



ViSi Reading Text Information from Webpage

DOCUMENT DATE: **21st MAY 2019**
DOCUMENT REVISION: **1.1**



Description

The coverage of this application note demonstrates the way of accessing webpages and saving the content into a file in the micro-SD.



Pre-requisite Application Note:

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

[ViSi Resolving Remote Address](#)

[ViSi TCP Socket Connections](#)

Before getting started, the following are required:

- The target module can also be a Diablo16 display

[gen4-uLCD-24D series](#) [gen4-uLCD-28D series](#) [gen4-uLCD-32D series](#)
[gen4-uLCD-35D series](#) [gen4-uLCD-43D series](#) [gen4-uLCD-50D series](#)
[gen4-uLCD-70D series](#)
[uLCD-35DT](#) [uLCD-43D series](#) [uLCD-70DT](#)

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor. The display module used in this application note is the uLCD-32PTU, which is a Picaso display. This application note is applicable to Diablo16 display modules as well.

- [4D Programming Cable](#) / [uUSB-PA5/uUSB-PA5-II](#) for non-gen4 displays(uLCD-xxx)
- [4D Programming Cable](#) & [gen4-PA](#) / [gen4-IB](#) / [4D-UPA](#) for gen4 displays (gen4-uLCD-xxx)
- [micro-SD \(μSD\)](#) 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.

Content

Description 2

Content 3

Application Overview 3

Setup Procedure 4

Create a New Project 4

Design of the Project 5

Layout of the Project 5

 Connection Status 5

The ViSi-based CC3000 Program 5

 The HTTPS /1.1 related commands 5

Run the Program 8

ViSi Getting Started - First Project for Picaso and Diablo16 8

Proprietary Information 9

Disclaimer of Warranties & Limitation of Liability 9

Application Overview

In this application project, a set of text information is being accessed from the internet and then saved to a file in the micro-SD. The target webpage content type used in this application is limited to a text-based webpage content and information. This application note is limited to the related basics of the HTTP 1.1 as used in the application project. It is not intended to be a reference material for the aforementioned topic.



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

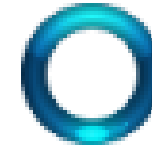
Layout of the Project

In this project couple of objects were used to provide a simple user interface form for the CC3000. In this project the information contained in one webpage is being downloaded using the CC3000 and then saved to a micro-SD.



Connection Status

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 HTTPS /1.1 related commands

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems.[1]HTTP is the foundation of data communication for the World Wide Web.

After being able to connect to a Wi-Fi network and to successfully connect a remote host to a socket, we will now try to take data from the internet to the display device. For demonstration purposes, we will only make a copy of the information downloaded to a file in the micro-SD.

Let this us point to the important commands used in the pre-requisite application notes. Under the sub-routine wifi_connect(), commands are used to connect the CC3000 to a certain WI-FI network. The command is given by the statement below.

```
wlan(WCONNECT      ,WPA2,ssid,key,0,0,0,result);
```

After connecting to the network, the next step is to retrieve the remote host address. Following a successful query of the remote host address, we then open a communication socket according to the communication protocol that we are to use either TCP or UDP data communication is supported by the CC3000. For this application note a TCP data communication is being established with the remote host.

```
wlan(WSOCKET_OPEN   ,SOCK_STREAM,IPPROTO_TCP,0,0,0,0,result);
if(result[0] != -1)
    spinning_progress(1);
    socket[0]:=result[0];
endif

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

If a socket is successfully opened using the WSOCKET_OPEN command the next step to take is connecting the resulting IP address of the remote host. Connecting to a remote host will require the use of the WSOCKET_CONNECT command.

Having been able to establish connection to the remote host, we can now proceed with requesting information from the remote host. Above are the statements used to request the content of a webpage located in the specified website host.

```
to(data_);putstr("GET /c3000/c3000.txt HTTPS/1.1\r\n");
wlan(WSEND      ,0,0, data_,0,0,0,result);

to(data_);putstr("Host: www.4dsystems.com.au\r\n");
wlan(WSEND      ,0,0, data_,0,0,0,result);

to(data_);putstr("\r\n");
wlan(WSEND      ,0,0, data_,0,0,0,result);
```

Let us try to interpret the meaning of the statements shown above. The HTTP 1.1 protocol involves the command GET. The “GET” command is a method that can be used to retrieve whatever information in the form of an entity that is identified by the Request-URL. In this application project we are requesting the content of the URL:

www.4dsystems.com.au/c3000/c3000.txt

To do we need to separate the page content URL with the Host address. And send the informations separately. The first one to be sent will be the URL of the desired page and preceeded by the GET method of HTTP 1.1.

```
to(data_);putstr("GET /c3000/c3000.txt HTTPS/1.1\r\n");
wlan(WSEND      ,0,0, data_,0,0,0,result);
```

After the ‘GET’ related command has been sent, this is followed by the ‘Host’ address information.

```
to(data_);putstr("Host: www.4dsystems.com.au\r\n");
wlan(WSEND ,0,0, data_,0,0,result);

to(data_);putstr("\r\n");
wlan(WSEND ,0,0, data_,0,0,result);
```

To better understand this the user can refer to this additional reading –

[Hypertext Transfer Protocol -- HTTP/1.1](#)

If the GET and Host statements have been sent, the next is to add a last statement that contains a '\r\n' statement. Now that we have sent all the statements needed to request for the resources of a webpage, let us now move to reading the data from the CC3000 and saving it to a file in a micro-SD.

```
wlan(WREAD_DATA ,0,1024,0,0,0,data_);

if(file_Exists("SAMP.txt")) file_Erase("SAMP.txt");
file:=file_Open("SAMP.txt", 'a');
```

Next is read the available data on the CC3000. To read the available information from the CC3000 we use the WREAD_DATA command. This command checks the CC3000 for availability of data and then followed by a continuous read according to the length specified.

```
wlan(WREAD_DATA ,0,1024,0,0,0,data_);

if(file_Exists("SAMP.txt")) file_Erase("SAMP.txt");
file:=file_Open("SAMP.txt", 'a');
|
for(n:=1; n< 1024; n++)
    file_Putc(data_[n], file);
    //print([CHR] data_[n]);
next
file_Close(file);

gfx_Panel(PANEL_RAISED, 88, 209, 143, 28, WHITE);
img_Show(hndl,iStaticText6);
spinning_progress("complete");
file_Unmount();
```

The statements shown above puts the read characters to an array named data_ and then saves them one by one on the text file previously created. The file_Close(file) statement saves and closes the created text file.

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.