AWaDH
Agriculture and Water
Technology Development Hub

INTERFACING HALL EFFECT SENSOR WITH DEV BOARD/NODE

What will you learn from this module:

Indication of magnetic field using Hall Effect sensor and Development Board/Node.

Requirements:

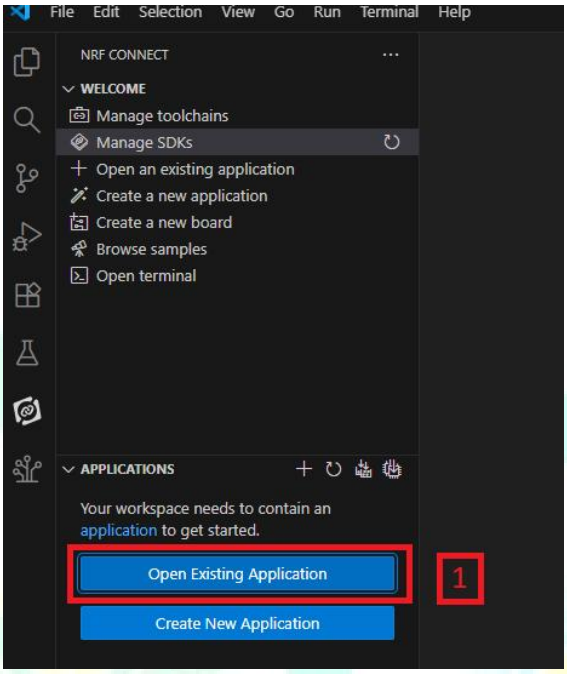
- nRF connect desktop software.
- nRF Command line tools.
- Visual studio code.
- USB cable.
- nRF52832 Development Board/Node.
- Hall Effect Sensor.

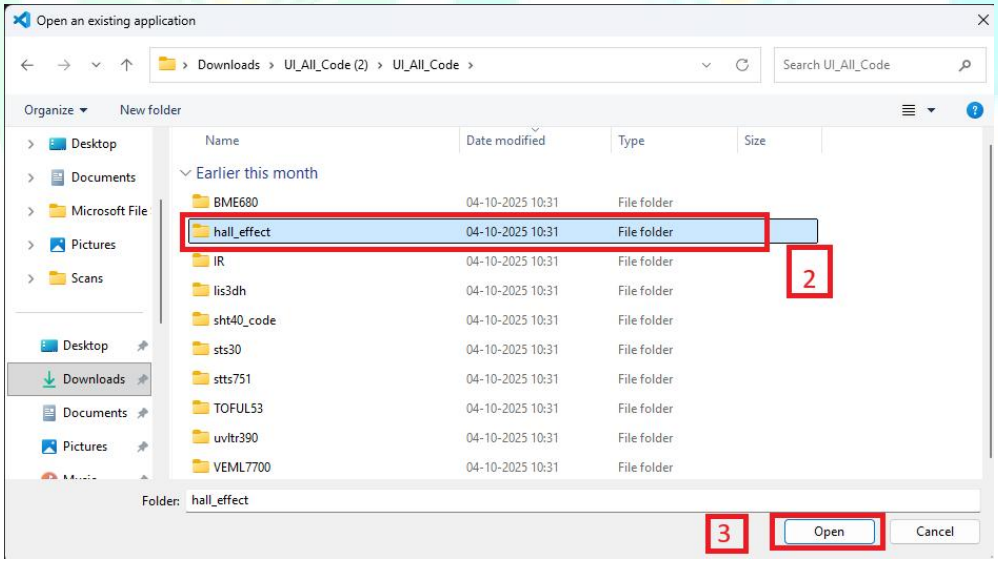
Prerequisites:

- Basic knowledge of C/C++
- Basic knowledge of communication protocol.
- Basic project setup.

