôleur

Une solution programmée peut être envisagée pour déterminer la **distance autonomie totale Nt** à partir des valeurs de **Nc** (distance relative à un complément de carburant) et **Na** (**distance autonomie initiale**).

On propose d'utiliser le microcontrôleur **16F877A** et le langage MikroC comme langage de programmation.

On suppose que les nombres **Nc** et **Na** sont des mots binaires à 8 bits chacun et que **Nt** est un mot binaire à 9 bits (le neuvième bit étant la retenue sortante du résultat de l'addition des nombres **Nc** et **Na**). On donne à la figure 6 le schéma de simulation relatif à cette application.

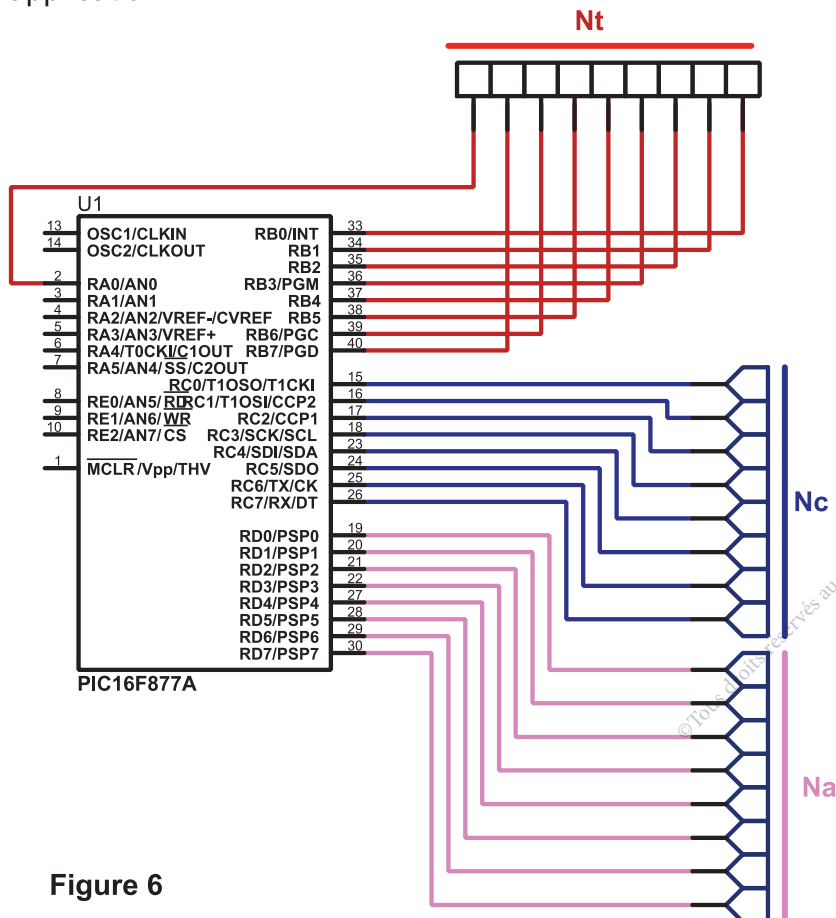


Figure 6

3.1. L'affichage du résultat de l'addition des nombres **Nc** et **Na** est en binaire. Compléter, en MikroC, le programme ci-après.

```

int Nt; // Déclaration de la variable Nt
char Na .....; // Déclaration de la variable Na
..... at PORTC; // Déclaration de la variable Nc
void main() { // Début du programme principal

TRISC=0x.....; TRISD=0x.....; // Configuration des registres
TRISA.B0=.....; TRISB=.....;

PORTB=.....; .....=0; // Initialisation des portB et portA
while (1) { // Début de la boucle tant que
Nt= .....; // Calcul du nombre Nt

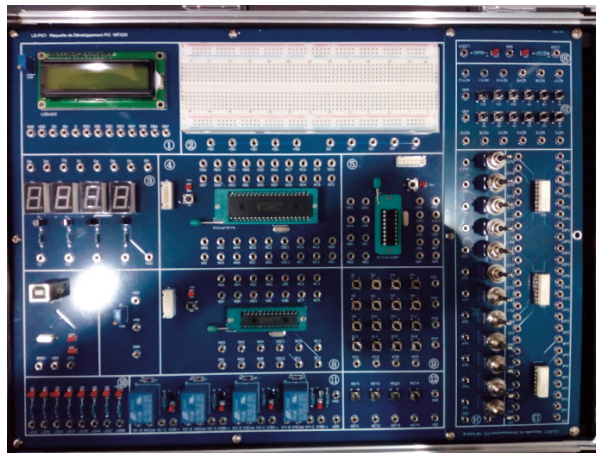
PORTB=.....; .....=Nt >>8 ; /* Affichage de la valeur de Nt en binaire
sur 9 bits */
} // Fin de la boucle tant que
} // Fin du programme principal

```



En petits groupes :

- 3.2.** Saisir le programme précédent à l'aide du compilateur MikroC puis simuler son fonctionnement à l'aide d'un logiciel de simulation.
- 3.3.** Sur maquette d'expérimentation, vérifier le fonctionnement du système.



4. Mise en œuvre de la fonction FP1 par le circuit intégré 4560

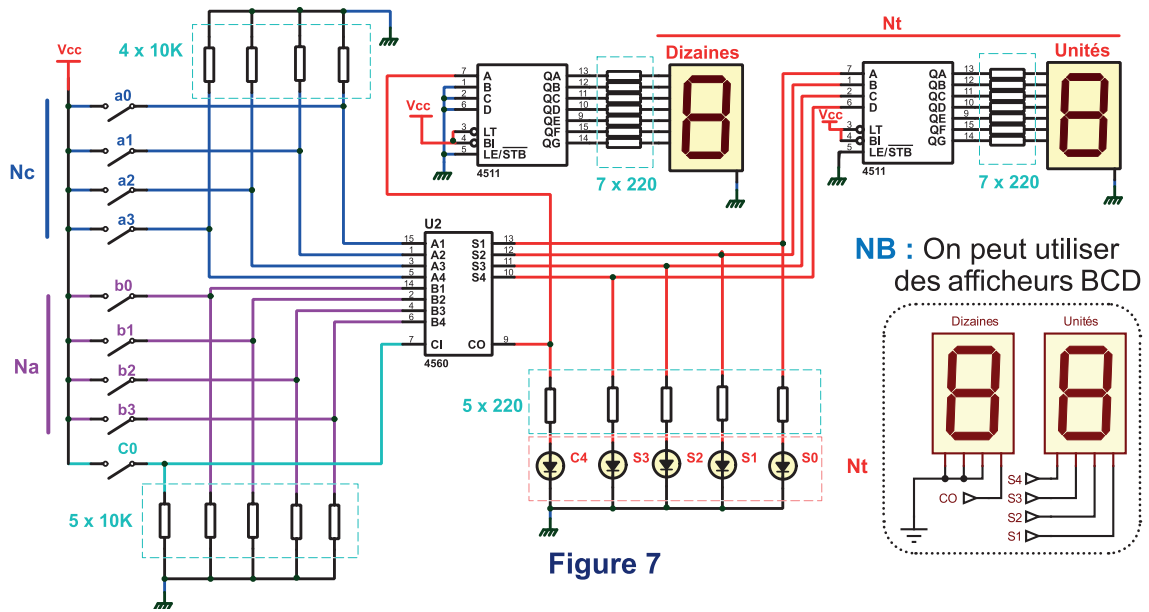


La fonction FP1 est assurée maintenant par le circuit intégré **4560**.
On donne à la figure 7 le schéma de câblage correspondant.



En petits groupes :

- 4.1.** A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage de la figure 7 et vérifier le fonctionnement du circuit intégré **4560**.



4.2. Compléter le tableau ci-dessous :

	Opérande A (Nc)				Opérande B (Na)				Opérande S (Nt) (sortie de U2 : 4560)					Valeur affichée (Nt)	
C0	a3	a2	a1	a0	b3	b2	b1	b0	C4	S3	S2	S1	S0	Dizaines	Unités
0	1	0	0	1	0	1	1	0
1	0	0	1	1	0	1	1	1

4.3. Déduire le rôle du circuit intégré 4560.



.....

.....

