I2C WebLogger

1. Introduction

The I2C-WebLogger is a stand alone I2C Network device, that allows reading *any* I2C sensor or device. It gives the functionality of the BL233 and I2C2PC for network use. Uniquely in the world of IoT, there is *no lock-in* to our servers or infrastructure.

- Wifi LAN, Wifi AccessPoint, and Ethernet Interfaces
- Internal Webserver and log data storage for direct operation over LAN
- Datalogging and display of formatted data from I2C sensors over internet.
- Stand-alone operation possible without internet connection, or even LAN.
- Remote webcam photos
- No compilers or software needed to read and display I2C sensors over the Internet or LAN.
- Three I2C, SPI, 1 Wire Ports
- Remote serial-over-ip operation e.g. from Labview, Matlab

The Modules hardware pcb can also be used with a wide range of other Linux devices or serial modules (Raspberry Pi, XBee, Bluetooth BLE).

Contents

1.1. Key Advantages

Compared to other IoT devices commonly available:

- No Lock-In: Everyone else is trying to lock you in to their services. WebI2C will work without any servers, and you can run your own servers if you need.
- *Any* type of sensor can be used. If it is I2C, SPI, or 1Wire, you can use it. You do not depend on us to support new sensors
- Learn once: Our ASCII Text commands and data are the same for all interfaces today and tomorrow: PC/Linux, USB, RS232, Bluetooth, Fibre, Wifi, Ethernet, Sigfox, GSM, GPRS
- There is NO programming of the device, and no compilers, or development enviroments, to customise it for *any* I2C sensors. Customisation is done entirely remotely, via the webpage, after installation.
- Operation is possible *without* internet access.
  - Remote locations without internet, or local access when the internet fails
  - Secure systems where connection to the public internet is undesireable. (IoT devices will *never* be secure, it is delusional to think so.)
  - Reliability: Operation must continue during cellular/internet/power outages
- Long Term Security: History has already shown that big (e.g. google@Home) and small service providers are turning services off after 3 years or less, with no warning and no recourse. No prudent industrial or infrastructure monitor should be dependent on them.
- 15 years and counting: We have current industrial customers who are using and buying our I2C devices for more than a decade.

2. Overview

The Weblogger hardware is based on the BL233 chip with a Linux router, and has very similar I2C capability as ourI2C2PC.

- I2C, SPI, 1Wire
- 3 Ports (2x CMOS, 1x TTL)
- 5V and 3V capable.

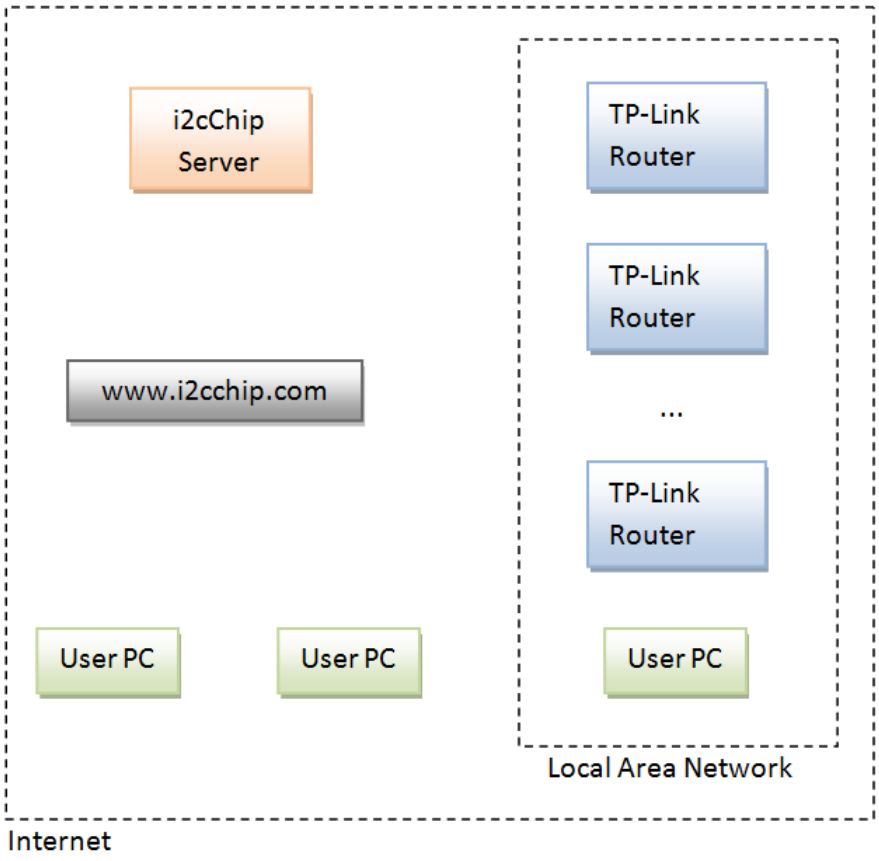
- Internal Power Switch to control router power supply from I2C side - Allows low power intermittent operation eg solar power, and true independent hardware watchdog

It allows four modes of use:

- Serial-over-IP: Telnet from a computer to the BL233 chip, and control over the LAN, as if it was a local serial device
- Local Logging: WebI2C logs sensors to internal webserver. Access data files over LAN, even without internet connection
- Stand-Alone: Where the is not even a lan, use built-in Wifi Access point
- Remote Internet: Data can be read remotely via internet

The Linux system is completly based on scripting,w ith no special modules, and can be used with any linux system, into the future. There is no compiled binary lock-in.

Below is an overview diagram of what an I2C Web Logger system looks like:



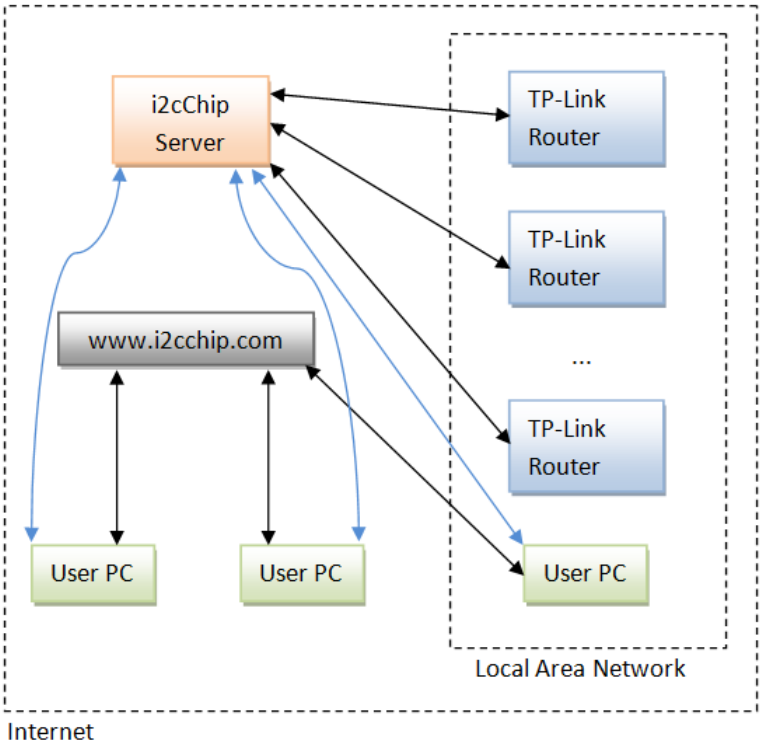
The router supported is TP-Link TL-WR703N. Other linux devices can easily work, as the whole software uses standard Linux commandline tools. Note that the i2cChip server is a separate server from the server hosting www.i2cchip.com.

There are two ways the user can communicate with devices connected to the network:

- Over the internet getting its information from the I2CCHIP server
- Over the Local Area Network

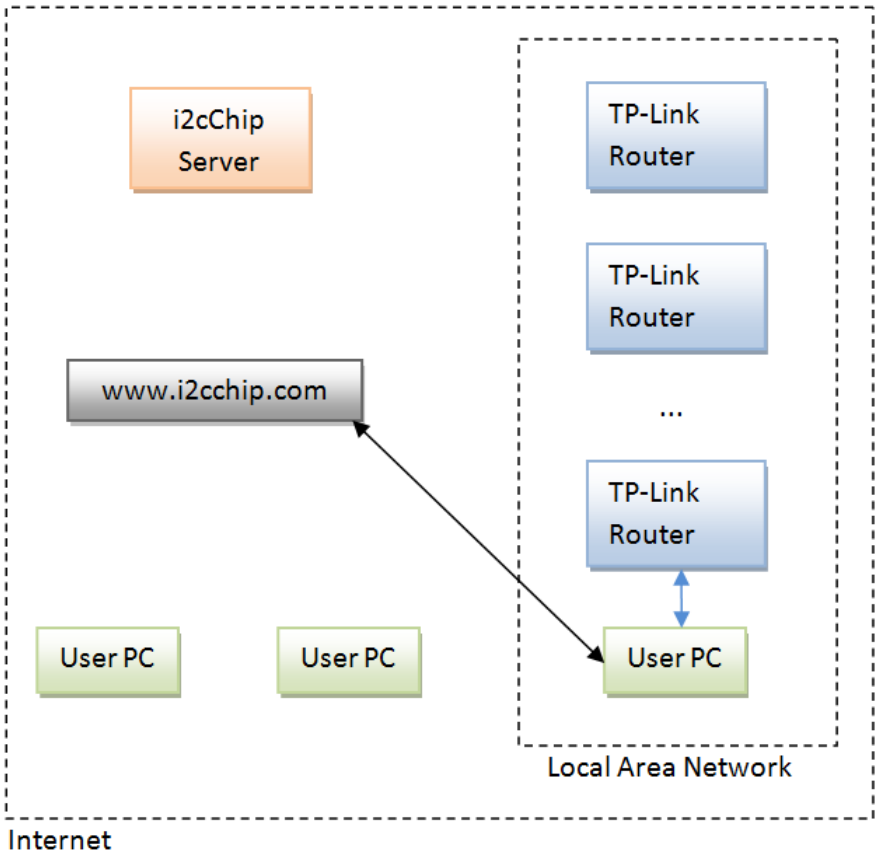
The next two sections will explain these two options in more detail.

2.1. Access device over the internet



Each time the device gets a sample it will be uploaded to the i2cChip server. It will also upload status information regularly. For a user to access the information, go to http://www.i2cchip.com/webDataLogger in your favorite browser. A full explanation of how to use the website will be given later in this document. The javascipt that executes in the user's browser connects to the i2cChip server (shown as blue arrows in the image above). The user will be able to see the samples as they come in, edit code for formatting of the sampled information and even download files from the device. None of the scripts running on the device can be edited through the internet for security reasons. All changes will be local to the browser.

2.2. Access device over the local area network

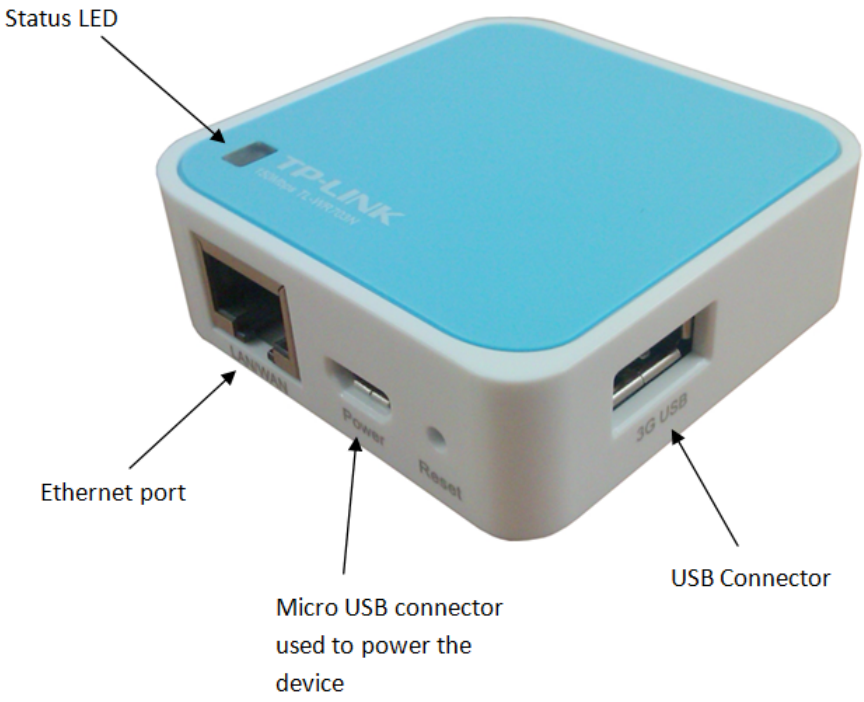


When the user is on the same LAN as the devices then he can select the “Local LAN” option which allows him to directly talk to the devices. The user still accesses the device on http://www.i2cchip.com/webDataLogger but the javascript communicates with a device on its local IP address (shown by the blue arrow). This is much faster than the internet and the user has the capability to edit the scripts on the device. Any device on the same LAN as the user can be accessed this way.

