



Trio Motion Technology Ltd.  
Shannon Way, Tewkesbury,  
Gloucestershire. GL20 8ND  
United Kingdom  
Tel: +44 (0)1684 292333  
Fax: +44 (0)1684 297929

1000 Gamma Drive  
Suite 206  
Pittsburgh, PA 15238  
United States of America  
Ph: +1 412.968.9744  
Fx: +1 412.968.9746

**Doc No.:** TN20-74  
**Version:** 1.1  
**Date:** 30<sup>th</sup> Nov 2004  
**Subject:** Ethernet IP Daughter Board

---

## Application Note

### 1. Introduction:

The new P298 Ethernet IP Daughter Board requires some specific setup sequences, written in Trio BASIC, to be run when used with the first release of system software. (v1.63 Dev 39 or later)

### 2. Applicability:

Ethernet IP Daughter Board P298.  
MC206/MC224 System Software V1.63 Dev 39 or higher.

### 3. Interface to HMS Anybus module:

The P298 is designed around the Ethernet IP Anybus module from HMS. Access to the memory and functions of the module are via a 16 bit read/write interface with an address range from 0 to 65535. (\$0000 to \$FFFF) A new command has been added to the Trio BASIC language that allows a programmer to read and write the Anybus module's memory.

Read:

```
ETHERNET_IP(slot_number, function, index, #params, vr_base_index)
```

Write:

```
ETHERNET_IP(slot_number, function, index, param1, [param2, param3, param4])
```

slot\_number : the physical daughter board slot where the P298 is located (0 for MC206).

function : 0 = read, 1 = write.

index : memory address to be accessed.

#params : number of parameters to be read consecutively from the address.

1 = only the parameter at the address is read.

2 = address and address+1 are read.

3 = address > address+2 are read.

4 = address > address+3 are read.

vr\_base\_index : the starting index of a block of up to 4 VRs to read data into.

param1 ..param4 : values to be written to address, address+1, address+2, address+3.

The Ethernet\_IP function returns either TRUE (-1) or FALSE (0) to indicate the success or otherwise of the function call.

## 4. Starting the Ethernet IP board:

The following sequence checks that the communication to the P298 is good and then sets the mode bit to start the Ethernet IP board operation.

```
' Get status to establish communications
FOR i=1 TO 10
  cf=ETHERNET_IP(slot_no,rd,$5002,1,vbase)
  IF cf=TRUE THEN
    GOTO setmode
  ENDIF
NEXT i
PRINT #5,"Error - can't communicate with EthernetIP db"

setmode:
ETHERNET_IP(slot_no,wr,$5001,1) ' set mode word to 1
```

(See (1) Pages 3-1, 7-1 and 7-2)

## 5. Setting IP address and subnet mask:

Before the P298 can be used on an Ethernet system, the IP address, subnet mask, default gateway etc. must be set. This routine writes the values to the registers in the Anybus module. Note that the values are stored in FLASH EPROM and thus are retained after power is removed. The power must be cycled off/on before these address values are initialised and the ethernet connection can be used.

```
setup_ethernet:
' IP address
v1=192 : v2=168 : v3=0 : v4=252
p1=v1*256+v2
p2=v3*256+v4
PRINT #5,HEX(p1),HEX(p2)
ETHERNET_IP(slot_no,wr,$7004,p1,p2)
' subnet mask
v1=255 : v2=255 : v3=255 : v4=0
p1=v1*256+v2
p2=v3*256+v4
PRINT #5,HEX(p1),HEX(p2)
ETHERNET_IP(slot_no,wr,$7009,p1,p2)
' Default Gateway
v1=192 : v2=168 : v3=0 : v4=225
p1=v1*256+v2
p2=v3*256+v4
```

```

PRINT #5,HEX(p1),HEX(p2)
'  ETHERNET_IP(slot_no,wr,$700d,p1,p2)
  ETHERNET_IP(slot_no,wr,$7019,0) ' DHCP Enable OFF
  ETHERNET_IP(slot_no,wr,$701e,0) ' Data Rate = Auto
RETURN

```

(See (2) pages 13-1 and 13-2)

