

## РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ РУХУ НА БАЗІ STM32 З ВИКОРИСТАННЯМ PIR-ДАТЧИКА ТА ВИВЕДЕННЯМ ПОВІДОМЛЕННЯ НА OLED-ДИСПЛЕЙ SSD1306.

**Мета:** дослідити принцип роботи інфрачервоного датчика руху (PIR), реалізувати обробку сигналу через зовнішнє переривання (EXTI) мікроконтролера STM32.

### Теоретична інформація

PIR-сенсори призначені для фіксації руху шляхом аналізу змін інфрачервоного випромінювання в зоні спостереження. Вони широко застосовуються в охоронних системах, автоматичному освітленні, енергозберігаючих пристроях та системах «розумного дому». Такі датчики відзначаються компактністю, низькою вартістю, мінімальним енергоспоживанням і відсутністю механічних елементів, що забезпечує їх довговічність та стійкість до зносу.

PIR (Passive Infrared) датчики також називають піроелектричними або інфрачервоними датчиками руху, оскільки їх робота базується на піроелектричному ефекті — здатності певних матеріалів генерувати електричний заряд при зміні температури. Основним елементом сенсора є піроелектричний кристал, розташований у спеціальному корпусі, який реагує на зміну інтенсивності інфрачервоного випромінювання, що випромінюється об'єктами з температурою вище абсолютного нуля. Чим вища температура об'єкта, тим інтенсивніше його інфрачервоне випромінювання.

Конструктивно чутливий елемент поділений на дві симетричні зони. Це зроблено для того, щоб сенсор реагував не на абсолютний рівень випромінювання, а саме на його зміну. Коли теплий об'єкт (наприклад, людина) переміщується в межах зони чутливості, відбувається різниця сигналів між двома половинами елемента. Саме ця диференціальна зміна формує на виході датчика цифровий сигнал — логічний рівень High або Low.

Таким чином, PIR-датчик не «бачить» об'єкт безпосередньо, а реєструє динаміку зміни теплового фону, що дозволяє ефективно виявляти рух при мінімальному споживанні ресурсів системи.

### Характеристики модуля

- напруга живлення постійного струму: 4,5 - 20 В;
- струм на OUT: 60 мА;
- напруга на виході: високі і низькі рівні в 3, В TTL логіки;
- дистанція виявлення: 3 - 7 м (налаштовується);
- кут виявлення: від 120 ° до 140 °;
- тривалість імпульсу при виявленні: 5 - 300 сек (налаштовується);
- час блокування до наступного виміру: 2,5 сек. (але можна змінити заміною SMD-резисторів);
- робоча температура: від -20 до + 80 ° С;
- режими роботи: L - одиночне захоплення, H - повторювані вимірювання;
- габарити модуля: 32 x 24 x 18 мм.

Піроелектричний датчик руху складається з двох основних частин. Кожна з частин включає в себе спеціальний матеріал, чутливий до інфрачервоного випромінювання. В даному

випадку лінзи особливо не впливають на роботу датчика, так що ми бачимо дві ділянки чутливості всього модуля. Коли датчик знаходиться в стані спокою, обидва сенсори визначають однакову кількість випромінювання. Наприклад, це може бути випромінювання приміщення або навколишнього середовища на вулиці. Коли теплокровний об'єкт (людина або тварина), проходить повз, він перетинає зону чутливості першого сенсора, в результаті чого на модулі PIR датчика генеруються два різних значення випромінювання. Коли людина залишає зону чутливості першого сенсора, значення вирівнюються. Саме зміни в показах двох датчиків реєструються і генерують імпульси HIGH або LOW на виході.