4.4. Identifier les différentes broches du circuit intégré 4560 en complétant le tableau suivant :

Désignation	N° de broche	Fonction
A(1...4)	15, 1, 3, 5	Bits de l'opérande A en BCD
B(1...4)
CI	Retenue à l'entrée
CO
S(1...4)	Bits du résultat de l'addition en BCD

4.5. Compléter le schéma de câblage de la figure 8 afin d'additionner en BCD deux nombres Na et Nc à deux digits chacun.

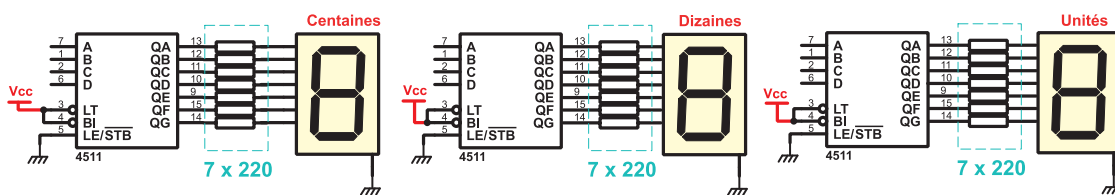
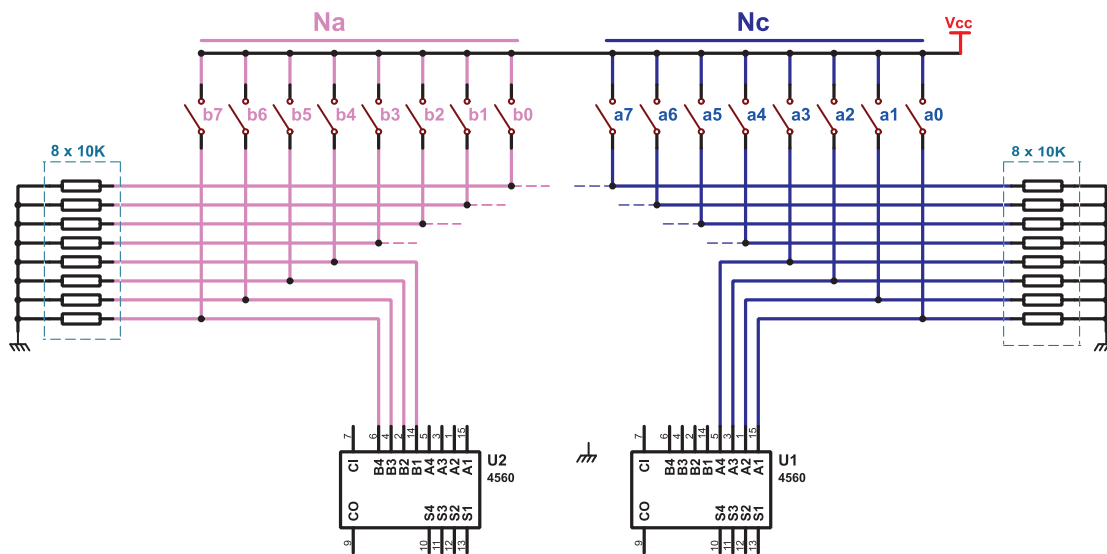


Figure 8

5. Par pair, récapituler les nouveaux savoirs



.....

.....

.....

.....

.....

.....

.....

ACTIVITÉ 3 : Etude de la fonction FP2

1. Mise en œuvre avec le circuit intégré 74283

La distance restante à parcourir Nr s'exprime par la relation suivante:

$$Nr = Nt - Np$$

1.1. Donner l'expression de Nr en utilisant la notation en complément à 2.



1.2. Compléter le schéma de câblage donné à la figure 9 pour obtenir la distance restante à parcourir Nr (utiliser un format à 8 bits y compris le bit de signe).



Aide à l'activité 3

© Tous droits réservés au Centre National Pédagogique

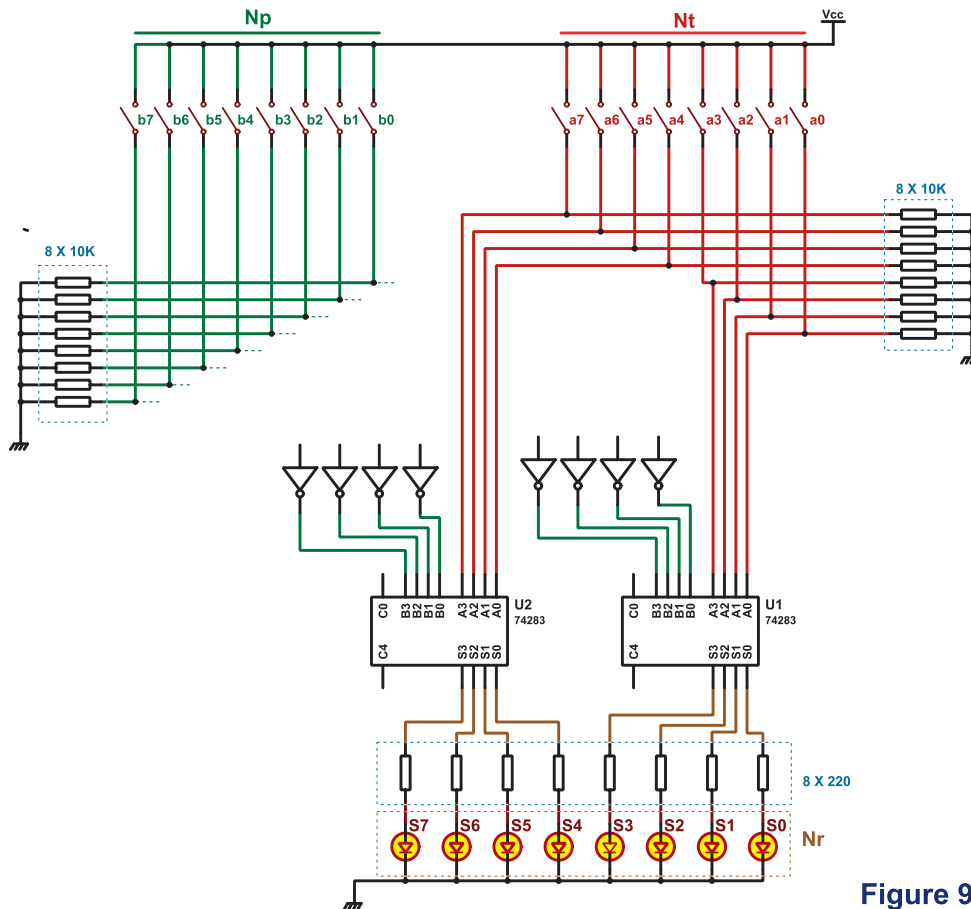


Figure 9



En petits groupes :

1.3. A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage de la figure 9 et compléter le tableau ci-dessous.

Nt								Np							
a7	a6	a5	a4	a3	a2	a1	a0	b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	0	0	0	0	0	1	0	0	1	1	1	1

Nr							
S7	S6	S5	S4	S3	S2	S1	S0
.....

2. Mise en œuvre d'une solution à base de microcontrôleur



Une solution programmée peut être envisagée pour déterminer la distance autonomie restante **Nr** à partir des valeurs de **Nt** (distance autonomie totale) et **Np** (distance parcourue).

On propose d'utiliser le microcontrôleur **16F877A** et le langage **MikroC** comme langage de programmation.

On suppose que les nombres **Nt**, **Np** et **Nr** sont des mots binaires à 8 bits chacun. On donne à la figure 10 le schéma de simulation relatif à cette application.

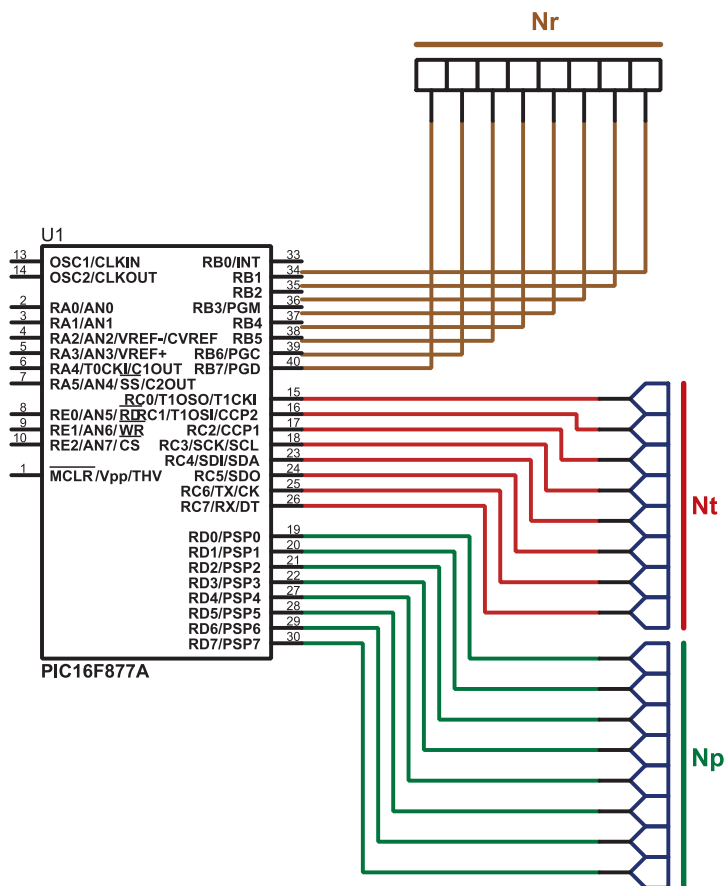


Figure 10



En petits groupes :

2.1. L'affichage du résultat de la soustraction des nombres **Nt** et **Np** est en binaire. Compléter, en MikroC, le programme suivant :

```

char ..... ; // Déclaration de la variable Nr
char ..... ; // Déclaration de la variable Nt
..... Np at PORTD ; // Déclaration de la variable Np

void main() {

TRISB=..... ; TRISC=0x..... ; // Configuration des registres
TRISD=0x..... ;

..... ; // Initialisation du portB à 0

while (1) { // Boucle infinie
..... ; // Calcul du nombre Nr
..... ; // Affichage en binaire de la valeur de Nr
} // Fin boucle infinie
} // Fin du programme

```

- 2.2. Saisir le programme précédent à l'aide du compilateur MikroC puis simuler son fonctionnement à l'aide d'un logiciel de simulation.
- 2.3. Sur maquette d'expérimentation, vérifier le fonctionnement du système.

3. En petits groupes, synthétisons les nouveaux savoirs



.....

.....

.....

.....

.....

ACTIVITÉ 4 : Etude de la fonction FP3

1. Mise en œuvre du circuit intégré 7485



La fonction FP3 est assurée par le circuit intégré 7485.

On donne à la figure 11 le schéma de câblage correspondant.



