



Controlling a WF32 from a Computer

by [joshwoldstad](#) on June 24, 2015

Table of Contents

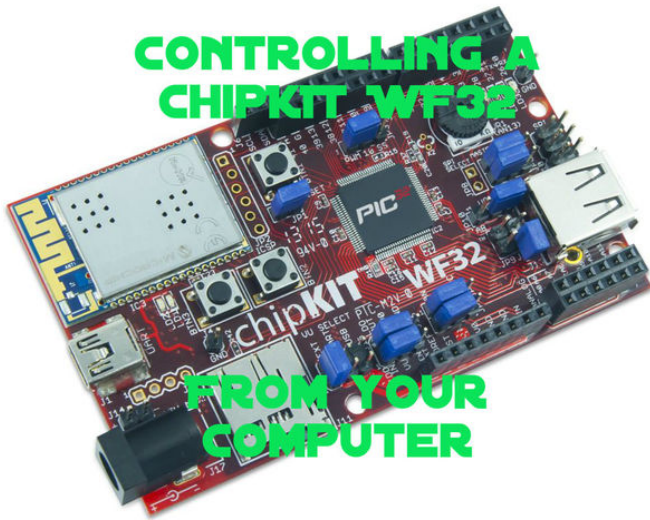
Controlling a WF32 from a Computer	1
Intro: Controlling a WF32 from a Computer	2
Step 1: Downloading the Test Version of MPIDE	2
Step 2: Download deWebIOServer	3
File Downloads	3
Step 3: Moving Content into your SD Card.	3
Step 4: Opening the MPIDE File	3
Step 5: Adding your Wifi Credentials	4
Step 6: Plug your SD Card into the WF32	4
Step 7: Changing your Board Settings	5
Step 8: Selecting your Serial Port	5
Step 9: Uploading the Code	6
Step 10: Finding the IP Address of the WF32	6
Step 11: Playing around with the Pins	7
Related Instructables	7
Advertisements	7
Comments	7

Intro: Controlling a WF32 from a Computer

The chipKIT WF32 is Wifi-capable and contains an SD card device. That means we can load host a server on the WF32 and access it from other computers! This Instructable is about setting up a webserver that allows you to access the WF32 pins from a web page.

You will need:

- A chipKIT WF32
- Micro SD Card
- A Computer with internet access
- Access to a wifi network.



Step 1: Downloading the Test Version of MPIDE

In order for us to properly compile everything, we need to use a test version of MPIDE.

Here is the download [link](#) to the recommended test version.

Note: MPIDE test builds may interact with stable builds of MPIDE, so be sure to back up anything crucial.

Extract the files to your desired location. You can open MPIDE by going into the file and double-click on the **MPIDE.exe** file.



Step 2: Download deWebIOServer

Download this file and extract it. Take note of where you are extracting the folder, you'll need to access it soon.

The file should look like the picture above after it is done downloading/extracting.

Content	6/25/2015 2:26 PM	File folder	
18020_EWN_LabManual.pdf	6/8/2015 7:07 PM	Adobe Acrobat D...	5,790 KB
deWebIOServer.pde	6/24/2015 4:52 PM	PDE File	14 KB
HTMLGetPins.cpp	6/24/2015 5:40 PM	CPP File	34 KB
HTTPServerConfig.h	6/24/2015 1:58 PM	H File	12 KB
IOConfig.h	6/24/2015 5:30 PM	H File	6 KB

File Downloads



deWebIOServer.zip (6 MB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'deWebIOServer.zip']

Step 3: Moving Content into your SD Card.

Now we're going to move all the contents of the Content folder onto the SD card.

1. Plug your SD Card into the computer.
2. Open up the Content folder
3. Copy everything in the folder. Note: You have to copy all the files inside Content, not Content itself
4. Paste everything into the SD Card and safely eject.

