STM32WL Lora Phy hands-on

Point to Point application

life.augmented

P2P

Key learning

- Touch a real RF application
- Proprietary protocol on the top of Lora modulation
- Sequencer approach of low-power application
- Channel Sensing

life.augmented

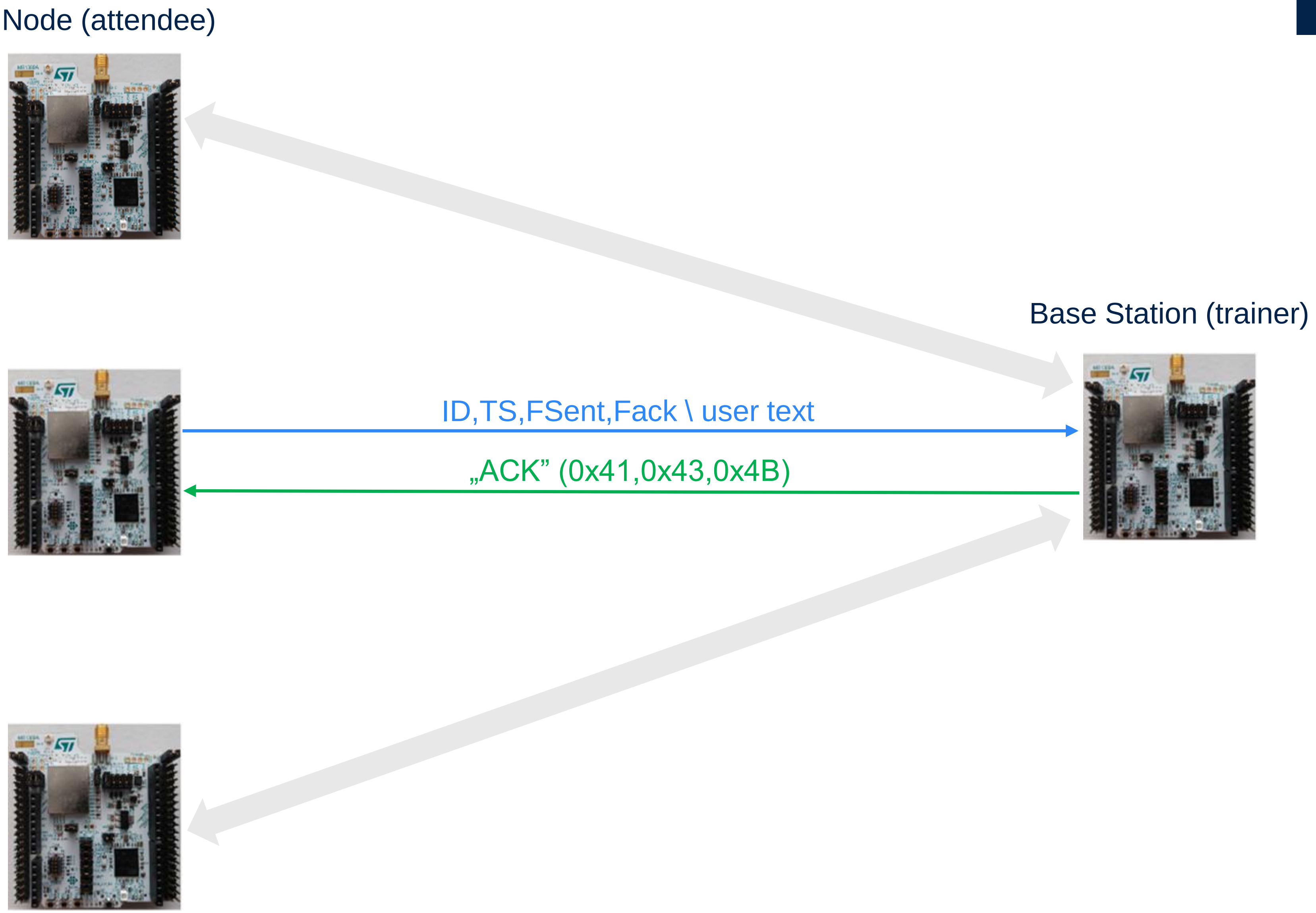
P2P: network

Node periodically sends, if RF channel is free, following data:

- Node ID: 1 byte,
- Temperature: 1 byte,
- Frame Sent Counter: 2 bytes,
- Frame Ack Counter: 2 bytes.
- App log is printed out: VCP, 115200,8,N,1,
- Alternative frame format is possible: user defined text

Base Station receives data and send back, if RF channel is free, the acknowledge frame:

