Groupe 84
Yasmine Tligui, Zaynab Hajroun // 341215 ,346235

30 mai 2023

RAPPORT DE PROJET : MICROCONTRÔLEURS
Your Smart Home

1. Introduction et description générale

Ce rapport présente le projet que nous avons réalisé dans le cadre du cours de microcontrôleur où nous avons développé un appareil "Smart Home". Ce projet axé sur la domotique vise à créer une solution qui permettra aux utilisateurs de contrôler et de gérer divers aspects de leur maison de manière efficace et centralisée. Il s'agit d'un système qui servira de centre de contrôle et de coordination des différents systèmes et appareils domestiques.

Le système est entièrement contrôlé par la télécommande qui permet à l'utilisateur d'afficher la température actuelle, régler la température, activer l'alarme anti-intrusion ou encore jouer de la musique.

Le réglage de la température permet de contrôler un système de climatisation et l'alarme permet d'informer le propriétaire qu'il y a eu une intrusion en son absence par un système de notification. Ces deux aspects ne sont pas gérés par notre microcontrolleur et aucun logement n'est réellement contrôlé, il s'agit ici uniquement d'une simulation.

Nous utilisons plusieurs modules différents dont le buzzer (M2) pour jouer la musique ou déclencher l'alarme. Cette dernière est déclenchée si le détecteur de distance (Sharp GP2Y0A21) détecte une présence à une certaine distance. La température est détectée par un capteur de température communiquant en 1-wire (M5). De plus, l'utilisateur est en mesure de régler la température qui sera affichée sur l'écran LCD, et on supposera qu'il existe un système de climatisation et de chauffage externe auquel l'information est envoyée.

2. Mode d'emploi

Nous avons imaginé un système qui présente une interface d'accueil lorsqu'aucune action n'est entreprise par l'utilisateur. Le système affiche le message d'accueil "Your smart home" et est dans l'attente d'une action provoquée par la télécommande pour effectuer l'une des actions suivantes :

a. Alarme anti-intrusion :

Notre système présente un mode de sécurité qui permet à l'utilisateur d'activer une alarme anti - intrusion en son absence, lorsqu'une présence est détectée à 20cm de notre capteur.

Ce mode peut être activé en appuyant sur le bouton 1 de la télécommande. Lors d'une détection l'alarme retentit et un message "Warning" apparaît sur le LCD.
A la fin de la sonnerie d'alarme, le système de notre smart home retourne à l'écran d'accueil. Lorsque ce mode est activé, l'utilisateur ne peut accéder aux autres fonctionnalités de notre système. Il est identifiable grâce au texte : "Alarm on", qui s'affiche lorsqu'il est en cours. Si le mode de sécurité n'est pas activé, il est possible d'accéder aux autres modes offerts par notre système. C'est un mode qui est supposé être activé lorsque l'utilisateur quitte son domicile, c'est pourquoi nous avons jugé inutile d'avoir accès aux autres fonctionnalités.

b. Mesurer la température actuelle :

Il est possible d'afficher la température du logement en appuyant sur la touche numéro 2. Au début de l'activation s'affiche le message "Current temp", avant d'afficher la température de la maison. La température s'affiche sur l'écran pendant 8 secondes pendant lesquels nous pouvons la voir varier. A la fin des 8 secondes, le message "end" s'affiche et l'utilisateur sera renvoyé à la page d'accueil.

c. Climatisation et chauffage :

