

Compétences attendues :

- Proposer une note de clarification définissant les objectifs, les tâches et le planning
- Communiquer au sein du groupe de projet et rendre compte de son travail oralement et par écrit
- Produire différentes architectures de solutions sous forme de schémas, d'algorithmes.
- Evaluer la réponse des solutions proposées au cahier des charges et les comparer
- Choisir le cas échéant des éléments standards
- Rechercher et partager des données informatiques
- Représenter la solution en utilisant l'outil informatique

Problématique :

Proposer, décrire et justifier les modifications nécessaires des paramètres et/ou du comportement d'un système pluritechnique existant pour répondre à l'évolution du besoin défini par un cahier des charges. (Configuration, commande, programmation, détection,...)

Support / Ressources :

- SYSTEME PORTAIL FAAC.
- DOSSIER TECHNIQUE.
- ELEMENTS DE COURS.

Evaluation :

Le mini-projet est évalué.

L'évaluation prend en compte les points suivants :

- Les travaux effectués pendant la période concernée.
- Le dossier fourni par le groupe de travail sous format numérique.
- Chaque élève présente sa contribution personnelle aux travaux du groupe (5 minutes de présentation et 5 minutes de questions).

-I- MISE EN SITUATION :

Cette étude s'appuie sur un système automatisé grand public : l'OUVRE PORTAIL FAAC.

L'ouverture et la fermeture d'une propriété peuvent être particulièrement contraignantes dans les situations suivantes :

- Manœuvre d'un portail grand et lourd nécessitant un certain effort.
- Portail éloigné de l'habitation demandant un déplacement important.
- Passage d'un véhicule devant s'arrêter avant et après l'ouverture du portail.
- Manœuvre du portail par un enfant ou une personne handicapée.
- Manœuvre d'un portail par mauvais temps.

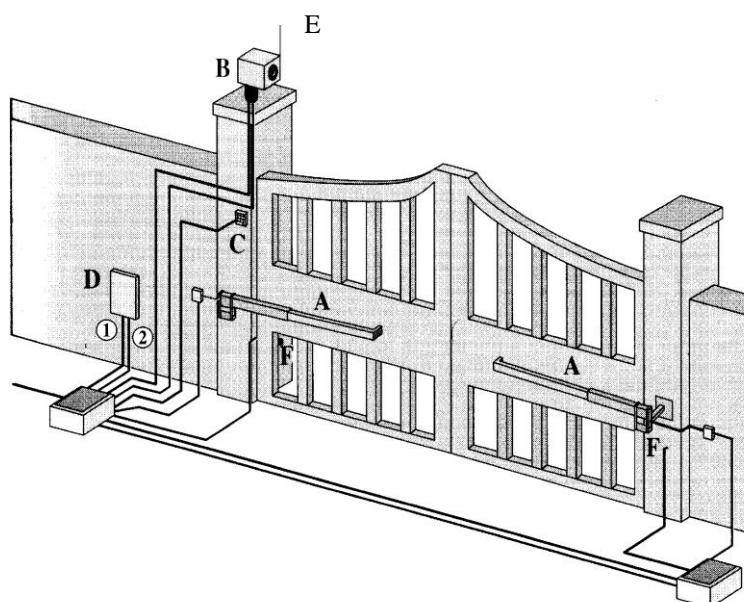
L'ouvre portail automatisé doit donc permettre de réduire, voire de supprimer ces contraintes d'utilisation.

Un portail est généralement constitué de deux vantaux, identiques ou différents. Pour automatiser le portail, chaque vantail doit être équipé d'un actionneur.

La partie commande est commune aux deux actionneurs.

Les ordres d'ouverture peuvent être donnés de différentes façons :

- au clavier à code (Digikey).
- A l'aide d'une télécommande.
- A l'aide d'une commande à clé.
- A l'aide de boutons poussoirs, présents dans la maison, par exemple.



(A) Opérateur FAAC 402 CBC, englobant le moteur, la pompe et le vérin.

(B) Lampe clignotante.

