

DOCUMENTATION TECHNIQUE DES TESTS EN ELECTRONIQUE

Test 1 : Gyroscope et accéléromètre

S'orienter dans l'espace, savoir distinguer le haut, le bas, la gauche, la droite, l'avant et l'arrière, est une compétence naturelle chez l'être humain. Mais comment faire pour qu'un robot ou un système automatisé reproduise cette capacité ?

C'est là qu'interviennent les capteurs inertiels : des composants électroniques capables de « sentir » les mouvements et rotations d'un objet dans l'espace.

Nous avons choisi d'explorer cette technologie avec un capteur MPU6050, un module combinant :

- ✓ un accéléromètre 3 axes, pour détecter les accélérations (y compris la gravité),
- ✓ un gyroscope 3 axes, pour mesurer les vitesses de rotation autour de ces 3 axes.

Ce capteur sera placé dans la paume d'une main et connecté à un microcontrôleur ATmega328P. À chaque geste : lever la main, la baisser, la tourner, les informations seront affichées en temps réel sur un écran LCD.

✓ **En quoi ce test relève d'une importance capitale ?**

Ce test constitue une immersion concrète dans les systèmes embarqués et la robotique, où les capteurs inertiels comme ceux que l'on retrouve dans les drones, smartphones ou robots sont essentiels pour mesurer la position, la vitesse et la stabilité du système. En combinant un accéléromètre (qui mesure les accélérations linéaires) et un gyroscope (qui mesure la vitesse de rotation), on peut reconstruire en temps réel la trajectoire et l'orientation d'un objet dans l'espace : c'est ce que l'on appelle la **fusion de données**, un processus indispensable dans les applications à 3D.

Grâce à la simplicité du module MPU6050 (interface I2C standard) et la restitution des résultats sur un écran LCD, ce test offre une approche pédagogique efficace : il permet de visualiser directement comment un geste manuel se traduit en direction détectée, sans complexité inutile. Ce compromis entre technique, accessibilité, et interactivité en fait un excellent point de départ pour quiconque souhaite comprendre la détection de mouvement, l'électronique embarquée et

l'orientation spatiale avant de creuser des solutions plus complexes (filtres avancés, modèles AHRS, etc.).

1. Objectifs de ce test :

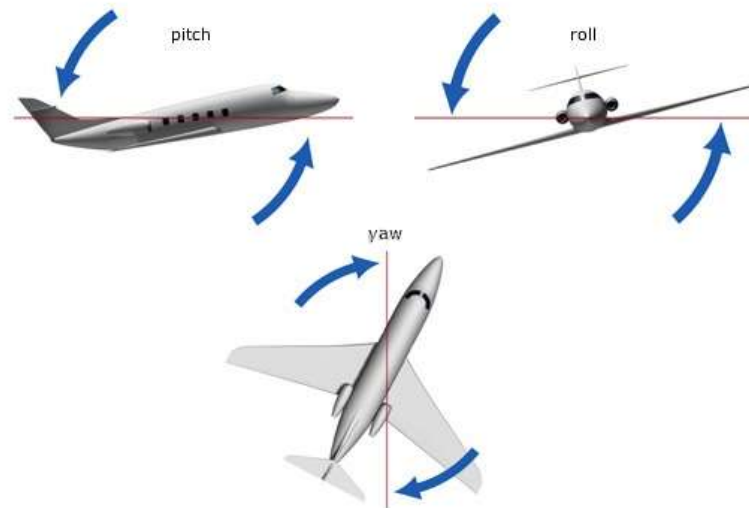
Ce test vise à développer et valider plusieurs compétences techniques et pratiques essentielles dans le domaine des systèmes embarqués et de la détection de mouvement. Plus précisément, il s'agit de :