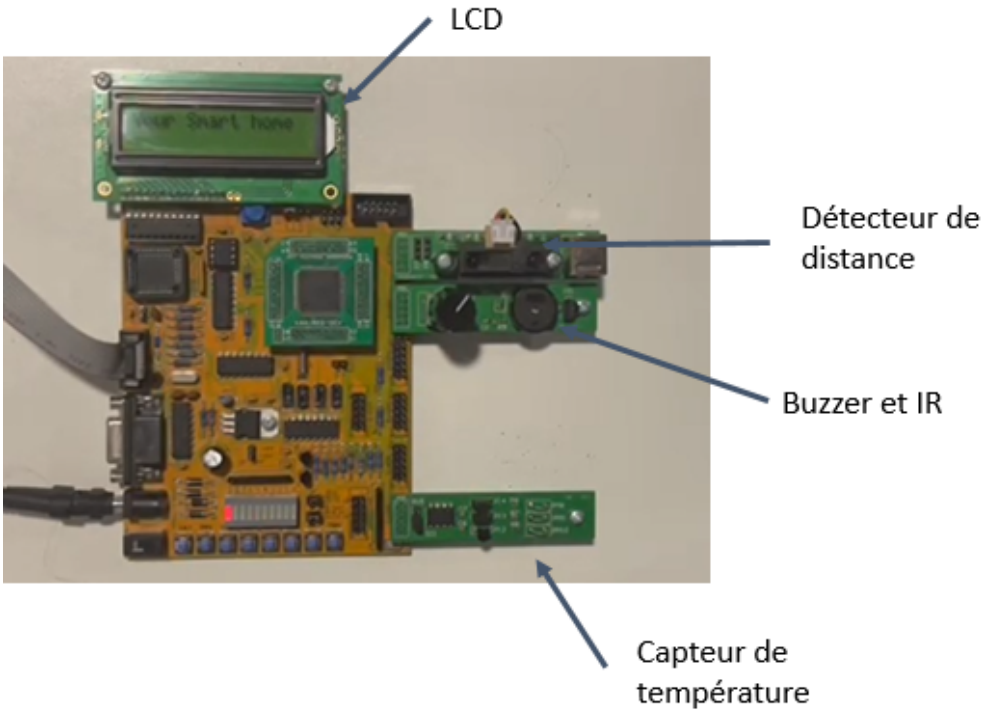
En plus de pouvoir mesurer la température du logement, il est également possible de la régler grâce aux boutons 5 pour l'augmenter, et 8 pour la diminuer. La température chosie sera envoyée au système de climatisation de la maison. L'augmentation ou la diminution de cette dernière se fait unité par unité. Pour quitter ce mode, il suffit d'appuyer sur la touche 4 de la télécommande.

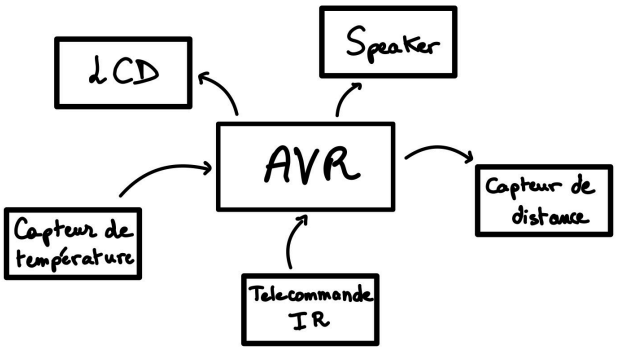
d. Musique du jour :

Outre les fonctionnalités citées, il est également possible de découvrir la musique du jour proposée par l'intelligence artificielle en appuyant sur la touche + de la télécommande.
En réalité, il s'agit d'une musique que nous avons programmé mais nous imaginons le microcontrôleur fonctionner de cette manière.

3. Description technique de l'application et du matériel

Comme expliqué précédemment, nous utilisons les cinq périphériques suivants : le LCD, la télécommande, le buzzer, le 1-wire (capteur de température) ainsi que le détecteur de distance. Ce choix de périphériques impose une gestion des ports particulière.
Premièrement, le LCD constitue notre principal moyen de communiquer des informations avec l'utilisateur. Ce périphérique influence les ports A et C c'est pourquoi nous avons fait le choix de ne pas les utiliser.
Par ailleurs, le détecteur de distance devant être branché sur le port F pour son bon fonctionnement, nous utilisons celui-ci dans ce but.
Afin d'utiliser le buzzer et le capteur infrarouge ainsi que le capteur de température, nous les branchons respectivement aux ports E et D.