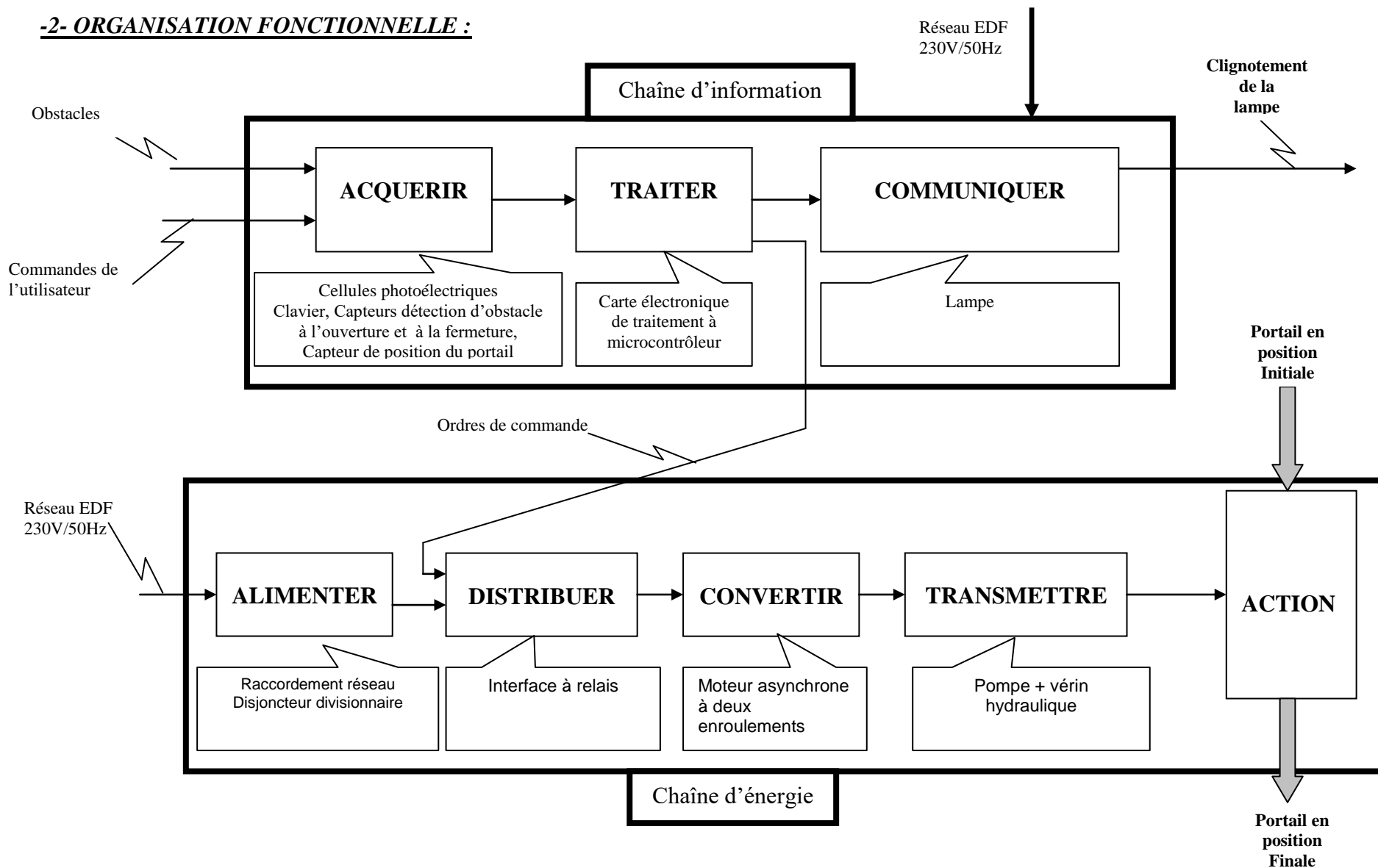
(C) Clavier à touches Digikey.

(D) Boîtier électronique de commande.

(E) Antenne 433 MHz.

(F) Cellules photoélectriques.