- Identifier et comprendre le fonctionnement d'un capteur inertiel combiné (MPU6050) intégrant un accéléromètre et un gyroscope, et savoir exploiter ses données pour mesurer l'orientation et les mouvements dans l'espace.
- Mettre en œuvre la communication I2C entre le capteur et le microcontrôleur ATmega328P, en maîtrisant l'échange de données numériques via ce protocole standardisé.
- Programmer un microcontrôleur Arduino pour lire les données brutes du capteur, appliquer un traitement logiciel (filtrage, calibration) et calculer les angles d'orientation (yaw, pitch, roll).
- Développer une logique de détection de direction capable d'interpréter les variations d'angles et d'accélération pour déterminer les mouvements de la main dans les six directions principales (haut, bas, gauche, droite, avant, arrière).
- Assurer l'affichage en temps réel des résultats sur un écran LCD I2C, permettant une visualisation claire et intuitive des mouvements détectés.
- Comprendre l'importance de la calibration et du filtrage pour améliorer la précision et la fiabilité des mesures issues du capteur inertiel.
- Acquérir une démarche méthodique de test et validation, en vérifiant le bon fonctionnement du système dans des conditions réelles d'utilisation.

2. Comprendre les orientations : roulis, tangage et lacet :

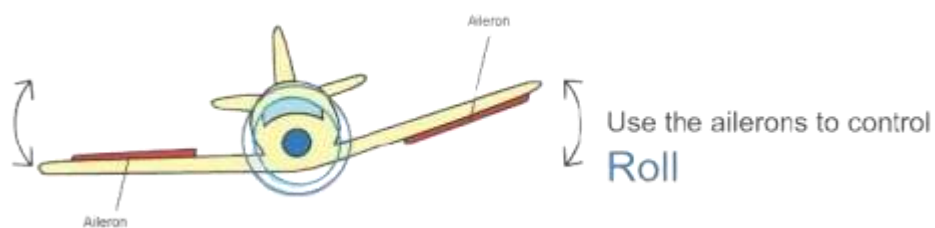
Pour décrire précisément comment un objet (comme ta main équipée du capteur MPU6050) s'oriente et se déplace dans l'espace, on utilise trois angles fondamentaux appelés roulis (roll), tangage (pitch) et lacet (yaw). Ces trois mouvements correspondent à des rotations autour de

trois axes perpendiculaires, et permettent de définir la position et l'orientation d'un objet dans un espace à trois dimensions.



a) Le roulis (Roll)

C'est la rotation autour de l'axe longitudinal, c'est-à-dire un mouvement d'inclinaison latérale, comme si ta main penchait vers la droite ou la gauche.



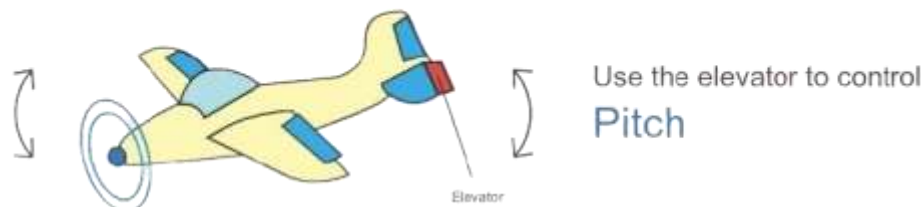
Démonstration pratique : Imagine que tu tiens ta main devant toi et que tu la fais basculer sur le côté, comme si tu voulais verser un verre d'eau. Ce mouvement est un roulis.

Importance dans notre système :

Le roulis permet de détecter si la main s'incline sur la gauche ou la droite, ce qui est essentiel pour comprendre les mouvements latéraux.

b) Le tangage (Pitch)

C'est la rotation autour de l'axe transversal, correspondant à un mouvement de bascule avant-arrière.