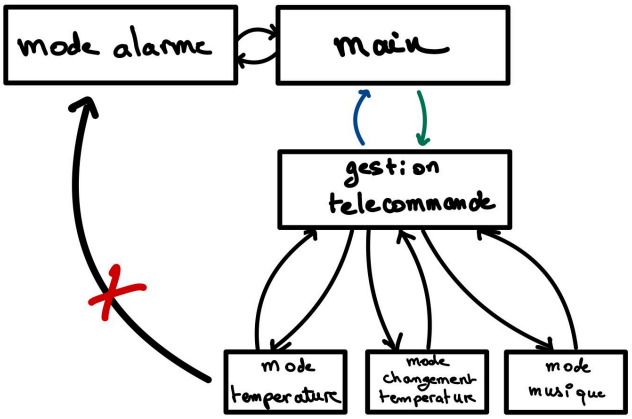




4. Fonctionnement du programme et présentation des modules

Notre programme est organisé en modules indépendants reliés par un “main” qui constitue le fichier que nous téléchargeons sur la carte.
Dans celui-ci, nous configurons tous les ports dans un premier temps, nous initialisons l’affichage et autorisons l’interruption pour le timer0.
L’affichage du message d'accueil est inconditionnel, puis le système vérifie si le mode alarme est demandé par l’utilisateur et effectue le branchement dans ce cas là. Sinon, le programme vérifie si une autre touche est activée en faisant appel au module gestion_télécommande.
Chacun des modules gère un aspect différent de notre programme tel que la télécommande, la musique, l’alarme ou encore les différents modules de température. Ainsi, à la fin de chaque appel de routine le programme est renvoyé vers le main, ce qui nous permet de rester dans le mode d'accueil et d’afficher le message “Your smart home”

Voici un schéma expliquant la relation entre les différents modules auxquels nous faisons appel :



Main

Ce fichier contient tous les includes et sert de “centre de contrôle”.
Le main contient le resetMain qui constitue le reset principal de notre programme. Il permet de configurer les ports, nettoyer le LCD et configurer les interruptions du timer. Il utilise également de la mémoire SRAM afin de stocker une valeur qui sera utilisée pour régler la température. En effet, la température étant une donnée sur 16 bits, nous avons trouvé plus judicieux de la stocker directement dans la SRAM dans une seule adresse au lieu de faire appel à deux registres qui doivent êtres conservées entre les fichiers (le set temperature utilise la température mesurée au préalable par le mode température).
Nous vérifions dans le main deux conditions : l'alarme et les autres boutons de la télécommande. Nous avons fait le choix de cette séparation car le capteur de distance ainsi que la télécommande sont composés chacun d’une boucle qui tourne jusqu’à ce que l'événement correspondant provoque un changement d’état. Nous avons exploré l’idée de faire une interruption avec la télécommande pour ce faire mais nous trouvons qu’un branchement est suffisant et est plus simple de réalisation.

Module mode_alarme

Lorsqu’il est appelé, ce module permet d’activer le mode alarme en affichant le message “ALARM ON” sur le LCD et en permettant au détecteur de distance de détecter un obstacle. Pendant la détection, les signaux analogiques sont convertis en signaux numériques ce qui permet de lire le voltage correspondant à la distance de l’obstacle. Ce voltage stocké dans les registre a0 et a1 sera ensuite converti en centimètres selon la courbe qui se trouve dans le data sheet.
Ainsi, nous avons choisi une distance pour laquelle une intrusion est dite détectée et qui enclenche l’alarme à travers le buzzer par un branchement conditionnel. Le buzzer est ici utilisé de la même manière que pour jouer de la musique et nous expliquerons cela plus en détail dans la partie correspondante.
Pendant que ce mode est activé, il n’est pas possible d’utiliser les autres boutons, l’utilisateur peut donc soit activer le mode alarme, soit décider de faire usage de sa smart

home et d'accéder à ses autres fonctionnalités, ce qui explique la séparation claire de ces deux cas de figure dans le main.

