
wireless control for everyone

# ONE-NET EVALUATION MODULE REFERENCE GUIDE

**Document Version 2.3.0**

**August 14, 2012**

# Table of Contents

# 1 Overview

The ONE-NET Evaluation Module is designed to provide convenient hands-on experience with the MAC layer of the ONE-NET protocol. The module is delivered with an on-board implementation of the serial interface protocol defined in the separate *Command-Line Specification* document.

## 1.1 ONE-NET Device Overview

- All ONE-NET based implementations are called `devices.`

- `Devices` are either `Masters` or `Clients`.

- A `Master` or `Client` can contain up 16 logical, addressable `Units`.

- The ONE-NET Evaluation Module described in this document contains 4 definable User I/O Pins as addressable `Units`.

# 2 Components

## 2.1 Included

- This Evaluation Module circuit board is based on a Renesas R8C/23 processor and an Analog Devices ADF7025 ISM band FSK transceiver.

- Separate 915MHz or 868MHz monopole RF antenna, which must be attached to the Antenna (J6) connector on the circuit board.

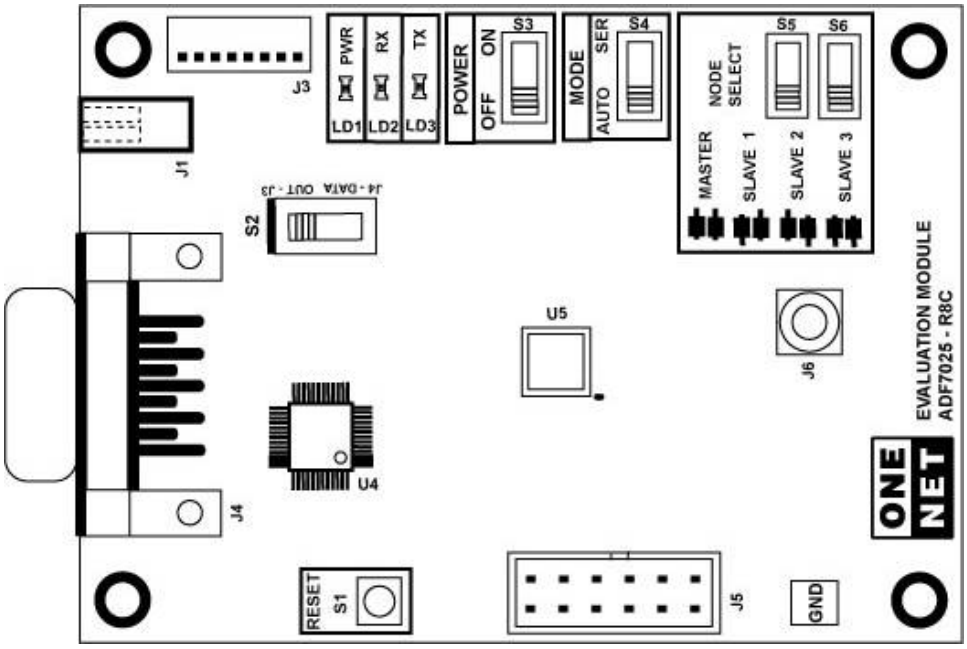
- Quad battery pack, with detachable cable, which must be attached to the Power (J1) connector on the circuit board.

## 2.2 User-Supplied

- Four (4) AA-sized batteries.

- Computer with any standard terminal emulation software (such as the *HyperTerminal* accessory provided with *Microsoft Windows, Tera Term, or Minicom for Linux*).

**Evaluation Board Diagram:**



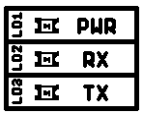
Standard serial communication cable with 9-pin connectors (male to female), which must be attached to the Serial (J4) connector on the circuit board, and to a serial port on the computer.

# 3 Module Characteristics

The ONE-NET Evaluation Module can be reprogrammable with different features. The RAM and code space resources are insufficient to support all features. People with access to a Renesas HEW IDE and E8 or E8A chip programmer can experiment with different configuration options.. Please see the ***ONE-NET CONFIGURATION OPTIONS, FEATURES, AND PORT CONSTANTS GUIDE*** document for a list of configuration options. The default configuration does not include Block And Stream Messaging. The Evaluation Module uses an RF data rate of 38,400 bits/s. A 115,200 bits/s data rate is used for the serial (terminal) connection. The serial data rate bitrate is one of the options that can be configured when reprogramming the Evaluation Module. When configured as a Master, the Device can manage up to five (5) Clients; its identifier (DID) is fixed as 0x001. The Device implements four (4) Units, each of which can operate as either an input or an output; Unit addresses are fixed as 0x00–0x03. All application layer data messages are output via the serial connection.

## 3.1 Status LEDs

Three light-emitting diodes (LEDs) are used to display Device status information; they are defined as follows:




### 3.1.1 PWR (LD1)

LED is illuminated whenever power is applied to connector J1, and the POWER switch is set to the ON position.

### 3.1.2 RX (LD2)

LED flashes whenever data is being received.

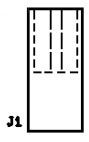
### 3.1.3 TX (LD3)

LED flashes whenever data is being transmitted.

## 3.2 Connectors

### 3.2.1 Power (J1)

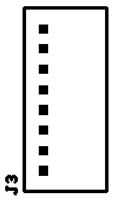
Coaxial DC center-terminal-positive connector (3.5mm outside diameter, 1.0mm inside diameter), for power source in the range 3.8–16.0VDC (60ma maximum draw)



### 3.2.2 Aux. Power (J2)

Unused in current implementation. This jack is not installed on the Evaluation Modules. The board is designed to allow a 2-pin 1.5mm connector (JST part #B2B-ZR) for solder side power connections.

### 3.2.3 Auxiliary (J3)

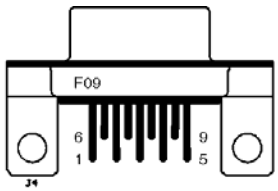



8-pin 1.5mm pitch white right-angle header (JST part #S8B-ZR); for matching connector, use JST part #ZHR-8. Electrical connections for J3 are defined in Table 1: ***Auxiliary Connector Pin Assignments***

| Pin | Function | I/O | |
|---|---|---|---|
| 8 | User I/O Pin 0 (Unit 0) | CLI Configurable | Pins 3-8 use standard 3.3V digital logic levels. |
| 7 | User I/O Pin 1 (Unit 1) | CLI Configurable | |
| 6 | User I/O Pin 2 (Unit 2) | CLI Configurable | |
| 5 | User I/O Pin 3 (Unit 3) | CLI Configurable | |
| 4 | UART RX | Input | |
| 3 | UART RX | Output | |
| 2 | Power (3.8-16DC) | Power Input | |
| 1 | Ground | Ground | |

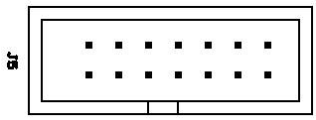
***Table 1: Auxiliary Connector Pin Assignments***

### 3.2.4 Serial I/O (J4)




Standard (DB-9) 9-pin female type, for RS232 serial communication connection. The communication is standard level shifted bipolar RS-232 signaling.