Aide à l'activité 4

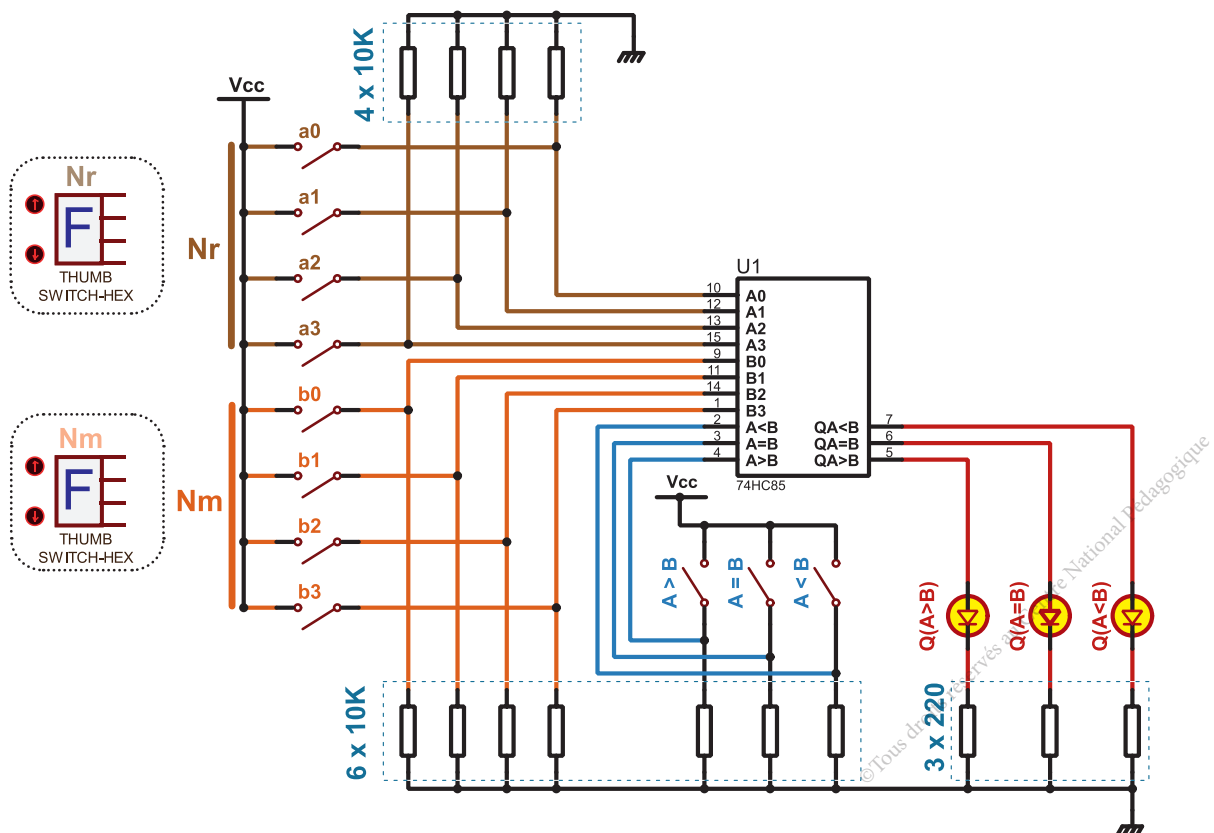


Figure 11



En petits groupes :

- 1.1. A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage donné à la figure 11 et vérifier le fonctionnement du circuit 7485.
- 1.2. Compléter le tableau ci-après par l'état logique des sorties $Q_{A>B}$, $Q_{A=B}$ et $Q_{A<B}$.

Opérande A (Nr)				Opérande B (Nm)				Entrées de mise en cascade			Sorties du comparateur		
a ₃	a ₂	a ₁	a ₀	b ₃	b ₂	b ₁	b ₀	A>B	A=B	A<B	Q _(A>B)	Q _(A=B)	Q _(A<B)
1	0	1	1	1	0	0	1	1	0	1
1	0	1	1	1	0	1	1	1	0	1
0	0	1	1	1	0	1	1	1	0	0
1	0	1	1	1	0	1	1	0	0	0
1	0	1	1	1	0	1	1	1	1	0
1	0	1	1	1	0	1	1	0	1	1

1.3. Déduire le rôle du circuit intégré **7485**.



1.4. Comparer les résultats trouvés précédemment avec ceux figurant sur la fiche technique du circuit intégré **7485** (voir "Aide à l'activité 4").



1.5. Compléter les phrases suivantes par les termes:

entrées de mise en cascade ; des sorties ; logique haut.



D'après les résultats obtenus et la fiche technique du circuit intégré **7485**, si (A₃=B₃) et (A₂=B₂) et (A₁=B₁) et (A₀=B₀), l'état dépend de l'état logique des



La sortie Q_{A=B} (broche 6) ne prend 1 que si l'entrée de mise en cascade A=B (broche 3) est au niveau

1.6. Identifier les différentes broches du circuit intégré **7485** en complétant le tableau suivant :

Désignation	N° de broches	Fonction
A(0...3)	10, 12, 13,15	Bits de l'opérande A
B(0...3)
A<B, A=B, A>B
Q _{A<B} , Q _{A>B} , Q _{A=B}

1.7. Le circuit intégré **7485** fait partie de la famille TTL. Proposer une référence de circuit intégré de la famille C-MOS assurant la même fonction que le circuit **7485**.



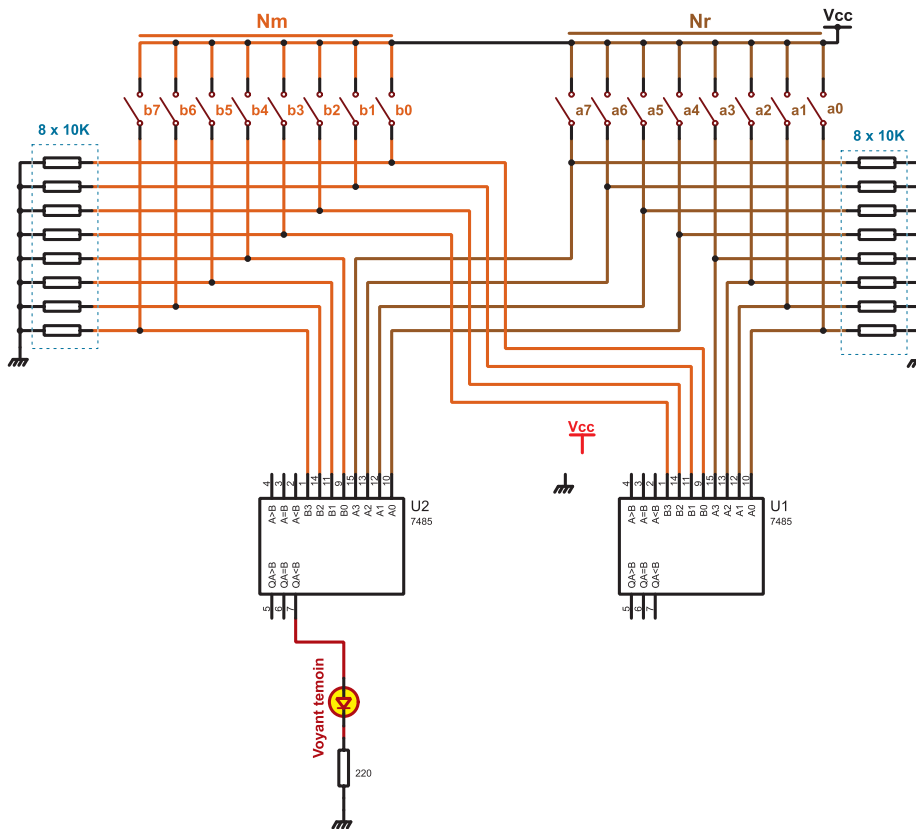
2. Mise en cascade des circuits intégrés **7485**

On désire étendre les nombres **Nr** et **Nm** à 8 bits lors de la réalisation de la fonction **FP3** avec le circuit intégré **7485**.

2.1. Donner le nombre de circuits intégrés **7485** nécessaires pour effectuer cette opération.



2.2. Compléter le schéma de câblage de la figure 12 correspondant.



En petits groupes :

Figure 12

2.3. A l'aide d'un logiciel de simulation, d'un simulateur ou d'une maquette, câbler le montage de la figure 12 et vérifier son fonctionnement en complétant le tableau suivant pour $Nr = 71_{(10)}$ et $Nm = 30_{(10)}$ puis pour $Nr = 25_{(10)}$ et $Nm = 30_{(10)}$.

Nr								Nm								Sortie
a7	a6	a5	a4	a3	a2	a1	a0	b7	b6	b5	b4	b3	b2	b1	b0	Voyant témoin
...
...

3. Mise en œuvre d'une solution à base de microcontrôleur

On se propose de compléter le programme ci-après assurant la comparaison de deux mots binaires Nr et Nm à 8 bits chacun et d'afficher le résultat sur trois diodes LEDs : H_1 , H_2 et H_3 .

- ◆ Si $Nr > Nm$ alors $H_1=1$, $H_2=0$ et $H_3=0$.
- ◆ Si $Nr = Nm$ alors $H_1=0$, $H_2=1$ et $H_3=0$.
- ◆ Si $Nr < Nm$ alors $H_1=0$, $H_2=0$ et $H_3=1$.

Une entrée de validation **EV** est reliée à la broche RE0 permet d'autoriser la comparaison.

