RFXCOM

RFXMeter

RF transmitter For Power/Gas/Water/Pulse metering

www.rfxcom.com

Table of Contents

1. RFXMeter FEATURES.

The RFXMeter is a device that has an RF transmitter and a microcontroller to count pulses from up to 3 metering modules.
The size of each counter is 24 bits so the maximum count is 16777215 pulses. The counter values of all 3 counters will be saved in non-volatile memory when the RFXMeter is disconnected from the power or a power loss occurs. The counter values are restored when the power returns.

The RFXMeter contains:

- 3 slots for metering modules,
- An RF transmitter with microcontroller to process the metering pulses,
- A 5V DC power supply for the transmitter, the microcontroller and the metering modules.

1.1. RFXPwr power metering module.

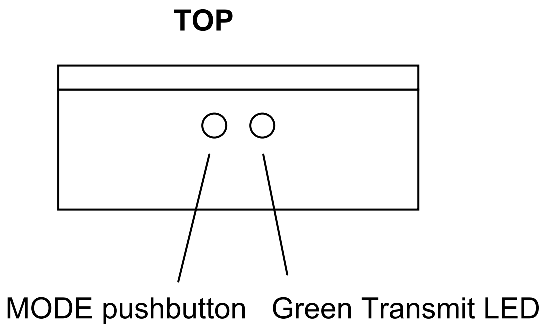
This module can be inserted in the RFXMeter and is able to measure the whole house power usage or a single phase for up to 150A.

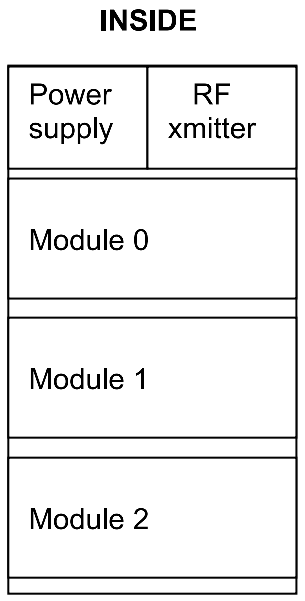
1.2. RFXPulse pulse metering module.

This module can be inserted in the RFXMeter and measures pulses coming from a relay or a photo transistor or a reflective-optical sensor. It can be used for example to count blinking LED's on an electricity meter or the mirror pass by on the Dutch gas meter.

2. Technical description of the RFXMeter unit.

2.1. Physical layout.





2.2. *Powering the RFXMeter.*

The RFXMeter has a power input connector for a power adaptor.
The adaptor output must be:
9V AC – 250mA
or 9 to 12V DC – 250mA (center pin = -, outer contact = +)
The center pin size is 2.1mm.

It is not necessary to use a power adaptor for the RFXMeter unit if an RFXPwr-module is installed at location for Module 0. In this case the 9V AC adaptor of the RFXPwr-module will be used to power the RFXMeter also.

2.3. *Power fuse in the RFXMeter.*

The RFXMeter has a 250mA fuse build in for the 5V circuit. The total maximum load for the 5V power supply is 200mA.

2.4. *Power usage.*

The RFXMeter uses 60mA from the 5V circuit so for the sub-modules a total of 140mA is allowed.

3. How to reset an RFXMeter.

Remove the power for at least 10 seconds.
The counter values will be saved in non-volatile memory when the power is removed.

4. How to install a module.

- Remove the power from the RFXMeter unit and open the enclosure.
- Insert the module.
- Close the enclosure.
- Connect the RFXMeter to the power.

5. Configure the RFXMeter modules in the software.

5.1. *Configure an RFXMeter module in Homeseer - ACRF.*

Use the RFreceiver program to find the ACRF-ID of the RFXMeter module(s).
The ACRF Device ID is the decimal number behind the slash.
In this example the sensor Device ID for the ACRF is 2296.

```
3008F8D25A1809 RFXMeter addr:08F8 = ACRF-ID:2296 RFXMeter: 1626714 bits=48
```

5.2. *Configure a module.*

If the RFXMeter is supported by a software product then check the documentation of this software how to configure the RFXMeter module.
If you want to write your own software to support the RFXMeter modules then have a look in the RFreceiver source how to decode the RFXMeter packets. This source file can be of help and can be found at the download page of www.rfxcom.com .

