

1) SNMPv2c Demo Application Introduction

This information on the SNMPV2c agent should answer the following FAQs:

1. Why are there two separate mib scripts (mchip.mib and snmp.mib) with Microchip TCP/IP Stack SNMP support?
2. What are the Tools needed to compile and upload the MIB scripts to the agent and the manager?
3. How will the SNMPv2c agent in version 5.00+ of the TCP/IP stack support the existing SNMPv1 applications implemented with previous stack releases?
4. How is the SNMP MIB browser used, and how are MIB scripts uploaded and interpreted?

The SNMP stack in the Microchip TCP/IP Stack suite is separated in two parts:

1. SNMP stack implementation (SNMP.c)
2. Application and demo specific implementation (CustomSNMPApp.c).

The SNMPv2c agent is provided with the Microchip TCP/IP Stack v5.00 and later. The earlier releases of the stack included the SNMPv1 agent (referred to as the V1 agent in this document). The Microchip SNMPv2c agent (referred to as the V2c agent) follows RFC 3416 and is implemented to address the requirements of embedded applications. The V2c agent supports and responds to V1 queries too.

The V2c agent is implemented with multiple community name configuration support. The configured community names are stored in the selected non volatile memory (SPI EEPROM or SPI Flash). The community defaults can be configured through the *TCP/IP Configuration Wizard*. The communities can also be dynamically configured using the HTTP interface. A separate, access restricted web page is provided in the Microchip TCP/IP Stack v5.00+ HTTP2 server for dynamic configuration of SNMP communities.

1.1) Note: For existing Microchip SNMPv1 users

The Microchip TCP/IP Stack SNMPv1 user who wants to upgrade the Microchip TCP/IP Stack from older versions to the latest version but still want to use SNMP v1 can get the SNMP v1 services from the V2c agent, provided they do not modify the default settings of the SNMP module in v5.00+.

The implementation framework for V1 and V2c remains the same, except there are new features, new functions have been added, and some of the function name and parameters have changed. The existing users of V1 may need to make some changes to their application-specific code. There should not be any changes in the V2c stack code unless the user has incorporated application code in the SNMP stack.

2) Microchip SNMPv2c MIB Scripts and Corresponding Tools

The Microchip SNMP agent has 2 MIB files: **snmp.mib** (Microchip Custom MIB script) and **mchip.mib** (ASN.1 MIB script).

The Microchip TCP/IP Stack includes the “**mib2bib.exe**” utility, which compiles the Microchip MIB script and outputs the “**snmp.bib**” and “**mib.h**” files. The “**snmp.bib**” file should be copied to “C:\Microchip Solutions\TCPIP Demo App\WebPages2\” directory. The “**mib.h**” file should be copied to “C:\Microchip Solutions\TCPIP Demo App\”.

Next, use the “**MPFS2.exe**” tool. The “Getting Started \ Uploading Web Pages” section gives more insight on this tool. With this utility the compiled SNMP MIB script is bundled into the “**MPFS2Img.bin**” file and is stored to the volatile memory. The SNMP agent searches for the required MIB variable information in this MPFS image using MPFS file system support.

Compile the “**TCPIP Demo App-Cxx**” project along with the “**mib.h**” file copied in “TCPIP Demo APP”. The `#defines` in “**mib.h**” file are required by the SNMP agent. DO NOT make any changes in the “**mib.h**” file. The “**mib2bib.exe**” utility will automatically generate this file.

The “**mchip.mib**” file is the ASN.1 MIB script for the SNMP managers. The SNMP managers (browsers) understand the ASN.1 format. Once the SNMP manager installation is done, this file should be uploaded to the MIB directory of the manager. The following sections will provide more information on how to use SNMP managers and how to upload the MIB script.

2.1) SNMP Browsers and MIB scripts

There are large numbers of SNMP MIB browsers available. Also, the user can implement a customized MIB browser specific to their application. For the purposes of this getting started guide, we will be discussing the **iReasoning** MIB browser for the Microchip SNMP Demo application. The procedure for MIB script upload, the MIB tree structure display and the SNMP query mechanism varies slightly from browser to browser. The browser used for SNMP specific applications completely depends on the user’s discretion.

2.2) SNMP Browser and MIB script Upload

The iReasoning MIB browser can be downloaded from <http://www.ireasoning.com/downloadmibbrowserlicense.shtml>. Once the browser installation is done, copy the “**mchip.mib**” file to “C:\Program Files\ireasoning\mibbrowser\mibs”.

