



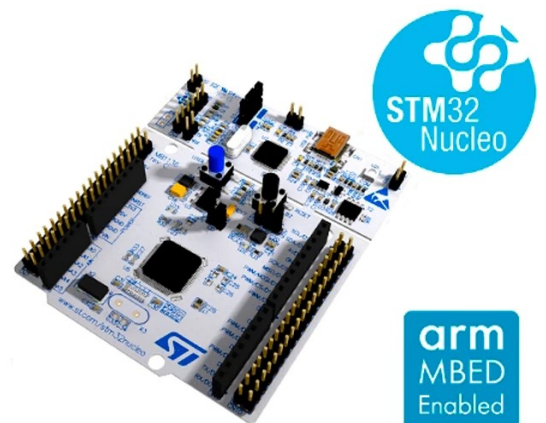
UNIVERSITÉ  
TOULOUSE III  
PAUL SABATIER



# [Rapport : BE Microcontrôleur STM32]

Master 1 EEA Systèmes et Microsystèmes embarqués

*Auteur : BA WAZIR AHMED  
MBOUNGOU FRANGLY*



15/05/2023

# Introduction :

La sécurité et la surveillance sont des préoccupations majeures pour de nombreux propriétaires de maison, d'entreprises et d'autres institutions. Les systèmes de surveillance traditionnels peuvent être coûteux et difficiles à mettre en œuvre, mais grâce aux avancées technologiques récentes, il est devenu plus facile et plus abordable de créer un système de surveillance simple mais efficace.

Dans ce compte rendu, nous allons présenter notre système de surveillance, le projet est constitué de deux cartes Nucleo-L476RF, un capteur Me PIR motion sensor, un module de buzzer, un module de communication radio APC220.

Nous avons fait notre projet en deux parties, dans une première partie le système est composé d'une seule carte Nucleo, le capteur et le buzzer, si le capteur détecte un mouvement alors le buzzer fait un son.

Ensuite pour améliorer le système nous avons utilisé le module APC220 pour faire une communication entre les deux cartes Nucleo où une carte est connectée avec le capteur et la partie transmission de APC220, et la deuxième carte est connectée avec la partie réception de APC220 et le buzzer, si le capteur de la première carte détecte un mouvement alors la partie émission envoie un signal à la partie réception qui ensuite active le buzzer pour faire un son. Nous allons examiner les différentes composantes du système, comment elles fonctionnent ensemble pour former un système de surveillance et les avantages et les limites du système.

# 1- Première Partie de système :

La première partie du système utilise la carte Nucleo, le capteur de mouvement PIR et le module de buzzer. Le but de cette partie est de détecter les mouvements et de faire un son d'alarme lorsque le capteur détecte un mouvement.

## Composants du système :

1- Une carte Nucleo-L476RF : les cartes Nucleo sont des microcontrôleurs programmables qui permettent de contrôler les différentes parties du système de surveillance.

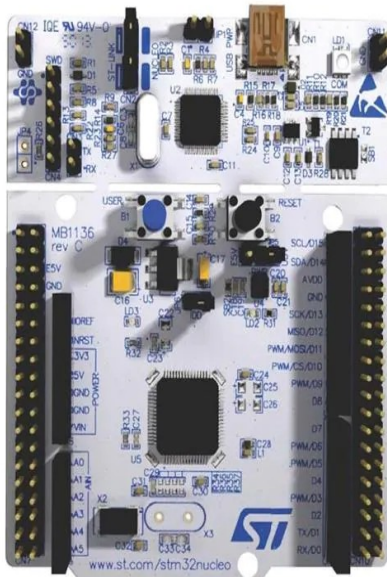
2- Capteur de mouvement PIR : Le capteur de mouvement PIR (Pyroélectrique Infrarouge) est un capteur passif qui détecte les mouvements en détectant les changements de température dans son champ de vision. Le capteur est équipé d'un détecteur de mouvement infrarouge passif qui est sensible aux changements de température. Lorsque quelque chose se déplace dans le champ de vision du capteur, la température augmente et le capteur détecte ce changement.

3- Module de buzzer : Le module de buzzer est un petit haut-parleur qui peut être utilisé pour émettre des sons et des alarmes. Il est facile à utiliser et peut être connecté directement à la carte Nucleo. Le module est activé par un signal de sortie de la carte Nucleo.