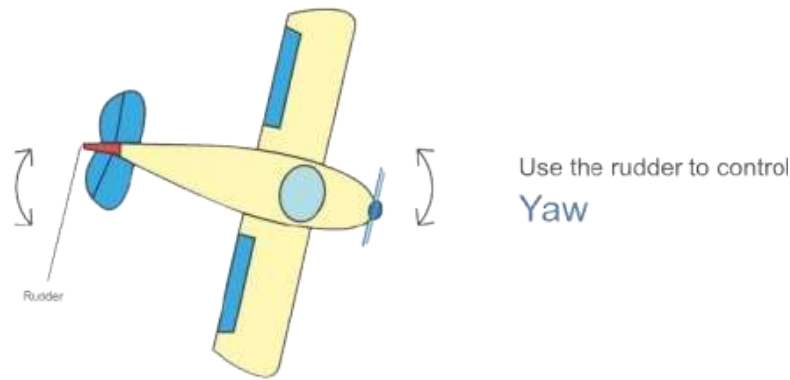
Démonstration pratique : Imagine que tu fais un signe de tête pour dire « oui », en penchant ta main vers l'avant ou l'arrière. Ce mouvement est un tangage.

Importance dans notre système :

Le tangage permet de détecter si la main se penche vers l'avant ou vers l'arrière, ce qui correspond aux mouvements de translation avant/arrière.

c) Le lacet (Yaw)

C'est la rotation autour de l'axe vertical, c'est-à-dire un mouvement de rotation horizontale, comme si ta main tournait sur elle-même à plat.



Démonstration pratique : Imagine que tu tournes ta main vers la gauche ou la droite sans la pencher, juste en la faisant pivoter sur elle-même. Ce mouvement est un lacet.

Importance dans notre système :

Le lacet permet de détecter les rotations horizontales, ce qui est crucial pour comprendre les mouvements de rotation autour de l'axe vertical.

3. Pourquoi ces angles sont-ils importants dans notre système ?

Le MPU6050 mesure ces trois rotations grâce à son gyroscope, et les angles correspondants (roll, pitch, yaw) sont calculés pour déterminer précisément l'orientation de la main dans l'espace. En combinant ces données avec celles de l'accéléromètre (qui mesure les accélérations linéaires), le système peut détecter non seulement la position, mais aussi le mouvement et la direction de la main.

Cette compréhension des trois axes de rotation est essentielle pour :

- ✓ Interpréter correctement les gestes (lever, baisser, tourner la main).
- ✓ Détecter les directions de mouvement (haut, bas, gauche, droite, avant, arrière).
- ✓ Améliorer la précision des mesures en filtrant et fusionnant les données des capteurs.

4. Choix du matériel pour ce test :

Pour réaliser ce projet de détection d'orientation et de mouvement, le choix du matériel est crucial afin d'assurer un fonctionnement fiable, précis et adapté à une intégration finale. Nous avons opté pour des composants permettant à la fois simplicité, compacité et performance.

✓ Microcontrôleur : ATmega328P seul

Plutôt que d'utiliser une carte Arduino Uno complète, nous avons choisi d'employer le microcontrôleur ATmega328P en version « puce seule ». Cette approche présente plusieurs avantages :

- Miniaturisation : Le microcontrôleur seul occupe beaucoup moins de place, facilitant l'intégration dans un boîtier ou un montage personnalisé.
- Coût réduit : Le prix du microcontrôleur nu est nettement inférieur à celui d'une carte Arduino complète.
- Personnalisation du circuit : Cela permet de sélectionner précisément les composants périphériques nécessaires (oscillateur, alimentation, reset) et d'optimiser le montage.
- Approche pédagogique : Cette méthode offre une meilleure compréhension du fonctionnement matériel d'un système embarqué en réalisant soi-même le circuit minimal.

Composants nécessaires pour faire fonctionner l'ATmega328P

- Cristal oscillateur 16 MHz avec condensateurs pour fournir une horloge stable.
- Résistance pull-up sur la broche RESET pour éviter les réinitialisations intempestives.
- Alimentation 5 V régulée avec condensateurs de découplage.
- Convertisseur USB-série externe (comme un module FTDI) pour programmer le microcontrôleur via USB.
- Câblage soigné pour relier le microcontrôleur aux capteurs et à l'écran.