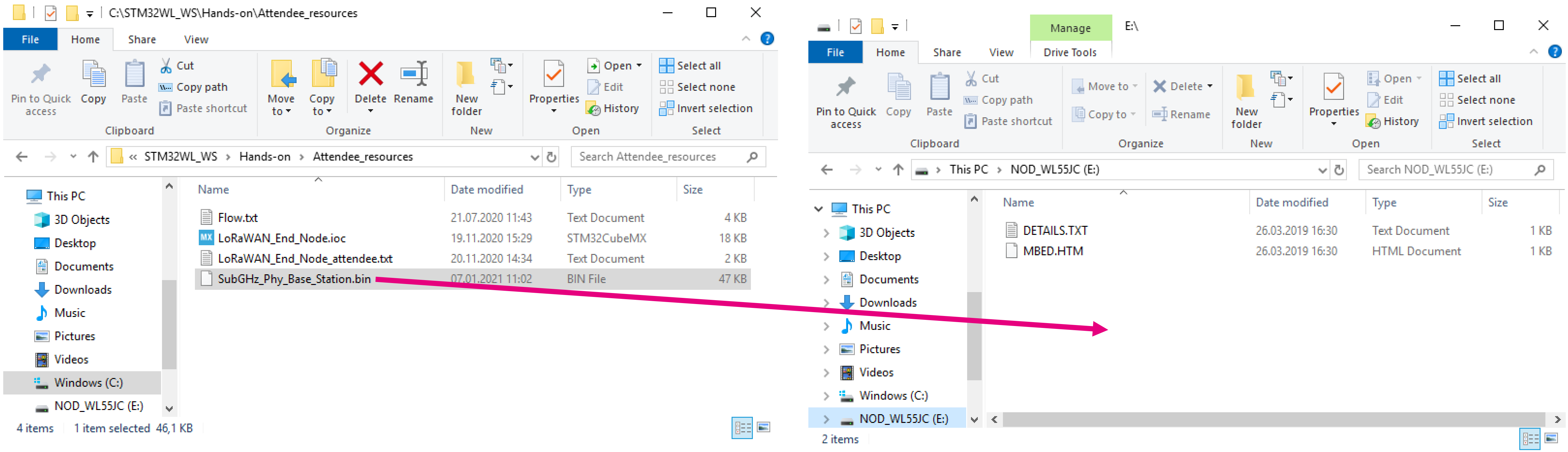
- Ascii string: „ACK” is send back,
- App log is printed out: VCP,115200,8,N,1



P2P: hands-on

Prepare Base Station: connecte to the PC one of the Nucleo-WL board and download the binary …\STM32WL_WS\Hands-on\Attendee_resources\SubGHz_Phy_Base_Station.bin

You can just **copy/paste** the binary file from WS repository to ST-Link debugger disk drive instance using Windows Explorer



C:\STM32WL_WS\Hands-on\Attendee_resources

Name	Date modified	Type	Size
Flow.txt	21.07.2020 11:43	Text Document	4 KB
LoRaWAN_End_Node.ioc	19.11.2020 15:29	STM32CubeMX	18 KB
LoRaWAN_End_Node_attendee.txt	20.11.2020 14:34	Text Document	2 KB
SubGHz_Phy_Base_Station.bin	07.01.2021 11:02	BIN File	47 KB

4 items 1 item selected 46,1 KB

E:\

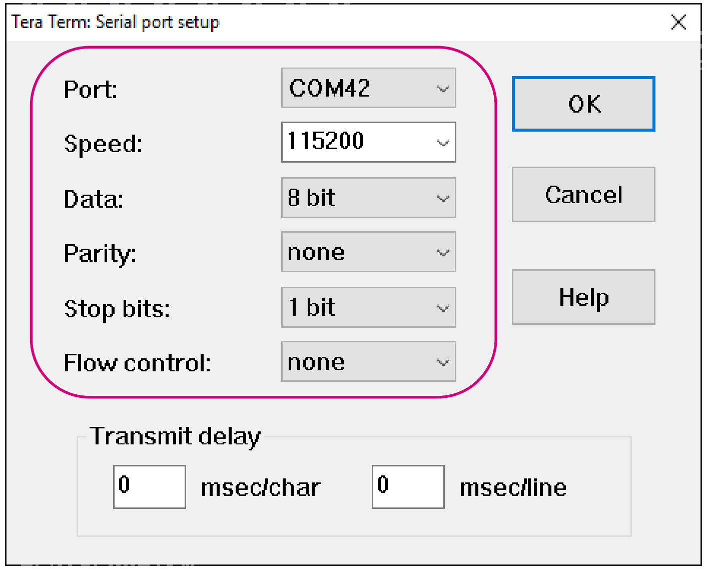
Name	Date modified	Type	Size
DETAILS.TXT	26.03.2019 16:30	Text Document	1 KB
MBED.HTM	26.03.2019 16:30	HTML Document	1 KB

2 items

Source code of Base Station project is in below folder of the WS repository
…\Hands-on\Point2Point_Phy\Projects\NUCLEO-WL55JC\Applications\SubGHz_Phy\P2P_Base_Station

P2P: hands-on

Start Nucleo VCP terminal: 115200,8,N,1 and test if Base Station board is working, press Nucleo-WL reset button when connected



