

```
'nRF24L01 LIB with PIC SSP MODULE, Hardware SPI Modul
' 20.09.2017
```

```
'Memory Map - register address defines
```

```
#define CONFIG      0x00
#define EN_AA       0x01
#define EN_RXADDR    0x02
#define SETUP_AW     0x03
#define SETUP_RETR   0x04
#define RF_CH        0x05
#define RF_SETUP     0x06
#define STATUS_REG   0x07 'page 55, STATUS is a PIC register -> STATUS_REG for
nRF24L01
#define OBSERVE_TX   0x08
#define RPD          0x09 'Received Power Detector
#define RX_ADDR_P0   0x0A
#define TX_ADDR      0x10
#define RX_PW_P0     0x11
#define FIFO_STATUS  0x17
```

```
'Command Name Mnemonics (Instructions)
```

```
#define R_REGISTER   0x00
#define W_REGISTER   0x20 '001A AAAA
#define R_RX_PAYLOAD 0x61
#define W_TX_PAYLOAD 0xA0
#define FLUSH_TX     0xE1
#define FLUSH_RX     0xE2
#define WL_NOP       0xFF
```

```
'----- subs -----
sub Read_Reg(in Register as byte) 'Read content of a 1 byte register
    Dummy = 0
    Register = R_REGISTER + Register
```

```
    set CSN off
    SPITransfer Register, Dummy
    SPITransfer R_REGISTER, Dummy
    set CSN on
end sub 'Read_Reg
```

```
'-----
sub Write_Reg(in Register as byte, in Data as byte) 'Write 1 byte to register
    Dummy = 0
    Register = W_REGISTER + Register
```

```
    set CSN off
    SPITransfer Register, Dummy
    SPITransfer Data, Dummy
    set CSN on
end sub 'Write_Reg
```

```
'-----
sub Write_Addr_Pipe(in Register as byte)
    Register = W_REGISTER + Register

    set CSN off
    SPITransfer Register, Dummy

    for byte_num = 1 to 3
        SPITransfer RXTX_ADDR(byte_num), Dummy
    next
    set CSN on
end sub 'Write_Addr_Pipe
```

```
'-----
sub Write_Cmd(in command as byte)
    set CSN off
    SPITransfer command, Dummy
    set CSN on
end sub 'Write_Cmd
```

```
'-----
sub RX_Mode
    WRITE_REG(CONFIG, 0b1111011) '1 byte CRC, POWER UP, PRX, INT's disabled
    set CE on
    wait 130 us 'wait for RX ready
end sub 'RX_Mode
```

```

'-----
sub Read_Payload
  set CSN off
  SPITransfer R_RX_PAYLOAD, Dummy

  for num = 1 to RX_LENGTH
    SPITransfer WL_NOP, rx_data(num)
  next

  set CSN on
end sub 'Read_Payload
'-----
sub Write_Payload
  set CSN off
  SPITransfer W_TX_PAYLOAD, Dummy
  for num = 1 to TX_LENGTH
    SPITransfer tx_data(num), Dummy
  next
  set CSN on

  set CE on
  wait 150 us
  set CE off
end sub 'Write_Payload
'-----
sub TX_Mode
  set CE off
  WRITE_REG(CONFIG, 0b1111010) '1 byte CRC, POWER UP, PTX, INT's disabled
end sub 'TX_Mode
'-----

