# 2 Kit Setup:

Kits will be delivered from an Arrow FAE with a version of the AT\_Command code preloaded to work over a simple USB interface.

The following instructions allow a kit to be programmed

Following instructions to request AT code from Murata then modify the process to allow a virtual COM port for the UART interface.

Murata will allow their AT Command code to be pre-programmed for demonstration purposes. Murata requires your agreement with their conditions before providing the AT Command source code. Until the details of programming a compiled version of the code onto a kit are documented (Method #1), the process to program a kit will require source code and use of WICED SDK (Method #2).

Method #1: (This process requires development)

* 1.1 Download Single Firmware file from GitHub
* 1.2 Use Modus Shell to load Firmware into Kit

Method #2:

* 2.1 Request access to Murata AT Command source code
* 2.2 Import Murata Source code into WICED SDK file structure
* 2.3 Modify Platform File to utilize the UART through the USB debug interface
* 2.4 Compile Boot Code
* 2.5 Compile and program Murata AT Command program

Communication with the CYWxx907 based kit:

* 3.1 Setup Virtual UART Connection
* 3.2 Setup Terminal Emulator
* 3.3 Test an AT Command

## 2.1 Request Access to murata at command source code:

AT Command Code from Murata = atcmd\_release\_20211129.zip

* + Use this draft e-mail to request the code



* + – or - Send a request from your company e-mail using the following format exactly with your personal information filled in after each colon. A bot is looking for the key words below to respond to your request:

To

[wifi\_at\_command@murata.com](mailto:wifi_at_command@murata.com)

Subject

Murata AT Command Software Request

Body

I would like to try Murata AT Command Software. Here is my project information.

Company Name:

Country:

Company Address:

Murata/Distributor Contact Person’s Name:

Application:

Module (Type1GC or Type1LD):

Prototype Schedule:

Production Schedule:

Annual & Lifetime Volume (kpcs):

* + When you receive an e-mail response, reply with a copy of the exact format of the response, appending a “yes” to the end

Body

Accept Terms for Use of the Software (Yes or No): Yes

Murata will respond with the following documents:

* atcmd\_release\_20211129.zip
  + This zip file contains two directories and a license
    - TEST\_TOOL 🡸 We won’t be using this
    - WICED 🡸 contains two versions of the AT Command, one for each WICED 6.4 and 6.6
      * Copying the 43xxx\_Wi-Fi directory onto your existing WICED directory of the same name will add some files and replace others.

## 2.2 Import Murata Source code into WICED SDK file structure:

Graphical user interface, text, application, email

Description automatically generated

The 43xxx\_Wi-Fi folder from Murata contains 18 files.

Copying the entire 43xxx\_Wi-Fi directory from Murata onto the directory of the same name in your existing installation of WICED will add 13 files under the new directory “\43xxx\_Wi-Fi\apps\test\at\_cmd”.

Into directory \43xxx\_Wi-Fi\apps\test\at\_cmd

11/29/2021 12:00 PM 3,725 **at\_cmd.h**

11/26/2021 07:48 PM 25,846 **at\_cmd.mk**

11/29/2021 12:00 PM 2,118 **debug\_log.h**

08/23/2021 07:02 PM 2,029 **makefile**

11/29/2021 12:00 PM 3,989 **os\_wrapper.h**

11/29/2021 12:00 PM 10,680 **os\_wrapper\_wiced.c**

11/29/2021 12:00 PM 5,149 **process.h**

11/29/2021 12:00 PM 54,967 **process\_wiced.c**

11/29/2021 12:00 PM 24,649 **recv\_thread.c**

11/29/2021 12:00 PM 1,621 **recv\_thread.h**

08/23/2021 07:02 PM 21,671 **usb\_device\_cdc\_acm\_atuart.c**

08/23/2021 07:02 PM 3,560 **usb\_device\_cdc\_acm\_atuart.h**

08/23/2021 07:02 PM 4,306 **wiced\_init.c**

Copying Murata’s 43xxx\_Wi-Fi folder over your existing WICED folder of the same name will also replace the following 5 files:

* ota2\_extract.mk
* ota2\_bootloader.mk
* ota2\_failsafe.mk
* WICED.mk
* Graphical user interface, text, application

  Description automatically generatedBCM94390x\_targets.mk

If you prefer, copy or move the five files before copying Murata’s files in.