Go to the “File->Load MIBs” menu and select the “**mchip.mib**” file.
 You should be able to see the Microchip MIB directory in the left pane (“SNMP MIBs” pane) of the browser (Figure 1).

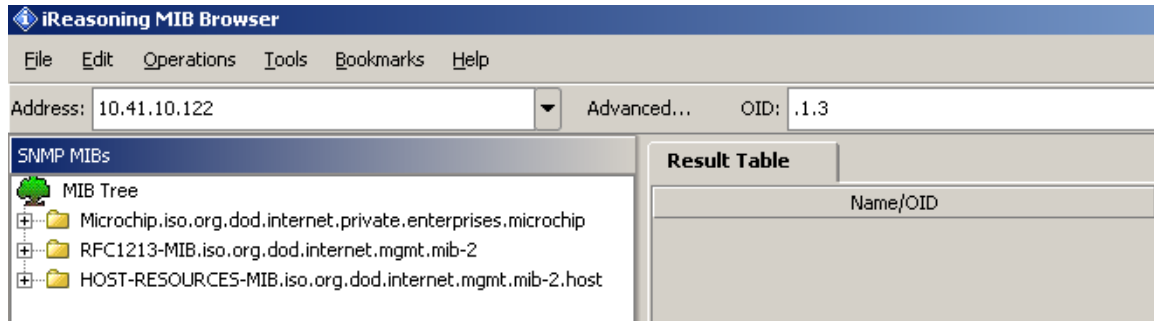


Figure 1: The Microchip “SNMP MIBs” Directory

Figure 2 shows the RFC1213- MIB2 variables implemented in the Microchip SNMP agent. This is the minimum set of variables required to identify the node as an SNMP node in a network. These variables can be accessed by any SNMP browser with a “public” type community name. Please refer to the Microchip SNMP App Note (AN870) for more details on the MIB script and Demo SNMP MIB variable tree structure.

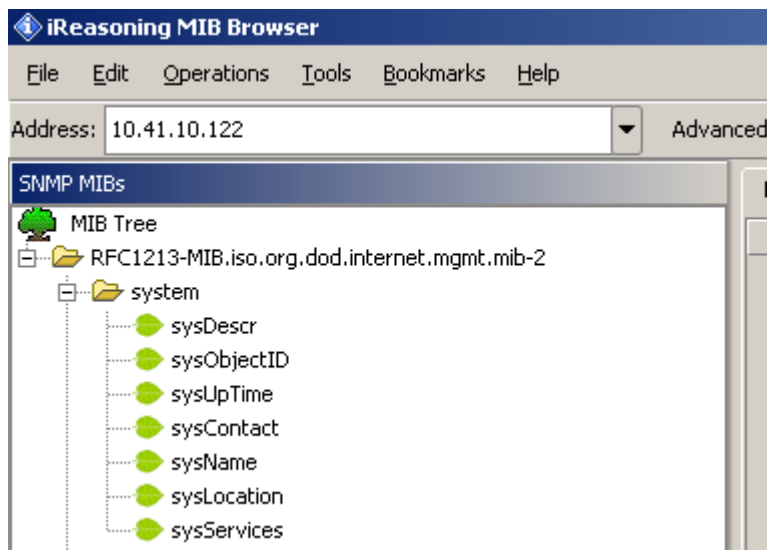


Figure 2: RFC1213 MIB2 Variables

Figure 3 shows the Microchip private MIB variable tree structure in the browser. The ASN.1 format “**mchip.mib**” file is defined with this tree structure for the MIB variables. The browser can access every variable in the MIB database provided the community name matches.

Note: Whenever user needs to modify the MIB variables, the corresponding changes should be made in both the MIB scripts (“snmp.mib” and “mchip.mib”).