## 6. Reading the IP address and subnet mask:

This routine reads the IP address and subnet mask that are currently in use. The values are printed in the Motion Perfect terminal in the usual "dot" format.

```

read_ethernet_settings:
  ETHERNET_IP(slot_no,rd,$7007,2,vbase)
  v1=(VR(vbase) AND $ff00)/256
  v2=VR(vbase) AND $ff
  v3=(VR(vbase+1) AND $ff00)/256
  v4=VR(vbase+1) AND $ff
  PRINT #5,"IP Address: ";v1[0];".";v2[0];".";v3[0];".";v4[0]
  PRINT #5,HEX(VR(vbase));HEX(VR(vbase+1))

  ETHERNET_IP(slot_no,rd,$700b,2,vbase)
  v1=(VR(vbase) AND $ff00)/256
  v2=VR(vbase) AND $ff
  v3=(VR(vbase+1) AND $ff00)/256
  v4=VR(vbase+1) AND $ff
  PRINT #5,"Subnet Mask: ";v1[0];".";v2[0];".";v3[0];".";v4[0]
  PRINT #5,HEX(VR(vbase));HEX(VR(vbase+1))

  ETHERNET_IP(slot_no,rd,$700f,2,vbase)
  v1=(VR(vbase) AND $ff00)/256
  v2=VR(vbase) AND $ff
  v3=(VR(vbase+1) AND $ff00)/256
  v4=VR(vbase+1) AND $ff
  PRINT #5,"Def Gateway: ";v1[0];".";v2[0];".";v3[0];".";v4[0]
  PRINT #5,HEX(VR(vbase));HEX(VR(vbase+1))

  ETHERNET_IP(slot_no,rd,$701a,1,vbase)
  PRINT #5,"DHCP Enable: ";VR(vbase)[0]

  ETHERNET_IP(slot_no,rd,$701f,1,vbase)
  PRINT #5,"Data Rate: ";VR(vbase)[0] ' 0=Auto, 1=10MHz, 2=100MHz
RETURN

```

(See (2) pages 13-1 and 13-2)

## 7. Reading/writing the fieldbus data

The Anybus module has a common data area that can be accessed by both the fieldbus and the Motion Coordinator. This data area can be mapped to Modbus and/or Ethernet IP by setting certain parameters in the module. As a simple example this routine maps the Modbus TCP to the data area so that a remote master can read and write data

using this protocol.

```
' set up the common data area
ETHERNET_IP(slot_no,wr,$6001,32) ' FB Out data area size (param 41)
ETHERNET_IP(slot_no,wr,$6018,32) ' SCI In data area size (param 64)

ETHERNET_IP(slot_no,wr,$601b,$0000) ' set offset in FB Out area (param 67)
ETHERNET_IP(slot_no,wr,$601c,32) ' set data size of FB Out area (param 68)
ETHERNET_IP(slot_no,wr,$6006,$0000) ' set offset in FB In area (param 46)
ETHERNET_IP(slot_no,wr,$6007,32) ' set data size of FB In area (param 47)

'ETHERNET_IP(slot_no,wr,$5001,1) ' set mode word to 1

REPEAT
' read and display the data from FB Out (data coming to us)
ETHERNET_IP(slot_no,rd,$1000,4,vbase)
PRINT #5,HEX(VR(vbase)),HEX(VR(vbase+1)),HEX(VR(vbase+2)),HEX(VR(vbase+3))
WA(500)
UNTIL FALSE
```

Notice that there are 2 separate 16-bit words (32 byte) areas, one for data coming into the Motion Coordinator and the other for data going out. The repeat loop above reads 4 words from this data area and prints them on the Motion Perfect terminal 5. The data is written to from Modbus TCP using the Modbus function 6, write single register or function 16, write multiple registers addressed at 40001 to 40016.

Data going out of the Motion Coordinator must be written using the command:

```
' Write my data out (var1..var4) to the PLC
ETHERNET_IP(slot_no,wr,$0000,var1,var2,var3,var4) `Sends 4 words to PLC
```

The above ETHERNET\_IP command is read within the daughter board over Modbus TCP using Modbus function 4, read input registers addressed at 30001 to 30016.

