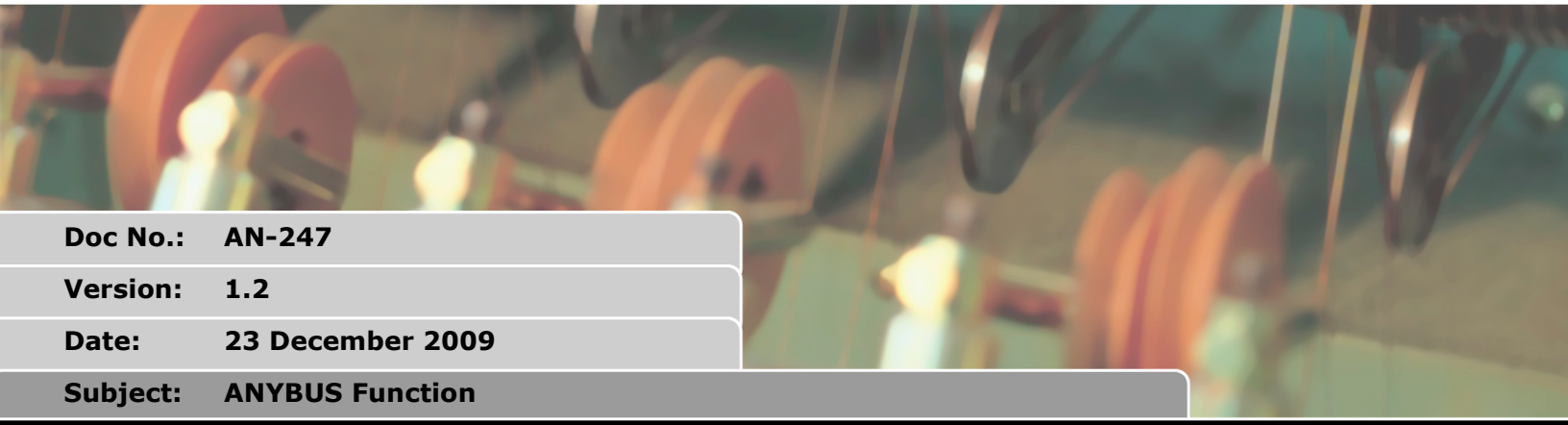
Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292333
Fax: +44 (0)1684 297929

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

Tomson Centre
118 Zhang Yang Rd., B1701
Pudong New Area, Shanghai,
Postal code: 200122
P. R. CHINA
Tel/Fax: +86-21-58797659



Doc No.: AN-247
Version: 1.2
Date: 23 December 2009
Subject: ANYBUS Function

APPLICATION NOTE

1. ANYBUS

1.1. Required Version: 2.0042

1.2. Type: System Function

1.3. Syntax: ANYBUS(function, slot [, parameters…])

1.4. Description:

This function allows the user to configure the communication module and set the network to an operation state. Some networks have limitations on data types and size, please refer the Anybus data sheet for details.

The ANYBUS function provides access to 3 different operations.

Operation:	0	Configure map
	1	Configure module and start protocol
	2	Stop protocol
	3	Read status byte
	4	Auto configure mapping

1.5. Parameters:

Assigns a VR or table point to the memory area that is updated over the network. Returns a true if the function was successful.

Individual or all maps can be deleted using the first 5 parameters.

The current mapping can be printed to the terminal using the first 2 parameters.

Function 0	ANYBUS(0,slot [, map, data source [, index, data type, count, read/write]])
Slot	Module slot in which the Anybus is fitted
Map	Map number, use -1 to delete all maps
Data source	location for data on the MC464

	-1	delete map
	0	VR
	1	Table
Index	Start position in data source	
Data type	the size and type of data that is sent across the bus	
	0	boolean
	1	signed 8 bit integer
	2	signed 16 bit integer
	3	signed 32 bit integer
	4	unsigned 8 bit integer
	5	unsigned 16 bit integer
	6	unsigned 32 bit integer
	7	character
	8	enumeration
	9-15	Reserved
	16	signed 64 bit integer
	17	unsigned 64 bit integer
	18	floating point/real number
Count	Number of data types mapped	
Read/ write	Data direction	

1.6. Parameters

Resets the Anybus module, loads the mapping and then sets the network to operational mode using the parameters provided. Returns a true if the function was successful.

