

Système de Détection de Tremblements de Terre avec Arduino

Module : M.5.3.2 - Développement des applications embarquées

Filière : Ingénierie de la Sécurité des Systèmes d'Information et Cyberdéfense

Membres du groupe :

Sami BOUCHNAFA

Mohammed MOUADDEN

Date de remise : 27/10/2024

Table des matières

1	Introduction	2
1.1	Présentation de l'idée	2
1.2	Contextualisation	2
1.3	Problématique	2
2	Cahier des charges détaillé	2
2.1	Spécifications fonctionnelles et techniques	2
2.2	Description de la carte retenue	3
2.2.1	Caractéristiques techniques adaptées	3
2.2.2	Avantages pour le projet	3
2.3	Description des capteurs/actionneurs	3
3	Conception du système embarqué	4
3.1	Diagramme d'états-transitions	4
3.2	Explications des états	5
4	Simulation et implémentation	6
4.1	Capture d'écran de la simulation dans le cas sans alerte	6
4.2	Capture d'écran de la simulation dans le cas avec alerte	7
4.3	Lien vers le projet	7
4.4	Code source	7
5	Conclusion	8
5.1	Synthèse du projet	8
5.2	Innovation	8
5.3	Difficultés et challenges	8
6	Références	8

1. Introduction

1.1. Présentation de l'idée

Notre projet consiste en un système de détection de tremblements de terre low-cost basé sur Arduino. Ce système utilise un accéléromètre MPU6050 pour mesurer en temps réel les vibrations et mouvements sismiques, couplé à un système d'alerte sonore via un buzzer pour avertir les occupants d'un bâtiment.

1.2. Contextualisation

Dans de nombreuses régions sismiques, l'accès à des systèmes d'alerte précoce reste limité en raison des coûts élevés des équipements professionnels. Ce projet s'inscrit dans une démarche de démocratisation des systèmes de sécurité sismique, en proposant une solution abordable et facilement déployable pour les particuliers et petites structures.

1.3. Problématique

Comment concevoir un système de détection sismique fiable, économique et accessible, capable d'alerter rapidement les occupants d'un bâtiment en cas de secousses significatives ?

2. Cahier des charges détaillé

2.1. Spécifications fonctionnelles et techniques

1. Fonctionnalités principales :

- Détection continue des mouvements sismiques sur 3 axes
- Calcul en temps réel de l'intensité des secousses
- Système d'alerte sonore en cas de dépassement de seuil
- Monitoring des données via interface série
- Système d'arrêt de sécurité automatique
- Réinitialisation manuelle post-alerte

2. Contraintes techniques :

- Temps de réponse < 500ms
- Précision de mesure : $\pm 2g$
- Fréquence d'échantillonnage : 10Hz
- Consommation énergétique < 100mA
- Autonomie sur batterie > 24h

2.2. Description de la carte retenue

La carte Arduino UNO R3 a été choisie pour les raisons suivantes :

2.2.1 Caractéristiques techniques adaptées

- Microcontrôleur ATmega328P à 16MHz
- 32KB de mémoire flash
- 2KB de RAM
- Bus I2C pour la communication avec le MPU6050
- Alimentation 5V stable

2.2.2 Avantages pour le projet

- Coût abordable (<30€)
- Large communauté et documentation extensive
- Compatible avec de nombreux capteurs
- Environnement de développement simple
- Possibilité d'alimentation par batterie

2.3. Description des capteurs/actionneurs

1. Capteur MPU6050 :

- *Caractéristiques :*
 - Accéléromètre 3 axes avec gyroscope intégré
 - Plage de mesure : $\pm 2g$ à $\pm 16g$
 - Résolution : 16 bits
 - Interface I2C
 - Consommation : 3.6mA
- *Justification :*
 - Haute précision
 - Prix abordable
 - Communication numérique stable
 - Filtrage intégré du bruit

2. Buzzer :

- *Caractéristiques :*

- Tension : 5V
- Fréquence : 2000-4000Hz
- Volume : >85dB
- *Justification* :
 - Volume suffisant pour alerter
 - Consommation faible
 - Facilité d'intégration

3. Bouton de Reset :

- *Caractéristiques* :
 - Type : Bouton poussoir
 - Configuration : Pull-up interne
 - Connecté à la pin 2 (Digital)
- *Justification* :
 - Réinitialisation manuelle sécurisée
 - Prévention des redémarrages automatiques
 - Interface utilisateur simple

