**Exercices du cours d’algorithmique **

**D**

**d**

**Cours :** [**https://bit.ly/35cZ1Wa**](https://bit.ly/35cZ1Wa)

* **Utilisation d’une structure alternative \_(page 8\_du cours)\_\_\_\_\_\_\_**

**Exercice 1**

**Modifiez** le pseudo-code de l’algorithme « ***Calcul\_Surface\_Disque*** *»* (page 4 du cours) de telle sorte que dans le cas où **D ≤ d** le message : "Erreur de mesure" soit affiché. Ce nouvel algorithme sera identifié par « ***Calcul\_Surface\_Disque\_1*** *»*

**Exercice 2**

**Écrivez** le pseudo-code de l’algorithme « ***Déplacement\_1*** » décrivant l’arrêt d’un robot lorsque son capteur tactile (état logique vrai ou faux) heurte un obstacle. La variable de type booléen recevant l’état logique du capteur sera nommée **CTactile**.



**Capteur tactile**

Un sous-programme **Déplacer(Avant)** assure le déplacement du robot s’il n’y a pas d’obstacle. Un sous-programme **Arrêt()** assure l’arrêt du robot.

* **Utilisation d’une structure alternative imbriquée\_(page 9\_du cours)\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

D

d

**Exercice 3.** Pour préciser l’erreur prise en compte dans l’exercice 1, on vous demande de **modifier** le pseudo-code de la partie exécutive de l’algorithme « ***Calcul\_Surface\_Disque\_1****»* de telle sorte que

* Si D < d on affiche le message : "Erreur D < d",
* Si D > d on calcule S et on affiche sa valeur,
* Si D = d on afficher le message : "Erreur D = d"

Ce nouvel algorithme sera noté « ***Calcul\_Surface\_Disque\_2****»*

Contrainte : utiliser une structure alternative imbriquée.

**Exercice 4**

Face à un obstacle, un robot mobile doit avoir le comportement suivant :

- Si l’obstacle est situé à une distance D > 100cm, il se déplace en utilisant 70% de la puissance électrique disponible pour ses moteurs,

* Si l’obstacle est à une distance D < 30cm, il s’arrête.



* Dans l’intervalle [30cm, 100cm] le robot se déplace en utilisant 40% de la puissance.

La variable recevant la mesure de la distance est notée **D**. La variable recevant la puissance est notée **P.**

Un sous-programme **Déplacer(Avant, P)** assure le déplacement du robot. Un sous-programme **Arrêt()** assure l’arrêt du robot.

**Écrivez** le pseudo-code de l’algorithme « ***Déplacement\_2*** » décrivant le programme à implanter dans le robot.

Contrainte : utilisez une structure alternative imbriquée.

* **Utilisation d’une structure de choix multiple\_\_(page 10\_du cours)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercice 5**. Un capteur délivre un chiffre N représentatif de la couleur détectée.

AffCo

Couleur



**Capteur de couleur**

N

**Algorithme**

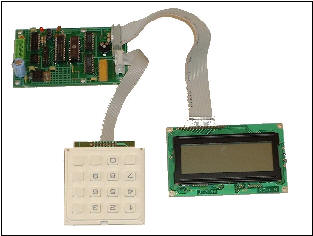
**Capteur de couleur**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **G. Phys** | **Couleur** | **Noir** | **Bleu** | **Vert** | **Jaune** | **Rouge** | **Blanc** |
| **Variables** | **N** | **1** | **2** | **3** | **4** | **5** | **6** |
| **AffCo** | **"Noir"** | **"Bleu"** | **"Vert"** | **"jaune"** | **"Rouge"** | **"Blanc"** |

**Écrivez** le pseudo-code de l’algorithme « ***Couleurs*** » affichant la couleur mesurée par le capteur. On appelle « AffCo » la variable contenant le texte à afficher.

Contrainte : utilisez une structure de choix multiple.

