

Date de rendu : 02/02/2024

MISE EN OEUVRE DES DSP

CONCEPTION, PROGRAMMATION ET MISE EN OEUVRE D'UN TRAITEMENT
NUMÉRIQUE DU SIGNAL EN TEMPS RÉEL

Projet : Interrupteur Clap



Fait par:
DIA Saliou
DIKOUANGA OGOMBE Lucia
TOURE Mouhamadou

Encadré par:
Farouk Benmeddour

Sommaire

CONCEPTION INITIALE DU PROJET : Fondations en DSP et dsPIC	2
I) Introduction aux DSP et dsPIC	3
1) Définition des DSP et dsPIC	3
2) Comparaison entre un dsp et les processeurs généraux	3
3) Applications en mécanique	4
II) Caractéristiques des microcontrôleurs dsPIC	4
1) Architecture dsPIC intégré	5
2) Unités de traitement du signal	6
III) Préparation du terrain	7
1) Applications possibles pour le Dspic	7
2) Le choix du projet	8
3) Analyse des besoins spécifiques au projet	8
4) Etude des contraintes matérielles et logicielles	8
IV) Choix du Microcontrôleur et Environnement	9
1) Évaluation des microcontrôleurs disponibles	9
2) Sélection en fonction des exigences du projet	10
PHASE OPÉRATIONNELLE DU PROJET : De la Création à l'Implémentation	11
I) Création du projet	11
1) Création d'un nouveau projet dans l'IDE	11
2) Configuration initiale du projet (fréquence d'horloge, broches d'E/S, etc)	11
II) Introduction aux Outils dsPIC	11
1) Présentation des outils DSP disponibles dans l'IDE	11
2) Utilisation des bibliothèques DSP pré intégrées	12
3) Codage sous le logiciel MPLAB	12
PHASE FINALE DU PROJET : Tests, validation, et communication des résultats	13
I) Tests et validation	14
II) Perspectives	15
Conclusion	17
Annexes	18

Partie 1

CONCEPTION INITIALE DU PROJET : Fondations en DSP et dsPIC

I) Introduction aux DSP et dsPIC

1) Définition des DSP et dsPIC

❖ Définition DSP

Les DSP “Digital Signal Processors” en anglais sont des processeurs optimisés permettant la manipulation et le traitement de signaux tels que le son, la vidéo à l’aide de son unité de calcul spécifique (multiplicateur, additionneur, accumulateur de données), tout cela dans le but d’exécuter plusieurs fonctions telles que le filtrage, la détection, l’extraction, la compression, etc. Ils sont dans la plupart des cas utilisés dans le cadre d’applications de traitement numérique du signal en temps réel.

❖ Définition dsPic

Le microcontrôleur dsPIC fait partie de la famille des microcontrôleurs 16 bits de Microchip. L’architecture du dsPIC ou contrôleur de signal numérique est la même que celle de la famille PIC 16 bits, avec des blocs supplémentaires de traitement du signal numérique. Cela permet de renforcer le microcontrôleur standard avec un puissant processeur mathématique nécessaire à un processeur de signal numérique. Cela offre une quantité considérable de fonctionnalités dans une seule puce. Cette intégration du DSP dans le MCU standard offre de grandes performances au microcontrôleur.

2) Comparaison entre un dsp et les processeurs généraux

Les microprocesseurs et les processeurs de traitement numérique du signal (DSP) se distinguent fondamentalement par leur application spécifique. Le microprocesseur est polyvalent, adapté à diverses applications logicielles, tandis que le DSP est spécifiquement conçu pour le traitement du signal, bénéficiant d’une architecture optimisée.

Les DSP excellent dans les calculs arithmétiques complexes, comme la multiplication, les rendant efficaces pour des applications telles que le traitement d’image, la reconnaissance vocale et les télécommunications. Leur présence est notable dans des

dispositifs nécessitant un traitement rapide des données audio ou vidéo en temps réel, comme les téléphones cellulaires, les lecteurs DVD et les appareils photo numériques.

