**UNIVERSITE D’ANTANANARIVO**

**ECOLE SUPERIEURE POLYTECHNIQUE D’ANTANANARIVO**

**MENTION ELECTRONIQUE**

**PROJET PIC DU COURS DE CIRCUITS PROGRAMMABLES**

**Domaine :** Sciences de l’Ingénieur

**Mention :** Electronique

**Parcours :** Electronique Automatiques

Interfacing RTC Module (DS3231) with PIC Microcontroller (PIC16F877A) Digital Clock

**Rédigé par :**

RANDRIANJATOVO Vahiniaina et **RAHARIJAONA Valicio**

**Année universitaire :** 2023-2024

# RESUME

# SOMMAIRE

# LISTE DES ABREVIATIONS ET SYMBOLES

# LISTE DES FIGURES

# 1. Introduction

Dans ce projet, on vise à construire une horloge digitale utilisant un circuit intégré RTC et un microcontrôleur PIC. Le RTC permet de maintenir une heure précise même en cas de coupure de courant, tandis que le microcontrôleur est responsable de l'affichage de l'heure et de la gestion des interactions utilisateur. On a choisi le microcontrôleur PIC16F877A et le RTC DS3123.

# 2. Matériel nécessaire

Dressez la liste complète des composants matériels utilisés :

## Module RTC DS3231

Le module RTC (Real-Time Clock) DS3231 permet à votre microcontrôleur PIC16F877A de garder une trace précise de l'heure, même en cas de coupure de courant grâce à sa pile de sauvegarde. La connexion du module RTC DS3231 avec notre PIC16F877A se fait :

**Brochage :**

* **Alimentation :**
  + VCC (broche 1 du DS3231) à +5V du PIC (Vdd).
  + GND (broche 3 du DS3231) à la masse du PIC (Vss).
* **Communication I2C :**
  + SDA (broche 5 du DS3231) à la broche SDA du PIC (broche 23, RC4/SDI/SDA).
  + SCL (broche 6 du DS3231) à la broche SCL du PIC (broche 18, RC3/SCK/SCL).

Le DS3231 communique via le bus I2C (Inter-Integrated Circuit), nécessitant deux lignes de signalisation :

* **SDA (Serial Data Line) :** ligne de données bidirectionnelle pour transférer les informations.
* **SCL (Serial Clock Line) :** signal d'horloge synchronisant la communication.

Le PIC devra initier la communication I2C en tant que maître pour envoyer et recevoir des données du module RTC esclave (DS3231).

## Microcontrôleur PIC16F877A

Configuration du PIC16F877A :

**Alimentation :**

* **Vdd (broche 20) :** Alimentation positive (+5V).
* **Vss (broche 18) :** Masse (0V).
* **MCLR (broche 1) :** Reset (à maintenir à Vdd pour un fonctionnement normal) connecte à +5V en passant par une résistance de 10K.

**Oscillateur à quartz :**

* **OSC1 (broche 17) :** Entrée d'horloge (connecter à une broche du quartz).
* **OSC2 (broche 16) :** Sortie d'horloge (connecter à l'autre broche du quartz).

**Programmation :**

* **MCLR (broche 1) :** connecte directement a Vdd pour un fonctionnement de programmation (connecte à +5V).
* **PGD (broche 40) :** Programmation données (à connecter au programmateur).
* **PGC (broche 39) :** Programmation horloge (à connecter au programmateur).

**Broches d'entrée/sortie (E/S) :**

**Port RD (broches 21 à 30):** 4 broches E/S configurables.

L’oscillateur a quartz doit être connectés aux broches OSC1 et OSC2 pour générer l'horloge du microcontrôleur.

La configuration des broches E/S (direction, mode) se fait par programmation.

## Afficheur LCD 16x2

Pour afficher du texte sur un écran LCD 16x2 avec le microcontrôleur PIC16F877A, il faut connecter les broches du LCD aux ports du PIC et configurer le contrôleur LCD via le firmware. La connexion avec le PIC se fait comme suit :

* **Broches d’alimentation :**

VDD (broche 1) du LCD à Vdd (+5V) du PIC.

