

Micro-E-02

Revision 1.10



Micro Connect Pty Ltd

www.microconnect.com.au

Overview

The Micro-E-02 is an advanced Ethernet access device with an RS-232 or RS-422 serial port, temperature sensors and digital I/O.

All programming and settings are achieved via the Ethernet port with the aid of the Micro-E configuration utility software. Updated versions of the GUI can be obtained for free at www.microconnect.com.au

This manual requires that both the Micro-E firmware & the GUI configuration software be V1.09 or greater.

The configuration software uses UDP ports 50,000 and 50,001 to access the unit.

Setting the IP Address

Direct ethernet connection

The GUI software has the ability to configure the Micro-E even if the IP address in the Micro-E is different from the connected network or the IP address is not known. Multiple devices with the same IP address can also be configured. The default address is 11.0.0.3

To use the direct ethernet connection the Micro-E must be connected via a hub/switch etc on the same subnet. Connection through a gateway/router will not work for the direct connection method.

The correct local ethernet IP address must be entered when prompted. A temporary IP address will be assigned a broadcast MAC address to detect the Micro-E device(s) to be configured.

When an item is found and selected the temporary IP address will be modified for the MAC address of the selected unit.

You can then configure the Micro-E for different settings.

The ARP entry will be removed on program exit.

NOTE: The temporary IP address assigned to the Micro-E assumes that the subnet is a minimum of "255.255.255.0" a subnet of "255.255.255.224" for example can fail to detect the Micro-E.

WARNING: During direct ethernet connection some network connected devices may not function (ie. if the temporary address is used for another device). When the program exits normal network functions should return.

Only 1 instance of the config software should be run when using the direct ethernet connection mode.

IP connection

If the Micro-E is already configured on the network, enter the IP address or DNS name in the IP address field or a global search can be performed using a broadcast packet (subnet search).

Once found, select the Micro-E to be configured and press continue.

A broadcast search may not work properly if your network blocks broadcast UDP packets.

The Direct Ethernet Connection method should be used if a Micro-E is found on the local network with a different subnet address configured in the unit.