-2- ORGANISATION FONCTIONNELLE :



-3- PROBLEME TECHNIQUE POSE :

Le système étudié possède un seul vantail

La transmission des informations par la télécommande n'est pas à étudier

Cahier des charges.

- Les réglages de l'intensité de la force d'ouverture et de fermeture sont indépendants l'un de l'autre. Ils sont réalisés par l'intermédiaire de soupapes by-pass et peuvent être dangereusement réglés suivant les régions sollicitées par des vents de fortes puissances, ce qui risque d'occasionner des écrasements de personnes ou d'enfants en bas âge. La version de la carte de commande du portail étudié ne prévoit pas de sécurité supplémentaire concernant un éventuel écrasement d'une personne lors d'un mouvement d'ouverture du battant du portail.
- Cette version ne permet pas un échange communicatif satisfaisant entre l'utilisateur et le système.
- Cette version ne détecte pas les butées extrêmes (ouverture/ fermeture) du vantail, elle est basée sur une temporisation et donc dépense de l'énergie inutilement. Il est donc possible d'économiser significativement (Nombre de cycles par an) une énergie non négligeable.

On se propose de résoudre les problèmes suivants :

- Insérer un écran LCD permettant de renseigner l'utilisateur sur la procédure à suivre pour mettre en mouvement le système ?
- Choisir et insérer de nouveaux capteurs permettant d'assurer la sécurité anti-écrasement et de détecter les butées extrêmes ?
(Types de capteurs, implantation, schéma de raccordement,...)

Cycle de fonctionnement à prendre en compte pour les capteurs de sécurité:

Après un appui sur une des commandes, deux cas sont à envisager :

➤ **Phase d'ouverture.**

Capteur de sécurité fermeture \Rightarrow aucun effet.

Capteur de sécurité ouverture \Rightarrow le vantail se ferme immédiatement.

➤ **Phase de fermeture.**

Capteur de sécurité ouverture \Rightarrow aucun effet.

Capteur de sécurité fermeture \Rightarrow le vantail s'ouvre immédiatement.

- Programmer une nouvelle carte à microcontrôleur en utilisant un langage approprié (Flowcode ou Arduino) afin de prendre en compte les nouvelles exigences ?
Inventaires des entrées et des sorties, schémas de raccordement, algorithme, programme, modèles numériques, essais, analyse des écarts
- Concevoir le positionnement des capteurs sur la solution existant du laboratoire.

Attention : Avant de commencer le mini-projet, il faudra effectuer une analyse complète du travail à faire afin de déterminer le type et le nombre des entrées/sorties ainsi que les contraintes spécifiques liées au projet (choix et implantation des capteurs, logiciels et langage de programmation utilisés) dans le but de faire le lien entre les différentes tâches de chaque intervenant.

Groupe	Répartition des tâches
Elève 1	<p>Concevoir et réaliser une maquette numérique du schéma électrique permettant de relier toutes les parties du système (chaîne d'information et chaîne d'énergie sans oublier l'alimentation en 5V de la carte arduino à partir du secondaire du transformateur 25V-50hz)</p> <ul style="list-style-type: none"> - Elaboration du schéma structurel complet sur proteus. - Simulation et essais. Analyse des écarts. Etablir le lien avec les parties étudiées par les coéquipiers.
Elève 2	<p>Concevoir et réaliser une maquette numérique de la carte de traitement des données qui consiste à acquérir les informations provenant d'un clavier à touches. On s'attachera à instaurer sur un écran LCD un dialogue entre l'utilisateur et le système (échange interactif) pour le guider dans la procédure d'ouverture ou de fermeture du portail.</p> <ul style="list-style-type: none"> - Elaboration et simulation du programme (Flowcode ou Arduino). - Simulation et essais. Analyse des écarts. Etablir le lien avec les parties étudiées par les coéquipiers.
Elève 3	<p>Concevoir et réaliser une maquette numérique permettant la commande et l'alimentation en énergie du moteur du portail lorsque le code d'accès est correct (une clé ou bouton poussoir pourront également commander le système. Il faudra prendre en compte les positions extrêmes du portail et la sécurité anti-écrasement.</p> <ul style="list-style-type: none"> - Choix des capteurs. (détection des positions extrêmes du portail et de la sécurité anti-écrasement) - Elaboration du programme (Flowcode ou Arduino). - Simulation et essais. Analyse des écarts. Etablir le lien avec les parties étudiées par les coéquipiers.