**Exercice 6**



Lors de l’appui sur la touche d’un clavier, celui-ci envoie un code (**IBM**). Ce code n’étant pas compatible avec celui utilisé par un afficheur LCD (**ASCII**), il est nécessaire d’effectuer une opération de **transcodage**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Touche appuyée** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Code touche clavier (IBM)** | 16h | 1Eh | 26h | 25h | 2Eh | 36h | 3Dh | 3Eh | 46h | 45h |
| **Code afficheur (ASCII)** | 30h | 31h | 32h | 33h | 34h | 35h | 36h | 37h | 38h | 39h |

Remarque: h signifie que la valeur est en hexadécimal (base 16).

**Algorithme**

**Transcodage**

**Code\_Touche**

**Code\_Afficheur**

**Clavier**

**Afficheur**

|  |  |
| --- | --- |
| Lors de l’appui sur une touche, le code IBM correspondant est placé dans une variable « **Code\_Touche** » de type entier.  Vous utiliserez la variable « **Code\_Afficheur** » pour mémoriser le code ASCII à envoyer à l’afficheur.  **Écrivez** le pseudo-code de l’algorithme "***Transcodage***".  Contrainte : utilisez une structure de choix multiple. | table-code-ascii  Table des codes ASCII |

**Exercice 7 (synthèse : la structure est à choisir parmi celles vues précédemment)**

Un site de contrôle des grandeurs physiques (température, humidité, etc.) mesurées dans une habitation doit pouvoir représenter la température entre 0 et 50°C. Le concepteur à retenu un widget jauge pour la température.

Cette jauge doit changer la couleur de son bord extérieur de telle sorte qu’il prenne la couleur de l’intervalle dans lequel se situe l’aiguille.

- 0 ≤ Temp ≤ 15°C : gris

- 15°C < Temp ≤ 30°C : jaune

- 30°C < Temp ≤ 40°C : orange

- 40°C < Temp ≤ 50°C : rouge

- > 50°C ou < 0 : noir

L’échelle de la jauge doit être modifiée comme ci-dessous lorsque la température est comprise dans l’intervalle ]18°C,23°C]

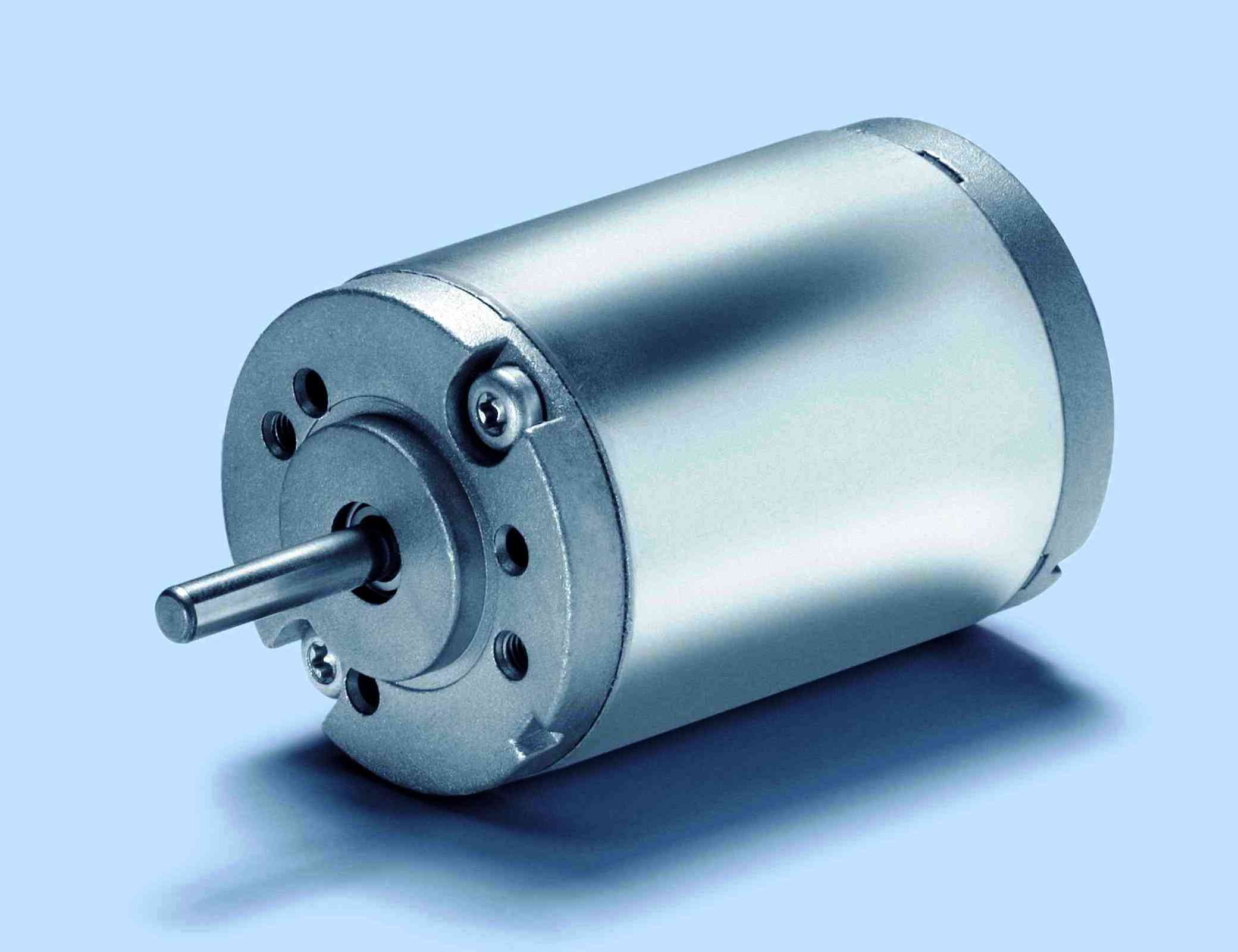
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 ≤ Temp ≤ 15°C | 15°C < Temp ≤18°C | 18°C < Temp ≤ 23°C | 23°C < Temp ≤ 30°C | Temp > 50°C |

