

MINI-PROJET : HORLOGE NUMÉRIQUE ET RÉVEIL.

Projet réalisé par :

DHISSI AYMAN
MOUTAWADI ALI

Projet encadré par :

Mme Chakir

REMERCIEMENTS

Cher Mme Chakir,

Nous tenons à vous exprimer nos sincères remerciements pour votre soutien et votre guidance tout au long de notre mini projet sur l'horloge. Votre expertise et vos précieux conseils ont grandement contribué à la réussite de notre projet.

Nous tenons également à souligner l'importance de votre rôle dans notre apprentissage et développement de compétences. Votre volonté de partager vos connaissances et votre disponibilité pour répondre à nos questions ont été extrêmement appréciées.

Ce projet nous a permis d'explorer les aspects pratiques de la programmation Arduino, la manipulation des composants électroniques et la conception de systèmes embarqués. Grâce à votre encadrement, nous avons pu acquérir une compréhension approfondie des principes de base de l'horlogerie numérique et développer nos compétences techniques.

Nous tenons également à exprimer notre gratitude pour avoir créé un environnement d'apprentissage stimulant et encourageant. Votre encouragement constant et votre confiance en nos capacités ont été des moteurs pour notre motivation et notre engagement envers le projet.

Enfin, nous vous remercions de nous avoir donné l'opportunité de travailler sur ce projet. Il nous a permis d'appliquer les connaissances théoriques acquises en classe et de développer notre créativité et notre esprit d'innovation.

Nous vous sommes extrêmement reconnaissants pour votre soutien et vos conseils tout au long de ce mini projet. Votre engagement envers notre réussite a été inestimable, et nous espérons pouvoir bénéficier de votre expertise à l'avenir.

Sincèrement :
DHISSI AYMAN,
MOUTAWADI ALI
Année 2023-2024

SOMMAIRE

1. Introduction

- 1.1 Contexte
- 1.2 Objectif du projet
- 1.3 Méthodologie

2. Architecture de l'horloge avec réveil

- 2.1 Langage de programmation et outils utilisés
- 2.2 Structure
- 2.3 Principales fonctionnalités

3. Conception

- 3.1 Explications du code

4. Résultats et discussion

- 4.1 Limitations et Améliorations potentielles

5. Conclusion

1. Introduction :

Le projet consiste en la création d'une horloge avec affichage sur un écran LCD et une fonctionnalité de réveil. L'objectif principal est de concevoir et développer un système qui affiche l'heure et la date actuelles, tout en permettant à l'utilisateur de définir une heure de réveil pour déclencher une sonnerie.

L'horloge sera basée sur la plateforme Arduino, qui est largement utilisée pour le développement de projets électroniques. Elle sera équipée d'un écran LCD 16x2 pour afficher l'heure, la date et d'autres informations, ainsi que d'un buzzer qui servira de sonnerie pour le réveil.

Le projet nécessitera une programmation en langage Arduino pour gérer l'affichage de l'heure et de la date, la configuration du réveil et le déclenchement de la sonnerie. Des bibliothèques telles que LiquidCrystal.h seront utilisées pour faciliter l'interaction avec l'écran LCD.

L'horloge offrira également une fonctionnalité de configuration initiale permettant à l'utilisateur de définir l'heure et la date au démarrage. Cela garantira que l'horloge est synchronisée avec l'heure réelle.

Dans ce rapport de projet, nous présenterons en détail les différentes étapes de conception, de développement et de mise en œuvre de l'horloge. Nous discuterons également des fonctionnalités principales, des limitations potentielles et des améliorations envisageables. Enfin, nous conclurons sur les résultats obtenus et les apprentissages réalisés au cours de ce projet.

1.1 Contexte :

L'utilisation des horloges est omniprésente dans notre vie quotidienne, que ce soit pour se réveiller le matin, planifier des rendez-vous ou simplement vérifier l'heure. Avec les avancées technologiques, les horloges traditionnelles ont évolué vers des horloges numériques plus précises et polyvalentes.

