

# Taller de Programación y Robótica en CMM BML 2023T2 – CL24 – PIR Alarm

[R-mP] email Alarma de presencia (PIR) – 90'

1. Planteamiento del mProyecto – 15'
2. [HW] – PIR y hw test PIR con interrupciones -15'
3. Alarma PIR operática Buzer y reset (sin conex.)-20'
4. Enviar un correo con PICO W x IFTTT
  1. Opciones de diseño -5'
  2. Conocer el servicio IFTTT -10'  
(ver IFTTT, Logarse y crear servicio)
5. Unir PIR y envio de webhook -> PICOW – 15'
6. Un paso + : añadir info al correo – 10'



Voluntario : J.C. Santamaria



# Clase 24.1 [R-mP] email Alarma de presencia (PIR)

## Planteamiento del mini Proyecto

### Objetivo:

**Hacer una alarma, que detecte presencia y envíe un e-mail + Aprender IoT básico**

### Decisiones

➤ **Sensor PIR + Servicio IoT IFTTT**

### Razones de las decisiones

- Simplicidad : tanto del sensor PIR como de IFTTT
- Coste: el sensor es muy barato + IFTTT gratis
- Tutoriales Disponibles: sunfounder
- Interes en conocer servicio IFTTT, para otros posibles proyectos

### **Pasos**

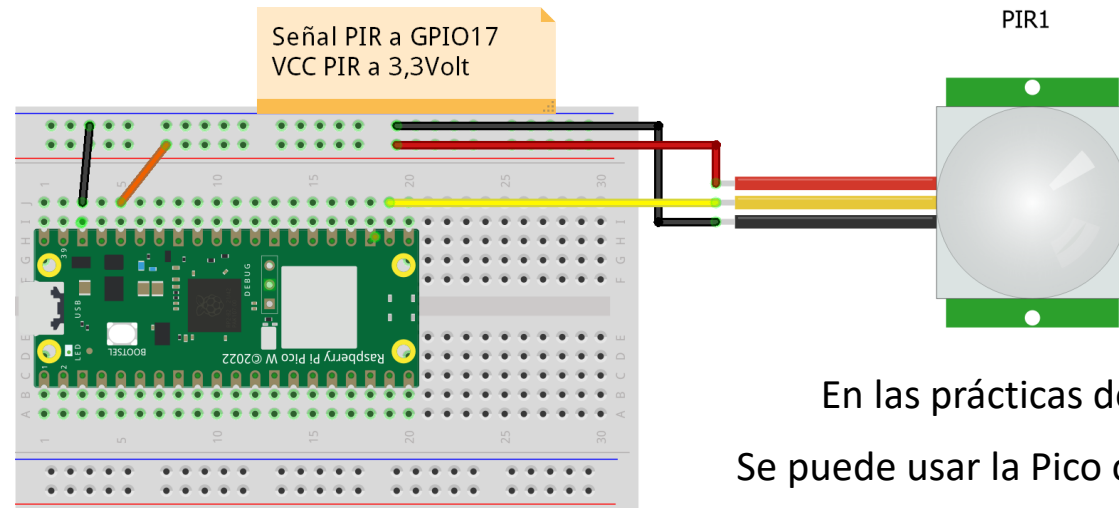
- Investigar proyectos similares
  - [Tutorial de sunfounder](#)
  - Otros tutoriales , también con ESP32
- Decisiones de diseño: Dividir en partes , hay 2 partes + sub -alternativas
  1. Detección de presencia
    - A. PIR
    - B. Radares GHz, ....
  2. Envío de email
    - A. Enviar el email desde PICOW
    - B. Usar un servicio que solo necesite un “disparo”
- 1.A - Conocer y hacer alarmas con PIR locales
- 2.B - Conocer y probar servicio IFTTT
- Montar PIR-alarma + disparo de IFTTT para enviar email
- Ir un paso + allá y To-dos