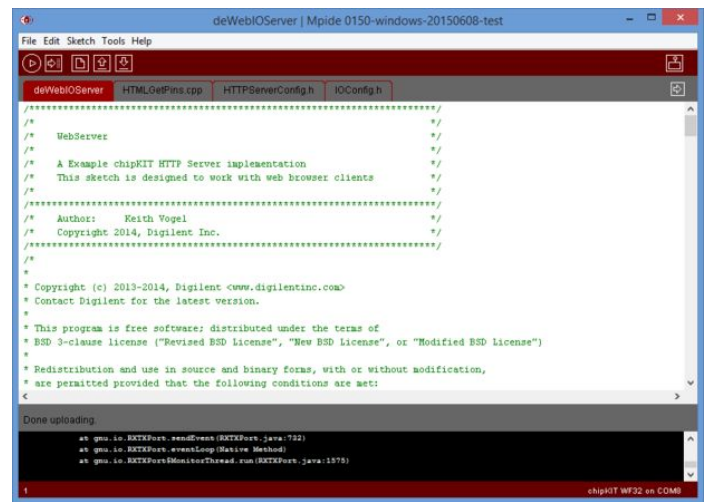
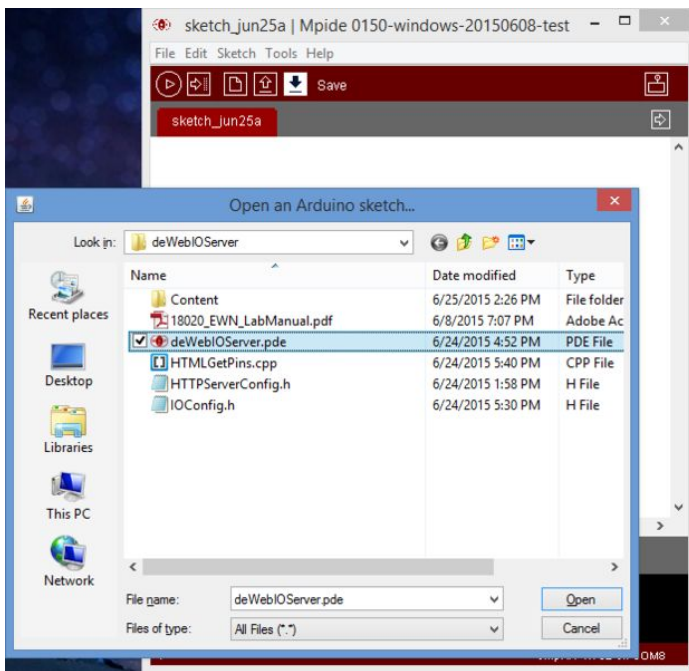
Everything in the Content folder has to be in the Root file in the SD Card. (It just has to look like the above picture when you open up the SD Card.)

aboutck.htm	6/8/2015 7:07 PM	Chrome HTML Do...	4 KB
aboutdm.htm	6/8/2015 7:07 PM	Chrome HTML Do...	3 KB
aboutmax.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
aboutmx3.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
aboutmx4.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
aboutmx7.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
aboutuc.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
aboutuno.htm	6/8/2015 7:07 PM	Chrome HTML Do...	2 KB
Digilent.png	6/8/2015 7:07 PM	PNG File	20 KB
DPSK.PNG	6/8/2015 7:07 PM	PNG File	4 KB
favicon.ico	6/8/2015 7:07 PM	ICO File	2 KB
HomePage.htm	6/24/2015 4:52 PM	Chrome HTML Do...	4 KB
HTTPrv.zip	6/8/2015 7:07 PM	Compressed (zipp...	1 KB
Max32.jpg	6/8/2015 7:07 PM	JPG File	78 KB
MX3cK.png	6/8/2015 7:07 PM	PNG File	184 KB
MX4cK.png	6/8/2015 7:07 PM	PNG File	185 KB
MX7cK.png	6/8/2015 7:07 PM	PNG File	191 KB
NetShld.jpg	6/8/2015 7:07 PM	JPG File	97 KB
PSK.PNG	6/8/2015 7:07 PM	PNG File	3 KB
SerMon.PNG	6/8/2015 7:07 PM	PNG File	18 KB
SrvSetup.htm	6/8/2015 7:07 PM	Chrome HTML Do...	17 KB
uC32.jpg	6/8/2015 7:07 PM	JPG File	212 KB
Uno32.jpg	6/8/2015 7:07 PM	JPG File	84 KB
WiFire.png	6/8/2015 7:07 PM	PNG File	327 KB

Step 4: Opening the MPIDE File

Now it's time to open the MPIDE file.

