

Raspberry Pi Pico တွင် Interrupt အားအသုံးပြုခြင်း

အလုပ်လုပ်နေစဉ်အတွင်း ကြားဖြတ်ဝင်ရောက်လာတဲ့ အလုပ် လုပ်လိုက်ရတာကို interrupt လို့ခေါ်ပါတယ်။ ဥပမာ - စာအုပ်ဖတ်နေရင်း ဖုန်းလာလို့ကိုင်လိုက်တယ်။ စာအုပ်ဖတ်တဲ့အလုပ်ကို လုပ်နေစဉ်အတွင်း ဖုန်းလာလို့ဖုန်းကိုင်လိုက်ရတာကို “Interrupt” လို့ခေါ်ပါတယ်။

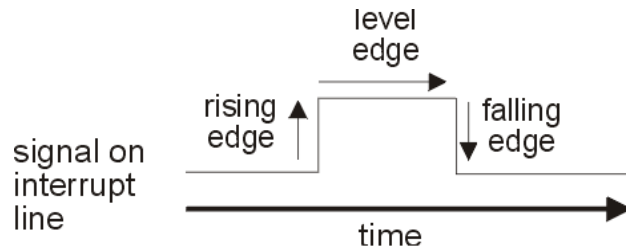
RP2040 ဟာ Dual-Core ARM processor ဖြစ်ပါတယ်။ core 0 ရော core 1 ရောက interrupt input (၃၂) ခုပါတဲ့ ARM ရဲ့ NVIC (Nested Vectored Interrupt Controller) နဲ့ချိတ်ဆက်ထား ပါတယ်။

IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source	IRQ	Interrupt Source
0	TIMER_IRQ_0	6	XIP_IRQ	12	DMA_IRQ_1	18	SPI0_IRQ	24	I2C1_IRQ
1	TIMER_IRQ_1	7	PIO0_IRQ_0	13	IO_IRQ_BANK0	19	SPI1_IRQ	25	RTC_IRQ
2	TIMER_IRQ_2	8	PIO0_IRQ_1	14	IO_IRQ_QSPI	20	UART0_IRQ		
3	TIMER_IRQ_3	9	PIO1_IRQ_0	15	SIO_IRQ_PROC0	21	UART1_IRQ		
4	PWM_IRQ_WRAP	10	PIO1_IRQ_1	16	SIO_IRQ_PROC1	22	ADC_IRQ_FIFO		
5	USBCTRL_IRQ	11	DMA_IRQ_0	17	CLOCKS_IRQ	23	I2C0_IRQ		

Raspberry Pi Pico မှာပါတဲ့ RP2040 မိုက်ခရိုကွန်ထရိုလာရဲ့ lower IRQ (၂၆) ခုဟာ NVIC နဲ့ချိတ်ဆက်ထားပါတယ်။ ကျန် IRQ ၂၆ ကနေ ၃၁ အထိကတော့ zero နဲ့ချိတ်ထားပြီး မသုံးပါဘူး။ Interrupt ကို Priority (ဘယ် interrupt က ပိုအရေးကြီးသလဲ) ခွဲတဲ့နေရာမှာ IRQ နံပါတ် အငယ်ဆုံး က Priority အမြင့်ဆုံးဖြစ်ပါတယ်။ IRQ0 က Priority အမြင့်ဆုံးပါ။

ဒီ ဥပမာတော့ External Hardware Interrupt ကို သုံးပြုမှာဖြစ်ပါတယ်။

External Interrupt ဆိုတဲ့နေရာမှာ ပြင်ပကနေ “Rising Edge” ဝင်လာရင် Interrupt ဖြစ်စေမှာလား၊ “Falling Edge” ဝင်လာရင် Interrupt ဖြစ်စေမှာလားဆိုပြီးတော့ (၂) မျိုးရှိပါတယ်။ Interrupt ဖြစ် လာရင် ပုံမှန်အလုပ်လုပ်နေတဲ့ ပရိုဂရမ်ကို ကြားဖြတ်ပြီးတော့ ISR လို့ခေါ်တဲ့ Interrupt Service Routine ထဲကို သွားအလုပ်လုပ်ပါတယ်။ Rising Edge ဆိုတာကတော့ “0” ကနေ “1” ပြောင်းလဲသွားတဲ့ အခြေအနေ digital wave “အတက်” ကိုပြောတာပါ။ Falling Edge ဆိုတာကတော့ “1” ကနေ “0” ပြောင်းလဲသွားတဲ့အခြေအနေ digital wave အကျ ကို ပြောတာဖြစ် ပါတယ်။ အောက်ပါပုံမှာကြည့်လိုက်ပါ။