Interface Ports

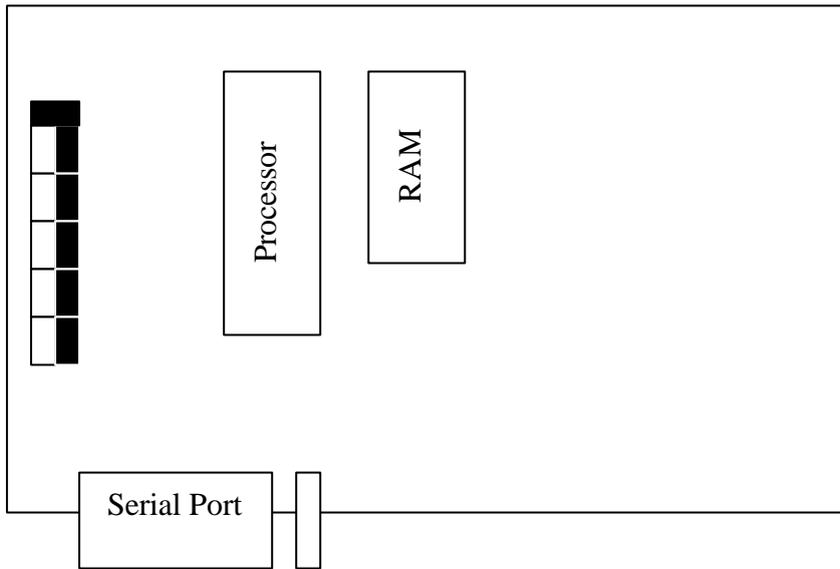


View of RS-232/RS-422 Serial and 2 x I2C ports

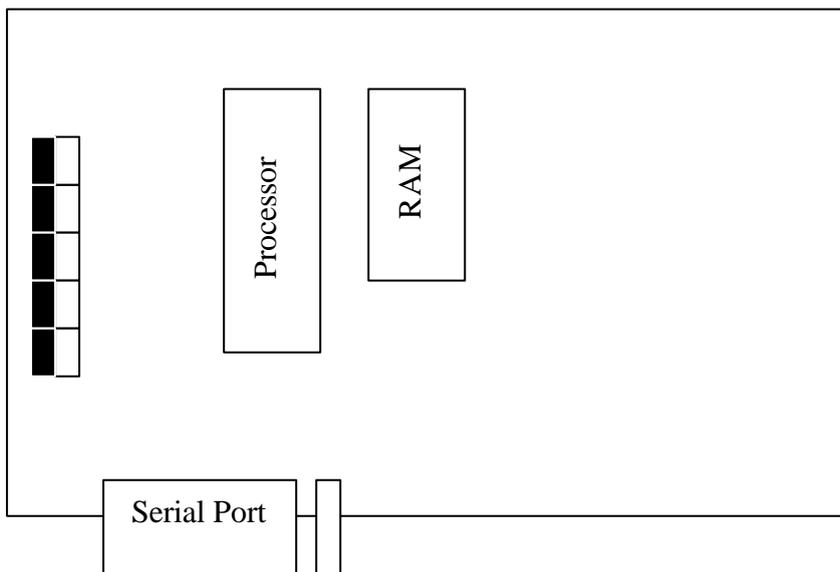
Serial Port

Item	Unit
Interface	RS-232 or RS-422 (jumper selectable)
Speed	1200 – 115200 baud
Parity	Even, Odd, None
Data Bits	7 or 8
Stop Bits	1
Mode	DTE, 3-wire interface

RS-422 Jumper Settings



RS-232 Jumper Settings



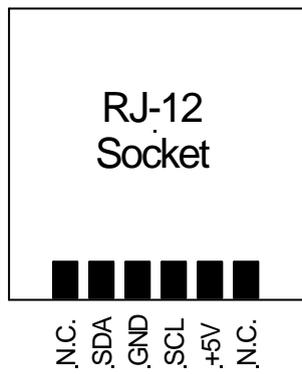
To change the jumpers the Micro-E must have the lid removed via the 2 screws on the base and 2 on the side.

The Serial port is configured as a DTE in RS-232 mode. Handshaking signals are not required but the DTR and RTS are active.

The Serial port config software differs from version 1.09 compared to version 1.10.

Firmware versions > 1.10 should only be used the configuration software of >1.10.

I2C (Temperature Port)



Viewed from the front looking at the socket.

The I2C port allows the optional MC-T1 temperature sensors to be attached to the Micro-E.

Temperature Sensor

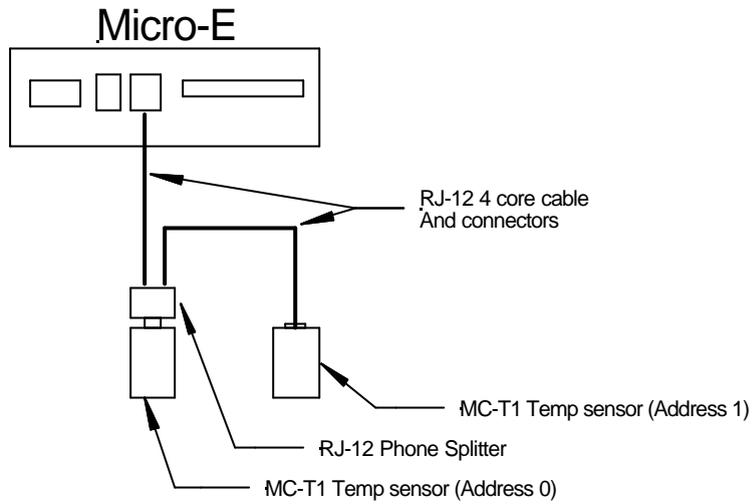
Optional MC-T1 temperature sensors are available for the Micro-E.

The sensors have the following features:

1. Up to 4 sensors on the 1 port
2. Addressable
3. Port powered
4. Uses standard RJ-12 phone cabling
5. Daisy chain connection
6. Up to 10M from Micro-E to sensors

The MC-T1 has +/- 1.5 Deg accuracy.