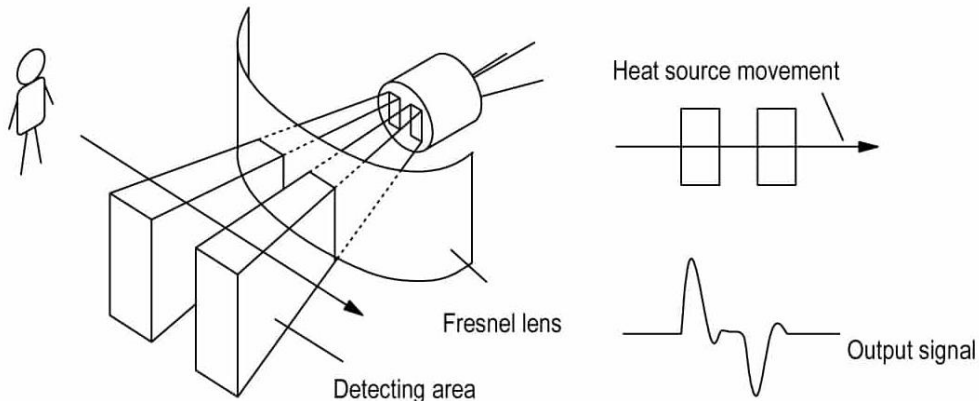
# Clase 24.2 [R-mP] email Alarma de presencia (PIR) [HW] – PIR y hw test PIR con interrupciones

## Sensor PIR (tuto sunfounder)

El sensor PIR es un doble sensor de infrarrojos. Lo que mide son diferencias de luz IR entre un sensor y el adyacente. Los objetos estáticos emisores de IR hacen que los 2 sensores reciban = cantidad de luz, pero si el objeto se mueve cada sensor “vera” diferente cantidad de luz IR. Un amplificador diferencial se encarga de aumentar esta diferencia, dando un valor DIGITAL



En las prácticas de CL24  
Se puede usar la Pico o la PICO W



En vez de estar comprobando el valor del pin de señal digital del PIR, continuamente para comprobar un cambio a '1', usaremos interrupciones para que el uC haga ese trabajo por nosotros.

Creamos una función para manejar la interrupción, y al crear la interrupción pasamos esta función como parámetro (las funciones también son objetos)

`BMMRCL24_pir_irq_bhwt_1_0.py`



## Clase 24.3 [R-mP] email Alarma de presencia (PIR)

### Alarma PIR operática Buzer y reset (sin conex)

BMMR\_CL24\_pir\_irq\_buzled\_but\_1\_3.py

Para “armar” una alarma operativa (o algo asi como un PoC) **necesitamos añadir** al circuito PIR de HW-test anterior:

**1- Buzzer para sonido si alarma**

**2- Led para luz si alarma**

**3- Botón para “resetear”**

Buzzer y Led pueden compartir PIN, asi que usaremos 3 pines en modo digital. El buzzer será de tipo activo por simplicidad => suena con “1” sin mas

Si se necesita +sonido, sustituir buzzer por altavoz con transistor

TO-DO : +display con mensajes

**PoC : proof of Concept**

# 0- Creamos 3 objetos PIN : PIR, botón, Buzzer + Inicializar variables

# F.1 Int-handler -Función si salta PIR => flag alarma **YES** & **reset interrup Alarma**

# F.2 Int-handler Función reset Alarma => flag alarma **NO** & **re-arma interrup Alarma**

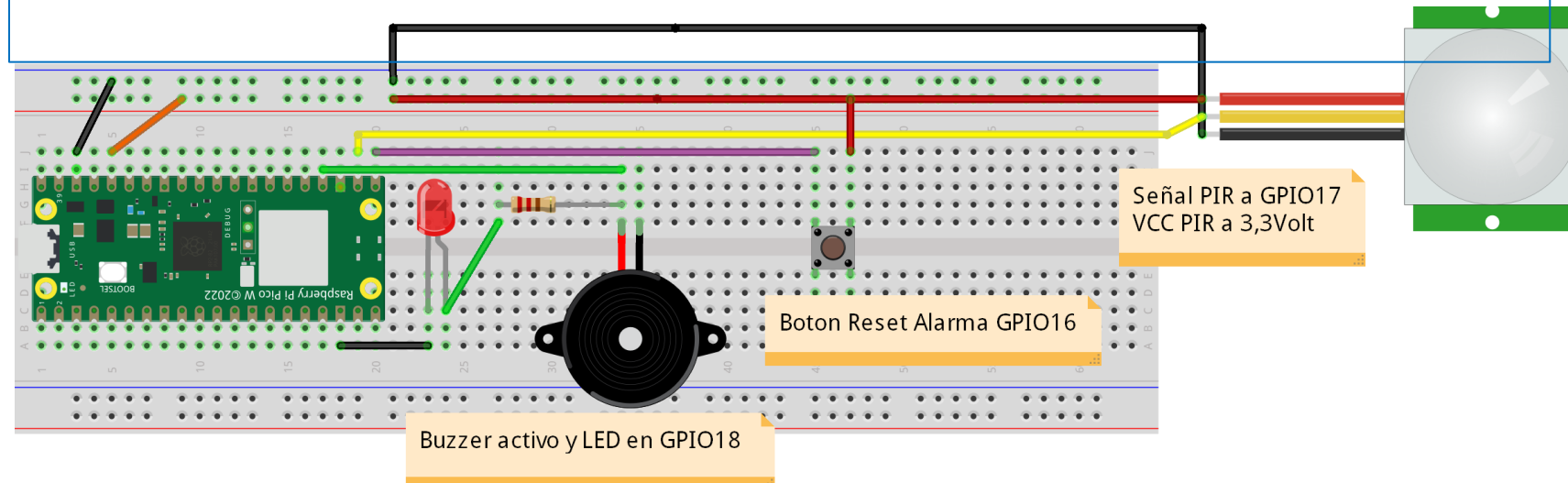
# 1 - Set interrupción PIR & Set interrupción Boton reset

