

## SAE2 – Centrale d'alarme

Fredéric Imbert, Wilfried Uhring

### Objectif de la SAE

Le but du projet développé dans la SAE (Situation d'évaluation et d'enseignement) porte sur la conception d'une centrale d'alarme. Les ressources associées à cette SAE sont principalement le module d'enseignement Info2 ainsi que les bases en électronique vues au premier semestre ainsi que les sites web des fabricants des modules et cartes utilisés.

### Description du projet

Le cœur de la centrale d'alarme est une carte Curiosity HPC (voir figure 1) dont le coût est d'environ 30 €, embarquant un microcontrôleur de type 18F47Q10 très semblable à celui utilisé dans les ressources. Elle sera programmée à l'aide de l'environnement MPLAB X à travers le programmeur intégré directement sur la carte Curiosity. Il n'est donc pas nécessaire de recourir à un programmeur externe. On utilise le code configurateur MCC pour tous les périphériques.

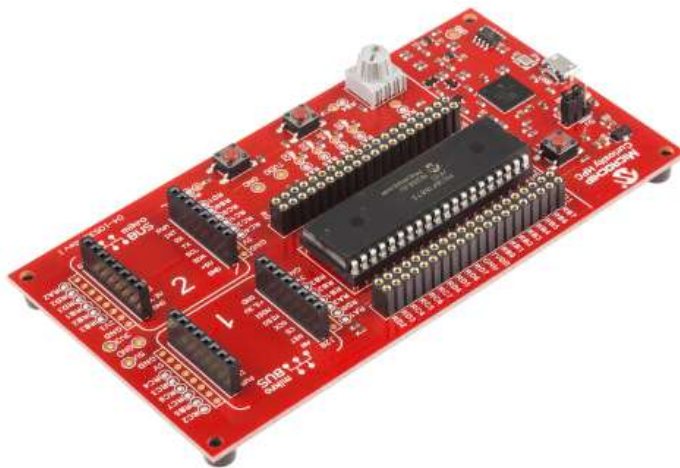


Figure 1: Carte Curiosity HPC

L'alarme est connectée à divers capteurs et actionneurs parmi lesquels on retrouve :

- ⇒ Clavier
- ⇒ 1 capteur IR pyrométrique
- ⇒ Ecran LCD 2 lignes
- ⇒ Accéléromètre
- ⇒ Un interrupteur (émulation fenêtre)
- ⇒ Une horloge temps réel (RTC Real Time Clock)
- ⇒ Transducteur Piézo pour émettre un son
- ⇒ Un potentiomètre
- ⇒ Un thermomètre LM35CZ
- ⇒ Une carte de liaison Bluetooth

Des capteurs supplémentaires peuvent être ajoutés en option selon votre avancée dans le projet.

- ⇒ Capteur de Gaz et humidité
- ⇒ Capteur de pression, baromètre
- ⇒ Un capteur connecté par liaison radio 433MHz - 868MHz

Le programme doit être écrit selon la philosophie suivante (voir figure 2).

- 1) L'ensemble de l'extraction des données issues des capteurs sera réalisé par interruption uniquement.
- 2) Les états et informations des capteurs sont placés dans un ensemble de données structuré.
- 3) La mise à jour des consignes de la centrale d'alarme est également effectuée dans la routine d'interruption.

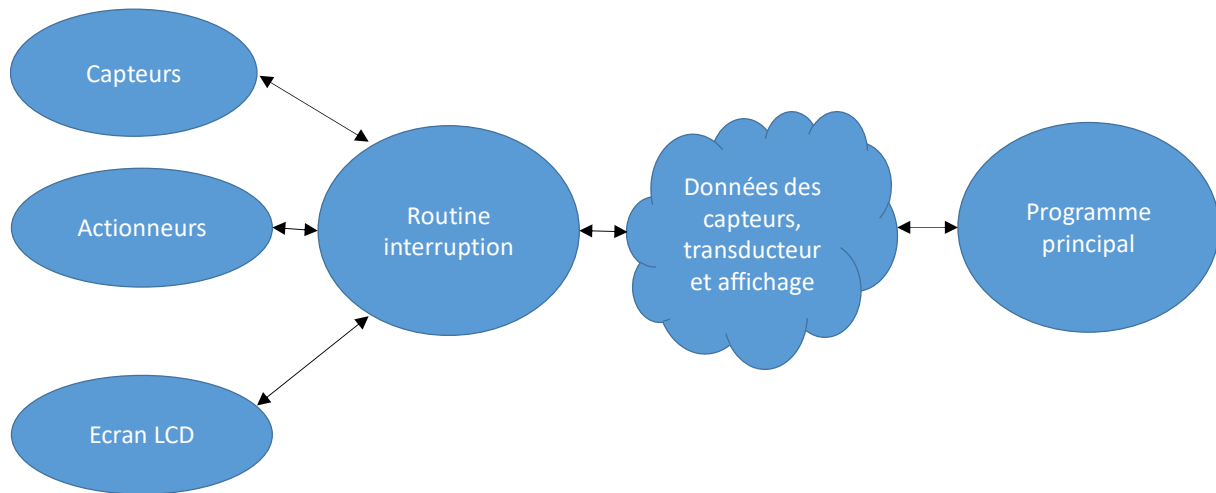


Figure 2. Architecture logicielle du projet d'alarme

Le comportement de la centrale d'alarme est décrit par une machine à état qui évolue par scrutation des données des capteurs. Cette machine à état constitue le programme principal. Le programme principal ne doit donc en aucun cas discuter directement avec un périphérique.