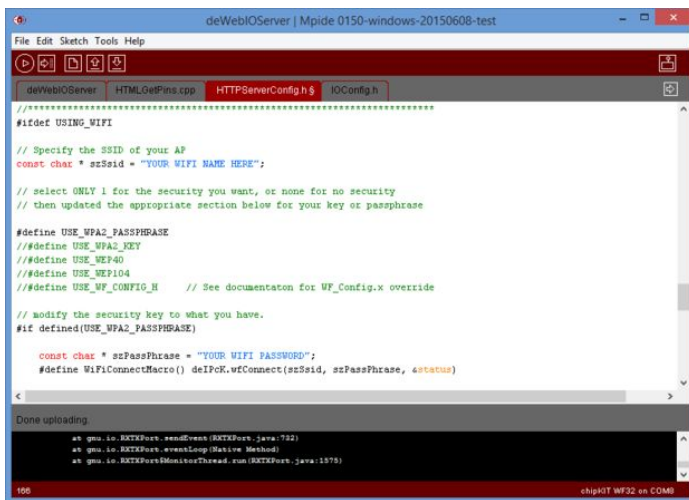
1. In the test version of MPIDE we downloaded, enter the command **CTRL + o**.
2. Open deWebIOServer.pde
3. A new window of MPIDE should pop up, and there will be 4 tabs opened.



Step 5: Adding your Wifi Credentials

In order for the WF32 to "work" you need to connect it to your network.

1. Go to the HTTPServerConfig.h file.
2. Enter the command **CTRL + f** (search), and enter szSsid, this will take us to the area where we want to enter our Credentials
3. In the string "YOUR WIFI NAME HERE", enter the wifi you use.
4. In the string "YOUR WIFI PASSWORD", enter your password.
5. Save using **CTRL + s**.

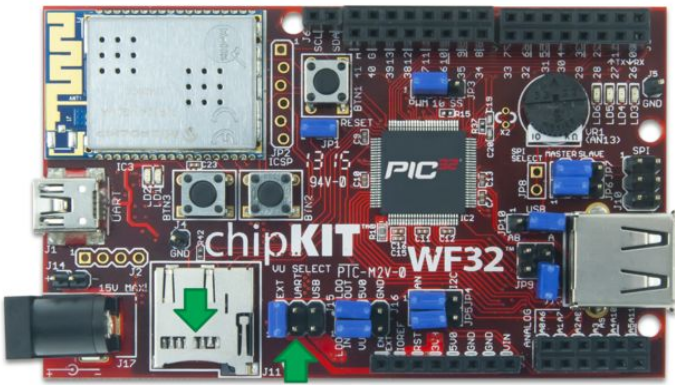


Step 6: Plug your SD Card into the WF32

Check out the green arrow to see where the SD Card should go.

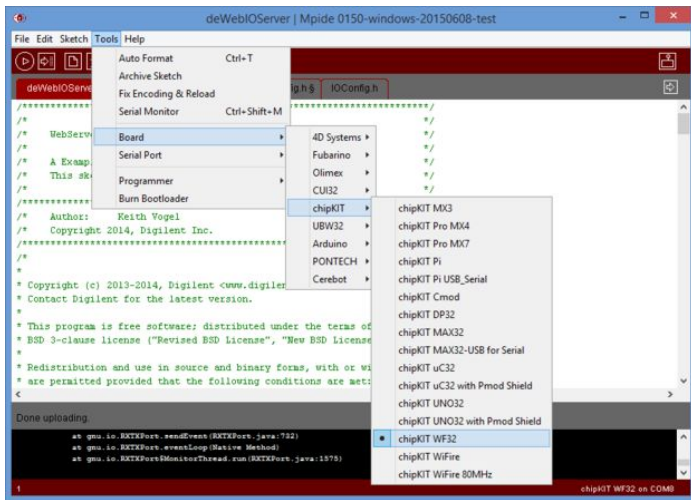
After the SD Card is plugged into the WF32, connect the WF32 to your Computer. Use a Micro USB Cable to connect the two.