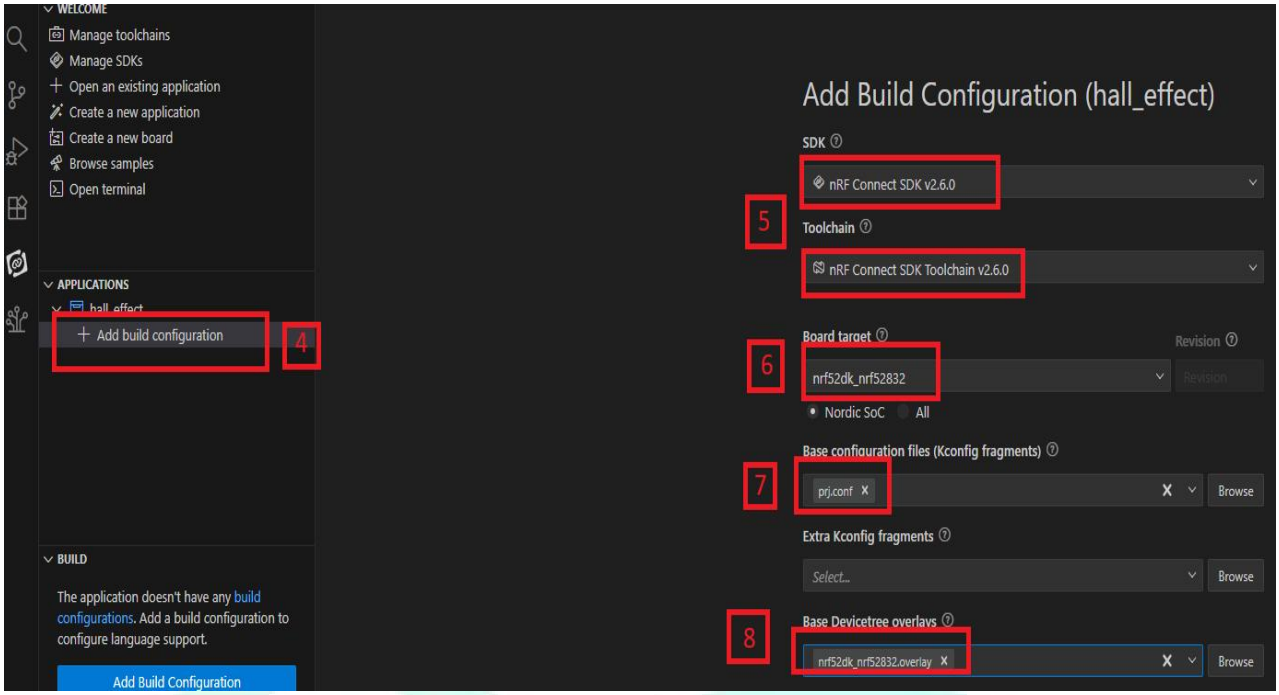
Setup and Configuration:

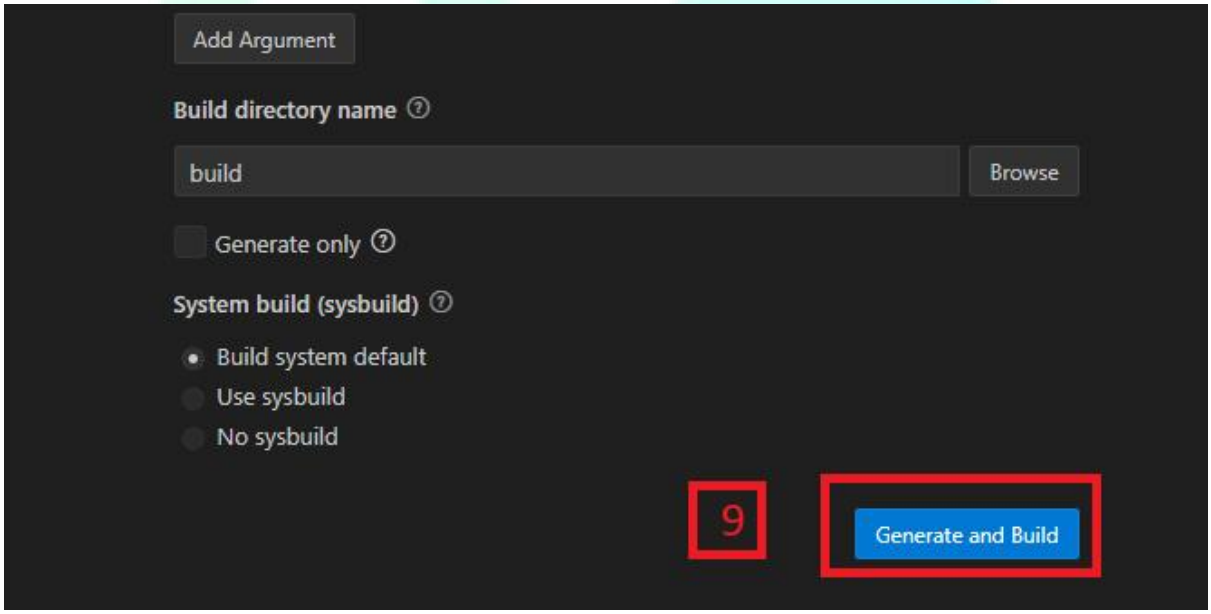
- Open VS Code and click on **Open Existing Application [1]** > click on **Hall effect [2]** > **Open [3]** as shown in the pictures below.