Port: COM42

Speed: 115200

Data: 8 bit

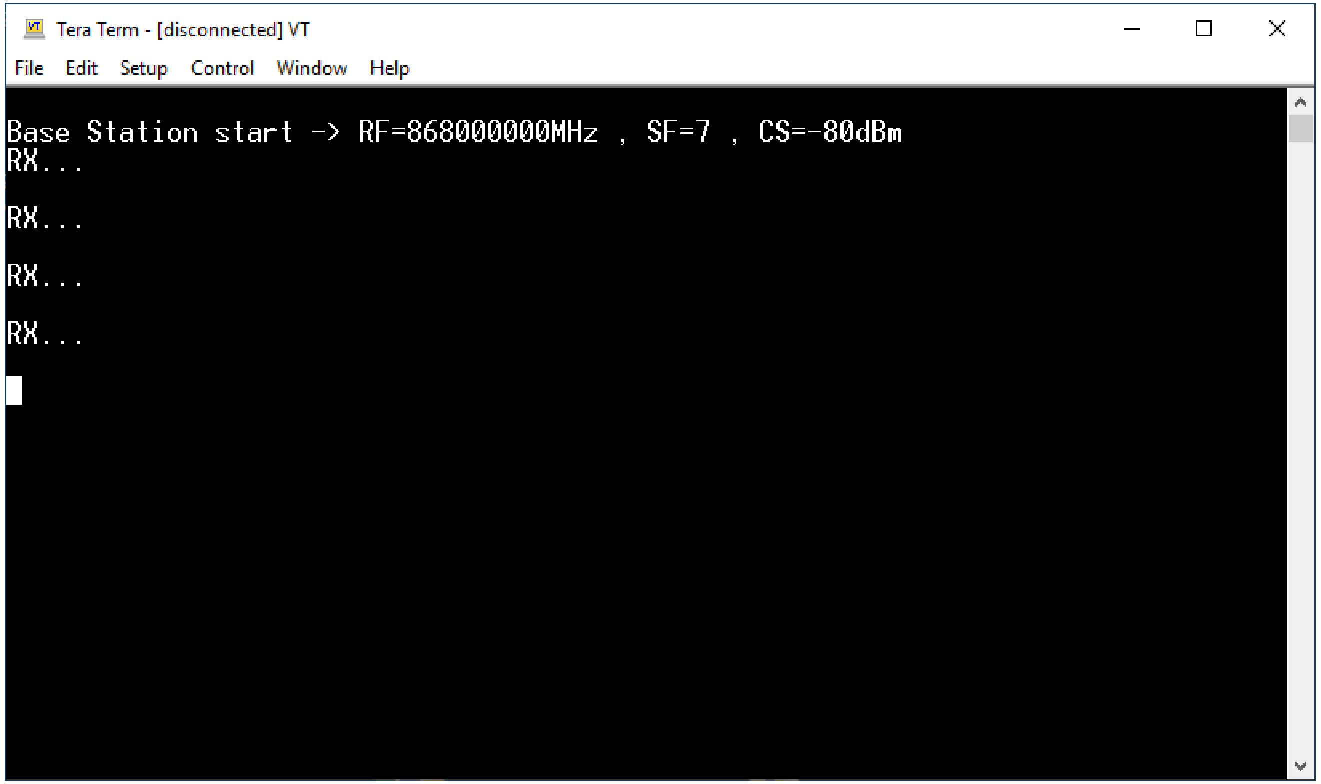
Parity: none

Stop bits: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line



```
Base Station start -> RF=868000000MHz , SF=7 , CS=-80dBm
RX...

RX...

RX...

RX...
```

When Base Station board is working, **disconnect** the board from the PC

P2P: Parameters of the RF link

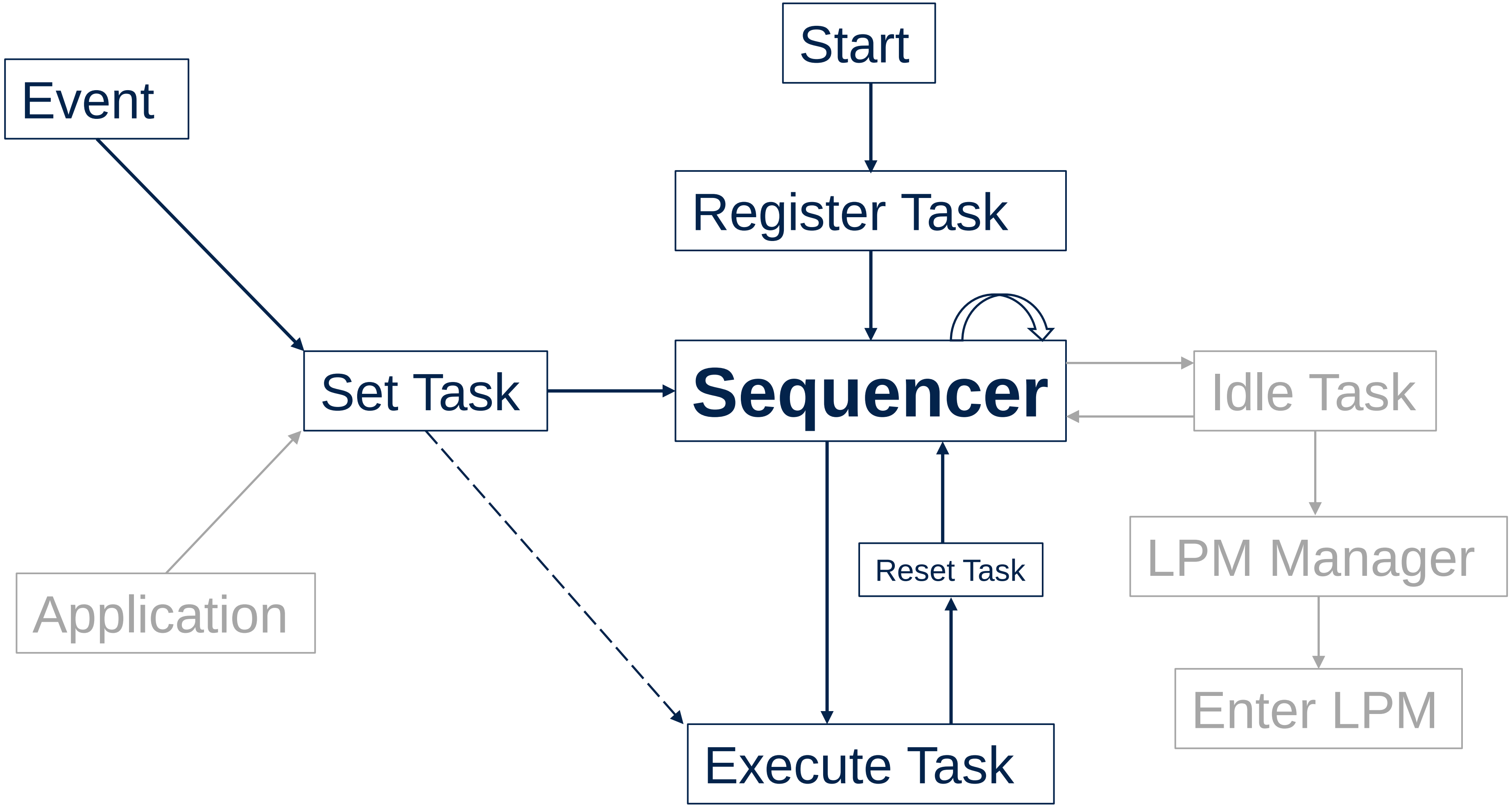
File: subghz_phy_app.h

```
#define RF_FREQUENCY                   868000000 /* Hz */
#define TX_OUTPUT_POWER                10        /* dBm */
#define LORA_BANDWIDTH                  0         /* [0: 125 kHz, 1: 250 kHz, 2: 500 kHz */
#define LORA_SPREADING_FACTOR          7         /* [SF7..SF12] */

#define RF_CHANNEL_FREE_TRIALS_MAX     5
#define RF_CHANNEL_FREE_RSSI_TRESHOLD  -80  /* dBm */
#define RSSI_SENSING_TIME              10   /* ms */
#define CS_BACKOFF_TIME_UNIT           20   /* ms */
```