3. Hardware

3.1. TP-Link TL-WR703N

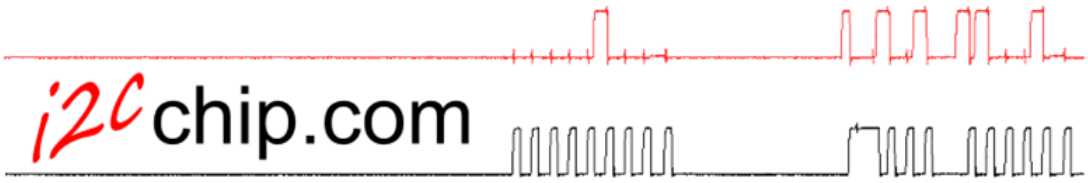
3.1.1 Hardware



The device is powered from a Micro USB connector. It cannot be powered from the Host USB connector. The Ethernet connector is used to connect to your LAN. The device also has wifi built in. When something goes wrong with the wifi setup then the user can be locked out of the device. The Ethernet port is then used to change the settings and get access to the device again. The status LED flashes while the device is booting and stays on when running. The LED can be flashed from the website. This is usefull when a user has multiple devices and wants to check which one he is actually communicating with.

3.1.2 Configuration

When the device is accessed locally using it's IP address, the following screen is displayed:



I2Cchip Web Data Logger

Links for device:

View logger data on www.i2cchip.com (choose the option connected to the internet):
Ethernet: www.i2cchip.com/webDataLogger (IP address: 192.168.1.13)
Wifi: www.i2cchip.com/webDataLogger (IP address: 192.168.2.1) ← 1

Device is currently configured as: Ethernet DHCP Client, Wifi Access Point Fixed IP

Configure the device as a wifi access point and to connect to the rest of your network through ethernet:
Wifi access point ← 2

Configure the device to connect to an existing wifi network and to have a fixed IP address on the ethernet used for configuration:
Wifi client ← 3

View logged data:
All JSON data
All raw data ← 4
Last JSON data sample
Last raw data sample

View Format functions:
Format function
Development format function ← 5

View user javascript:
User javascript ← 6

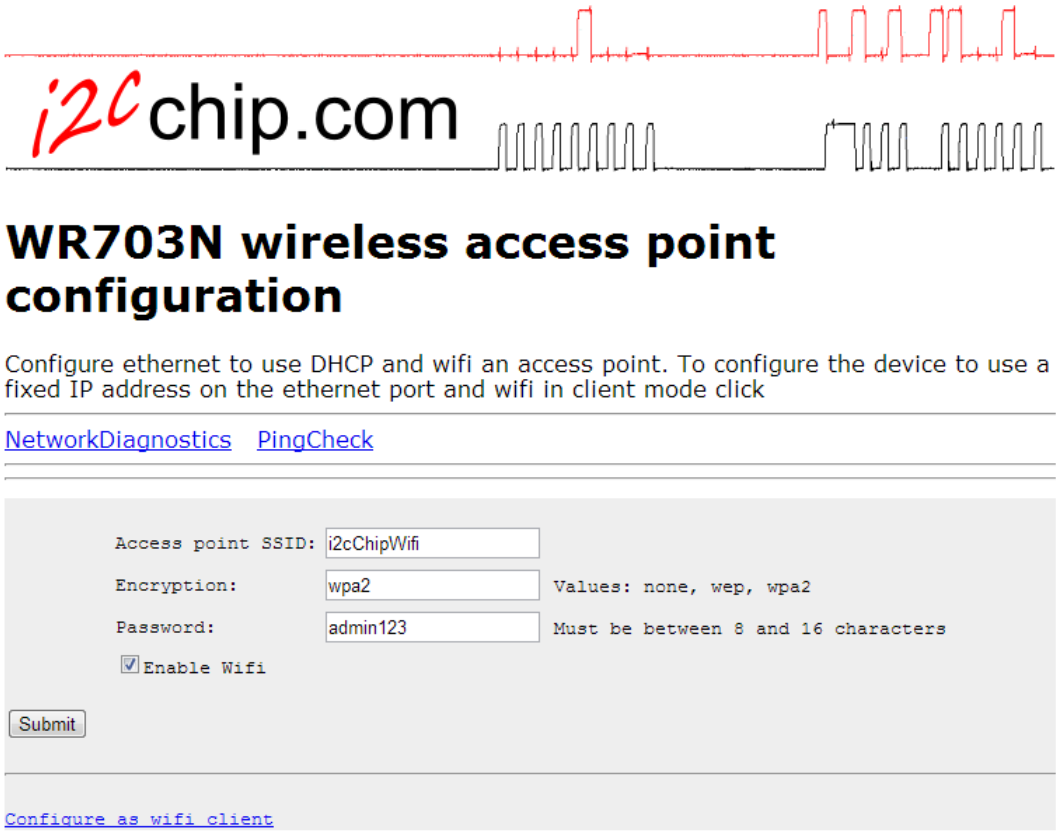
View shell scripts:
Shell script
Config script ← 7

1. Clicking this link redirects the user to the i2cChip website. The URL contains the channel information of the device, so the device will be correctly configured when the page loads. This is the easiest way to start using the device.
2. Wifi access point is used to configure the wifi as an access point and the Ethernet will use DHCP so it can be connected to the user's LAN. This is the default way the device is configured when shipped. This will be explained in more detail later in this document.
3. Wifi client configuration is used to configure the wifi as a client and the Ethernet as a fixed IP address. In this case the Ethernet is only used if something goes wrong in wifi configuration and then the user is locked out of the device. Using the Ethernet the user can get control back of the device. Wifi client configuration will be explained in more detail later in this document.
4. This section allows the user to view the captured data. The data can either be downloaded in JSON format or as raw comma delimited data. The user also has the choice to download all the captured data or only the last captured sample.
5. The format function is used on the website to format the captured data when it is displayed in the formatted data tab. This is explained in more detail later in this document in the website section. These links allow the user to view the format functions stored on the device.

6. The user javascript is one level up from the format function. The user javascript contains user defined functions that can be called from the format function. Please note that the format function contains raw code and the user javascript contains functions. More about this in the website section later in this document.
7. The scripts are executed on the router. This is responsible for sending and receiving data to the devices connected to the router and for uploading and downloading data from the web. See the scripts section below for more details.

3.1.3 Wifi access point setup

Below is an image of what the page looks like for configuring the device as an access point. To have internet access the ethernet port must be connected to a LAN using DHCP. The access point SSID is the name that will show up when you scan for networks on your PC or tablet to connect to. To implement these changes, click on Submit and then click on "Restart network with newly saved settings" on the next page.



3.1.4 Wifi client setup

Select this option to configure the device to connect to an existing wifi network. The SSID is the name of the wifi network you want to connect to. At the bottom the user can enter an IP address that will be associated with the ethernet port. In this scenario the ethernet port is used to access the device if

something went wrong with the wifi setup, for example if the password was incorrect. Configure your PC ethernet port to be on the same subnet as the device, for example 192.168.2.2 and then you should be able to connect to the device and fix the problem. It is very important that the ethernet IP address is on a different subnet as the wifi network. If you are unsure what the range of IP addresses are that your wifi network is using then log into your wireless modem and look under the DHCP section what IP addresses it is assigning to devices.

3.1.5 Scripts

4. Starting Out with WebLogger Starter Kit

Your kit consists of the I2C Starter Kit, WebLogger, and USB Hub & power supply.

- compatible browser: Chrome, Firefox, Safari, (Opera works via internet, not on LAN). Unfortunately IE is not supported

- Connect the 8Bit I/O to I2C2PC port1, and I2C2PC via hub to router USB. (All USB devices must be connected before power on / reboot to be detected.)
- Plug into Ethernet LAN. (Your device is shipped ready to go as an Ethernet DHCP Client)
- Go to the WebDataLogger

Your device should now automatically connect to the internet and begin logging.

4.1. Configuring the Network Interface

Ethernet LAN (DHCP Client)

Plug the unit into an ethernet cable connected to your LAN. It will get an IP address by DHCP, and automatically connect to the internet.

Now you can either scan the LAN to find the device http://www.i2cchip.com/webDataLogger/scanDevices.html, or use the ChannelCode given on the lid.

The device does have internet access.

If you use this mode, the device can be simply plugged into the Ethernet LAN anywhere, and it will just work, without any setting up.

Wifi Access Point

Connect to *I2CchipWifi*. Password *admin123*

Default IP is 192.168.2.1

You are now connected by Wifi with the device as an access point. If you connected the device's ethernet to your LAN then you will have internet access through the access point.

If you want to use as a WiFi Client, then you can now change the settings http://192.168.2.1/cgi-bin/wifi_client

Otherwise, change the default password, or disable the WiFi.

4.2. Wifi Client

If you have a Wifi network, connected to the internet, then you want to connect as a Wifi Client. This will allow the device to connect to the internet via WiFi

On the setup page, select WiFi Client, and enter your network details.

The *Ethernet IP Address* is needed if WiFi setup does not suceed (eg wrong password etc). See below.

4.3. Ethernet Fixed IP Address

Unplug your laptop from the LAN ethernet cable. Plug an ethernet cable directly between your laptop and the device. Go to the fixed address, default is 192.168.2.1.
This is *not* in the usual 192.168.1.xxx range. This is to ensure that it should not conflict with the existing LAN addresses. Do not have both the WiFi and Ethernet connected, if the IP addresses are in the same subnet.

The device does not have internet access through the fixed ethernet, it is standalone.

When you plug your laptop into the device ethernet port, you will be able to access the device, but you will lose your laptops internet connection.

5. Setting up Your Device

Setting up your device to do something new and useful, requires that you configure three things

- Network Settings
- I2C Command Strings: Initialise and scan the sensors
- Format Function: Tiny javascript that converts and scales the hex sensor readings, into meaningful numbers to display.

This will only be 4 or 5 lines: ***Too Easy!***

You must be connected locally on the LAN to save changes to the config files, scripts etc to the router.

5.1. I2C Command Strings

There are two strings in the Config Script. The *InitString*

5.2. Friendly Name, Location etc

Change these in the Config Script

friendlyName: Something you can remember.

jsonLocation: So someone else can find it when you have a new job! Perhaps an address, or “On top of tool cabinet in Room 1237 Bldg 345”

jsonGPSCoordinates: Enter decimal coordinates eg “39.211374,-82.978277”

Getting GPS Coodinates from Google Maps

In Google Maps, simply right-click on your selected spot on the map, to call up a menu. Select ”What’s here” from the menu, and a green arrow will appear on the map on your spot. Click on the green arrow to see the numeric latitude and longitude for the location. Now copy and paste these coordinates from the search box, eg “39.211374,-82.978277”

Google Maps URLs

6. USB Devices

When using USB devices use the Hub provided.

If WiFi is disabled, the hub is not required and USB devices may be connected directly to the device.

6.1. WebCam

A webcam can be connected to take still pictures. Pictures can be taken automatically at intervals set in the Config, or when desired from the webLogger. The last 10 photos are kept on the internet server.

7. Ser2Net Serial Port Redirection

The device has Ser2Net. Ser2Net enables you to make a serial connection over the LAN.

This allows you to:

- Test and debug the I2C commands and return values using Realterm on your pc
- Use a serial port program on your PC to remotely control the I2C over the LAN.

Realterm can connect to the device by setting the Port on the Port tab to the Telnet IP value below:

I2C	Port	IP	
Internal I2C adaptor	ttyATH0	192.168.2.1:3008	telnet
External USB I2C2PC	ttyUSB0	192.168.2.1:3009	telnet
Internal I2C adaptor	ttyATH0	192.168.2.1:2008	raw
External USB I2C2PC	ttyUSB0	192.168.2.1:2009	raw

When Ser2Net is enabled, then data logging via the WebDatalogger system is disabled.