Temperature sensor connection diagram



Standard RJ-12 phone cabling can be used but it must be 4 core straight through cabled ie pin 1 to pin 1 etc. Some pre made cables are crossed and will not work with the sensor.

Accessing the Sensors via TCP/IP

The temperature sensor data is accessed via the TCP/IP utility port (also used for the digital I/O). This is setup in the Utility configuration page. The sensors can also be accessed via UDP port 50,001.

The protocol is ASCII based and is as follows.

1. Checking the temperature reading

Tx

T or t = Temperature request
x = The sensor address to read (0-3)

The Micro-E will reply with:

yyyy<CR><LF>

y = 4 characters containing the temperature data.

If the temperature is negative a “-“ sign will replace the first “0”. A carriage return and a line feed are also appended to the reading. Leading zero’s are not blanked.

If the sensor reply is “-001” a sensor may not be connected to the unit.

2. Unknown command

If the command is not able to be processed ie syntax error the Micro-E will reply with a “NO” + <CR><LF>.

Accessing the sensors via UDP

The sensor data can also be accessed on the UDP configuration port of 50,001.

The UDP access is intended for simple external program access without the TCP overhead and complexity.

Data is HEX ONLY.

If the format is incorrect there will not be any response.

3. Checking the temperature reading

Tx

T or t = Temperature request

x = The sensor address to read (00h-03h)

The Micro-E will reply with:

tyy – where y is the 2 byte sensor reading.

This reading is the direct reply from the LM77 sensors (National Semiconductor) and will be required to be decoded as described in the LM77 datasheet.

Digital I/O



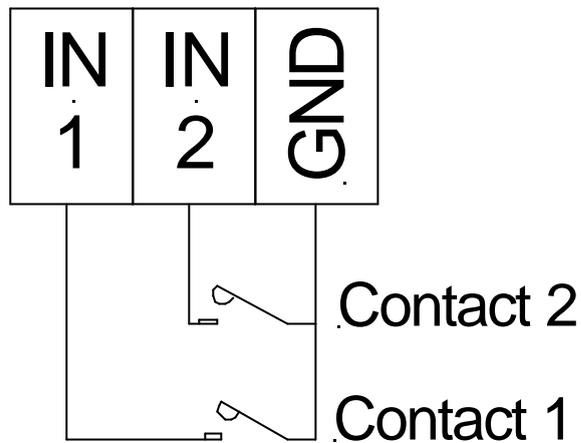
View of Digital I/O (Phoenix connector removed)

Inputs

The 8 inputs of the Micro-E-02 have the following specifications:

1. Maximum 30VDC input.
2. Internal 5VDC pull-up resistor.
3. Current limited input.

For most applications the inputs need only be connected through a set of contacts to the GND connection. An example of the wiring is shown below.



Example of inputs using the internal pull-up

Outputs

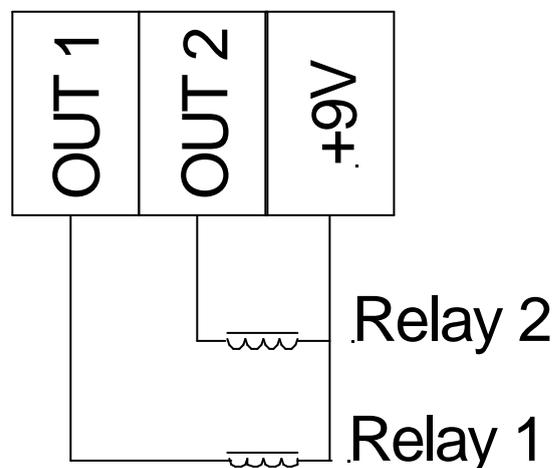
The 8 outputs of the Micro-E-02 have the following specifications:

1. Maximum 30VDC.
2. Open collector drive.
3. Max 150mA current per output (Current limited).
4. Relay clamping diode (relay direct drive).

The outputs do not have a pull up resistor. They have only 2 states:

1. OFF - High resistance
2. ON – Connected to GND

A typical connection diagram is shown below for powering a relay from the outputs.