Function 1	ANYBUS(1,slot, address [, baud])
Slot	Module slot in which the Anybus is fitted
Address	Module address, node number, MAC id. etc
Baud	Network baud rate
	Baud rate CC Link - required
	0 156 kbps
	1 625 kbps
	2 2.5 mbps
	3 5 mbps
	4 10 mbps
	Baud rate Devicenet – optional
	0 125kbps
	1 250kbps
	2 500kbps
	3 autobaud (default)

Baud rate Profibus – automatic, not required

1.7. Parameters:

Stops the cyclic data transfer. Returns a true if the function was successful.

Function 2	ANYBUS(2,slot)
Slot	Module slot in which the Anybus is fitted

1.8. Parameters:

Reads the status byte from the Anybus module. Returns the status byte.

Function 3	ANYBUS(3,slot)
Slot	Module slot in which the Anybus is fitted

Anybus status byte:

Bits 0-2	Anybus State	
	0	SETUP
	1	NW_INIT
	2	WAIT_PROCESS
	3	IDLE
	4	PROCESS_ACTIVE
	5	ERROR
	6	(reserved)
	7	EXCEPTION
Bit 3	Supervisory bit	
	0	Module is not supervised.
	1	Module is supervised by another network device
Bits 4-7	Reserved	

1.9. Parameters

Auto configure and start the cyclic network. The mapping can still be read using function 0. Currently only available for the Profibus network.

Function 4	ANYBUS(4,slot, address, type, inoff, outoff)	
Slot	Module slot in which the Anybus is fitted	
Address	Module address, node number, MAC id. Etc	
Type	Data type and location	
	0	VR Integer
	1	Table Integer
	2	VR Float
	3	Table Float
Inoff	Offset for inputs	
Outoff	Offset for outputs	

1.10. Example 1:

Configure Device Net with 2 16-bit integer inputs and 2 16-bit integer outputs. This data is transmitted cyclically using the 'Polled Connection' method. Ensure to configure the master identically to the slave otherwise the data will not transmit.

```
device_net:

slotnum=0 'Local variable with module slot number

'Map data
  map=FALSE
'Map received data
  map= ANYBUS(0, slotnum, 1, 0, 0, 2, 4, 0) '4*16-bit Int Rx

  IF map=TRUE THEN
'Map transmit data
    map= ANYBUS(0, slotnum, 2, 0, 4, 2, 4, 1) '4*16-bit Int Tx
  ENDIF

  IF map=FALSE THEN
    PRINT#term, "Mapping failed"
    STOP
  ENDIF

'Print mapped data to the terminal
```

```
  ANYBUS(0,slotnum)

'Start Network
  map= ANYBUS(1, slotnum, 3, 2)  'MAC ID=3, Baud=500k
  IF map=FALSE THEN
    PRINT#term, "Failed to start network"
    STOP
    ELSE
    PRINT#term, "Network Started"
  ENDIF
  RETURN
```

1.11. Example 2:

Configure CC-Link with 2 stations, both with 16 bits in, 16 bits out, 2 SINT16 in and 2 SINT16 out. Ensure that the master is configured identically and that the handshaking bits are implemented.

```
cc_link:
'Function 0 - Set up mapping
'station 1
  map = ANYBUS(0, slotnum, 0, 0, 0, 0, 16, 0) '16*BOOL Rx
  map = ANYBUS(0, slotnum, 1, 0, 1, 0, 16, 1) '16*BOOL Tx
  map = ANYBUS(0, slotnum, 2, 0, 2, 2, 2, 0)'2*16-bit Int Rx
  map = ANYBUS(0, slotnum, 3, 0, 4, 2, 2, 1) '2*16-bit Int Tx
'station 2
  map = ANYBUS(0, slotnum, 4, 0, 6, 0, 16, 0) '16*BOOL Rx
  map = ANYBUS(0, slotnum, 5, 0, 7, 0, 16, 1) '16*BOOL Tx
  map = ANYBUS(0, slotnum, 6, 0, 8, 2, 2, 0) '2*16-bit Int Rx
  map = ANYBUS(0, slotnum, 7, 0, 10, 2, 2, 1) '2*16-bit Int Tx

  ANYBUS(0,slotnum) 'print mapping to terminal

'Function 1 - Start Protocol
  IF map = FALSE THEN
  map = ANYBUS(1, slotnum, 1, 2)
  ENDIF
```

1.12. Example 3

Configure Profibus by manually mapping the data.

