Twitter V1
Module Application Guide

Description

This module allows you to emulate the functionality of the micro blogging service Twitter. To see more information on the service go see www.twitter.com. All you need is an Ethernet enabled processor and a Twitter account. Functions include viewing both public and friend posts as well as updating your own posts. Use the module to follow your friends or even track late breaking news from media outlets such as Fox, The New York Times, CNN, MSNBC and NPR.

Internet Disclaimer

Before using this module you should read the Twitter terms of service available at https://twitter.com/terms to ensure that your application of the module meets these terms and conditions. This module supports the Twitter REST API.

This module requires Internet access and depends on the output of one or more 3rd-party Internet websites. As such, ControlWorks Consulting is unable to guarantee the formatting, content, availability, or reliability thereof. In the event of a permanent outage or change in formatting, ControlWorks may or may not elect to issue an update to purchasers of the module.

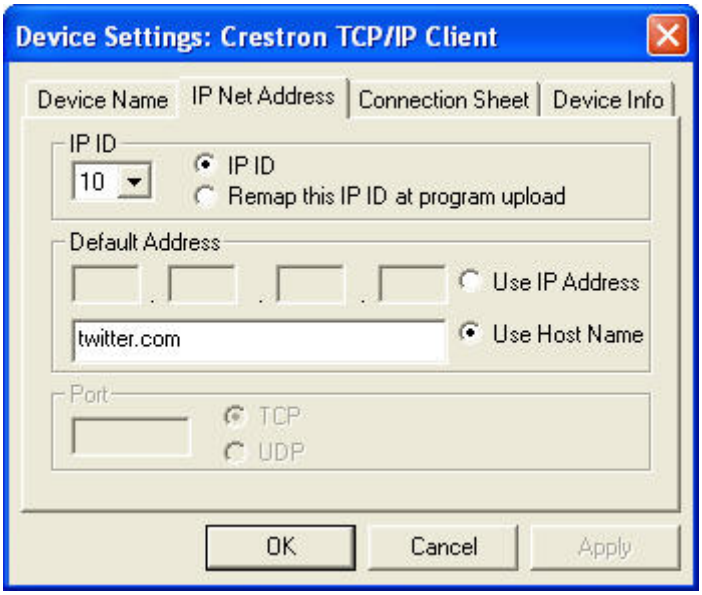
ControlWorks Consulting accepts no responsibility for the accuracy or your usage of data returned by the module.

Compatibility			Processor Requirements	
2-Series Compatible	NOT CNMSX Compatible	System Builder Compatible	Ethernet REQUIRED	Compact Flash NOT NEEDED

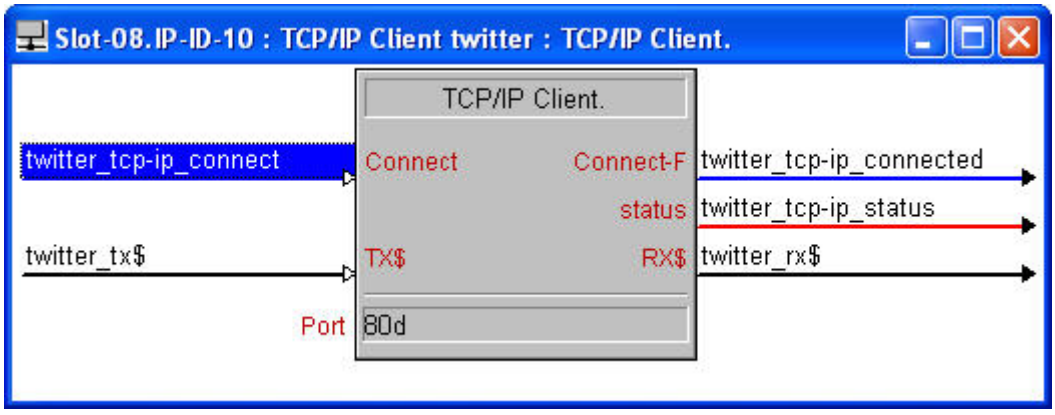
Ethernet Configuration Information

The processor must have DNS servers and hostname declared in the DNS Management and Ethernet Addressing setup. To declare DNS servers from Toolbox, connect to your processor and select “DNS Management …” from the “Functions” menu. Use the DNS server IP addresses provided by the Internet Service Provider of the project, or from open DNS servers available on the internet. Be sure to test the settings by using the “Test…” button and entering a domain name such as www.twitter.com. Declare a Hostname by selecting “Ethernet Addressing…” from the “Functions” menu. Enter a Hostname in the “Host Name” field. The Hostname can be any combination of letters, numbers and select symbols, as long as the Hostname is unique on the network.