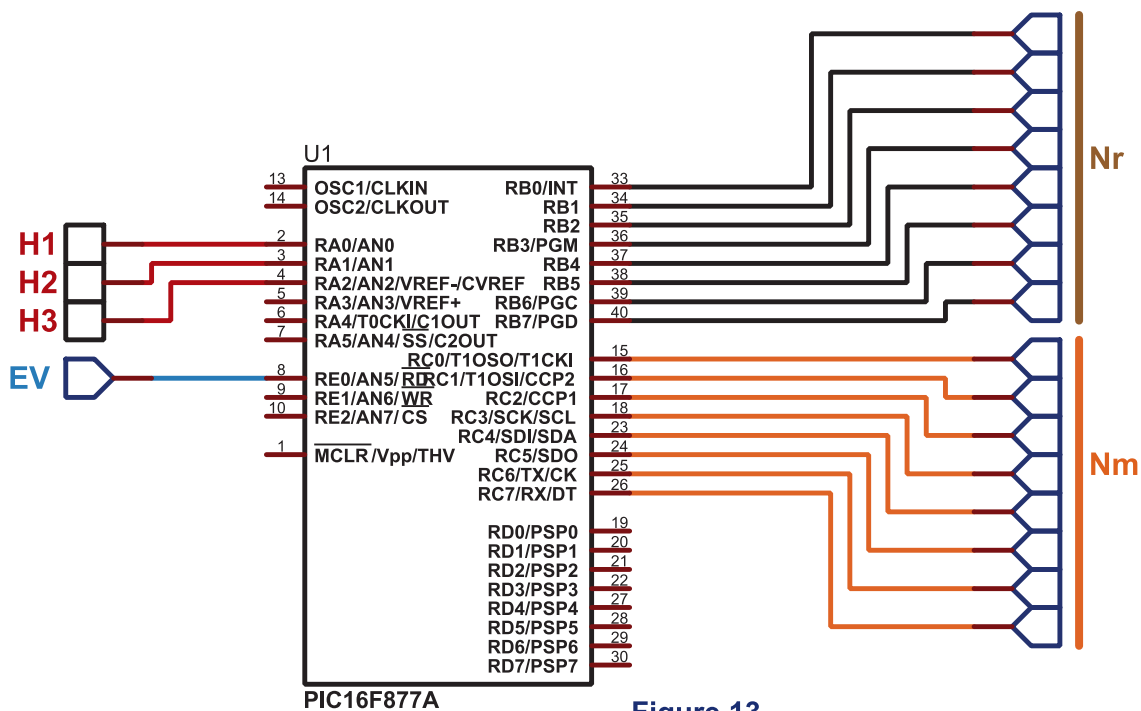


Figure 13



En petits groupes :

3.1. En se référant à la description du fonctionnement du comparateur et à la figure 13, compléter le programme ci-dessous par les instructions convenables afin d'assurer la comparaison des deux nombres **Nr** et **Nm** et de commander les LEDs H1, H2 et H3.

```

char Nr at ..... ;
char Nm ..... ;
..... at PORTA.B0 ; // Déclaration des variables
sbit H2 at PORTA.B1 ; // Nr, Nm, H1, H3 et EV
..... at PORTA.B2 ;
..... at PORTE.B0 ;

void main()
{
    ..... ; // Début du programme
    ..... ; // Configuration des portB et portC en entrées
    ..... ; ADCON1=7 ; // Configuration du portE en entrées numériques
    ..... ; // Configuration du portA en sorties
    ..... ; // Initialisation du portA à 0
while (1) // Boucle infinie
{ // Début de la boucle infinie
while(EV==1) { // Tanque EV=1
if (.....) // Commande de H2
{H1=0; H2=1; H3=0;}
}
}
}

```

```

if (.....) // Commande de H1
{H1=1; H2=0; H3=0;}
if (.....) // Commande de H3
{H1=0; H2=0; H3=1;}
}}} // Fin du programme

```

3.2. Saisir le programme précédent à l'aide du compilateur MikroC puis simuler son fonctionnement à l'aide d'un logiciel de simulation.

3.3. Sur maquette d'expérimentation, vérifier le fonctionnement du système.

II- GESTION D'AUTRES FONCTIONS ELECTRONIQUES DANS UNE VOITURE

ACTIVITÉ 5 : Etude de la fonction multiplexage

J'observe l'objet d'apprentissage

De nos jours, les fonctions électroniques sont très présentes dans les véhicules. L'ajout de fonctions nécessite plus de câblages. L'augmentation du nombre de fonctions électroniques (options) dans l'automobile engendre l'augmentation du nombre de fils.

Le schéma classique de câblage par une communication entre boîtiers électroniques peut embarquer plus de 4000 mètres de fils d'où la nécessité d'adopter une solution pour réduire le nombre de fils et minimiser le poids.



Vidéo2 TH2_seq1



Problématique : Quelle solution électronique utilisée pour répondre à ce besoin?



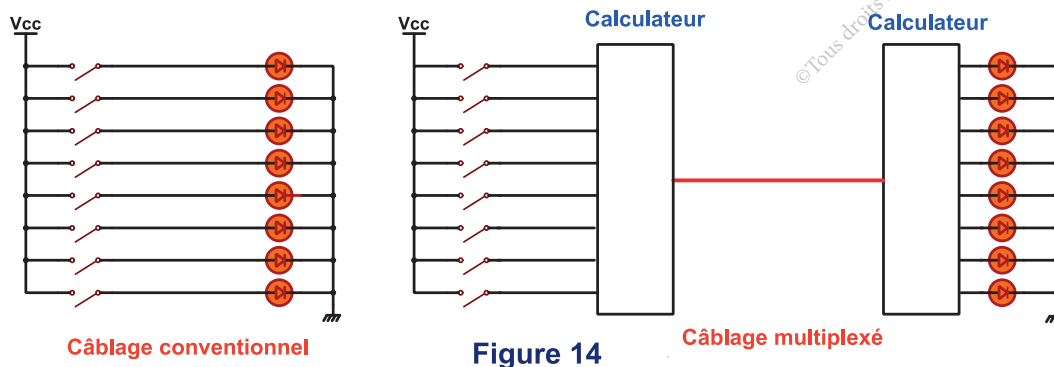
En petits groupes :

Essayons de répondre aux questions suivantes :

- Comment simplifier l'architecture électrique des véhicules?
- Comment augmenter le nombre de fonctions et réduire le nombre de câbles?
- Par quel moyen peut-on regrouper plusieurs fonctions électroniques (options) dans un seul boîtier?

1. Etude comparative entre le câblage classique et le câblage multiplexé

La figure 14 représente le schéma synoptique de deux solutions proposées :



1.1. Etude du câblage conventionnel

La figure 15 représente un exemple de schéma de câblage classique de quelques fonctions électroniques dans une voiture.

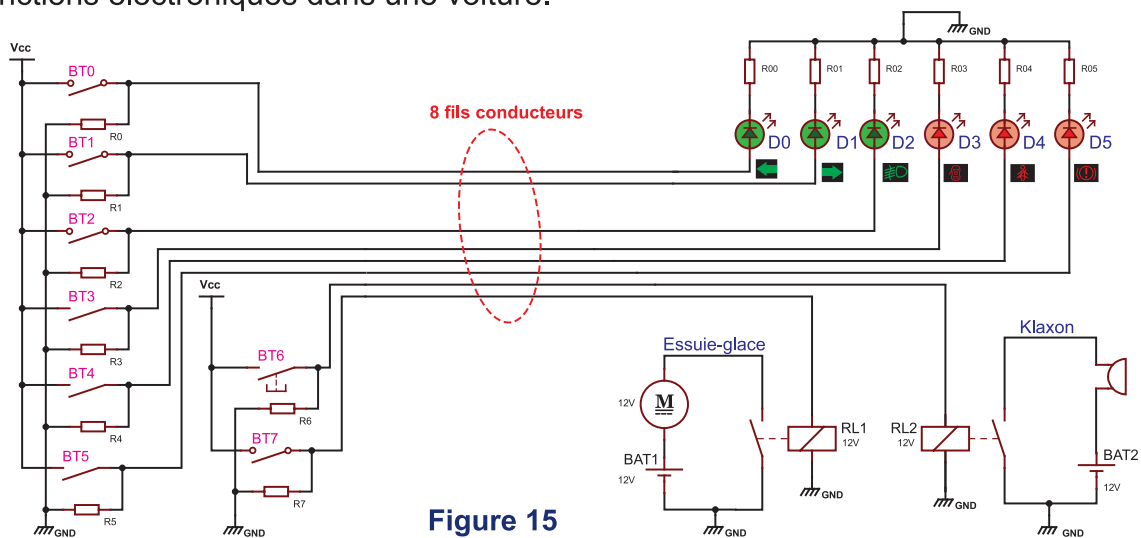


Figure 15

Déterminer :

- a- Le nombre des entrées :
- b- Le nombre des sorties :
- c- Le nombre de fils conducteurs reliant les entrées aux sorties :

1.2. Etude du câblage multiplexé

La figure 16 représente un exemple de schéma de câblage multiplexé assurant les fonctions électroniques obtenues par la figure 15.