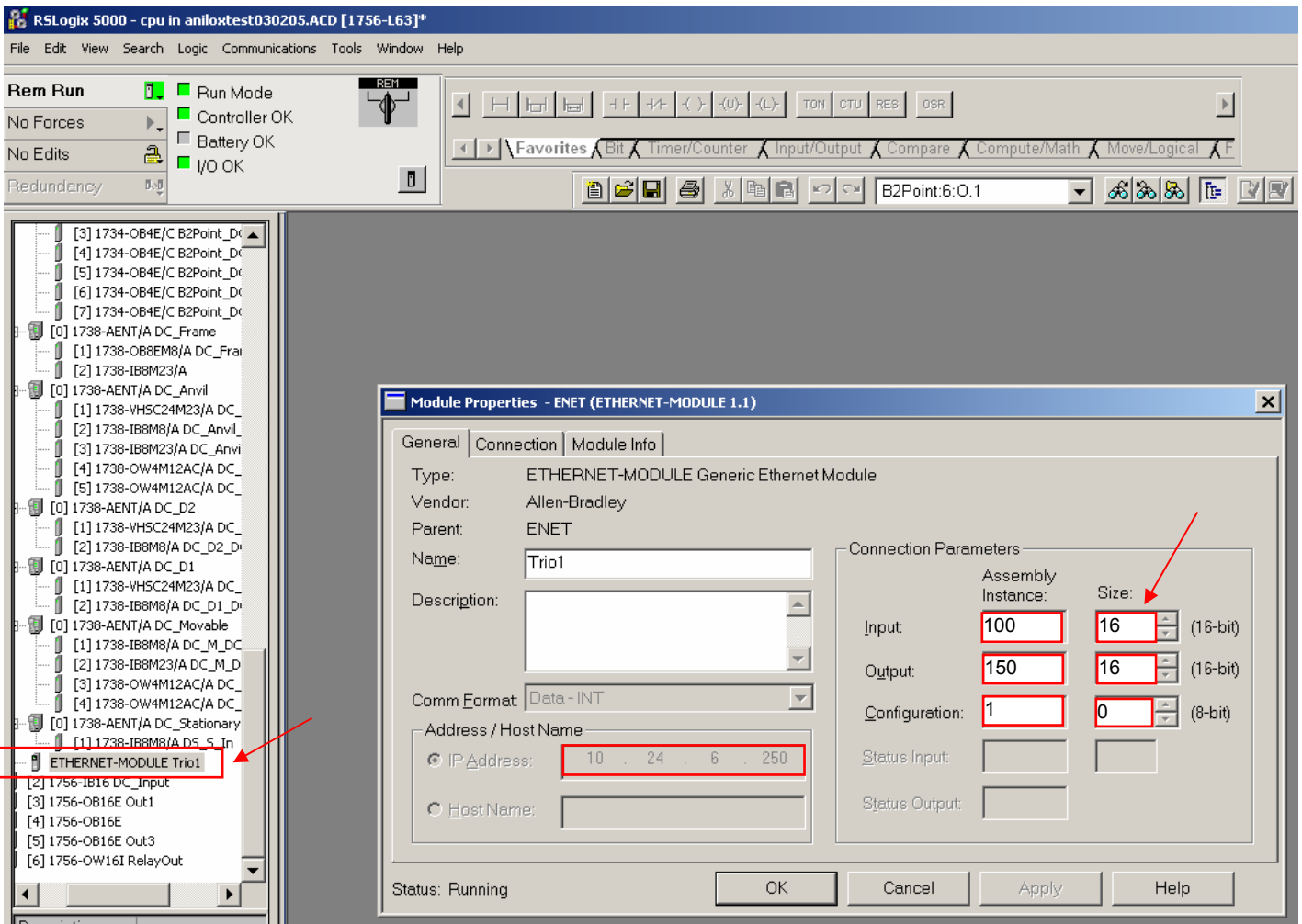
(See also (2) chapters 5, 6 and 11)

## 8. Ethernet-IP protocol configuration.

This section explains how to set up the PLC side (field bus) for use with the Trio P298 Ethernet-IP daughter board. The Trio P298 daughter board is permanently set as a server (slave) device on the network. The PLC (or other) must be a client (master) to communicate with the Trio. Specifically, the Allen Bradley RSLogix™ 5000 PLC is used here as an example. The Trio's IP address and subnet mask should have been set as defined in Section 5 of this note.

To allow the PLC to identify the Trio unit, the supplied EDS file must be loaded into the PLC. This is done using the RSLogix™ 5000 software. Once the Trio is found on the network it will show up in the RSLogix™ 5000 software device list (screen1). Several connection parameters need to be set for proper communications. The RSLogix™ 5000 software Connection Parameter window can be found in Module Properties for Trio (screen1). The six parameters below define the Trio properties.