Accessing the Digital I/O via TCP/IP

The TCP/IP protocol uses standard ASCII to allow a standard telnet session to implement changes. After a socket connection is established to the port number of the GUI configuration software the following commands are available:

1. Changing an OUTPUT:

Ox=y<CR>

Where:

O = Output

x = The output address to change (output 0 to 8).

if x = 0 indicates to address all of the outputs

y = 0 or 1

0 means turn output to high impedance state

1 means output connected to GND.

Default power up state of each output is controlled via the GUI setup.

The Micro-E will respond with either:

OK<CR> (processed successfully)

NO<CR> (command unrecognized or incorrect)

2. Checking the OUTPUT STATE:

?Ox=<CR>

Where:

? = Asking the state of the output

O = Output

x = The output address to check (output 0 to 8).

x=0 can be used to get a binary value of the outputs, Bit 0 is output 0, Bit 8 is output 8.

The Micro-E will respond with either:

OSx=y<CR> (command was successful)

Where:

OS = Output State (*case sensitive*)

x = The output address checked (output 0 to 8).

*x=0 will get a binary value of the outputs as 1 byte, Bit 0 is output 0, Bit 8 is output 8.
This will not be displayed on a telnet session correctly.*

y = 0 or 1

0 means the output is in the high impedance state

1 means output is connected to GND.

or if x=0 the data is 1 byte (decode in binary)

NO<CR> (command unrecognized or incorrect)

3. Checking the INPUT STATE:

?Ix=<CR>

Where:

? = Asking the state of the output

I = Input

x = The input address to check (output 0 to 8).

x=0 can be used to get a binary value of the inputs, Bit 0 is output 0, Bit 8 is output 8.

The Micro-E will respond with either:

ISx=y<CR> (command was successful)

Where:

IS = Input State (*case sensitive*)

x = The input address to check (output 0 to 8).

x=0 can be used to get a binary value of the inputs, Bit 0 is output 0, Bit 8 is output 8.

This will not be displayed on a telnet session correctly.

y = 0 or 1

0 means the input is in the low (ie GND).

1 means the input is high or open circuit

or if x=0 the data is 1 byte (decode in binary)

NO<CR> (command unrecognized or incorrect)

Accessing the digital I/O via UDP

The UDP digital I/O is available on port 50,001.

The UDP access is intended for simple external programming access without the TCP overhead and complexity.

The UDP access only sends or receives 2 bytes.

The first byte is the control or command byte.

The second byte is the data.

All of the data is a single binary encoded byte with:

Bit 0 as the output/input 0.

Bit 8 as the output/input 8 etc.

If the format is incorrect there will not be any response.

1. Changing an OUTPUT:

Ox

O = Outputs

x = The output data in binary encoded values.

The Micro-E will reply with:

ox

o = Outputs

x = The outputs in data binary encoded (echoed for confirmation).

Checking the OUTPUT STATE:

Ix

I = Inputs

x = The input data.

The Micro-E will reply with:

ix

i = Inputs

x = The input data in binary encoded values.

2. Checking the INPUT STATE:

Sx

s = Output States

x = The output data in bit values.

The Micro-E will reply with:

sx

o = Outputs

x = The output data in binary encoded values.

Index

Overview	2
Setting the IP Address.....	3
Direct ethernet connection.....	3
IP connection.....	4
Interface Ports	5
Serial Port.....	5
RS-422 Jumper Settings.....	6
RS-232 Jumper Settings.....	6
I2C (Temperature Port).....	8
Temperature Sensor	9
Temperature sensor connection diagram	10
Accessing the Sensors via TCP/IP	11
1. Checking the temperature reading	11
2. Unknown command	11
Accessing the sensors via UDP	12
3. Checking the temperature reading	12
Digital I/O	13
Inputs.....	13
Outputs.....	14
Accessing the Digital I/O via TCP/IP	15
1. Changing an OUTPUT:	15
2. Checking the OUTPUT STATE:.....	16
3. Checking the INPUT STATE:.....	17
Accessing the digital I/O via UDP	18
1. Changing an OUTPUT:	18
2. Checking the INPUT STATE:.....	19