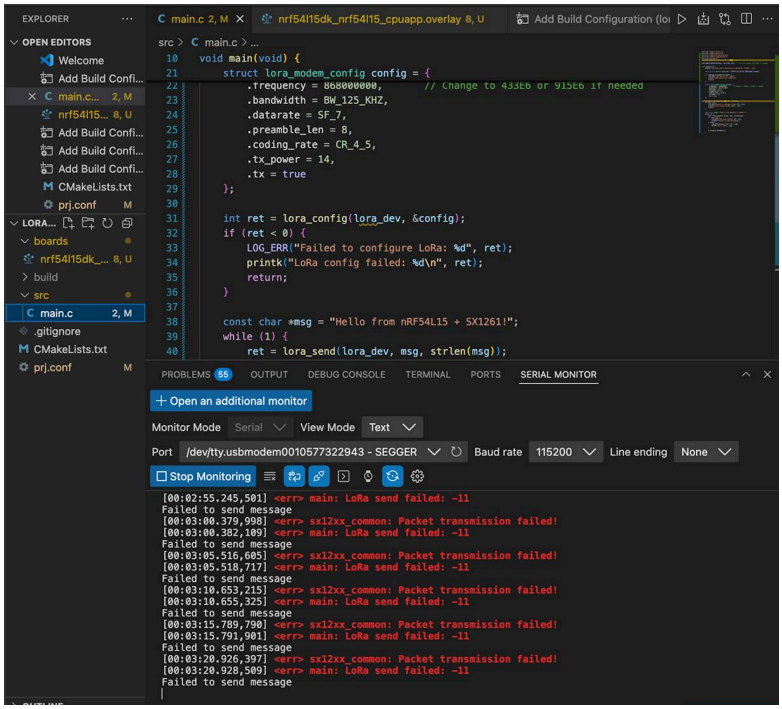
OUTPUT RECEIVING :



CODE FILE :

aanjaneyapandey@Mac lora_trx_app % tree
.
├── boards
│ └── nrf54l15dk_nrf54l15_cpuapp.overlay
├── CMakeLists.txt
├── prj.conf
└── src
└── main.c

nrf54l15dk_nrf54l15_cpuapp.overlay :

```
&spi00{
    status = "okay";
    cs-gpios = <&gpio2 5 GPIO_ACTIVE_LOW>; // P2.05 - CS

    lora0: sx1261@0 {
        compatible = "semtech,sx1261";
        reg = <0>; // CS index
        spi-max-frequency = <8000000>;

        reset-gpios = <&gpio2 6 GPIO_ACTIVE_LOW>;     // P2.06
        dio1-gpios = <&gpio1 15 GPIO_ACTIVE_HIGH>;     // P1.15
        busy-gpios = <&gpio1 11 GPIO_ACTIVE_HIGH>;     // P1.11
    };
};
```

Main.c :

```c
#include <zephyr/kernel.h>
#include <zephyr/device.h>
#include <zephyr/drivers/lora.h>
#include <zephyr/logging/log.h>
#include <string.h>

LOG_MODULE_REGISTER(main, LOG_LEVEL_DBG);  // Enables serial log over UART
/**
 * @brief Main function of the LoRa transmitter app.
 *
 * Initializes LoRa, sends periodic packets, and logs transmission status.
 *
 * @return int Always returns 0.
 */
int main(void) {
    const struct device *lora_dev = DEVICE_DT_GET(DT_NODELABEL(lora0));

    if (!device_is_ready(lora_dev)) {
        LOG_ERR("LoRa device not ready");
        return;
    }
    struct lora_modem_config config = {
        .frequency = 868000000,       // Use 433E6 / 915E6 if needed
        .bandwidth = BW_125_KHZ,
        .datarate = SF_7,
        .preamble_len = 8,
        .coding_rate = CR_4_5,
        .tx_power = 14,
        .tx = true
    };

    int ret = lora_config(lora_dev, &config);
    if (ret < 0) {
        LOG_ERR("Failed to configure LoRa: %d", ret);
        return;
    }

    LOG_INF("LoRa config done. Waiting before transmit...");
    k_sleep(K_MSEC(200));  // Give time to settle

    char msg[] = "Hello from nRF54L15 + SX1261!";
    while (1) {
        ret = lora_send(lora_dev, msg, strlen(msg));
        if (ret == -11) {
            LOG_WRN("SX1261 busy. Retrying...");
            k_sleep(K_MSEC(100));  // Wait before retry
            continue;
        } else if (ret < 0) {
            LOG_ERR("LoRa send failed: %d", ret);
        } else {
            LOG_INF("Message sent: %s", msg);
        }
        k_sleep(K_SECONDS(5));
    }
    return 0;
}
```

CMakeLists.txt :

```
cmake_minimum_required(VERSION 3.20.0)
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})

project(lora_trx_app)

target_sources(app PRIVATE src/main.c)
```

prj.conf :

```
# Core
CONFIG_GPIO=y
CONFIG_SPI=y

# LoRa
CONFIG_LORA=y
CONFIG_LORA_INIT_PRIORITY=90

# Serial + Logging
CONFIG_SERIAL=y
CONFIG_CONSOLE=y
CONFIG_UART_CONSOLE=y
CONFIG_LOG=y
CONFIG_LOG_MODE_DEFERRED=y
CONFIG_LOG_BACKEND_UART=y
CONFIG_LOG_DEFAULT_LEVEL=3
```

SDK : *nRF Connect SDK v2.9.1*

Toolchain : *nRF Connect SDK Toolchain v2.9.1*

Circuit Diagram and Pin connection :