

ДОДАТОК А  
ТЕКСТ ПІДПРОГРАМИ ВИВЕДЕННЯ ПОТОЧНИХ ЗНАЧЕНЬ  
НАЛАШТУВАННЯ МОДУЛЯ LORA

```

from SX127x.LoRa import * from SX127x.board_config import BOARD
import unittest

def get_reg(reg_addr):
    return lora.get_register(reg_addr)

def SaveState(reg_addr, n_registers=1):
    """ This decorator wraps a get/set_register around the function (unittest)
call.
:param reg_addr: Start of register addresses
:param n_registers: Number of registers to save. (Useful for MSB/LSB
register pairs, etc.) :return: """ def decorator(func): def
wrapper(self):
    reg_bkup = lora.get_register(reg_addr) func(self)
lora.set_register(reg_addr, reg_bkup) return wrapper return decorator

class TestSX127x(unittest.TestCase):

def test_setter_getter(self):
    bkup = lora.get_payload_length() for l in [1,50, 128, bkup]:
lora.set_payload_length(l) self.assertEqual(lora.get_payload_length(), l)

@SaveState(REG.LORA.OP_MODE) def test_mode(self):

```

```

        mode = lora.get_mode()                for m in [MODE.STDBY,
MODE.SLEEP, mode]:        lora.set_mode(m)
        #self.assertEqual(lora.get_mode(), m)

        @SaveState(REG.LORA.FR_MSB,    n_registers=3)                def
test_set_freq(self):        freq = lora.get_freq()        for f in [433.5, 434.5, 434.0,
freq]:        lora.set_freq(f)        self.assertEqual(lora.get_freq(), f)

        @SaveState(REG.LORA.MODEM_CONFIG_3)                def
test_set_agc_on(self):
        lora.set_agc_auto_on(True)
        self.assertEqual((get_reg(REG.LORA.MODEM_CONFIG_3)
&
0b100) >> 2, 1)        lora.set_agc_auto_on(False)
        self.assertEqual((get_reg(REG.LORA.MODEM_CONFIG_3)
&
0b100) >> 2, 0)

        @SaveState(REG.LORA.MODEM_CONFIG_3)
def test_set_low_data_rate_optim(self):
        lora.set_low_data_rate_optim(True)
        self.assertEqual((get_reg(REG.LORA.MODEM_CONFIG_3)
0b1000) >> 3, 1)        lora.set_low_data_rate_optim(False)        &
        self.assertEqual((get_reg(REG.LORA.MODEM_CONFIG_3) &
0b1000) >> 3, 0)

        @SaveState(REG.LORA.DIO_MAPPING_1,    2)                def
test_set_dio_mapping(self):

```

```

        dio_mapping = [1] * 6          lora.set_dio_mapping(dio_mapping)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_1),
                 0b01010101)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_2),
                 0b01010000)    self.assertEqual(lora.get_dio_mapping(), dio_mapping)

```

```

        dio_mapping = [2] * 6          lora.set_dio_mapping(dio_mapping)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_1),
                 0b10101010)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_2),
                 0b10100000)    self.assertEqual(lora.get_dio_mapping(), dio_mapping)

```

```

        dio_mapping = [0] * 6          lora.set_dio_mapping(dio_mapping)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_1),
                 0b00000000)
        self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_2),
                 0b00000000)    self.assertEqual(lora.get_dio_mapping(), dio_mapping)

```

```

        dio_mapping = [0,1,2,0,1,2]    lora.set_dio_mapping(dio_mapping)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_1),
                 0b00011000)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_2),
                 0b01100000)    self.assertEqual(lora.get_dio_mapping(), dio_mapping)

```

```

        dio_mapping = [1,2,0,1,2,0]    lora.set_dio_mapping(dio_mapping)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_1),

```

```

0b01100001)
self.assertEqual(get_reg(REG.LORA.DIO_MAPPING_2),
0b10000000)    self.assertEqual(lora.get_dio_mapping(), dio_mapping)

# def test_set_lna_gain(self):
#     bkup_lna_gain = lora.get_lna()['lna_gain']
#     for target_gain in [GAIN.NOT_USED, GAIN.G1, GAIN.G2,
GAIN.G6, GAIN.NOT_USED, bkup_lna_gain]:
#         print(target_gain)
#         lora.set_lna_gain(target_gain)
#         actual_gain = lora.get_lna()['lna_gain']
#
self.assertEqual(GAIN.lookup[actual_gain],
GAIN.lookup[target_gain])

if __name__ == '__main__':

    BOARD.setup()    lora = LoRa(verbose=False)    unittest.main()
BOARD.teardown()

```

ДОДАТОК Б  
ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