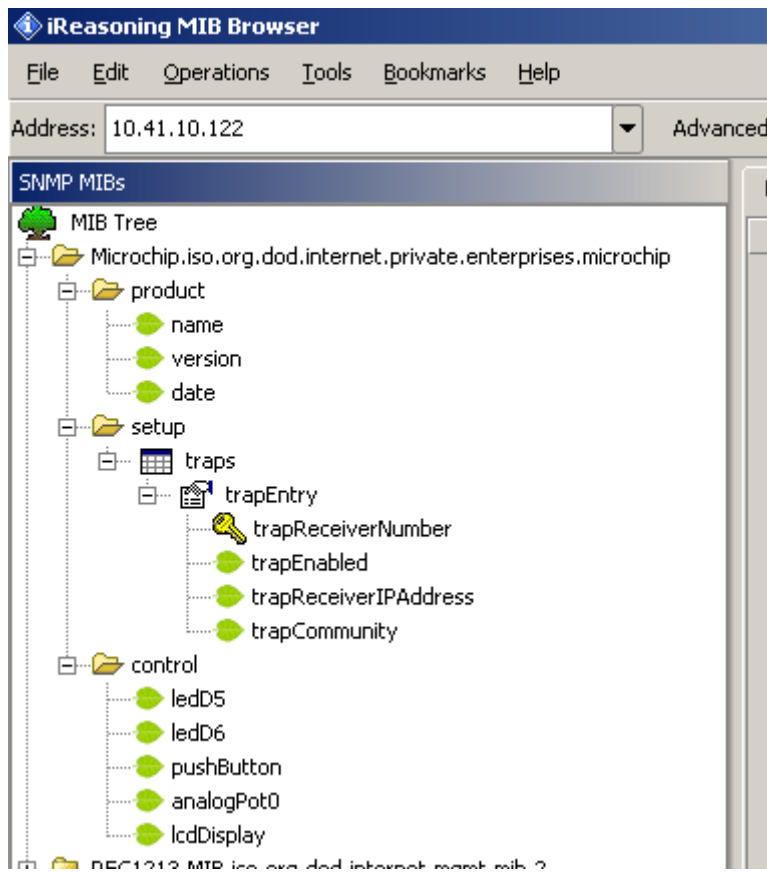


Figure 3: Private MIB Variable Tree Structure

2.3) Configuring the Browser for SNMP Version and Community

Select the “Advanced” Tab in the browser

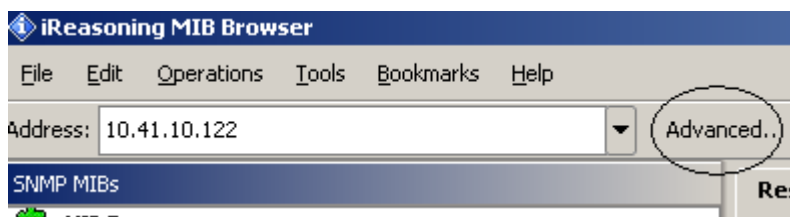


Figure 4: Advanced Tab Location

The Configuration Window shown in Figure 5 will appear.

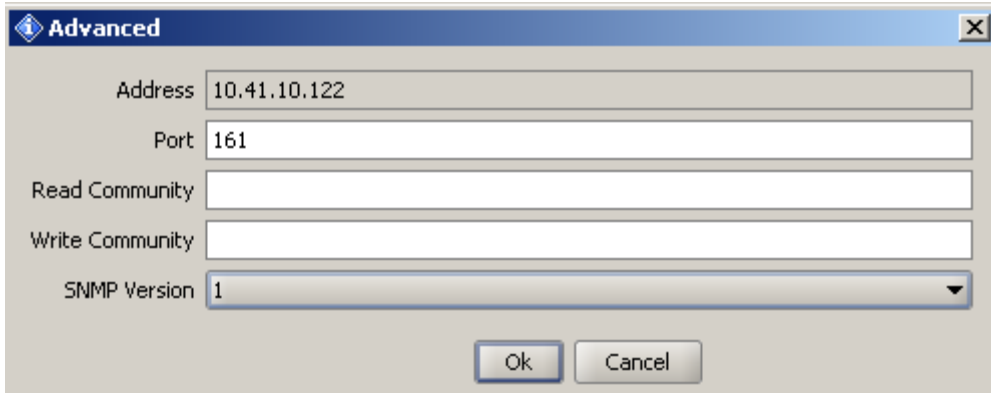


Figure 5: Advanced Configuration Window

Configure the Read and Write community to the browser. If the community fields are left blank, the manager sends the SNMP Request with the community name specified as “public”. The V2c agent is configured by default with 3 Read {“public”, “read”, “ ”} and 3 Write communities {“private”, “write”, “public”}, the default maximum community length is 8 characters. As the default communities also contains the “public” community name, the agent will respond to all the browsers requesting with “public” community. The **TCPIP Configuration Wizard** can be used to configure the default SNMP community names. At run time, the community names can be dynamically configured using the HTTP interface dedicated for SNMP community name configuration.