La variable recevant la valeur de la température sera nommée **temp,** la variable recevant la couleur du bord sera nommée **couleurBord.** On dispose **des fonctions changerCouleurBord("**couleurBord**") et changerEchelle(**min,max,intervalle,pas**).** intervalle est l’écart entre 2 grandes graduations, le pas est l’écart entre deux petites graduations. L’algorithme demandé s’applique à une jauge existante.Aucune boucle n’est nécessaire.

**Écrivez** le pseudo-code de l’algorithme « **Widget*Température*** » assurant le comportement de la jauge.

* **Utilisation d’une boucle de comptage\_\_(page 11\_du cours)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercice 8** : **Commande d’un moteur à courant continu**



La vitesse angulaire d’un moteur à courant continu (aimants permanents) est proportionnelle à sa tension d’alimentation si la charge est constante (I=cst). L’allure de la représentation graphique (U) est donnée ci-dessous.

Ω(rad/s)

U(V)

I = cst

Un microcontrôleur peut régler la valeur moyenne d’une tension périodique avec une structure appelée timer.

Il est donc possible de faire varier la vitesse d’un moteur à courant continu avec un microcontrôleur.

Microcontrôleur

**Algorithme**

**Timer**

**Amplificateur de puissance**

**Moteur**

N

UT (V)

U(V)

Ω(rad/s)

Le moteur étant initialement à l’arrêt, on souhaite le démarrer « en douceur » en augmentant progressivement sa fréquence de rotation.

200

**N**

0 1 2 10

**t(s)**

20

Pour répondre au CdC, onpropose de faire évoluer le nombre entier N pour produire le signal numérique ci-contre.

Dans l’intervalle [0,10s] ce signal est appelé « **rampe numérique** », au-delà, il est appelé « **palier**».

On dispose d’un **sous-programme** **Attendre(t)**. t : nombre de secondes

**Écrivez** le pseudo-code de l’algorithme « ***RampeNumerique »*** qui génère ce signal.

**Exercice 9 : Commande d’un moteur à courant continu**

**Écrivez** le pseudo-code de l’algorithme « ***Profil*** » qui génère le signal ci-dessous.

0 1 2 15 30 45 t(s)

15

N

1

On applique ce signal au timer de l’exercice précédent. Quel est le comportement du moteur ?