### 3.2.5 Programmer (J5)




14-pin blue vertical header, for connection to a Renesas device programming tool such as the E8 Emulator

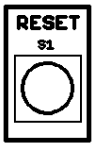
### 3.2.6 Antenna (J6)


A Coaxial (SMA) screw-type connector, for a 915MHz or 858MHz antenna (a monopole antenna is supplied with the board). This connector allows the user to either remote locate an antenna or

experiment with other antenna types.

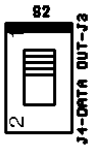
## 3.3 Switches

### 3.3.1 Reset (S1)



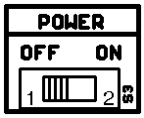

The RESET push-button switch resets the microprocessor (losing any unsaved data) and restarts the Device. It is the functional equivalent of turning off, and then reapplying, power to the circuit board.

### 3.3.2 Data Out (S2)




When the DATA OUT switch (near the serial connector) is in the J4 position (towards the serial I/O connector), serial I/O is performed via the Serial (J4) connector. When the switch is in the J3 position (down), serial I/O is performed via the Auxiliary (J3) connector. Note that the signal levels on the Auxiliary connector are not the same as on the Serial connector.

### 3.3.3 Power (S3)

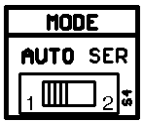



The POWER switch simply controls the power supply to the circuit board; when it is in the OFF position, all power is disconnected.

### 3.3.4 Mode (S4)

The MODE switch selects one of two operating modes for the Device.

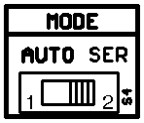
#### *3.3.4.1 Auto*




The AUTO position invokes a preconfigured board state, one which defines a network with one (1) Master and three (3) Client Devices. Clients are assigned DIDs 0x002, 0x003, and 0x004 (as determined by the NODE SELECT switches). It is important that the NODE SELECT switches be correctly set when in Auto mode, as they are used without evaluation to assign the DIDs (unlike normal assignment via the protocol), and duplicate DIDs can result in addressing errors.

The Master automatically attempts to send one text packet (containing “11”, “22”, or “33”) every two and a half seconds to each Client; if user (serial) input is detected, the Master delays sending this message for twenty (20) seconds. Device removal, and any commands (such as `channel`, `join`, `sniff`, etc.) that change operating modes, are disallowed in AUTO mode.

### *3.3.4.2 Serial*

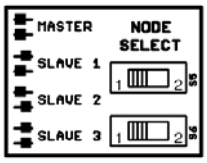



The SER (serial) position invokes a preconfigured board state with an empty ONE-NET network. In this state, each Device will start up just as in a regular network, first examining the local data ROM to see if a saved instance *of the same Device type* (i.e., Master or Client) exists. If so, it loads the saved network parameters and resumes operation; otherwise, it initiates the proper join process.

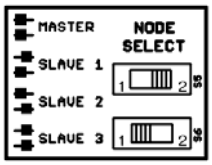
The Master will scan channels until it finds a clear channel (or the most clear channel), and then sets up a network on that channel. Clients will automatically scan channels, looking for a Master Invite New Client packet. The first Client to join the network receives DID 0x002, the next 0x003, etc. The user may also select a new channel using the `channel` command.

## 3.3.5 Node Select (S5, S6)

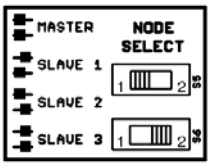
A pair of node select switches specifies one of four optional Master/Client states for the Device. When set as illustrated on the circuit board, they determine whether the Device is to operate as a Master (`MASTER`) or as one of three possible Clients (`SLAVE 1`, `SLAVE 2`, or `SLAVE 3`). In serial mode, the three Client switch positions are equivalent.



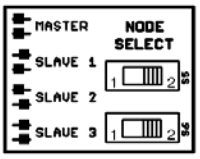

Master




Client (Slave 1)




Client (Slave 2)




Client (Slave 3)

# 4 Command-Determined Modes

Please see the ***Command-Line Specification*** and ***ONE-NET Debugging Tools And Techniques*** documents for more details on how to implement the options / modes listed below.

## 4.1 Packet Sniffer

This state (which removes the Device from the ONE-NET network) is invoked by the sniff command, which turns the Device into a packet-sniffing node. All data bytes for each received packet on a given channel are output via the serial connection. Decoding, decrypting, and packet details can be adjusting using the verbose level command (see below).

When the board is in packet-sniffing mode, bytes received over the RF interface will be output only when there is no user input to the Command Line Interpreter. Once the user types something, packet-sniffing mode is suspended until the serial input data is cleared (i.e., when the user types "Enter" or deletes the input line), at which point it resumes.

## 4.2 Verbosity Level

This state is controlled by the `verbose level` command. Higher verbosity levels result in more detailed printouts. Please see the ***Command-Line Specification*** and ***ONE-NET Debugging Tools And Techniques*** documents for more details

## 4.3 Echo

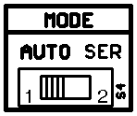
This state is controlled by the `ECHO` command, which enables/disables command echoing for the Device. The default state is **on**.

# 5 Working with The Evaluation boards

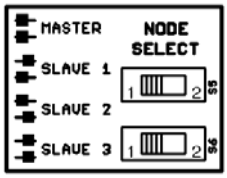
## 5.1 Initial self-test: The "auto" mode

All Evaluation modules are supplied with identical hardware and identical software content. For purposes of the following descriptions, the board that is configured as a Master (both `Node Select` switches set to their left positions) is denoted the `Master`. A board with any other `Node Select` switch settings is denoted a `Client`. Note: Other than a power supply for each, no external connections are necessary for the self-test.

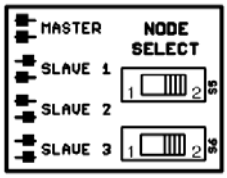
**The Setup:**




Set the `Mode` switches on both boards to **Auto**




Set the `Node Select` switches on one board to their left positions. This board will be the `Master`.