Note: The V2c agent will respond only to the queries from the SNMP browsers of the same community. That is, the V2c agent and the browser should be the member of the same community.

If the V2c agent receives an SNMP request with an unknown community name specified, the agent will generate an Authentication Trap.

The V2c agent’s multiple community support enables the user implementation to provide limited access to the requesting browser, depending on the community name used by the browser, to access the MIB database variables of the agent.

2.4) MIB Variable and Operation Selection

The variable to be accessed from the agent MIB database can be selected from the “SNMP MIBs” pane. The selected variable’s OID can be seen in the OID tab.

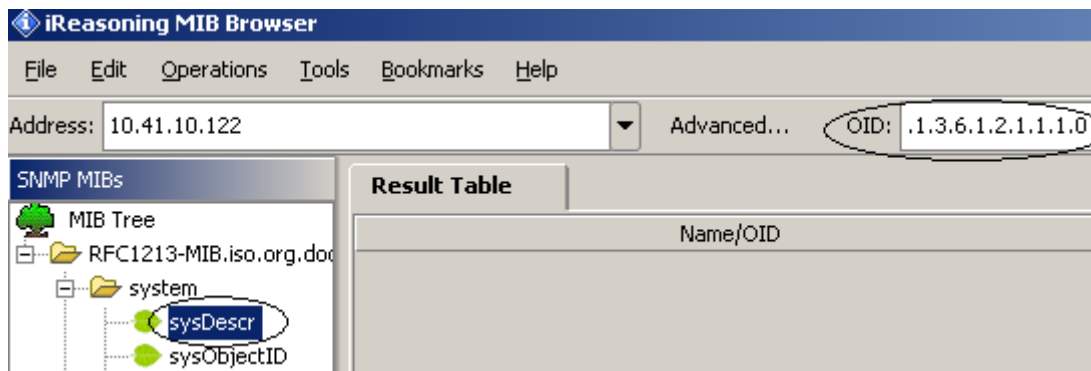


Figure 6: Variable in Agent MIB Database and its OID

The SNMP Request is selected from the “Operations” tab.



Figure 7: The SNMP Request and Operations tab

3) Exploring the Microchip SNMPv2C agent Demo

After the MIB script has been uploaded to the SNMP browser, the MIB tree structure will be displayed in the browser as shown in previous figures. Any of these variables can be accessed (SNMP operations can be performed) from the agent if the agent supports these variables. The browser and agent should be members of the same community. To learn more about the SNMP operations, PDU types, and terminologies, refer to SNMP App note AN870.

Configure the IP address of the V2c agent to the “Address” tab.

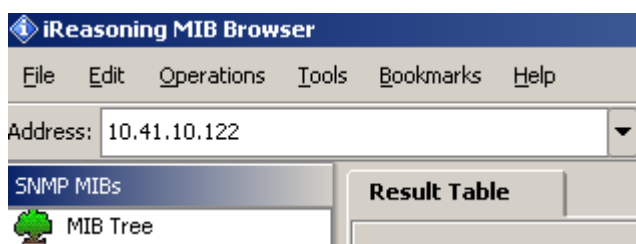


Figure 8: Configuring the V2c IP Address

3.1) Get

1. Select 'Advanced'. Configure the SNMP version as 'V1', and the Read Community as 'public'.
2. Select the *system* variable 'SysDescr' from the MIB structure. Select 'Get' from 'Operations' and press 'Go'.

3. 'Result Table' will display information of the 'SysDescr' variable.

This procedure can be repeated for any other MIB variables. For SNMP version V2c, repeat the above procedure, substituting V2 for V1 in the Advanced->SNMP Version configuration.

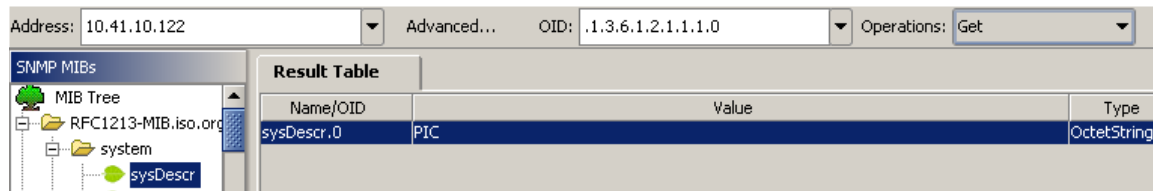


Figure 9: Configuration of a Get Operation

The V2c agent is provided with three Read and Write community defaults as explained earlier. Configure any of these communities to the browser and try accessing the MIB variables. You should be able to access all the MIB variables even with 'write' communities configured as Read Community. For GET operations, if the Read or Write community matches, the agent processes the request. For SET operations, the received community name must match to any of the configured 'write' community names.