✓ Capteur inertiel : MPU6050

Le choix du capteur s'est porté sur le module MPU6050, qui combine un accéléromètre 3 axes et un gyroscope 3 axes dans un seul composant.

Pourquoi le MPU6050 ?

- ❖ Mesure complète du mouvement : Il fournit à la fois les accélérations linéaires et les vitesses angulaires, indispensables pour déterminer l'orientation dans l'espace.
- ❖ Interface I2C : La communication via un bus I2C standard simplifie le câblage et permet de connecter plusieurs périphériques sur les mêmes lignes SDA et SCL.
- ❖ Intégration du DMP : Le MPU6050 intègre un processeur de mouvement numérique (DMP) qui réalise la fusion des données pour fournir des angles d'orientation stables.
- ❖ Compatibilité avec ATmega328P : Fonctionne sous une alimentation de 3,3 V à 5 V, compatible avec le microcontrôleur utilisé.

NB : Particularités à prendre en compte

- Le module MPU6050 fonctionne généralement sous 3,3 V, mais la plupart des modules GY-521 intègrent un régulateur 3,3 V et des résistances pull-up compatibles 5 V, ce qui facilite l'utilisation avec un ATmega328P alimenté en 5 V.
- Il faut veiller à connecter SDA (données) et SCL (horloge) aux broches A4 et A5 du microcontrôleur.
- L'adresse I2C par défaut est 0x68, modifiable en connectant la broche AD0 à 3,3 V (adresse 0x69).

✓ Affichage : écran LCD I2C 16x2

Pour visualiser les résultats en temps réel, un écran LCD 16 colonnes x 2 lignes avec interface I2C a été choisi.

Avantages de l'écran LCD I2C

- ❖ Câblage simplifié : Utilise seulement 4 fils (VCC, GND, SDA, SCL) pour la communication et l'alimentation.
- ❖ Compatibilité I2C : Partage le bus I2C avec le MPU6050, réduisant le nombre de connexions nécessaires.
- ❖ Affichage clair et lisible : Permet de visualiser directement la direction détectée sans matériel supplémentaire.

5. Architecture et principe de fonctionnement :

Ce système utilise le capteur **MPU6050** pour détecter comment un objet (comme une main) bouge et s'oriente dans l'espace. Voici comment fonctionnera le système de notre test :

✓ *Acquisition des données*

Le MPU6050 contient deux capteurs importants :

- ❖ **Accéléromètre 3 axes** : il mesure les accélérations sur trois directions (avant/arrière, gauche/droite, haut/bas). Cela permet de savoir si la main accélère ou change d'inclinaison.
- ❖ **Gyroscope 3 axes** : il mesure la vitesse à laquelle la main tourne autour de ces trois axes.

Ces deux capteurs travaillent ensemble pour donner une image complète du mouvement.

✓ *Prétraitement des données*

Les mesures brutes du capteur peuvent contenir du bruit (petites erreurs ou fluctuations). Pour améliorer la qualité :

- On applique un **filtre passe-bas** qui lisse les données de l'accéléromètre, en gardant les tendances importantes et en supprimant les petites perturbations.
- Le capteur utilise aussi une fonction appelée **DMP (Digital Motion Processor)** qui prépare les données et envoie une alerte (interruption) quand elles sont prêtes à être lues. Cela permet au microcontrôleur de récupérer les données au bon moment, sans perdre d'informations.

✓ *Fusion et calcul des orientations*

Le DMP du MPU6050 combine les données de l'accéléromètre et du gyroscope pour calculer la position exacte de la main dans l'espace. Il utilise une méthode mathématique appelée **quaternion** pour représenter l'orientation sans ambiguïté.