Ser2Net ports are configured in /etc/ser2net.conf. You can change the timeout, handshaking, default baud rate etc.

Realterm does not have a virtual comport on the PC, it can directly talk to telnet ports. A free windows Virtual comport redirector is HW VSP3 Virtual Serial Port.

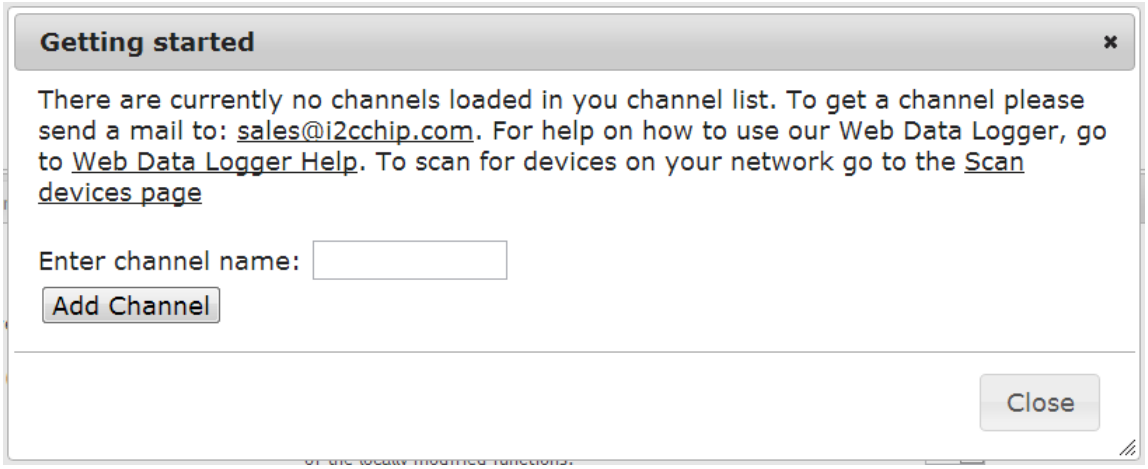
To use Ser2Net from beyond the LAN you need to change your internet firewall/gateway/adsl modem settings. If you are doing this you should setup the comms to tunnel through SSH.

8. Using the website

8.1. Getting started

When you first open the data logger website at http://www.i2cchip.com/webDataLogger/index.html and there are no channel information then you will be presented with a Getting Started dialog box where you can enter the channel number which is located on the router. An easier way to get started

is to click on the Scan Devices link also in this dialog page. The scan devices page will be discussed later in this document.

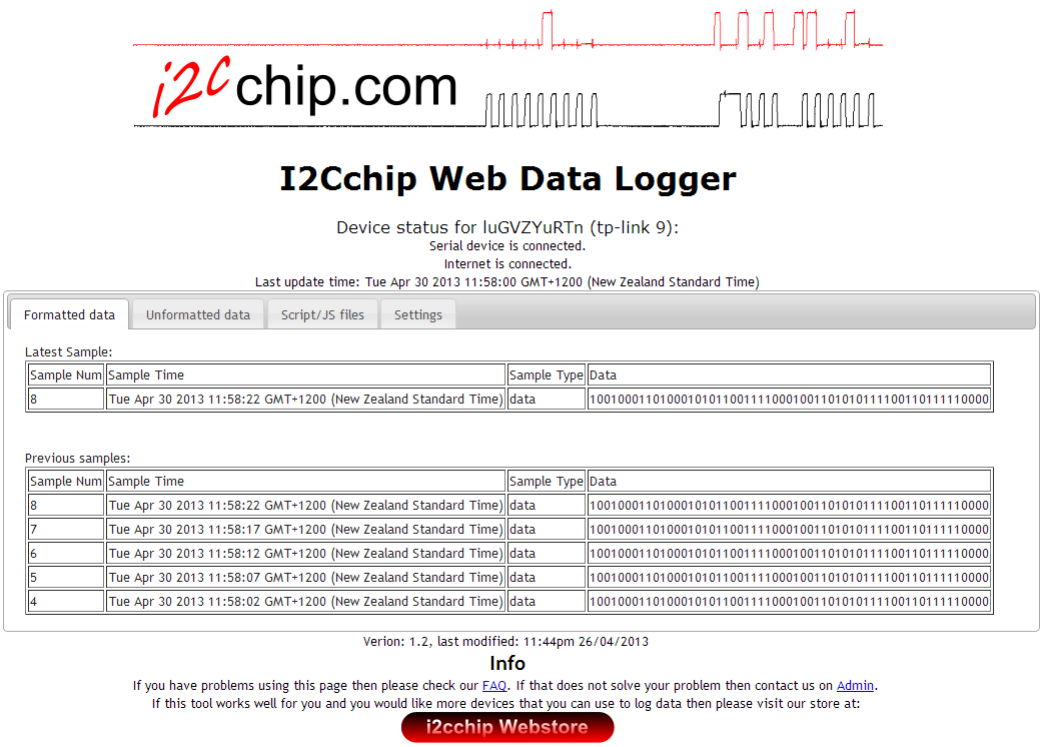


8.2. Tabs

The functionality of the website is broken up into the following tabs:

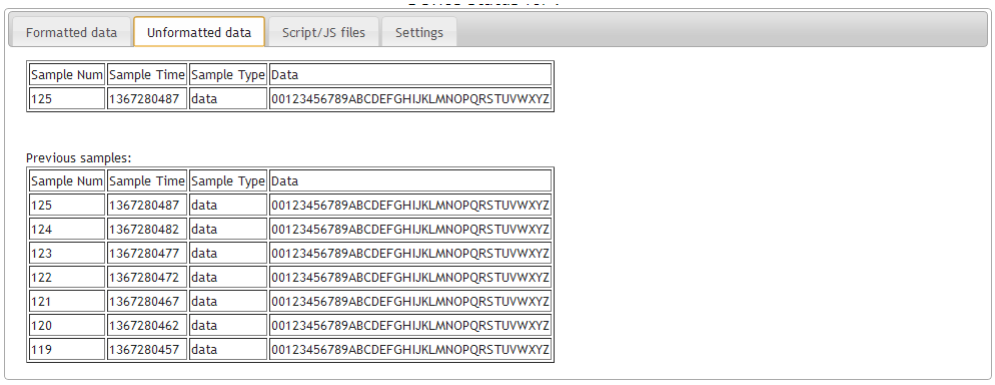
8.2.1 Formatted data

The data read from the datalogger is formatted by the format function and then displayed in this tab. It displays the latest sample at the top and all the samples that has been captured since the page was last refreshed under previous samples. The format function has the capability to display the data in any format. It is up to the user how many columns the table has and what the column headings and column widths are. See the format function section later in this document for more details on how to change the formatted data.



8.2.2 Unformatted Data

The unformatted tab shows the raw data captured from the datalogger. By default there are four columns: Sample num, Sample time, Sample Type and Data. The sample num is an incrementing number inside the router. The sample time is the number of seconds since 1/1/1970. If your router is not connected to the internet then this time will not be set correctly. The data type is set in the router config file. See the Display Javascipt section later in this document for how to edit the functions used for displaying the data.



8.2.3 Script/JS files

Under this section there are another set of tabs for viewing and editing important files.

8.2.3.1 Format Function

The format function is used to format the raw data into any format the user desires. The code in the format function is executed in line inside the function calling it, therefore the code in the format function is not defined inside a function. If the user wants to define their own functions to call from the format function then it is best to add the user functions to the Display Javascript file.

Development format function

Above the format function code there are two radio buttons where the user can select between the production format function and the development format function. The selected one will be used to format the data in the formatted data tab. The development format function is used for testing purposes before implementing the code in the production format function.

Apply and save buttons

After the user made changes to the format function and wants to store the code then these buttons are used. The apply button saves the changes to the browser's local storage. That means these changes will only be stored to this browser on this machine and nowhere else. If you want to revert back to the code stored on the router the press the "Reload user code" button in the settings tab. When the user is connected to the router over the lan then there is a save button as well. This allows the changes to be permanently saved to the router. This option should be used with caution. If there are mistakes in the code then a message will be shown telling the user that there is something wrong in the code.

Input variables

Variables available to the format function:

Variable Name	Description
ffOption	Contains the value selected in the "Select format function option" drop down list in the Settings tab
sampleData.sampleNum	Sample number
sampleData.time	Sample time
sampleData.dataType	The sample data type
sampleData.data	Sample data received from BL233
sampleData.seperatedData	Sample data that was split using comma as a delimiter. The first value is accessable as sampleData.seperatedData[0]
jsonDevData	Contains the JSON device data. To see all the variable available, click on "View device information" on the settings page. For example, to use the device friendly name use: jsonDevData.friendlyName

Variables

formattedDataStruct is the object populated with the formatted data. It contains the following members:

1. formattedDataStruct.headings is an array of strings containing the headings of the columns.
2. formattedDataStruct.colWidths is an array of strings containing the widths of each column. Set to -1 or leave undefined if the browser must autoscale the column.
3. formattedDataStruct.data is an array that contains the actual data. Each array entry must be an array with the same number of entries as the headings array.

Column headings and width

The format function is called once without data and setHeadings = true and after that it is called once for each line of data. Therefore it is important for the format function to check whether setHeading is true. The user can set the column heading names and the width of each column. The calling function initialises the column heading names as:

- Sample Num
- Sample Time
- Sample Type
- Data

If the user is happy with these names then it is not necessary to change them. The column widths are initialed to -1, which means they are not added and the browser can autoscale the table. The widths can be set to any allowed css unit, for example:

- 100px
- 50%
- 2em

See Measurement Values for more options.

Example 1, the user is happy with initialised values. The headings and width will not be changed. It is important that the format function must check the value of setHeading.

```
if (setHeadings === true)
{
} else {
// rest of the code
```

Example 2, set own column headings and widths:

```
if (setHeadings === true)
{


    formattedDataStruct.headings = ["Number", "Type", "Temperature", "Time"
    formattedDataStruct.colWidths = ["10%", "10%", "100px", "15%", -1];

} else {
```

Formatting data

The else part of (setHeadings === true) is executed for each sample. ffOption is set by a drop down list in the settings tab and is used to decide which switch statement to execute. This is useful for formatting your data in different formats before downloading. The formatted data must be stored in the formatted data array.

Example 1:

```
if (setHeadings === true) {

    //use default headings

} else {

    var tempNum;
    switch (ffOption) {
       case 0:

          formattedData.push(sampleData.sampleNum); //convert time to human
          formattedData.push(new Date(sampleData.time * 1000));
          formattedData.push(sampleData.dataType); //show data in binary
          tempNum = parseInt(sampleData.data, 16); formattedData.push(tempN
          break;
       case 1:
```

In this example the default headings are used. The data is pushed in the order: sample num, sample time, data type and formatted data. The time is converted into a human readable time. The data is converted into a binary string.

Formatting time

The time can be formatted using date.format.js. See date-time-format for more information on this.