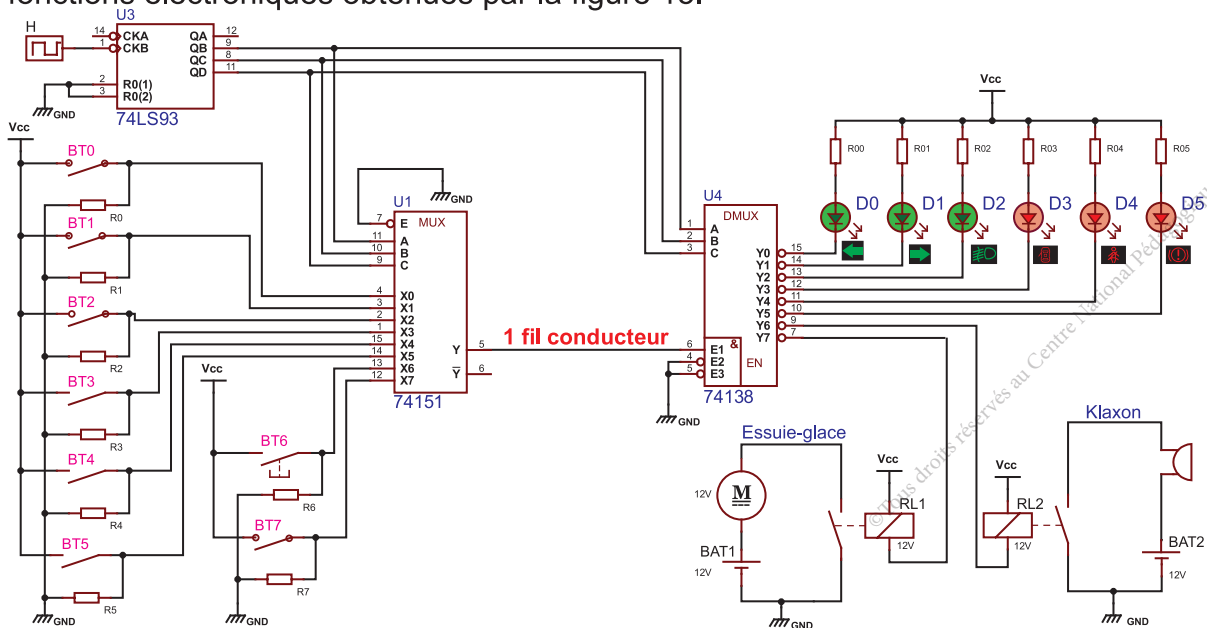


Figure 16

a- Déterminer :

- Le nombre des entrées :
- Le nombre des sorties :
- Le nombre de fils conducteurs reliant les entrées aux sorties :

b- Prélever la référence des deux circuits U₁ et U₄ et préciser le nombre des entrées et des sorties de chacun d'eux.

☞ Référence de U₁ :

☞ Nombre d'entrées de données: ; Nombre de sortie(s) :

☞ Référence de U₄ :

☞ Nombre d'entrée : 1 (E1) ; Nombre de sorties :

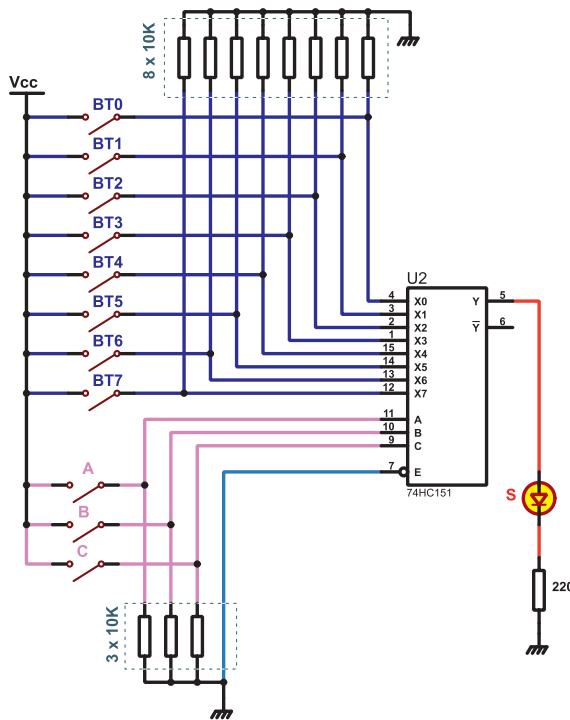
c- Préciser l'avantage du deuxième montage (figure 16) par rapport au premier (figure 15). ☞

ACTIVITÉ 6 : Mise en œuvre du circuit intégré 74151

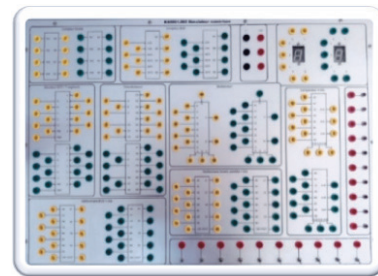


En petits groupes :

1. Câbler le montage ci-dessous (figure 17) sur un simulateur numérique ou le saisir à l'aide d'un logiciel de simulation puis observer son fonctionnement.



Aide à l'activité 6



Simulateur logique

Figure 17

2. Compléter la table de fonctionnement ci-dessous.

CBA	BT ₇	BT ₆	BT ₅	BT ₄	BT ₃	BT ₂	BT ₁	BT ₀	S
000	0	0	0	0	0	0	0	1	...
001	1	1	1	1	1	1	0	1	...
010	1	1	1	1	1	0	1	1	...
011	1	1	1	1	0	1	1	1	...
100	0	0	0	1	0	0	0	0	...
101	0	0	1	0	0	0	0	0	...
110	1	0	1	1	1	1	1	1	...
111	0	1	1	1	1	1	1	1	...

3. Mettre l'entrée \bar{E} au niveau logique 1, manipuler A, B et C et déduire le rôle de cette entrée.

.....

4. Compléter les phrases ci-dessous par l'un des termes suivants afin d'identifier le rôle des broches du circuit intégré **74151**.

entrées de données ; entrées de sélection ; sortie du circuit 74151.

✎ A, B et C :

✎ $X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7$:

✎ Y :

5. Conclure sur le rôle du circuit Intégré **74151** en complétant le paragraphe ci-dessous par les termes suivants :

la sortie Y ; multiplexeur ; sélection ; commutateur ; combinatoire ;
 2^3 entrées de données ; X_0 ; X_1 ; sortie.

✎ Le circuit intégré **74151** est un circuit utilisé pour aiguiller
 8 = vers une seule

✎ Suivant l'état des 3 entrées de (CBA), recopie
 l'une des entrées de données X_0 ou jusqu'à X_7 . Le circuit intégré
 74151 est appelé **8 vers 1** (Mux 8 vers 1).

Exemples :

Si CBA = 000 alors Y = ; Si CBA = alors Y = X_2 .

✎ Le multiplexeur joue le rôle d'un à plusieurs positions
 orientant ainsi une information d'entrée parmi plusieurs vers la sortie.

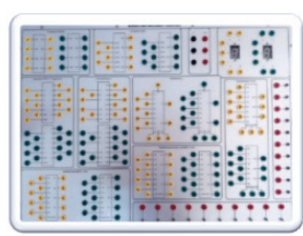
6. Compléter l'équation logique de la sortie Y du multiplexeur 8 vers 1 en fonction des entrées de données et de sélection.

✎ $Y = \bar{C}.B.\bar{A}.X_0 + \dots + \dots + \dots + \dots + \dots + \dots + C.B.A.X_7$

ACTIVITÉ 7 : Mise en œuvre du circuit intégré 74138

En petits groupes :

1. A l'aide d'une maquette, d'un simulateur ou d'un logiciel de simulation, câbler le logigramme de la figure 18.



Aide à l'activité 7

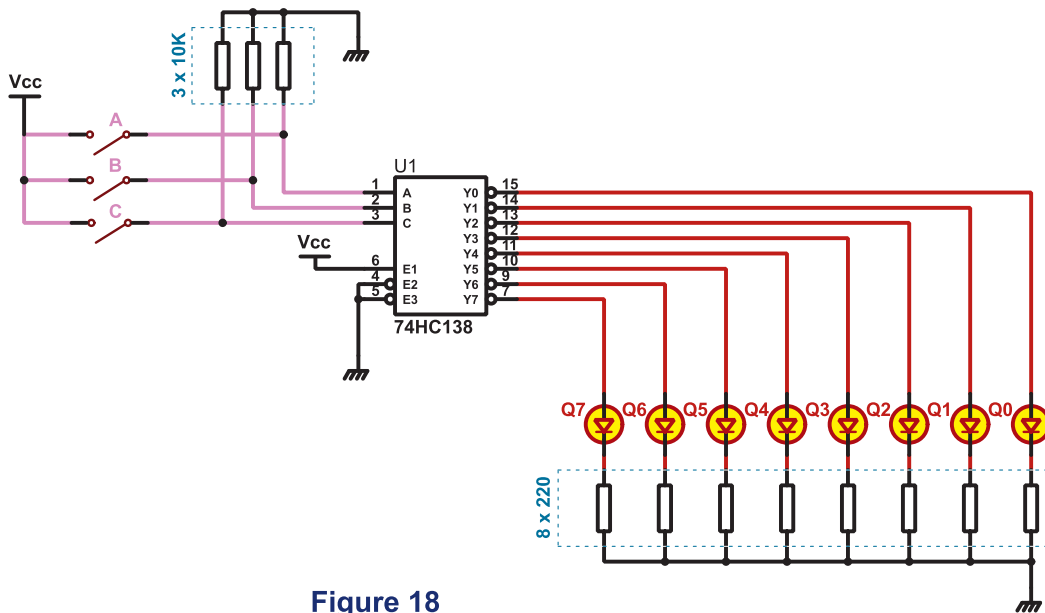


Figure 18

2. Vérifier le fonctionnement du montage précédent en complétant le tableau ci-dessous. (La sortie active prend un niveau logique 0).