Be sure to have the VU Select jumper at the UART pins, otherwise you won't be able to encode everything.



Step 7: Changing your Board Settings

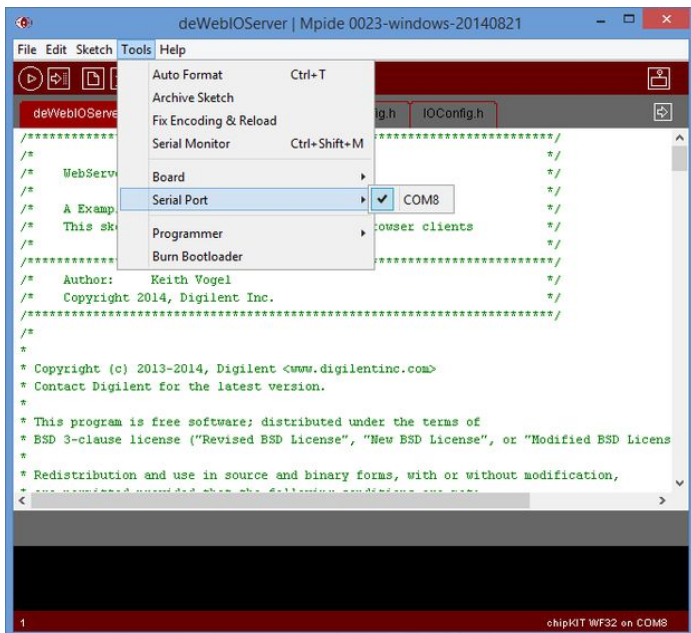
In order for everything to upload correctly, we need to make sure the "board" in MPIDE is set to WF32. In this case, a picture is worth a thousand words. Check out the picture to see where to change the board to WF32.



Step 8: Selecting your Serial Port

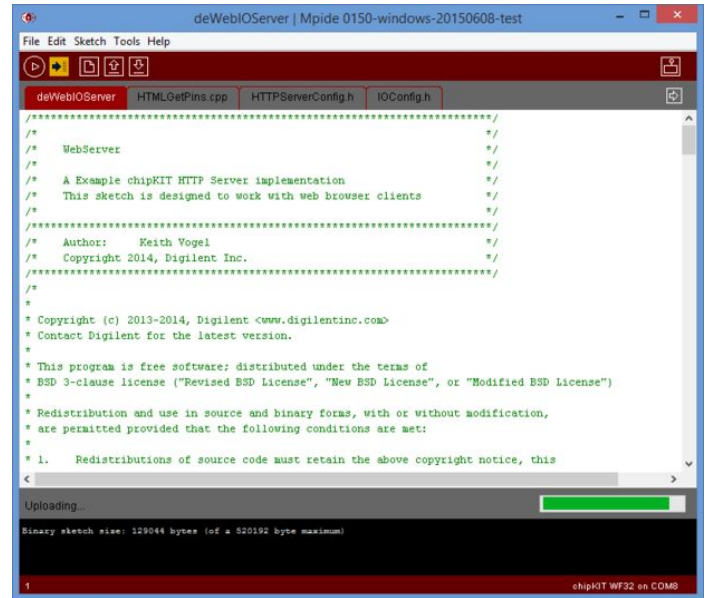
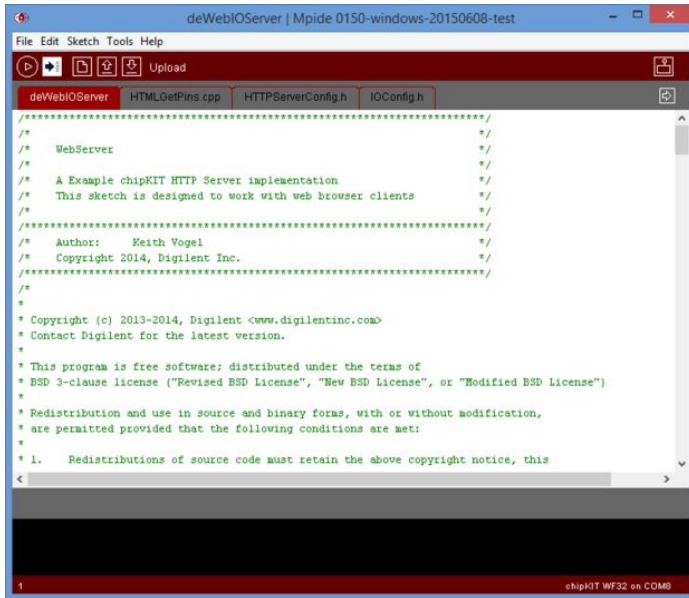
Before we can view our Serial Monitor, we need to select our Serial Port. Just like choosing the board, go to Tools, but select Serial Port, and then pick whatever COM port your WF32 is on.