1.2 Objectif du projet :

- Affichage de l'heure : L'horloge doit être capable d'afficher avec précision l'heure actuelle sur l'écran LCD. Cela permet à l'utilisateur de connaître l'heure instantanée en un coup d'œil.
- Fonction de réveil : L'horloge doit permettre à l'utilisateur de spécifier une heure de réveil. À cette heure spécifiée, l'horloge doit déclencher une sonnerie ou une alarme pour signaler le réveil. Cela aide les utilisateurs à se réveiller à l'heure souhaitée.
- Affichage de la date : En plus de l'heure, l'horloge doit également être capable d'afficher la date actuelle sur l'écran LCD. Cela permet à l'utilisateur de connaître la date en cours.
- Configuration initiale : L'utilisateur doit pouvoir configurer la date et l'heure initiales au démarrage de l'horloge. Cela garantit que l'horloge est correctement calibrée et synchronisée avec la date et l'heure réelles.
- Précision et fiabilité : L'horloge doit être précise et fiable pour fournir des informations temporelles précises. Cela peut être réalisé en utilisant des mécanismes de suivi du temps appropriés, tels que des compteurs de temps ou des horloges en temps réel.
- Interface utilisateur conviviale : L'horloge doit offrir une interface utilisateur conviviale pour la configuration et l'utilisation. Cela peut être réalisé en utilisant des boutons, un clavier ou d'autres méthodes d'interaction pour permettre à l'utilisateur de définir l'heure de réveil et de configurer d'autres paramètres.
- En atteignant ces objectifs, le projet aboutira à la création d'une horloge fonctionnelle et pratique, offrant à l'utilisateur la possibilité de connaître l'heure, la date et de se réveiller à l'heure souhaitée grâce à une fonctionnalité de réveil intégrée.

1.3 Méthodologie :

Étape 1 : Matériel requis

- Arduino Uno ou une autre carte compatible
- Écran LCD 16x2
- Buzzer actif ou passif
- Potentiomètre (pour ajuster le contraste de l'écran LCD)
- Fils de raccordement

Étape 2 : Câblage

- Connectez les broches VCC et GND de l'écran LCD à l'alimentation 5V et à la masse de l'Arduino respectivement.
- Connectez les broches RS, E, D4, D5, D6 et D7 de l'écran LCD aux broches numériques de l'Arduino.
- Connectez la broche positive du buzzer à une broche numérique de l'Arduino.
- Connectez la broche négative du buzzer à la masse de l'Arduino.

Étape 3 : Configuration du code

- Incluez la bibliothèque LiquidCrystal.h pour gérer l'écran LCD.
- Définissez les broches utilisées pour l'écran LCD et le buzzer en tant que variables globales.
- Déclarez les variables pour stocker l'heure, les minutes, le buzzer, etc.
- Initialiser l'écran LCD dans la fonction setup().
- Écrivez les fonctions pour afficher l'heure sur l'écran LCD et activer/désactiver le buzzer.

Étape 4 : Obtention de l'heure

- Utilisez la fonction millis() pour obtenir le temps écoulé depuis le démarrage de l'Arduino.
- Calculez l'heure, les minutes et les secondes à partir du temps écoulé.

Étape 5 : Affichage de l'heure sur l'écran LCD

- Utilisez la fonction lcd.setCursor() pour définir la position du curseur sur l'écran LCD.
- Utilisez la fonction lcd.print() pour afficher l'heure, les minutes et les secondes sur l'écran LCD.

Étape 6 : Activation du buzzer pour le réveil

- Comparez l'heure actuelle avec l'heure définie pour le réveil.
- Si les heures et les minutes correspondent, activez le buzzer en utilisant la fonction tone().
- Attendez un certain temps pour que le buzzer sonne pendant une durée spécifiée.
- Utilisez la fonction noTone() pour désactiver le buzzer.

Étape 7 : Boucle principale

- Dans la fonction loop(), appelez les fonctions nécessaires pour obtenir l'heure, afficher l'heure sur l'écran LCD et vérifier l'activation du réveil.

- Ajoutez des délais appropriés pour éviter une utilisation excessive du processeur.