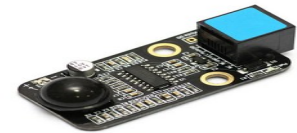
Le capteur de mouvement PIR est connecté à la carte Nucleo via une liaison numérique de 3 broches. La broche de sortie du capteur est connectée à une broche d'entrée (PA10) de la carte Nucleo. Le programme qui est exécuté sur la carte Nucleo lit l'état de la broche d'entrée du capteur de mouvement PIR. Lorsque le capteur détecte un mouvement, l'état de la broche de sortie change et la carte Nucleo détecte ce changement. Le programme active ensuite le module de buzzer (broche PA8) pour faire le son.

Le câblage est fait de la manière suivante :

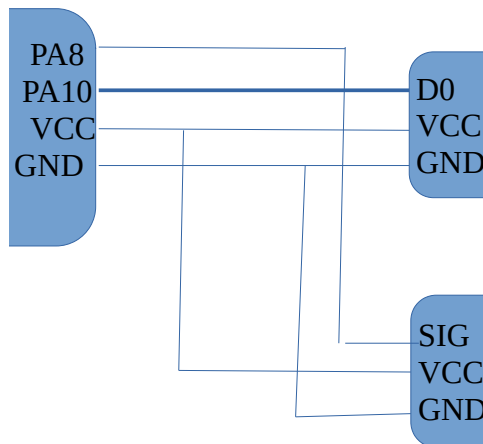
CARTE



Capteur Me PIR



Buzzer



Le système est alimenté par une batterie ou un adaptateur secteur, selon les préférences du propriétaire du système.

## Fonctionnement de Système avec le logiciel STM32 Cube IDE :

Pour avoir le bon fonctionnement de système , on a utilisé le logiciel STM32 Cube IDE pour faire la programmation de la carte , dans le programme qu'on a fait on a choisi PA10 comme entrée et qui est connecté avec la sortie de capteur , pour cela , on a écrit le code suivant :

```
/*Configure GPIO pin : PA10 */
GPIO_InitStruct.Pin = GPIO_PIN_10;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT; //mettre la broche PA10 en entrée
GPIO_InitStruct.Pull = GPIO_PULLUP;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

on obtient alors PA10 en entrée , cette entrée nous indique si le capteur détecte un mouvement ou non .

Pour le buzzer on a choisi PA8 comme sortie connecté avec l'entrée de buzzer , PA8 envoie un signal vers le buzzer pour faire activer le buzzer et donc avoir le son d'alarme .Voici le code fait concentrant PA8 :

```
/*Configure GPIO pins : LD2_Pin PA8 */
GPIO_InitStruct.Pin = LD2_Pin|GPIO_PIN_8; //choix de broche
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP; //mettre la broche PA8 en sortie
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

On a donc fait la bonne configuration de GPIO , maintenant on peut écrire la fonction **while** {...} dans laquelle on écrit le fonctionnement désiré de système , voici donc le contenu de la fonction :

```
while (1)
{
    /* USER CODE END WHILE */

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_10)==GPIO_PIN_SET){ //condition de détection de signal en PA10
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET); //la sortie PA8 est active
        HAL_Delay(800); // mettre une tempo
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET); //la sortie PA8 est désactivée
        HAL_Delay(300); // mettre une tempo
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET); //la sortie PA8 est active
        HAL_Delay(800); // mettre une tempo
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET); //la sortie PA8 est désactivée
        HAL_Delay(300); // mettre une tempo
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET); //la sortie PA8 est active
        HAL_Delay(800); // mettre une tempo
    }
    else { // si le capteur ne détecte rien alors le système n'active pas le buzzer
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET); //la sortie PA8 est désactivée
    }

    /* USER CODE BEGIN 3 */
}
```

## Avantages du système :

Le système de surveillance que nous avons présenté dans la première partie du projet présente plusieurs avantages :

1- Coût abordable : le système est relativement peu coûteux à mettre en œuvre, car il utilise des composants électroniques peu coûteux et facilement accessibles.

2- Facile à installer : le système est facile à installer et ne nécessite pas de connaissances approfondies en électronique. Il suffit de suivre les instructions d'installation fournies avec les différents composants du système. Simple à utiliser : le système est simple à utiliser et ne nécessite pas de formation particulière. Il suffit d'allumer le système et il commencera à surveiller la zone désignée.

3- Efficace : le système est efficace pour détecter les mouvements dans la zone surveillée et émettre un signal d'alarme. Cela peut aider à dissuader les intrus potentiels et à alerter les occupants du bâtiment en cas de danger.

## Limites du système :

Bien que le système de surveillance que nous avons présenté dans la première partie du projet soit efficace, il présente également certaines limites :

- Zone de surveillance limitée : le capteur de mouvement PIR a une portée limitée, ce qui signifie qu'il ne peut surveiller qu'une zone relativement restreinte.