6. RFXMeter mode settings and calibration.

The RFXMeter has a MODE pushbutton to set:

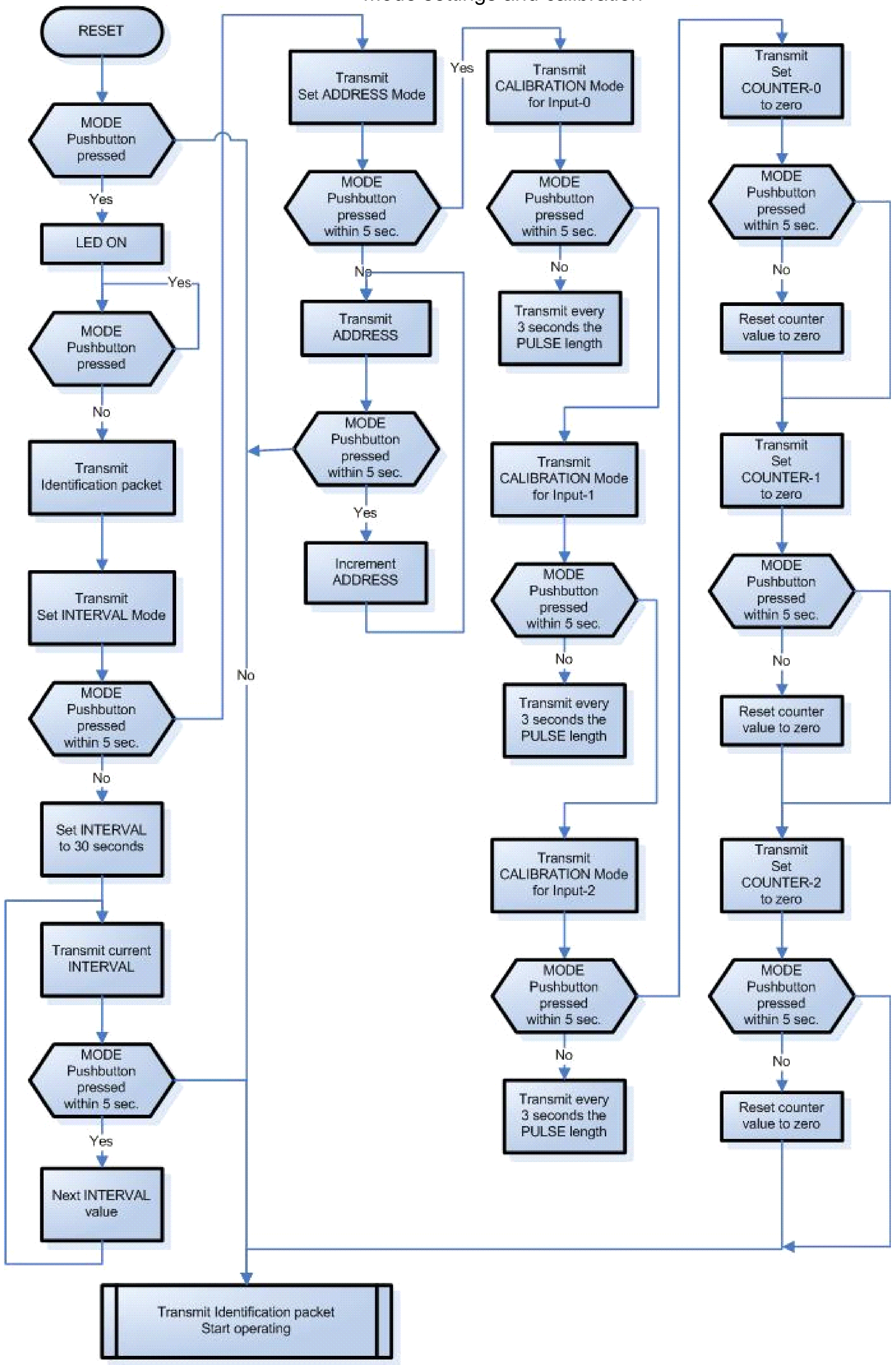
1. The transmit interval of the counter value(s),
2. Set the device address,
3. Enter calibration mode, (note: the device is already calibrated)
4. Reset the counter value(s) to zero.

To enter the settings or enable the calibration mode:

- Disconnect the power from the RFXMeter for at least 10 seconds,
- Hold the MODE pushbutton,
- Reconnect the power to the RFXMeter,
- The green LED will light indicating that the device has entered the settings mode.
- Release the MODE pushbutton.

The full process how to set the modes is displayed on the next page.