3.2) Get_Next

1. Repeat the process for 'Get'. Select 'SysDescr' variable from MIB. Select 'Get Next' from 'Operations'.
2. 'Result Table' will display information of 'sysObjectId' variable which is next to 'SysDescr' variable.

Repeat for different MIB variables and get information of corresponding next variable.

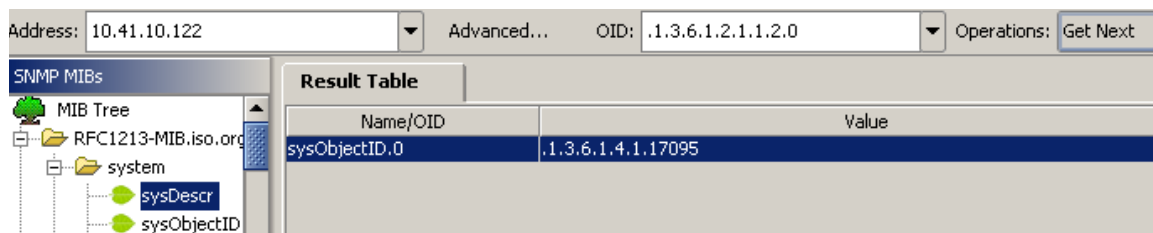


Figure 10: Configuration of a Get_Next operation

3.3) Get_Bulk

This operation is supported in SNMPv2c.

1. Configure SNMP version as V2 in SNMP browser.
2. Select 'SysDescr' variable from MIB.
3. Select 'Advanced'. Configure SNMP version = 'V2', Read Community = 'public' or 'read'.

4. Select 'Get Bulk' from 'Operations' and press 'Go'.

'Result Table' will display information for 10 MIB variables in a single request. These variables are lexicographical successors of the 'SysDescr'. Get_Bulk enables collection of bulk information from the agent with a single request from the manager. The number of variables that the agent will respond with is configurable and should be configured with the browser. It can be configured in 'Tools->Options->Non Repeaters' and 'Tools->Options->Max Repetititons' menus. These two numbers are extracted by the agent from the received Get_Bulk request, and the number of variables to insert in request is calculated.

Refer to RFC 3416 for detailed information on calculating the number of variables in response and what are Non-Repeaters and Max-Repetitions.

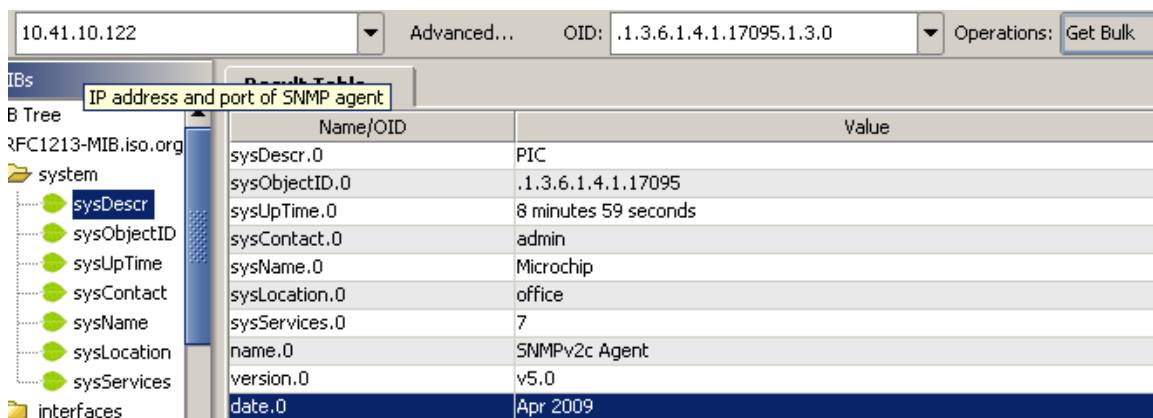


Figure 11: Configuration of a Get_Bulk Operation

3.4) Set

The **Set** command updates the variable information of the MIB database of the agent. The **Set** command can be performed only on those variables which are declared as 'READWRITE' in the MIB scripts, and only if the community name matches any one of the 'write' community names configured with the agent.