1.13. Example 4

Configure Profibus using the automated mapping.

```
Profibus:
 vrint=0
 tableint=1
 vrfloat=2
 tablefloat=3
 slotnum=0

  'Function 4, read network mapping, configure and start.
  map= ANYBUS(4, slotnum, 5, vrint, 100, 200)

  IF map=FALSE THEN
    PRINT#term, "Failed to start network"
    STOP
```

```
ENDIF
ANYBUS(0,slotnum) 'print mapping to terminal
```

2. CC-Link

The CC-Link module requires some handshaking to enter the process active state. This section details how to configure the handshaking in the master to enable communications. It is assumed that the user is experienced with the master and that they can configure the CC-Link as per the manufacturers' information.

2.1. Master configuration

The master must be configured as per the manufacturers' instructions. In this example the master is configured as shown in Figure 1.

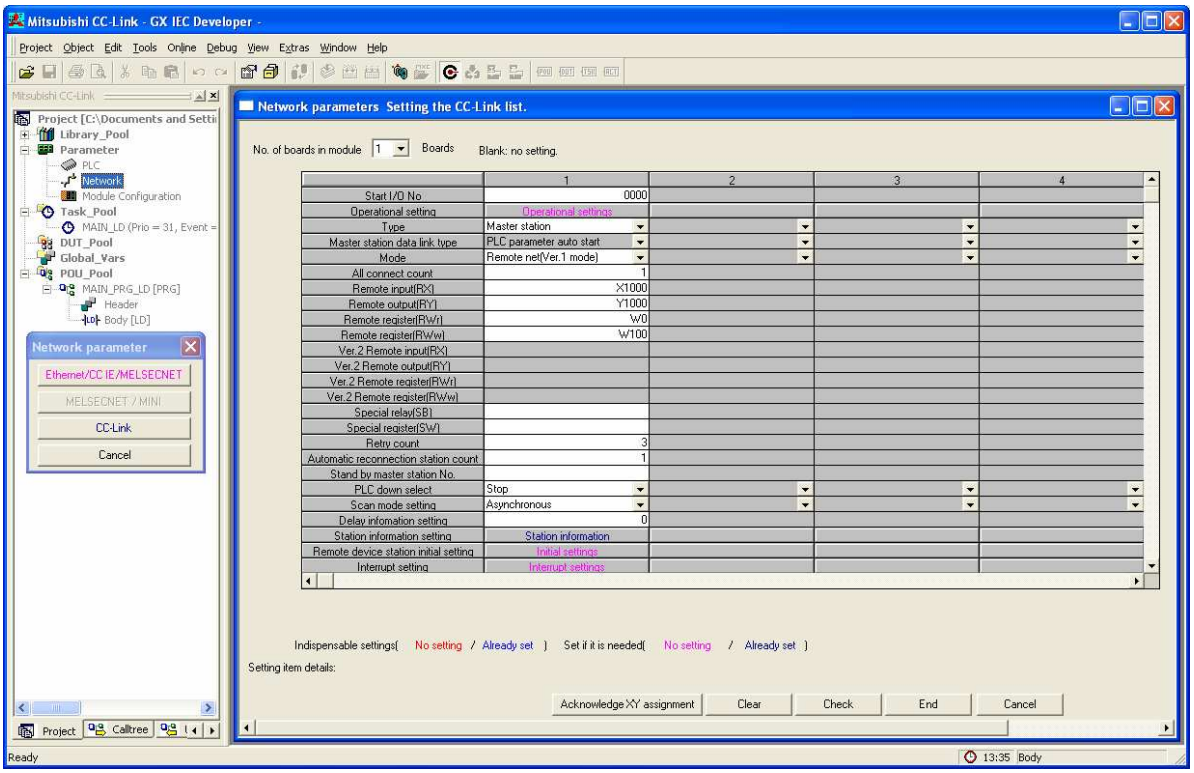


Figure 1: CC-Link master configuration

2.2. Slave configuration

The current implementation in the MC464 only supports CC-link Version 1. With up to 4 stations addressable. The final 16 bits are used for handshaking and the final word write point is used for an error word.