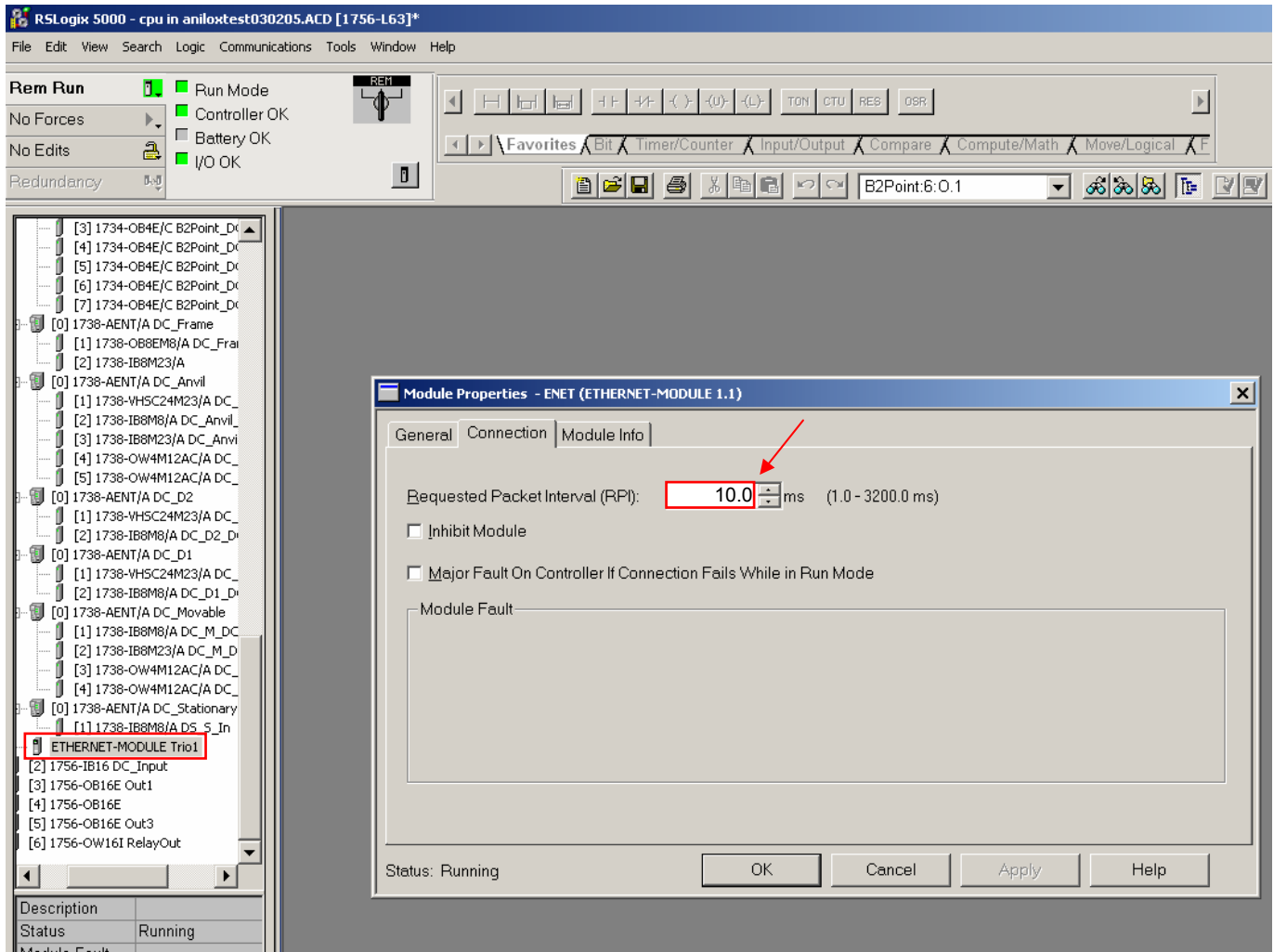
Assembly Instance [Input]:	100	Size [Input]:	16
Assembly Instance [Output]:	150	Size [Output]:	16
Assembly Instance [Configuration]:	1	Size [Configuration]:	0