CBA	000	001	010	011	100	101	110	111
Sortie active	Q ₀

3. Mettre l'entrée \bar{E}_2 au niveau logique 1, manipuler A, B et C puis conclure.

.....

4. Mettre l'entrée \bar{E}_3 au niveau logique 1, manipuler A, B et C puis conclure.

.....

5. Compléter la phrase suivante:

Les deux entrées \bar{E}_2 et \bar{E}_3 permettent de le circuit.

Ce sont des entrées de actives au niveau

6. Compléter les phrases ci-dessous par l'un des termes suivants afin d'identifier le rôle des broches du circuit intégré **74138**.

entrée de donnée ; entrées de sélection ; sorties du circuit 74138 ;

entrées de validation.

A, B, C :

Y₀, Y₁, Y₂, Y₃, Y₄, Y₅, Y₆, Y₇ :

\bar{E}_2 , \bar{E}_3 :

E₁:

7. Conclure sur le rôle du circuit Intégré **74138** en complétant le paragraphe ci-dessous par les termes suivants :

une sortie ; l'entrée de donnée ; démultiplexeur ; 3 entrées de sélection ; combinatoire ; commutateur ; entrées de validation ; multiplexage.

- ✎ Le circuit intégré **74138** est un circuit..... utilisé pour aiguiller E_1 vers l'une des huit sorties $Y_0, Y_1, Y_2, \dots, Y_7$.
- ✎ Selon l'état logique des (CBA), parmi les huit est active.

Exemples :

- ◆ Si CBA = 000 alors $Y_0 = 0$ (Seule la sortie Y_0 est active) ;
- ◆ Si CBA = 010 alors $Y_2 = 0$ (Seule la sortie Y_2 est active).
- ✎ Les deux entrées \bar{E}_2 et \bar{E}_3 permettent de valider le circuit. Ce sont des
- ✎ Le circuit intégré **74138** est appelé **1 vers 8** (DMUX 1 vers 8). Il joue le rôle d'un à plusieurs positions orientant ainsi l'information d'entrée vers une sortie choisie.
- ✎ Le démultiplexage est la fonction inverse du

III- MISE EN ŒUVRE D'UNE UNITÉ ARITHMÉTIQUE ET LOGIQUE (U.A.L)

ACTIVITÉ 8 : Mise en œuvre du circuit intégré 74381



Le calculateur (figure 19) assurant le pilotage et le contrôle des fonctions électroniques dans une voiture est généralement construit autour d'une Unité Arithmétique et Logique (U.A.L). Cette unité permet d'effectuer des opérations de calcul arithmétiques (addition, soustraction, ...) et des opérations logiques (ET, OU, XOR, ...).



Figure 19 : Calculateur



Aide à l'activité 8



En petits groupes :

1. A l'aide d'une maquette, d'un simulateur ou d'un logiciel de simulation, câbler le logigramme de la figure 20.

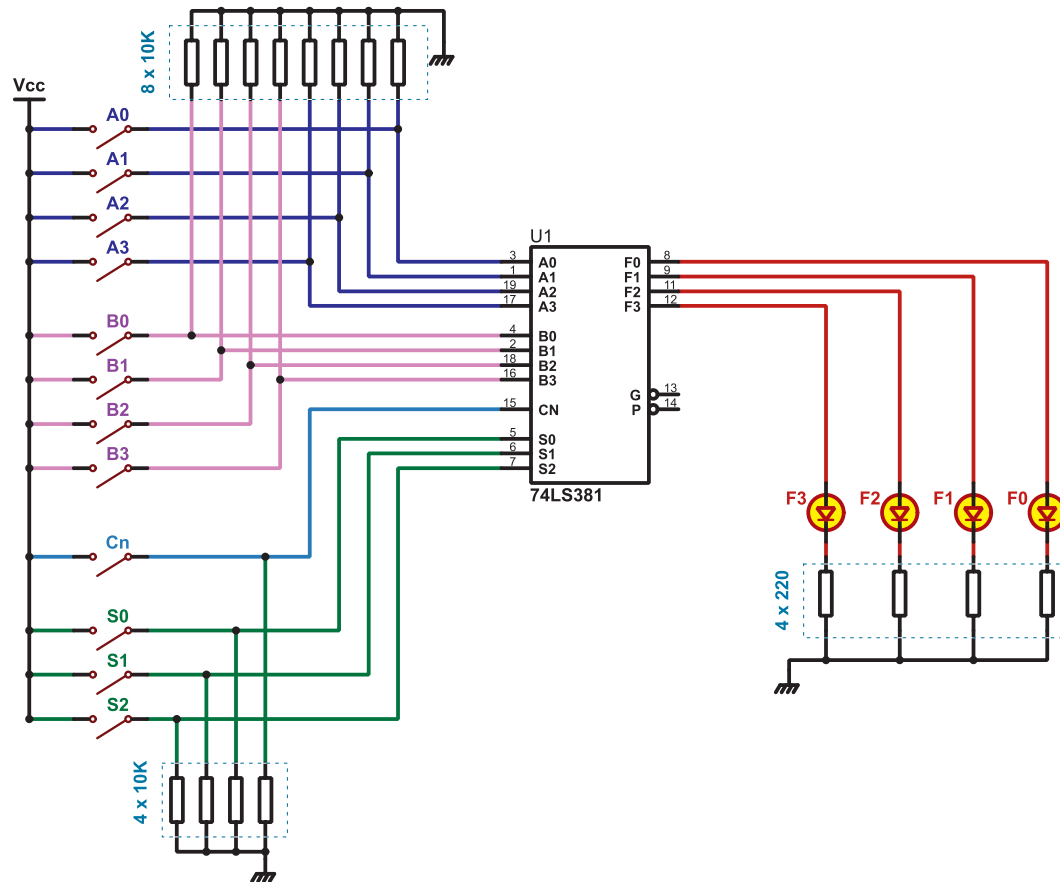


Figure 20

2. Compléter le tableau ci-dessous en indiquant l'état des sorties **F₃F₂F₁F₀**.

Entrées de sélection C _n S ₂ S ₁ S ₀	Opérande A A ₃ A ₂ A ₁ A ₀	Opérande B B ₃ B ₂ B ₁ B ₀	Sorties F ₃ F ₂ F ₁ F ₀	Nom de la fonction réalisée	Type de la fonction réalisée
0 0 1 1	0 1 1 0	0 1 1 0
1 0 1 0	1 0 0 0	0 1 1 1
0 1 1 0	0 1 1 0	0 1 1 1
1 1 1 0	0 1 1 0	0 1 1 1
1 0 0 1	0 1 1 0	1 1 1 0

3. En se référant au document constructeur du circuit intégré **74381** :

- Compléter le tableau ci-dessus, en précisant le nom et le type de la fonction réalisée.
- Vérifier les résultats obtenus dans le tableau précédent.
- Identifier le rôle des différentes broches du circuit intégré **74381** en complétant le tableau ci-après.

N° de la broche	Désignation	Fonction
.....	Bits de l'opérande B
.....	A ₃ A ₂ A ₁ A ₀
.....	Retenue à l'entrée
5, 6, 7
.....	F ₃ F ₂ F ₁ F ₀

ACTIVITÉ 9 : Mise en œuvre d'une UAL à base de microcontrôleur

Le microcontrôleur de type PIC **16F877A** est utilisé pour réaliser des opérations arithmétiques et logiques. Soient A et B deux mots binaires à 4 bits et F le résultat de l'opération effectuée.

$A = (A_3A_2A_1A_0)_2$ et $B = (B_3B_2B_1B_0)_2$ et $F = (C_4F_3F_2F_1F_0)_2$ où C_4 étant la retenue sortante.

Les entrées de sélection des opérations à réaliser sont affectées aux bits **RE0**, **RE1**, **RE2** du PORTE.

On donne à la figure 21, le schéma de simulation relatif à cette application. Le tableau ci-dessous résume le fonctionnement de l'UAL à concevoir.

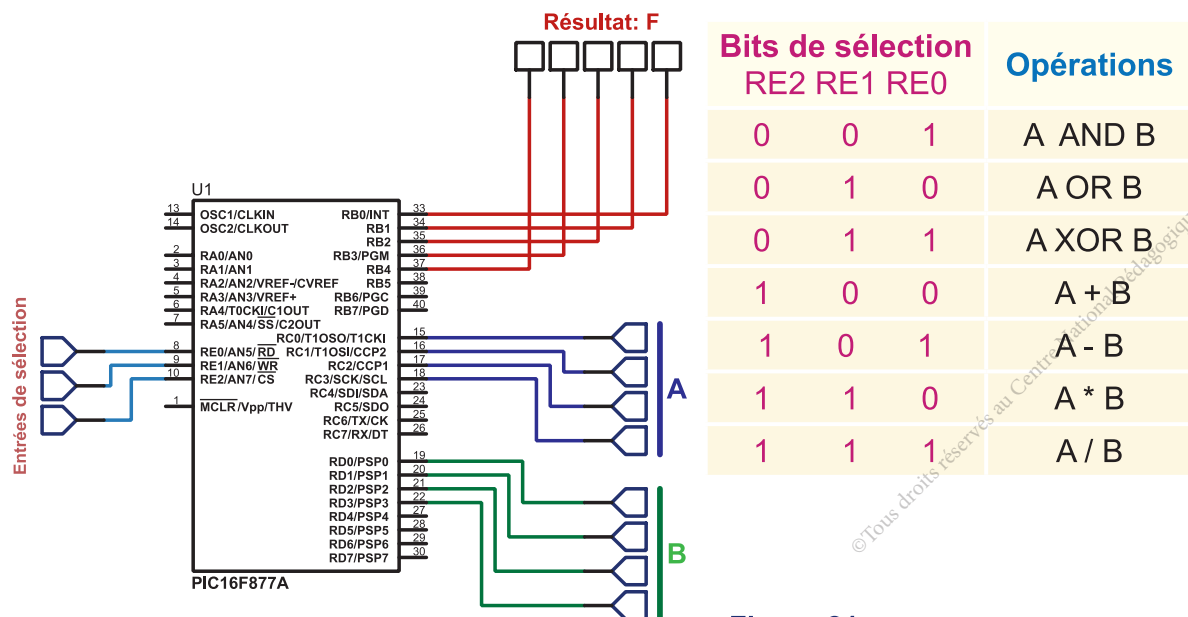


Figure 21

En petits groupes :

- En se référant au tableau qui résume le fonctionnement de l'U.A.L et au schéma de câblage (figure 21), compléter le programme en MikroC permettant de réaliser les différentes opérations arithmétiques et logiques.