- 1) L'alarme doit pouvoir être activée et désactivée dans un premier temps au clavier en entrant un code de 4 chiffres.
- 2) L'alarme peut se déclencher sur plusieurs événements selon son mode de configuration
  - a. Mode partiel
    - i. L'alarme se déclenche uniquement lorsque l'interrupteur « fenêtre » est activé.
  - b. Mode alarme totale
    - i. L'alarme se déclenche sur tous les événements détectables.
- 3) L'alarme affiche en permanence l'heure et la température sur la première ligne de l'écran LCD et affiche « désactivée » ou bien « partielle » ou « totale » selon le mode.
- 4) Le niveau sonore de l'alarme est réglé à l'aide du potentiomètre.
- 5) Dans un deuxième temps, l'alarme doit pouvoir être activée/désactivée à distance à l'aide d'une application sur smartphone et de la liaison Bluetooth.
- 6) L'application doit également permettre de modifier des paramètres de capteur (ex : seuil de déclenchement)
- 7) On peut en option demander à l'alarme de se désactiver automatiquement à une heure précise.

### Démarches à suivre pour réaliser le projet.

Etape 0 :

Affecter les connexions des périphériques sur les pattes disponibles de votre microcontrôleur.

Etape 1 :

Valider l'utilisation de l'écran LCD. Vous devez pouvoir afficher un code hexadécimal passé en paramètre.

Etape 2 :

Valider l'utilisation du clavier (4x4). Vous devez pouvoir fournir une donnée qui représente l'état du clavier. La mise à jour de l'état doit être effectuée toutes les 20 ms.

- Si aucune touche n'est pressée, la donnée fournie sera 0xFF
- Si plusieurs touches sont pressées à la fois, on retourne le code 0xFE
- Si une seule touche est pressée, la donnée retournée est le code ASCII (à ski) correspondant à la touche

Afficher le code hexadécimal de l'état du clavier dans le programme principal.

Etape 3 :

Valider la lecture des capteurs suivants à intervalle de temps régulier de 100 ms

- Potentiomètre
- PIR
- Température

Afficher l'état ou la valeur de ces capteurs sur l'écran

Etape 4 :

Mise en route des capteurs I2C.

- Accéléromètre, avec une mise à jour toutes les 20 ms. L'accéléromètre permet de détecter que la centrale n'est pas déplacée
- RTC, avec une mise à jour toutes les 1000 ms

Afficher l'état ou la valeur de ces capteurs sur l'écran

Etape 5 :

Valider la génération du son de la sirène de l'alarme. Son intensité doit être pilotable par la valeur du potentiomètre. Le rapport cyclique du signal (modulation en largeur d'impulsion, soit Pulse Width Modulation - PWM) envoyé vers le transducteur piézo vous permet de régler l'intensité. La fréquence générée est de 554Hz pendant 100ms et 440Hz pendant 400ms.

On génère également un retour son clavier (un bip à 440 Hz d'une durée 200ms), lors de l'appui valide d'une touche.

Etape 6 :

Le rafraîchissement de l'écran est effectué toutes les 100ms. En fait, il affiche le contenu d'un tableau de 2x16 caractères (mémoire alphanumérique). Ce tableau contiendra sur la première ligne l'heure et la température. On s'occupera des informations de la première ligne dans une étape ultérieure. Sur la deuxième, on a le retour clavier « les étoiles » sur les 4 premiers caractères, et sur la fin de l'écran, on a l'état de l'alarme « désactivé » « partielle » ou « totale ».

Etape 7 :

L'ensemble des capteurs et actionneurs étant maîtrisé, on passe à l'écriture de la machine à état qui décrit le comportement de l'alarme. Dans un premier temps :

l'alarme est activée / désactivée par la lecture d'un code de 4 chiffres au clavier suivi d'une commande (A pour arrêt, B pour le mode partiel, C pour le mode total). A la détection d'un premier appui, on entre dans le mode de saisie du code. La touche « \* » permet de remettre à zéro toute la séquence de touche entrée, la touche « # » permet d'annuler le dernier appui (on peut appuyer plusieurs fois sur « # » pour annuler plusieurs appuis). Si aucune touche n'est pressée durant plus de 5 secondes, on annule la saisie.

On affiche des étoiles lors de l'entrée du code sur l'écran LCD.