Étape 8 : Test et ajustement

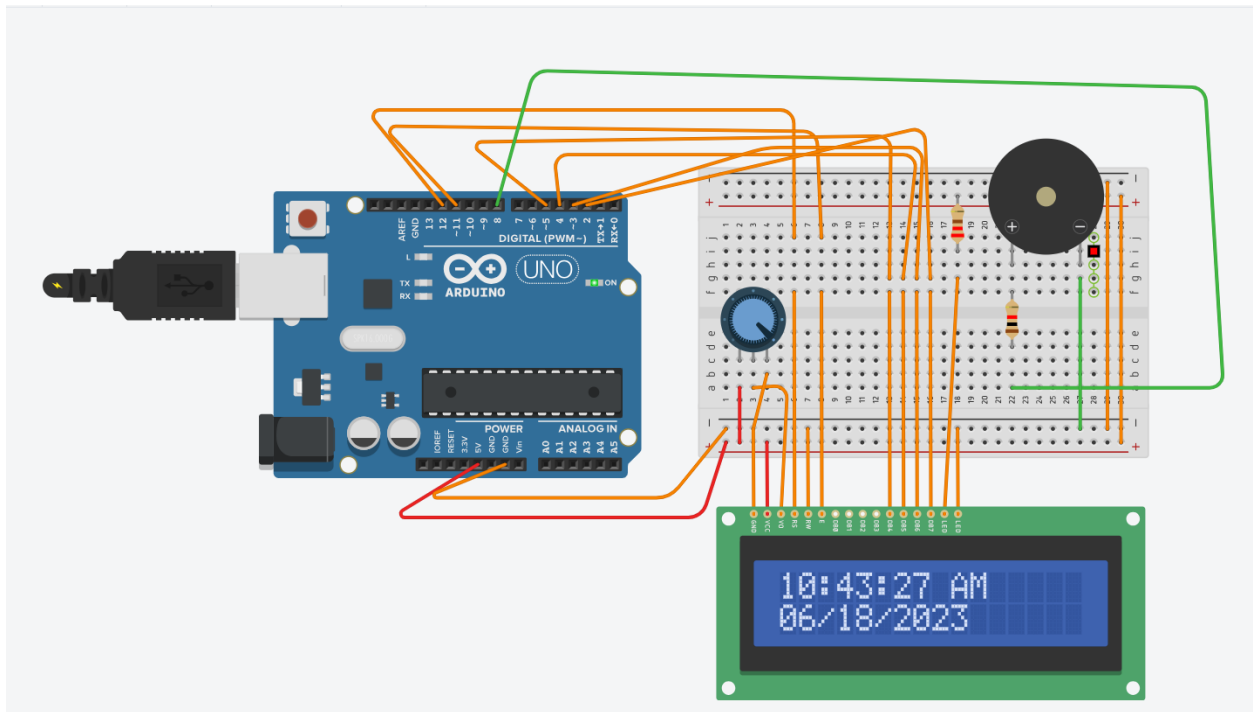
- Téléversez le code sur l'Arduino et observez le fonctionnement de l'horloge.
- Utilisez le potentiomètre pour ajuster le contraste de l'écran LCD si nécessaire.
- Testez le réveil en définissant une heure de réveil et en vérifiant si le buzzer s'active correctement.

2. Architecture de l'horloge avec réveil

2.1 Langage de programmation et outils utilisés

- Le langage de programmation utilisé dans ce code est le C++. L'outil utilisé est l'environnement de développement intégré (IDE) Arduino, qui est l'IDE standard pour développer des projets Arduino. Cet IDE offre une interface conviviale pour écrire, compiler et téléverser le code sur la carte Arduino. Il fournit également une bibliothèque appelée "LiquidCrystal" qui facilite la communication avec l'écran LCD 16x2.
- La bibliothèque "LiquidCrystal" permet de contrôler l'écran LCD en utilisant les fonctions fournies, telles que `begin()`, `setCursor()`, `print()`, etc. De plus, la fonction `tone()` est utilisée pour générer un signal sonore à l'aide d'un buzzer, et `noTone()` pour arrêter la génération du son.
- En résumé, le langage de programmation est le C++ et l'IDE Arduino est utilisé en conjonction avec la bibliothèque "LiquidCrystal" pour développer et téléverser le code sur la carte Arduino.