Module gestion_telecommande

Il s'agit du module qui permet d'utiliser la télécommande. Grâce au format RC5 pour les télécommandes à infrarouges, l'information est interceptée par le microcontrôleur et décodée par notre programme.
Ce module décode l'information puis stocke celle-ci dans les registres b0 et b1. Tout de même, la comparaison entre les registres b0 est suffisante car ils nous permettent déjà de différencier toutes les touches que nous avons choisi d'utiliser dans notre cas. Le bit de poids fort de notre signal reçu nous est donc inutile mais afin de ne pas perdre l'information, nous le stockons dans b1, comme le veut le code fourni par le manuel pour gérer les signaux RC5. Le registre b0 sera comparé aux valeurs connues (grâce à une mesure au préalable), qui sont déclarées comme constantes. Elles sont désignées par des étiquettes spécifiques et leur adresse pointe vers l'espace mémoire où seront stockées ces constantes dans la SRAM grâce à l'instruction .equ. Cette instruction nous permet également d'interdire la modification de ces constantes au cours de l'exécution du programme.
Après avoir effectué les comparaisons nécessaires, un branchement conditionnel est effectué permettant d'appeler le module correspondant, vérifier si aucun bouton n'est pressé ou si le mode alarme doit être déclenché.

Module mode_changement_temperature

Ce module permet à l'utilisateur de choisir la température avec la télécommande en incrémentant à l'aide du bouton 5 ou en décrémentant à l'aide du bouton 8. Les valeurs des températures sont stockées dans la SRAM lorsqu'elles sont incrémentées pour les mêmes raisons que dans le resetMain. Un affichage sur le LCD de ce réglage de la température est généré.
Le module stocke la température à modifier dans les registres a0 et a1. Il s'agira de la température actuelle si la mesure a déjà été faite ou une valeure par défaut (25.5) stockée dans la mémoire.
Le module est principalement constitué de branchements conditionnels. qui permettent d'augmenter, de diminuer la température ou de quitter le mode (bouton 4).
Pendant la vidéo, nous pouvons observer que la température passe de 31,43 degrés à 16,43 degrés. Il est à noter que nous avons fait le choix de pouvoir régler la température entre 16 et 31 degrés (sans prêter attention aux chiffres après la virgule). Ici, nous avons émis 2 signaux et donc augmenté la température à 31,43 degrés, c'est pour cette raison que celle-ci repasse à 16 degrés.

Module mode_température

Le module température utilise le 1-wire pour mesurer la température puis l'afficher. Au début, un message est affiché pour une durée de 2 secondes, avant que nous puissions voir la température mesurée par notre système.

Pour pouvoir sortir de ce mode sans l'intervention de l'utilisateur, nous avons décidé d'utiliser une interruption par timer.
En effet, lorsque le mode température est activé, nous lançons le timer0. Lorsqu'il atteindra 8 secondes, il exécutera la sous routine d'interruption après l'instruction courante. L'adresse d'interruption a été placée au début du programme du fichier main aux adresses données, grâce aux directives .org. L'adresse OVF0addr est quant à elle définie dans le fichier m128def.inc.
Ici, au lieu d'avoir un branchement vers une sous routine dans notre vecteur d'interruption, nous avons directement logé la sous routine à l'adresse de son vecteur d'interruption, ce qui ne perturbe pas le déroulement de la requête d'instruction. De plus, il est important de sauvegarder le contexte pour pouvoir reprendre l'exécution du main à la fin de l'exécution de la sous-routine d'interruption.
Nous affichons le texte qui indique que la fin du mode est en cours et nous stockons la valeur 1 dans le registre r16. Ainsi, lors du retour au main à l'adresse stockée dans la pile, le PC sautera vers l'étiquette end_mode_temperature. Dans les autres cas, notre registre r16 est a 0 et permet donc l'affichage de la température.
Notre timer fonctionne donc par principe d'overflow.