Comparativement, les microprocesseurs sont des dispositifs polyvalents destinés à des applications logicielles générales, telles que les traitements de texte et les navigateurs Web. Ce sont des applications où la vitesse est importante, mais pas critique. Les progrès technologiques, notamment en termes de rapidité (grâce au faible temps de commutation) et de puissance de calcul (grâce au nombre de bits des bus internes) des DSP, ont marqué une évolution significative parallèlement aux microprocesseurs et aux microcontrôleurs dans le domaine de l'informatique.

3) Applications en mécatronique

Les DSP dans le domaine de la mécatronique sont utilisés pour :

- **La commande et l'asservissement** en temps réel de systèmes dynamiques tels qu'un robot, un système de positionnement, un actionneur, etc
- **L'obtention d'informations** précises sur l'état d'un système mécatronique par le traitement, l'interprétation, le filtrage de données d'un capteur (radar, transmissomètre, etc)
- **Le traitement d'images et de vidéos** dans les systèmes de vision par ordinateur que l'on retrouve dans les robots autonomes, les drones, les systèmes de sécurité, etc
- **La communication sans fil** pour la coordination et la synchronisation des actions entre des composants ou des systèmes mécatroniques (aéronefs, sous-marins, etc)
- **Le contrôle, la conversion et la distribution d'énergie** dans les systèmes de production d'énergie renouvelable
- **Le diagnostic de panne** par l'analyse des signaux et la détection d'éventuelles pannes ou problèmes dans les systèmes mécatroniques
- **La réduction ou l'amplification de son** comme un réducteur de bruit ambiant utilisé dans les casques audios pour permettre à l'utilisateur une meilleure expérience sonore

II) Caractéristiques des microcontrôleurs dsPIC

1) Architecture dsPIC intégré

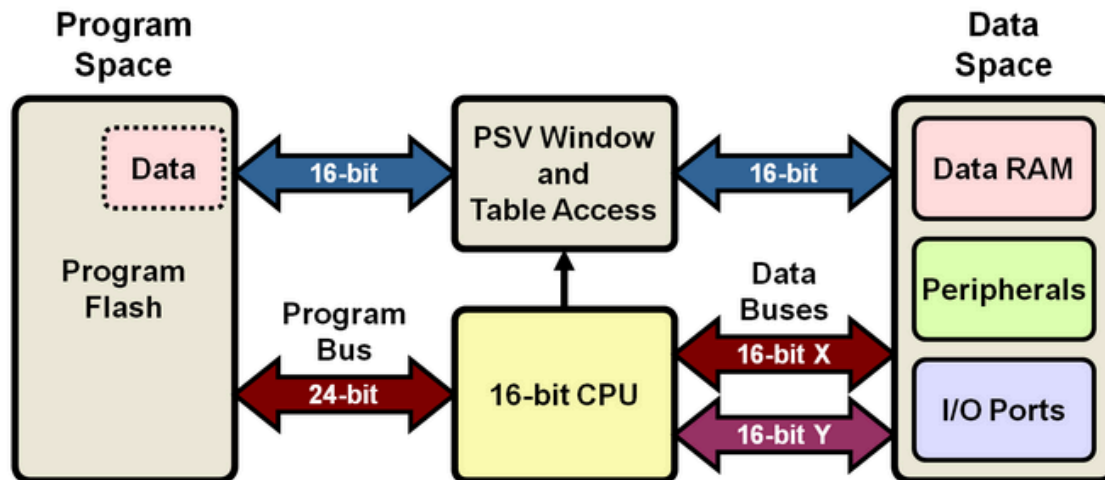


Figure 1: Architecture

Le dsPIC présente les caractéristiques clés suivantes :