Set the `Node Select` switches on the other board to their right positions. This board will be the `Client`.

**The Test:**

1. Apply power to the `Master`. The "`PWR`" LED will be illuminated as long as power is applied.
2. After about a second, the "TX" led is dimly lit, then the "`RX`" led on the `Master` is lit for about a half second. You may see a slight flickering at this point, which is normal.

3. Apply power to the `Client`. The "`PWR`" LED will be illuminated as long as power is applied.
4. After about a second, the "TX" led is dimly lit, then the "`RX`" led on the `Master` is lit for about a half second. You may see a slight flickering at this point, which is normal. If no master is present, the LEDs may flash for a few seconds and then will remain off. If a `Master` is present and the `Client` can receive its signal, the "`RX`" LED and the "`TX`" led will flash alternately.
5. If `Master` and `Client` are working properly, "`TX`" and "`RX`" LEDs on both units will flash alternately.
6. For one `Master and fewer than three Clients`, the `Master` will show more "`TX`" than "`RX`", and the client will show more "`RX`" than "`TX`". This is because the Master sending to non-existent devices. The Client devices should always have their "RX" LEDs blink more than their "TX" LEDs blink because they will hear all messages to other clients but only respond to messages to themselves.
7. It is possible to have up to three different `Clients`. (There will always be exactly one master.) Each client should have its own Device ID, as determined by the "Node Select" switch settings on each Evaluation Module.
8. When Client devices are attached to a desktop computer via serial cable, you should see regular messages output.
    1. Client 1 (device 002) should output "Received text from 001: 11"
    2. Client 2 (device 003) should output "Received text from 001: 22"
    3. Client 3 (device 004) should output "Received text from 001: 33"
9. The "`Auto`" mode can be used to determine the operational limits of the range of the devices in your environment. With `Master` and `Client` units in close proximity, apply power and verify that "`TX`" and "`RX`" lights on all units are flashing alternately.
   Then move one of the units farther and farther away. Successful communications are verified if the printouts above take place. If the "RX" LEDs blink, that means that the devices can at least hear each other.
10. The channel used in Auto Mode is US Channel 2 (904 MHz) if US Channels are used.
11. The channel used in Auto Mode is European Channel 2 (866.5 MHz) if US Channels are not used.
12. The data rate used in Auto Mode is 38,400 bits per second.
13. The Serial Connection bitrate is 115,200 bits per second.

## 5.2 Functional Tests

Before performing the first test, it is recommending that the basic functionality be verified using the "Auto" Mode initial self test, as described in the preceding paragraph.

For purposes of the following descriptions, the board that is configured as a Master (both Node Select switches set to their left positions) is denoted the Master. A board with any other Node Select switch settings is denoted a Client.

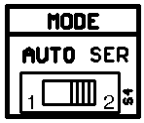
Upon initial startup, user pins 1 and 3 of a board that is configured as a Master are automatically set to be inputs and user pins 0 and 2 of a Master are automatically set to be outputs. Client devices are opposite.

| Default Settings | Master | Client |
|---|---|---|
| User Pin 0 | output | input |
| User Pin 1 | input | output |
| User Pin 2 | output | input |
| User Pin 3 | input | output |

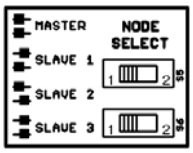
The pin direction configurations of an Evaluation Module can be changed and saved from the Command-Line Interface, using an RS232 terminal connected to that module

**The Setup:** Two evaluation boards are needed for this test. You can use serial interace programs such as Tera Term, Microsoft's *Hyperterminal* or Linux's *Minicom* on your workstation. The RS-232 port setup is 38,400 bits/second, 8 data bits, one stop bit, and no parity. The software supplied with the Evaluation Modules does not support flow control.

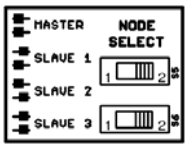
1. Set both evaluation boards to "Serial" mode.




2. Select one board to be the `Master` and the other the `Client` using the Node Select switches.

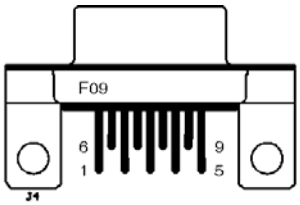



Master




Client

3. Attach an RS-232 terminal to the serial port of the board that is configured as the `Master`.




4. On the board that is configured as the `Master`, connect a switch between user pin 1 and ground. Note that battery life may be somewhat extended by connecting a 1K Ohm resistor in series with the switch.
5. Connect the anode of an LED to user pin 3 of the `Master` through a 150-390 Ohm resister.
6. Connect the cathode of the LED to the ground on the `Master`.
7. On the board that is configured as a `Client`, connect a switch between user pin 3 and ground. Note that battery life may be somewhat extended by connecting a 1K Ohm resistor in series with the switch.
8. Connect the anode of an LED to user pin 1 of the `Client` through a 150-390 Ohm resister.
9. Connect the cathode of the LED to the ground on the `Client`.
10. Apply Power to both boards, and set the `Power` switch to the **On** position.

## 5.3 Functional Test 2 – Terminals To Both Devices

In the following discussion, information printed by the program looks like this:

```
ONE-NET Evaluation Version 2.3.0 (Build 107)
SERIAL MODE
ocm-m>
```

If you want to use a specific channel for the network, specify that channel on the master. Otherwise, if you don't care, the master will randomly select a channel from a list of channels with low amounts of traffic. As of August 14, 2012, valid locales are "US" for the United States and "EUR" for Europe. Valid U.S. channels are 1 through 25. Valid European channels are 1 through 3.

```
channel:US:5
```

Information entered by the user looks like this:

```
invite:2222-2222
```

The initial prompt looks like this:

```
ONE-NET Evaluation Version 2.3.0 (Build 107)
SERIAL MODE
ocm-m>
```

Now, the "`invite`" command is used to get other devices to join the network. Evaluation Modules configured as `Clients` will respond to their unique invite code which is located on the bottom of the board. Note: The `invite` can also be retrieved by an RS-232 terminal to the serial port of the board that is configured as the `Client and using the` LIST command.

To invite a device to join the network, type the following command:

`ocm-m>` **`invite:xxxx-xxxx`** (the xxxx-xxxx represents the unique invite code of a

client, which is case sensitive)

Note that the "`invite`" command requires the word "invite", a colon, and a Device Invite Key consisting of two groups of four hex digits separated by a hyphen. It must be exactly as shown. Different devices will have different Device Keys. The example shows the key for an Evaluation Module that is configured as a `Client`.