Version number

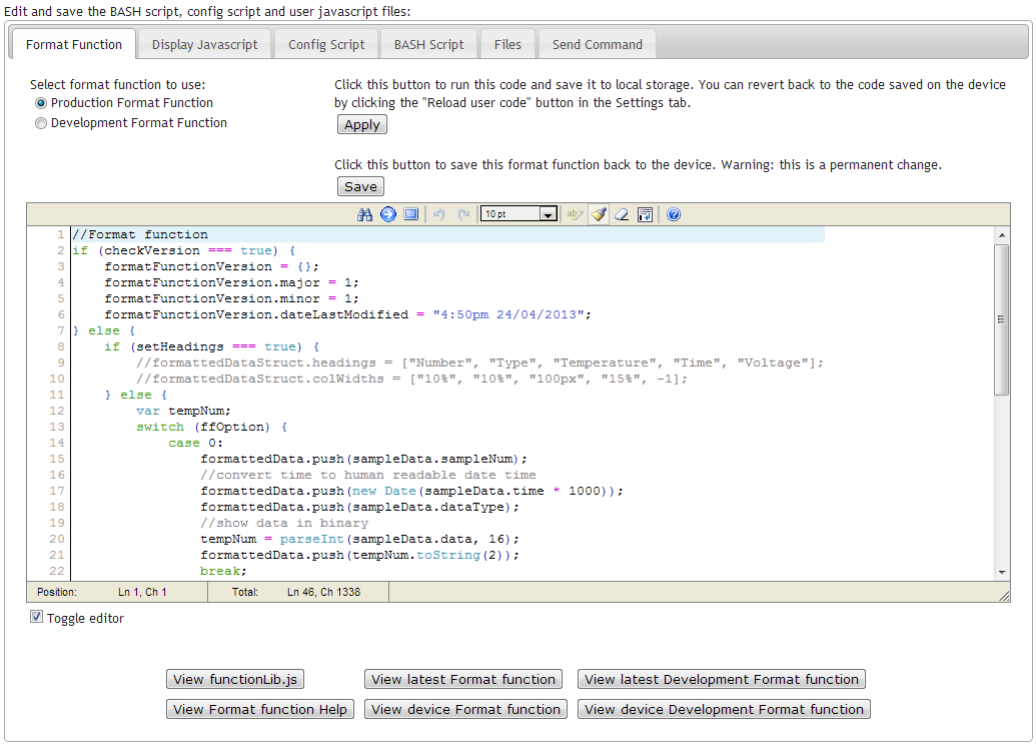
When there is an updated release of the website available and the user downloads it then the user will be notified if there are changes to the code that could make old code incompatible with the website. When this happens the user must merge his code with the new format function that can be downloaded from the website. The user must update the version numbers to be the same as the downloaded file.

The format function version is checked at the top of the format function code. This code should not be removed or changed unless a new version is available, otherwise the page will bring up messages that the format function is out of date.

Format function buttons

Below the format function edit box there are a couple of buttons that allow the user to view the format function code. Below is a description of these buttons.

- View functionLib.js: The functions declared in this file is available to the Format function and display javascript. It contains functions to display the data in html tables, functions called by the display javascript that executes the format function and i2cChip specialised string to number convert function and a real number to engineering number convert function. See the comments in functionLib.js for how to use these functions.
- View Format function Help: Displays a page containing the help you are reading at the moment.
- View latest Format function: Displays the latest version of the format function from the i2cchip.com website. If the website show a dialog box saying that the format function is outdated then this code must be merged with the existing code to make it compatable with the website again. It is important that the version numbers must be copied over as well.
- View device Format function: Displays the format function downloaded from the router. When the user press ”Reload user code” in the settings tab then this code will replace the code stored in your browser’s local storage.
- View latest Development Format function: The same as View latest Format function except that it displays the latest development format function from the server.
- View device Development Format function: The same as View device Format function except that it displays the development format function from the router.



8.2.3.2 Display javascript

The main function in the display javascript file is called: RawDataStruct2HtmlBlockLocal. This function is called by the i2cchip website javascript to render the tables in the formatted and unformatted tabs. The user can also add his own functions to this file, which can then be called from the format function. It is important that the function is declared like this:

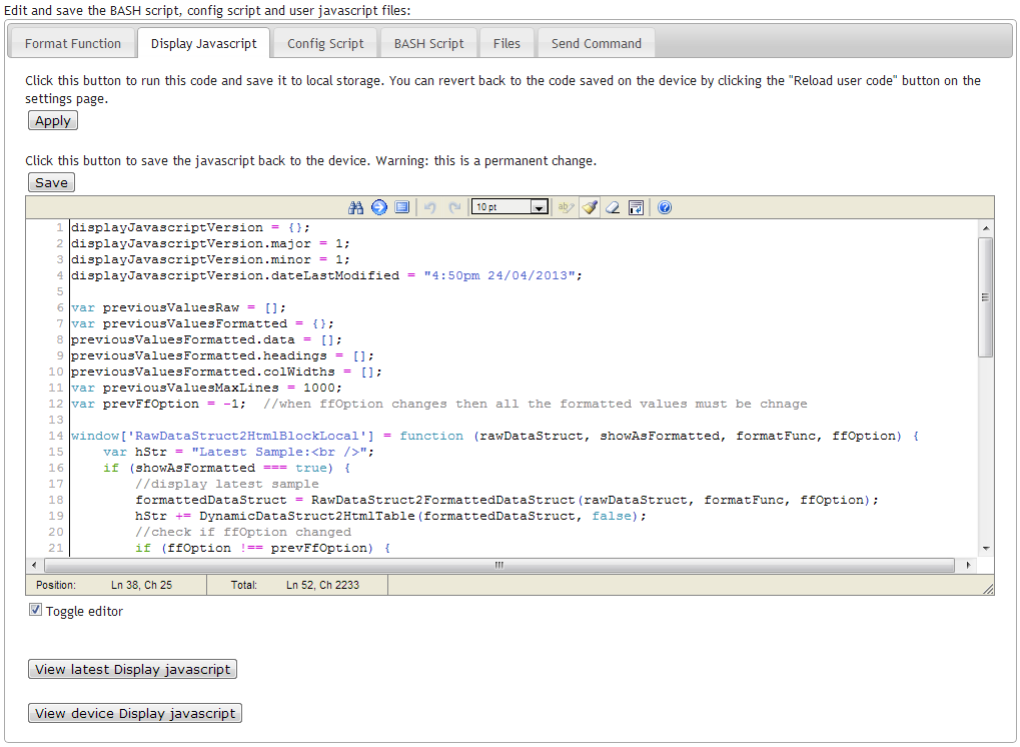
```
window['myFunctName'] = function (myVar1, myVar2) {
    //your code
}
```

where myFunctName will be the name of your function and myVar1 and 2 will be replaced by your input variables.

Above the edit box is an Apply button which saves the modifications made to the display javascript to the browser's local storage. If you are connected to the device on the LAN then there will also be a Save button. This button will save the changes back to the router. This should be used with caution because faulty code can break the product.

Below the edit box there are two buttons:

- View latest Display javascript: Displays the latest version of the display javascript from the i2cchip.com website. If the website show a dialog box saying that the display javascript is outdated then this code must be merged with the existing code to make it compatable with the website again. It is important that the version numbers must be copied over as well.
- View device Display javascript: Displays the display javascript downloaded from the router. When the user press "Reload user code" in the settings tab then this code will replace the code stored in your browser's local storage.



8.2.3.3 RawDataStruct2HtmlBlockLocal explained

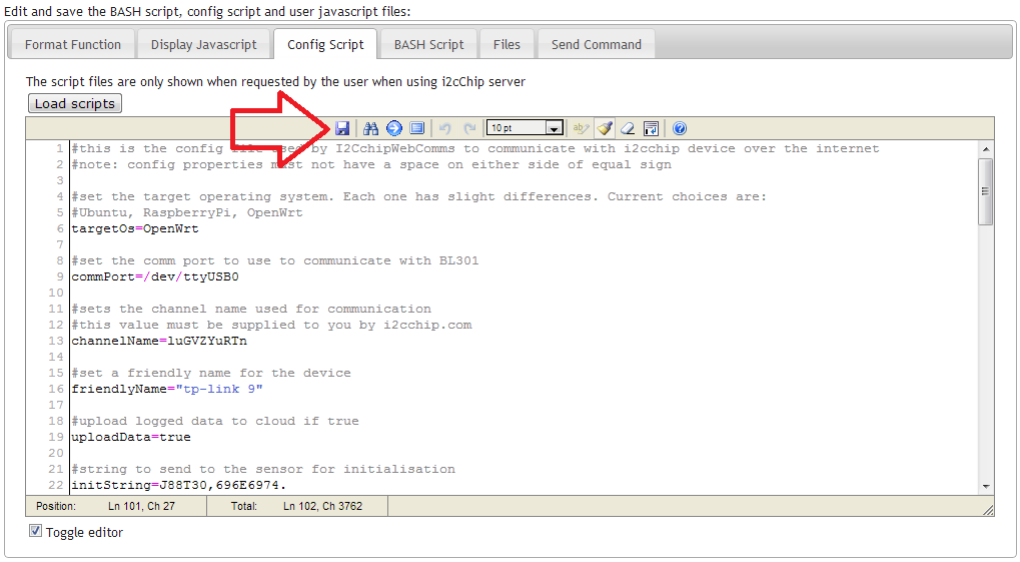
This function is important because it is called by the i2cchip.com javascript to display the formatted and unformatted data. This section gives a quick explanation about how this function works.

The first section executes if showAsFormatted == true is used to display the data in the formatted tab and it's else section is used to display the data in the unformatted tab.

1. Formatted code: RawDataStruct2FormattedDataStruct from functionLib.js is called to convert the latest raw data into a formatted data structure using the format function. See the format function help section earlier in this document to see the layout of the structure returned from this function. DynamicDataStruct2HtmlTable converts the latest sample into an html table which can be rendered by the browser. Next it checks if ffOption changed. If it hasn't changed then it only needs to convert the last sample and add it to the previous samples. If ffOption has changed then all the data has to be reformatted.
2. Unformatted code: This section always renders the html tables with four columns: Sample Num,Sample Time,Sample Type,Data.

8.2.3.4 Config script

The config script executes on the router and configures the device. It is not loaded by default from the router when connected through the i2cChip server. Press the Load scripts button to view the code. When connected to the router on the LAN, changes made to this file can be saved back to the router by clicking the save button:



The comment above each setting describes what the setting is used for. Below is extra information about the most important settings:

- channelName: Your unique channel name assigned by i2cChip. Without this channel name you won't be able to use this product.
- friendlyName: A human friendly name used to identify this device. This is very usefull if you have multiple devices.

- initString and scanString: The initString will only be sent to the BL233 once when the router starts and scanString will be sent to the router at an interval set by scanInterval. See the BL233 datasheet for how to build the commands that must be sent to your I2C device. These strings need to be quoted with single quotes. seeQuoting Strings in Bash

- dataType: The data type field returned with each sample. The initString datatype will always be 0. In the Send Command tab the user can set the datatype that will be sent back with the command's response.

- logFileMaxLines and logFileMaxLinesRun: When the raw samples file reaches the number of samples specified in logFileMaxLines then the oldest samples are deleted until the file contains the number of samples specified in logFileMaxLinesRun. By default, if the file reaches 100000 samples then the last 10000 samples will be deleted and the new file will contain 90000 samples. It stops the router RAM from filling up and crashing the device. These numbers can be made smaller if the user has problems downloading the files because they became to big.

- logFileMaxLinesJson and logFileMaxLinesRunJson: These two variables are the same as explained above except they are applicable to the JSON files.

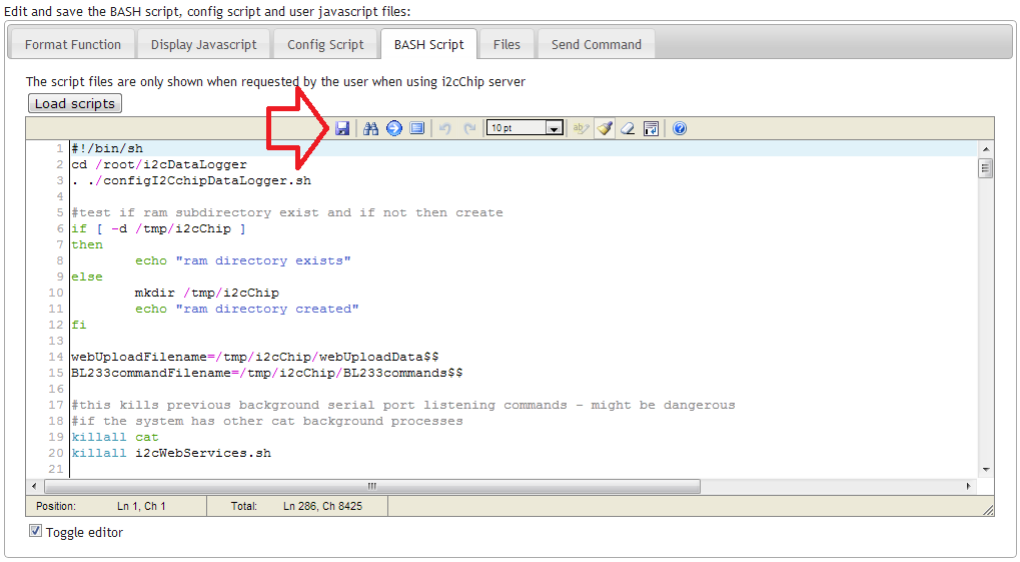
- jsonLocation and jsonGpsCoordinates: These values are user defined and forms part of the JSON data structure that is downloaded from the device. These values can be used in the format function. When you view the JSON data structure by clicking the View device information button on the settings tab you can view the device location on a Google map if the jsonGpsCoordinates contains good coordinates.