Now you'll be able to open the Serial Monitor. (You can do this now or in a few steps from now)



Step 9: Uploading the Code

Now that you've got your Wifi credentials and the MPIDE is set to WF32, it's time to upload the code. Click on the square upload button in the top left corner and MPIDE, and wait for everything to upload!

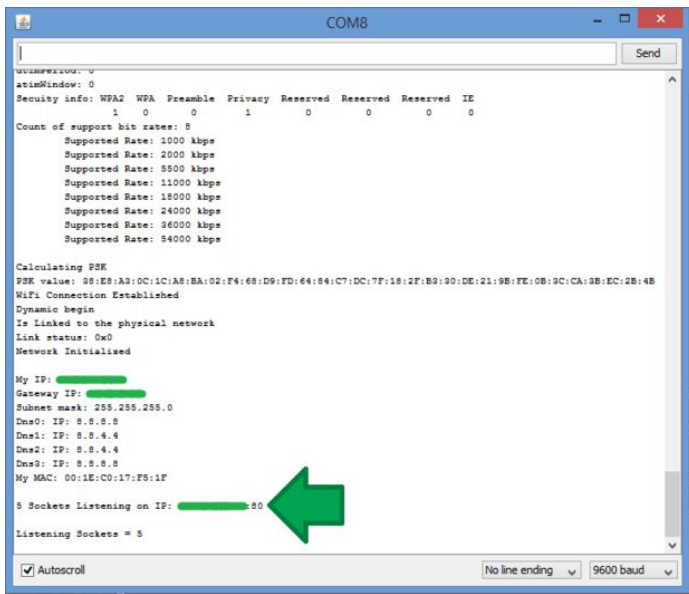


Step 10: Finding the IP Address of the WF32

After the code has finished uploading, open the Serial Monitor (with the button in the top right corner).

Initialization will happen, when it gets to "Listening Sockets = #" line, you can find the IP address of the WF32.

You can check it out in the picture, it will be something along the lines of 12.34.5.678:80 (It won't be this, but something like it). Enter the underlined portion into your browser. Be sure that the computer your using is on the same network as the WF32.



Step 11: Playing around with the Pins

Once you've entered the IP address, it will take you to a landing page about the server.

You can click on the Read and Modify Board Pins link to get to the Pin Page.

The easiest example is changing Pin 13 from the **Tristate** button to **Digital Output High**, then hit the **Refresh** button. This will change one of the LED pins from OFF to ON.

Of course this is just one example of PIN options you can do. You is access to all non-essential pins!