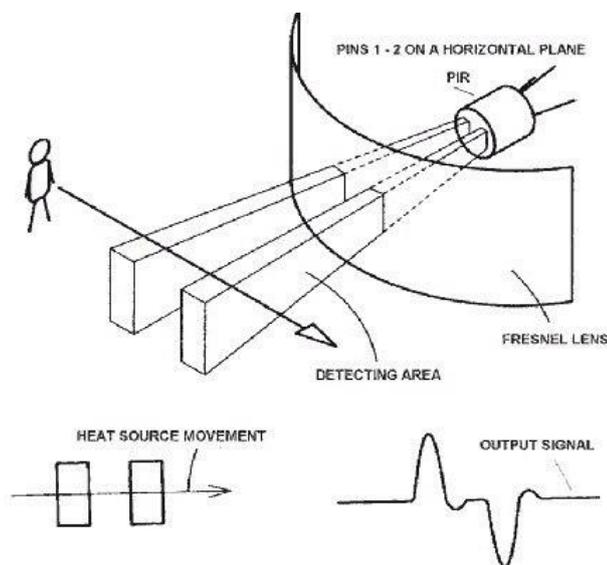


Рис.6.1 - Принцип дії PIR-сенсора

### Хід виконання роботи

1. Підключити датчик руху відповідно до розпіновки попередньо обравши на платі відповідний вільний пін для зчитування з інформаційного піна сенсора. Та увімкнути один з доступних світлодіодів. Налаштувати пін для світлодіоду на вихід GPIO\_Output, пін, що зчитує інформацію сенсора як GPIO\_EXTI1. Налаштувавши конфігурацію наступним чином

PA1 Configuration :

GPIO mode	External Interrupt Mode with Rising/Falling edge trigger detection
GPIO Pull-up/Pull-down	Pull-down
User Label	

1.1 Після чого увімкнути Sys->Debug->Serial Wire, RCC->High Speed Clock-> Cristal/Ceramic Resonator та в NVIC поставити галочку на дозволі зовнішніх переривань.

EXTI line1 interrupt	<input checked="" type="checkbox"/>	0	0
----------------------	-------------------------------------	---	---

Згенерувати код

2. Реалізувати програму відстеження руху. При виявленні сенсором руху, обраний світлодіод має загорітись. Активацію світлодіода реалізувати через зовнішні переривання.

2.1 У файлі main.c додати колбек функцію для переривання. В «х» підставивши власні піни та регістри до яких здійснено підключення відповідно до налаштувань :

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == GPIO_PIN_1)
    {
        GPIO_PinState s = HAL_GPIO_ReadPin(GPIOx, GPIO_PIN_x);

        if (s == GPIO_PIN_SET) {
            HAL_GPIO_WritePin(GPIOx, GPIO_PIN_x, GPIO_PIN_SET);
        } else {
            HAL_GPIO_WritePin(GPIOx, GPIO_PIN_x, GPIO_PIN_RESET);
        }
    }
}
```

Перевірити роботу пристрою.

3. Модернізувати попередній код та реалізувати симуляцію встановлення режиму охорони. Натискання кнопки переводить систему в «режим охорони» та у випадку виявлення руху запалюється індикатор. Допоки кнопка не натиснута система не в «режимі охорони». Натискання кнопки реалізовується через зовнішнє переривання.

3.1 Ініціювати змінні

```
// Прапорець активного режиму охорони
// 0 - система вимкнена
// 1 - система в режимі охорони
volatile uint8_t guard_mode = 0;

// Прапорець процесу активації (йде відлік часу перед включенням)
volatile uint8_t guard_arming = 0;

// Зберігає момент часу (в мс), коли було натиснуто кнопку
// Використовується для реалізації затримки перед активацією
```

3.2 В функції колбеку зовнішнього переривання дописати код:

```
if (GPIO_Pin == GPIO_PIN_1 && guard_mode)
{
    GPIO_PinState s = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1);

    if (s == GPIO_PIN_SET) {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
    } else {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
    }
}
if (GPIO_Pin == GPIO_PIN_0){
    guard_time = HAL_GetTick();
    guard_arming = 1;
}
```

3.3 В основну функцію додати

```
while (1)
{
    /* USER CODE END WHILE */
    if (guard_arming)
    {
        if (HAL_GetTick() - guard_time >= 5000)
        {
            guard_mode = 1;
            guard_arming = 0;
        }
    }
    /* USER CODE BEGIN 3 */
}
```

### **Індивідуальне завдання**

1. Підключити до системи OLED дисплей SSD1306 за допомогою SPI протоколу, налаштувати конфігурацію та реалізувати інформування користувача про стан системи. Вивід інформацію про поточний режим системи. (Не в режимі охорони, перевід в режим охорони та сам режим охорони, сповіщення про рух).

### **Контрольні запитання**

1. Чому не рекомендується використовувати HAL\_Delay() у callback переривання?
2. Що таке PIR-датчик і на якому фізичному принципі він працює?
3. Яку роль виконує піроелектричний елемент у датчику?
4. Які фактори можуть викликати хибне спрацювання датчика?