- snapImage and snapImageInterval: If you have a webcam connected to the router then these variables are used to capture and upload an image at set intervals.

- allowCommandsToBL233: If this value is true then the user can send commands to the BL233 using the Send Commands tab over the internet. This is disabled by default for security reasons.

8.2.3.5 Bash script

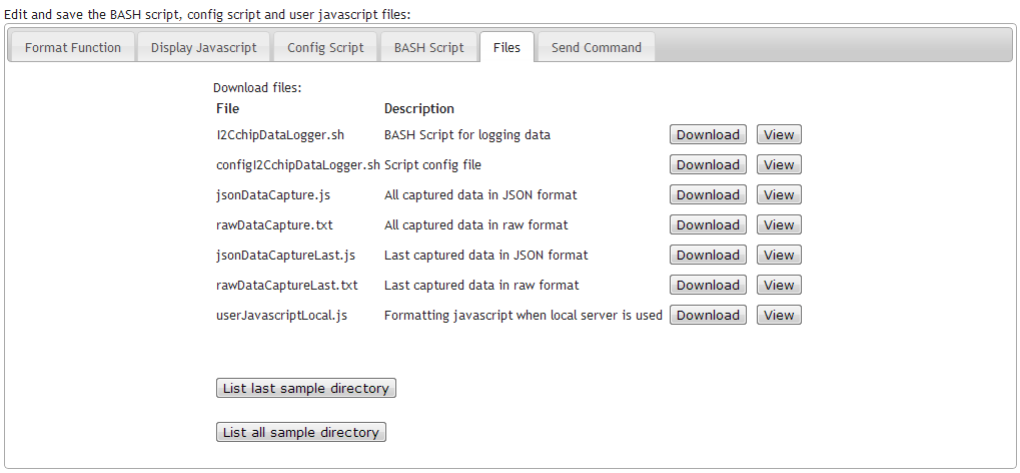
The bash script executes on the router and is responsible for communication with the BL233 and the script that does all the internet communication. It is not loaded by default from the router when connected through the i2cChip server. Press the Load scripts button to view the code. When connected to the router on the LAN, changes made to this file can be saved back to the router by clicking the save button:



This file will only be usefull to advanced users.

8.2.3.5 Files

This tab allows the user to view or download files from the router. At the bottom there are two buttons that allow the user to list the content of the sample directories.



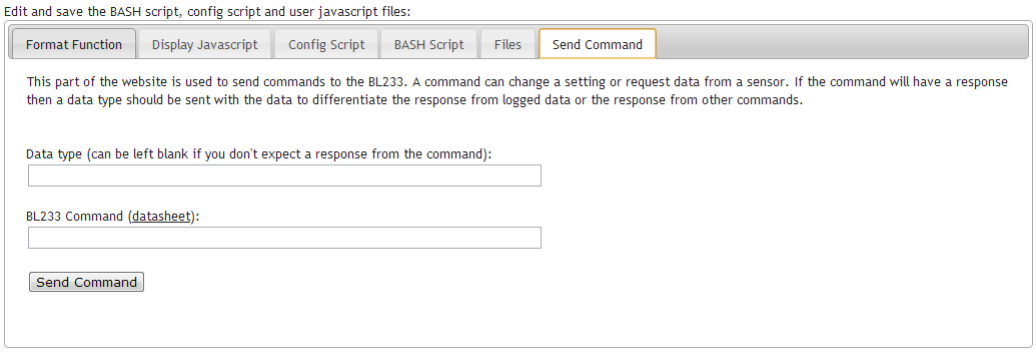
8.2.3.6 Send Command

This section allows the user to send commands to the BL233 from the internet. It can be very useful if settings on your I2C device needs to be changed. Two values must be sent:

- Data type: The response from a command is processed exactly the same as logged data and will also be stored in the sample files and uploaded to the i2cChip server. The data type distinguish it as a seperate command and can be processed differently by the format function. This value can be left blank if the command does not invoke a response from the BL233

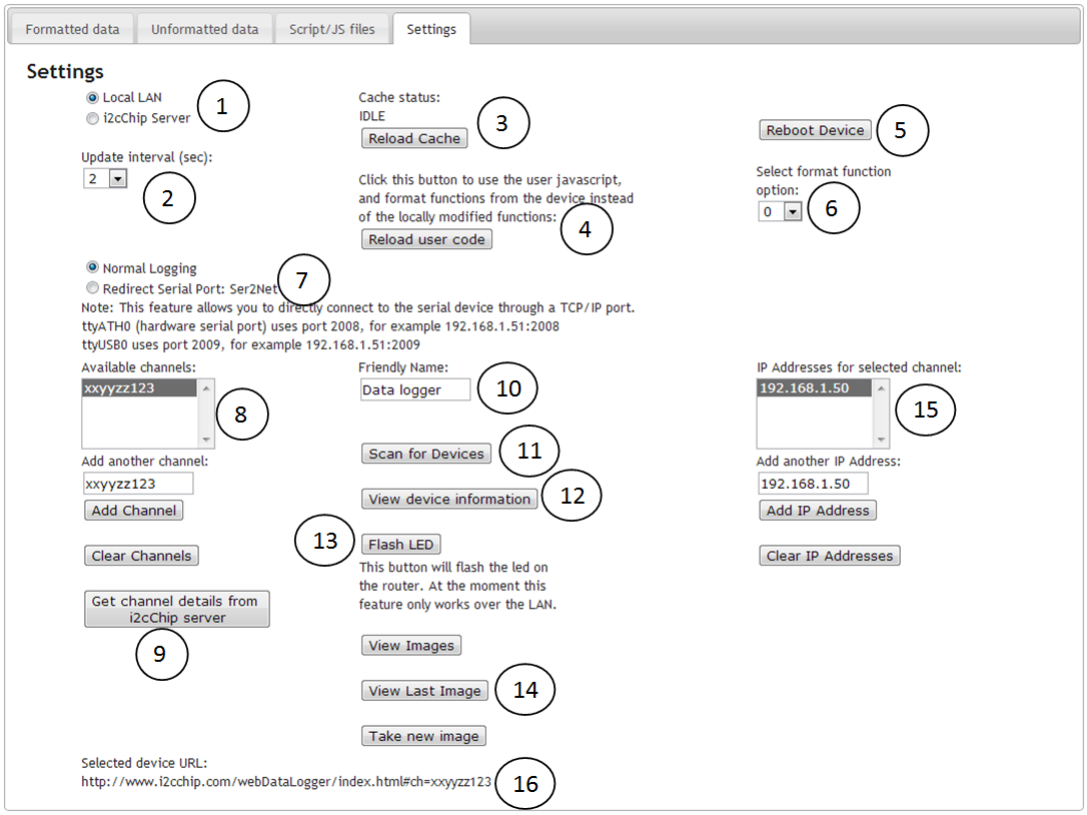
- BL233 Command: Actual command sent to the BL233. See BL233 datasheet for more details.

Sending commands will only work if it is enabled in the router config file by setting allowCommandsToBL233=true. This is disabled by default for security reasons.



8.2.4 Settings

Below is an image of the settings tab with a number by each section. Below is an explanation of each of the numbered sections:



1. This option box allows the user to choose which medium is used to communicate with the router. If the router is connected on the same local area network as the router then Local LAN can be selected. This option is much faster and allows the user to edit the scripts on the router. If you need to get data from the router over the internet then select the i2cChip Server option.

See sections 2.1 and 2.2 earlier in this document for diagrams how these two options connects to a router.

2. This interval is how often the page will connect to the server to check if a new sample is available. This interval must be less than your scan interval set in the router config file, otherwise the website will skip samples. Smaller scan intervals will make the website get samples process more responsive at the price that it will generate more internet traffic.

3. Occasionally there will be improvement uploaded to the website. The website is stored in the browser's local storage so that the website and router can work without being connected to the internet. The next time the user connected to the internet and a new version of the website is available then a diolog will pop up asking the user if he wants to download the latest version. If the user chooses not to upgrade at this point in time then this button allows the user to upgrade when he is ready.

4. The website will use the format function and display javascript stored in the browser's local storage if available to format the data. The website will also load the format function and display javascript from the router and keep it in the background. If the user wants to revert back to the router format function and display javascript then this button is used. Warning: using this button you can easily loose user defined code.

5. When connected to the device over the LAN this button is used to reboot the router. This is necessary if the user made changes to the config or bash scripts.

6. Inside the format function is a switch statement for ffOption. The number selected in this drop down list determines which section of the format function switch statement will be executed.

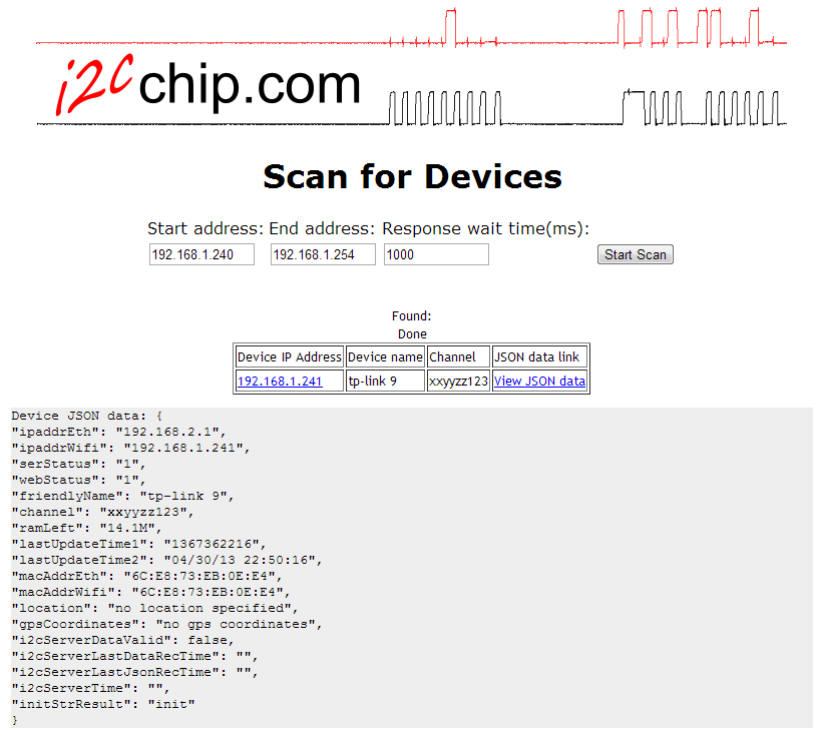
7. This option allows the user to select between normal data logging mode or ser2net mode which allows the user to connect straight to the router's serial port to send data. For this option to work the user must be on the same LAN as the the device. See section 7 earlier in this document for more details on how to use Ser2Net.

8. This section consists of a listbox containing all the currently loaded channels, an edit box and button for adding a channel and a button which allows you to clear all the channels. The channel name is the secret code which i2cChip supplies with each router which allows the device to connect to the i2cChip server. It is important that this code must be kept secret because anybody can connect to your device over the internet if they have this code. The channel name is located on a sticker on the router.

9. This button loads the latest device information Json data structure from the i2cChip server. Press the View device information button to view this data structure.

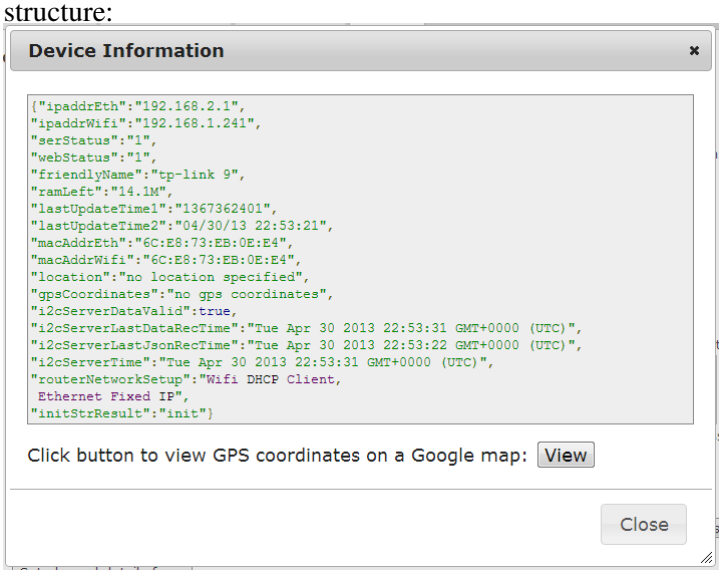
10. This field contains the human friendly name for the device. This name can be changed in the config file on the router.