-4- PISTES A SUIVRE POUR REALISER LE PROJET :

Pour le Choix du ou des capteurs et leurs implantations.

Vous devrez faire des recherches sur la documentation technique présente dans le laboratoire et/ou en utilisant Internet puis faire une synthèse de toutes les informations réunies.

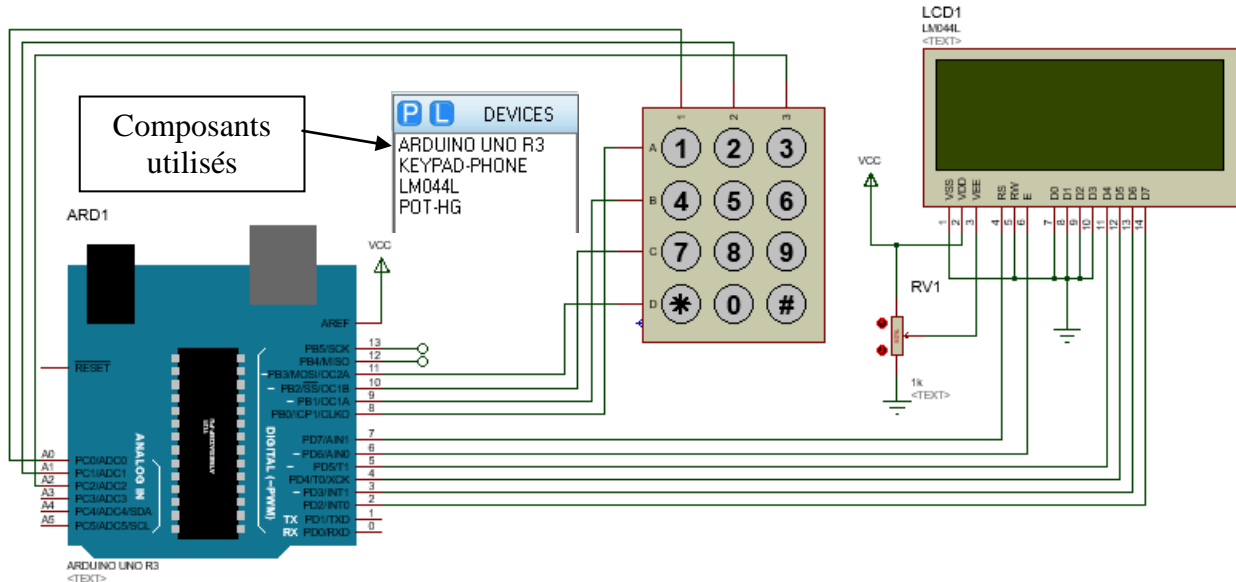
- N'oubliez pas que vous êtes plusieurs groupes à faire le même projet et la meilleure solution sera donc mieux appréciée.
- Pour l'implantation des capteurs, il faudra bien sûr utiliser un logiciel permettant éventuellement de dessiner les différentes pièces à fabriquer et/ou préciser à l'aide d'une maquette numérique leur positionnement.
- Il faudra déterminer le raccordement avec la carte de commande qui permettra le traitement des données.

Pour la programmation et les essais.

- Vous devrez dans un premier temps faire l'inventaire des entrées et des sorties en indiquant pour chacune d'elles le type d'information reçue ou envoyée ainsi que l'état des relais de sortie pour configurer le portail en ouverture ou en fermeture.
- Pour la programmation de la carte à microcontrôleur, vous demanderez au professeur le document d'aide à la programmation.
- Pour la recherche de la maquette numérique sur proteus, il faudra prendre en compte tous les TD et TP réalisés précédemment.
- Le raccordement des différents éléments composant le système sera effectué par le professeur mais vous devrez établir au préalable les schémas électriques.
- Enfin des essais seront effectués pour vérifier à la fois, le bon fonctionnement du système, et, le respect du cahier des charges.