Vss (broche 2) du LCD à Vss (masse) du PIC.

* **Broches de contraste :**

VEE (broche 3) du LCD à un potentiomètre 10K connecté à Vdd et Vss pour ajuster le contraste.

R/W (broche 5) du LCD à Vss (masse).

* **Broches de données :**

D4 à D7 du LCD aux ports RD4 à RD7(broches 27 à 30) du PIC.

* **Broches de contrôle :**

RS (broche 4) à la brocher RD2 du PIC

E (broche 6) à la broche RD3 du PIC

## Oscillateur a quartz

L'oscillateur à quartz génère un train continu d'impulsions carrées. La fréquence fondamentale de ce signal est contrôlée par le cristal de quartz lui-même. Cet oscillateur nécessite 2 condensateurs 22pF connecté aux masses et un cristal quartz 20MHz dont les bornes sont connectées a chacun des condensateurs. Les 2 bornes du cristal est aussi l’entre et sortie vers le PIC16F877A.

## PICKIT 3.5 Programmer

Le Pickit 3.5 Programmer est le **programmateur/ débogueur matériel** utilisé pour transférer le code compilé vers le microcontrôleur PIC. Il agit comme un intermédiaire physique entre l’ordinateur et le microcontrôleur PIC.

### Fonctionnalités principales du Pickit 3 Programmer :

* **Programmation en circuit (ICSP - In-Circuit Serial Programming) :** Permet de programmer le microcontrôleur PIC directement sur la carte de circuit sans avoir besoin de le retirer. Cela simplifie le processus de développement et de test.
* **Débogage :** Cela permet d'analyser le code en cours d'exécution sur le microcontrôleur, d'identifier les erreurs et de les corriger plus facilement.
* **Facilité d’utilisation :** Le Pickit 3 Programmer est relativement facile à utiliser, même pour les débutants. Il se connecte à votre ordinateur via un port USB et communique avec le logiciel de développement, tel que MPLAB X IDE.

### Connexion :

Connecter le Pickit 3 Programmer à votre ordinateur via USB et à la carte de circuit imprimé contenant le microcontrôleur PIC avec les connexion correspondant

* MCLR/Vpp

Activation de la programmation haute tension (Vpp) : Dans certains cas, les anciens microcontrôleurs PIC peuvent nécessiter une tension plus élevée (généralement 12V) sur cette broche pour activer le mode de programmation. Les PIC modernes n'en ont généralement pas besoin et obtiennent leur tension de programmation du Pickit 3.5 lui-même. Pendant la programmation, une brève impulsion basse sur cette broche peut être utilisée pour réinitialiser le microcontrôleur.

* Vdd

Cette broche alimente le programmateur Pickit 3.5 en tension (dans notre cas +5V).

* GND

Cette broche est la connexion à la masse du programmateur Pickit 3.

* PGD

Program Data (Données de programmation). C'est une ligne de données bidirectionnelle utilisée pour la communication entre le programmateur Pickit 3.5 et le microcontrôleur pendant la programmation.

* PGC

Program Clock (Horloge de programmation). C'est un signal d'horloge utilisé par le programmateur Pickit 3.5 pour synchroniser le transfert de données avec le microcontrôleur pendant la programmation.

Il y a une derniere broche non utilisée, cette broche n'est actuellement pas utilisée et doit rester non connectée.

### Programmation :

Utilisez le compilateur Xc8 pour générer un fichier .hex à partir du code source, puis dans MPLAB IPE ou Pickit3 programmer Application sélectionner le fichier de code compilé et lancer le processus de programmation. Le Pickit 3 Programmer transfère alors le code vers la mémoire du microcontrôleur PIC.

//Ajoutez des schémas de câblage ou des photos du montage pour une meilleure illustration.

