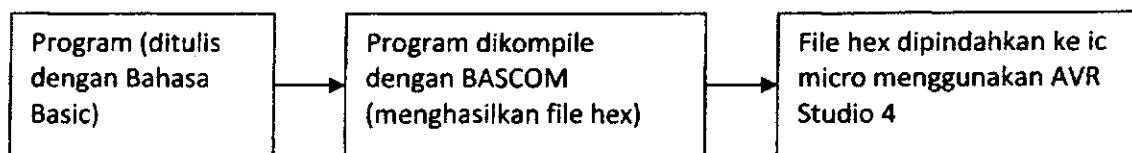


## LAMPIRAN

Program Transmit dan Receive modul RFM12. Program ditulis dengan bahasa Basic dan compiler BASCOM dari MCS Electronic, serta menggunakan AVR Studio 4 dari ATMEL untuk memindahkan file program ke ic microcontroller ATMEGA8



```
$regfile = "m8def.dat"  
$hwstack = 32  
$swstack = 10  
$framesize = 40  
$crystal = 8000000  
$baud = 19200
```

```
Const File = "TxRx 1.05 .bas"  
Const Description = "RFM12 monitor program"
```

```
Print  
Print Description  
Print File  
Print Version()
```

```
Rfm12_rst Alias Portc.5  
Config Rfm12_rst = Input
```

```
Config Portb = &B0010_1101
```

```
Test_pin Alias Portb.0
```

```
Rfm12_cs Alias Portb.2  
Set Rfm12_cs
```

```
Rfm12_sdi Alias Portb.3
```

```
Rfm12_sdo Alias Pinb.4  
Set Portb.4
```

```
Rfm12_sck Alias Portb.5
```

```
Rx_led Alias Portd.5  
Config Pind.5 = Output
```

```
Tx_led Alias Portd.6
```

*Config Pind.6 = Output*

*Error\_led Alias Portd.7*  
*Config Pind.7 = Output*

*Config Spi = Hard , Interrupt = Off , Data Order = Msb , Master = Yes , \_*  
*Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 1*

*Spiinit*

*Declare Sub Rfm12\_spi\_wrt*  
*Declare Sub Rfm12\_tx(txarray As Byte)*  
*Declare Sub Rfm12\_rx(rxarray As Byte)*  
*Declare Sub Rfm12\_tx\_ready*  
*Declare Sub Rfm12\_rx\_wait*  
*Declare Sub Rfm12\_get\_status\_bit*  
*Declare Sub Saveflags*  
*Declare Sub Rfm12\_clear\_fifo*  
*Declare Sub Rfm12\_rx\_on*  
*Declare Sub Rfm12\_init*  
*Declare Sub Printflags*