# 2- Bucle infinito

# 2.1 Si flag Alarma YES => PIN buzzer = 1 => Sonar buzzer Lucir LED

# 2.1 Si flag Alarma NO => PIN buzzer = 0

**TRUCO**  
Maquina  
estados





## Clase 24.4 Enviar un correo con PICO W x IFTTT

Conocer el servicio IFTTT - > [Ver tutorial de Sunfounder](#)

### Opciones

- 3 opciones para enviar emails por eventos HW
  1. **Servicios IoT que ante valores limite envían e-mails** Ej Adafruit IO
  2. Hacer el envío del e-mail **directamente desde nuestro programa** en micropython con una libreria como [umail](#), [ver tutorial](#)
  3. **Usar un servicio como [IFTTT](#) que conecta eventos con acciones**

### Razones para escoger IFTTT

- **Simplicidad**
  - La opción 2) es compleja
  - La opción de Adafruit, ya la conocemos +/- , exploremos +cosas
- **Tutoriales Disponibles**
  - [Tutorial de Sunfounder alarma PIR](#)
- **Versatilidad** : si conoces IFTTT, puedes conectar otras acciones: enviar mensaje de Telegram, guardar en hoja de cálculo de Google, etc.



## Clase 24.4 Enviar un correo con PICO W x IFTTT

### Conocer el servicio IFTTT -2 - > [Ver tutorial de Sunfounder](#)

- **IFTTT** son las siglas de **IF This, Then That** que viene a significar **Si esto, entonces aquello**. La gracia está en que IFTTT conecta aplicaciones de todo tipo de forma sencilla, y el trigger puede ser un evento de nuestra PICO W enviado a través de HTTP (webhook)
- **Applets:** son un tipo de acciones concretas con las que puedes automatizar un proceso. **La cuenta gratuita dispone de 2**
- **Servicios:** es el lugar donde están todos los applets (acciones) que puedes tener de un mismo programa/ tipo. **Nosotros usaremos el servicio Webhooks que puede conectarnos con +750 aplicaciones**

1. Hay que crear una cuenta en IFTTT
2. Crear un Applet, usando como “if this” un webhook , con un Then email

[Ver tutorial de Sunfounder](#)

La parte complicada es ver la clave a usar en el url del GET de HTTP.

Ver clave : cuenta> my services > webhook > documentation



Your key is: **b6mIB0SPIZyvHmPV7SKN\_S**

◀ Back to service





# Clase 24.5 [R-mP] email Alarma de presencia (PIR) => Envio de correo

## Alarma PIR operática Buzer y reset (CON conexion)

BMMR\_CL24\_pir\_ifttt\_1\_1.py

Solo hay que hacer unos cambios muy pequeños al programa anterior de PoC :

PIR, buzzer/led + botón reset

En la función de Interrup – handler de alarma PIR hay que añadir el disparo del WebHook IFTTT, que es tan simple como hacer un **HTTP. GET**, a la **URL** que nos indica la documentación de la API de IFTTT, con el **nombre del webhook** ( creado por nosotros) y la **clave** asociada a nuestra cuenta