Etape 8 :

Prise en compte des capteurs selon le mode de fonctionnement de l'alarme. Après activation, la sirène entre soit dans le mode alarme partielle, ou totale :

- En alarme partielle, la sirène se déclenche lors de la détection de l'interrupteur « fenêtre »
- En alarme totale, la sirène se déclenche lorsqu'un au moins un des 3 capteurs se déclenche (accéléromètre, fenêtre et PIR)

Lors d'un déclenchement de l'alarme, la sirène ne commence à sonner qu'après une durée de 30 secondes. Elle émet ensuite pendant 3 minutes puis s'arrête automatiquement. On peut arrêter la sirène à tout moment en rentrant à nouveau le code au clavier suivi de la touche « A ».

Etape 9 :

Contrôle à distance. On interface une liaison série/Bluetooth pour contrôler l'alarme à distance. L'application smartphone envoie une séquence de caractères correspondant au code à entrer au clavier avec le code de configuration A, B ou C suivi des caractères Carrier Return (0x13) Line Feed (0x10). Si le mot reçu n'est pas cohérent, aucune action n'est engendrée.

Etape 10 :

Il faut faire en sorte que l'alarme se désactive automatiquement à l'heure renseignée grâce à l'application smartphone. Pour mettre à jour l'heure, l'application envoie le code de l'alarme, suivi du code D, de l'heure d'arrêt puis les caractères Carriage Return (0x13) Line Feed (0x10). Si le mot reçu n'est pas cohérent, aucune action n'est engendrée.

Bonus

- Qualité de l'air
- Pression
- Détection d'ouverture de porte par analyse signal
- Liaison radio 433MHz - 868MHz

### Description du matériel

Le clavier proposé est un clavier matriciel 4x4 à membrane (voir figure 3), le détecteur pyrométrique dispose de 3 connections, deux pour l'alimenter et une pour le signal (figure 3). L'écran LCD dispose de 4 broches, deux pour la communication I2C et deux pour l'alimentation.



Figure 3: Photographie du clavier (gauche), du détecteur pyrométrique (centre) et de l'écran LCD avec communication I2C (droite).

L'horloge temps réel (RTC) dispose d'une pile et d'un oscillateur à Quartz d'horlogerie pour pouvoir indiquer le temps même lorsque le circuit n'est pas sous tension. Il dispose d'un bus de communication I2C (figure 4.a). Le capteur de température utilisé est un LM35CZ qui délivre une tension analogique proportionnelle à la température. Il dispose d'une broche de signal ainsi que deux broches d'alimentation (figure 4.b). Le module de communication Bluetooth est composé d'un circuit Bluetooth avec une antenne PCB intégrée. Il communique avec le microcontrôleur à l'aide d'une interface de type série (figure 4.c). L'accéléromètre est basé sur une puce MPU6050. Il délivre 3 accélérations et 3 axes de position (gyroscope). Il est interfacé à l'aide d'un bus I2C (figure 4.d)

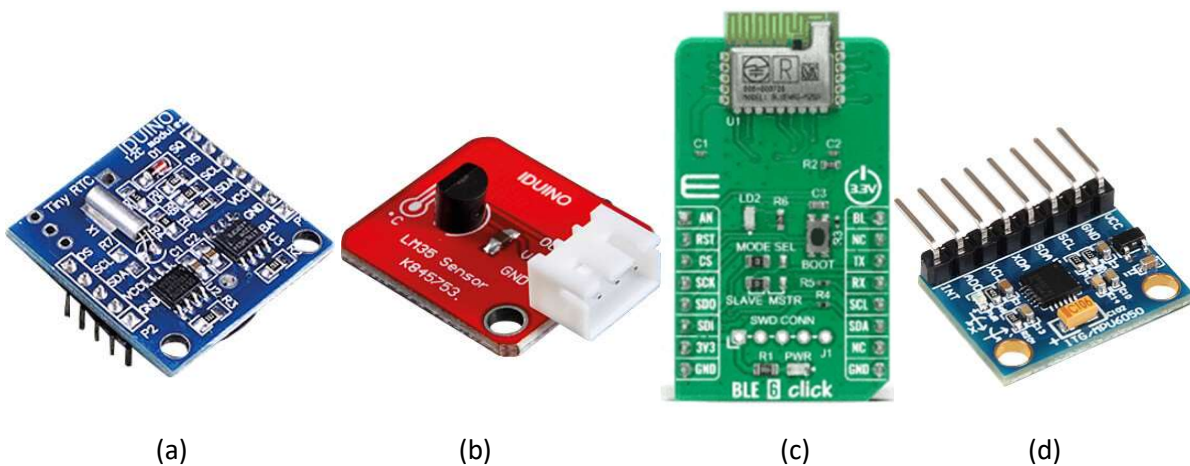


Figure 4 : (a) horloge temps réel, (b) capteur de température LM35CZ, (c) module de communication Bluetooth/série, (d) accéléromètre MPU6050