# WiFi Reset Button Implementation

Using the BOOT button to disconnect ESP32 and clear the credentials

# Our Implementation

# Disconnecting WiFi and Clearing Credentials Using the BOOT Button on the DevKit

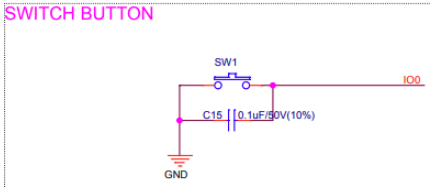
- About the Implementation
  - The BOOT button will be configured to generate an interrupt on IO0.
  - When the interrupt occurs, a message will be sent to the WiFi Application about the user request to disconnect/clear credentials.
  - Upon receiving the message, the WiFi Application will check if there really is an active connection prior to disconnecting and clearing credentials.

# Additional Information About the Implementation

GPIO Interrupt on the BOOT Button and Binary Semaphores

# GPIO Interrupt for WiFi Reset Button

- About
  - Background info: an interrupt is a signal that indicates the occurrence of a specific event that requires immediate attention and blocks the normal program execution to run an interrupt service routine (ISR) – which reacts to the event that occurred.
    - *Note: ISRs must have a short execution.*
  - The GPIO interrupt in our case, we are configuring the interrupt to trigger on a falling edge signal, because when you press the BOOT button, the pin connects to ground as shown here in the ESP32 DevKit-C schematic:




  - Due to the short execution time of ISRs; a binary semaphore will be used to notify a FreeRTOS task, which will handle the actions performed when the button is pressed.

# About Binary Semaphore APIs from ESP-IDF

- Binary Semaphore (from ISR)
  - Binary semaphores are semaphores which can assume the values of 0 and 1 only → Hence, they can be used as a signaling mechanism.
  - The ESP-IDF API for Semaphores can be found here → https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html#semaphore-api
  - Within the ISR, the API we need is → xSemaphoreGiveFromISR.
  - Within the FreeRTOS task, the API we need is → xSemaphoreTake.

# ESP-IDF APIs Used for the WiFi Reset Button GPIO Configuration

# Utilizing ESP-IDF for GPIO Interrupt Configuration

- Suggested Reading & About
  - About GPIO ESP-IDF → https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/gpio.html
  - Set the direction of the GPIO we are working with as an input → gpio_set_direction.
  - Set the interrupt type of the GPIO e.g., Falling Edge (GPIO_INTR_NEGEDGE) → gpio_set_intr_type.
  - Install the ISR service → gpio_install_isr_service.
  - Specify the ISR that will run when the interrupt for that pin is triggered → gpio_isr_handler_add.
  - Define the ISR and place it into the IRAM region → Using the IRAM_ATTR (attribute).

# Let’s get started!