1. Select 'ledD5' variable from MIB.
2. Select 'Advanced'. Configure SNMP version = 'V1 or V2'. Write Community = 'public' or 'write' or 'private'.
3. Select 'Set' from 'Operations'.
4. The "SNMP SET" window will pop up (Figure 13). Enter the value for the browser.
5. A success message should appear in the browser.



Figure 12: Configuration of a Set Operation

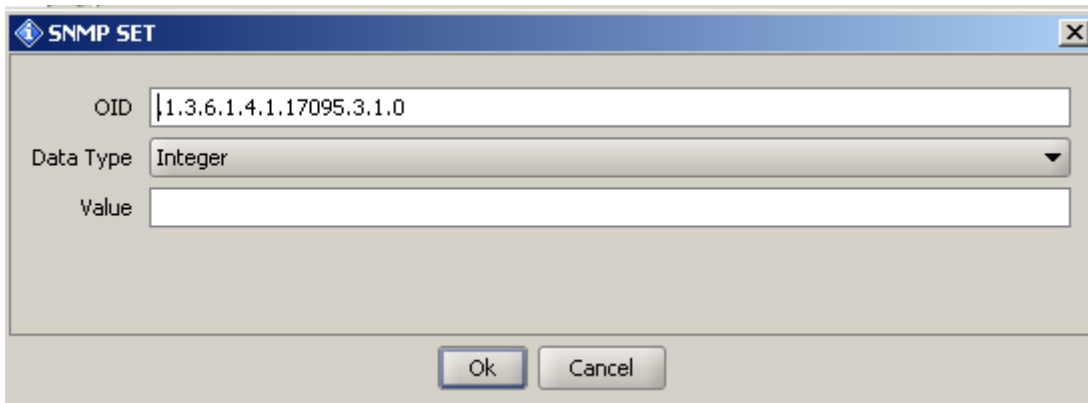


Figure 13: The “SNMP Set” Window

A ‘Get’ operation for the same variable should now return the new ‘Set’ value for this variable. The LED5 on the PICDEM.net 2 or the Explorer16 Board should be UP. A ‘Get’ operation on the *LED5* will display the LED5 as ‘ON’. Repeat the procedure to ‘Set’ LED5 to OFF. LED6 can also be set ON and OFF.

3.5) Traps

Traps are notifications from the agent to the manager that are used if a predefined event occurs at the agent. For the Trap Demonstration, two events are defined with the V2c agent.

- 1) If the Analog Pot value is more than 512, the agent sends Trap every 5 seconds to the configured “*trapReceiverIPAddress*”.
- 2) If the ‘Push Button 0’ is closed, a trap will be sent.

For the Trap Demo, make sure that the following `#define` preprocessor directive in the TCPIPConfig.h file is defined:

```
// Comment following line if SNMP TRAP DEMO not required
#define SNMP_TRAP_DEMO_ENABLED
```

The current implementation of the V2c agent also generates an “Authentication Failure Trap”, if the private MIB variable values are requested to modify (Set) a variable or if the value of the variable is requested (Get) by browsers with the wrong community name.

1. Select ‘*Advanced*’. Configure SNMP version = ‘V2’ and Write Community = ‘*public*’ or ‘*write*’ or ‘*private*’.
2. Select ‘*trapEnabled.0*’ variable from MIB.
3. Select ‘*Set*’ from ‘*Operations*’.
4. The “SNMP SET” window will pop up. Enter ‘1’ in the value field.
5. Select ‘*trapReceiverIPAddress.0*’. **Set** the IP address if the PC on which the SNMP browser is installed and running.
6. Select ‘*trapCommunity.0*’. **Set** the community name of the SNMP browser. The ‘*trapCommunity*’ should be configured to the community name at which the browser will receive the trap. If the ‘*trapCommunity*’ name is not configured, then every SNMP browser on the PC will receive the trap. The ‘*trapCommunity*’ name will work as a filter for the SNMP browser to receive the trap.
7. Open the Trap Receiver utility of the MIB iReasoning MIB browser (Start->All Programs->iReasoning->MIB Browser->Trap Receiver).