P2P: Sequencer

Low-power app: event driven application flow



P2P: Sequencer

Register task

```
UTIL_SEQ_RegTask((1 << CFG_SEQ_Task_Sensor_Process),0,Sensor_Process);
```

Run Sequencer in main loop

```
UTIL_SEQ_Run(UTIL_SEQ_DEFAULT);
```

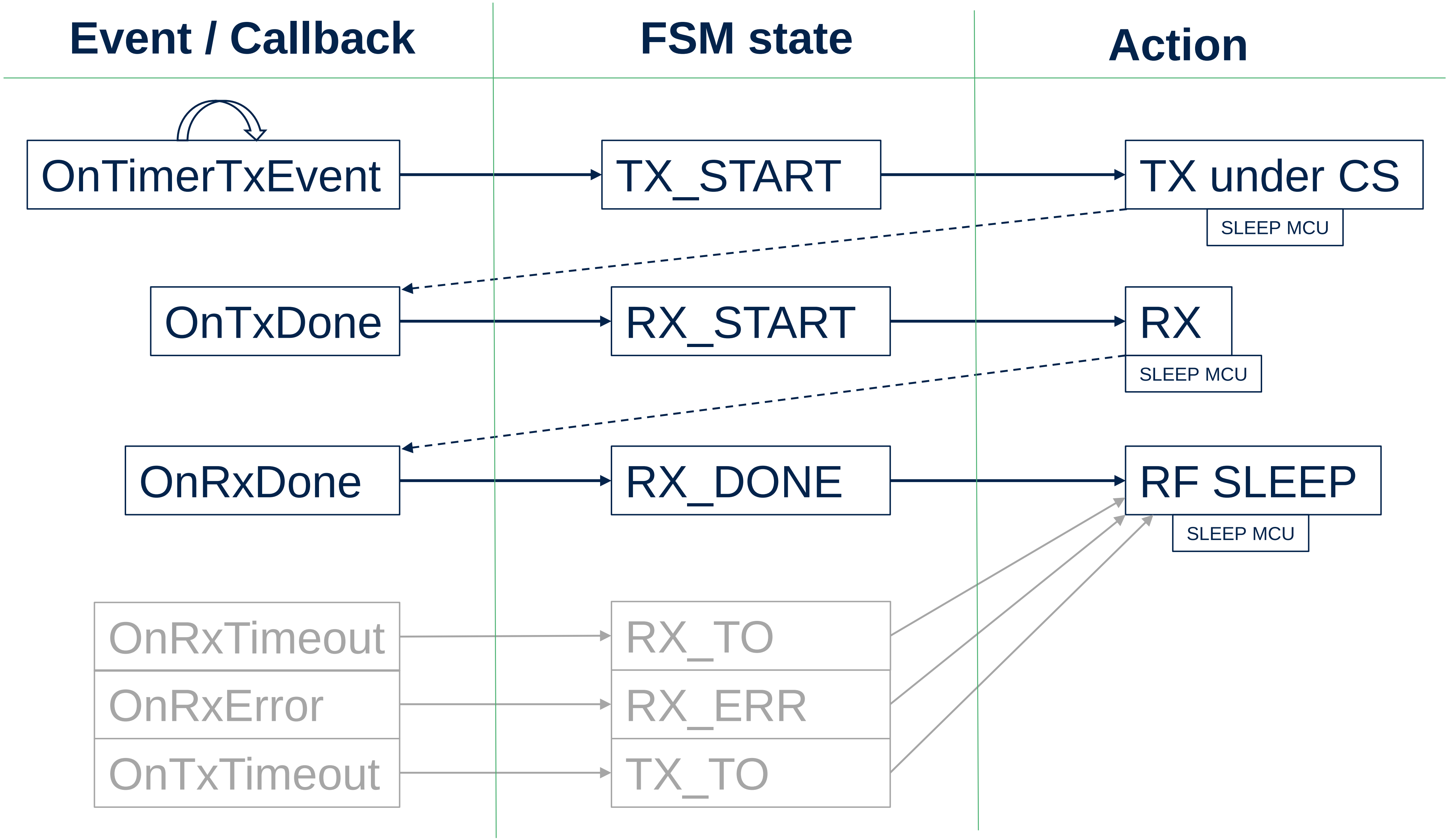
Set task within callback function

```
static void OnTxDone(void)
{
  State = RX_START;
  UTIL_SEQ_SetTask((1 << CFG_SEQ_Task_Sensor_Process), CFG_SEQ_Prio_0);
}
```