Insert a TCP/IP client into your program. Configure the client as shown below, making sure you use the host name twitter.com in the “Use Host Name” field. The IP ID will vary depending on where it is inserted into the program.

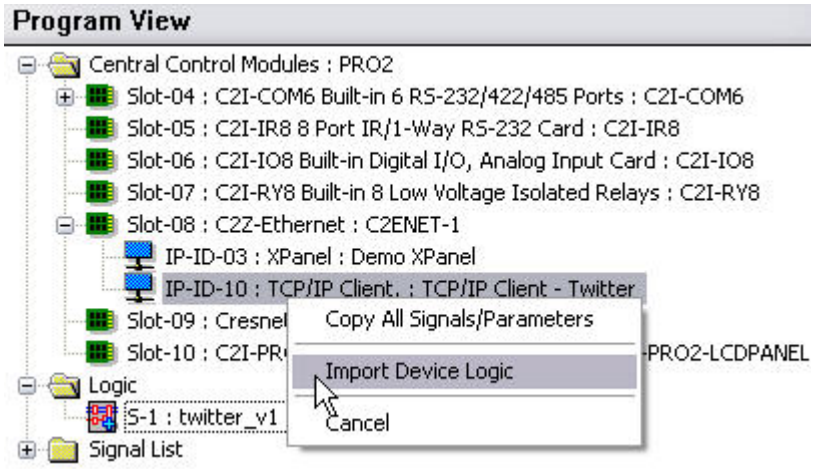


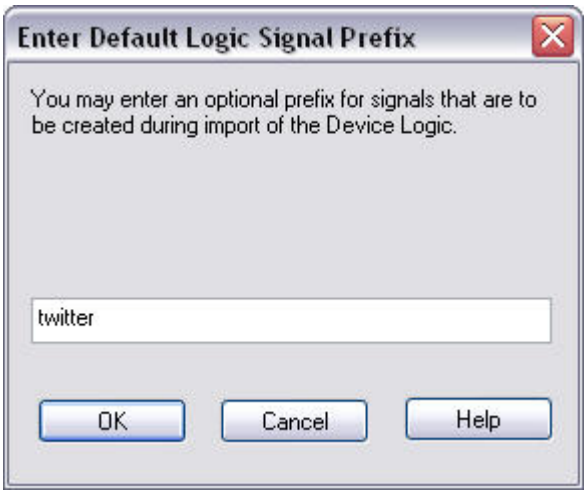
Declare the TCP/IP client symbol as shown below, making sure the port is set to 80d.



Virtual Connection Logic

This module is built utilizing virtual connection logic that will allow you to right-click and drag the module symbol to a TCP/IP client symbol, select “Import Device Logic” to auto-connect the two symbols. You will be prompted to add an optional signal prefix to the auto-connected signals. SIMPL will add an underscore character at the end of the prefix you enter automatically, so a space or underscore does not need to be entered in this dialog. Due to Crestron’s implementation of this feature, the programmer will still need to manually enter the port number (80d) on the TCP/IP client symbol.



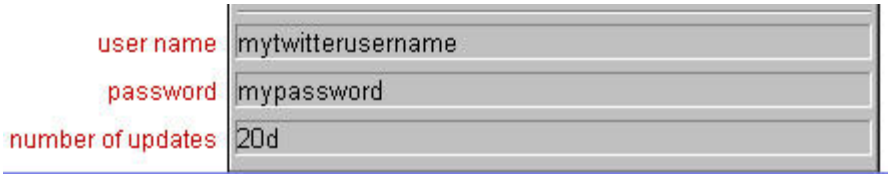


Twitter Account Setup

The first step in getting the module running is to setup a free account. Go visit www.twitter.com and setup your account. Once your account is setup you can go to the Twitter **Settings** tab to update your profile with your location, a personal Bio, picture and other relevant information to allow friends to find you more easily. Under the **Find People** tab you can search for friends on Twitter or on other services such as Gmail, Yahoo!, AOL, Hotmail and MSN. Here you can also search for services to follow like CNN. Make note of your user name and password as you will need them in the next step.