GP15 မှာ External Interrupt ဝင်စေမည့် MicroPython ပရိုဂရမ်ရေးသားခြင်း

```
import machine
import utime

led = machine.Pin(0, machine.Pin.OUT)
ledBuiltIn = machine.Pin(25, machine.Pin.OUT)

led.value(0)
ledBuiltIn.value(0)

button = machine.Pin(15, machine.Pin.IN, machine.Pin.PULL_DOWN)

def int_handler(pin):
    button.irq(handler=None)
    print("Interrupt Detected")
    led.value(1)
    ledBuiltIn.value(0)
    utime.sleep(3)
    led.value(0)
    button.irq(handler=int_handler)

button.irq(trigger=machine.Pin.IRQ_RISING, handler=int_handler)

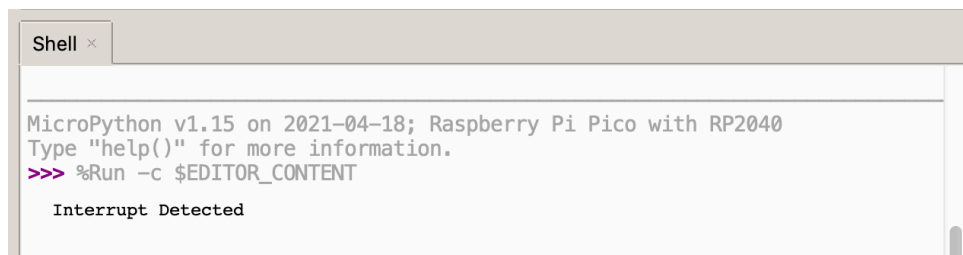
while True:
    ledBuiltIn.toggle()
    utime.sleep(1)
```

ပရိုဂရမ်ကိုကြည့်လိုက်မယ်ဆိုရင် GP15 မှာ External Interrupt ဝင်စေဖို့ရန်အတွက် Push button တစ်လုံးတပ်ထားပါတယ်။ Push button ကို Pull Down Resistor ခံပြီးတွေ့တပ်ထားတာပါ။ ဒါကြောင့် ခလုတ်နှိပ်လိုက်တဲ့အချိန်မှာ “1” ထွက်ပြီး ခလုတ်မနှိပ်ထားတဲ့အခြေအနေမှာ “0” ထွက်နေပါလိမ့်မယ်။ trigger=machine.Pin.IRQ_RISING လို့ရေးထားတဲ့အတွက်ကြောင့် ပြင်ပကနေ GP15 မှာ Rising Edge (“1”) ဝင်လာခါကျမှ Interrupt Request Handler ထဲကိုသွား အလုပ်လုပ်မှာဖြစ်ပါတယ်။

GP25 မှာတပ်ထားတဲ့ LED က ပုံမှန် ၁ စက္ကန့် delay နဲ့ပိတ်လိုက်ဖွင့်လိုက်ဖြစ်နေပါလိမ့်မယ်။ ပြင်ပကနေ GP15 ကို Rising Edge (“1”) ဝင်လာတဲ့အခါမှာ Interrupt ဖြစ်ပေါ် ပြီး Interrupt Handler (*int_handler(pin):*) ထဲမှာရေးထားတဲ့အတိုင်း GP0 မှာတပ်ထားတဲ့ LED ကို ၃ စက္ကန့်လင်းစေပါလိမ့်မယ်။ ပြီးတာနဲ့ ပြန်ပိတ်သွားပြီး ပုံမှန်ပရိုဂရမ်ဖြစ်တဲ့ GP25 မှာပါတဲ့ Raspberry Pi Pico ရဲ့ built-in led ကို ၁ စက္ကန့် delay နဲ့ ပြန်အလုပ်လုပ်စေမှာဖြစ်ပါတယ်။

ပရိုဂရမ်ရဲ့ Interrupt Handler (*def int_handler(pin):*) ထဲရေးထားတဲ့ ကုဒ်တွေကို တစ်ကြောင်း ချင်းစီ လေ့လာကြည့်ရအောင် -

- `button.irq(handler=None)`
ပြင်ပကနေ Interrupt ဝင်လာပြီးသည့်နောက် - Interrupt တွေကို ခဏပိတ်လိုက်ပါတယ်။
- `print("Interrupt Detected")`
“ပြီးတာနဲ့ “Interrupt Detected” ဆိုပြီးတော့ ThonnyIDE ရဲ့ Shell မှာ ထုတ်ပြလိုက်ပါ တယ်။



```
Shell x
MicroPython v1.15 on 2021-04-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Interrupt Detected
```

- `led.value(1)`
GP0 မှာတပ်ထားတဲ့ LED ကို ဖွင့်ပေးလိုက်ပါတယ်။
- `ledBuiltIn.value(0)`
အဲဒီနောက် GP25 မှာတပ်ထားတဲ့ ပုံမှန် blink နေသော Raspberry Pi Pico မှာ built-in ပါပြီးသား LED ကို ပိတ်လိုက်ပါတယ်။
- `utime.sleep(3)`
အချိန် (၃) စက္ကန့်လောက် ခဏနားလိုက်ပါတယ်။
- `led.value(0)`
Interrupt ဝင်လာကြောင်း Status ပြထားတဲ့ GP0 က LED ကိုပြန်ပိတ်လိုက်ပါတယ်။
- `button.irq(handler=int_handler)`
စောနက ပထမဦးဆုံး အဆင့်မှာပိတ်ထားတဲ့ Interrupt တွေအားလုံးကို ပြန်ဖွင့်ပေးလိုက်ပါတယ်။