3. Conception du système embarqué

3.1. Diagramme d'états-transitions

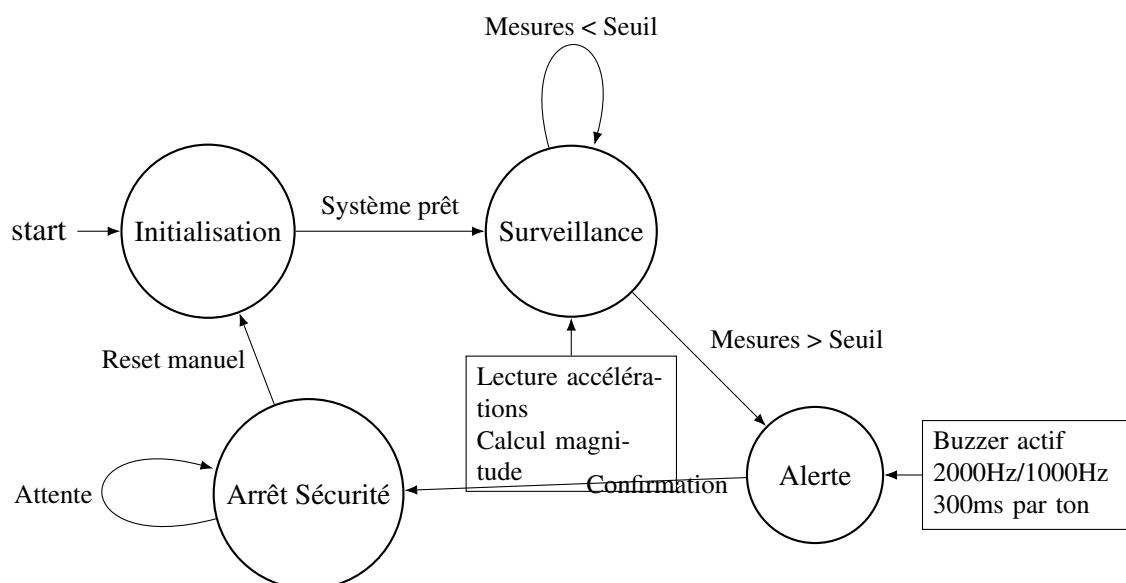


FIGURE 1 – Diagramme d'états-transitions du système

3.2. Explications des états

1. Initialisation :

- Configuration des composants
- Calibration des capteurs
- Vérification du système

2. Surveillance :

- Lecture continue des accélérations
- Calcul de la magnitude totale
- Comparaison avec le seuil

3. Alerte :

- Activation du buzzer avec pattern sonore alterné
- Envoi des données
- Préparation à l'arrêt de sécurité

4. Arrêt Sécurité :

- Système en attente
- Maintien de l'alarme sonore
- Attente d'intervention manuelle