Module Setup

Open up the demo program and scroll down to the bottom of the module. Here you will find where to enter your own user name and password on the module. Replace "mytwitterusername" and "mypassword" with your own Twitter credentials.



Signal And Parameter Descriptions

Bracketed signals such as "[signal_name]" are optional signals

DIGITAL INPUTS

tcp-ip_connected ..... Routed from the tcp-ip client symbol. High indicates that the tcp-ip connection is open.

[get_friends_updates] ..... Pulse to query for the user's (defined in the parameters) and user's friends posts. Friends are the other twitter accounts that the user is following.

[get_user_updates] ..... Pulse to query for the user's (defined in the parameters) posts.

[get_public_updates] ..... Pulse to query for all public posts

[send_update] ..... After entering text using the keyboard, pulse this to post the update contained in [update_text$] to Twitter.

[keyboard_shift] – [keyboard_space] ..... Standard QWERTY keyboard functions. Pulse each to enter text into the string output [update_text$].

ANALOG INPUTS

tcp-ip_status..... Routed from the tcp-ip client symbol. Indicates the status of the tcp-ip client.

SERIAL INPUTS

rx$. ..... Routed from the tcp-ip client symbol. Contains all the data received from the server.

DIGITAL OUTPUTS

tcp-ip_connect ..... Route to the tcp-ip client symbol. Latched high when the module requires the tcp-ip connection to be opened. Do not use any other logic to drive the connect input on the TCP-IP client symbol.

ANALOG OUTPUTS

[keyboard_multimode] ..... Route to touchpanel definitions for keyboard buttons. Enables modes for shifted and non-shifted keys.

[update_characters_left] ..... Indicates the number of characters available for the current update text. Twitter limits updates to 140 characters.

SERIAL OUTPUTS

[tweet_created_at$[1]] - [tweet_created_at$[20]]..... String containing the timestamp for the received update

[tweet_text$[1]] – [tweet_text$[20]]..... String containing the body of the received update

[tweet_source$[1]] – [tweet_source$[20]]..... String containing the source of the received update

[tweet_screen_name$[1]] – [tweet_screen_name$[20]] String containing the screen name of the user that posted the received update

[tweet_image_url$[1]] – [tweet_image_url$[20]] ..... String containing the url to the image of the user that posted the received update. Route this to a dynamic graphic element on a VTPro project.

[update_text$]..... String containing the output of the keyboard. This is also sent to twitter upon pulsing the [send_update] digital input.

The following strings are returned from the server upon successful transmission of an update:

[last_tweet_created_at$]..... String containing the timestamp for the last update sent

[last_tweet_text$] ..... String containing the body of the last update sent

[last_tweet_source$] ..... String containing the source of the last update sent

[last_tweet_screen_name$] ..... String containing the screen name of the last update sent

[last_tweet_image_url$]..... String containing the URL to the user image from the last update sent

PARAMETERS

user name ..... User name of the Twitter account to use for sending and receiving updates

password..... Password for the above Twitter user

number of updates..... Number of updates to include when requesting user, friend, or public updates. Twitter limits clients to a maximum of 20. Acceptable values are 1d to 20d.

port ..... The tcp-ip port to be used on the tcp-ip client symbol. This parameter has a default value of 80d and cannot be changed to a different value. This parameter is defined to enable Systembuilder compatibility.

Support

This module is supported by ControlWorks Consulting, LLC. Should you need support for this module please email support@controlworks.com or call us at 440-449-1100. ControlWorks normal office hours are 9 AM to 5 PM Eastern, Monday through Friday, excluding holidays.

Before calling for support, please ensure that you have loaded and tested operation using the included demonstration program and touchpanel(s) to ensure that you understand the correct operation of the module. It may be difficult for ControlWorks to provide support until the demonstration program is loaded.

Updates, when available, are automatically distributed via Email notification to the address entered when the module was purchased. In addition, updates may be obtained using your username and password at http://www.thecontrolworks.com/customerlogin.aspx .