If you don’t first remove the five files, accept the replacements.

Following are the destinations for the 5 replacement files

Into directory \43xxx\_Wi-Fi\apps\snip\ota2\_extract

08/23/2021 07:02 PM 3,194 **ota2\_extract.mk**

Into directory \43xxx\_Wi-Fi\apps\waf\ota2\_bootloader

08/23/2021 07:02 PM 3,648 **ota2\_bootloader.mk**

Into directory \43xxx\_Wi-Fi\apps\waf\ota2\_failsafe

08/23/2021 07:02 PM 3,075 **ota2\_failsafe.mk**

Into directory \43xxx\_Wi-Fi\WICED

08/23/2021 07:02 PM 6,590 **WICED.mk**

Into directory \43xxx\_Wi-Fi\WICED\platform\MCU\BCM4390x

10/08/2021 07:42 PM 30,232 **BCM94390x\_targets.mk**

Other support documents from Murata

* Murata’s AT Command Specification
  + N3-0568\_Murata\_Wi-Fi\_AT\_Command-1.1-Specification\_Rev.C (Available with AT Command Code from Murata described in the Software section above)
* Murata’s AT Command Quick Start Guides for UART
  + N3-0567\_Type1GC\_1PS\_AT\_Command(UART)-Quick\_Start\_Guide\_RevD @ <https://www.murata.com/-/media/webrenewal/products/connectivitymodule/asset/pub/rfm/data/n3-0567_type1gc_1ps_at_command_uart-quick_start_guide.ashx?la=en&cvid=20211022010000000000>

## 2.3 Update Quicksilver Platform and add UART SWAP Platforms

Option #1 – Copy in Platform File Updates

Option #2 – Update and Create your own files

**Option #1 – Copy in Platform File Updates and update one AT\_CMD file**

Find and unzip “WICED-Studio\_Platform\_Updates\_To\_Swap\_UART.zip” from github.com/ArrowElectronicsESC/QuicksilverWorkshop

Copy the directory “43xxx\_Wi-Fi” folder over the existing 43xxx\_Wi-Fi folder in your installation of “WICED-Studio-6.6” (or current version of WICED). This will replace three files as described below and add three additional platform folders.

The following files/directories have been updated to support this AT Code Example

|  |  |
| --- | --- |
| FIle | Changes |
| apps/ waf/ ota2\_boollader/ ota2\_bootloader.mk | Add a wildcard after CYWxx907 based kit names to allow additional platform names  **VALID\_PLATFORMS :**= BCM943909WCD1\_3\* BCM943907WAE\_1\* BCM943907WAE2\_1\* BCM943907AEVAL1\_1 BCM943907AEVAL1F\* BCM943907AEVAL2F\* BCM943907WCD1 CYW943907WAE\* CYW943907AEVAL\*  **VALID\_PLATFORMS +**= BCM943907WCD2 CYW954907AEVAL1F\* BCM943903PS Quicksilver\_EVL\* MurataType1GC MurataType1PS |
| apps/ waf/ ota2\_failsafe/ ota2\_failsafe.mk | Add a wildcard after CYWxx907 based kit names to allow additional platform names  **VALID\_PLATFORMS :**= BCM943909WCD1\_3\* BCM943907WAE\_1\* BCM943907WAE2\_1\* BCM943907AEVAL1\_1 BCM943907AEVAL1F\* BCM943907AEVAL2F\* BCM943907WCD1 CYW943907WAE\* CYW943907AEVAL\*  **VALID\_PLATFORMS +**= BCM943907WCD2 CYW954907AEVAL1F\* CYW9MCU7X9N364 BCM943903PS Quicksilver\_EVL\* MurataType1GC MurataType1PS |
| apps/snip/ota2\_extract.mk | Add a wildcard after CYWxx907 based kit names to allow additional platform names  **VALID\_PLATFORMS :**= BCM943909WCD1\_3\* BCM943907WAE\_1\* BCM943907WAE2\_1\* BCM943907AEVAL1\_1 BCM943907AEVAL1F\* BCM943907AEVAL2F\* BCM943907WCD1 CYW943907WAE\* CYW943907AEVAL\*  **VALID\_PLATFORMS +**= BCM943907WCD2 CYW954907AEVAL1F\* CYW9MCU7X9N364 BCM943903PS Quicksilver\_EVL\* MurataType1GC MurataType1PS |
| platforms/ Quicksilver\_EVL/ ota2\_image\_defines.mk  Note: This change fixes an issue with the Quicksilver platform. | Changed the starting address for the Over The Air (OTA) Image  # File system (start on 4k boundary) <-- default  #ifeq (1, $(SECURE\_SFLASH))  #OTA2\_IMAGE\_CURR\_FS\_AREA\_BASE := 0x00264000 # 852k 0x000D5000  #else  #OTA2\_IMAGE\_CURR\_FS\_AREA\_BASE := 0x00262000 # 860k 0x000D7000  #endif  # File system (start on 4k boundary) <-- changed to this by Carson 2/3/22  **ifeq** (1, $(SECURE\_SFLASH))  **OTA2\_IMAGE\_CURR\_FS\_AREA\_BASE :**= 0x00265000 # 852k 0x000D5000  **else**  **OTA2\_IMAGE\_CURR\_FS\_AREA\_BASE :**= 0x00263000 # 860k 0x000D7000  **endif** |
| platforms/ Quicksilver\_EVL\_UART\_SWAP | Identical to Quicksilver\_EVL platform with one modification made to the platform.h file to set UART2 for STDIO |
| platforms/ CYW943907AEVAL1F\_UART\_SWAP | Identical to CYW943907AEVAL1F platform with one modification made to the platform.h file to set UART2 for STDIO |
| platforms/ CYW954907AEVAL1F\_UART\_SWAP | Identical to CYW954907AEVAL1F platform with one modification made to the platform.h file to set UART2 for STDIO |