11. This button brings up a page which allows the user to scan the local LAN for i2cChip data logger routers:



Enter the start and end IP addresses and press the scan button. If you are unsure what address range to use then log into your router (the one giving you access to your network and internet, not the one supplied by i2cChip) and look under the DHCP section to see what range of addresses is assigned. When a device is found then it is displayed in a table. It shows the device IP address, the device friendly name, the secret channel name and a link which displays the router's device information JSON structure as can be seen above.

12. Pressing this button brings up a dialog box showing the router's device information JSON data structure:



13. This button allows the user to flash the LED on the router it is currently connected to. This is very handy when you have multiple devices and you want to make sure that you are communi-

cating with the correct one.

14. When you have a webcam connected to the device this option allows you to view and take new images. If the config file on the router is correctly configured then the router will take images at a regular interval and upload it to the server. The View Images and View Last Image buttons allows the user to view taken images. The Take new image button will take a new image and upload it.

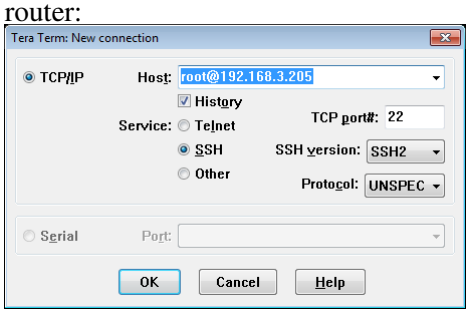
15. This is the IP addresses of the router. There can be two IP addresses, one for the router's ethernet port and the second for it's wifi. Make sure you select the correct one when trying to connect to the device over the LAN. You can enter the IP addresses yourself or allow the i2cChip server to populate then by selecting the i2cChip server option in number 1 above.

16. When you want to view your data logger information on a different browser or device then you have to enter the channel name again before you can get data from the router. This URL simplifies this process because the channel name is part of the URL. This makes it easy to email or copy the URL to another machine and get the website up and running instantaneously.

9. Upgrading scripts on router

Upgrading the scripts on the router should be a relatively painless exercise because the router is released with a script that does the upgrading from the internet. Upgrading the scripts is seen as an advanced feature and therefore we do not allow this to be done from the i2cChip website. Follow these steps to upgrade the firmware:

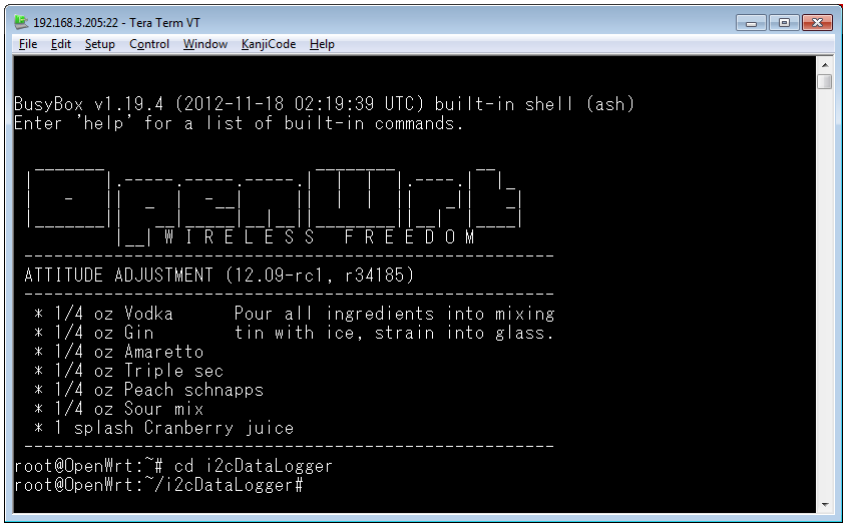
1. Connect to the router using SSH. Our preference is to use TeraTerm. Enter the IP address of the router:



The default credentials of the router is:
Username: root
Password: admin

In the image above the username is entered as part of the IP address, in which case TeraTerm will only ask for the password. If you are unsure what the IP address of the router is, check the IP address listbox in the settings tab in http://www.i2cchip.com/webDataLogger/index.html.

2. Enter: cd i2cDataLogger
Your screen should look like this:



3. To upgrade the scripts enter the following command:
   ./updateScripts.sh

   This will download and unzip the new scripts and set all links and permissions.

4. There are 4 scripts which are user configurable:

   a) configI2CchipDataLogger.sh

   b) displayJavascript.js

   c) formatFunction.js

   d) formatFunctionDev.js

Copies of these four files before the upgrade are copied into a directory called: oldScripts. The user can use these files to copy changes over to the new files.There is also a script that allows the user to copy the old user configurable files back into the new scripts called: useOldScripts.sh. This should be used with caution because if there are important changes to any of these four files then those changes will be lost when this script overwrites the four files with their old copies.

10. Digging into OpenWRT

10.1. Tools

- TeraTerm: Telnet and SSH terminal
- WinSCP: file upload/download through SSH
- Realterm:

10.2. SSH Connection

Connect using TeraTerm, SSH to the IP address.

User: root

Password: admin

Changing the password: Type “passwd” at the prompt, and enter new password.

10.3. VI Editing Files

VI commands are less obscure than Linear A.

VI Cheat Sheet

- “i” insert mode, “I” insert line
- <esc> return to command mode
- “:x” save and quit
- “:q!” force quit

10.4. Network Statistics

To see how much network data is being used look at:

```
cat /sys/class/net/wlan0/statistics/rx_bytes
cat /sys/class/net/wlan0/statistics/tx_bytes
cat /sys/class/net/eth0/statistics/rx_bytes
cat /sys/class/net/eth0/statistics/tx_bytes
```

There are several packages to make neater stats of this eg *vnstats*

10.5. Storage Space

df shows disk free space

free shows ram space.

10.6. Useful links

WR703 Hardware Page

Freeing Serial of console messages

WR703 Schematic

11. Sample I2C Setups

11.1. SIP8574

Reads the 8 bit port. Flashes the LED on bit 7 during reads. Unconnected pins read as 1.

initString: J88T30,S40FFR01P.

scanString: J88T31,S4101,W7FPL0200WFFP.

12. Security

Change the passwords! There is a WiFi password, and an OpenWRT SSH password.

Your data on the internet is kept secret by the ChannelCode remaining secret. The ChannelCode is available to people in your building i.e. who can scan your LAN, or physically go and look at the device. I2CCHIP knows all ChannelCodes.

While the configuration and scripts can be seen from the internet, the configuration cannot be saved to the device unless you are on the LAN.

Commands can be sent from the browser to the device.

The device is not directly accessable from the internet, it only communicates directly with our server. So it is not opening up port access to your LAN from outside.

If you are not using WiFi, disable it.

If you are using the WiFi Access Point mode, change the password.

12.1. Intentional Leakage

You may not want to reveal some information:

- Friendly Name: “Unguarded Gold Brick” might not be the best choice, if true.
- Location and GPS Coordinates
- IP Addresses (lan) will be leaked by jsonData
- Channelcode is displayed on the webpage, visible to passers-by shoulder surfing.

13. Data Reflector

Internet access to the Weblogger requires a data reflector. This is a NodeJS server. Contact us if you want to run your own private reflector.

14. Hardware

The I2C Hardware is almost identical to that of the I2C2PC, and its datasheet and application notes should be referred to also.

The TPLink WR703N is reverse engineered here, notes, Openwrt page

Power (5V) comes from the router USB power in, and is unswitched and unregulated. The whole BL233B can be powered from this 5V or an internal 3.3V 250mA regulator.

Bus 3 VDD can be separately powered from 3.3V via J4.

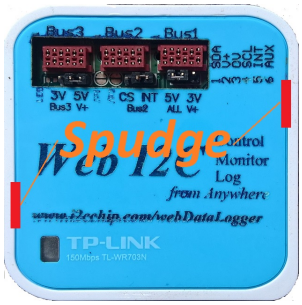
J1	VDD Select
1	5V (USB in)
3	3.3V 250mA

J3	Bus2 INT Pin
1	CS2 out
3	IRQ in

J4	Bus3 VDD
1	3.3V
3	VDD (J1)

	Solder Links	
JP1	Bus 3 INT to CS3	
JP2	Bus 3 INT to IRQ	
JP3	Power Switch to TPLINK	

To change the solder links you will need to open the case with a spudger.



14.1. Serial Interface

The serial port is buffered by U3 and has TTL input levels, which are compatible with 2.7 or 3.3V levels from router serial ports.

The ACTIVE pin (GPIO14) is used to block the serial port during boot, when used on Linux port TTYS0. This is because the default is for all boot messages to be streamed out the console (ttyS0) during boot. (these must be blocked from BL233)

14.2. Router Power Switch

The BL233 is powered directly from the +5 USB in, so it always has power, and can control the routers internal power.

Close JP3. The TPLINK must be modified to connect the power switch output to the regulator feedback. When Q1 is closed the feedback voltage to regulator U5 (R99+R100 untion) on the WR703 is raised above the reference level (0.63V), and the regulator will shut down, powering the WR703 off. This is done through a resistor connected right at R99+R100 junction. R99//R100=54k. 180k+10k@3.15V will raise it to 0.70V. So R should be 100k - 180k

CS3 = 0 turns thr router off. (“O00DF”). “O00FF” turns it on.

With the default timer divisor, the max delay / watchdog time is 65sec. You will need to change eeprom setting TimerDiv (see sec 14.1). When TimerDiv is 0xFFFF, the max delay/watchdog time is 19min 25sec, and each timer tick is 17.8ms

- Use deLay command to turn the router back on after a period. Change the eeprom timer settings to allow long periods eg “JA8 O00DF LFFFF O00FF”
- Put the BL233 into an interrupt loop, and use and interrupt macro to turn the router back on
- Set a serial watchdog macro to turn the router back on. (see sec 15)

If you are using the router power switch, you should probably always have a watchdog macro set, even if you are not using the watchdog. eg “O00FF JA8<” turn router on and re-enable watchdog

Note that as soon as the off command is executed, the flow of characters will stop, and any subsequent command will be lost.

- Put your stop command in a macro, subsequent commands are stored. “JA8 O00DF LFFFF O00FF<”
- use pause “:” command to ensure stop commands are in the buffer. “JA8 :O00DF LFFFF O00FF;”
- Put a delay before the stop commnd that is long enough to *guarantee* that the subsequent commands are in the buffer. “JA8 L0001 O00DF LFFFF O00FF”

14.3. Use with Rapberry Pi.

Fit female header J5. This plugs directly onto the Rapberry Pi expansion connector. see B.2

14.4. Options

Power LED: Remove R23, fit R20. LED becomes Power indicator.

ACTIVE Default to OFF when not driven: Remove R18, Fit R19.

