



Journée 1 - TP 1

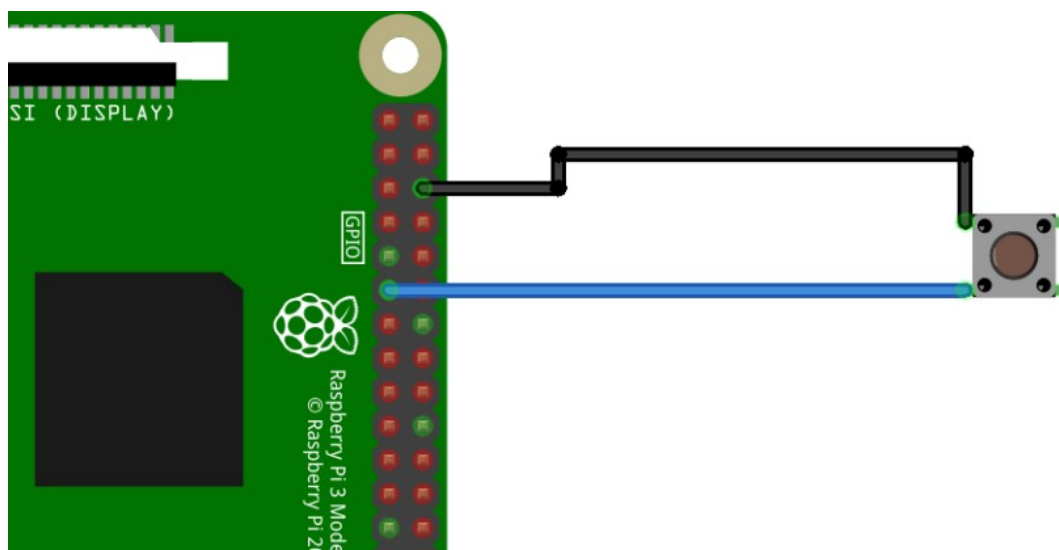
<i>BSP UBoot et Linux embarqué, construire son propre système</i>	<i>Date MAJ</i>	<i>21 août 2023</i>
<i>Chiheb Ameer ABID</i>	<i>Version</i>	<i>1.0</i>

Objectifs

- Configurer la production des interruptions à travers le DT
- Développer et configurer une fonction de gestion d'interruption (ISR)

I- Montage

On se propose de connecter un bouton poussoir à la carte RPi. L'objectif est de générer une interruption à l'appui du bouton. Pour simplifier, juste la routine ISR envoie un message au fichier journal. Le montage de la réalisation est illustré par la figure suivante :



II- Modification de l'arborescence de périphériques

Dans cette étape, on effectue des modifications dans l'arborescence de périphériques pour indiquer le numéro de l'IRQ et la broche GPIO dont la modification d'état provoquera une demande d'interruption. Ces modifications concernent le fichier **arch/arm/boot/dts/bcm2710-rpi-3-b.dts** et elles sont effectuées en deux étapes:

- La première étape concerne la spécification de la broche connectée au bouton poussoir : numéro de la broche, direction de la broche (entrée) et activation de la transition pull-up. Le nœud suivant doit être inséré dans le nœud gpio :

```
button_pin: key_pin {
```



```

    brcm,pins = <17>;
    brcm,function = <0>; /* Input */
    brcm,pull = <1>; /*Enable pullup */
};

```

- La deuxième étape concerne la configuration de ligne IRQ. Le nœud suivant est inséré dans le nœud SoC :

```

int_button {
    compatible = "training,intbutton";
    pinctrl-names = "default";
    pinctrl-0 = <&button_pin>;
    gpios = <&gpio 17 0>;
    interrupts = <17 1>;
    interrupt-parent = <&gpio>;
};

```

La propriété `gpios` précise quand la broche à utiliser (broche 17) et la configuration de cette broche :

- ```
#define GPIO_ACTIVE_HIGH 0
```
- ```
#define GPIO_ACTIVE_LOW 1
```

La propriété `interrupts` comporte un seul élément est constitué de deux cellules : le numéro de le numéro `hwirq` et le type de l'interruption générée :

```

#define IRQ_TYPE_NONE          0
#define IRQ_TYPE_EDGE_RISING   1
#define IRQ_TYPE_EDGE_FALLING  2
#define IRQ_TYPE_EDGE_BOTH     (IRQ_TYPE_EDGE_FALLING | IRQ_TYPE_EDGE_RISING)
#define IRQ_TYPE_LEVEL_HIGH    4
#define IRQ_TYPE_LEVEL_LOW     8

```

Une fois les descriptions nécessaires sont ajoutées dans le DT, il est nécessaire de re-compiler le noyau, et de transférer le fichier compilé `bcm2710-rpi-3-b.dtb` sur la RPi :

`arch/arm64/boot/dts/broadcom/bcm2710-rpi-3-b.dtb`.

Ce fichier doit être placé sur la carte RPi dans le répertoire `/boot/`. Il faut ensuite redémarrer la carte.

III- Implémenter une pilote de plateforme

Compléter le programme C `J1TP1.c` pour gérer les interruptions.

Vérifier en consultant le `/proc/interrupts` que les interruptions sont produites.

IV- Gérer plusieurs interruptions

- Connecter un deuxième bouton poussoir à la broche 23. Effectuer les modifications nécessaires pour gérer les interruptions produites par les broches 17 et 23 avec le même pilote de plateforme. On se propose de créer un seul périphérique misc dont le fichier spécial renvoie le nombre d'interruptions générées par chacune de deux broches.
- Connecter deux diodes LED. L'appuie sur un bouton poussoir permet de changer l'état d'une diode LED.