Define Idle task of Sequencer (weak function)

```
void UTIL_SEQ_Idle(void)
{
  UTIL_LPM_EnterLowPower();
}
```

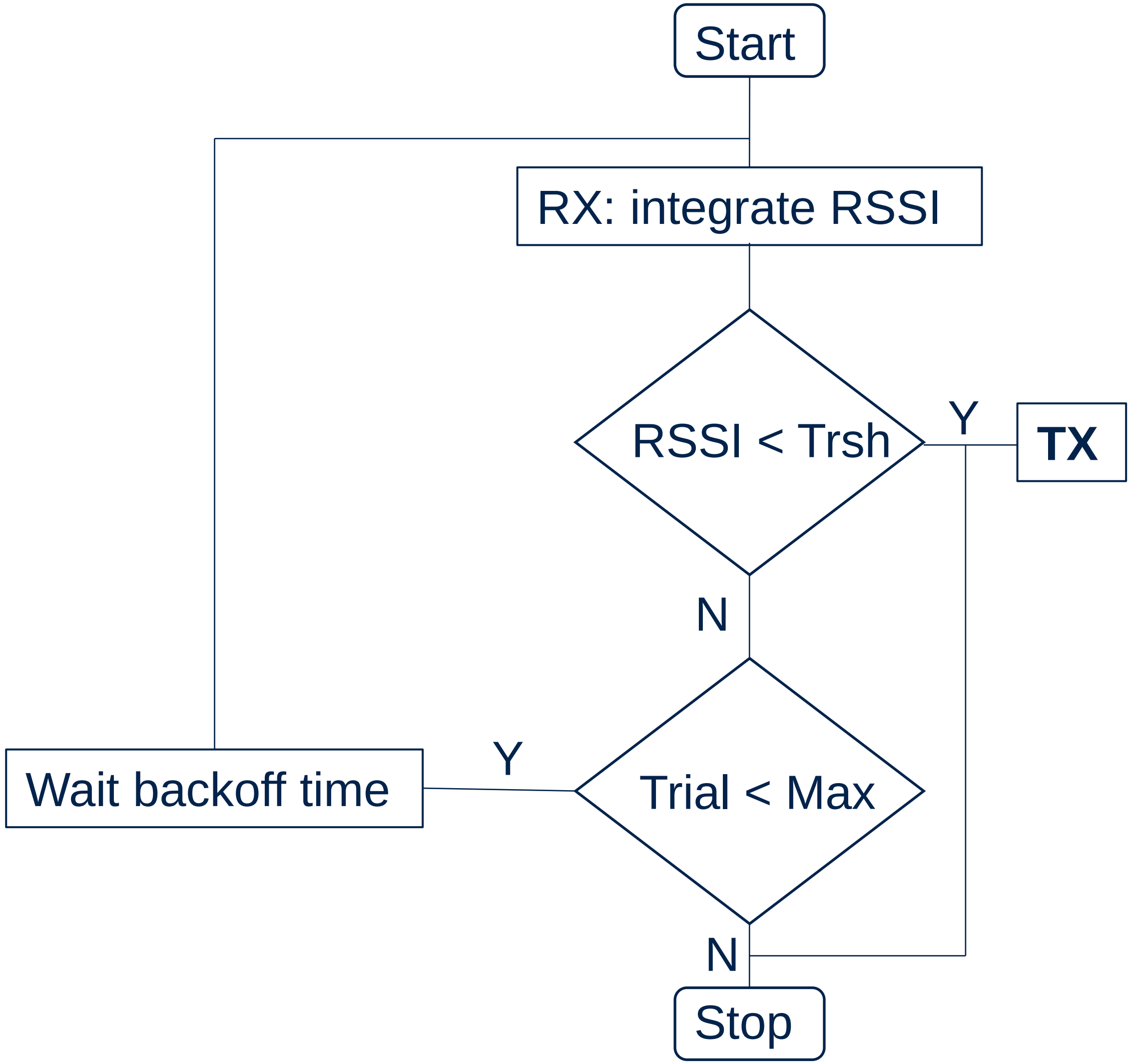
P2P: Sensor_Process & FSM



Event / Callback	FSM state	Action
OnTimerTxEvent	TX_START	TX under CS SLEEP MCU
OnTxDone	RX_START	RX SLEEP MCU
OnRxDone	RX_DONE	RF SLEEP SLEEP MCU
OnRxTimeout	RX_TO	
OnRxError	RX_ERR	
OnTxTimeout	TX_TO	