Pour surveiller de plus grandes zones, il faudra utiliser plusieurs capteurs.

- Faux positifs et faux négatifs : le capteur de mouvement PIR peut parfois détecter des mouvements qui ne sont pas réellement présents, ce qui peut entraîner des faux positifs. De même, il peut parfois ne pas détecter des mouvements réels, ce qui peut entraîner des faux négatifs.
- Dépendance à l'alimentation électrique : le système de surveillance dépend de l'alimentation électrique pour fonctionner. Si l'alimentation électrique est interrompue, le système ne fonctionnera pas.

## Conclusion :

Dans cette première partie de notre projet de système de surveillance, nous avons présenté une solution simple mais efficace pour surveiller une zone désignée en utilisant un capteur de mouvement PIR, une carte Nucleo et un module de buzzer. Bien que le système présente certaines limites, il peut être utile pour les propriétaires de maison, les entreprises et d'autres institutions qui souhaitent surveiller leurs locaux à moindre coût et avec un minimum d'efforts. Dans la deuxième partie du projet, nous avons amélioré le système en utilisant un module de communication radio APC220 pour permettre la transmission de signaux entre deux cartes Nucleo.



# La deuxième partie de notre projet :

Dans la deuxième partie de notre projet de système de surveillance, nous avons tenté d'améliorer la solution présentée dans la première partie en utilisant un module de communication radio APC220 pour permettre la transmission de signaux entre deux cartes Nucleo.

## Module de communication APC220 :



Le module de communication RF APC220 DFRobot TEL0005 composé de 2 modules RF, 2 antennes et un convertisseur USB-TTL. Permet la communication entre 1 PC et une carte compatible Arduino ou deux cartes compatibles Arduino.

Le module communique avec le PC ou une carte compatible via une connexion série TTL.

Alimentation: 3,3 à 5,5 Vcc

Consommation: 25 à 35 mA

Puissance: réglable jusqu'à 20 mW (10 mW maxi en France)

Vitesse: 1200 à 19200 bauds

Fréquence: 431 à 478 MHz

T° de fonctionnement: -20 à -70 °C

Dimensions: 47 x 19 x 13 mm (chaque module sans antenne)

Référence DFRobot: TEL0005

Cette amélioration permet à l'utilisateur qui se situe loin de système de savoir s'il y a un mouvement détecté .Pour faire cette amélioration il faut savoir faire le bon câblage pour le APC220 :

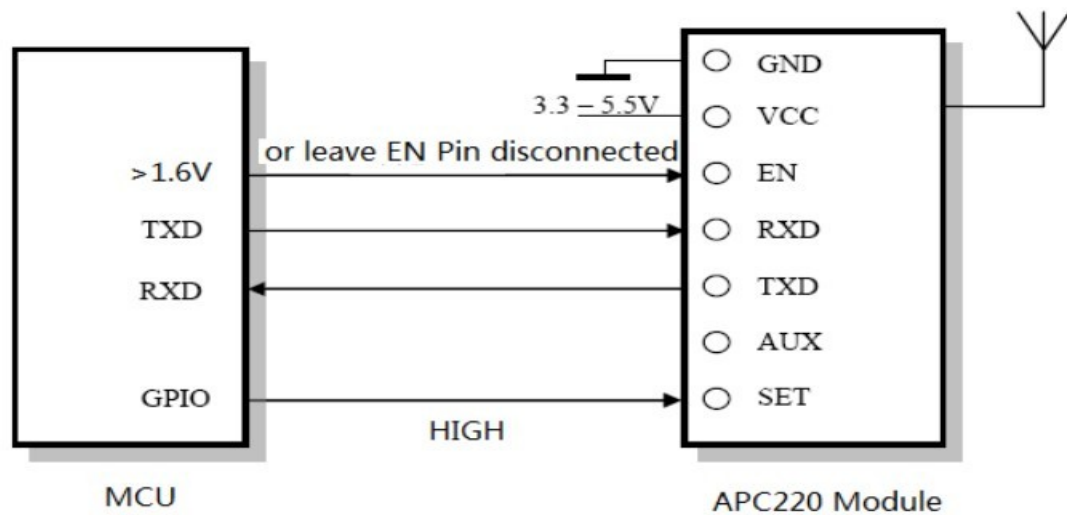
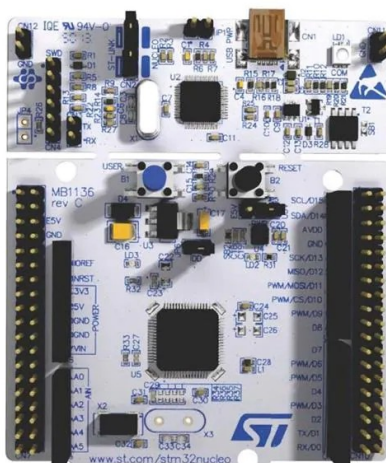