Mode settings and calibration


RESET
MODE Pushbutton pressed
Yes
LED ON
Yes
MODE Pushbutton pressed
No
Transmit Identification packet
Transmit Set INTERVAL Mode
MODE Pushbutton pressed within 5 sec.
No
Set INTERVAL to 30 seconds
Transmit current INTERVAL
MODE Pushbutton pressed within 5 sec.
Yes
Next INTERVAL value
No
Transmit Set ADDRESS Mode
MODE Pushbutton pressed within 5 sec.
Yes
No
Transmit ADDRESS
MODE Pushbutton pressed within 5 sec.
Yes
Increment ADDRESS
Transmit CALIBRATION Mode for Input-0
MODE Pushbutton pressed within 5 sec.
No
Transmit every 3 seconds the PULSE length
Transmit CALIBRATION Mode for Input-1
MODE Pushbutton pressed within 5 sec.
No
Transmit every 3 seconds the PULSE length
Transmit CALIBRATION Mode for Input-2
MODE Pushbutton pressed within 5 sec.
No
Transmit every 3 seconds the PULSE length
Transmit Set COUNTER-0 to zero
MODE Pushbutton pressed within 5 sec.
No
Reset counter value to zero
Transmit Set COUNTER-1 to zero
MODE Pushbutton pressed within 5 sec.
No
Reset counter value to zero
Transmit Set COUNTER-2 to zero
MODE Pushbutton pressed within 5 sec.
No
Reset counter value to zero
Transmit Identification packet Start operating

7. How to set the transmit Interval Rate.

- Start the RFreceiver program using an RFXCOM X10 receiver.
- Disconnect the power from the RFXMeter for at least 10 seconds.
- Hold the MODE pushbutton and connect the power.
- The green LED will light indicating that the device has entered the settings mode.
- Release the MODE pushbutton.
- The message **SET INTERVAL RATE** is transmitted.
- After 5 seconds the initial interval value of 30 seconds is set and transmitted.
- When the pushbutton is pressed again within 5 seconds the next interval value is set and the new value will be transmitted. The RFXMeter will return to operation when the pushbutton is not pressed for 5 seconds.

Interval Rates are: 30 seconds, 1 minute, 6, 12, 15, 30, 45 and 60 minutes.

8. How to set the base address of the RFXMeter device.

- Start the RFreceiver program using an RFXCOM X10 receiver.
- Disconnect the power from the RFXMeter for at least 10 seconds.
- Hold the MODE pushbutton and connect the power.
- The green LED will light indicating that the device has entered the settings mode.
- Release the MODE pushbutton.
- The message **SET INTERVAL RATE** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **SET ADDRESS mode** is transmitted.
- After 5 seconds a normal data packet with the current address is transmitted.
- When the pushbutton is pressed again within 5 seconds the address is incremented and the new value will be transmitted. The RFXMeter will return to operation when the pushbutton is not pressed for 5 seconds.

9. How to calibrate the RFXPwr module.

Note: The RFXPwr module is already calibrated in factory.

- Turn the potentiometer R12 a little to the right if the unit runs to slow.
- Turn the potentiometer R12 a little to the left if the unit runs to fast.

The next procedure can be used to calibrate the RFXPwr module using the software.
There must be a constant load of at least 8 Ampere (@230V 1840Watt, @120V 960Watt) and the power usage must be measured with a calibrated Watt meter.

- Start the RFreceiver program using an RFXCOM X10 receiver.
- Disconnect the power from the RFXMeter for at least 10 seconds.
- Hold the MODE pushbutton and connect the power.
- The green LED will light indicating that the device has entered the settings mode.
- Release the MODE pushbutton.
- The message **SET INTERVAL RATE** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **SET ADDRESS mode** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **CALIBRATION mode for Input-0** is transmitted.
- If RFXPwr module 0 must be calibrated wait until the calibration mode is entered otherwise press within 5 seconds the MODE pushbutton.
    - The message **CALIBRATION mode for Input-1** is transmitted.
    - If RFXPwr module 1 must be calibrated wait until the calibration mode is entered otherwise press within 5 seconds the MODE pushbutton.
    - The message **CALIBRATION mode for Input-2** is transmitted.
    - Wait until the calibration mode for RFXPwr module 2 is entered.
- Adjust the potentiometer until the power usage measured by the RFXPwr module is equal to the power usage on the calibrated Watt meter.
- When finished, remove the power from the RFXMeter.