- **Architecture 16 bits** : Le dsPIC est un microcontrôleur 16 bits, ce qui lui confère une capacité de traitement plus élevée par rapport aux microcontrôleurs 8 bits.
- **Traitement du signal numérique (DSP)** : Le dsPIC dispose de blocs supplémentaires dédiés au traitement du signal numérique, ce qui lui permet d'effectuer des opérations mathématiques complexes et des algorithmes de traitement du signal avec une grande précision.
- **Capacités de contrôle en temps réel** : Le dsPIC combine les fonctionnalités d'un microcontrôleur traditionnel avec les capacités de contrôle en temps réel nécessaires pour les applications critiques. Il peut gérer efficacement les interruptions prioritaires et les tâches simultanées.
- **Périphériques intégrés étendus** : Le dsPIC offre une large gamme de périphériques intégrés, tels que des convertisseurs analogique-numérique (CAN), des interfaces de communication (UART, SPI, I2C), des temporisateurs, des entrées/sorties numériques, etc.
- **Gestion de l'alimentation** : Le dsPIC intègre des fonctionnalités avancées de gestion de l'alimentation, ce qui permet d'optimiser la consommation d'énergie et de prolonger la durée de vie de la batterie dans les applications mobiles.
- **Moteur DSP complet** : Le DSPIC dispose d'un moteur DSP complet qui permet d'effectuer des opérations de traitement du signal avec une grande efficacité et une faible latence.

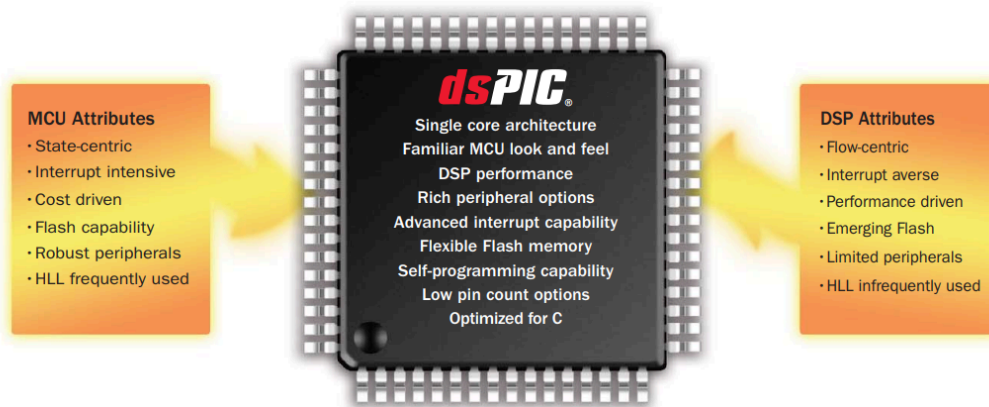


Figure 2: Le dsPIC

2) Unités de traitement du signal

Les unités de traitement du signal des DSpic (DSP) sont des composants électroniques spécialisés conçus pour traiter des signaux numériques en temps réel. Le traitement **numérique** du signal s'oppose au traitement **analogique** du signal, sur lequel se basaient tous les anciens systèmes électroniques et qui manipule les signaux à leur état naturel, c'est-à-dire sous forme d'ondes (signaux *analogiques*). Le traitement numérique consiste à manipuler les signaux sous forme de suites d'échantillons numériques obtenus à partir de signaux physiques réels au moyen de convertisseurs analogique-numériques (CAN), leur appliquer les traitements mathématiques appropriés, puis éventuellement les restituer sous leur forme originelle à l'aide d'un convertisseur numérique-analogique (CNA).

Les systèmes numériques présentent plusieurs avantages par rapport aux systèmes analogiques. Le plus significatif d'entre eux est qu'un système numérique permet d'effectuer à peu de frais des tâches qu'il serait très difficile voire même impossible de réaliser en analogique.

Toutes ces considérations, ajoutées au fait que les systèmes numériques profitent des progrès toujours croissants réalisés dans le domaine de la VLSI (*Very Large Scale Integration*), font des systèmes numériques un support privilégié pour les applications de traitement du signal.

Voici les caractéristique de l'unité de traitement de signal du dspic :

- **Architecture de pipeline** : Les DSP utilisent une architecture de pipeline pour optimiser le débit de traitement. Les étapes du traitement du signal sont divisées en segments, chacun effectuant une opération spécifique. Ces segments sont disposés en série, permettant à chaque cycle d'horloge de traiter une nouvelle portion du signal.