Your key is: b6mIB0SPIZyvHmPV7SKN\_S

**TO-DO : probar CURL , para testeo. Es una herramienta de Windows ( y otros) para probar comandos HTTP**

# 0- Creamos 3 objetos PIN : PIR, botón, Buzzer + inicializar variables + url, key IFTTT

### # 0.2 Conectarse a wifi

# F.1 Int-handler -Función si salta PIR => flag alarma **YES** & reset interrup Alarma

+ Disparar Webhook de IFTTT

# F.2 Int-handler Función reset Alarma => flag alarma **NO** & re-arma interrup Alarma

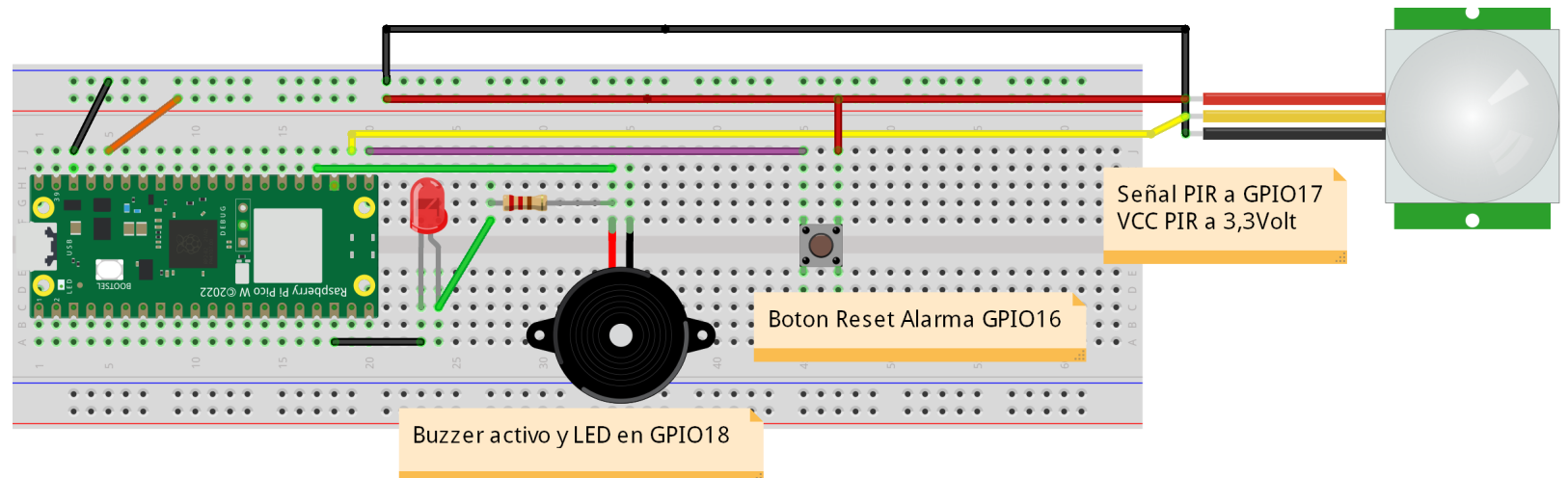
# 1 - Set interrupción PIR & Set interrupción Boton reset

# 2- Bucle infinito

# 2.1 Si flag Alarma YES => PIN buzzer = 1 => Sonar buzzer Lucir LED

# 2.1 Si flag Alarma NO => PIN buzzer = 0

**TRUCO**  
Maquina estados





# Clase 24.6 [R-mP] email Alarma de presencia (PIR) => Envio de correo

## Un paso + : añadir info al correo

BMMR\_CL24\_pir\_ifttt\_2\_0.py

Se puede explotar mejor el servicio IFTTT, enviando además del “disparo” del webhook hasta 3 valores de datos.

Vamos simplemente a enviar un valor fijo como prueba

TO-DO : enviar como dato la distancia al objeto -sensor HCSR04

# 0- Creamos 3 objetos PIN : PIR, botón, Buzzer + inicializar variables + url, key IFTTT

### # 0.2 Conectarse a wifi

# F.1 Int-handler -Función si salta PIR => flag alarma **YES** & reset interrup Alarma

+ Disparar Webhook de IFTTT + incluir valor de sensores

# F.2 Int-handler Función reset Alarma => flag alarma **NO** & re-arma interrup Alarma

# 1 - Set interrupción PIR & Set interrupción Boton reset

# 2- Bucle infinito

# 2.1 Si flag Alarma YES => PIN buzzer = 1 => Sonar buzzer Lucir LED

# 2.1 Si flag Alarma NO => PIN buzzer = 0