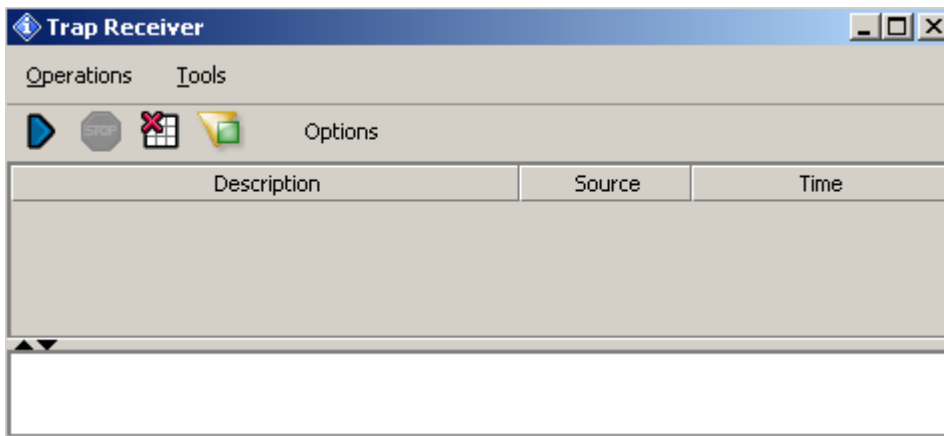


Figure 14: The Trap Receiver Utility

All the necessary configurations as explained for the trap demo are followed.

1. **Analog Pot Trap:** Adjust the Analog Pot on the PICDEM.net2 board or Explorer16 Board to a value of more than 512. The received trap is shown in Figure 15. This is enterprise specific trap. The SNMP manager will receive the source IP address, the OID (as the name of the variable), the value, the time stamp, etc. for each specific event that occurs. The browser will interpret the data as AnalogPot variable information based on the OID name.