À partir de ces quaternions, on calcule trois angles simples à comprendre :

- ✓ **Roll (roulis)** : inclinaison gauche-droite.
- ✓ **Pitch (tangage)** : inclinaison avant-arrière.
- ✓ **Yaw (lacet)** : rotation sur place à gauche ou à droite.

Le système corrige aussi l'accélération mesurée pour enlever l'effet de la gravité, afin de détecter uniquement les mouvements réels.

✓ *Détection de la direction*

Pour savoir dans quelle direction la main se déplace, le système :

- Surveille l'accélération sur l'axe vertical (Z) pour détecter un mouvement vers le haut ou vers le bas.
- Analyse les variations des angles roll, pitch et yaw pour détecter les mouvements avant/arrière, gauche/droite et les rotations.
- Si aucun mouvement significatif n'est détecté, l'état est considéré comme étant **stable**.

✓ *Affichage et retour utilisateur*

- 🌈 La direction détectée est affichée en temps réel sur un écran LCD I2C 16×2, facile à lire et à comprendre.
- 🌈 Les données peuvent aussi être envoyées sur un moniteur série (ordinateur) pour suivre les mesures, calibrer le système et améliorer la précision.

6. Liste du matériel

Composants	Spécifications principales	Rôle dans le test
<i>ATmega328P</i>	Horloge 16 MHz, 32 kB Flash	Microcontrôleur principal, traitement des données et contrôle du système
<i>MPU6050 (GY-521)</i>	Interface I2C, DMP intégré, alimentation 3 V–5 V	Capteur inertiel combinant accéléromètre et gyroscope MEMS pour mesurer mouvements et orientation
<i>Écran LCD 16×2 I²C</i>	Adresse I2C 0x27 ou 0x3F	Interface utilisateur pour affichage en temps réel des directions détectées
<i>Module FTDI</i>	Interface USB-TTL, 6 broches	Programmation du microcontrôleur et communication série pour réaliser un debug
<i>Veroboard ou PCB</i>	24×16 bandes, format Eurorack	Support mécanique pour montage et soudure des composants
<i>Fer à souder + étain</i>	Pointe fine 0,4 mm	Assemblage des composants électroniques par soudure
<i>Multimètre</i>	Mesure de continuité, tension, résistance	Vérification des connexions et tests électriques
<i>Breadboard + câbles Dupont</i>	Montage sans soudure, prototypage rapide	Réalisation de prototypes et tests avant soudure définitive

i. Microcontrôleur

- ATmega 328P – PU (version en boîtier DIP28)

- Source (7 à 12V)
- Condensateurs de filtrage
 - ✓ $1 \times 10 \mu\text{F}$ électrolytique (entre VCC et GND pour l'alimentation générale)



- ✓ $2 \times 100 \text{ nF}$ céramique (entre VCC et GND et AVCC et GND pour le découplage)



iii. Oscillateur

- Quartz 16 MHz



- 2 condensateurs 22 pF



iv. Broche Reset

- Résistance 10 k Ω (entre RESET et +5V pull-up)



- Bouton poussoir pour forcer le RESET à GND



v. Connexion pour la programmation du microcontrôleur

- Une carte Arduino UNO
- Convertisseur USB-TTL



vi. Test visuel du programme

- Une LED (pour le clignotement du test Blynk)



- Une résistance 220 Ω - 330 Ω (en série avec la LED pour sa protection)



7. Réalisation du circuit :

A. Test du microcontrôleur ATmega328P :

Avant d'intégrer l'ATmega328P à ton circuit définitif, il est conseillé de le tester sur une breadboard (plaque d'essai).

- Montage minimal : Place le microcontrôleur sur la breadboard avec une alimentation 5 V, un cristal 16 MHz (et ses deux condensateurs de 22 pF), une résistance de 10 kΩ entre RESET et Vcc, et relie les broches d'alimentation (Vcc, GND, AVcc).
- Test de fonctionnement : Téléverse un code simple (ex. blink LED) pour vérifier que la puce fonctionne correctement.