10. How to reset the counter value for a module.

- Disconnect the power from the RFXMeter for at least 10 seconds.
- Hold the MODE pushbutton and connect the power.
- The green LED will light indicating that the device has entered the settings mode.
- Release the MODE pushbutton.
- The message **SET INTERVAL RATE** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **SET ADDRESS mode** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **CALIBRATION mode for input-0** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **CALIBRATION mode for input-1** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **CALIBRATION mode for input-2** is transmitted.
- Press within 5 seconds the MODE pushbutton.
- The message **COUNTER for input-0 will be set to zero** is transmitted.
- If the counter for input-0 must be set to zero wait 5 seconds otherwise press the MODE pushbutton.
    - The message **COUNTER for input-1 will be set to zero** is transmitted.
    - If the counter for input-1 must be set to zero wait 5 seconds otherwise press the MODE pushbutton.
    - The message **COUNTER for input-2 will be set to zero** is transmitted.
    - If the counter for input-2 must be set to zero wait 5 seconds otherwise press the MODE pushbutton.
- The RFXMeter will return to operation now.

11. RFXMeter RF data format.

Message length 48 bits (decimal)

```
|    Byte 0     |    Byte 1     |    Byte 2     |    Byte 3     |    Byte 4     |    Byte 5     |
 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
 4               4                   3                   2                   1
 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
 < - - - - - - - address - - - - - > < - - - - - - - - - counter value - - - - - - - - > < - - - > <parity>
                                 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 2 2 2 2 1 1 1 1    |
                                 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 3 2 1 0 9 8 7 6    packet type
```

2 bytes address. Byte 2 = byte 1 with the complement of the upper nibble (bit 7-4). This way max 256 metering modules can be used.

Counter value for an RFXPwr module measures with an accuracy of 0.001kWh and thus a value from 0 to 16777.215 kWh.

Packet type:

0000	normal data packet		
0001	new interval time set.		
	Byte 2	0x01	30 seconds
		0x02	1 minute
		0x04	6 minutes (RFXPower = 5 minutes)
		0x08	12 minutes (RFXPower = 10 minutes)
		0x10	15 minutes
		0x20	30 minutes
		0x40	45 minutes
		0x80	60 minutes
0010	calibrate value in <counter value> in µsec.		
0011	new address set		
0100	counter value reset to zero		
1011	counter value set		
1100	set interval mode within 5 seconds		
1101	calibration mode within 5 seconds		
1110	set address mode within 5 seconds		
1111	identification packet		
	Byte 2 = firmware version		
		0x00 – 0x3F = RFXPower	
		0x40 – 0x7F = RFU	
		0x80 – 0xBF = RFU	
		0xC0 – 0xFF = RFXMeter	
	Byte 3 = interval time (see packet type 0001)		

4 bits parity. This is the complement of:

byte 0 bit 7-4 + byte 0 bit 3-0 + byte 1 bit 7-4 + byte 1 bit 3-0
+ byte 2 bit 7-4 + byte 2 bit 3-0 + byte 3 bit 7-4 + byte 3 bit 3-0
+ byte 4 bit 7-4 + byte 4 bit 3-0 + byte 5 bit 7-4