When you issue an "`invite`" command, the `TX` LED on the master flashes periodically until it hears a response from the device that was invited. The `Client` device is scanning all frequencies within the ONE-NET range, and it may take several seconds up to a minute for it to receive the particular frequency being used by the `Master` and respond. Note that you can optionally specify a channel on the client, which makes the invite process much quicker. On the client, type…

```
ocm-c> join:US:5
```

The client will no longer listen on random channels, but will instead only listen on US Channel 5. Obviously this needs to match the Master's channel. At any time, type the list command on either device to get pertinent information.

If the `Master` doesn't get a response within about 10 minutes, it will terminate the "invite" process with a message that the command failed.

If a Client does respond and joins the network, you will see the following on the Client:

```
Successfully joined network as 002
```

And on the Master:

```
Updating ADD DEVICE on 002 succeeded.
```

From now on, the `Client` device will be known to the `Master` by its Device ID (DID) of 002.

Now, to let the client know that its input unit number 2 is to be connected to the Master's output unit number 2, you issue a "assign peer" command. Note: please see the ***ONE-NET Peer Messaging*** document for more information on ONE-NET peer assignments.

An "assign peer" command consists of the words "assign peer" followed by a colon and the Device ID of the source of the connection, another colon, and an input unit number on the source device . Then, after another colon you enter the DID of the destination device and an output unit number on the destination device.

So, to connect input unit number 3 on the `Client` device that was just added to the network to output unit number 2 on the `Master` device, you enter:

```
ocm-m> assign peer:002:2:001:2
```

If the command is correct and if Device with DID 002 accepts the assignment, you will see:

```
ocm-m> assign peer:002:2:001:2
OK

ocm-m> Updating ASSIGN PEER on 002 succeeded
```

The "OK" means that the command was accepted at the `Master`, and the following line means that the `Client` accepted the assignment after receiving the packet. The "list" command on the Client will reflect the new peer assignment.

```
Peer Table:
  002:2:001:2
```

To assign unit number 3 on the Master to unit number 1 on the Client that has joined the network, you would enter the following:

```
ocm-m> assign peer:001:3:002:1
OK
```

Note that no messages are sent, so the RX and TX LEDs will not blink. That's because the peer assignment is to the Master and the command was typed in on the Master. Since the device that needs to be informed is the device that the command was typed into, no message passing is required. A "list" command on the Maser reflects this new peer assignment.

```
Peer Table:
  001:3:002:1
```

Now, whenever input unit 2 on the `Client` changes state, a "switch status changed" message is sent to Unit 2 of the `Master` so that it can act appropriately (set output pin 2's state to the Client's pin 2 state). Whenever input unit 3 on the `Master` changes state, a "switch status changed" message is sent to the `Client` and the `Client` acts appropriately (set output pin 1's state to the Master's pin 3 state).

With the software supplied with the Evaluation Modules, using peer assignments as described above, whenever the logic value of an input pin changes from zero to one, the output pin on the target device goes from zero to one, and vice versa.

Bottom line: if everything works up to this point, flipping a switch connected to user pin 3 of the `Master` causes the LED connected to user pin 1 of the `Client` to change. Flipping a switch connected to user pin 2 of the `Client` causes the LED connected to user pin 2 of the `Master` to change.

At this point, nothing has been saved in either unit's non-volatile memory, so if you power-cycle (or reset) both units, you will have to go through the entire process again.

Now, flip the switch connected to user pin 2 of the Client on, then off. Here's what you should see. We'll assume that unit 2 is currently off on both devices. The list command produces the following on the Master. See red text in particular

```
 ocm-m> list
ONE-NET Evaluation Version 2.3.0 (Build 107)
# of Network MH Devices : 2
# of Network MH Repeaters : 1
Message key : (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-
0e-0f)
Old Message key : (00-00-00-00) - (00-01-02-03) - (04-05-06-07) - (08-
09-0a-0b)

NID: 0x000000001
DID: 0x001

Channel: US 5
Client count: 1

  Client 1 : DID: 0x002

Peer table:
  001:3:002:1

User pins:
  0 output state: 0
  1 input state: 0
  2 output state: 0
  3 input state: 1
```

And on the Client.

```
 ocm-m> list
ONE-NET Evaluation Version 2.3.0 (Build 107)
Invite code: 2222-2222
# of Network MH Devices : 2
# of Network MH Repeaters : 1
Message key : (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-
0e-0f)
Old Message key : (00-00-00-00) - (00-01-02-03) - (04-05-06-07) - (08-
09-0a-0b)

NID: 0x000000001
DID: 0x002

Channel: US 5
Keep-Alive Interval:1800000 ms
Send To Master: True
Reject Bad Msg ID: True
Peer table:
  002:2:001:2

User pins:
  0 input state: 0
  1 output state: 0
  2 input state: 0
  3 output state: 1
```

The relevant portions are the following.

**Master**

```
Peer table:
  001:3:002:1
User pins:
  0 output state: 0
  1 input state: 0
   2 output state: 0
  3 input state: 1
```

**Client**

```
Peer table:
  002:2:001:2
User pins:
  0 input state: 0
  1 output state: 0
   2 input state: 0
  3 output state: 1
```

The relevant peer entry and switch and relay pairing are in red (switch / input pin on the Client and relay / output pin on the Master). Note that the source unit of the peer entry and the recipient is an output. Note also that the two pins have the same value (off). Flipping switch 2 of the client on yields the following.

**Client Output**

```
Pin 2 has changed to state 1
Single transaction with 001; return status: SUCCESS
Unit 2 of 001 has state 1.
```

**Master Output**

```
Unit 2 of 002 has state 1.
Pin 2 has changed to state 1.
```

The list command confirms the change, as should any LED attached to the Master's unit 2.

**Master**

```
Peer table:
  001:3:002:1
User pins:
  0 output state: 0
  1 input state: 0
   2 output state: 1
  3 input state: 1
```

**Client**

```
Peer table:
  002:2:001:2
User pins:
  0 input state: 0
  1 output state: 0
   2 input state: 1
  3 output state: 1
```

Flipping the switch off again will yield the same results, only this time the pin values will once again be 0 / off.

Flipping a switch attached to the Master's pin 3 will yield the same results, only this time, the Client's unit 1 will be the unit changing.

You can also set a pin from the command line. From the client.

```
ocm-c> set pin:001:1
```

Pin 2 of the Master will turn on.