*Min Freq 430.24*

*'Max Freq 439.75*

*'in 2.5KHz steps*

*,*

*'Fc = the center Freq. in MHz*

*'Fr = the Freq. register value.*

*'Fc = 430 + Fr \* 0.0025 (MHz)*

*'Fr = (Fc - 430) \* 400*

*Const Fc = 435.00*

*Const Fr =(fc - 430) \* 400*

*Const Fcmd = &HA000 + Fr*

*'const Payload\_Size = 128*

*Const Payload\_size = 32*

*'const Payload\_Size = 16*

*Const Pls\_plus1 = Payload\_size + 1*

*Const Beacon = "123456789012345678901234567890"*

*Dim Tempbit As Bit*

*Dim Temp As Byte*

*Dim Tempw As Word*

*Dim Spi\_tx\_wrd As Word*

*Dim Spi\_tx\_l As Byte At Spi\_tx\_wrd Overlay*

*Dim Spi\_tx\_h As Byte At Spi\_tx\_wrd + 1 Overlay*

*Dim Spi\_rx\_wrd As Word*

*Dim Spi\_rx\_l As Byte At Spi\_rx\_wrd Overlay*

*Dim Spi\_rx\_h As Byte At Spi\_rx\_wrd + 1 Overlay*

*Dim Rx\_status\_bit As Bit*

*Dim Rxbuffer(pls\_plus1) As Byte*

*Dim Rxstring As String \* Payload\_size At Rxbuffer(1) Overlay*

*Rxbuffer(pls\_plus1) = 0*

*Dim Txindex As Byte*

*Dim Txbuffer(pls\_plus1) As Byte*

*Dim Txstring As String \* Payload\_size At Txbuffer(1) Overlay*

*Txbuffer(pls\_plus1) = 0*

*Dim Timeout As Word*

*Dim Cmnd As String \* 4*

*Dim Flags As Byte*

*Echoflag Alias Flags.0*

*Beaconflag Alias Flags.1*

*Sermmsgflag Alias Flags.2*

*Rxonflag Alias Flags.3*

*Txonflag Alias Flags.4*

*Dim Dummy As Eram Word At 0*

*Dim Progcntr As Eram Word At 2*

*Dim Savedflags As Eram Byte*

*Tempw = Progcntr*

*Print "uC flashed " ; Tempw ; " times."*

*Print*

*'kill some time here while*

*'rfm12 does a power on reset.*

*Reset Rx\_led*

*Reset Tx\_led*

*Reset Error\_led*

*Waitms 500*

*Set Rx\_led*

*Set Tx\_led*

*Set Error\_led*

*Rfm12\_init*

*Rfm12\_rx\_on*

*Flags = Savedflags*

*Print "Echo = " ; Echoflag*

*Print "Beacon = " ; Beaconflag*

*Cmnd = ""*

*Do*

```

If Beaconflag = 1 Then
    'Tx a test beacon about once a second.
    Txstring = Beacon
    Print "Tx: " ; Txstring
    Rfm12_tx Txbuffer(1)
    'now loop waiting for a reply msg.
    For Temp = 1 To 255
        'poll RFM12 for Rx data.
        Rfm12_get_status_bit
        If Rx_status_bit = 1 Then
            'Rx RxBuffer
            Rfm12_rx Rxbuffer(1)
            Print "Rx: " ; Rxstring
            If Rxstring <> Txstring Then
                Reset Error_led
                Print "<>"
            End If
            Rxstring = ""
            Exit For
        End If
    End If
    If Temp = 255 Then
        Reset Error_led
        Print "no echo"
    End If

    'waitus 50      'at 38.3kbps, 32 byte payload
    'waitus 350    'at 19.2kbps, 128 bytes
    Waitus 150      'at 19.2kbps, 32 bytes
    Set Error_led
    Next Temp
Else
    'poll RFM12 for Rx data.
    Rfm12_get_status_bit
    If Rx_status_bit = 1 Then
        'reset RX_LED
        'Rx RxBuffer
        Rfm12_rx Rxbuffer(1)
        Print "Rx: " ; Rxstring
        If Echoflag = 1 Then
            Print "Ec: " ; Rxstring
            'Tx RxBuffer.
            Rfm12_tx Rxbuffer(1)
        End If
        'set RX_LED
    End If
End If

Temp = Inkey()
If Temp <> 0 Then

```

```

If Temp = 27 Then
    'press "esc" to get a command prompt.
    Spi_tx_wrd = 0
    Rfm12_spi_wrt
    Print
    Print Description
    Print File
    Print Version()
    Print
    Tempw = Progcnt
    Print "uC flashed "; Tempw ; " times."
    'print
    'print "Status word = "; bin(SPI_Rx_H) ; " "; bin(SPI_Rx_L)
    Printflags
    Print
    Input "Command >", Cmnd
    Print
    Rfm12_rx_on
Elseif Temp = 13 Then
    Print
    Print "Tx: "; Txstring
    'Tx TxBuffer.
    Rfm12_tx Txbuffer(1)
    Txstring = ""
    Txindex = 0
Else
    'fill the TxBuffer until <cr> or full.
    If Txindex < Payload_size Then
        Print Chr(temp);
        Incr Txindex
        Txbuffer(txindex) = Temp
        Temp = Txindex + 1
        Txbuffer(temp) = 0
    End If
End If
End If

If Cmnd <> "" Then
    Select Case Cmnd
    Case "u"
        'Press "u <CR>"
        Print
        Print "Press F4 to start upload."
        Waitms 3000

        Goto &HC00

    Case "c"

```

```

Print
Input "Enter a Command Word in HEX >" , Cmnd
Spi_tx_wrd = Hexval(cmnd)
Rfm12_spi_wrt
Print "RFM12 returned >" ; Hex(spi_rx_wrd)

Case "b"
Toggle Beaconflag
If Beaconflag = 1 Then Echoflag = 0
Saveflags
Printflags

Case "e"
Toggle Echoflag
If Echoflag = 1 Then Beaconflag = 0
Saveflags
Printflags

Case Else
Print "Unknown Command"

End Select
Print
Cmnd = ""
End If
Loop

Sub Printflags
If Beaconflag = 1 Then
Print "Beacon ON"
Else
Print "Beacon OFF"
End If
If Echoflag = 1 Then
Print "Echo ON"
Else
Print "Echo OFF"
End If
End Sub

Sub Saveflags
'save to eram
Flags = Flags And 3
Savedflags = Flags
End Sub

Sub Rfm12_init
Local I As Byte
Print "Init values:"
'Initialize RFM12

```