12. VB.NET example

```
If recbits = 48 Then 'if packet length is 48 bits check if RFXMeter
   parity = Not ((recbuf(0) >> 4) + (recbuf(0) And &HF) _
            + (recbuf(1) >> 4) + (recbuf(1) And &HF) _
            + (recbuf(2) >> 4) + (recbuf(2) And &HF) _
            + (recbuf(3) >> 4) + (recbuf(3) And &HF) _
            + (recbuf(4) >> 4) + (recbuf(4) And &HF) _
            + (recbuf(5) >> 4)) And &HF
   If (parity = (recbuf(5) And &HF)) And _
          (recbuf(0) + (recbuf(1) Xor &HF) = &HFF) Then
      rfxpower = True
   End If
End If


Sub processrfxmeter()
    Dim measured_value As Single
    WriteMessage("           RFXMeter[" & (recbuf(0) * 256 + recbuf(1)).ToString & "]M", False)
    WriteMessage(" RFXMeter addr:" & VB.Right("0" & Hex(recbuf(0)), 2), False)
    WriteMessage(VB.Right("0" & Hex(recbuf(1)), 2), False)
    WriteMessage(" ID:" & Convert.ToString(recbuf(1) + (recbuf(0) * 256)) & " ", False)
    Select Case recbuf(5) And &HF0
        Case &H0
            measured_value = ((recbuf(4) * 65536) + (recbuf(2) * 256) + recbuf(3))
            WriteMessage("RFXMeter: " & Convert.ToString(measured_value), False)
            WriteMessage(";  RFXPower: " & Convert.ToString(measured_value / 100) & " kWh", False)
            WriteMessage(";  RFXPwr-Module: " & Convert.ToString(measured_value / 1000) & " kWh", False)
        Case &H10
            WriteMessage("Interval: ", False)
            Select Case recbuf(2)
                    Case &H1
                        WriteMessage("30 sec.", False)
                    Case &H2
                        WriteMessage("1 min.", False)
                    Case &H4
                        WriteMessage("6 (old=5) min.", False)
                    Case &H8
                        WriteMessage("12 (old=10) min.", False)
                    Case &H10
                        WriteMessage("15 min.", False)
                    Case &H20
                        WriteMessage("30 min.", False)
                    Case &H40
                        WriteMessage("45 min.", False)
                    Case &H80
                        WriteMessage("60 min.", False)
                    Case Else
                        WriteMessage("illegal value", False)
                End Select
        Case &H20
                Select Case (recbuf(4) And &HC0)
                    Case &H0
                        WriteMessage("Input-0 ", False)
                    Case &H40
                        WriteMessage("Input-1 ", False)
                    Case &H80
                        WriteMessage("Input-2 ", False)
                    Case Else
                        WriteMessage("Error, unknown input ", False)
            End Select
            measured_value = (((recbuf(4) And &H3F) * 65536) + (recbuf(2) * 256) + recbuf(3)) / 1000
            WriteMessage("Calibration: " & Convert.ToString(measured_value) & "msec ", False)
            If measured_value <> 0 Then
                WriteMessage("RFXPower= " & Convert.ToString(Round(1 / ((16 * measured_value) / (3600000 /
100)), 3)) & "kW", False)
                WriteMessage(" RFXPwr= " & Convert.ToString(Round(1 / ((16 * measured_value) / (3600000 /
62.5)), 3)) & "|" & Convert.ToString(Round((1 / ((16 * measured_value) / (3600000 / 62.5))) * 1.917, 3)) &
"kW", False)
            End If
        Case &H30
            WriteMessage("New address set", False)
        Case &H40
            Select Case (recbuf(4) And &HC0)
                Case &H0
                    WriteMessage("Counter for Input-0 will be set to zero within 5 seconds OR push MODE
button for next command.", False)
                Case &H40
```

```
                    WriteMessage("Counter for Input-1 will be set to zero within 5 seconds OR push MODE
button for next command.", False)
                Case &H80
                    WriteMessage("Counter for Input-2 will be set to zero within 5 seconds OR push MODE
button for next command.", False)
                Case Else
                    WriteMessage("Error, unknown input ", False)
            End Select
        Case &H50
            WriteMessage("Push MODE push button within 5 seconds to increment the 1st digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &H60
            WriteMessage("Push MODE push button within 5 seconds to increment the 2nd digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &H70
            WriteMessage("Push MODE push button within 5 seconds to increment the 3rd digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &H80
            WriteMessage("Push MODE push button within 5 seconds to increment the 4th digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &H90
            WriteMessage("Push MODE push button within 5 seconds to increment the 5th digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &HA0
            WriteMessage("Push MODE push button within 5 seconds to increment the 6th digit.", False)
            measured_value = (recbuf(2) >> 4) * 100000 + (recbuf(2) And &HF) * 10000 + (recbuf(3) >> 4) _
            * 1000 + (recbuf(3) And &HF) * 100 + (recbuf(4) >> 4) * 10 + (recbuf(4) And &HF)
            WriteMessage("Counter value = " & VB.Right("00000" & Convert.ToString(measured_value), 6),
False)
        Case &HB0
                Select Case recbuf(4)
                    Case &H0
                        WriteMessage("Counter for Input-0 reset to zero.", False)
                    Case &H40
                        WriteMessage("Counter for Input-1 reset to zero.", False)
                    Case &H80
                        WriteMessage("Counter for Input-2 reset to zero.", False)
                    Case Else
                        WriteMessage("protocol error.", False)
                End Select
        Case &HC0
            WriteMessage("Enter SET INTERVAL RATE mode within 5 seconds OR push MODE button for next
command.", False)
        Case &HD0
            Select Case (recbuf(4) And &HC0)
                Case &H0
                    WriteMessage("Enter CALIBRATION mode for Input-0 within 5 seconds OR push MODE button
for next command.", False)
                Case &H40
                    WriteMessage("Enter CALIBRATION mode for Input-1 within 5 seconds OR push MODE button
for next command.", False)
                Case &H80
                    WriteMessage("Enter CALIBRATION mode for Input-2 within 5 seconds OR push MODE button
for next command.", False)
                Case Else
                    WriteMessage("Error, unknown input ", False)
            End Select
        Case &HE0
            WriteMessage("Enter SET ADDRESS mode within 5 seconds OR push MODE button for next command.",
False)
        Case &HF0
            If recbuf(2) < &H40 Then
                WriteMessage("RFXPower Identification,", False)
            ElseIf recbuf(2) < &H80 Then
                WriteMessage("RFXWater Identification,", False)
            ElseIf recbuf(2) < &HC0 Then
```

```
                WriteMessage("RFXGas Identification,", False)
            Else
                WriteMessage("RFXMeter Identification,", False)
            End If
            WriteMessage(" Firmware Version: " & VB.Right("0" & Hex(recbuf(2)), 2), False)
            WriteMessage(", Interval rate: ", False)
            Select Case recbuf(3)
                    Case &H1
                        WriteMessage("30 seconds", False)
                    Case &H2
                        WriteMessage("1 minute", False)
                    Case &H4
                        WriteMessage("6 minutes", False)
                    Case &H8
                        WriteMessage("12 minutes", False)
                    Case &H10
                        WriteMessage("15 minutes", False)
                    Case &H20
                        WriteMessage("30 minutes", False)
                    Case &H40
                        WriteMessage("45 minutes", False)
                    Case &H80
                        WriteMessage("60 minutes", False)
                    Case Else
                        WriteMessage("illegal value", False)
                End Select
        Case Else
            WriteMessage("illegal packet type", False)
        End Select
    End Sub
```