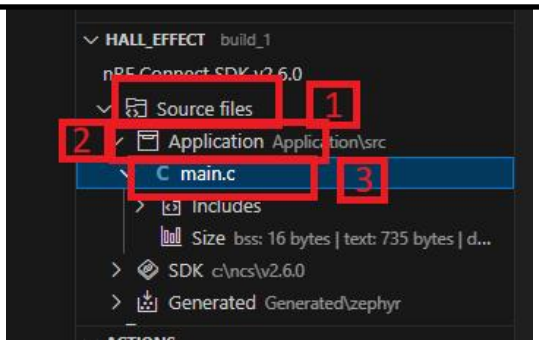


- Under nRf Connect, click on **Create new build configuration [4]**. Here, select **toolchain and SDK versions [5]** to be used. You can aso change the board version, if you are using nRF52832, then select **nrf52dk_nrf52832 [6]** or you can change from dropdown menu for another version like nRF52833 etc.
- Click on the Configuration and select **prj.conf [7]** from dropdown menu and then click on the devicetree overlay & select **nrf52dk_nrf52832.overlay [8]**.
- Then click on the **Generate and Build [9]** as shown below in the picture.





- Go to source file, click **source file [1]** > click on **Application [2]** > click on **main.c [3]**.
- By clicking on **main.c** file and you will see the code on your screen.