```

```

' nRF24L01 Library with Software SPI
' 24.09.2017
'
'nRF24L01 Settings and defined commands
#define Carry STATUS.0 ' Carry Bit of STATUS register

'Memory Map - register address defines
#define CONFIG      0x00
#define EN_AA       0x01
#define EN_RXADDR    0x02
#define SETUP_AW     0x03
#define SETUP_RETR   0x04
#define RF_CH        0x05
#define RF_SETUP     0x06
#define STATUS_REG   0x07 'page 55, STATUS is a PIC register -> STATUS_REG for
nRF24L01
#define OBSERVE_TX   0x08
#define RPD          0x09 'Received Power Detector
#define RX_ADDR_P0   0x0A
#define TX_ADDR      0x10
#define RX_PW_P0     0x11
#define FIFO_STATUS  0x17

'Command Name Mnemonics (Instructions)
#define R_REGISTER    0x00
#define W_REGISTER    0x20 '001A AAAA
#define R_RX_PAYLOAD  0x61
#define W_TX_PAYLOAD  0xA0
#define FLUSH_TX       0xE1
#define FLUSH_RX       0xE2
#define WL_NOP         0xFF

'-----
sub spi_tx_byte(in Dout as byte) 'Software SPI Serial Data Out (SDO)
    set SDO off

    Repeat 8
        rlf Dout,f 'rotate left asm instruction
        if (Carry = 1) then set SDO on
        if (Carry = 0) then set SDO off
        set SCK on
        wait 8 us
        set SCK off
    End Repeat
end sub ' spi_tx_byte

'-----
sub Write_Reg(in Register as byte, in Data as byte) 'Write 1 byte to register
    Register = W_REGISTER + Register

    set CSN off
    spi_tx_byte(Register)
    spi_tx_byte(Data)
    set CSN on
end sub 'Write_Reg

'-----
sub Write_Addr_Pipe(in Register as byte)
    Register = W_REGISTER + Register

    set CSN off
    spi_tx_byte(Register)

    for byte_num = 1 to 3
        spi_tx_byte(RXTX_ADDR(byte_num))
    next
    set CSN on
end sub 'Write_Addr_Pipe

'-----
sub Write_Cmd(in command as byte)
    set CSN off
    spi_tx_byte(command)
    set CSN on
end sub 'Write_Cmd

'-----

```

```

sub Write_Payload
    set CSN  off
    spi_tx_byte(W_TX_PAYLOAD)

    for num = 1 to TX_LENGTH
        spi_tx_byte(tx_data(num))
    next
    set CSN  on

    set CE on
    wait 150 us
    set CE off
end sub 'Write_Payload
'-----
sub TX_Mode
    set CE off
    WRITE_REG(CONFIG, 0b1111010) '1 byte CRC, POWER UP, PTX, INT's disabled
end sub 'TX_Mode
'-----

