Contexte :

On reprend le cas de l’alarme verisecure :

Vous êtes technicien pour la société VERISECURE. Une commande d’alarme de voiture pour la société ALSET vient d’être passée.

Sauf que cette fois on veut garder la possibilité de modifier le comportement de l’alarme. Pour ce faire on utilise une carte arduino à la place des portes logique.

Rappelle l’alarme est déclenché si :

* Les feux sont allumés et le contact est coupé.
* La porte est ouverte et le contact est mis.

Question préliminaire :

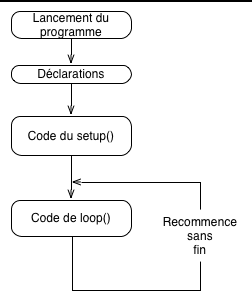
1. **Quelle sont les entrées du système ?**
2. **Quelle est la sortie du système ?**
3. **Faire une phrase qui relie la sortie aux entrées (les mots et/ou doive apparaitre)**
4. **Réaliser l’algorigramme du système**

Ressource

* Projet Proteus : Alarme.pdsprj
* Projet arduino : Alarme.ino
* Projet arduino : moniteur.ino
* Le logiciel arduino V1.8.11

Objectif :

**Le projet n’est pas fini c’est à vous de compléter le programme puis de le transférer dans le logiciel Proteus**

IDE arduino

1. **Ouvrez la dernière version d’arduino**

Dans tout projet arduino il y a :

* Déclaration des variables
  + Bool -> variable qui a la valeur 0 ou 1
* Deux fonctions :
  + Void setup ()
    - Permet d’initialiser la carte
  + Void loop ()
    - Fonction qui se répète indéfiniment

1. **Dans la dernière version d’arduino ouvrez le projet Alarme.ino**

Lecture des entrées

1. **A l’aide d’internet trouver à quoi sert l’instruction pinMode**
2. **Il manque l’instruction pinMode pour l’entrée contact. Ecrire l’instruction pinMode pour l’entrée correspondant au contact.**
3. **A l’aide d’internet trouver à quoi sert l’instruction digitalRead**

Les variables sont la mémoire d’un programme.

1. **Compléter le programme pour mémoriser l’état de l’entrée contact.**

Ressource de programmation

Il est possible de modifier l’exécution d’un programme avec des instructions de contrôle. L’une de ces instructions est l’**if** qui exécute ses instructions que si sa condition est vraie. Si sa condition est fausse il exécute l’**else** (optionnel).

If (condition)   
{  
 instruction  
} else   
{

Instruction

}

|  |  |
| --- | --- |
| Opérateur mathématique | Syntaxe |
| Addition | + |
| Soustraction | - |
| Division | / |
| Multiplication | \* |
| Affectation | = |

Tableau syntaxes

|  |  |
| --- | --- |
| Opérateur logique | Syntaxe |
| Ou | || |
| Et | && |

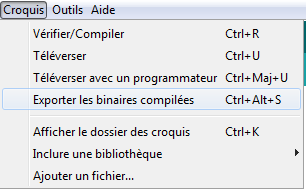
|  |  |
| --- | --- |
| Comparaison | Syntaxe |
| Egale | == |
| Supérieur | > |
| Supérieur ou égale | >= |
| Inférieur | < |
| Inférieur ou égale | <= |
| Différent | = ! |

Affectation des sorties

1. **Compléter le programme pour pouvoir contrôler la variable alarme**
2. **A quoi sert l’instruction digitalWrite**
3. **Affecter la sortie qui pilote l’alarme**

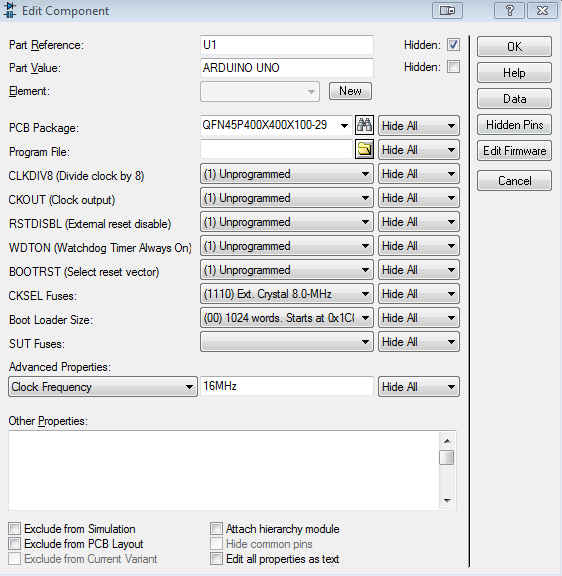
Transférer le programme

1. **A l’aide des indications suivante, transférer le programme dans Proteus**



Pour transférer le programme dans Proteus il faut

1. Exporter les binaires compilées dans arduino
2. Double cliquer sur l’arduino dans Proteus



1. Cliquer sur le symbole dossier et choisir le fichier Alarme.ino.standard.hex

Tester le programme

1. **Test si la simulation Proteus fonctionne comme prévus (faite vérifier par le professeur)**

Exercice pour approfondir

Exercice 1

Ecrivez un programme qui :

1. **Initialise 2 variables entières a et b (déclaration en int)**
2. **Déclare une variable nbmin également entière.**
3. **A l’aide d’un if, fait en sorte que la variable nbmin contienne la valeur minimale**
4. **Trouver comment utiliser le moniteur série intégré à arduino.** (S’inspirer du programme moniteur.ino)
5. **Affiche cette valeur minimale ainsi déterminée.**

Exercice 2

**Même exercice mais pour 3 nombres entiers**

Exercice 3

**Maintenant on veut trouver le maximum**

Exercice 4

**Modifier le programme pour que ce soit l’utilisateur qui init les variables**

Utilisé la fonction Serial.read () et la table ASCII

Exercice 5

**Ecrivez un programme qui affiche le signe du produit de a et b sans faire la multiplication.**

Exercice 6

Nous désirons afficher la mention obtenue par un élève en fonction de la moyenne de ses notes. S’il a une moyenne strictement **inférieure à 10, il est recalé**. S’il a une moyenne **entre 10 (inclus) et 12, il obtient la mention passable.** S’il a une moyenne **entre 12 (inclus) et 14, il obtient la mention assez bien.** S’il a une moyenne **entre** **14 (inclus) et 16, il obtient la mention bien.** S’il a une moyenne **supérieure à 16 (inclus) il obtient la mention très bien.** Ecrire les instructions nécessaires.

Exercice 7

Ecrire un programme qui teste si un nombre entier est pair. Dans le cas où le chiffre est **pair, vous afficherez un message du type ”Le chiffre X est pair”**, où X sera le chiffre que vous aurez préalablement déclaré. **Modifiez ensuite votre programme pour afficher également un message si le chiffre est impair.**

Vous utiliserez pour cela la fonction modulo (%). Rappel : l’opérateur “%” (ou modulo) renvoie le reste de la division (entière) entre 2 nombres (ex : 11%3 = 2).

Exercice 8

Une année est bissextile si

* L’année A est divisible par 4 et pas par 100
* L’année A est divisible par 400

Exemples :

* 1901 n’est pas bissextile car non divisible par 4
* 2004 est bissextile car divisible par 4 et pas par 100
* 2100 n’est pas bissextile car divisible par 4, divisible par 100 mais pas par 400
* 2000 est bissextile car divisible 400

**Ecrire un programme qui détermine si une année est bissextile ou non**