

GPIO PROJECT

GPIO → **Input Subsystem** projects lined up.

Let's break them down carefully so you understand the **flow across U-Boot** → **Kernel** → **Input subsystem** → `/dev/input/event0`.

◆ **Project 1: Enter Switch (Rising Edge Interrupt)**

🔴 **Goal**

- Port a push button (Enter switch) on **GPIO11**.
 - Configure it for **rising edge interrupt**.
 - Expose it as an **input device** → events available at `/dev/input/event0`.
-

1 **U-Boot Level (Troubleshooting stage)**

This ensures the **pinmux** and GPIO hardware are correct **before kernel boots**.

- **Step 1: Pinmux configuration**
 - Modify `board/ti/am335x/mux.c` (or relevant mux file in U-Boot).
 - Configure GPIO11's pad (example: `conf_gpmc_adX` or whatever pin corresponds).
 - Set mode = GPIO, input enabled, pull-up/down as per button circuit.
- **Step 2: Test GPIO from U-Boot prompt**
 - => `gpio status`
 - => `gpio input 11` # check if button press changes value
 - => `gpio set 11` # (if output mode test)

If this works, hardware and mux are good ✅.

2 **Kernel Level (Driver + DTS)**

- **Step 3: Device Tree changes**
In `arch/arm/boot/dts/km-bbb-am335x.dts`, add an entry under `gpio-keys`:
 - `gpio-keys {`
 - `compatible = "gpio-keys";`
 - `enter_key {`

- `label = "enter";`
- `linux,code = <28>; /* KEY_ENTER */`
- `gpios = <&gpio1 11 GPIO_ACTIVE_HIGH>;`
- `interrupt-parent = <&gpio1>;`
- `interrupts = <11 IRQ_TYPE_EDGE_RISING>;`
- `};`
- `};`

Here:

- `linux,code` = input event code (`KEY_ENTER` = 28).
- `IRQ_TYPE_EDGE_RISING` = rising edge interrupt.
- **Step 4: Kernel config**
Enable **GPIO keys driver**:
 - Device Drivers → Input device support → Keyboards → GPIO Keys

Set it as module (`gpio_keys.ko`).

Cross-compile, deploy, and load.

- **Step 5: Test in user space**
- `sudo evtest /dev/input/event0`

Press Enter switch → you should see events:

Event: time ..., type 1 (EV_KEY), code 28 (KEY_ENTER), value 1

Event: time ..., type 1 (EV_KEY), code 28 (KEY_ENTER), value 0

◆ Project 2: Up & Down Switches (Falling Edge Interrupt)

📌 Goal

- Add **two switches**: UP and DOWN.
- Trigger on **falling edge interrupts**.
- Still exposed via `/dev/input/event0`.

Kernel DTS entry

Extend the `gpio-keys` node:

```

gpio-keys {
    compatible = "gpio-keys";

    enter_key {
        label = "enter";

        linux,code = <28>;    /* KEY_ENTER */

        gpios = <&gpio1 11 GPIO_ACTIVE_HIGH>;

        interrupts = <11 IRQ_TYPE_EDGE_RISING>;
    };

    up_key {
        label = "up";

        linux,code = <103>;    /* KEY_UP */

        gpios = <&gpio2 5 GPIO_ACTIVE_LOW>;

        interrupts = <5 IRQ_TYPE_EDGE_FALLING>;
    };

    down_key {
        label = "down";

        linux,code = <108>;    /* KEY_DOWN */

        gpios = <&gpio2 6 GPIO_ACTIVE_LOW>;

        interrupts = <6 IRQ_TYPE_EDGE_FALLING>;
    };
};

```

- Codes are Linux input keycodes:
 - KEY_UP = 103
 - KEY_DOWN = 108
 - GPIO pins (gpio2_5, gpio2_6) are just examples — replace with your actual pins.
-

User space test

Run:

```
sudo evtest /dev/input/event0
```

Press UP or DOWN → you should see:

Event: time ..., type 1 (EV_KEY), code 103 (KEY_UP), value 1

Event: time ..., type 1 (EV_KEY), code 103 (KEY_UP), value 0

Event: time ..., type 1 (EV_KEY), code 108 (KEY_DOWN), value 1

Event: time ..., type 1 (EV_KEY), code 108 (KEY_DOWN), value 0

Git Commit Info (for your log)

- **Tag name:** gpioproj1
 - **Commit message:**
 - Enter Switch (GPIO11: Rising Edge Interrupt) porting on KM-BBB target board
 - **Tag name:** gpioproj2
 - **Commit message:**
 - UP & Down Switches (Falling Edge Interrupt) porting on KM-BBB target board
-

By the end, all switches (Enter, Up, Down) will be visible via /dev/input/event0 (through **input subsystem**), and you can capture them with evtest, cat /dev/input/event0, or even user-space applications.

Step	Action	Command(s)	Expected Output	Debug / Troubleshooting if Fail
1	Boot into U-Boot	=> prompt	U-Boot prompt appears	If board hangs → check serial console, SD/eMMC boot pins
2	Check GPIO mux	=> gpio status	Shows GPIO11 (or UP/DOWN pins) configured	If not → fix mux in board/ti/am335x/mux.c
3	Test GPIO input in U-Boot	=> gpio input 11	Value toggles (0/1) when button pressed	If always 0/1 → check schematic, pull-ups

Step	Action	Command(s)	Expected Output	Debug / Troubleshooting if Fail
4	Boot into Linux	Login prompt	System boots fully	If kernel panic → recheck DTS edits
5	Export GPIO (raw test)	echo 49 > /sys/class/gpio/export; cat gpio49/value	Value changes on press/release	If not → wrong GPIO number in DTS
6	Verify input driver loaded	`dmesg`	grep gpio-keys`	Logs like: input: gpio_keys as /devices/.../input/input0
7	Identify event node	ls -l /dev/input/	event0 or event1 shows up	If no eventX → check input subsystem config
8	Run event test	sudo evtest /dev/input/event0	Key codes: KEY_ENTER (28) or KEY_UP (103) / KEY_DOWN (108) appear	If no event → wrong linux,code in DTS
9	Press Enter switch	Physically press button	EV_KEY code 28, value 1 (press) + value 0 (release)	If missing → check DTS: rising edge config
10	Press UP/DOWN switches	Press buttons	EV_KEY code 103 / 108 appear	If inverted (press=release) → fix GPIO_ACTIVE_LOW/HIGH in DTS