You can check out the Pin-out tables [here](#).

DIGILENT
BEYOND THEORY

DEIPeK HTTP Example Server
by Keith Vogel

The HTTP server example abstracts the complexities of creating HTTP Web server hosting pages that can be written in almost any HTML editor. Once created, just copy your pages onto an SD card and plug it into the SD card reader on your chipKIT board. Restart the server. All links specified in the pages should be relative to the current page - or relative to the root of the SD file system. The default page is called `newpage.htm`, and this page must exist at the root of the SD file system. All files on the SD file system must be limited to the [3.1 file naming convention](#); appropriate extensions should be used on your files. The SD file system can contain HTML pages, JPEGs, GIFs, Icons, TXT, MP3s, WAVs, PNGs, and XMLs for download to the browser; however, only the two three-letter extensions may be used for each file type. The content type specified to the requesting browser is determined by the two-letter file extension.

In addition to pages stored on the SD card, it is possible to add active pages that are dynamically created by writing a concise function and specifying the accessing URL to the server. These dynamic pages can respond to browser HTTP GETs, POSTs, or PUTs and can dynamically interact with the resources on the chipKIT board. One example is to create a dynamic page that talks to a camera that is connected to the board, and then have it take a picture and compose a JPEG picture. In designing your HTML pages you can use URLs that reference local SD files, dynamically created pages, or absolute URLs to other sites embedded in your HTML pages. For example, if you want to use some locally hosted pages on the SD card, check out the page about [The chipKIT MPIDE System](#). Or, you can go to another site like the Digilent products page at [Digilent Inc.](#)

For an example of how to use a `Fetch` to read and modify the GPIO pins on the board, check out [Read and Modify Board Pins](#).

The HTTP server uses a highly cooperative multitasking programming model where multiple connections and pages can be processed concurrently. To ensure this works reliably, any code written for the server's `loop()` function or component pages should be written as a state machine, with each state only doing a small portion of work. The HTTP server loops repeatedly, calling your code as well as the HTTP server code. As part of the HTTP server code, it listens for connections. When a connection is made, the appropriate `connect()` function is called based on the URL. If no URL matches a connect page, the default `connect()` function is called. Typically, the SD file system connect function is used as the default `connect()` function, as this will look on the SD file system for a matching file. If no file exists, the SD `connect()` function will return an "HTTP File Not Found (404)" error to the browser.

For specific information on how to set up this example server, go to [Getting Started](#).

GPIO Status

[Refresh](#)

Pin 0: DEDICATED

Pin 1: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 2: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 3: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 4: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 5: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 6: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 7: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 8: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 9: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 10: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 11: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 12: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 13: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 14: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 15: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 16: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 17: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 18: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 19: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

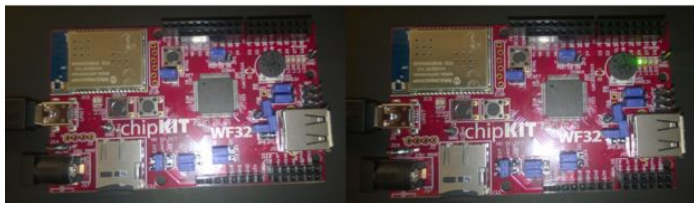
Pin 20: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 21: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Pin 22: * Tristate | Digital Input | Digital Output LOW | Digital Output HIGH | Pin Value

Before

After



Related Instructables



Getting Started With the WF32!

by JayWeeks



Getting Started with the ChipKIT WF32 (LabVIEW) by Sudharsan Sukumar



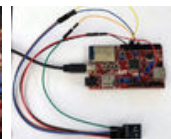
Health and Security Cloud System - Digilent Design Contest 2015 by burlacu.eusebiu



Watt-a-save by AndreiAnghel



Display Weather and Location Using chipKIT WF32 and LabVIEW by Sudharsan Sukumar



SPI in LabVIEW Using MakerHub LINX, MmodALS, and chipKIT WF32 by Sudharsan Sukumar

Comments