13. DIY options.

The RFXMeter device has 3 slots for RFXCOM modules.
The user can also create his design for a metering module.

13.1. *Header connections:*

JP0, JP1, JP2

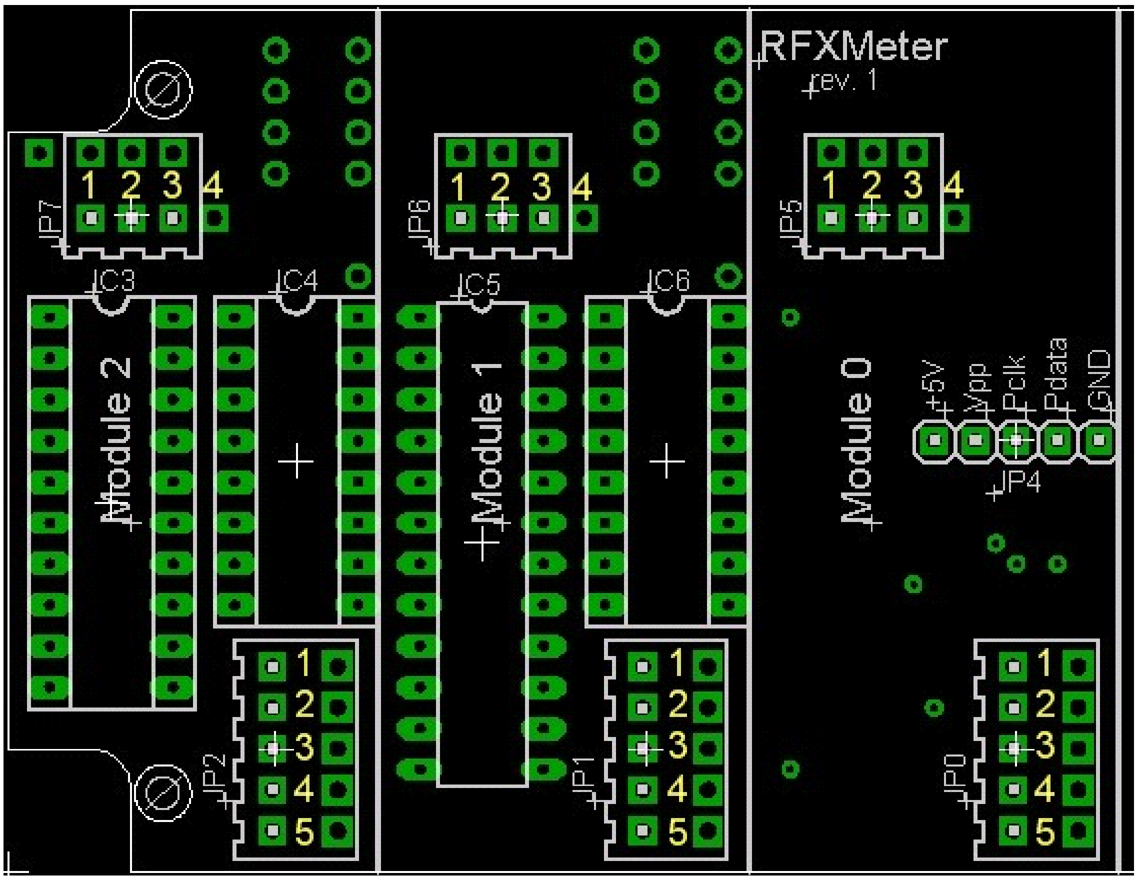
Pin 1	counter input to the RFXMeter
Pin 2	not connected
Pin 3	connected to Pin 3 of JP0, JP1, JP2.
Pin 4	Ground
Pin 5	connected to Mode pushbutton.