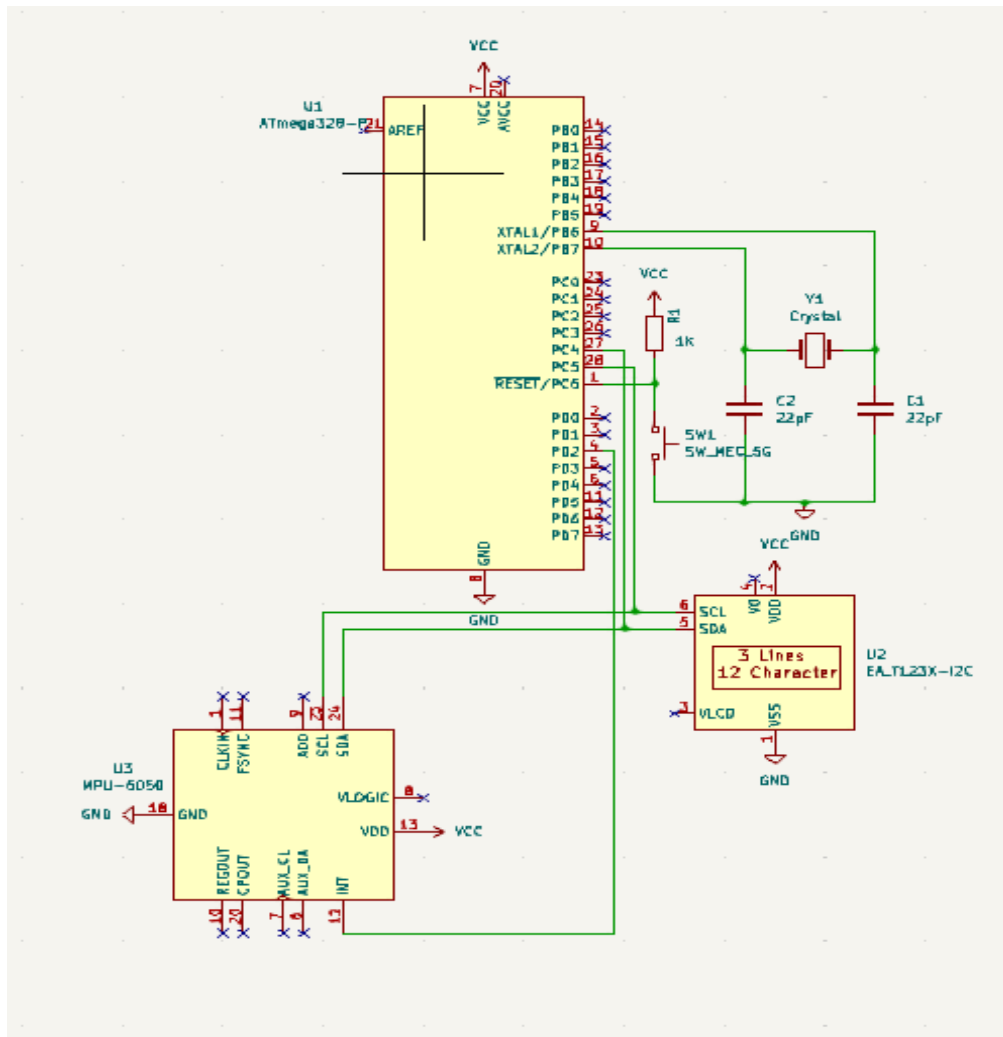
B. Gravure du bootloader :

Le bootloader permet à l'ATmega328P d'être programmé comme un Arduino classique. Si tu utilises une puce neuve, il faut graver ce bootloader :

- Utilise une carte Arduino comme programmeur ou un programmeur externe.
- Branchements : Connecte les broches MOSI, MISO, SCK, RESET, Vcc et GND de l'Arduino à celles de l'ATmega328P.
- Gravure : Dans l'IDE Arduino, sélectionne « Arduino as ISP » puis « Burn Bootloader ». Choisis le bon modèle (par ex. « ATmega328 on a breadboard (8 MHz internal clock) » si tu utilises l'oscillateur interne, ou « Arduino Uno » si tu utilises un cristal 16 MHz).