- **Unités de multiplication-accumulation (MAC)** : Les DSP sont souvent dotés d'unités spécialisées appelées MAC, conçues pour effectuer rapidement des opérations de multiplication suivies d'une addition (multiplication-accumulation). Ces opérations sont courantes dans de nombreux algorithmes de traitement du signal, tels que les filtres numériques.
- **Jeu d'instructions spécialisé** : Les DSP ont un jeu d'instructions optimisé pour les opérations de traitement du signal. Cela peut inclure des instructions SIMD pour le traitement en parallèle de plusieurs données, accélérant ainsi les calculs impliquant des tableaux de données.
- **Gestion de la mémoire** : Les DSP intègrent souvent plusieurs niveaux de mémoire, y compris une mémoire cache rapide et une mémoire RAM plus grande. La gestion efficace de la mémoire est cruciale pour minimiser les temps d'accès et garantir un traitement en temps réel. Certains DSP incluent également des mémoires spéciales pour stocker des coefficients de filtres fréquemment utilisés.
- **Instructions de contrôle de flux** : Les DSP incluent des instructions spéciales pour faciliter le contrôle de flux, telles que les boucles optimisées pour les traitements répétitifs. Cela améliore l'efficacité de l'exécution des algorithmes complexes.
- **Interfaces d'E/S spécialisées** : Les DSP sont équipés d'interfaces spécifiques pour faciliter la communication avec d'autres composants du système. Cela peut inclure des interfaces série telles que I2C, SPI, ou des interfaces parallèles adaptées aux besoins des applications visées.
- **Optimisation de la consommation d'énergie** : Les DSP sont conçus pour maintenir un équilibre entre puissance de calcul et consommation d'énergie. Certains DSP peuvent ajuster dynamiquement leur fréquence d'horloge ou désactiver des parties non utilisées pour économiser de l'énergie lorsqu'ils ne sont pas sollicités.
- **Traitement en virgule fixe ou en virgule flottante** : Les DSP peuvent être configurés pour effectuer des calculs en virgule fixe ou en virgule flottante en fonction des exigences de l'application. Le choix dépend souvent de la précision requise pour le traitement du signal.

III) Préparation du terrain

1) Applications possibles pour le Dspic

Il est possible d'utiliser des dsPIC pour les applications suivantes :

- Thermocouple : Acquisition de mesures de température sur un système tournant
- Génération de signaux pour allumer ou éteindre des dispositifs domotiques
- Commande d'une machine à réluctance variable linéaire
- Commande d'un moteur à courant continu
- Accéléromètre : Acquisition des composantes d'un accéléromètre
- Affichage de données (heure, température ambiante, ...) sur un écran LCD
- Détecteur de lumière, de son ou de mouvement : Actionne un dispositif lorsqu'il détecte la donnée souhaitée

- AC/DC : Montage pour transformer du courant alternatif en courant continu
- Jeu de buzzer avec plusieurs joueurs : Dispositifs à base de relais permettant de connaître lors d'un jeu le joueur qui répond en premier
- Modulation à largeur d'impulsion : Faire varier un signal (forme, fréquence, amplitude) d'un signal à la sortie d'une broche
- Interrupteur clap : Dispositif permettant d'allumer une ampoule (ou led) par claquements de mains

2) Le choix du projet

Pour ce projet, nous avons décidé de nous concentrer sur l'une des applications du dsPic, à savoir l'interrupteur clap. L'objectif du projet est d'allumer une LED RGB en effectuant simplement des claquements de mains (rouge si 1 claquement, vert si 2 claquements, bleu si 3 claquements)

3) Analyse des besoins spécifiques au projet

Pour mener à bien ce projet, notre système devra répondre à plusieurs sous-fonctions :

- 1) FC1 : Capter le signal sonore émis par le claquement des mains
- 2) FC2 : Traiter le signal afin de reconnaître parmi tous les sons, celui des claquements des mains, de les compter et choisir quelle couleur allumer.
- 3) FC3 : Allumer la LED

4) Etude des contraintes matérielles et logicielles

Pour répondre aux fonctions énoncées précédemment, nous aurons besoins :