15. Specifications

Spec	value		unit
Current Drain Wifi ON	200		mA
Current Drain Wifi OFF			mA
Current Drain Sleeping	20		mA
Current for USB Hub			mA
Current for I2C2PC			mA
Current for WebCam			mA
Router Internal I/O Voltage	2.7		V
Boot Time	25		sec

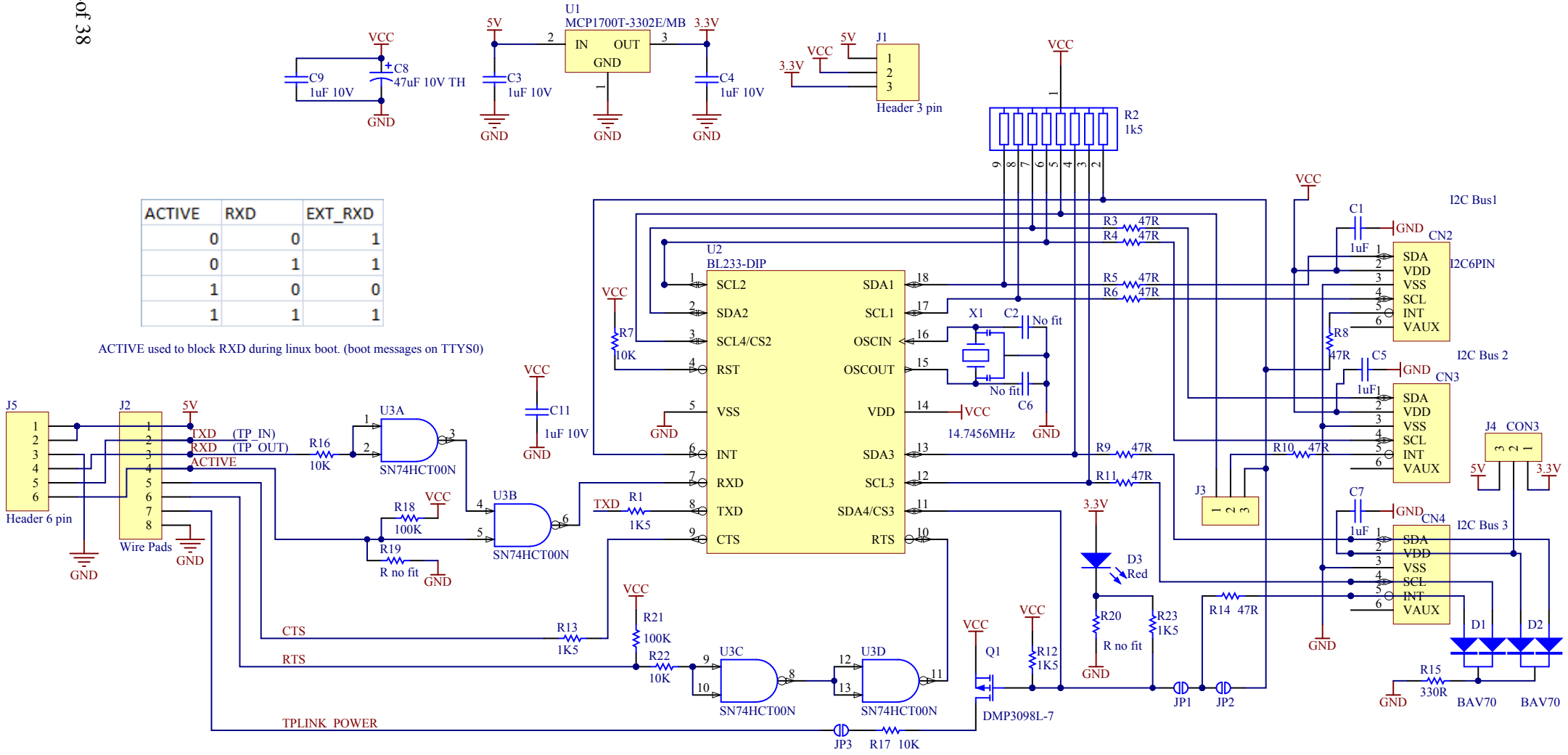
A. Revision History

Date	Rev#	Changes
31 Aug 2015	0	new
14 Sep 2017	1	

B. Schematics and Drawingsa

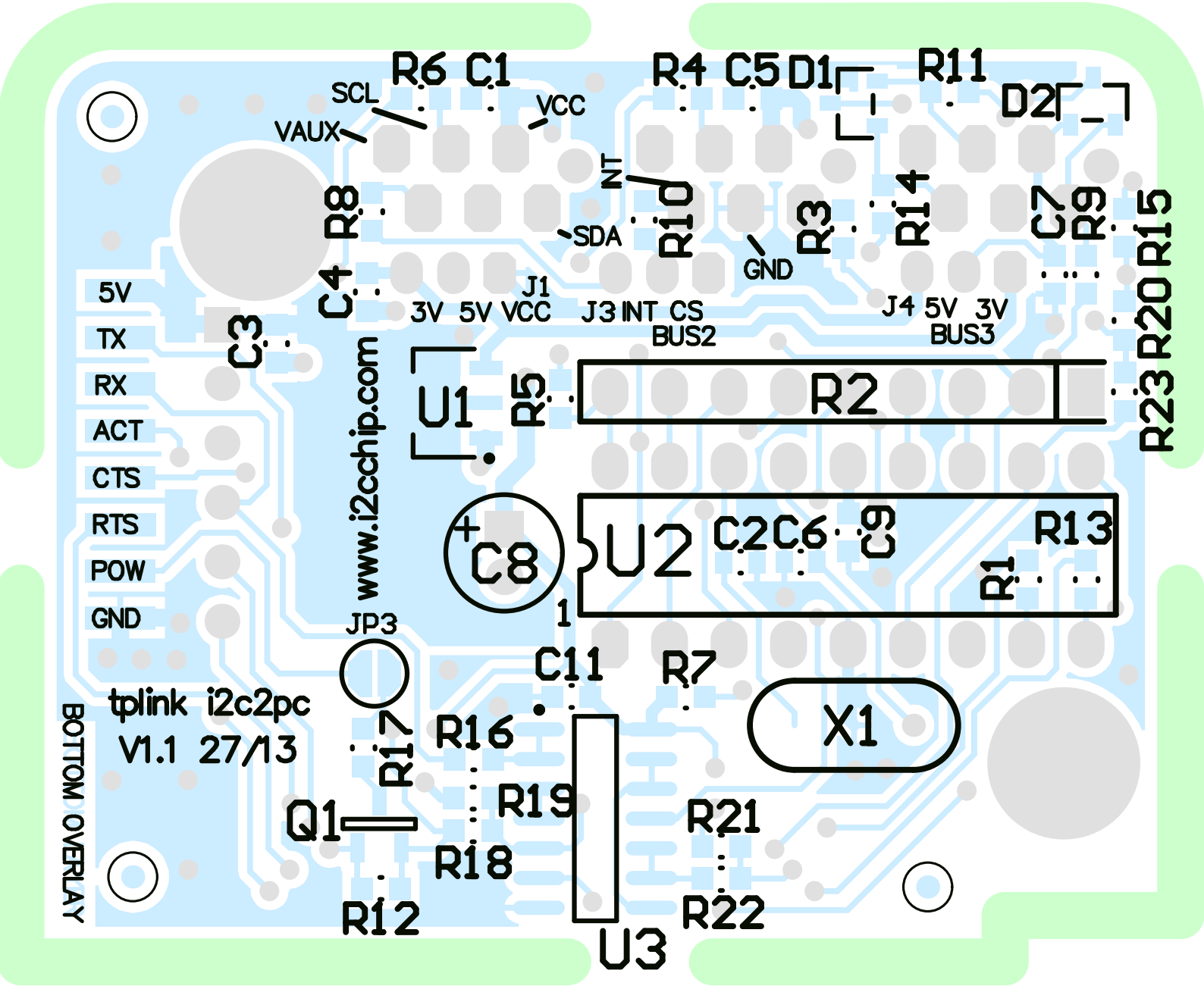
ACTIVE	RXD	EXT_RXD
0	0	1
0	1	1
1	0	0
1	1	1

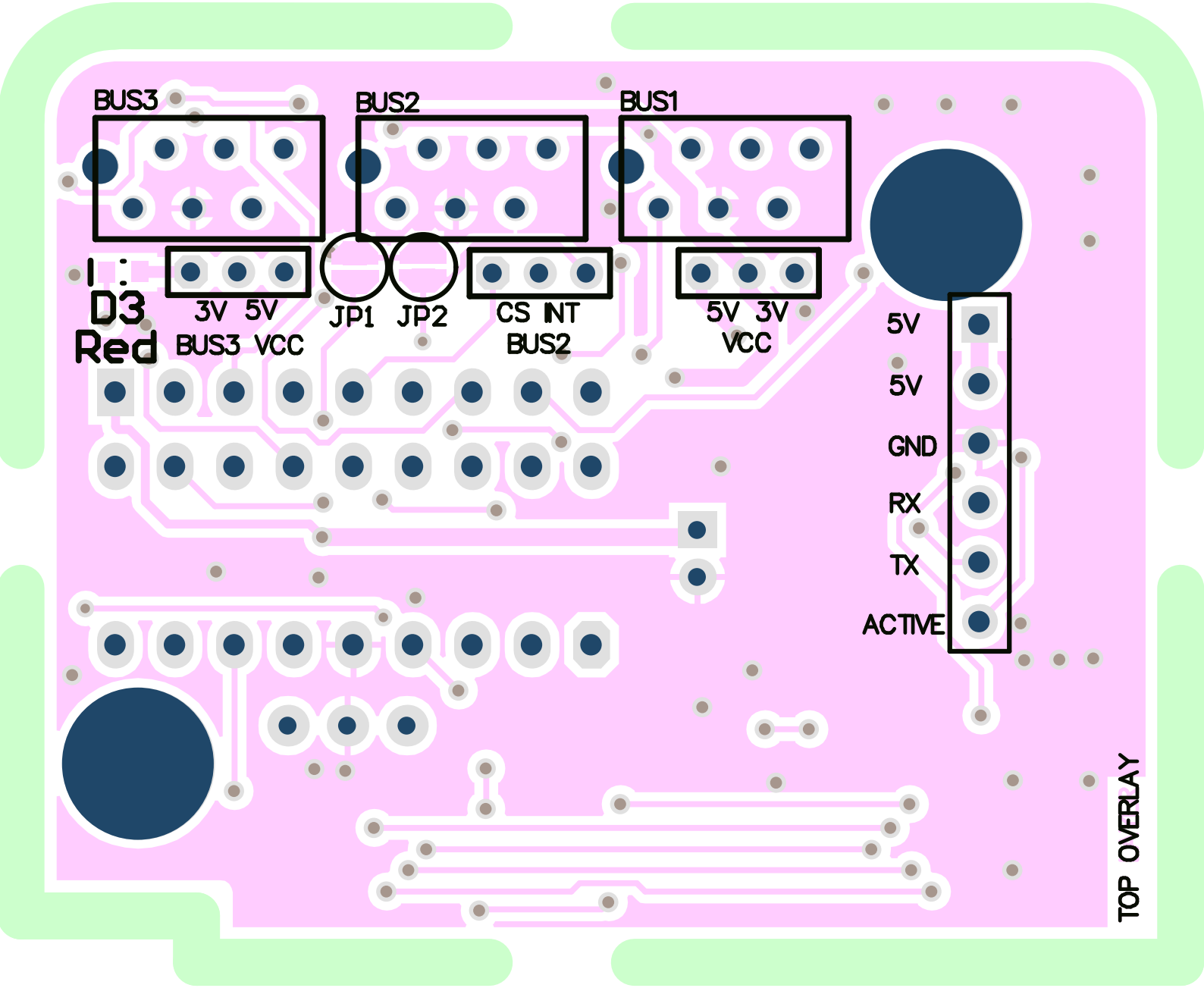
ACTIVE used to block RXD during linux boot. (boot messages on TTYS0)