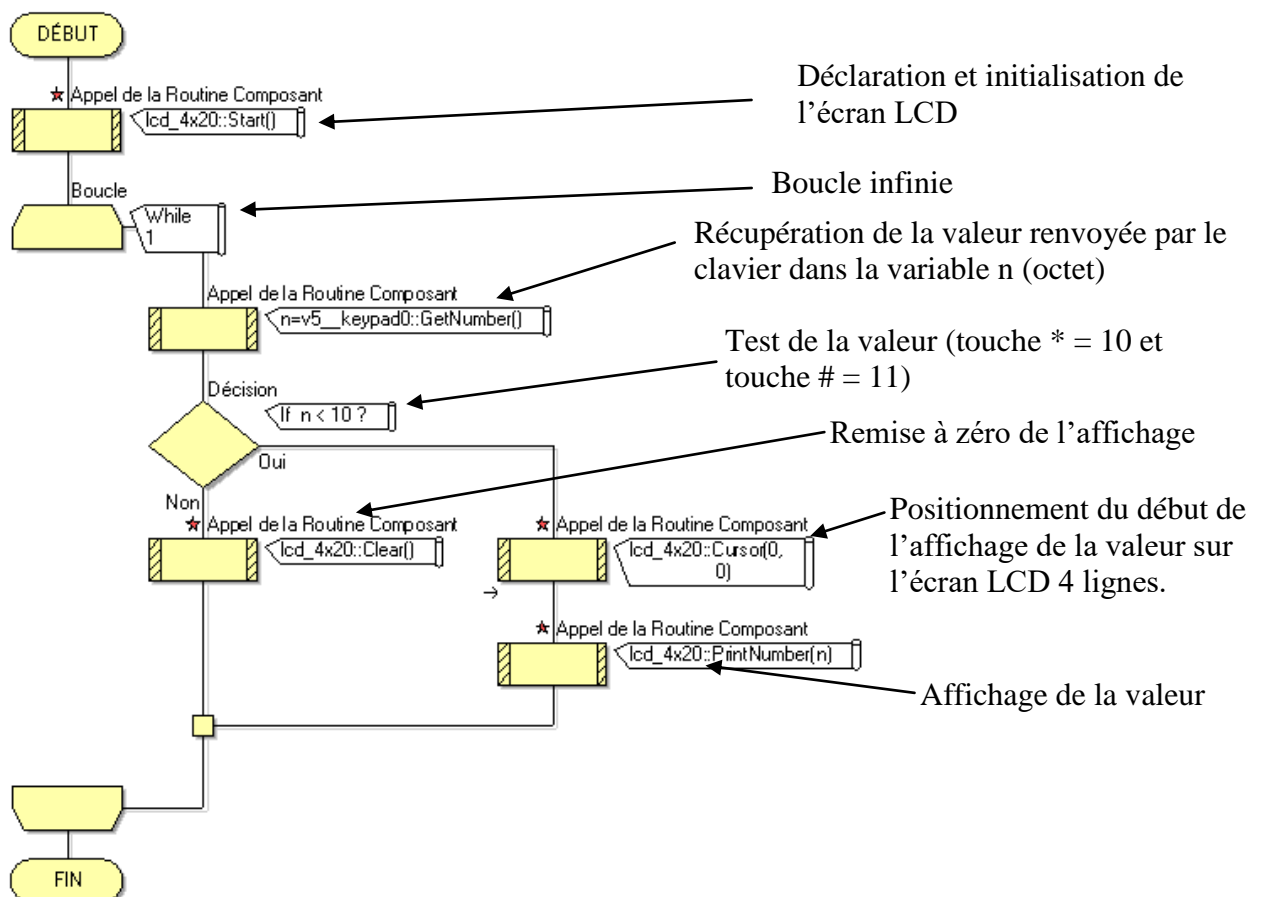
Pour vous aider dans votre démarche de programmation et de conception de la carte de commande, les documents qui suivent vous seront utiles.