# 3. Logiciel de développement

## Realisation

On a expérimenté avec plusieurs logiciels de développement utilisé pour programmer le microcontrôleur PIC.

De base on a utilisé le compilateur Xc8 avec le MPLABX IDE.

Pour programmer le PIC, on a utilisé MPLABX IPE et Pickit 3 Programmer pour transverser le code.

* MPLAB X IDE :
  + Éditeur de code source avec coloration syntaxique et complétion automatique.
  + Compilateur C et assembleur.
  + Débogueur intégré avec affichage de la mémoire et des registres.
  + Simulateur pour tester le code avant de le programmer sur le microcontrôleur.
  + Prise en charge de plusieurs programmateurs, dont le Pickit 3.
* MPLAB IPE :
  + **Interface simple et intuitive**
  + Programmation en production
  + **Deux modes de fonctionnement** 
    1. **Mode Production :** Limité aux actions autorisées par l'ingénieur en conception, ce mode permet aux **techniciens de production de programmer des appareils sans risque de modifier accidentellement la configuration.**
    2. **Mode Avancé : Destiné aux ingénieurs, ce mode offre un accès complet à toutes les fonctionnalités de programmation, y compris la configuration des paramètres d'alimentation, la sélection des**
* **PICkit3 Programmer :**
  + **Programmation**
  + Vérification
  + **Effacement**
  + Débogage (sur certaines versions)

## **Simulation**

Proteus **est un logiciel de simulation de circuits électroniques développé par Labcenter Electronics. Il permet de créer des schémas électroniques, de les simuler et de les déboguer. Proteus est un outil puissant et polyvalent qui peut être utilisé pour une large gamme d'applications, de l'éducation à la recherche en passant par la conception de circuits électroniques professionnels.**

**Pour programmer le module Pic dans la simulation, on a procédé a deux méthodes différentes :**

* **Téléversement du code hexadécimal du programme après compilation avec Xc8 dans MPLAB X IDE**
* **Edition direct du code source du programme du PIC dans Proteus avec l’utilisation du compilateur Xc8**

// Mentionnez si vous avez utilisé des bibliothèques spécifiques pour interagir avec le RTC ou l'afficheur.

# 4. Conception du circuit

Détaillez le schéma électrique du circuit.

Expliquez les connexions entre le RTC, le microcontrôleur et l'afficheur.

Vous pouvez inclure des captures d'écran du schéma réalisé sur un logiciel de CAO électronique (ex: KiCad).

# 5. Développement du logiciel

Décrivez l'algorithme principal utilisé pour lire l'heure et la date à partir du RTC et les afficher sur l'écran.

Expliquez les fonctions principales du code source, en mettant l'accent sur la communication I2C avec le RTC et le pilotage de l'afficheur.

Fournissez des extraits de code source commentés pour illustrer les points clés.

# 6. Test et résultats

Décrivez la procédure de test mise en place pour vérifier le fonctionnement de l'horloge digitale.

Présentez les résultats obtenus : l'horloge affiche-t-elle correctement l'heure et la date ?

Discutez des éventuels problèmes rencontrés et comment vous les avez résolus.

# 7. Conclusion

Résumez les points clés du projet.

Mentionnez les points forts et les points faibles de la réalisation.

Suggérez des améliorations possibles pour des projets futurs.

# 8. Annexes

Ajoutez en annexe le code source complet du projet.

Incluez la fiche technique du module RTC DS3231 et du microcontrôleur PIC16F877A (si pertinent).

Ajoutez tout autre document utile à la compréhension du projet.

Conseils supplémentaires

Utilisez des illustrations claires et des captures d'écran pour améliorer la compréhension du rapport.

Soignez la mise en page et la présentation du rapport.

Rédigez dans un style clair, concis et professionnel.

En suivant ce plan de rédaction, vous disposerez d'un rapport de réalisation complet et documenté de votre projet d'horloge digitale.

jjjj