N.B : Utiliser des fonctions spécifiques (routines) pour chaque opération.

```

char A,B,E,Fc;

void ET_logique ()

{ Fc= ..... ; }

void OU_logique ()

{ Fc= ..... ; }

void OU_exclusive ()

{Fc= ..... ; }

void addition ()

{Fc= ..... ; }

void soustraction ()

{Fc= ..... ; }

void multiplication ()

{Fc= ..... ; }

void division ()

{Fc= ..... ; }

void main()

{

  ADCON1=0x17;

```

```

// Suite du programme
TRISB=0B11100000; TRISC=255;
TRISD=255;TRISE=7;

PORTB=0;

while (1)

{

  A= (PORTC & 0X0F);
  B=(PORTD & 0X0F);
  E= PORTE;

  PORTB=Fc;

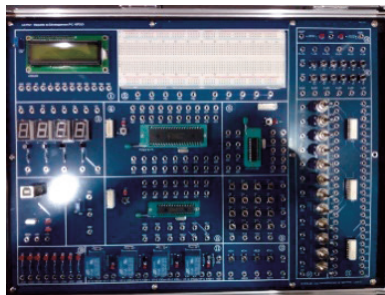
  if (E==.....) ET_logique();
  if (E==.....) OU_logique();
  if (E==.....) OU_exclusive ();
  if (E==4) .....
  if (E==5) .....
  if (E==6) .....
  if (E==7) .....

}

}

```

2. Saisir le programme précédent à l'aide du compilateur MikroC puis simuler son fonctionnement à l'aide d'un logiciel de simulation.
3. Sur maquette d'expérimentation, vérifier le fonctionnement de l'U.A.L.

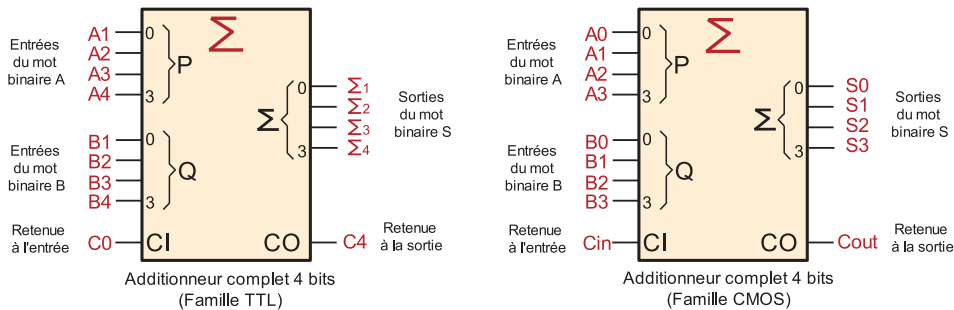


© Tous droits réservés au Centre National Pédagogique

1 Addition binaire

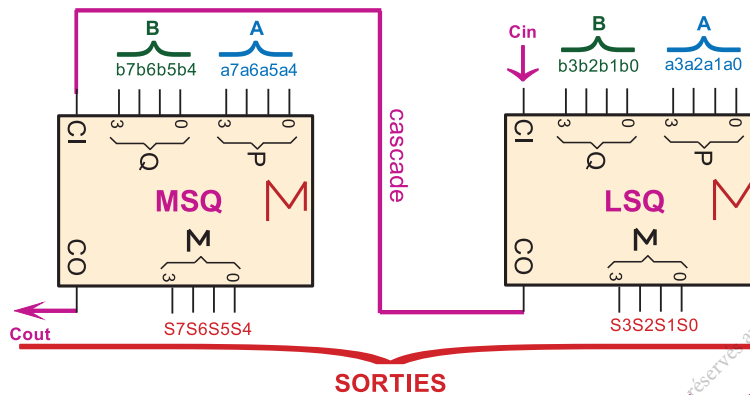
- Un additionneur complet 4 bits (4-bit binary full adder) permet d'additionner **2 nombres de 4 bits** ($A=A_4A_3A_2A_1$, $B=B_4B_3B_2B_1$) et éventuellement la retenue à l'entrée (CI).
- $A_4, A_3, A_2, A_1, B_4, B_3, B_2, B_1$ et CI constituent les entrées de l'additionneur et CO, S4, S3, S2, S1 représentent les bits du résultat de l'addition en binaire (CO : retenue à la sortie représentant le bit de poids le plus fort et S1 : bit de poids le plus faible du résultat).

Symboles



Mise en cascade de deux additionneurs à 4 bits

- Pour additionner deux nombres de **plus de 4 bits**, il faut associer **en cascade** plusieurs additionneurs de 4 bits. La mise en cascade consiste à **relier** la retenue de sortie du circuit (LSQ) avec la retenue d'entrée du circuit (MSQ).



MSQ : Quartet du poids le plus fort.
LSQ : Quartet du poids le plus faible.

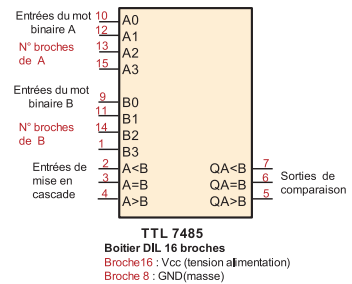
2 Comparaison binaire

- Il s'agit de **comparer deux** nombres binaires A et B pour indiquer en sortie si $A > B$; $A < B$ ou $A = B$.
- Le circuit intégré 7485 est un **comparateur** 4 bits, c'est-à-dire qu'il effectue la **comparaison** de deux nombres de 4 bits chacun. De plus, il dispose de **3 entrées** notées (=), (>) et (<) qui autorisent la mise en **cascade** de plusieurs circuits comparateurs de même type.

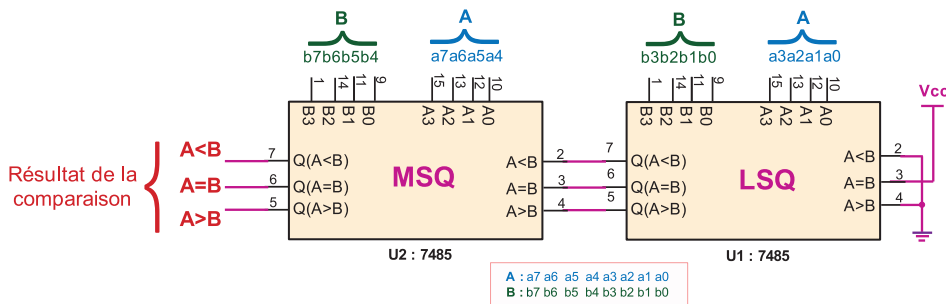


Symbole

Pour le circuit intégré 7485, il faut que l'entrée ($A=B$) du circuit LSQ soit reliée à V_{cc} .



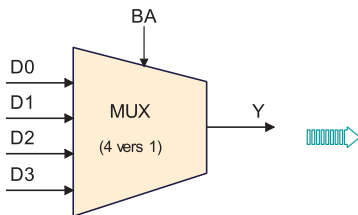
Mise en cascade de deux comparateurs à 4 bits



3 Multiplexage

- Le multiplexage consiste à envoyer sur **une même ligne** de transmission des informations provenant de **sources différentes**.
- Le multiplexeur désigné généralement par (**MUX**) est un circuit logique ayant **plusieurs entrées** de données (N) et **une seule sortie** communiquant ces données.
- L'aiguillage de l'entrée de données désirée vers **la seule sortie** est conditionné par la combinaison de (n) entrées de **sélection** (appelées aussi entrées **d'adresse** ou **d'aiguillage**). Avec $N = 2^n$.

Exemple : Mux 4 vers 1



- Si $BA = 00$ alors $Y = D0$
- Si $BA = 01$ alors $Y = D1$
- Si $BA = 10$ alors $Y = D2$
- Si $BA = 11$ alors $Y = D3$

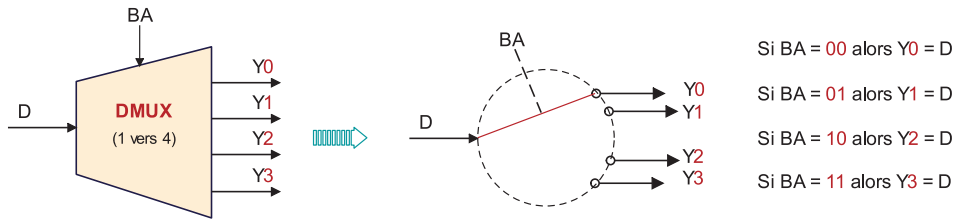
4 Démultiplexage

- Le **démultiplexage** consiste à **aiguiller** une information série provenant d'une **seule et unique entrée** vers **des sorties différentes**.
- Le **démultiplexeur (DMUX)** est un circuit logique ayant **une entrée** de données et **plusieurs sorties** (N).
- L'aiguillage de la donnée vers l'une des (N) sorties est conditionné par la combinaison de (n) entrées de **sélection** (appelées aussi entrées **d'adresse** ou **d'aiguillage**). Avec $N = 2^n$.