```

```
nRF24L01 Telemetrie_RX_V1
21.09.2017

-----
V3.3P          -| 1      20 | - GND
PORTA.5        -| 2      19 | - PORTA.0 SER_OUT
PORTA.4        -| 3      18 | - PORTA.1
PORTA.3    PUSH_BUTTON -| 4      17 | - PORTA.2
PORTC.5        -| 5      16 | - PORTC.0
PORTC.4        -| 6      15 | - PORTC.1
PORTC.3    ALARM   -| 7      14 | - PORTC.2
PORTC.6        CSN -| 8      13 | - PORTB.4 SDI
PORTC.7        SDO -| 9      12 | - PORTB.5
PORTB.7        CE  -| 10     11 | - PORTB.6 SCK
-----

nRF24L01 Pins top view, SMD view
-----
GND           1 *      * 2 V3.3P
CE            3 *      * 4 CSN
SCK           5 *      * 6 MOSI (SDO)
MISO (SDI) 7 *      * 8 IRQ
-----
Quarz
-----

#chip 16F690,8
#config OSC=INTRC_OSC_NOCLKOUT, PWRTE = ON

#include "C:\gcbasic_src\software_uart_v2.h"
#include "nRF24L01.h"

'nRF24L01 Settings and defined commands
#define FlushRX Write_Cmd(FLUSH_RX) 'used in RX mode, discard packet

' received data array index
#define FRAME_COUNTER 1
#define VOLTAGE       2

'Alarm threshold LiPo
#define VOLT_9P9 139 'Voltage divider with 22uF, 10K/(10K+47K), Vref = 3.3 Volt nominal
#define CONF_13SEC 26

#define STANDBY_I 0b01110010 'Power up, Interrupt's disabled
#define RX_LENGTH 2 ' Number of bytes to receive

'Serial settings
#define BAUD_RATE 9600
#define SER_OUT PORTA.0
dir SER_OUT out

bsf IOC,IOCA3 'enable Interrupt PORTABChange on PORTA.3
On Interrupt PORTABChange Call ISR1 'Interrupt Service Routine 1

#define ALARM PORTC.3 'Buzzer
dir ALARM out

'SPI pin settings
#define SDO PORTC.7
dir SDO out

#define SDI PORTB.4
dir SDI in

#define CSN PORTC.6
dir CSN out

#define SCK PORTB.6
dir SCK out

#define CE PORTB.7
dir CE out
```

```

'variables
dim Dummy,conf_counter,err_counter,f_counter_old,rx_counter,lq as byte

Dummy = 0
f_counter_old = 0
conf_counter = 0
err_counter = 0
rx_counter = 0
lq = 0

dim RXTX_ADDR(3) 'RX TX 3 byte address data pipe 0
dim rx_data(RX_LENGTH) 'data bytes to receive

rx_data(FRAME_COUNTER) = 0
rx_data(VOLTAGE) = 0

dim PUSH_BUTTON as bit 'later Menu setup
PUSH_BUTTON = 0

'Start up
wait 1 s 'wait for LCD ready
LCD_Clear
SerPrint("V1.1")
wait 2 s
LCD_Clear

'nRF_Setup
'init SPI & CE pin

SPIMode MasterSlow,0 'SPI Mode Setup

set SDO off
set SCK off
set CSN on
set CE off

' nRF Setup
RXTX_ADDR(1) = 0xB1
RXTX_ADDR(2) = 0xB2
RXTX_ADDR(3) = 0xB3

Write_Reg(CONFIG, STANDBY_I) ' STANDBY_I Mode
Write_Reg(EN_AA, 0x00) ' Disable auto ack
Write_Reg(EN_RXADDR, 0x01) ' Enable data pipe 0
Write_Reg(SETUP_AW, 0x01) ' 3 byte address
Write_Reg(SETUP_RETR, 0x00) ' Retransmit disabled
Write_Reg(RF_CH, 0x3F) ' RF channel 63
Write_Reg(RF_SETUP, 0x26) ' 250kbps, 0dBm
Write_Reg(RX_PW_P0, RX_LENGTH) ' RX payload, num of bytes, max 32
Write_Addr_Pipe(RX_ADDR_P0) ' 3 byte RX address data pipe 0
RX_Mode
FlushRX
' end nRF_Setup

'Power Up Test Buzzer
set ALARM on
wait 1 s
set ALARM off

'----- main -----
Do
  Read_Payload

  if (rx_data(VOLTAGE) < VOLT_9P9) then
    conf_counter++
    if (PUSH_BUTTON = 0) and (conf_counter >= CONF_13SEC) then
      set ALARM on
      conf_counter = CONF_13SEC
    end if
  end if
  if (rx_data(VOLTAGE) >= VOLT_9P9) then
    set ALARM off
  end if
end Do

```

```

        conf_counter = 0
    end if
    if (PUSH_BUTTON = 1) then
        set ALARM off
        conf_counter = CONF_13SEC
    end if

    if (f_counter_old = rx_data(FRAME_COUNTER)) then err_counter++
    rx_counter++
    if (rx_counter = 10) then
        lq = (10 - err_counter) * 10
        rx_counter = 0
        err_counter = 0
    end if

    LCD_Clear
    SerPrint("LQ:")
    print_byte(lq)
    LCD_Line2
    print_2p1(rx_data(VOLTAGE)*72)
    SerPrint(" V")
    f_counter_old = rx_data(FRAME_COUNTER)

    wait 500 ms
Loop

'----- subs -----
sub ISR1 'Interrupt Service Routine
    PUSH_BUTTON = 1
    wait 255 ms
end sub 'ISR1