Screen1 – RSLogix™ 5000 Program showing Trio identified and setup parameters

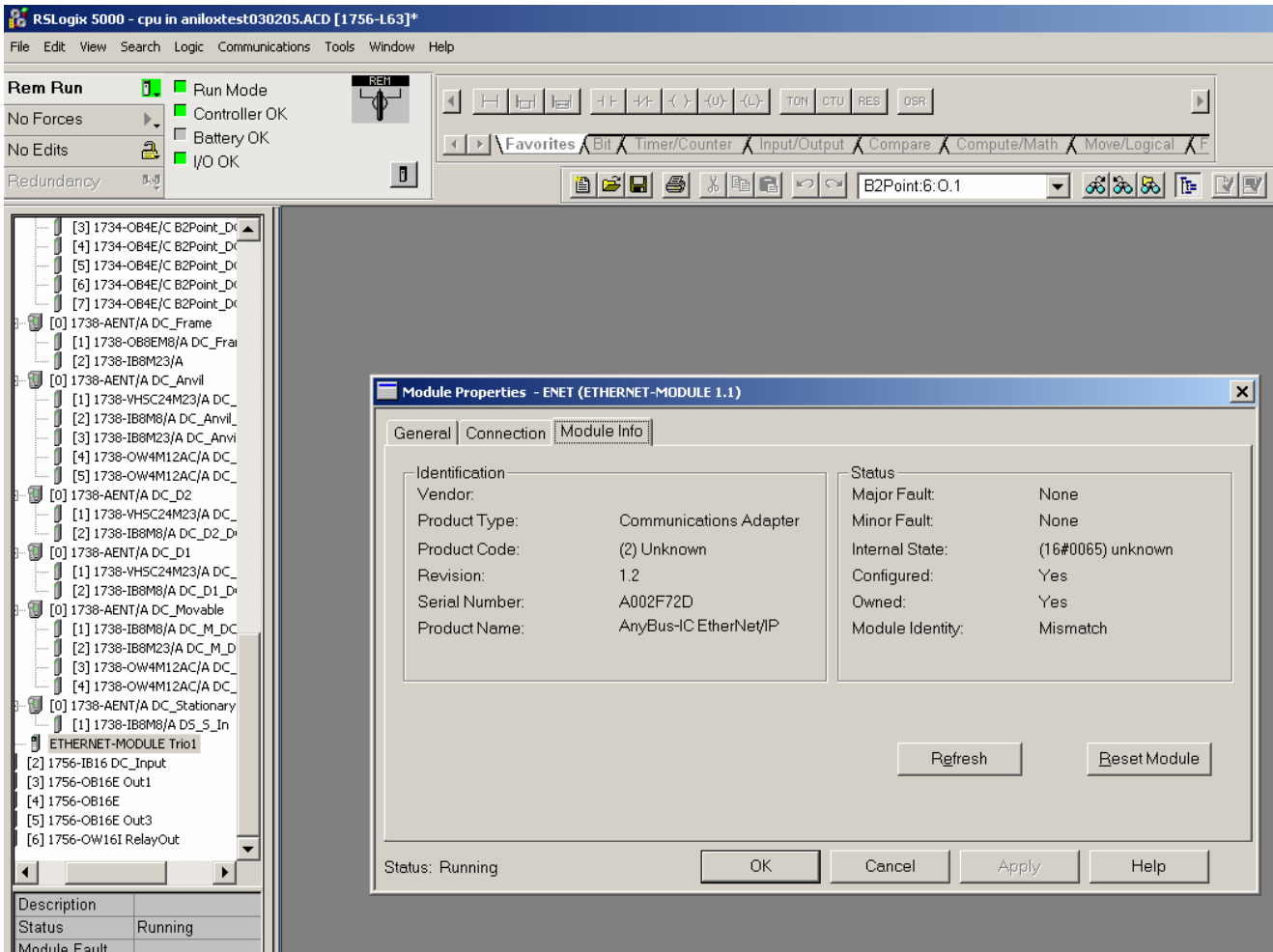
The Trio's IP address is also visible in the "General" tab of the Module Properties (screen1).

Moving to the "Connection" tab within Module Properties shows the Requested Packet Interval. This should be set to 10ms (screen2)



Screen2 – RSLogix™ 5000 Program showing Requested Packet Interval

The Trio's identification (displays as AnyBus-IC EtherNet/IP) can be seen at the "Module Info" tab within the Module Properties window (screen3).