When you map data to the MC464 it automatically determines how many stations to activate from the amount of data.

The maximum amount of data that can be mapped is detailed in Figure 2.

Occupied Stations	Bit points read	Bit points write	Word points read	Word points write
1	16	16	4	3

2	48	48	8	7
3	80	80	12	11
4	112	112	16	15

Figure 2: Station data

2.3. Handshaking

The handshaking is performed using the final 16 bits read and write.

Bits	Slave -> Master	Master -> Slave
0 - 7	Reserved	Reserved
8	Initial data processing request flag	Initial data processing complete flag
9	Initial data setting complete flag	Initial data setting request flag
A	Error status flag	Error reset request flag
B	Remote READY	Reserved
C - F	Reserved	Reserved

Figure 3 Handhake word

Using the configuration in Figure 1, the formula for the memory offset is 10(m+n)0 Hex, where m is depending on the station number and n on the number of occupied stations. The relations are:

m=(station number-1)*2 and

n=number of occupied stations*2-1.

In this example the station number is 1 therefore the m-factor is zero. The n-factor will have the values 1,3,5,7 for 1,2,3,4 occupied station(s). In this example the number of occupied stations is 2 and the m+n-factor consequently is 3. So the memory offset for the handshaking area is 1030 Hex and the complete address for the remote ready flag will be 103B as shown in figure 10.

The handshaking requires of waiting for bit 8, Initial data processing complete flag to be true. Then set bit 8 in the write area. You should then see bit B, remote ready become true and the Anybus module will enter process active.

This can be manually done using the Entry Data Monitor (on a Mitsubishi PLC) or automatically in a program. The following example could be used in structured text.

```
IF (X1038=1 AND X103B=0) THEN
          Y1038=1;
ELSE
          Y1038=0;
END_IF;
```

3. Anybus Status byte

The following example shows how the Anybus statues can be read. This checks for a change in state and if the module is supervised on the network. It displays the information on one of the terminal channels.

```
Read_state:
 VR(0)=ANYBUS(3,slotnum)
    rdanybus_state=READ_BIT(2,0)*$4+READ_BIT(1,0)*$2+READ_BIT(0,0)
    IF rdanybus_state<>anybus_state THEN
      anybus_state=rdanybus_state
      PRINT#term, "ANYBUS CC CHANGED STATE"
      PRINT#term, "    Anybus State = ";
      ON anybus_state+1 GOSUB s0,s1,s2,s3,s4,s5,s6,s7
      PRINT#term, ""
      anybus_state=rdanybus_state
    ENDIF

    'check for change in supervisory bit
    IF supbit<>READ_BIT(3,readbit) THEN
      supbit=READ_BIT(3,readbit)
      IF READ_BIT(3,readbit)=0 THEN
        PRINT#term, "Module is not supervised"
      ELSE
        PRINT#term, "Module is supervised by another network device"
      ENDIF
    ENDIF

RETURN

'Anybus State list
s0:
  PRINT#term, "SETUP"
  RETURN
s1:
  PRINT#term, "NW_INIT"
  RETURN
s2:
  PRINT#term, "WAIT_PROCESS"
  RETURN
s3:
  PRINT#term, "IDLE"
  RETURN
s4:
  PRINT#term, "PROCESS_ACTIVE"
  RETURN
s5:
  PRINT#term, "ERROR"
  RETURN
s6:
  PRINT#term, "(reserved)"
  RETURN
s7:
  PRINT#term, "EXCEPTION"
  RETURN
```