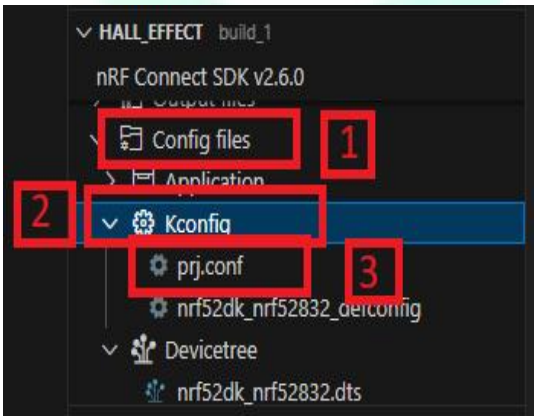


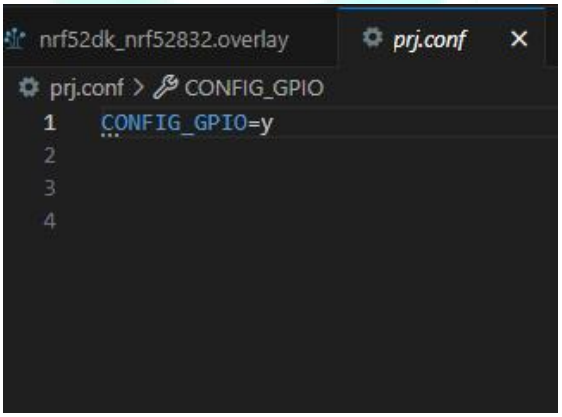
nrf52dk_nrf52832.overlay main.c

src > main.c > hall_sensor_cb

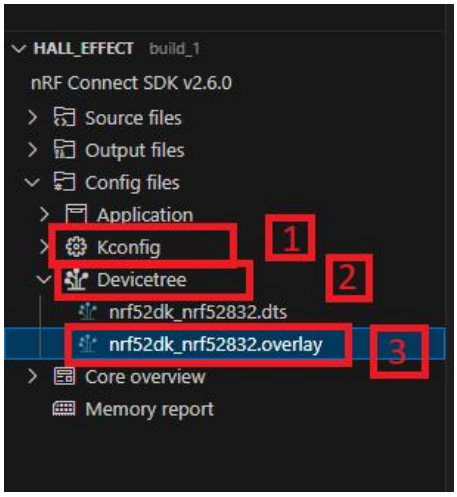
```c
#include <zephyr/kernel.h>
#include <zephyr/device.h>
#include <zephyr/drivers/gpio.h>
#include <zephyr/sys/printk.h>

#define SLEEP_TIME_MS 100  // Adjust as needed

/* Get Hall Effect Sensor configuration from devicetree */
#define HALL_SENSOR_NODE DT_NODELABEL(hall_sensor_0)

#if !DT_NODE_HAS_STATUS(HALL_SENSOR_NODE, okay)
#error "Unsupported board: hall_sensor devicetree alias is not defined"
#endif
static const struct gpio_dt_spec hall_sensor = GPIO_DT_SPEC_GET_OR(HALL_SENSOR_NODE, gpios, {0});
static struct gpio_callback hall_sensor_cb;

/* Optional LED configuration */
static const struct gpio_dt_spec led = GPIO_DT_SPEC_GET_OR(DT_ALIAS(led0), gpios, {0});
static int magnet_detected = 0; // 0 = No magnet, 1 = Magnet detected

/* Interrupt Handler for Hall Sensor */
void hall_sensor_triggered(const struct device *dev, struct gpio_callback *cb, uint32_t pins)
{
    magnet_detected = gpio_pin_get_dt(&hall_sensor); // Read sensor state
    printk("Magnet %s detected\n", magnet_detected ? "NOT" : "IS");

    if (led.port) {
        gpio_pin_set_dt(&led, !magnet_detected); // Turn LED on when magnet is detected
    }
}

int main(void)
{
    int ret;

    /* Check if Hall effect sensor is available */
    if (!gpio_is_ready_dt(&hall_sensor)) {
        printk("Error: Hall sensor device %s is not ready\n", hall_sensor.port->name);
        return 0;
    }

    /* Configure Hall effect sensor pin as input */
    ret = gpio_pin_configure_dt(&hall_sensor, GPIO_INPUT);
    if (ret != 0) {
        printk("Error %d: failed to configure %s pin %d\n", ret, hall_sensor.port->name, hall_sensor.pin);
        return 0;
    }
```

- To configure the prj configuration, click on **Config files [1]** > click on **Kconfig** [2] > click on **prj.conf [3]**.
- The prj configuration will appear on your screen as shown in the picture below.





- To configure the i2c protocol, you have to enable it in the **.overlay file**.
- Click on the **Config files** > click on **Kconfig [1]** > click on **Devicetree [2]** > click on **nrf52dk_nrf52832.overlay [3]**.
- The overlay file will appear on your screen and add the given code to the .overlay file as shown in the picture given below.