Module musique

Ce module nous permet de générer une musique avec le buzzer. Le fichier sound.asm est mis à notre disposition dans les librairies et c'est ce que nous utilisons pour jouer notre musique.
Nous avons établi une lookup table que nous parcourons avec le pointeur z et un registre utilisé comme compteur, ce qui permettra de notifier que l'entièreté des notes ont été jouées. Nous avons également ajouté un affichage "Discover :" avec le nom de la musique pour une expérience utilisateur complète. Le pointeur z nous permet de récupérer les notes à lire dans le registre r0, grâce à la commande lpm.

5. Description de détail de l'accès aux périphériques

Télécommande IR

Lorsque nous appuyant sur un bouton de la télécommande, des rayons infrarouges sont émis, il est donc normal que les pins qui reçoivent les données avant qu'elles soient converties soient configurés en entrées.

Détecteur de distance

Au vu du modèle que nous utilisons, le détecteur de distance doit être configuré en sortie. En effet, notre capteur est un capteur infrarouge, il va donc venir émettre des faisceaux lumineux infrarouges puis les récupérer par un récepteur infrarouge. En fonction de la quantité de lumière réfléchie, il nous sera possible de déterminer la distance qui sépare le système à l'objet. C'est pour cela que nous configurons ce type de capteur en sortie et non en entrée.

1-Wire

De même, le capteur de température va recevoir des informations du monde extérieur et les envoyer sous forme de signaux (électriques généralement pour avoir des pulses et des niveaux bas) au système qui va chercher à les convertir. De ce fait, il doit être configuré en entrée. Pour mesurer la température, nous utilisons le type de communication 1-Wire qui prescrit des timing stricts pour l'émission des signaux (niveaux 1 et niveaux 0). C'est un bus bidirectionnel développé pour établir une communication avec des périphériques à l'aide d'une seule ligne physique, dont le maître est le microcontrôleur.

Sorties : Buzzer et LCD

Le speaker et le lecteur LCD, quant à eux, vont recevoir des informations données dans le code afin que les utilisateurs puissent les percevoir ou les entendre. En effet, ce sont des interfaces qui permettent de communiquer efficacement avec leur environnement en transmettant des informations ou instructions sous forme de texte, de message, de son. Ils doivent donc être mis en sortie. Le LCD, afin de communiquer avec le microcontrôleur utilisé à la fois des bus d'adresses, de données et de configuration, car il dispose de plusieurs registres internes dont l'accès est donc faits grâce à des adresses et dont nous pouvons tirer certains types de données, notamment de la mémoire des caractères à afficher, nommées DDRAM, qui est parcourue par le registre AC. Nous devons à chaque fois le charger avec l'adresse du caractère à afficher et est toujours automatiquement après l'affichage de ce dernier. La connexion entre l'AVR et le LCD se fait tout de même par un circuit électronique pour recevoir des instructions précises tel que effacer l'écran (clear), remettre le curseur au début du LCD (home), ou autre.

6. Références

[1] listes des groupes de TPs: site Moodle, Mars 2023.

[2] https://www.epfl.ch/campus/security-safety/en/health/coronavirus-covid19/, December 5, 2022, et suivant.

[3] Vivanco, Universal TV- DVB Controller UR Z2, BDA 34873-Rev-RZ, 2013.

[4] Vishay Semiconductors, Data Formats for IR Remote Control, Document Number: 80071, Rev. 2.2, 2019.

[5] Vishay Semiconductors, IR Receiver Modules for Remote Control Systems, TSOP22.., TSOP24.., TSOP48.., TSOP44.., Document Number: 82459, 2016.

7. Annexes