Connexion d'un afficheur LCD et d'un clavier à une carte arduino dans Proteus



Pour afficher la touche saisie au clavier sur l'écran LCD. Nous n'afficherons pas les touches * et #.

Le programme flowcode est le suivant :



Le programme en langage C++ (logiciel arduino) est le suivant.

Il y a une petite différence entre ce programme et le précédent. Le chiffre restera affiché après l'appui sur une touche jusqu'au prochain appui. Le fait d'appuyer sur les touches * ou # annulera l'affichage.

```
// --- Que fait ce programme ? ---
/* Affiche sur un LCD les touches tapées sur un clavier
matriciel*/
// --- Fonctionnalités utilisées ---
// Utilise un afficheur LCD alphanumérique 4x20 en
mode 4 bits
// Utilise un Clavier matriciel 3x4 (12 touches)

//***** Entête déclarative *****
// A ce niveau sont déclarées les bibliothèques, les constantes,
les variables...
// --- Inclusion des bibliothèques utilisées ---
#include <LiquidCrystal.h> // Inclusion de la bibliothèque
pour afficheur LCD
#include <Keypad.h> // inclusion de la bibliothèque pour
clavier matriciel

// --- Déclaration des constantes ---
//--- Constantes utilisées avec le clavier 3x4
const byte LIGNES = 4; // 4 lignes
const byte COLONNES = 3; //3 colonnes
// --- constantes des broches ---
const int C3=A2; //déclaration constante de broche
const int C2=A1; //déclaration constante de broche
const int C1=A0; //déclaration constante de broche
const int RS=7; //déclaration constante de broche
const int E=6; //déclaration constante de broche
const int D4=5; //déclaration constante de broche
const int D5=4; //déclaration constante de broche
const int D6=3; //déclaration constante de broche
const int D7=2; //déclaration constante de broche
const int L4=11; //déclaration constante de broche
const int L3=10; //déclaration constante de broche
const int L2=9; //déclaration constante de broche
const int L1=8; //déclaration constante de broche

// --- Déclaration des variables globales ---
//--- Définition des touches
char touches[LIGNES][COLONNES] =
{
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'},
};
// tableaux de lignes et colonnes
byte BrochesLignes[LIGNES] = {L1, L2, L3, L4};