2.2 Structure :



2.3 Principales fonctionnalités :

- La fonction principale de ce projet est de créer une horloge avec un écran LCD et un buzzer qui sert de réveil. Voici les fonctionnalités principales :
- Affichage de l'heure : L'horloge affiche en temps réel l'heure, les minutes et les secondes sur l'écran LCD. L'heure est obtenue à partir des détails fournis par l'utilisateur ou à partir de l'heure actuelle de l'Arduino s'il n'y a pas de détails spécifiés.
- Affichage du mode AM/PM : L'horloge indique également le mode AM (matin) ou PM (après-midi) en fonction des détails fournis par l'utilisateur.
- Affichage de la date : L'écran LCD affiche également la date au format mois/jour/année.
- Réglage du réveil : L'utilisateur peut spécifier l'heure à laquelle il souhaite être réveillé en entrant les détails de l'heure du réveil via la communication série.
- Sonnerie du réveil : Lorsque l'heure actuelle correspond à l'heure du réveil spécifiée par l'utilisateur, un signal sonore est émis à l'aide du buzzer.
- Arrêt de la sonnerie : L'utilisateur peut arrêter la sonnerie en entrant le chiffre 1 via la communication série.
- Configuration initiale : Lors de la première exécution du code, l'utilisateur est invité à entrer les détails de l'heure et de la date pour initialiser l'horloge.

3. Conception :

3.1 Explication du code :

- `#include <LiquidCrystal.h>` : Cette ligne inclut la bibliothèque LiquidCrystal nécessaire pour utiliser un écran LCD.
- `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);` : Cela crée un objet lcd de la classe LiquidCrystal qui est utilisé pour communiquer avec l'écran LCD. Les nombres entre parenthèses représentent les broches Arduino connectées aux broches de contrôle de l'écran LCD.
- 5-12. Les variables ss, mm, hh, MM, DD, YYYY, AP, hr, k et a sont déclarées et initialisées.
- 14-18. La fonction `setup()` est une fonction spéciale dans Arduino qui est exécutée une fois au démarrage du programme. Ici, elle initialise la communication série avec une vitesse de 9600 bauds et initialise l'écran LCD avec une configuration de 16 colonnes et 2 lignes.
- 20-69. La fonction `loop()` est une autre fonction spéciale dans Arduino qui est exécutée en boucle indéfiniment. Le code à l'intérieur de cette fonction contrôle le déroulement du programme.
- Le programme commence par vérifier la valeur de la variable k. Si k est inférieur ou égal à 0, la fonction `Details()` est appelée pour demander à l'utilisateur d'entrer l'heure et la date via la communication série (port série). Les valeurs entrées sont stockées dans les variables correspondantes.
- Ensuite, une seconde (ss) est ajoutée. Si ss atteint 60, cela signifie qu'une minute est écoulée. Les minutes (mm) sont incrémentées et si mm atteint 60, cela signifie qu'une heure est écoulée. Les heures (hh) sont incrémentées et si hh atteint 12, cela signifie que le changement d'AM à PM ou vice-versa se produit. La fonction `printTime2()` est appelée pour afficher l'heure actuelle sur l'écran LCD.
- Si AP est pair (égal à 0), cela signifie que nous sommes dans la période AM et la variable DD (jour) est incrémentée.
- Si DD atteint 31, cela signifie que le mois (MM) doit être incrémenté. Si MM atteint 13, cela signifie que l'année (YYYY) doit être incrémentée.
- Si hh atteint 13, cela signifie que minuit est passé et hh est réinitialisé à 1.

- Ensuite, le programme entre dans une boucle while qui attend que a soit égal à 1. Si la valeur entrée pour hr est égale à l'heure actuelle (hh), une tonalité est générée pendant 5 secondes. Le programme demande ensuite à l'utilisateur d'entrer 1 pour arrêter la sonnerie. Si a est égal à 1, la sonnerie est arrêtée.
- Enfin, la fonction printTime2() est appelée à nouveau pour afficher l'heure mise à jour sur l'écran LCD, et la boucle loop() recommence.
- Les autres fonctions (line(), Details(), printTime()) sont des fonctions auxiliaires utilisées pour effectuer des tâches spécifiques dans le programme, comme l'affichage de lignes vides dans le moniteur série ou l'affichage du temps sur l'écran LCD avec un format spécifique.