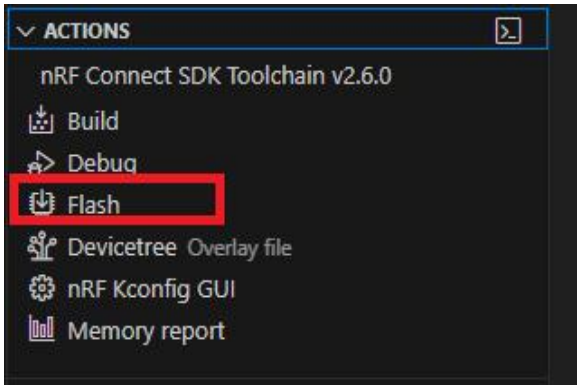


nrf52dk_nrf52832.overlay prj.conf

C:\Users\heeta\Downloads\hall_effect\hall_effect\nrf52dk_nrf52832.overlay

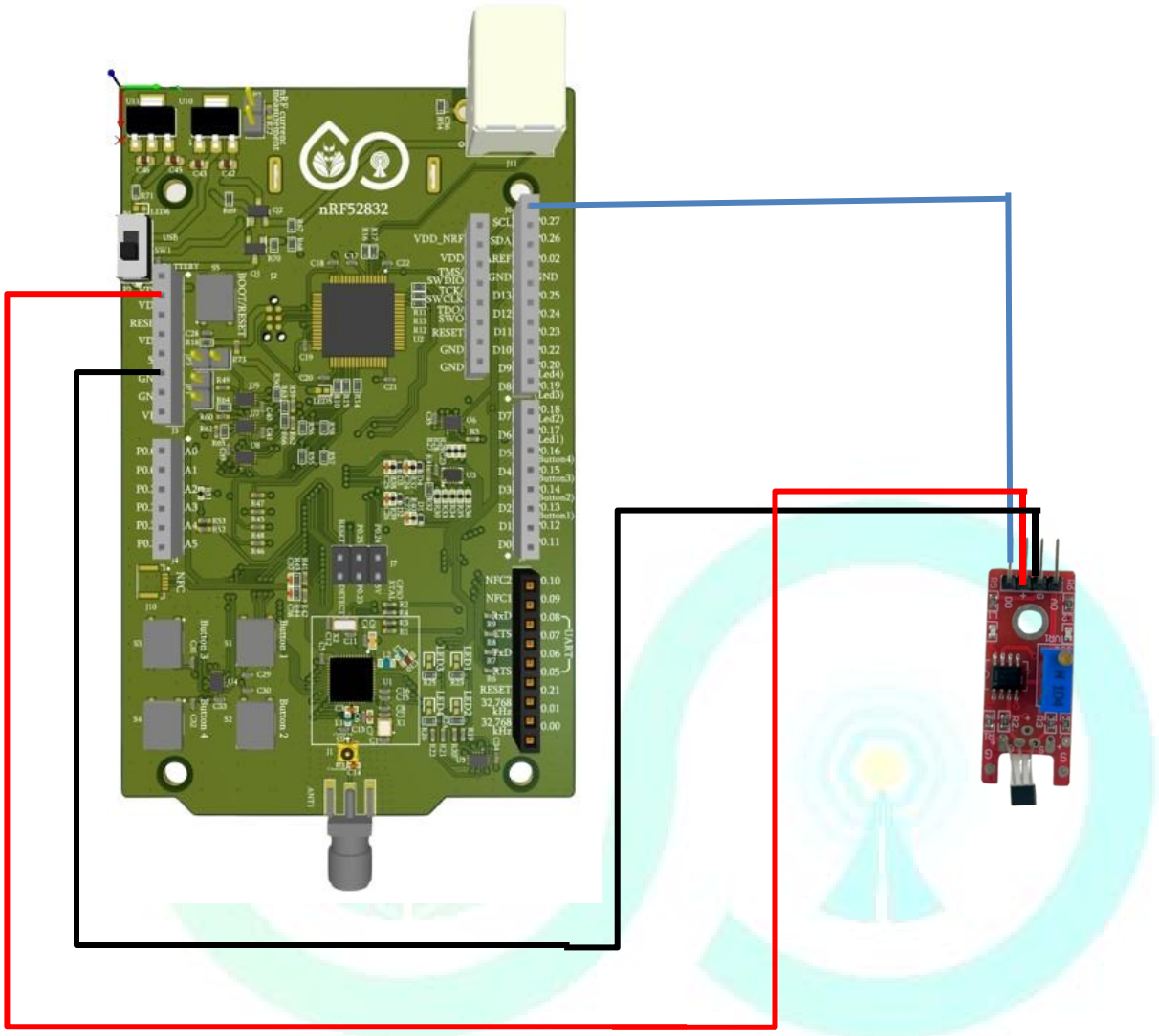
```
/ {
    hall_sensor {
        compatible = "gpio-keys";
        hall_sensor_0: hall_sensor_0 {
            gpios = <&gpio0 27 (GPIO_PULL_UP | GPIO_ACTIVE_LOW)>;
            label = "Hall Effect Sensor";
        };
    };
};



/ {
    model = "Nordic nRF52 DK NRF52832";
    compatible = "nordic,nrf52-dk-nrf52832";

    chosen {
        zephyr,console = &uart0;
        zephyr,shell-uart = &uart0;
        zephyr,uart-mcumgr = &uart0;
        zephyr,bt-mon-uart = &uart0;
        zephyr,bt-c2h-uart = &uart0;
        zephyr,sram = &sram0;
        zephyr,flash = &flash0;
        zephyr,code-partition = &slot0_partition;
    };

    leds {
        compatible = "gpio-leds";
        led0: led_0 {
            gpios = <&gpio0 17 GPIO_ACTIVE_LOW>;
            label = "Green LED 0";
        };
        led1: led_1 {
            gpios = <&gpio0 18 GPIO_ACTIVE_LOW>;
            label = "Green LED 1";
        };
        led2: led_2 {
            gpios = <&gpio0 19 GPIO_ACTIVE_LOW>;
            label = "Green LED 2";
        };
        led3: led_3 {
            gpios = <&gpio0 20 GPIO_ACTIVE_LOW>;
            label = "Green LED 3";
        };
    };
};
```

- In the overlay file, you can configure the gpio pins according to your board and sensor.
- Once you have made all the connections, including your nRF board and sensor, **flash** the code onto the board.



- If **flashed successfully** message is displayed on serial terminal, then flash process is complete.
- To check the output, go to **connected devices> select your board > select the COM** and view the output on serial monitor.

PIN CONFIGURATION



Board Pins -> Sensor Pins

VDD(3.3V) -> +

PO.27 -> DO

GND -> GND

OUTPUT

- To check the output, go to **connected devices> select your board > select the COM** and view the output on serial monitor.