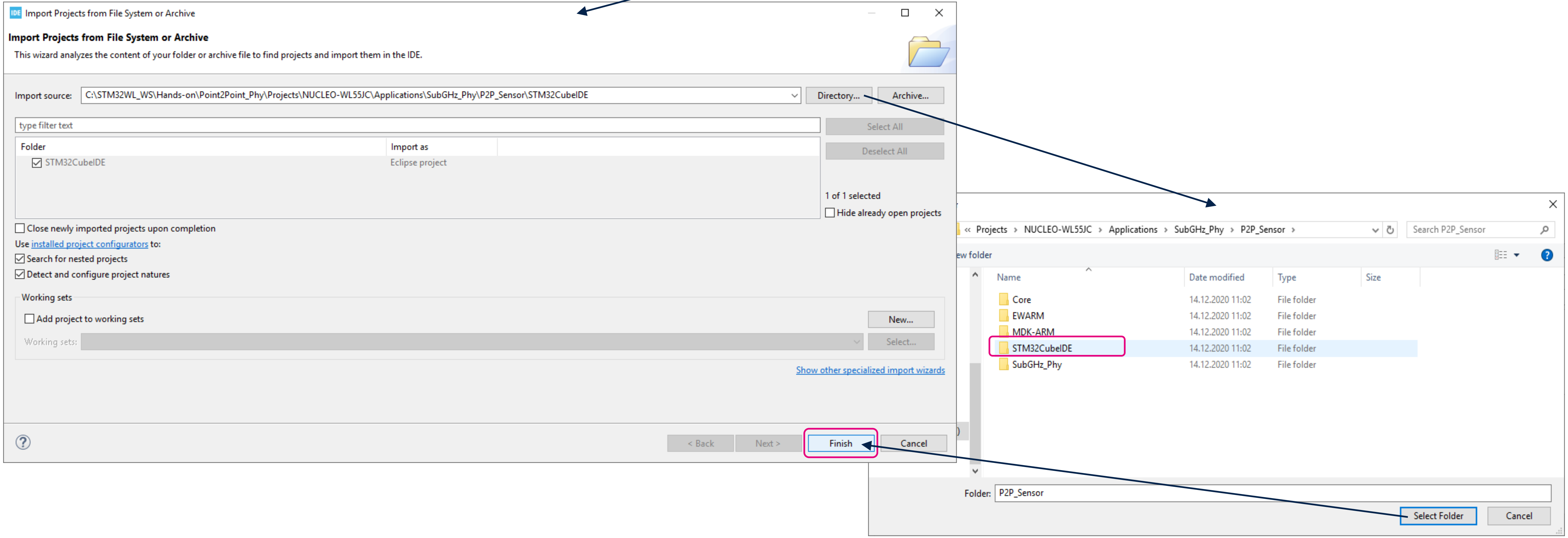
P2P: Channel Sensing

Channel Sensing mechanizm is introduced in order to reduce over the air collisions ratio



P2P: hands-on

Prepare Sensor node: connect to the PC 2nd Nucleo-WL and open project using CubeIDE
File → Open Projects from File System...
…\Hands-on\Point2Point_Phy\Projects\NUCLEO-WL55JC\Applications\SubGHz_Phy\P2P_Sensor\STM32CubeIDE



Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: C:\STM32WL_WS\Hands-on\Point2Point_Phy\Projects\NUCLEO-WL55JC\Applications\SubGHz_Phy\P2P_Sensor\STM32CubeIDE

type filter text

Folder	Import as
☑ STM32CubeIDE	Eclipse project

1 of 1 selected

□ Hide already open projects

□ Close newly imported projects upon completion

Use installed project configurators to:

☑ Search for nested projects

☑ Detect and configure project natures

Working sets

□ Add project to working sets

Working sets:

Show other specialized import wizards

« Projects › NUCLEO-WL55JC › Applications › SubGHz_Phy › P2P_Sensor ›

Search P2P_Sensor

Name	Date modified	Type	Size
Core	14.12.2020 11:02	File folder	
EWARM	14.12.2020 11:02	File folder	
MDK-ARM	14.12.2020 11:02	File folder	
STM32CubeIDE	14.12.2020 11:02	File folder	
SubGHz_Phy	14.12.2020 11:02	File folder	

Folder: P2P_Sensor

P2P: hands-on

App lower & higher layer

Core: main.c , etc.