4. Simulation et implémentation

4.1. Capture d'écran de la simulation dans le cas sans alerte

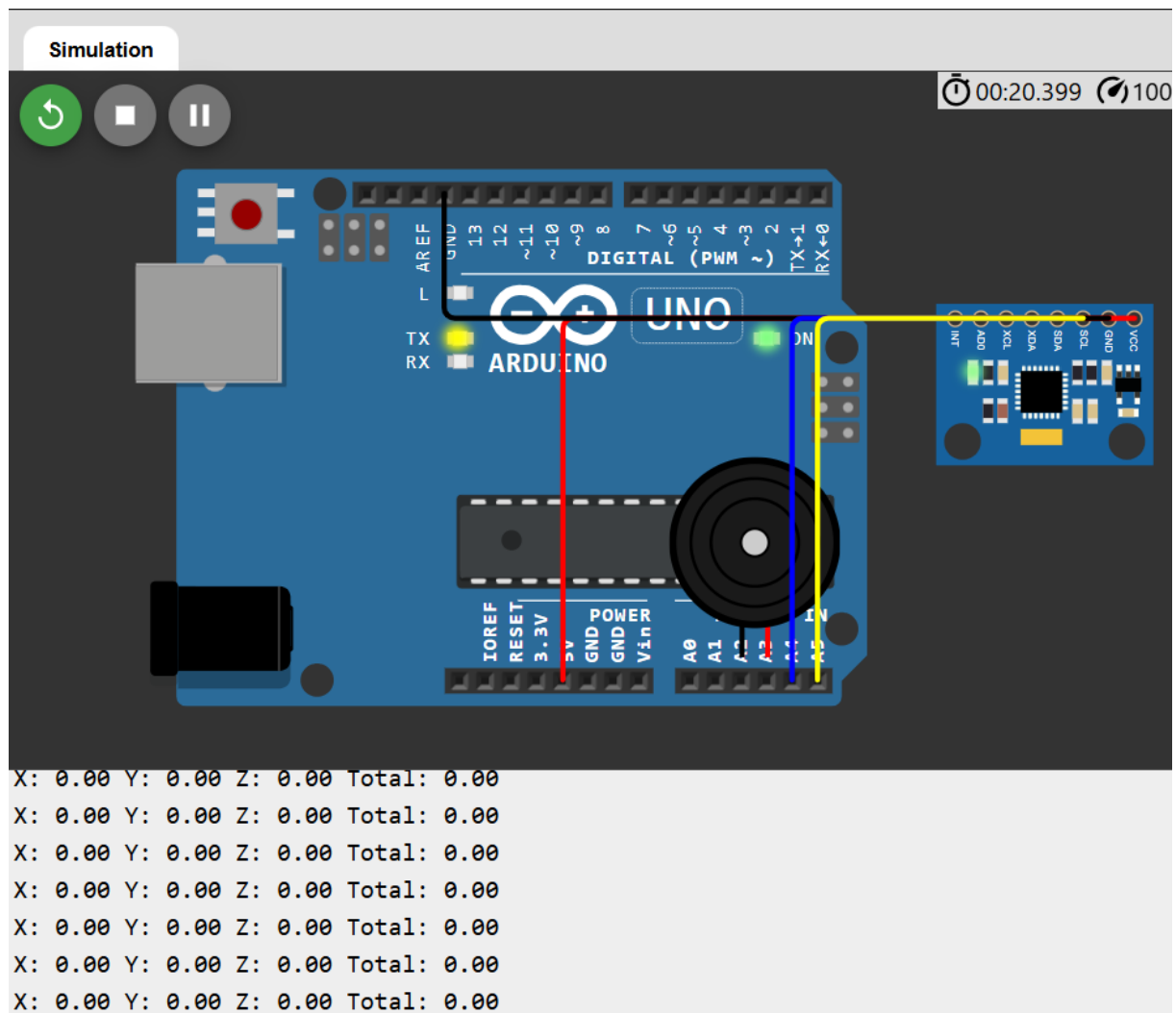


FIGURE 2 – Captures d'écran de la simulation Wokwi

4.2. Capture d'écran de la simulation dans le cas avec alerte

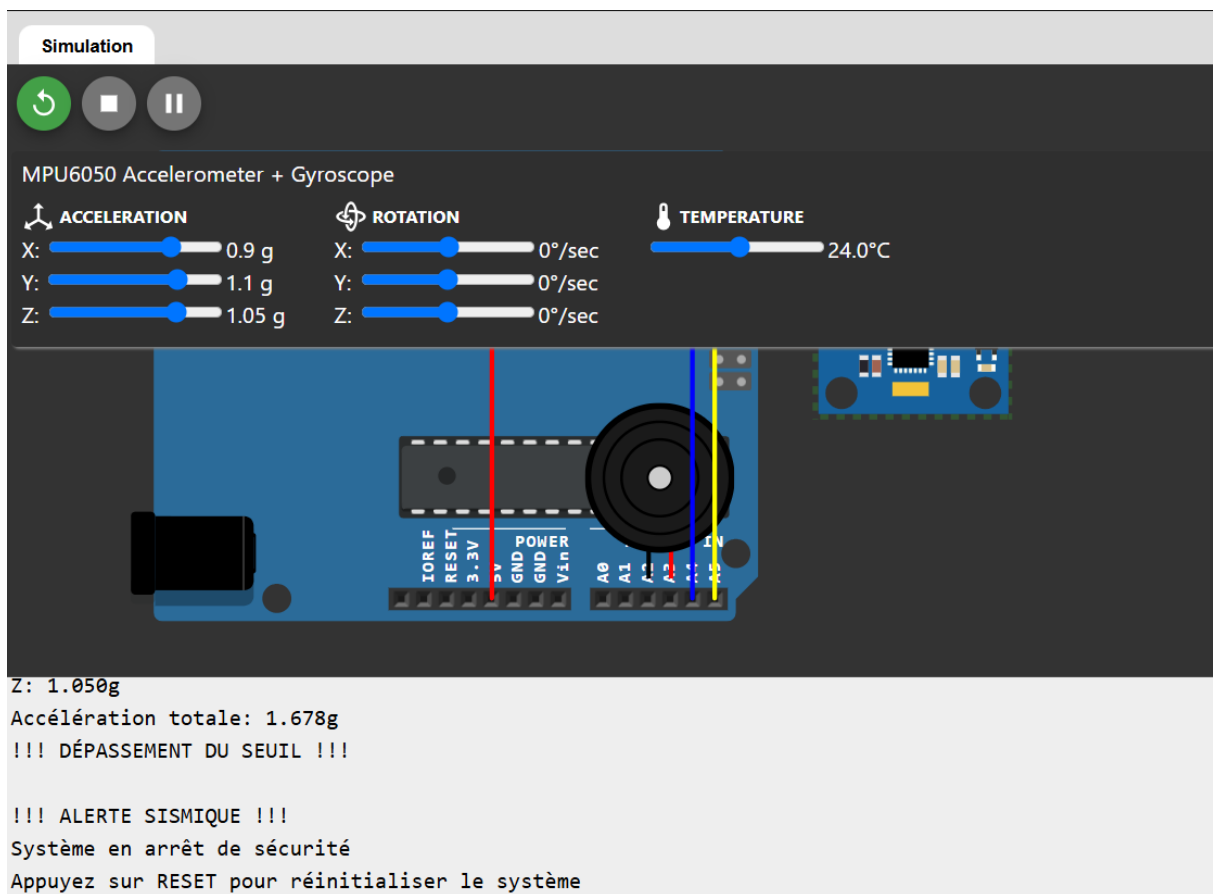


FIGURE 3 – Captures d'écran de la simulation Wokwi

4.3. Lien vers le projet

Simulation Wokwi : <https://wokwi.com/projects/412929632394461185>

4.4. Code source

Le code source complet est disponible sur GitHub : <https://github.com/zarathez/earthquake-detection-system>

Points clés du code :

- Fonction `triggerEmergencyState()` pour l'arrêt de sécurité
- Fonction `blinkAlarm()` pour le pattern sonore optimisé
- Fonction `resetSystem()` pour la réinitialisation sécurisée

5. Conclusion

5.1. Synthèse du projet

Le système développé répond aux objectifs initiaux en fournissant une solution de détection sismique économique et efficace. Les tests en simulation montrent une bonne réactivité du système et une détection fiable des mouvements significatifs, avec un mécanisme de sécurité robuste pour assurer la fiabilité post-détection.

5.2. Innovation

L'aspect innovant du projet réside dans :

