Produit : Barrière automatique

L'objectif de cette activité est de maquetter une solution de gestion de l'ouverture et de la fermeture d'une barrière automatique.

Description de la démarche :

- la première partie doit permettre d'appréhender le système et de comprendre la problématique posée ;
- dans la seconde partie, une solution décrivant le comportement du système est proposée et étudiée grâce à une simulation ;
- la troisième partie consiste à choisir les capteurs et à les intégrer à un modèle numérique représentant le câblage du système ;
- enfin, en dernière partie, une maquette est assemblée, puis un code source est intégré à cette maquette. La maquette permet de tester et valider le respect des exigences par la solution.

Les quatre parties doivent être traitées dans l'ordre proposé.

1. Découverte du produit et de la problématique technique

Découvrir le produit et prendre connaissance de la problématique, de son contexte, du diagramme des cas d'utilisation (Figure 2) et du diagramme des exigences (Figure 3).

- **Identifier** les conditions nécessaires pour que la barrière se lève. **Préciser** les deux exigences correspondant à ces conditions.
- **Identifier** les conditions nécessaires pour que la barrière se baisse. **Préciser** les deux exigences correspondant à ces conditions.
- **Identifier** la durée minimale durant laquelle un usager doit être prévenu avant que la barrière ne commence à se lever / baisser.

2. Simulation

L'objectif de cette partie est d'étudier et valider une proposition de solution décrivant le comportement du système, puis de convertir cette proposition en algorigramme.

- À l'aide du diagramme d'états simulant le comportement de la barrière automatique (fichier ba_stm.html), analyser le fonctionnement du système, puis **lister** la succession d'événements nécessaires pour que :
  - la barrière se lève, en partant de l'état *ATTENTE* ;
  - le système refuse d'ouvrir la barrière, en partant de l'état *ATTENTE* ;
  - la barrière se baisse, en partant de l'état de l'état *OUVERTURE* ;
- **Conclure** sur le respect des conditions nécessaires pour que la barrière s'ouvre ou se baisse par ce diagramme d'états.
- **Compléter** l'algorigramme partiel du système (Figure 5) à partir des observations faites sur le diagramme d'états, en indiquant à quoi corresponde les blocs numérotés 1 à 4 parmi ces propositions : *gateOpening()*, *accessDenied()*, *Véhicule en attente devant la barrière et barrière baissée ?* et *Aucun véhicule dans la zone de détection et barrière levée ?*

3. Conception

L'objectif de cette partie est de choisir un capteur parmi 3 propositions, puis d'intégrer le capteur choisi dans un modèle numérique en réalisant son branchement.

Rappel : la zone de détection fait 2 m de large, et le véhicule 1,4 m de large.

Contrainte de conception : Pour des raisons de simplicité dans le prototypage, un capteur produisant un simple signal de sortie logique est à privilégier par rapport aux capteurs à interface I²C, SPI, CAN, etc.

- Parmi les 3 capteurs à disposition, **choisir** le composant le plus adapté et **justifier** ce choix.
- **Réaliser** un schéma mettant en évidence la zone de détection et montrant comment les capteurs doivent être positionnés (schéma similaire à la Figure 1).

Choix de conception : La suite de l'étude est réalisée avec le capteur barrière F5-250. Ce capteur se compose de deux modules (un émetteur et un récepteur). Chaque module comporte 2 fils pour l'alimentation (rouge et noir). Le module récepteur possède un troisième fil (blanc) à relier à une entrée/sortie digitale.

Caméra : Le fonctionnement de la caméra n'est pas étudié. Dans la suite, la caméra est remplacée par un commutateur dont on fait passer la sortie à l'état haut si la caméra valide une plaque d'immatriculation, ou à l'état bas sinon.

- **Compléter** le modèle numérique du montage électronique fourni (fichier ba_cablage.fzz) de manière à connecter les 2 capteurs F5-250 à l'Arduino Uno. Ce schéma est aussi fourni en version PDF (fichier ba_cablage-base.pdf).

4. Expérimentation

L'objectif de cette dernière partie est de maquetter la solution puis de réaliser une expérimentation pour valider la solution retenue et son implémentation.

- À partir du matériel disponible, **assembler** la maquette en positionnant les 5 modules sur la plaquette perforée.

- **Câbler** la maquette, en suivant le modèle numérique du montage électronique.
- Ouvrir le code source fourni (fichier ba_code.ino), puis **compléter** le début du code source en remplaçant les symboles *??* par les bons numéros de broches (lignes 3 à 8).
- **Compléter** le code source en remplaçant les symboles *??* de la fonction **setup()** par le bon mode (lignes 16, 17 et 22) parmi : *OUTPUT*, *INPUT* et *INPUT_PULLUP*
- **Compléter** le code source en remplaçant les symboles *??* de la fonction **loop()** par la bonne fonction (lignes 28, 29 et 31) parmi : *digitalWrite*, *digitalRead*, *analogWrite* et *analogRead*
- **Compléter** le code source en remplaçant les symboles *????* de la fonction **loop()** par la bonne fonction (lignes 33, 35 et 39) entre : *gateOpening*, *gateLocking* et *accesDenied*

- **Procéder** à une série de tests de la maquette afin d'analyser son comportement, puis **conclure** sur le fait que la solution maquettée produit bien le comportement attendu.
- **Proposer** un protocole expérimental permettant de valider l'exigence 1r du diagramme des exigences.
- **Mettre en œuvre** le protocole de validation et **conclure** sur le respect de l'exigence 1r.
- **Conclure** sur la capacité de la solution à répondre à la problématique du sujet.