You can also check / query a pin's value from the command line. On the Client…

```
ocm-c> fast query pin:001:2
OK
ocm-c> Single transaction with 001; return status: SUCCESS
Unit 2 of 001 has state 1.
```

Perhaps the Master wants to send device 002 the value of pin 2. On the Master…

```
ocm-m> status pin:002:2
OK
ocm-m> Single transaction with 002; return status: SUCCESS
```

The resulting message on the Client.

```
ocm-m> Unit 2 of 001 has state 1.
```

Device 002 now knows the state of Device 001 / Unit 2.

Note that "`invite`" and "`assign peer`" commands can *only* be issued from the Master. "set pin", "fast query pin", and "status pin" can be issued by either a Master or a Client.

## 5.4 Saving To Non-Volatile Memory

At this point nothing from the user commands has been saved, and if you reset both units (or power-cycle them) you will have to go through the entire process again.

In order to save the network configuration and peer assignments, you enter "save" on the command line. In order to save the configuration on the client, you connect your RS-232 terminal to the client board and enter "save" there also.

If you have performed the "save" on both units, you can reset or power-cycle either or both units and the switch-to-led activity should resume without any command-line stuff from the user. You don't have to have an RS-232 terminal connected to either unit to have this happen once you have issued the "save" command at both units.

## 5.5 More Commands

### 5.5.1 unassign peer

Just as there is an “assign peer” command, there is an “unassign peer” command. The format is the same. It removes a peer assignment from the peer list.

```
ocm-m> unassign peer:001:3:002:1
OK

Peer Table:
  No peers.
```

### 5.5.2 single text

Send text from one device to another using the “single text” command.

```
ocm-m> single text:2:3:002:"ab"
OK
ocm-m> Single transaction with 002; return status: SUCCESS
```

On Device 002:

```
ocm-c> Received text from 001:
ab
```

### 5.5.3 verbose level

Change the verbosity of listings and transactions. Much more detail can be seen with higher verbosity levels. All printouts so far have been with verbosity level 1. See what happens when we change it to level 6.

```
ocm-m> single text:2:3:002:"ab"
OK
ocm-m> Single transaction with 002; return status: SUCCESS
```

#### 5.5.3.1 Verbosity Level 1

```
ocm-c> verbose level:1
OK
ocm-c> list
ONE-NET Evaluation Version 2.3.0 (Build 107)
```

```
Invite code: 2222-2222
# of Network MH Devices : 2
# of Network MH Repeaters : 1
Message key : (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-
0e-0f)
Old Message key : (00-00-00-00) - (00-01-02-03) - (04-05-06-07) - (08-
09-0a-0b)

NID: 0x000000001
DID: 0x002


Channel: US 5
Keep-Alive Interval:1800000 ms
Send To Master: True
Reject Bad Msg ID: True
Peer table:
  No peers.


User pins:
  0 input state: 0
  1 output state: 0
  2 input state: 0
  3 output state: 1
OK
ocm-c>
```

#### 5.5.3.2 Verbosity Level 6

```
ocm-c> verbose level:6
OK
ocm-c> list
ONE-NET Evaluation Version 2.3.0 (Build 107)
Invite code: 2222-2222
# of Network MH Devices : 2
# of Network MH Repeaters : 1
Message key : (00-01-02-03) - (04-05-06-07) - (08-09-0a-0b) - (0c-0d-
0e-0f)
Old Message key : (00-00-00-00) - (00-01-02-03) - (04-05-06-07) - (08-
09-0a-0b)

NID: 0x000000001
DID: 0x002


Device Features...
Feature Bytes : 6F C1 38 87
Max Hops : 7
Max Peers : 8
```

```
Smple Client : False
Multi-Hop : Capable
Multi-Hop Repeat : Capable
Block : Not Capable
Stream : Not Capable
Data Rate / Channel Chg. : Capable
Extended Single : Capable
Route : Capable
Device Sleeps : False

Data Rates...

Data rate 38,400 : Capable
Data rate 76,800 : Not Capable
Data rate 115,200 : Not Capable
Data rate 153,600 : Not Capable
Data rate 192,000 : Not Capable
Data rate 230,400 : Not Capable


Channel: US 5
Keep-Alive Interval:1800000 ms
Send To Master: True
Reject Bad Msg ID: True

Master Features...
Feature Bytes : 6F C1 38 87
Max Hops : 7
Max Peers : 8
Simple Client : False
Multi-Hop : Capable
Multi-Hop Repeat : Capable
Block : Not Capable
Stream : Not Capable
Data Rate / Channel Chg. : Capable
Extended Single : Capable
Route : Capable
Device Sleeps : False

Data Rates...

Data rate 38,400 : Capable
Data rate 76,800 : Not Capable
Data rate 115,200 : Not Capable
Data rate 153,600 : Not Capable
Data rate 192,000 : Not Capable
Data rate 230,400 : Not Capable


Enc. DID=0xB4BC, Raw DID=0x001, Hops=0, Max. Hops=7, Data Rate=0, Msg
ID=10, Cur. Time=593904 ms, Verify Time=0 ms
```

```
Peer table:
  No peers.


User pins:
  0 input state: 0
  1 output state: 0
  2 input state: 0
  3 output state: 1
OK
ocm-c>
```

### 5.5.4 user pin

Changes the user pin from an input to an output or vice-versa.

```
ocm-c> list
ONE-NET Evaluation Version 2.3.0 (Build 107)

NID: 0x000000001
DID: 0x002

...

User pins:
  0 input state: 0
  1 output state: 0
  2 input state: 0
  3 output state: 1
```

If the Master attempts to set pin 2 of unit 002, it will not work because pin 2 is an input pin (see red)

Note that the prompt tells you what device is being shown (“ocm-m” for Master, “ocm-c” for Client) Note also that the verbosity level has been raised for detailed printouts.

```
ocm-m> verbose level:6
OK
ocm-m> set pin:002:2:1
OK
ocm-m> ehanr : NACK : Nack Reason-->0x0E(No Resp) : Handle-->0x05(TIMEOUT MS) : Payload : 50 ms

ehanr : NACK : Nack Reason-->0x8A(Unit Is Input) : Handle-->0x00() : Payload : N/A

ests start:Hdr-->PID=0x100,Msg ID=0x0C,Msg Type=0x0(App)
Message Data:500F200001, retry=2,hops=0,dst=0020
ack_nack-->NACK : Nack Reason-->0x8A(Unit Is Input) : Handle-->0x00() : Payload : N/A

Single transaction with 002; return status: FAILED
ests end
```

Switch the pin to an output using the “user pin” command. Try again with the master.

```
ocm-c> user pin:2:output
OK
```