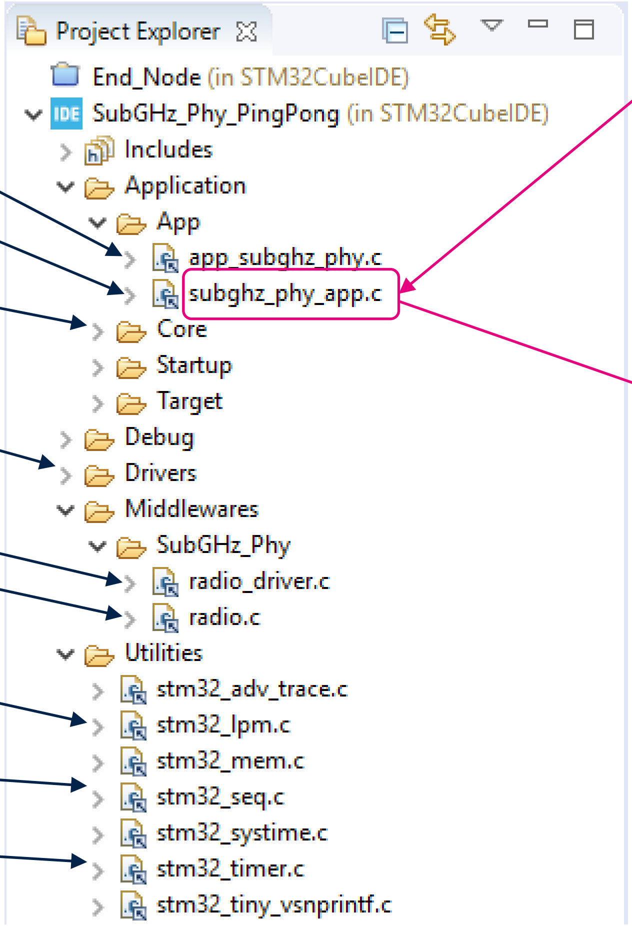
HAL library

Phy higher & lower layer

Low-Power Manager

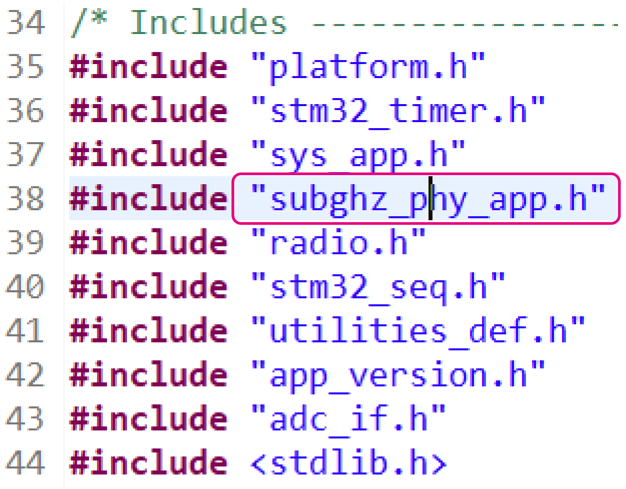
Sequencer

Software timers



```
Project Explorer
End_Node (in STM32CubeIDE)
SubGHz_Phy_PingPong (in STM32CubeIDE)
  Includes
  Application
    App
      app_subghz_phy.c
      subghz_phy_app.c
    Core
    Startup
    Target
  Debug
  Drivers
  Middlewares
    SubGHz_Phy
      radio_driver.c
      radio.c
  Utilities
    stm32_adv_trace.c
    stm32_lpm.c
    stm32_mem.c
    stm32_seq.c
    stm32_systime.c
    stm32_timer.c
    stm32_tiny_vsnprintf.c
```

Open in editor then put the cursor on subghz_phy_app.h and press F3 (open relevant file in editor as well)



```c
/* Includes ----------------
#include "platform.h"
#include "stm32_timer.h"
#include "sys_app.h"
#include "subghz_phy_app.h"
#include "radio.h"
#include "stm32_seq.h"
#include "utilities_def.h"
#include "app_version.h"
#include "adc_if.h"
#include <stdlib.h>
```

P2P: hands-on

File: subghz_phy_app.h

```
#define RX_TIMEOUT_VALUE                          2000  /* [ms] */
#define TX_TIMEOUT_VALUE                          3000  /* [ms] */
#define BUFFER_SIZE                               64  /* Define the payload size here */
#define LED_PERIOD_MS                             100
#define LED_ERROR_PERIOD_MS                       500
#define TX_PERIOD_MS                              10000  /* App TX duty cycle */

#define TCXO_WORKAROUND_TIME_MARGIN               50  /* 50ms margin */

#define RF_CHANNEL_FREE_TRIALS_MAX                5
#define RF_CHANNEL_FREE_RSSI_TRESHOLD             -70  /* [dBm] */
#define RSSI_SENSING_TIME                         10   /* [ms] */
#define CS_BACKOFF_TIME_UNIT                      20   /* [ms] */

#define NODE_ID                                   (uint8_t)(0x01)  /* Node address */
```

Modify NODE_ID following the given one by trainer

File: subghz_phy_app.c

P2P: hands-on

Sensor_Process task implementation

```c
static void Sensor_Process(void)
{
  int16_t temperatureDegC;
  uint32_t i,backoffTime,carrierSenseTime;
  int16_t rssi;
  bool isChannelFree = true;

  switch (State)
  {
    case TX_START:
  temperatureDegC = GetTemperatureLevel();
  i = 0;
#if 1  /* Byte data format */
  Buffer[i++] = NODE_ID;
  Buffer[i++] = temperatureDegC & 0xFF;
  Buffer[i++] = (temperatureDegC>>8) & 0xFF;
  Buffer[i++] = (FrameSentCnt>>8) & 0xFF;
  Buffer[i++] = FrameSentCnt & 0xFF;
  Buffer[i++] = (FrameAckCnt>>8) & 0xFF;
  Buffer[i++] = FrameAckCnt & 0xFF;
#endif
.
.
.
}
```

