



APPLICATION NOTE

4D SYSTEMS

TURNING TECHNOLOGY INTO ART

ViSi TCP Socket Connections

Document Date: 8th October, 2014

Document Revision: 1.0

Description

The coverage of this application note demonstrates TCP socket connections to resolved host names. It shows how to use the socket related commands of the CC3000.



Pre-requisite Application Note:

[ViSi Connecting the CC3000 to an AP](#)

[ViSi Resolving Remote Address](#)

Before getting started, the following are required:

- The target module can also be a Diablo16 display

[uLCD-35DT](#)

[uLCD-70DT](#)

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor.

- [4D Programming Cable](#) or [uUSB-PA5](#)
- [micro-SD \(uSD\)](#) memory card
- [Workshop 4 IDE](#) (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.
- [CC3000 WiFi Shield](#)

Content

Description	2
Content.....	3
Application Overview	4
Setup Procedure	4
Create a New Project.....	5
Design of the Project	5
<i>Layout of the Project</i>	5
The GUI string inputs and output display locations	5
Connection Status and	6
<i>The ViSi-based CC3000 Program</i>	6
The TCP Socket Connection	6
Run the Program	7
Proprietary Information	8
Disclaimer of Warranties & Limitation of Liability.....	8

Application Overview

The CC3000 uses network sockets to communicate through the internet. It uses socket addresses that has a specified IP address and port number. For this project a Transmission Control Protocol (TCP) socket connection is utilized.



Setup Procedure

For instructions on how to launch Workshop 4, how to open a **ViSi** project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Create a New Project

For instructions on how to create a new **ViSi** project, please refer to the section “**Create a New Project**” of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Design of the Project

Layout of the Project

In this project couple of objects were used to provide a simple user interface form for the CC3000. In this project the information related to a host se

The GUI string inputs and output display locations

Result Remote address

Connection Socket #



Connection Status and

These icons show the status of connecting to the network. The progress spinner is used to indicate that the processes under the connection icon is currently running or being executed.



PROGRESS INDICATOR



CONNECTED

The ViSi-based CC3000 Program

In this part of the application note statements that were written in 4DGL will be shown. These statements are used to produce the functionality desired for this project application. Each of the sub-routines has a group of statements that are intended for a designated purpose only.

The TCP Socket Connection

The statements presented here are called upon in the main function later in this document. It is best to present the purpose of each sub-routine for us to be able to easily understand the main program statements.

The starting part of this application project are very similar to the one used in the pre-requisite application notes indicated. Again, in this application project the IP address is being resolved using the statement shown below.

```
wlan(WGETHOST ,website,0,0,0,0,result);
ip2str(result,IP_add);
```

The first line in the statement uses the host name encoded in the space provided for the WEBSITE string. If the address resolving is done the IP address of the host will be saved to the array 'result'.

The content of the array 'result' will then be converted from separate bytes into a string using the IP2STR function.

```
ip2str(result,IP_add);
```

The need for converting the result IP address to string is a must due to socket related command which will be shown later. The function below denotes that the content of 'result' will be

Converted into a string that will be saved in IP_add array. After a successful retrieval of IP address and converting this to a string we now move to the connection preparation. Before any connection is made, make sure that a TCP socket is opened similar to the one below.

```
wlan(WSOCKET_OPEN ,SOCK_STREAM,IPPROTO_TCP,0,0,0,result);
```

The opened socket will be saved on result[0]. If the value of result[0] is not -1 and less than 8, then the process have successfully opened the TCP socket. The next step is to connect the address to the socket. This command simply attempt to connect the IP_add to the socket with a port number of 80.

```
wlan(WSOCKET_CONNECT, IP_add, 80, result[0], 0, 0, 0, result);  
if(result[0] == 1)  
    spinning_progress("connected");  
endif
```

Again, the status of the command WSOCKET_CONNECT will be saved on the array result. If the value is 1 then it successfully connected to the host IP address.

Run the Program

Refer to the section "Connect the WiFi Module to the uLCD-35DT" of the application note

[ViSi Connecting the CC3000 to an AP](#)

for the connection diagram of the WiFi module to the 4D display.

For instructions on how to save a **ViSi** project, how to connect the target display to the PC, how to select the program destination, and how to compile and download a program, please refer to the section "**Run the Program**" of the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.