Figure: Connect APC220 to MCU

## 1-Câblage de la partie **émission** :

CARTE



Capteur Me PIR



PA8  
PA10  
VCC  
GND  
TXD

D0  
VCC  
GND

SET  
VCC  
GND  
RXD

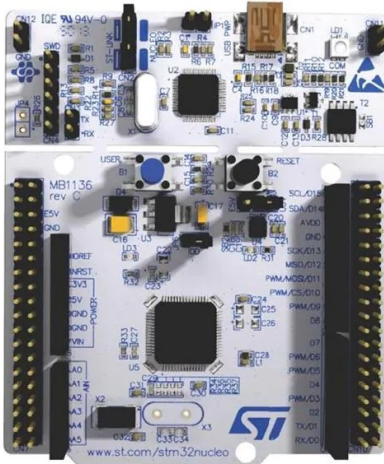


APC220 émission



## 2-Câblage de la partie **réception**:

CARTE



PA8  
PA9  
VCC  
GND  
RXD

SIG  
VCC  
GND

SET  
VCC  
GND  
TXD

Buzzer



APC220 réception

## Fonctionnement de Système avec le logiciel STM32 Cube IDE :

Comme dans cette partie de projet on a deux carte , une pour l'émission et l'autre pour la réception , alors on a fait deux programme , chaque programme est fait pour assurer le bon fonctionnement de sa carte avec sa partie émission/réception :

### 1-Programme de la partie émission :

Paramètres de transmission :

```
/* USER CODE END USART2_Init 1 */
huart2.Instance = USART2; //activation de USART2
huart2.Init.BaudRate = 9600; //paramètre de transmission
huart2.Init.WordLength = UART_WORDLENGTH_8B;
huart2.Init.StopBits = UART_STOPBITS_1;
huart2.Init.Parity = UART_PARITY_NONE;
huart2.Init.Mode = UART_MODE_TX_RX;
huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart2.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart2) != HAL_OK)
{
    Error_Handler();
}
```

code de la fonction While(1){...} :

```
while (1)
{
    /* USER CODE END WHILE */

    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_10)==GPIO_PIN_SET){//condition de détection sur PA10
        // Envoyer des données depuis NUCLEO 1 vers NUCLEO 2
        HAL_UART_Transmit(&huart2, txBuffer, sizeof(txBuffer), HAL_MAX_DELAY);//envoi de signal via APC220
    }

    /* USER CODE BEGIN 3 */
}
```

## 2-Partie réception :

Dans la carte de réception on a mis les mêmes paramètres de transmission , et la fonction while(1) {...} est la suivante :

```
while (1)
{
    /* USER CODE END WHILE */

    // Recevoir des données depuis NUCLEO 2 vers NUCLEO 1
    HAL_UART_Receive(&huart2, (uint8_t*)rxData, BUFFER_SIZE, 1000);//fonction de réception
    if (rxData[0] != 0)// si la réception de APC220 recoit un signal
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);//on active la sortie de buzzer
        HAL_Delay(800);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);///on désactive la sortie de buzzer
        HAL_Delay(300);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);//on active la sortie de buzzer
        HAL_Delay(800);
    }

    /* USER CODE BEGIN 3 */
}
```

Cependant, nous avons rencontré des difficultés lors de l'utilisation de ce module, ce qui nous a empêché d'achever cette partie du projet. Nous avons essayé de suivre les instructions du fabricant pour configurer le module et le faire fonctionner correctement, mais nous avons rencontré des problèmes lors de la communication entre les deux cartes Nucleo. Il est possible que les difficultés que nous avons rencontrées soient dues à un manque de connaissances techniques approfondies ou à un défaut du module lui-même. Cependant, nous avons réalisé que la mise en place de la communication sans

fil est un processus complexe qui nécessite des compétences et des connaissances spécifiques.

## **Conclusion :**

Dans l'ensemble, même si nous n'avons pas réussi à mettre en place la communication sans fil entre les deux cartes Nucleo à l'aide du module APC220, nous avons appris de précieuses leçons qui pourraient être utiles pour d'autres projets futurs. Nous avons également constaté que le développement d'un système de surveillance efficace nécessite des compétences techniques et des connaissances spécifiques, ainsi qu'un certain degré de patience et de persévérance.