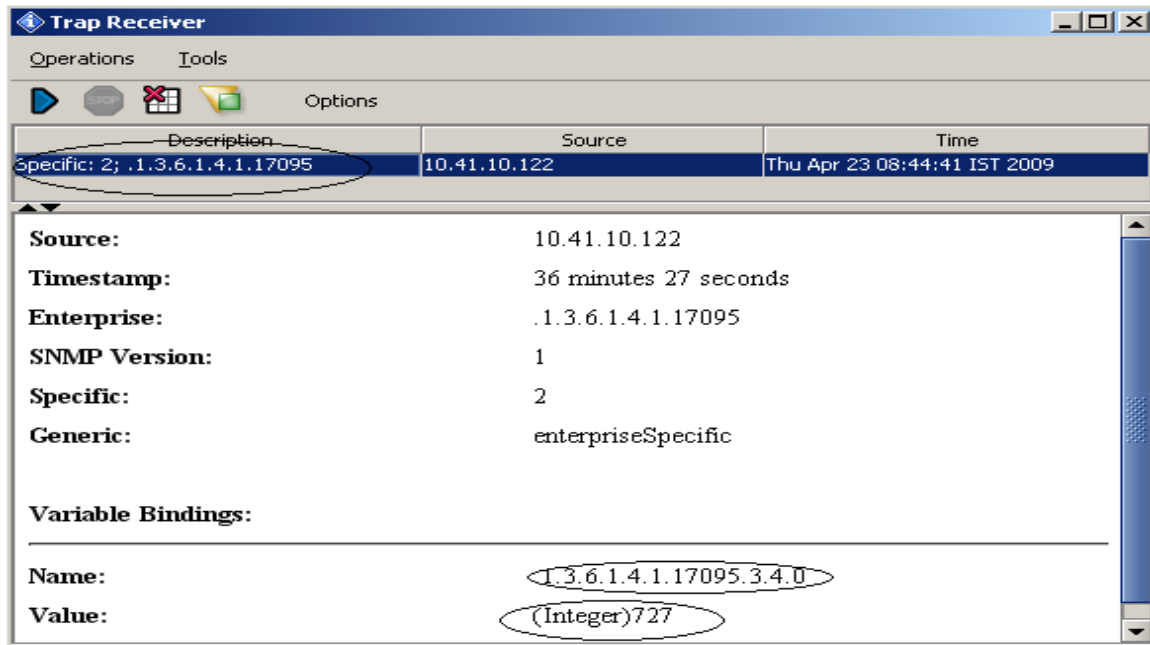


Figure 15: Analog Potentiometer Trap

2. **Push Button Trap:** Pressing the appropriate button on the development board ('RB0' on PICDEM.net Board, 'S3' on Explorer 16 Board) will cause the corresponding trap to be received by the '**Trap Receiver**'.
3. **Authentication Failure Trap:** By default the Read Communities supported are 'public' and 'read'. Configure a Read Community name in the 'Advanced' tab that is not one of the Read community names supported by the V2c agent. For example, configure 'mchp' as a Read Community to the browser. Select the private MIB variable '*LED5*' from the MIB tree and issue a 'Get' operation from the browser. The '*Result Table*' of the browser will not display any result but the '*Trap Receiver*' receives an 'Authentication Failure' Trap. This is an intimation from the agent to the SNMP manager that there is an unauthorized attempt to access the private MIB variable of the data base.

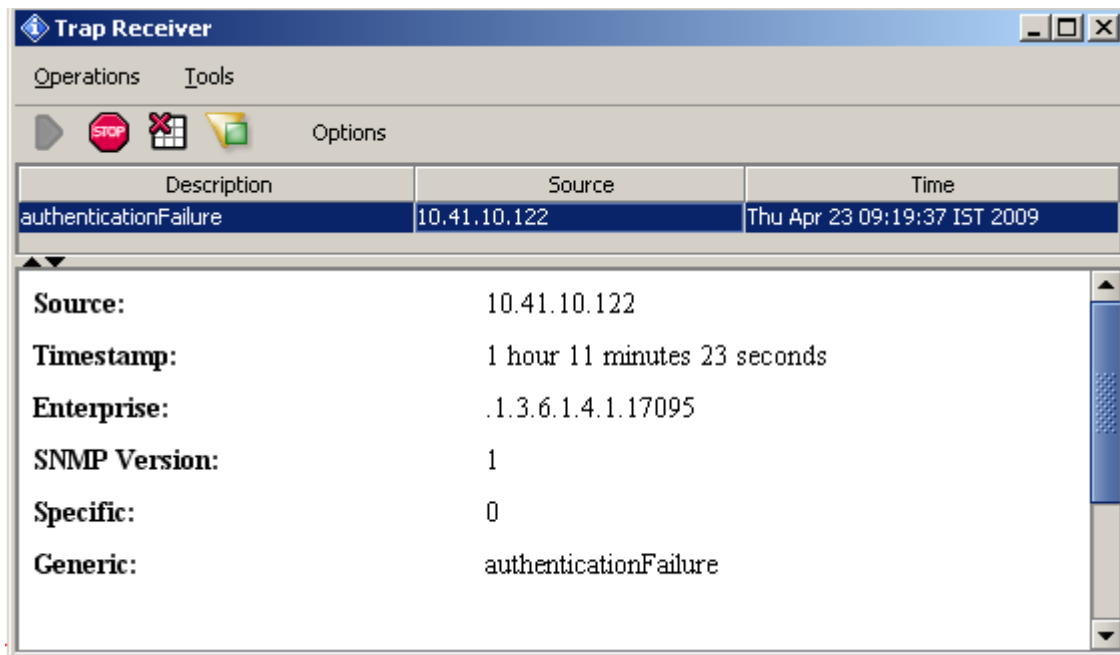


Figure 16: An Authentication Failure Trap

4) Configuring Multiple Communities to the V2c agent

If you are using an HTTP2 server with the Microchip TCPIP Stack, you can dynamically configure the Read and Write community names to the V2c agent from the “SNMP Configuration” web page. Authentication will be required to access this web page. By default, the username is “admin” and the password is “microchip”.

Overview

Dynamic Variables

Form Processing

Authentication

Cookies

File Uploads

Send E-mail

Dynamic DNS

Network
Configuration

SNMP
Configuration

SNMP Community Configuration

Read/Write Community String configuration for SNMPv2c Agent.

Configure multiple community names if you want the SNMP agent to respond to the NMS/SNMP manager with different read and write community names. If less than three communities are needed, leave extra fields blank to disable them.

Read Comm1 :	<input type="text" value="public"/>
Read Comm2 :	<input type="text" value="read"/>
Read Comm3 :	<input type="text"/>
Write Comm1:	<input type="text" value="private"/>
Write Comm2:	<input type="text" value="write"/>
Write Comm3:	<input type="text" value="public"/>
<input type="button" value="Save Config"/>	

Figure 17: The SNMP Configuration Web Page