C. Conception du schéma sous KiCad :

Ce circuit a pour but de mesurer les mouvements à l'aide du capteur MPU-6050 (accéléromètre et gyroscope). Ces mesures seront traitées et formatées par le microcontrôleur ATmega328P qui seront affichées sur un écran LCD. L'ensemble des composants communiquent grâce au protocole I2C, ce qui permet une réduction du nombre de connexions nécessaires .



SCHEMA DU CIRCUIT DU MONTAGE DU CAPTEUR MPU-6050

Le schéma a été réalisé dans KiCad. Les composants ont été choisis depuis la librairie KICad officielle. Les connexions sont nommées de façon explicite pour faciliter la lecture.

LISTE ET DESCRIPTION DES COMPOSANTS

- **ATmega328P**

C'est un microcontrôleur 8 bits (même que celui de l'arduino Uno) . Il est utilisé ici pour l'acquisition , le traitement et la transmission de données . Il fonctionne avec un quartz externe pour une meilleure précision d'horloge .

- **MPU-6050**

C'est un capteur inertielle 6 axes (accéléromètre, gyroscope) communiquant en *I2C*. Il donne des mesures d'accélérations, vitesse angulaire et température.

- **LCD I2C**

Il s'agit d'un écran alphanumérique (2 lignes x 16 caractères). Il est interfacé via un module I2C et affiche les valeurs acquises par le MPU-6050 pour surveillance en temps réels.

- **QUARTZ, CONDENSATEURS ET RESISTANCE**

Le quartz génère l'horloge du microcontrôleur.