Distribution Package Contents

The distribution package for this module should include:

twitter_v1.umc	Crestron User Module
twitter_v1.usp	SIMPL+ file used within the module
twitter_v1.ush	SIMPL+ header file
Base64 Encode Engine v1.usp	SIMPL+ file used within the module
Base64 Encode Engine v1.ush	SIMPL+ header file
twitter_xpanel_v1.vtp	Demo touchpanel file
twitter_demo_v1.smw	Demo program for a PRO2 processor
Twitter_help_v1.pdf	Module help file

Revision History

V1 gary@controlworks.com 2009.02.05
-Initial release

Development Environment

This version of the module was developed on the following hardware and software. Different versions of hardware or software may or may not operate properly. If you have questions, please contact us.

Crestron Hardware	**Firmware Version**
Crestron PRO2 Processor	3.155.1240
Crestron Software	**Software Version**
Crestron SIMPL Windows	2.11.03
Crestron Vision Tools Pro-e	3.9.23
Crestron Database	20.03.008
Crestron Library	508

ControlWorks Consulting, LLC Module License Agreement

Definitions:
ControlWorks, *We*, and *Us* refer to ControlWorks Consulting, LLC, with headquarters located at 701 Beta Drive, Suite 22 Mayfield Village, Ohio 44143-2330. *You* and *Dealer* refer to the entity purchasing the module. *Client* and *End User* refer to the person or entity for whom the Crestron hardware is being installed and/or will utilize the installed system. *System* refers to all components described herein as well as other components, services, or utilities required to achieve the functionality described herein. *Module* refers to files required to implement the functionality provided by the module and may include source files with extensions such as UMC, USP, SMW and VTP. *Demo Program* refers to a group of files used to demonstrate the capabilities of the Module, for example a SIMPL Windows program and VisionTools Touchpanel file(s) illustrating the use of the Module but not including the Module. *Software* refers to the Module and the Demo Program.

Disclaimer of Warranties
ControlWorks Consulting, LLC software is licensed to You as is. You, the consumer, bear the entire risk relating to the quality and performance of the Software. In no event will ControlWorks Consulting, LLC be liable for direct, indirect, incidental or consequential damages resulting from any defect in the Software, even if ControlWorks Consulting, LLC had reason to know of the possibility of such damage. If the Software proves to have defects, You and not Us must assume the cost of any necessary service or repair resulting from such defects.

Provision of Support
We provide limited levels of technical support only for the most recent version of the Module as determined by Us. We do not provide support for previous version of the module, modifications to the module not made by Us, to persons who have not purchased the module from Us. In addition, we may decline to provide support if the Demo Program has not been utilized. We may withdraw a module from sale and discontinue providing support at any time and for any reason, including, for example, if the equipment for which the Module is written is discontinued or substantially modified. The remainder of your rights and obligations pursuant to this license will not be affected should ControlWorks discontinue support for a module.

Modification of Software
You may not decrypt (if encrypted), reverse engineer, modify, translate, disassemble, or de-compile the Module in whole or part. You may modify the Demo Program. In no event will ControlWorks Consulting, LLC be liable for direct, indirect, incidental or consequential damages resulting from You modifying the Software in any manner.

Indemnification/Hold Harmless
ControlWorks, in its sole and absolute discretion may refuse to provide support for the application of the Module in such a manner that We feel has the potential for property damage, or physical injury to any person. Dealer shall indemnify and hold harmless ControlWorks Consulting LLC, its employees, agents, and owners from any and all liability, including direct, indirect, and consequential damages, including but not limited to personal injury, property damage, or lost profits which may result from the operation of a program containing a ControlWorks Consulting, LLC Module or any component thereof.

License Grant
Software authored by ControlWorks remains the property of ControlWorks. ControlWorks grants You the non-exclusive, non-transferable, perpetual license to use the Software authored by ControlWorks as a component of Systems programmed by You. This Software is the intellectual property of ControlWorks Consulting, LLC and is protected by law, including United States and International copyright laws. This Software and the accompanying license may not be transferred, resold, or assigned to other persons, organizations or other Crestron Dealers via any means.

The use of this software indicates acceptance of the terms of this agreement.