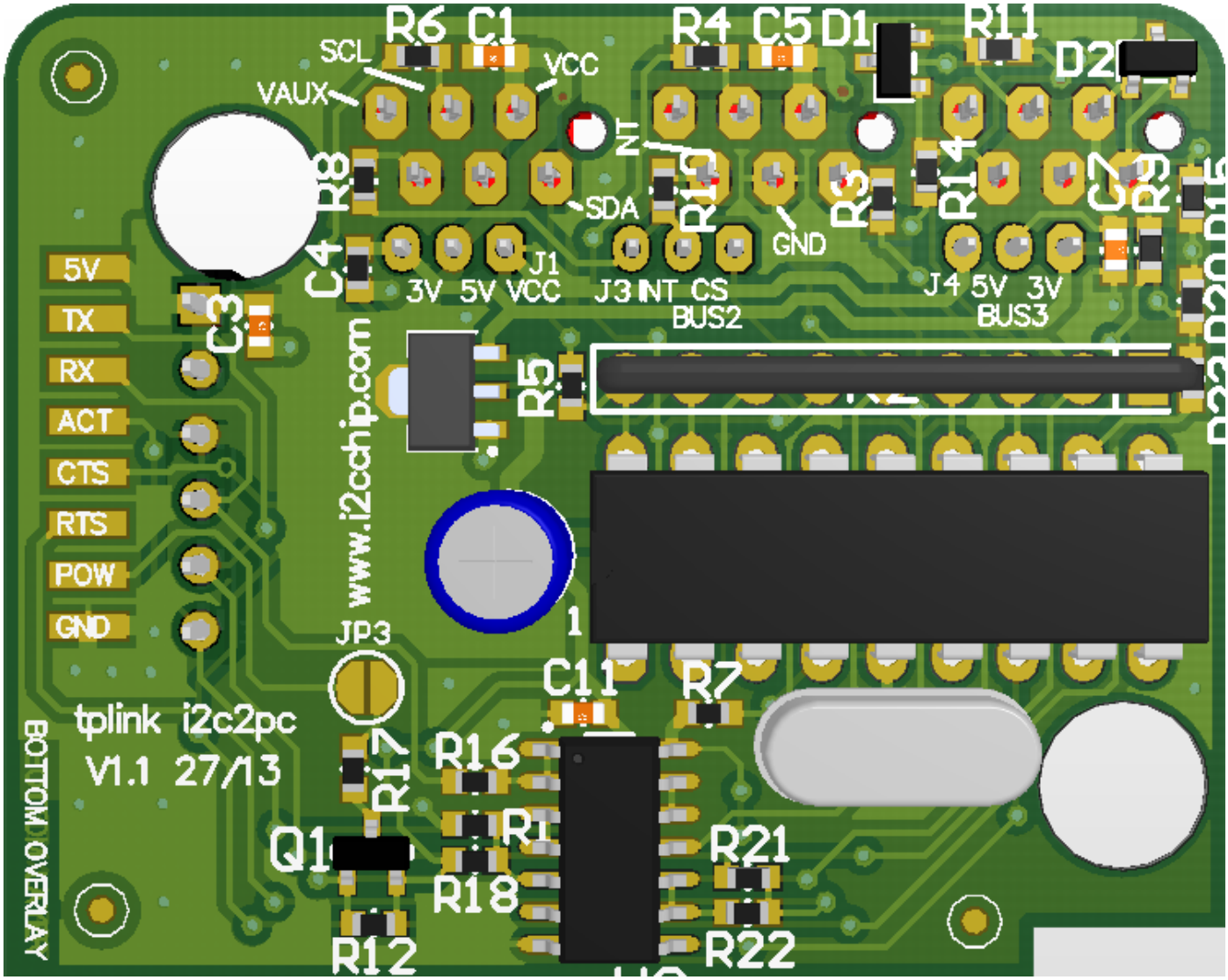


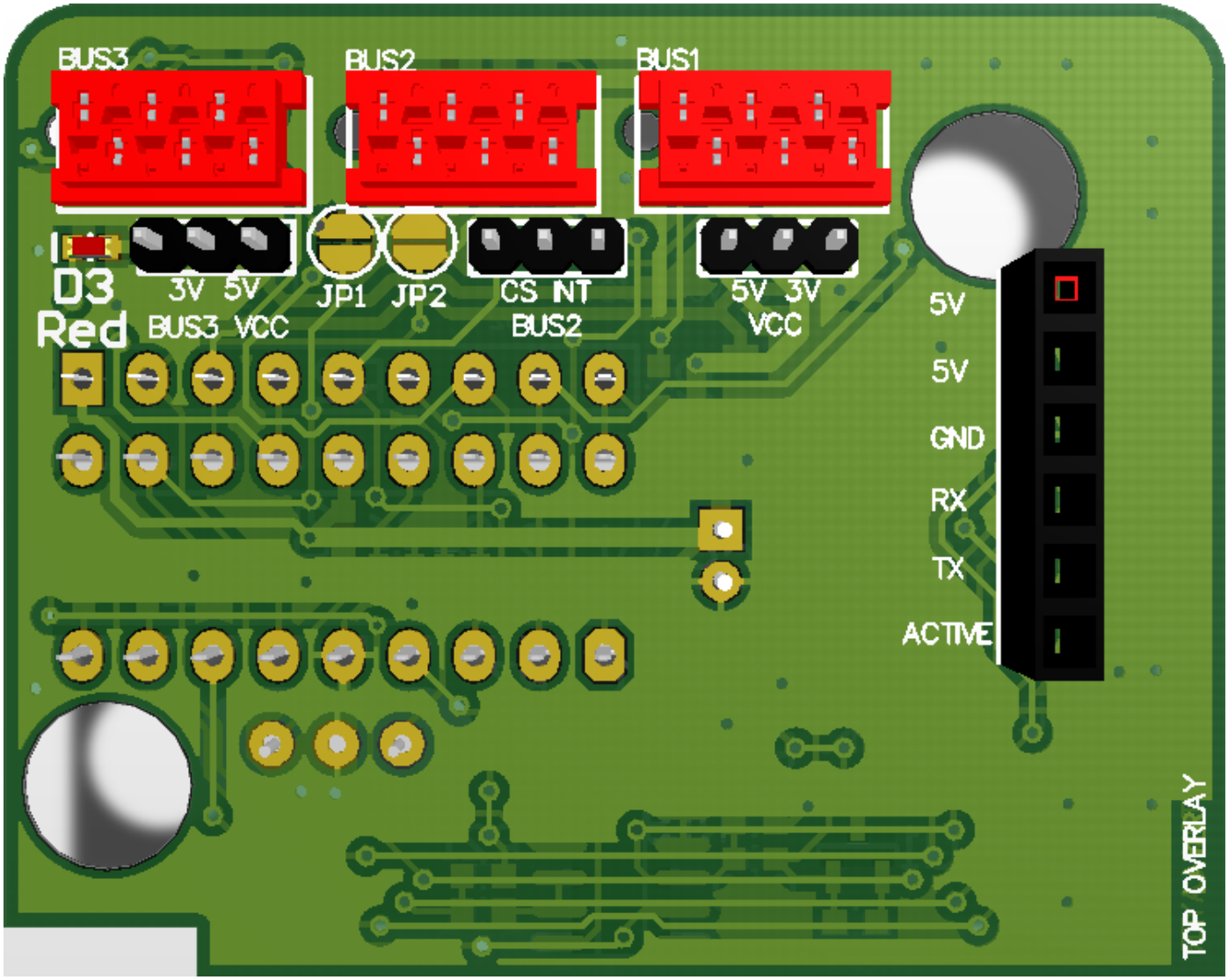
X1 is Ceralok with built in capacitors. C2,6 only fitted if X1 is Crystal

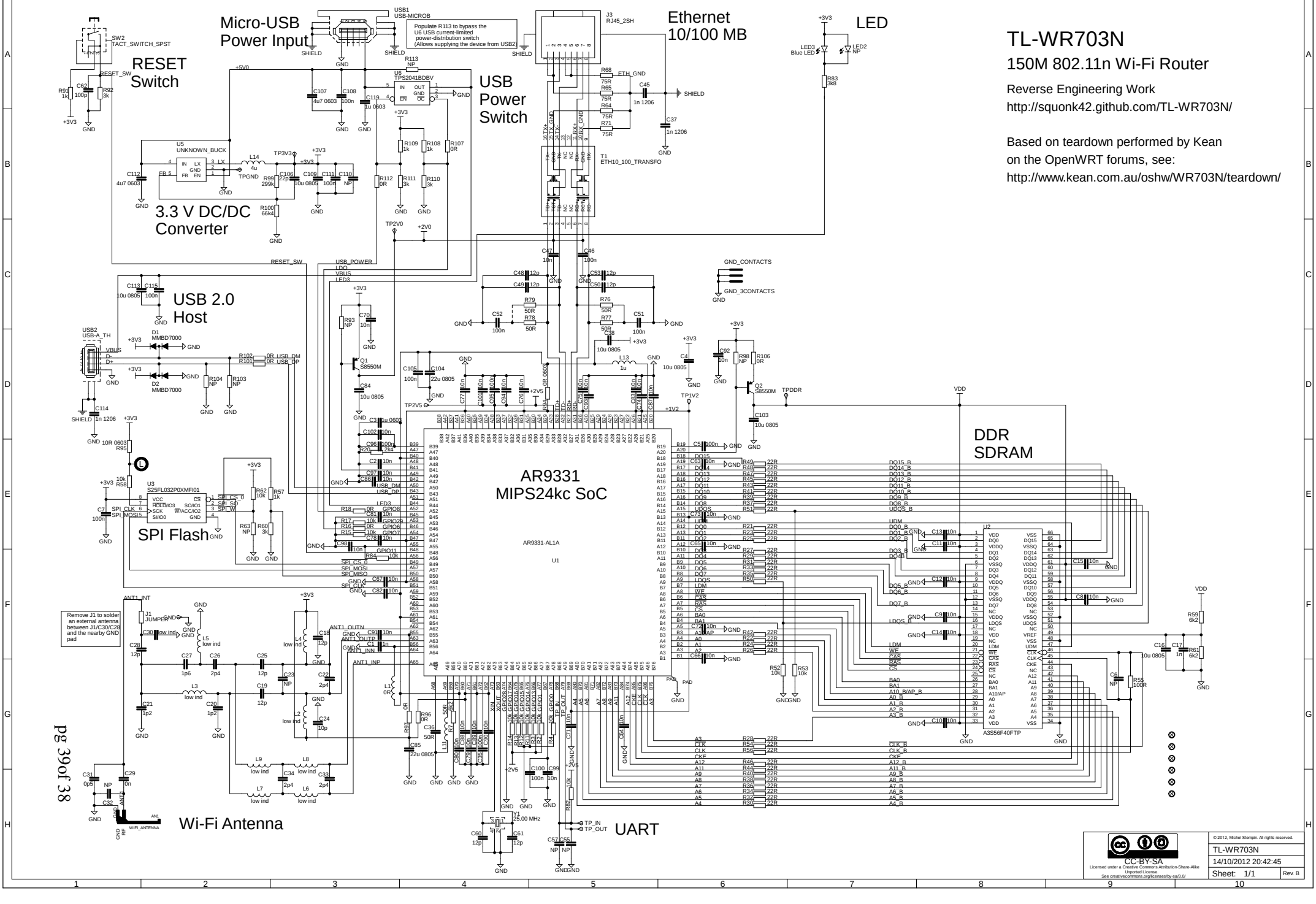
Title	I2C Adaptor for TpLink / Raspberry Pi	
Size: A4	Number	Revision: V1.1
Date: 8/30/2015		Sheet 1of 1
File: C:\Users\..\tplink_i2c2pc_A4.SchDoc		Drawn By: H Jetschko


R6 C1
SCL
VAUX
VCC
R4 C5 D1
R11
D2
INT
SDA
GND
R8
R10
R3
R14
C7
R9
R15
R20
R23
C4
C3
J1
3V 5V VCC
J3 INT CS
BUS2
J4 5V 3V
BUS3
5V
TX
RX
ACT
CTS
RTS
POW
GND
U1
R5
R2
www.i2cchip.com
C8
U2
C2
C6
C9
R13
R1
JP3
C11
R7
X1
tplink i2c2pc
V1.1 27/13
R17
R16
R19
R21
Q1
R18
R12
R22
U3
BOTTOM OVERLAY




R6 C1
SCL
VAUX
VCC
R4 C5 D1
R11
D2
INT
SDA
R8
R10
R3
R14
C7
R9
C4
J1
3V 5V VCC
J3 INT CS
BUS2
GND
J4 5V 3V
BUS3
5V
TX
RX
ACT
CTS
RTS
POW
GND
C3
R5
www.i2cchip.com
JP3
C11
R7
tplink i2c2pc
V1.1 27/13
R16
R17
R1
Q1
R18
R21
R12
R22
BOTTOM OVERLAY




TL-WR703N
150M 802.11n Wi-Fi Router
Reverse Engineering Work
http://squonk42.github.com/TL-WR703N/
Based on teardown performed by Kean
on the OpenWRT forums, see:
http://www.kean.com.au/oshw/WR703N/teardown/
Micro-USB Power Input
RESET Switch
USB Power Switch
Ethernet 10/100 MB
LED
3.3 V DC/DC Converter
USB 2.0 Host
SPI Flash
AR9331 MIPS24kc SoC
AR9331-AL1A
U1
DDR SDRAM
A3S56F40FTP
Wi-Fi Antenna
UART
Remove J1 to solder an external antenna between J1/C30/C28 and the nearby GND pad
pg 39of 38
CC-BY-SA
© 2012, Michel Stempin. All rights reserved.
14/10/2012 20:42:45
Sheet: 1/1
Rev. B