Try again on the Master. This time it works (see red).

```
ocm-m> set pin:002:2:1
OK
ocm-m> ehanr : NACK : Nack Reason-->0x0E(No Resp) : Handle-->0x05(TIMEOUT MS) : Payload : 50 ms

ehanr : ACK : Nack Reason-->0x00(No Err) : Handle-->0x0D(APP) : Payload :
App payload : 0x4002F00001 : Class-->0400 : Type-->0x00 : Src Unit-->0x2 : Dst Unit-->0xF : Data-->0x00001

ests start:Hdr-->PID=0x100,Msg ID=0x0D,Msg Type=0x0(App)
Message Data:500F200001, retry=1,hops=0,dst=0020
ack_nack-->ACK : Nack Reason-->0x00(No Err) : Handle-->0x0D(APP) : Payload :
App payload : 0x4002F00001 : Class-->0x400 : Type-->0x00 : Src Unit-->0x2 : Dst Unit-->0xF : Data-->0x00001

Single transaction with 002; return status: SUCCESS
ests end
Unit 2 of 002 has state 1.
Device 002 has checked in.


ocm-c> list
ONE-NET Evaluation Version 2.3.0 (Build 107)

NID: 0x000000001
DID: 0x002

...

User pins:
  0 input state: 0
  1 output state: 0
  2 output state: 1
  3 output state: 1
```

### 5.5.5 Other Commands

Please see the ONE-NET Command Line Specification for a more thorough list of commands.

## 5.6 Functional Test Notes – User I/O Pins

Hardware:

- Logic levels for user pins is 3.3 Volts. User pins are connected to J3 through 50 Ohm resistors on the Evaluation Modules.
- Upon initialization, User pins 1 and 3 of an Evaluation Module configured as a `Master` are assigned to be inputs, and user pins 0 and 2 are outputs.
- Upon initialization, user pins 1 and 3 of an Evaluation Module configured as a `Client` are assigned to be outputs, and user pins 0 and 2 are inputs.
- These assignments can be changed by the "`user pin`" command from the Command Line interface. Please see the ***ONE-NET Command-Line Specification*** document.

Outputs:

- For demonstration purposes, these outputs can deliver a couple of milliamps to an LED.
- Depending on the LED, series resistors in the range of 150 Ohms and 390 Ohms may be good choices.
- To control heavier loads, a simple transistor or other driver can be used.
- The ADI-Renesas ONE-NET Evaluation Module operate at 3.3 Volts internally, and for loads less than a couple of milliamps, output values swing between approximately zero Volts and 3.3 Volts.

Inputs:

- When a pin is configured as an input, there is an internal pull-up resistor, so that an open input is treated as a logic one.
- If you want to make the input see a logic zero, you can connect the input to ground. To reduce battery current, you can connect a 1K Ohm resistor to ground for a logic zero.

## 5.7 Functional Test Notes – Software

Software notes:

- The command line interface is very specific to format such as spacing and capitalization.
- The colons must appear exactly in the places that are shown in the examples above. Extra spaces or other superfluous inputs will cause commands to fail.
- There is no command-line history (so you can't repeat the previous command with an up-arrow or any other command). Ctrl-C (or just about any other non-printing character) will abort any command in progress.