➤ **Matériel Électronique :**

- Microcontrôleur dsPIC (FC2)
- LED RGB (FC3)
- Résistance pour limiter le courant traversant la LED (FC3)
- Microphone (FC1)

➤ **Outils de Développement :**

- MPLAB IDE (logiciel pour programmer le dsPIC)
- MPLAB IPE (logiciel pour télécharger le programme sur le dsPIC)

➤ **Matériel support :**

- Plaque d'essai (Breadboard) et fils de connexion (Prototypage)
- Alimentation électrique

➤ **Documentation**

- Datasheet du dsPIC
- Documentation du microphone
- Manuels MPLAB et XC16 (ou XC32) compiler.

IV) Choix du Microcontrôleur et Environnement

1) Évaluation des microcontrôleurs disponibles

Nous avons à notre disposition trois microcontrôleurs :

- dsPIC33FJ128MC
- dsPIC33EP512MC
- dsPIC30F3013

Nous allons donner certaines caractéristiques de ces dsPIC sur lesquelles nous nous baserons pour faire le choix final de celui qui répond le plus à notre besoin.

Caractéristiques	dsPIC33FJ128MC	dsPIC33EP512	dsPIC30F3013
Taille	16-bits	16-bits	16-bits
Communication	CANbus I2C IrDA LIN bus SPI UART/USART	I2C IrDA LIN QEI SPI UART/USART	I2C SPI UART/USART
Vitesse	40 MIPS	70 MIPS	20 MIPS
Tension d'alimentation(Vcc)	3V ~ 3.6V	3 V ~ 3.6 V	2.5 V ~ 5.5 V

Figure 3: Comparaison des différents dspic

2) Sélection en fonction des exigences du projet

Nous utiliserons pour notre application le microphone Sound Sensor V1.6. Ses caractéristiques sont les suivantes :

Caractéristiques	Valeurs
Tension d'alimentation (Vcc)	5V
Plage de tension de fonctionnement	3,3~5V
Gain de tension (V=6V, f=1kHz)	26dB
Sensibilité (1kHz)	-60~-56 dBV/Pa (52-48dB)
Plage de fréquence	16-20kHz

Figure 4: Caractéristiques du microphone

Au vu de la tension d'alimentation de notre microphone, le microcontrôleur compatible avec notre application est le DSPIC30F3013. Seulement, après des essais de connexion, nous avons constaté que nous n'arrivions pas à communiquer avec lui depuis le logiciel MPLAB.

Nous avons finalement opté pour le DSPIC33FJ128MC. Et afin de pourvoir à l'alimentation du microphone en 5V, nous utilisons une alimentation externe.

Partie 2

PHASE OPÉRATIONNELLE DU PROJET : De la Création à l'Implémentation

I) Création du projet

1) Création d'un nouveau projet dans l'IDE

Le projet a été créé. Nous avons effectué un test de bon fonctionnement en utilisant un programme qui permet l'allumage et l'extinction de la led intégré au DSPIC toutes les secondes. Celui-ci a été validé.

2) Configuration initiale du projet (fréquence d'horloge, broches d'E/S, etc)

Pour notre projet, nous avons en entrée le signal du microphone et en sortie le signal de commande de trois LEDs. Ainsi, nous avons câblé comme suit :

- Anode de la LED rouge -> Broche 4 (RB0)
- Cathode de la LED rouge -> Broche 8 (GND)
- Sortie du microphone -> Broche 3 (AN1)
- Vdd du microphone -> Broche + de l'alimentation 5V
- GND du microphone -> Broche - de l'alimentation 5V
- Broche 8 (GND) -> Broche - de l'alimentation 5V (Masses communes)

II) Introduction aux Outils dsPIC

1) Présentation des outils DSP disponibles dans l'IDE

Dans l'IDE MPLAB, on peut trouver plusieurs outils DSP pour le développement des projets. Parmi ces outils les principaux sont les suivants:

- Bibliothèques de fonctions DSP
- Outils de simulation DSP
- Outils d'analyse spectrale
- Autres fonctionnalités liées au traitement du signal numérique

2) Utilisation des bibliothèques DSP pré intégrées

Nous utiliserons les bibliothèques :

- xc.h : Elle permet d'accéder aux registres spécifiques du microcontrôleur et de tirer parti des fonctionnalités matérielles.
- stdio.h et stdlib.h : Ces bibliothèques fournissent des fonctionnalités liées à l'entrée/sortie standard et à la gestion de la mémoire, respectivement.