Callback examples

```c
static void OnTimerTxEvent(void *context)
{
  State = TX_START;
  UTIL_TIMER_Start(&timerTx);
  UTIL_TIMER_Start(&timerLedTx);
  SYS_LED_On(SYS_LED_BLUE);
  UTIL_SEQ_SetTask((1 << CFG_SEQ_Task_Sensor_Process), CFG_SEQ_Prio_0);
}
.
.
.
static void OnRxDone(uint8_t *payload, uint16_t size, int16_t rssi,
int8_t snr)
{
  BufferSize = size;
  memcpy(Buffer, payload, BufferSize);
  RssiValue = rssi;
  SnrValue = snr;

  State = RX_DONE;
  UTIL_SEQ_SetTask((1 << CFG_SEQ_Task_Sensor_Process), CFG_SEQ_Prio_0);
}
```

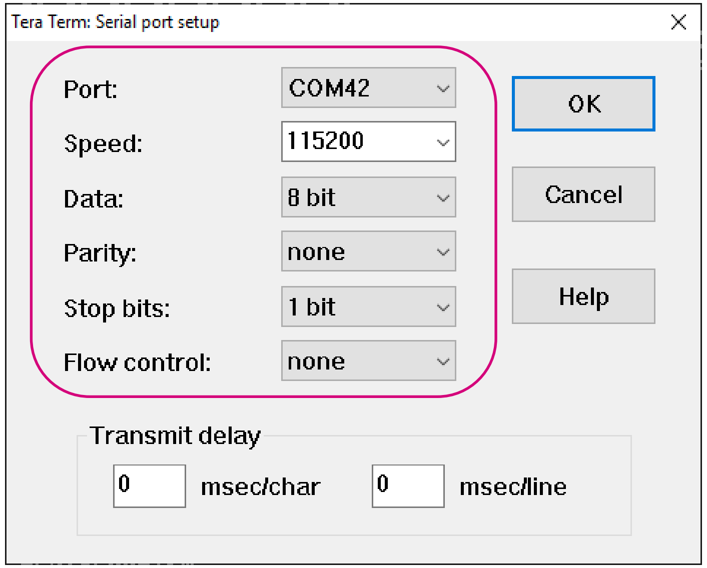
P2P: hands-on

1. Build

2. Debug

3. Stop debug when MCU is flashed

4. Start Nucleo VCP terminal: 115200,8,N,1



Tera Term: Serial port setup

Port: COM42

Speed: 115200

Data: 8 bit

Parity: none

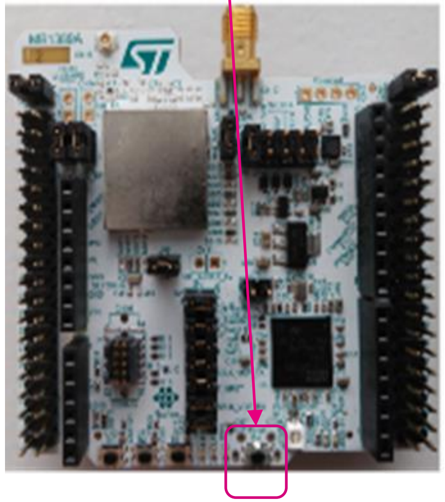
Stop bits: 1 bit

Flow control: none

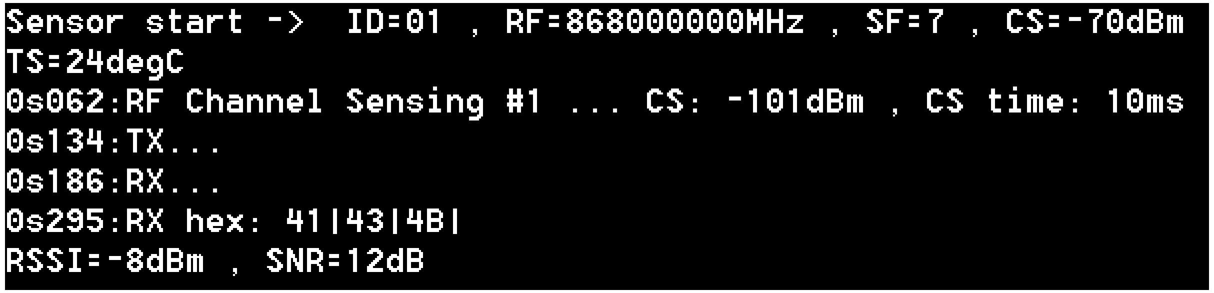
Transmit delay

0 msec/char 0 msec/line

5. Reset MCU



6. Follow terminal log



```
Sensor start ->  ID=01 , RF=868000000MHz , SF=7 , CS=-70dBm
TS=24degC
0s062:RF Channel Sensing #1 ... CS: -101dBm , CS time: 10ms
0s134:TX...
0s186:RX...
0s295:RX hex: 41|43|4B|
RSSI=-8dBm , SNR=12dB
```

Thank you

life.augmented