



ViSi Resolving Remote Address

DOCUMENT DATE: **22th JANUARY 2019**
DOCUMENT REVISION: **1.1**



Description

This application note requires that the reader have already read the brief presentation related to initializing and establishing connection to an access point. This documentation is limited to discussing the addition of a string. The additional string is the identification of a server from the internet. The remote IP address of the string will be resolved in this project.



Pre-requisite Application Note:

[ViSi Connecting the CC300 to an AP](#)

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
ACCESS POINT SSID AND PASSKEY INPUT LOCATIONS	5
CONNECTION ACCESS POINT STATUS ICONS	6
EXECUTE CONNECTION TO ACCESS POINT ICON	6
<i>The ViSi-based CC3000 Program</i>	6
The sub-routines	6
Wifi_connect() subroutine	6
Run the Program	7
Proprietary Information	8
Disclaimer of Warranties & Limitation of Liability	8

Application Overview

This application is focused to demonstrating the use of CC3000 commands to retrieve the IP address of a given host name.

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

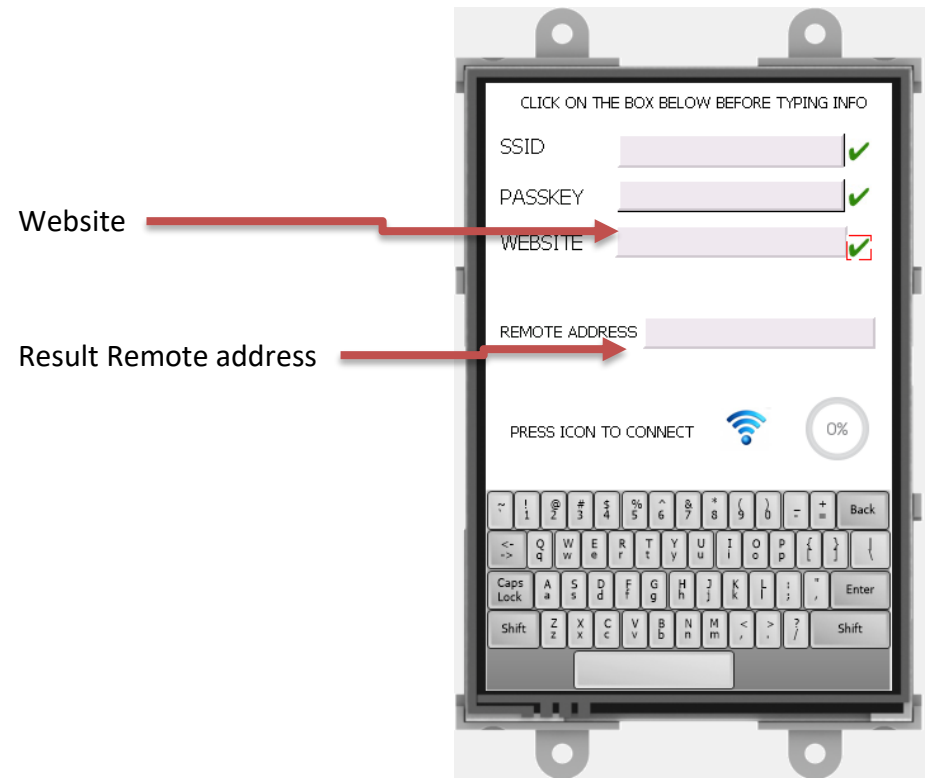
[ViSi Getting Started - First Project for Picaso and Diablo16](#)

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, a space is provided to display the encoded website address. The information given in the space provided for the website will have is IP address resolved in this project.

ACCESS POINT SSID AND PASKEY INPUT LOCATIONS



CONNECTION ACCESS POINT STATUS ICONS

These icons show the status of connecting to the network. The progress percentage display at the internal functions are executed and completed.



NOT CONNECTED



CONNECTED

EXECUTE CONNECTION TO ACCESS POINT ICON

If the access point SSID and PASSKEY have all been entered as represented by the check mark beside the spaces. The next icon to consider is the connect command execute icon.



This icon contains the command that takes the input strings from the SSID and PASSKEY which was entered by the user and sends it to the CC3000 via the SPI port 2 of the Diablo 16.

** NOTE: A child program is run in flash bank 2. The program included in the zip file with this project.

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 sub-routines

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.

Most of the subroutines in this projects are already used and mentioned from the pre-requisite application notes listed at the overview.

Wifi_connect() subroutine

This sub-routine contains the command to transport data to flashbank_2. Notice that there is memory allocation in the statements. The memory set allows data to be saved to the location while a program is run in the other flash bank locations.

The wifi_connect() executes commands from the secondary flash, which is this case is FLASHBANK 2, by passing parameters. To pass the parameters, wlan(...) subroutine is used the first seven parameters are used to identify the command to be used and the data needed by the command. On the other hand the 8th parameter of the command identifies an array where the data will be returned. For example, the command below direct that the website's address be resolved and save the remote IP address to array 'result'. The statement below is the actual command to retrieve the address.

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

```

func wifi_connect()
xfer:= mem_Alloc(512);
mem_Set(xfer, 0,512);

wlan(WBEGIN ,NORMAL,0,0,0,0,0);
wlan(WSP_VERSION ,0,0,0,0,0,0);
wlan(WEVENT_MASK ,unsol_keep_alive || unsol_ping_report || unsol_init|| unsol_DHCP,0,0,0,0,0);
connection(1);

to(IP_add); putstr("000.000.000.000");
to(subnet); putstr("000.000.000.000");
to(gateway);putstr("000.000.000.000");
to(DNS); putstr("000.000.000.000");
wlan(WSET_DHCP ,IP_add,subnet,gateway,DNS,0,0,0);
wlan(WARP_FLUSH ,0,0,0,0,0,0);
wlan(WSCAN_PARAMETER,0000,0,0,0,0,0);
wlan(WSCAN_RESULT ,0,0,0,0,0,0);
connection(2);

wlan(WCONNECTION_POLICY ,NO_AUTO_CONNECT,0,0,0,0,0);
wlan(WSET_TIMER,60,60,6012,0,0,0);
connection(3);
wlan(WCONNECT ,WPA2,ssid,key,0,0,0,result);
connection(4);
wlan(WGETHOST ,website,0,0,0,0,result);
connection(5);
endfunc

```

The commands being executed in this subroutine covers the essential commands for initializing and setting up the CC3000. After initialization has been made, only then connections to access points can be done. If a successful retrieval of IP address is attained the project should display the remote address of the specified host name. Regardless of success of the address retrieval the output will be displayed on the screen using the statements below.

```

wifi_connect();
connection("REMOTE");
print(result[0],".", result[1],".", result[2],".", result[3]);

```

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](#)

After downloading the project, if using a uLCD-35DT, wire the 30-way header on the Diablo display similar to the one below.

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.