3) Codage sous le logiciel MPLAB

Pour cette partie, nous nous sommes inspirés du code ADC donné par le professeur et nous l'avons modifié pour l'adapter aux besoins du projet. A un clap de main si la valeur du son produit et capté par le microphone est convertit , cette valeur convertie est comparée à un seuil si elle est supérieure la LED s'allume.

```
void PinA1OUT(void)
{
    TRISAbits.TRISA1 = 0;
    LATAbits.LATA1 = 1;
}

void PinBOUT(int ADCValue)
{
    TRISBbits.TRISB0 = 0;

    if(ADCValue > 100){
        LATBbits.LATB0 = 1;
    } else LATBbits.LATB0 = 0;
}

int main(int argc, char** argv) {
    unsigned int ADCValue;
    initAdc1();
    PinA1OUT();
    while(1){
        while (!AD1CON1bits.DONE); // Wait for the conversion to complete
        ADCValue = ADC1BUF0; // Read the conversion result
        PinBOUT(ADCValue);
    }
    return (EXIT_SUCCESS);
}
```

Figure 5:Une partie du code pour l'interrupteur clap de main

Partie 3

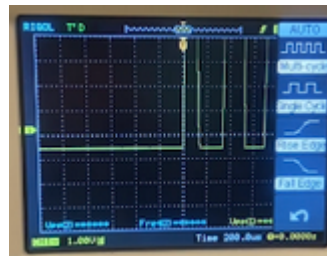
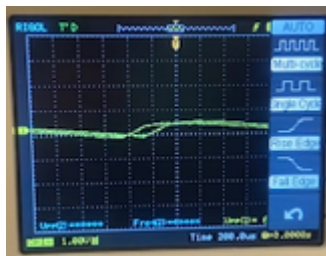
PHASE FINALE DU PROJET : Tests, validation, et communication des résultats

I) Tests et validation

- ☑ Allumage d'une LED rouge : Nous avons tout d'abord tenter d'allumer une led rouge afin de nous assurer que tout était en ordre. Le code est le suivant :

```
void main()
{
    TRISBbits.TRISB0 = 0;
    LATBbits.LATB0 = 1;
}
```

- ☑ Lecture des signaux du microphone : Afin de nous assurer que le microphone capte bien les sons des claps, nous avons visualiser son signal de sortie sur un oscilloscope. Nous pouvons observer la hausse de tension à la sortie lorsque l'on claque des mains.



- ☑ Allumage de la LED rouge par clap : Maintenant que nous sommes certains que la led s'allume bien lorsqu'on lui en donne l'ordre et que les informations du microphone sont bien transmise au dsPIC, nous écrivons le code de telle façon à contrôler l'allumage de la led par clap. Nous avons mis un premier un premier seuil à 100 ce qui correspond plus ou moins à 60 dB.

```
void initAdc1(void)
{
    //*****
    AD1CON1bits.AD12B = 0;
    AD1CON2bits.CHPS = 0;
    AD1CON2bits.ALTS = 0;

    AD1PCFGL = 0xFFFF;
    AD1CHS0bits.CH0SA = 0;
    AD1CON2bits.CSCNA = 0;

    AD1CHS0bits.CH0NA = 0;

    AD1CON1bits.SIMSAM = 0;
    AD1CON1bits.ASAM = 1;

    AD1CON1bits.SSRC = 7;

    AD1CON3bits.SAMC = 1;
    AD1CON3bits.ADRS = 0;

    AD1CON3bits.ADCS = 2;

    AD1CON2bits.VCFG = 0;
    AD1CON2bits.BUFM = 0;
    AD1CON1bits.FORM = 0;
    AD1CON1bits.ADON = 1;
}
```

```

void PinA1OUT(void)
{
    TRISAbits.TRISA1 = 0;
    LATAbits.LATA1 = 1;
}

void PinBOUT(int ADCValue)
{
    TRISBbits.TRISB0 = 0;
    if(ADCValue > 100){
        LATBbits.LATB0 = 1;
    } else LATBbits.LATB0 = 0;
}

int main(int argc, char** argv) {
    unsigned int ADCValue;
    initAdcl();
    PinA1OUT();
    while(1){
        while (!AD1CON1bits.DONE); // Wait for the conversion to complete
        ADCValue = ADC1BUF0; // Read the conversion result
        PinBOUT(ADCValue);
    }
    return (EXIT_SUCCESS);
}