```

```

nRF24L01 Telemetrie_TX_V1
21.09.2017
|
|-----|
| V3.3P          -| 1          20 | - GND
| PORTA.5        -| 2          19 | - PORTA.0
| PORTA.4        -| 3          18 | - PORTA.1 ACCU
| PORTA.3        -| 4          17 | - PORTA.2
| PORTC.5        -| 5          16 | - PORTC.0
| PORTC.4        -| 6          15 | - PORTC.1
| PORTC.3        -| 7          14 | - PORTC.2
| PORTC.6        CSN -| 8          13 | - PORTB.4 SDI
| PORTC.7        SDO -| 9          12 | - PORTB.5
| PORTB.7        CE  -| 10         11 | - PORTB.6 SCK
|-----|
|
| nRF24L01 Pins top view, SMD view
|-----|
| GND            1 *      * 2 V3.3P
| CE             3 *      * 4 CSN   CSN
| SCK            5 *      * 6 MOSI   (SDO)
| MISO (SDI)    7 *      * 8 IRQ
|
|-----|
|
|-----|
| Quartz |
|-----|
|
|-----|
#chip 16F690,8
#config OSC=INTRC_OSC_NOCLKOUT, PWRTE = ON

#include "nRF24L01.h"

'nRF24L01 Settings and defined commands
#define FlushTX Write_Cmd(FLUSH_TX) 'used in TX mode, flush TX FIFO

' TX data array index
#define FRAME_COUNTER 1
#define VOLTAGE        2

#define STANDBY_I 0b01110010 'Power up, Interrupt's disabled
#define TX_LENGTH 2 ' Number of bytes to send

'ADC channel AN1, Vref = 3.3 Volt
'Voltage gain 10K / (10K + 47K)
#define ACCU AN1
dir PORTA.1 in

'SPI pin settings

#define SDO PORTC.7
dir SDO out

#define SDI PORTB.4
dir SDI in

#define CSN PORTC.6
dir CSN out

#define SCK PORTB.6
dir SCK out

#define CE PORTB.7
dir CE out

dim Dummy,f_counter as byte
Dummy = 0
f_counter = 0 'frame counter

dim RXTX_ADDR(3) 'RX TX 3 byte address data pipe 0
dim tx_data(TX_LENGTH) 'data bytes to send

wait 1 s 'wait for voltage low pass filter ready

'nRF Setup

```



```

'init SPI & CE pin

SPIMode MasterSlow,0 'SPI Mode Setup

set SDO off
set SCK off
set CSN on
set CE off

' nRF Setup
RXTX_ADDR(1) = 0xB1
RXTX_ADDR(2) = 0xB2
RXTX_ADDR(3) = 0xB3

Write_Reg(CONFIG, STANDBY_I)      ' STANDBY_I Mode
Write_Reg(EN_AA, 0x00)           ' Disable auto ack
Write_Reg(SETUP_AW, 0x01)         ' 3 byte address
Write_Reg(SETUP_RETR, 0x00)       ' Retransmit disabled
Write_Reg(RF_CH, 0x3F)            ' RF channel 63
Write_Reg(RF_SETUP, 0x26)         ' 250kbps, 0dBm
Write_Addr_Pipe(TX_ADDR)          ' 3 byte TX address data pipe 0
TX_Mode
FlushTX
' end nRF_Setup

'----- main -----
Do
  tx_data(FRAME_COUNTER) = f_counter
  tx_data(VOLTAGE) = Average(ReadAD(ACCU),ReadAD(ACCU))
  f_counter++
  Write_Payload
  wait 500 ms
  FlushTX
Loop

```



```

set CSN on
set CE off

RXTX_ADDR(1) = 0xB1
RXTX_ADDR(2) = 0xB2
RXTX_ADDR(3) = 0xB3

Write_Reg(CONFIG, STANDBY_I)      ' STANDBY_I Mode
Write_Reg(EN_AA, 0x00)           ' Disable auto ack
Write_Reg(SETUP_AW, 0x01)         ' 3 byte address
Write_Reg(SETUP_RETR, 0x00)       ' Retransmit disabled
Write_Reg(RF_CH, 0x3F)           ' RF channel 63
Write_Reg(RF_SETUP, 0x26)        ' 250kbps, 0dBm
Write_Addr_Pipe(TX_ADDR)         ' 3 byte TX address data pipe 0
TX_Mode
FlushTX
' end nRF_Setup

'----- main -----
Do
  tx_data(FRAME_COUNTER) = f_counter
  accu_volt = Average(ReadAD(ACCU), ReadAD(ACCU))
  accu_volt = ([word]accu_volt * 101) / 100 'ADC gain correction
  tx_data(VOLTAGE) = accu_volt
  f_counter++
  Write_Payload
  wait 500 ms
  FlushTX
Loop

```