- L'utilisation d'un algorithme de détection optimisé pour minimiser les faux positifs
- L'intégration de seuils adaptatifs basés sur l'historique des mesures
- La possibilité d'extension future via le bus I2C

5.3. Difficultés et challenges

1. Challenges techniques :

- Calibration précise du MPU6050
- Filtrage du bruit et des vibrations parasites
- Optimisation de la consommation énergétique
- Gestion des faux positifs

2. Solutions apportées :

- Implémentation d'un filtre numérique
- Utilisation de moyennes glissantes
- Optimisation du code pour réduire les calculs
- Système d'arrêt de sécurité avec réinitialisation manuelle

6. Références

1. Documentation technique :

- Arduino UNO : <https://docs.arduino.cc/hardware/uno-rev3>
- MPU6050 : <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- Wokwi : <https://docs.wokwi.com/>

2. Articles scientifiques :

- "Low-Cost Earthquake Early Warning Systems" - Journal of Seismology, 2023
- "MEMS Accelerometers for Seismic Monitoring" - Sensors Journal, 2022

3. Ressources en ligne :

- Arduino Project Hub : <https://projecthub.arduino.cc>
- MPU6050 Library Documentation : <https://github.com/electronccats/mpu6050>
- Seismic Detection Algorithms : <https://www.seismicsource.com>