JP5, JP6, JP7

Pin 1	+5 Volt in from RFXMeter power supply (total for all 3 modules is max 140mA)
Pin 2	Ground
Pin 3	JP5 is 9V AC output to RFXMeter power supply. JP6, JP7 not connected
Pin 4	not part of JP but connected to voltage input of the RFXMeter power supply.

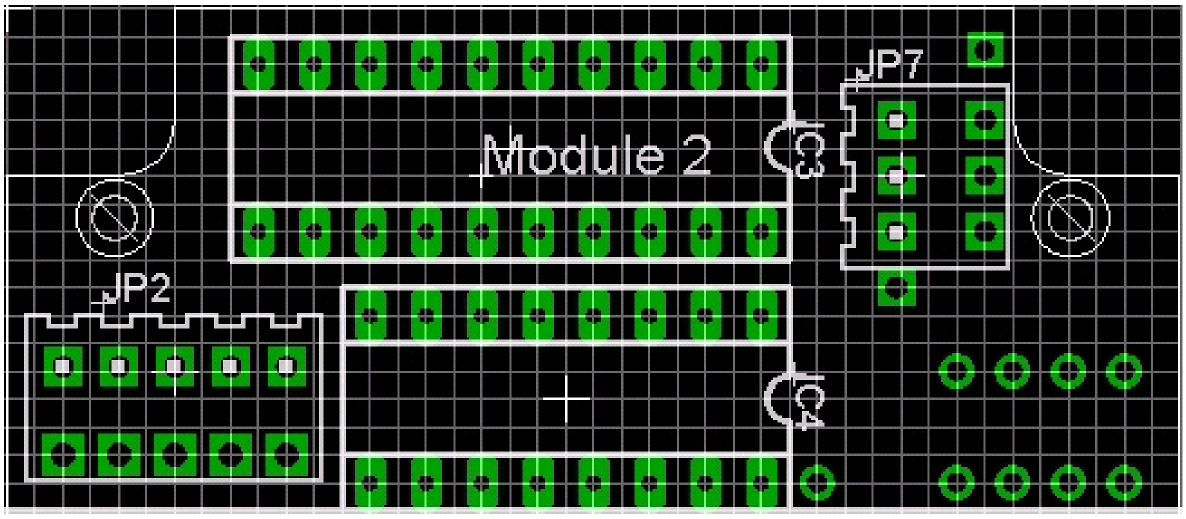
13.2. *Board layout.*

It is also possible to use the free IC3, IC4, IC5 and IC6 locations.



13.3. Module dimensions.

Grid used is 0.05 inch.
Module dimensions 0.9 inch x 2.1 inch



14. Warnings:

RF signals are possible disturbed and it has not been justified for this equipment at uses in circumstances where life-threatening or dangerous situations are possible.

15. Copyright notice.

All materials contained in this document are protected by copyright laws, and may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of RFXCOM.

16. Revision history.

Version 3.0 - August 2, 2007
RFXPwr module description moved to a dedicated RFXPwr document.
Text layout changed.

Version 4.0 - July, 2009
VB.NET example code added