Three Platform folders are added to the three existing platform folders for CYWxx907 based kits.

Note: The platform folders that support a UART Connection thru USB are modified copies of the corresponding default platforms that were provided with your WICED installation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kit | UART Connection | Platform File | UART TX | UART RX | UART CTS | UART RTR |
| Quicksilver | Thru USB | Quicksilver\_EVL\_UART\_SWAP | N/A | N/A | N/A | N/A |
| Quicksilver | Direct to J6 | Quicksilver\_EVL |  |  |  |  |
| CYW943907 | Thru USB | CYW943907AEVAL1F\_UART\_SWAP | N/A | N/A | N/A | N/A |
| CYW943907 |  | CYW943907AEVAL1F |  |  |  |  |
| CYW954907 | Thru USB | CYW954907AEVAL1F\_UART\_SWAP | N/A | N/A | N/A | N/A |
| CYW954907 |  | CYW954907AEVAL1F |  |  |  |  |

**Update two files**

a) Update file: 43xxx\_Wi-Fi/apps/test/at\_cmd/os\_wrapper\_wiced.c

Disable 3M Baud rate and Flow Control to use the UART through the USB option or a direct UART if not using CTS and RTR

Around line #52

Change from:

**#define** UART\_BAUDRATE\_3M\_ENABLE (1)

Change to:

**#define** UART\_BAUDRATE\_3M\_ENABLE (0) /\* Disable 3Mbit/second Baud Rate \*/

==========================================

Around line #116

Disable Flow Control

Replace “flow\_control=FLOW\_CONTROL\_CTS\_RTS ,” with “flow\_control = FLOW\_CONTROL\_DISABLED,”

**#if** defined(PLATFORM\_1LD)

.flow\_control = FLOW\_CONTROL\_DISABLED,

**#else**

// .flow\_control = FLOW\_CONTROL\_CTS\_RTS,

.flow\_control = *FLOW\_CONTROL\_DISABLED*,

**#endif**

**#endif**

b) Update file: 43xxx\_Wi-Fi/apps/test/at\_cmd/wiced\_init.c

Change to UART 1 from UART 2

Around line #89

**#if** defined(PLATFORM\_1LD)

wiced\_uart\_t wiced\_uart = WICED\_UART\_1;

**#else**

// wiced\_uart\_t wiced\_uart = WICED\_UART\_2; /\* UART\_2 is on J6 \*/

wiced\_uart\_t wiced\_uart = WICED\_UART\_1; /\* UART\_1 is Virtual Com Port via USB Debug Port J1 \*/

**#endif**

**#endif**

**Option #2 – Update files yourself.**

1. Within WICED, modify the “platform.h” file within the “platforms/Quicksilver\_EVL” files by changing the targeted UART on the Quicksilver\_EVL kit from UART\_1 to UART\_2
   1. Change STDIO\_UART from WICED\_UART\_1 to WICED\_UART\_2.