* **Utilisation d’une boucle de comptage\_(page 11\_du cours)\_et d’un tableau\_(page 13)**
* **Le nombre de répétitions est connu**

**Exercice 10** : **calcul de l’énergie électrique consommée par un circuit alimenté en C.C.**

**Rappel : E(J) = U(V) × I(A) × t(s)** avec U : tension en Volt, I : intensité en Ampère et t : temps en secondes

La tension **U(v)** peut être considérée comme constante. L'intensité du courant **I(A)** varie dans le temps. La période de mesure (échantillonnage) est constante. On la nomme **Te.**

Exemple : U = 12V, I varie, Te = 5s

**Te**

**U(V)**

12

2

2,1

2,2

t(s)

t(s)

**I(A)**

0 5 10 15 20 25 30

0

|  |  |
| --- | --- |
| k | TabI[k] |
| 0 | 2 |
| 1 | 2,1 |
| 2 | 2,2 |
| 3 | 2 |
| 4 | 2,3 |
| 5 | 2,1 |

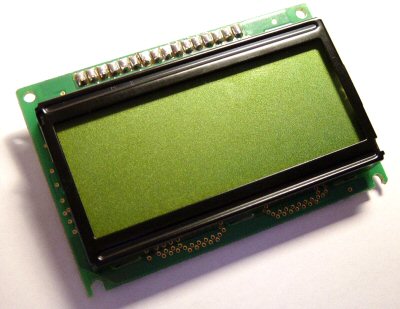
Le traitement à réaliser s’effectueaprès avoir placé **dix valeurs de I** dans un tableau tabI. **Écrivez** le pseudo-code de l’algorithme « ***Energie*** » qui calcule E.

**Exercice 11** : conversion binaire -> décimal

Rappel sur la conversion binaire -> décimal : https://bit.ly/3lvQbaM

**Écrivez** le pseudo-code de l’algorithme « **BinDec** » qui calcule la valeur décimale **N10** d’un nombre binaire **N2**positif codé sur **n** bits. N2 est contenu dans un tableau **tabN**.

**Exemple :** si N2 = 10000110 alors tabN = [1,0,0,0,0,1,1,0] => N10 = 134



**Exercice 12**

**Écrivez** l’algorithme d’un programme qui place successivement chacun des éléments du vecteur « **Code** » (tableau de dix octets) ci-dessous dans une variable « **Sortie** ». Une attente d’une seconde doit séparer chaque écriture.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Code** | 30h | 31h | 32h | 33h | 34h | 35h | 36h | 37h | 38h | 39h |

Quel est le résultat de ce programme si le contenu de la variable « Sortie » est envoyé sur un LCD affichant des codes ASCII ?

* **Le nombre de répétitions est inconnu**

**Exercice 13 : Gestion des feux d’une maquette de carrefour**

Les feux d’une maquette de carrefour sont gérés par un **mot de 6 bits** comme sur le schéma ci-dessous.

**b5** V

**b4** O

**b3** R

R **b3**

O **b4**

## V **b5**

**b2** **b1** **b0**

V O R

R O V

**b0** **b1** **b2**

Capteur

Jour/nuit

**Voie B**

**Voie A**

**Voie A**

**Voie B**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| **0** | **0** | V | O | R | V | O | R |
|  | | **Voie B** | | | **Voie A** | | |

On nomme « **Feux »** la variable de type **octet** associée à ce mot de six bits. Le contenu de cette variable est donné ci-contre.

On nomme « **jour »** la variable de type **booléen** associée au capteur Jour/nuit.

Le fonctionnement des feux doit répondre aux deux **cycles** ci-dessous.

**Mode jour (jour = vrai)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Voie A** | V | O | R | R | R | R |
| **Voie B** | R | R | R | V | O | R |
| **Temps** | 4s | 2s | 1s | 4s | 2s | 1s |

**Mode nuit (jour = faux)**

|  |  |  |
| --- | --- | --- |
| **Voie A** | **O** |  |
| **Voie B** | **O** |  |
| **Temps** | **1s** | **1s** |

**Écrivez** le pseudo-code de **l’algorithme « Carrefour ». Utilisez** deux structures répétitives **(tant que…)**