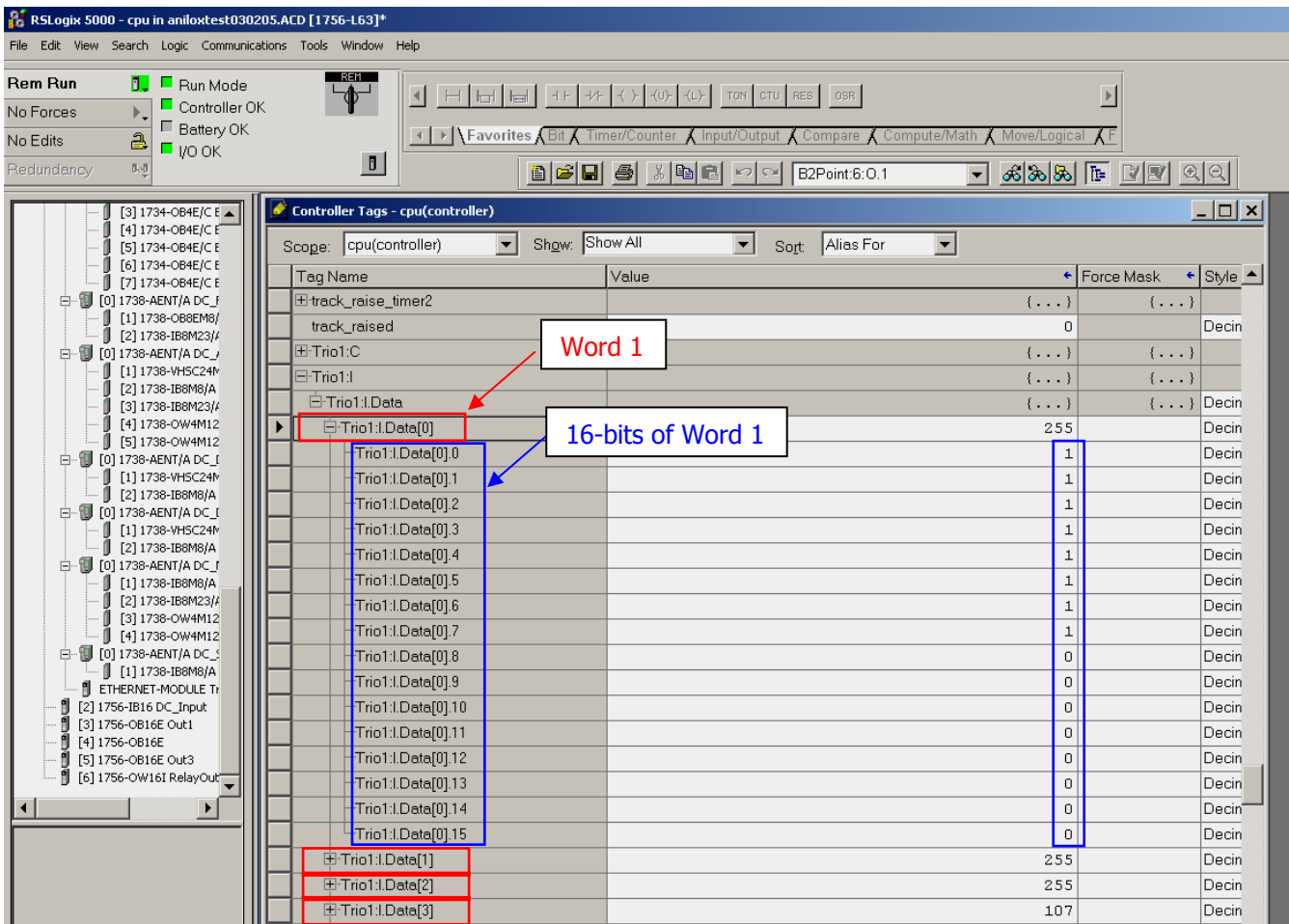


Screen3 – RSLogix™ 5000 Program showing Trio Module Info

After you confirm proper communications and setup of the Trio Ethernet-IP link, you can begin sending and receiving up to (4) 16-bit words. Data sent to the PLC from the Trio BASIC program is accomplished using

```
' Write my data out (var1..var4) to the PLC
ETHERNET_IP(slot_no,wr,$0000,var1,var2,var3,var4) `Sends 4 words to PLC
```

Note these are (4) words of 16-bit integer values. Some PLCs' flip the order of least significant byte (LSB) and most significant byte (MSB). This is hardware dependent based on the processor used in the PLC. In the case of the RSLogix™ 5000 PLC, the two 8-bit bytes per 16-bit word are flipped (screen4).



Screen4 – RSLogix™ 5000 Program showing Trio 16-bit word values



## 9. Bibliography

Please refer to the following documents for further information. Some suggested page references are given here, but these are not the only sections covering the subject(s) discussed.

- 1) Anybus IC - Design Guide, revision 1.30
- 2) Anybus IC EIP - Fieldbus Appendix, revision 1.12