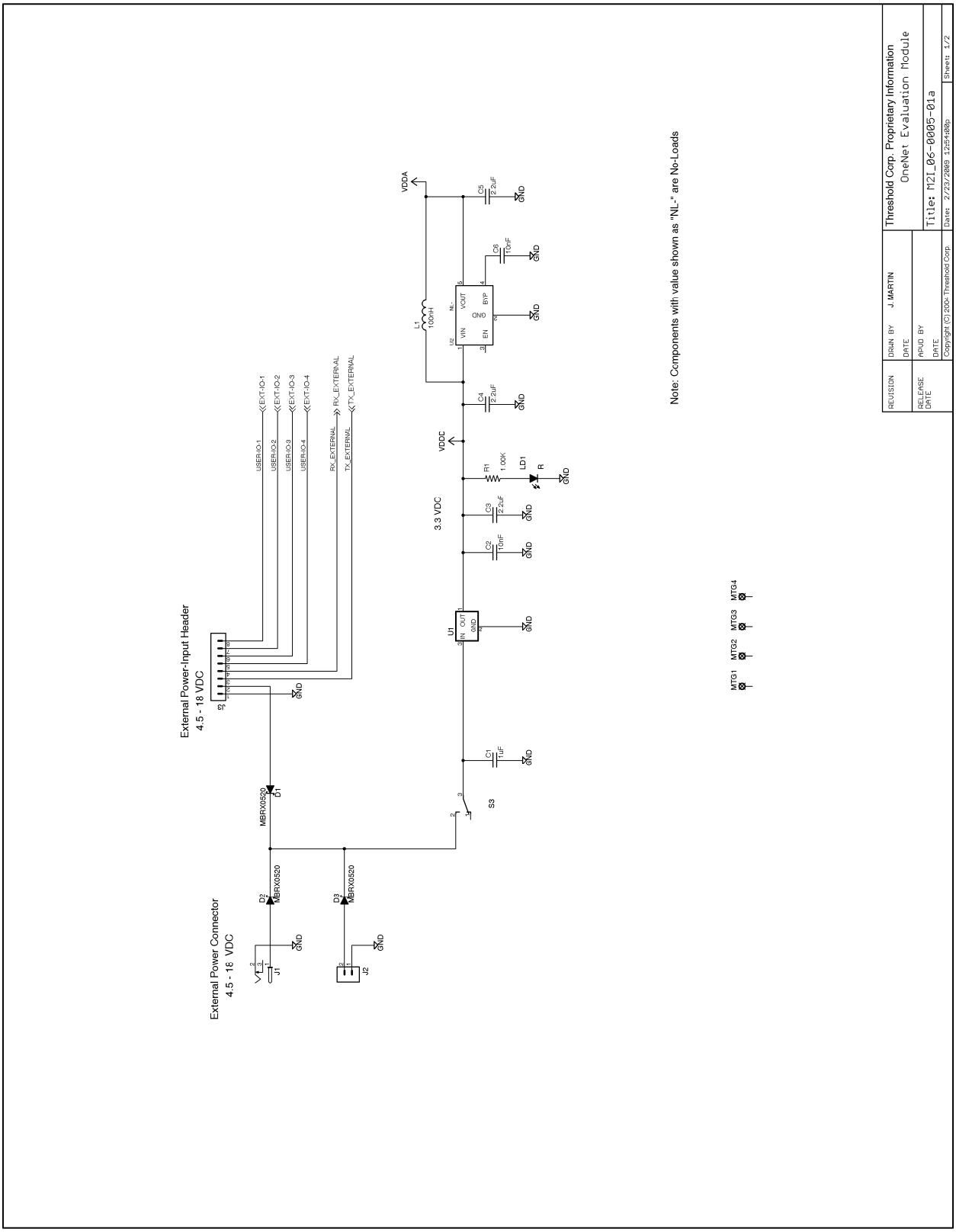
# 1 Appendix 1: Bill of Materials

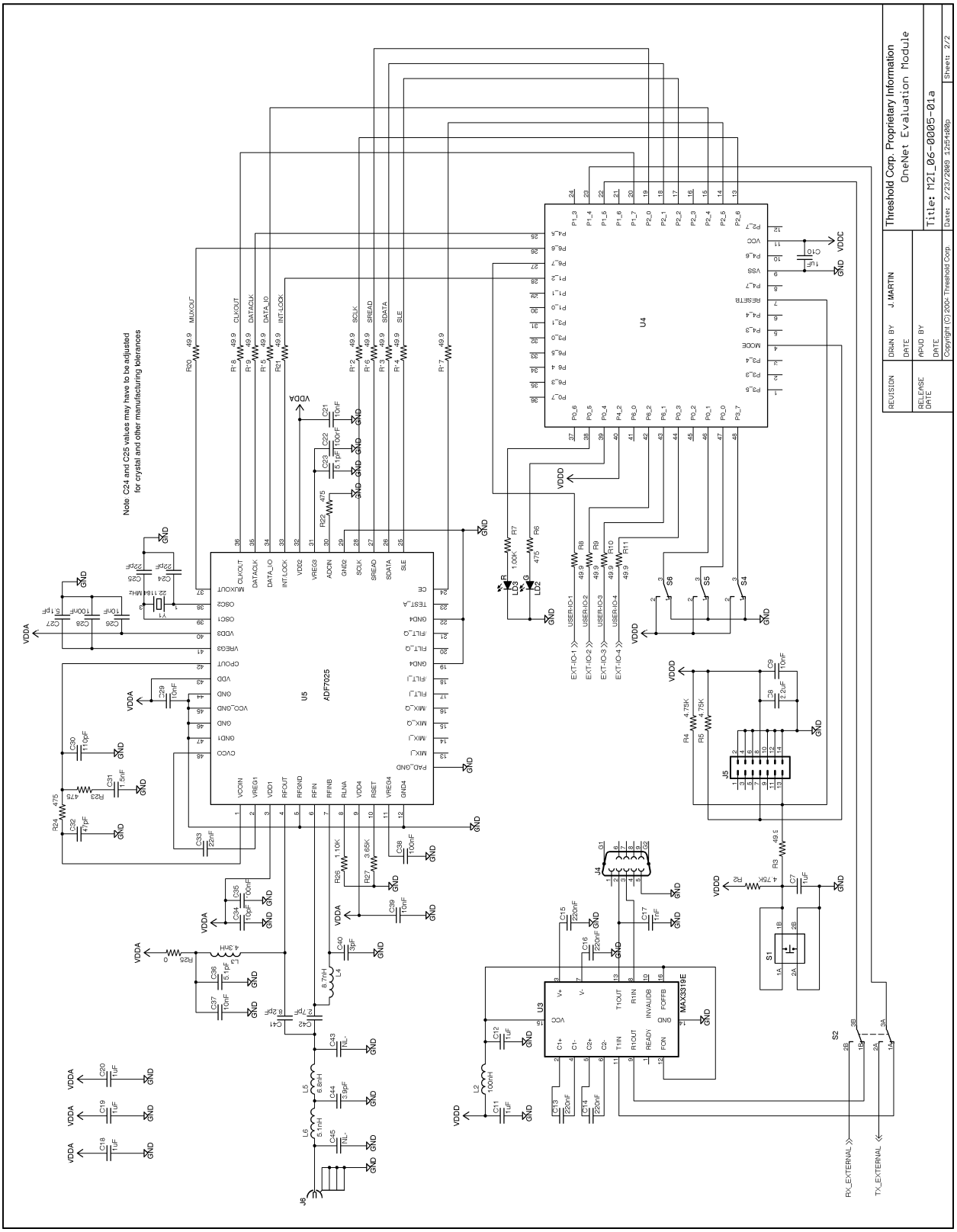
| ONE-NET ADI-REN Eval Board (M2I 05-0005-01) parts list | | | | | |
|---|---|---|---|---|---|
| qty | parts | description | package | mfr | part number |
| 1 | C1 | Capacitor, 1uF, ceramic, X5R, 20%, 16V | SMT-0603 | NIC | NMC0603X5R105M16TRPF |
| 4 | C13, C14, C15, C16 | Capacitor, 220nF, ceramic, X7R, 10%, 16V | SMT-0603 | NIC | NMC0603X7R224K16TRPF |
| 1 | C17 | Capacitor, 1nF, ceramic, X7R, 10%, 25V | SMT-0603 | NIC | NMC0603X7R102J25TRPF |
| 7 | C2, C9, C21, C26, C29, C37, C39 | Capacitor, 10nF, ceramic, X7R, 10%, 25V | SMT-0603 | NIC | NMC0603X7R103K25TRPF |
| 4 | C22, C28, C35, C38 | Capacitor, 100nF, ceramic, X7R, 10%, 16V | SMT-0603 | NIC | NMC0603X7R104K16TRPF |
| 3 | C23,C27, C36 | Capacitor, 5.1pF, ceramic, NPO, +/- .1pF, 50V | SMT-0603 | NIC | NMC-M0603NPO5R1B50TRPF |
| 2 | C24, C25 | Not Used | SMT-0603 | NIC | NMC0603NPO220F50TRP |
| 4 | C3, C4, C5, C8 | Capacitor, 2.2uF, ceramic, X5R, 10%, 6.3V | SMT-0603 | NIC | NMC0603X5R225K6.3TRPF |
| 1 | C30 | Capacitor, 110pF, ceramic, NPO,5%, 50V | SMT-0603 | NIC | NMC0603NPO111J50TRPF |
| 1 | C31 | Capacitor, 1.5nF, ceramic, NPO, 5%, 25V | SMT-0603 | NIC | NMC0603X7R152J50TRPF |
| 1 | C32 | Capacitor, 47pF, ceramic, NPO, 5%, 50V | SMT-0603 | NIC | NMC0603NPO470J50TRP |
| 1 | C33 | Capacitor, 22nF, ceramic, X7R, 5%, 25V | SMT-0603 | NIC | NMC0603X7R223J25TRPF |
| 1 | C34 | Capacitor, 10pF, ceramic, NPO, 5%, 50V | SMT-0603 | NIC | NMC0603NPO100B50TRPF |
| 1 | C40 | Capacitor, 3pF, ceramic, NPO, +/- .1pF, 50V | SMT-0603 | NIC | NMC-M0603NPO3R0B50TRPF |
| 1 | C41 | Capacitor, 8.2pF, ceramic, NPO, +/1 .25pF, 50V | SMT-0603 | NIC | NMC-M0603NPO8R2C50TRPF |
| 1 | C42 | Capacitor, 2.7pF, ceramic, NPO, +/- .1pF, 50V | SMT-0603 | NIC | NMC-M0603NPO2R7B50TRPF |
| 2 | C43, C45 | Not Used | SMT-0603 | TBD | N/A |