//connexions utilisées pour les broches de lignes du
clavier
byte BrochesColonnes[COLONNES] = {C1, C2, C3};
//connexions utilisées pour les broches de colonnes du
clavier

char touche; // variable de stockage valeur touche
appuyée

// --- Déclaration des objets utiles pour les fonctionnalités
utilisées ---
LiquidCrystal lcd(RS, E, D4, D5, D6, D7); // Création
d'un objet LiquidCrystal = initialisation LCD en mode 4
bits
```

```
// création d'un objet keypad = initialisation clavier
Keypad clavier = Keypad( makeKeymap(touches),
BrochesLignes, BrochesColonnes, LIGNES,
COLONNES );

//***** FONCTION SETUP = Code
d'initialisation *****
// La fonction setup() est exécutée en premier et 1 seule
fois, au démarrage du programme

void setup() { // debut de la fonction setup()

// --- ici instructions à exécuter au démarrage ---

lcd.begin(20,4); // Initialise le LCD avec 16 colonnes x 2
lignes

delay(10); // pause rapide pour laisser temps initialisation

// Test du LCD

lcd.print("LCD OK"); // affiche la chaîne texte - message
de test
delay(2000); // pause de 2 secondes

lcd.clear(); // // efface écran et met le curseur en haut à
gauche
delay(10); // pour laisser temps effacer écran

} // fin de la fonction setup()

//***** FONCTION LOOP = Boucle sans
fin = coeur du programme *****
// la fonction loop() s'exécute sans fin en boucle aussi
longtemps que l'Arduino est sous tension

void loop(){ // debut de la fonction loop()

// --- ici instructions à exécuter par le programme
principal ---

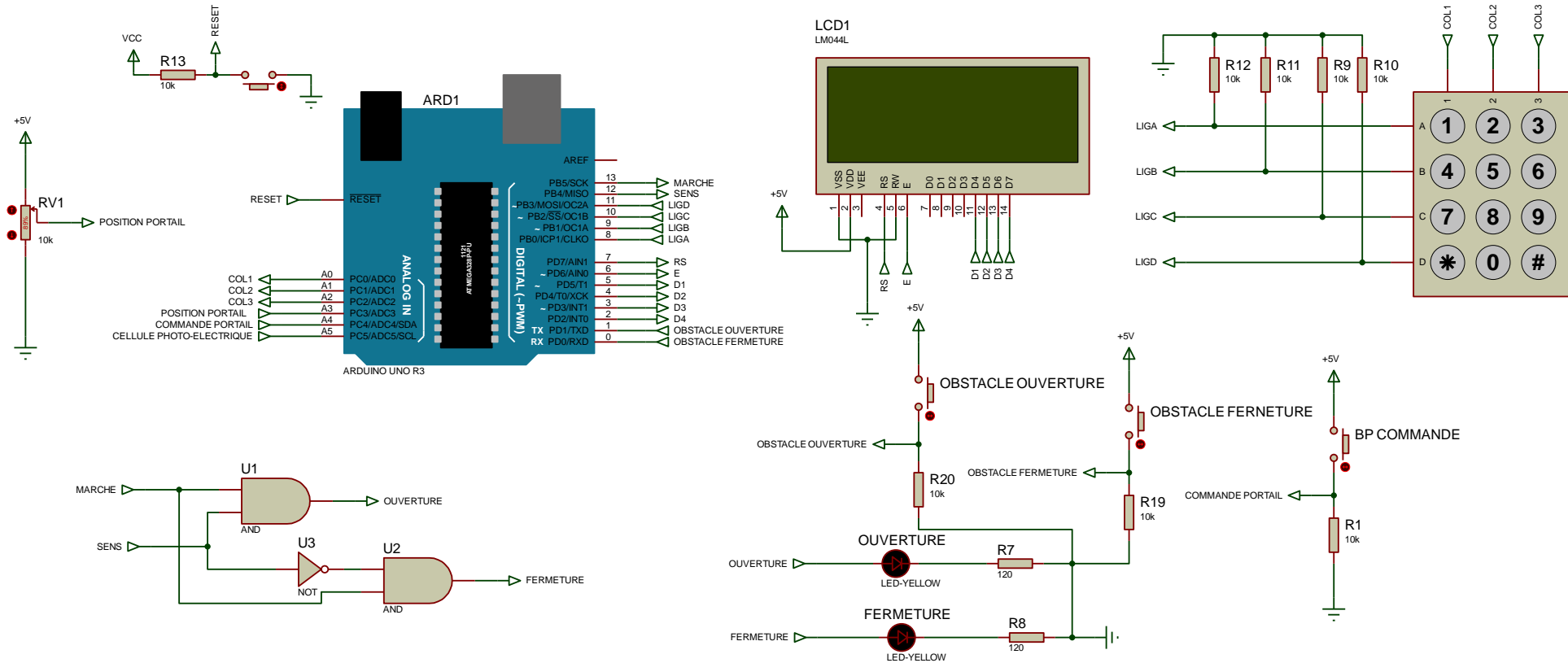
touche = clavier.getKey(); // lecture de la touche appuyée

if (touche != NO_KEY)
{ // si une touche a été frappée -- gestion de la
touche appuyée
  if (touche=='#'or touche=='*')
  {
    lcd.clear(); // efface écran si appui #
ou *
  }
  else //sinon affiche touche
  {
    lcd.setCursor(0,0);
    lcd.print(touche);
    delay(300); // pause entre 2 appuis
  }

} // fin gestion de la touche appuyée

} // fin de la fonction loop() - le programme recommence
au début de la fonction loop //sans fin. Fin du programme
```


SCHEMA DE LA CARTE DE COMMANDE



SCHEMA DE L'INTERFACE DE PUISSANCE