4. Résultats et discussion :

4.1 Limitations et Améliorations potentielles :

Dépendance à la communication série :

- Limitation : Le projet nécessite une connexion à un terminal série pour recevoir les détails de l'heure, de la date et de l'heure du réveil.
- Amélioration potentielle : Ajouter des boutons ou un clavier à l'Arduino pour permettre à l'utilisateur de saisir directement les détails sans avoir besoin d'une connexion à un terminal série. Cela rendrait le projet plus autonome et plus convivial.

Absence de sauvegarde des données :

- Limitation : Les détails de l'heure, de la date et de l'heure du réveil doivent être entrés à chaque redémarrage du projet.
- Amélioration potentielle : Intégrer une mémoire non volatile, telle qu'une mémoire EEPROM, pour stocker les détails de manière permanente. Ainsi, les détails seraient rappelés automatiquement au redémarrage de l'Arduino, évitant ainsi la nécessité de les saisir à chaque fois.

Limitation de la gestion des mois et des jours :

- Limitation : Le projet suppose que tous les mois ont 28 jours et qu'il y a une avance d'un jour lorsque l'on passe de l'après-midi (PM) au matin (AM).
- Amélioration potentielle : Améliorer la gestion des mois en tenant compte de leurs différentes longueurs et ajuster la date en conséquence. De cette manière, l'affichage de la date sera plus précis et correspondra à la réalité.

Sonnerie limitée :

- Limitation : Le projet utilise uniquement un signal sonore basique en utilisant un buzzer.
- Amélioration potentielle : Ajouter des fonctionnalités avancées à la sonnerie, telles que la possibilité de sélectionner différents sons ou de régler le volume. Il serait

également possible d'intégrer une bibliothèque de fichiers audio pour jouer des sons personnalisés, offrant ainsi une plus grande variété et une meilleure expérience sonore.

Interface utilisateur limitée :

- Limitation : L'interface utilisateur actuelle se limite à la communication série, ce qui peut être peu convivial pour les utilisateurs non techniques.
- Amélioration potentielle : Intégrer un écran tactile ou des boutons à l'Arduino pour permettre à l'utilisateur de configurer l'horloge et le réveil directement sur l'appareil. Cela offrirait une meilleure expérience utilisateur, rendant le projet plus convivial et intuitif à utiliser.

5. CONCLUSION :

En conclusion, le mini-projet d'horloge avec écran LCD et buzzer de réveil offre une fonctionnalité de base pour afficher l'heure, la date et activer une sonnerie à une heure prédéfinie. Cependant, quelques limitations ont été identifiées, notamment la dépendance à la communication série, l'absence de sauvegarde des données, la gestion limitée des mois et des jours, la sonnerie basique et l'interface utilisateur limitée.

Pour améliorer ce projet, plusieurs pistes ont été suggérées, telles que l'ajout de boutons ou d'un clavier pour faciliter la saisie des détails, l'intégration d'une mémoire non volatile pour la sauvegarde permanente des données, l'amélioration de la gestion des mois et des jours pour une plus grande précision, l'ajout de fonctionnalités avancées à la sonnerie et l'amélioration de l'interface utilisateur avec un écran tactile ou des boutons.

En mettant en œuvre ces améliorations, le projet pourrait devenir plus autonome, offrir une meilleure expérience utilisateur, une gestion plus précise de l'heure et de la date, ainsi qu'une sonnerie plus personnalisable. Ces améliorations contribuent à rendre l'horloge plus conviviale, plus pratique et plus polyvalente.

En conclusion, ce mini-projet d'horloge constitue une base solide pour une application d'affichage de l'heure avec fonction de réveil. Cependant, il offre des possibilités d'amélioration intéressantes pour répondre aux besoins spécifiques des utilisateurs et offrir une expérience plus complète.