```

- ☐ Allumage de LED RGB par l'intensité du clap : Nous avons ajouté au code deux autres seuils 150 et 200 respectivement pour que la led s'allume en vert ou bleu. Malheureusement, ce test n'a pas été validé puisque nous n'avons pas à notre disposition la led RGB.

```

void PinBOUT(int ADCValue)
{
    TRISBbits.TRISB0 = 0;
    TRISBbits.TRISB1 = 0;
    TRISBbits.TRISB2 = 0;
    if(ADCValue > 100){
        LATBbits.LATB0 = 1;
    } else LATBbits.LATB0 = 0;

    if(ADCValue > 150){
        LATBbits.LATB1 = 1;
    } else LATBbits.LATB1 = 0;
    if(ADCValue > 200){
        LATBbits.LATB2 = 1;
    } else LATBbits.LATB2 = 0;
}

```

- ☐ Allumage de LED RGB par nombre de claps : Nous n'avons pas pu par manque de temps arriver au bout de ce test.

II) Perspectives

Afin d'améliorer nos résultats, nous avons pensé à plusieurs perspectives :

- **Filtrage du bruit** : Intégrer des filtres ou des algorithmes de traitement du signal pour minimiser les déclenchements indésirables liés aux bruits non liés aux claps (fréquences indésirables), améliorant ainsi la précision de la détection du clap.
- **Sensibilité réglable** : Intégrer un mécanisme permettant à l'utilisateur de personnaliser la sensibilité du clap, adaptant ainsi le niveau de bruit requis et même le nombre de clap pour déclencher l'interrupteur.
- **Retard réglable** : Intégrer une fonction de retard réglable pour définir un laps de temps spécifique pendant lequel le clap doit être détecté, prévenant ainsi les activations accidentelles de l'interrupteur.

Conclusion

Les DSP et les microcontrôleurs DSPIC sont des composants essentiels pour le traitement numérique du signal et la mécatronique. Les DSP sont conçus pour manipuler et traiter des signaux tels que le son et la vidéo en temps réel, tandis que les microcontrôleurs DSPIC intègrent des fonctionnalités de DSP dans un microcontrôleur standard, offrant ainsi des performances élevées pour le traitement du signal numérique. Ils sont utilisés dans une variété d'applications, et le choix du microcontrôleur adapté est crucial pour assurer le bon fonctionnement du système. En résumé, ces composants offrent des capacités avancées pour le traitement des signaux en temps réel dans diverses applications.

Ce module nous a permis non seulement d'avoir une compréhension approfondie sur le DSP et les dspic, leurs domaines d'applications, leur utilité mais aussi de découvrir le logiciel de programmation MPLAB.

Annexes

Datasheet du dspic33FJ128MC802 :

<https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/70139G.pdf> (consulté le 01/02/2024)

DataSheet du microphone Sound Sensor V1.6 :

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/1285/Grove_Sound_Sensor_Web.pdf (consulté le 01/02/2024)

Caractéristiques des DSP :

<https://www.technologuepro.com/cours-dsp/chapitre-2-Caracteristiques-des-DSP.pdf> (consulté le 01/02/2024)

Architecture des dsPICs :

<https://skills.microchip.com/dsp-features-of-the-microchip-dspic-dsc/693195> (consulté le 01/02/2024)