Les condensateurs C1 et C2 sont typiquement de 22 pF et la résistance de 1K

- **BOUTON POUSSOIR**

Il permet de réinitialiser manuellement le microcontrôleur

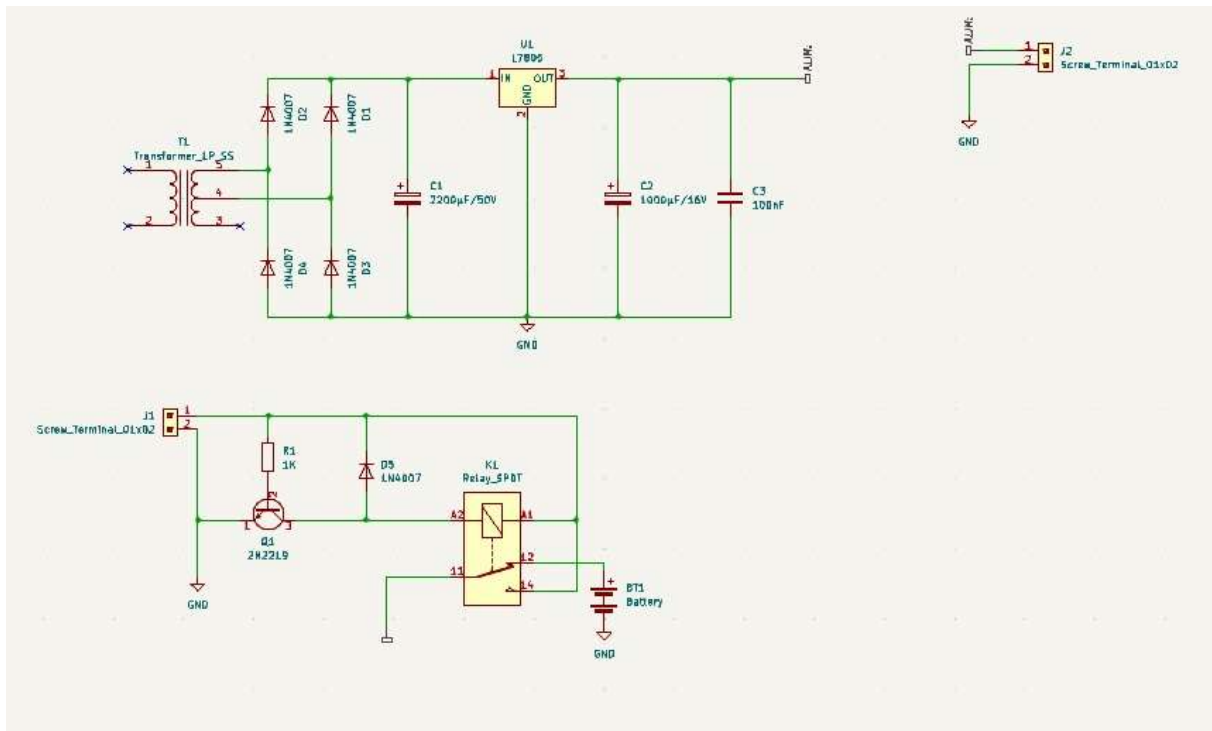
FONCTIONNEMENT GLOBAL

A la mise sous tension, le microcontrôleur initialise le capteur MPU-6050 via le module I2C. Il lit cycliquement les valeurs d'accélérations et de rotation. Les données sont ensuite traitées et converties dans un format lisible. Les résultats sont envoyés à l'écran LCD I2C pour affichage.

ALIMENTATION

Tous les composants sont alimentés en +5V (VCC).

Le MPU-6050 possède une broche VLOGIC pour s'adapter aux logiques de 3.3V ou 5V. On peut alimenter par deux manières : une par un adaptateur et une autre par une batterie.



PROTOCOLE DE COMMUNICATION UTILISE

Les capteurs d'accélération et de déplacement doivent pouvoir soumettre des informations précises et lisibles par l'utilisateur. C'est là qu'interviennent les broches analogiques de traitement bidirectionnel de données :

- La broche ***SDA*** : permet de recevoir, traiter et transmettre les données provenant des capteurs à l'afficheur LCD.
- La broche ***SCL*** : elle génère un signal d'horloge qui s'occupe du temps de transmission et réception des informations numériques entre les capteurs et l'afficheur LCD : elle réalise une ***synchronisation de données***.

D. Soudure des composants :

- ***Préparation*** : préchauffer le fer à souder (300–350 °C), nettoyer la panne, préparer les composants et le veroboard.

- Soudure des composants traversants (THT) : placer chaque composant, souder une patte, ajuster, puis souder les autres. Couper l'excédent de patte.
- Soudure des composants SMD (si applicable) : utiliser de la pâte à souder et du flux, commencer par fixer un coin, puis souder les autres broches (drag soldering ou soudure point par point).

NB : Utiliser du fil à souder fin (0,38 mm recommandé), éviter les ponts de soudure, et vérifier chaque joint (il doit être brillant et conique).

Inspecter visuellement chaque soudure, corriger les défauts (ponts, manque de soudure, etc.).

E. Programmation et test du circuit assemblé :

- ✓ Connexion du convertisseur USB-série : brancher le module FTDI (ou équivalent) aux broches RX, TX, Vcc, GND, (et DTR/RESET si votre convertisseur en possède) de l'ATmega328P.
- ✓ Téléversement du code : utiliser l'IDE Arduino pour charger un programme de test (ex. blink ou lecture capteur).
- ✓ Test fonctionnel : vérifier le bon fonctionnement du microcontrôleur, de la communication I2C (avec le MPU6050 et l'écran LCD), et de l'alimentation.