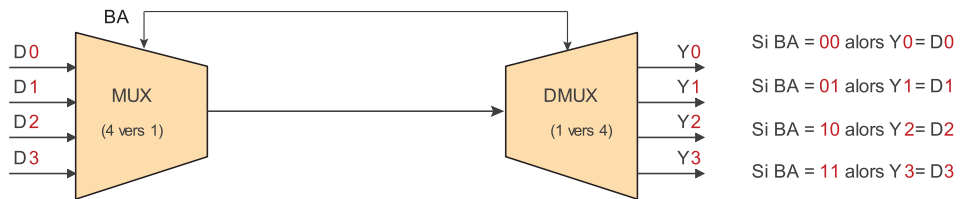
© Tous droits réservés au Centre National Pédagogique



Exemple : Dmux 1 vers 4

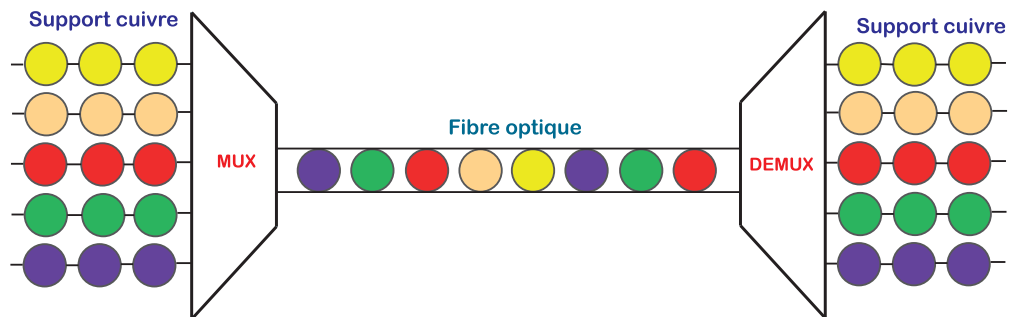


5 Multiplexage & Démultiplexage



Où cela se produit ?

*Transmission sur fibre optique :
 Découpage en temps entre les différentes connexions.*



Multiplexage dans les automobiles :

En exploitant le multiplexage, les constructeurs ont pu diminuer la quantité de câbles présents dans une voiture.



Multiplexage dans les réseaux de télécommunication :

L'interconnexion IP utilise le multiplexage et le démultiplexage dans les réseaux télécoms.



© Tous droits réservés au Centre National Pédagogique



6 Unité Arithmétique et Logique

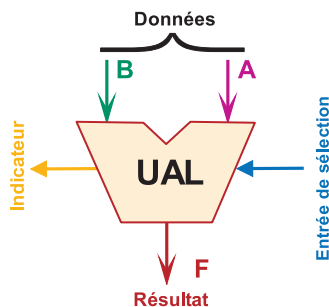
- L'Unité Arithmétique et Logique (**U.A.L** ou **A.L.U**: Arithmetic and Logic Unit) est un circuit **combinatoire** capable d'effectuer une grande variété d'opérations **logiques** et **arithmétiques**.
- L'U.A.L existe sous forme de circuit intégré indépendant, ou **intégrée** dans d'autres circuits numériques spécialisés tels que les microprocesseurs ou les microcontrôleurs.

L'unité arithmétique et logique, dispose de :

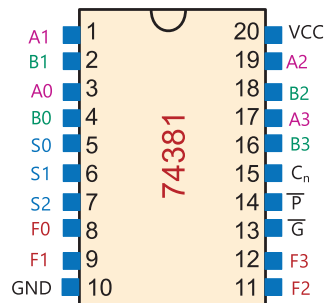
- **Deux entrées** de données A et B sur lesquelles on présente les données à traiter, «les opérands».
- **Une sortie** F donnant le résultat de l'opération effectuée.
- **Des entrées de sélection** permettant le choix de l'opération à effectuer.
- **Un indicateur** donnant l'état du résultat après exécution de l'opération. (Drapeau qui indique s'il ya eu une erreur : division par zéro, résultat négatif, dépassement de capacité...)

Symbole & brochage

Symbole:



Brochage:



Evaluation TH2

© Tous droits réservés au Centre National Pédagogique



JE REALISE MON PROJET

On propose 4 exemples de projets à réaliser et à mettre en œuvre.

Remarque :

Ces projets sont à titre indicatif. Les élèves peuvent choisir leur propre projet ou bien l'enseignant peut fournir d'autres projets.

Projet n°1 :

Gérer l'autonomie de distance d'une voiture et assurer les fonctions FP1, FP2 et FP3. Envisager une solution câblée (voir le schéma synoptique donné à la figure 2 et le schéma de câblage donné à la figure 3).

Projet n°2 :

Gérer l'autonomie de distance d'une voiture et assurer les fonctions FP1, FP2 et FP3. Envisager une solution programmée (voir le schéma synoptique donné à la figure 2 et le schéma de câblage donné à la figure 22).

Projet n°3 :

Réaliser et mettre en œuvre une carte électronique à base de microcontrôleur simulant les fonctions d'une UAL élémentaire (se limiter, au maximum, à quatre opérations).

Projet n°4 :

Réaliser et mettre en œuvre une carte électronique à base de microcontrôleur décrivant la fonction comparaison binaire.

Cahier de charges du projet N°02 :

La carte électronique donnée à la figure 22 est à base du microcontrôleur de type PIC 16F877A qui assure :

- l'addition des deux nombres (NC) et (Na) pour établir le nombre (Nt),
- la soustraction des nombres (Nt) et (Np) pour retrouver le nombre (Nr),
- la comparaison des nombres (Nr) et (Nm) afin d'actionner le voyant témoin si $(Nr) < (Nm)$,
- l'affichage du nombre (Nr) sur trois afficheurs à 7 segments.

Carte électronique :

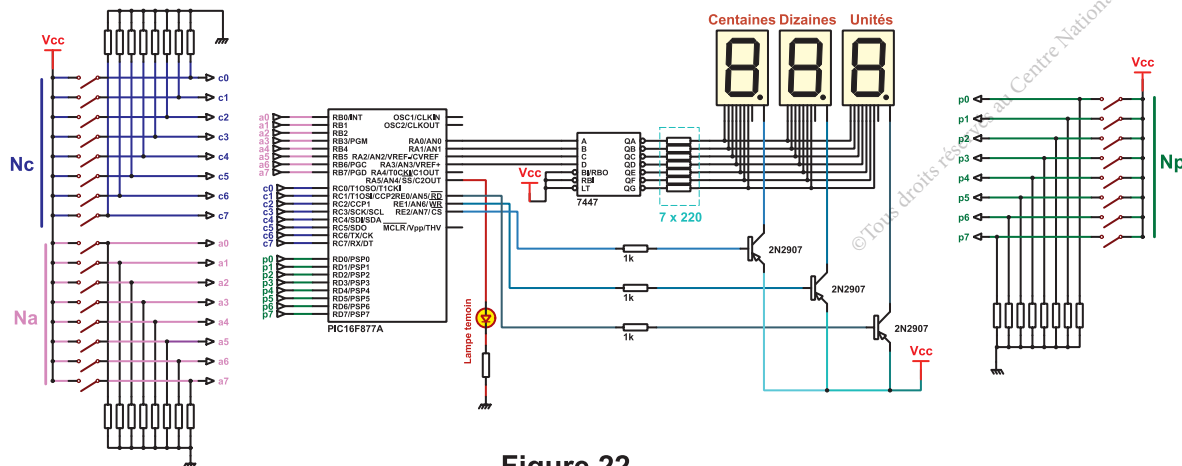


Figure 22



Projet n°1



Projet n°2



Projet n°3

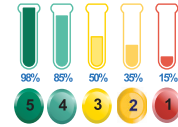


Projet n°4

Tous droits réservés au Centre National Pédagogique



Grille d'auto-évaluation



Thème 2	Activités				
Critères d'auto-évaluation:	Degrés d'appréciation:				
	5 Excellent	4 Très bien	3 Passable	2 Insuffisant	1 Faible
Compétences disciplinaires attendues					
CD 1.5: Résoudre un problème de logique combinatoire.					
J'ai réussi à identifier un circuit combinatoire.					
CD 2.5: Réaliser un montage à base de circuits combinatoires et analyser le fonctionnement					
J'ai réussi à analyser un circuit combinatoire.					
J'ai réussi à mettre en œuvre un circuit combinatoire intégré.					
J'ai réussi à mettre en œuvre une solution programmée.					
CD 3.5: Lire et décoder le schéma du circuit de commande d'un système combinatoire et en rendre compte.					
J'ai réussi à lire le schéma d'un circuit de commande d'un système combinatoire.					
J'ai réussi à décoder le schéma d'un circuit de commande d'un système combinatoire.					
Les compétences de vie et les éducations à...					
J'ai bien travaillé avec les autres en comprenant et en respectant leur diversité.					
J'ai appris à prendre les décisions d'une manière efficace.					
J'ai exprimé mes idées d'une manière claire, courte et dans un langage adapté à mes différents interlocuteurs.					

© Tous droits réservés au Centre National Pédagogique