| | | | | | |
|---|---|---|---|---|---|
| 1 | C44 | Capacitor, 3.9pF, ceramic, NPO, +/- .1pF, 50V | SMT-0603 | NIC | NMC-M0603NPO3R9B50TRPF |
| 1 | C6 | Capacitor, 10nF, ceramic, X7R, 10%, 25V | SMT-0603 | NIC | NMC0603X7R103K25TRPF |
| 7 | C7, C10, C11, C12, C18, C19, C20 | Capacitor, 1uF, ceramic, X5R, 10%, 6.3V | SMT-0603 | NIC | NMC0603X5R105K6.3TRPF |
| 3 | D1, D2, D3 | Diode, Schottky, 20V | SOD-323 | M.C.C. | MBRX0520 |
| 1 | J1 | Connector, Power, , 1.3mm I.D., 4mm O.D., R/A | T-H | Kycon | KLDX-0201B |
| 1 | J2 | Connector, 1.5mm, 2pin, Vert., Male | T-H | JST | B2B-ZR(LF)(SN) |
| 1 | J3 | Connector, 1.5mm, 8pin, R/A | T-H | JST | S8B-ZR |
| 1 | J4 | Connector, DB-9, R/A | T-H | Kobiconn | 152-3409 |
| 1 | J5 | Header, 14-pin, Dual-Straight, Shrouded | T-H | FCI | 66506-038 |
| 1 | J6 | Connector, SMA | T-H | Molex | 73391-0070 |
| 2 | L1, L2 | Inductor, 100nH, Ceramic, 10%, 300mA, | SMT-0603 | TDK | MLG1608BR10J |
| 1 | L3 | Inductor, 4.3nH, wirewound, +/- .5nH | SMT-0603 | Murata | LQW18AN4N3D00D |
| 1 | L4 | Inductor, 8.7nH, wirewound, +/- .5nH | SMT-0603 | Murata | LQW18AN8N7D00D |
| 1 | L5 | Inductor, 6.8nH, wirewound, +/- .5nH | SMT-0603 | Murata | LQW18AN6N8D00D |
| 1 | L6 | Inductor, 5.1nH, wirewound, 5% | SMT-0603 | Pulse | PE-0603CD5N1JTT |
| 2 | LD1, LD3 | LED, Red | SMT-0805 | Lite-on | LTST-C170KRKT |
| 1 | LD2 | LED, Green | SMT-0805 | Lite-on | LTST-C170KGKT |
| 2 | PCB1 | Printed Circuit Board | N/A | TBD | 06-0005-01 |
| 3 | R1, R7 | Resistor, 1.00Kohm, 1% | SMT-0603 | NIC | NRC06F1001TRF |
| 3 | R2, R4, R5 | Resistor, 4.75Kohm, 1% | SMT-0603 | NIC | NRC06F4751TRF |
| 1 | R22, R23, R24 | Resistor, 475ohm, 1% | SMT-0603 | NIC | NRC06F4750TRF |
| 1 | R25 | Resistor, Zero-Ohm | SMT-0603 | NIC | NRC06Z |
| 1 | R26 | Resistor, 1.10Kohm, 1% | SMT-0603 | NIC | NRC06F1101TRF |
| 15 | R27 | Resistor, 3.65Kohm, 1% | SMT-0603 | NIC | NRC06F3651TRF |
| 1 | R3, R8-R21 | Resistor, 49.9ohm, 1% | SMT-0603 | NIC | NRC06F51R1TRF |
| 1 | S1 | Switch, Pushbutton, Mom-SPST-NO | TH | Omron | B3W-1000BYOMZ |
| 4 | S2 | Switch, DPDT, Slide | T-H | Alcoswitch | SSA22G |
| 1 | S3, S4, S5, | Switch, SPDT, Slide | T-H | Alcoswitch | SSA12 |

| | | | | | |
|---|---|---|---|---|---|
| | S6 | | | | |
| 1 | U1 | IC, Regulator, 3.3V | SOT-89 | Torex | XC6202P332PR |
| 1 | U2 | Not Used | SOT-23-5 | N/A | N/A |
| 1 | U3 | IC, RS-232 Transceiver | SSOP-16 | Sipex | SP3220EBCT |
| 1 | U4 | IC, Microcontroller | LQFP-48 | Renesas | R5F21238JFP#U0 |
| 1 | U5 | IC, RF Transceiver | QFN-48 | ADI | ADF7025BCPZ |
| 1 | Y1 | Crystal, 22.1184MHz., 18pF, 10ppm, -20:+70C | SMD-3.2x2.5 | My Frequency | M8332-FX101018-22.1184 |

# 2 Appendix II: Board Schematics


External Power-Input Header
4.5 - 18 VDC
External Power Connector
4.5 - 18 VDC
Note: Components with value shown as "NL-" are No-Loads
Threshold Corp. Proprietary Information
OneNet Evaluation Module
Title: M2I_06-0005-01a
Sheet: 1/2
MTG1
MTG2
MTG3
MTG4
3.3 VDC
VDDA
VDDE
EXT-IO-1
EXT-IO-2
EXT-IO-3
EXT-IO-4
RX_EXTERNAL
TX_EXTERNAL
MBRX0520
GND


Threshold Corp. Proprietary Information
OneNet Evaluation Module
Title: M2I_06-0005-01a
Sheet: 2/2
Note: C24 and C25 values may have to be adjusted for crystal and other manufacturing tolerances
U4
U5
ADF7025
U3
MAX3319E
J4
J5
J6
S1
S2
S4
S5
S6
LD1
LD2
LD3
RX_EXTERNAL
TX_EXTERNAL
VDDA
VDDD
GND