NOTE: If you create a new platform file for the modification above, you’ll need to also modify the SNIP OTA2\_EXTRACT to accept the new platform name as a VALID PLATFORM

1. Within WICED, modify the “os\_wrapper\_wiced.c” file within the “apps/test/at\_cmd” folder. Disable flow control and change baudrate
   1. Change UART\_BAUDRATE\_3M\_ENABLE from 1 to 0.
   2. Change uart\_console\_config\_atcmd.flow\_control to FLOW\_CONTROL\_DISABLED.
2. Within WICED, modify the “wiced\_init.c” file within the “apps/test/at\_cmd” folder. Exchange STDIO UART and AT command UART
   1. Change variable wiced\_uart in wiced\_init.c from WICED\_UART\_2 to WICED\_UART\_1.

Below is differences for three files modified: platform.h, os\_wrapper\_wiced.c, wiced\_init.c

|  |
| --- |
| --- platforms/Quicksilver\_EVL/platform.h      Mon Aug 23 18:56:54 2021  +++ platforms/Quicksilver\_EVL/platform.h      Mon Aug 23 18:57:27 2021  @@ -52,7 +52,8 @@  #ifdef PLATFORM\_WL\_UART\_ENABLED  #define STDIO\_UART  ( WICED\_UART\_3 )  #else  -#define STDIO\_UART  ( WICED\_UART\_1 )  +#define STDIO\_UART  ( WICED\_UART\_2 )  #endif /\* PLATFORM\_WL\_UART\_ENABLED \*/   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --- apps/test/at\_cmd/os\_wrapper\_wiced.c       Fri Aug 20 11:23:44 2021  +++ apps/test/at\_cmd/os\_wrapper\_wiced.c       Mon Aug 23 18:54:19 2021  @@ -49,7 +49,7 @@  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    \*                      Macros    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -#define UART\_BAUDRATE\_3M\_ENABLE   (1)  +#define UART\_BAUDRATE\_3M\_ENABLE   (0)   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    \*                    Constants  @@ -114,7 +114,8 @@  #if defined(PLATFORM\_1LD)         .flow\_control = FLOW\_CONTROL\_DISABLED,  #else  -       .flow\_control = FLOW\_CONTROL\_CTS\_RTS,  +       .flow\_control = FLOW\_CONTROL\_DISABLED,  #endif  #endif  };  --- apps/test/at\_cmd/wiced\_init.c     Thu Jun 11 12:39:54 2020  +++ apps/test/at\_cmd/wiced\_init.c     Mon Aug 23 18:58:32 2021  @@ -89,7 +89,8 @@  #if defined(PLATFORM\_1LD)         wiced\_uart\_t wiced\_uart = WICED\_UART\_1;  #else  -       wiced\_uart\_t wiced\_uart = WICED\_UART\_2;  +       wiced\_uart\_t wiced\_uart = WICED\_UART\_1;  #endif  #endif |

## 2.4 Clean and Build Base OTA code

Expose all files in the Make Target window of WICED by clicking the “>” symbol to the left of 43xxx\_Wi-Fi

Make This Section 2.4 and Section 2.5 Clearer ---

The process of creating AT\_CMD code for the WICED module uses three steps, executed by running “Make Targets”:

1. clean – This step removes any residual information by deleting the “Build” folder and all contents
2. ota2\_extract – This step creates a base file, matched to your kit and configuration, to support Over The Air (OTA) updates.
3. at\_cmd – This step generates the code for your specific kit and configuration then loads that code into your kit.

A “Make Target” is simply a line of text containing special characters that provide instructions to a program. The format of the “Make Target” is as follows:

<application>-<Platform>-<RTOS>-<Network Stack>-<WLAN Interface Bus> <instructions>

<application> is the program to be performed or generated along with the path to that application. For example, “test.at\_cmd” will run the file “at\_cmd.mk” from the applications directory under Project Explorer, “43xxx\_Wi-Fi.apps.test.at\_cmd”. Note that “43xxx\_Wi-Fi” and “apps” are assumed.

<Platform> is a description of the kit or system the code is targeted to. Multiple platforms can be used for one kit to repurpose features of the kit. For example, if there are multiple UART ports on a kit, one platform may define a specific UART port A as UART 1, while a second platform may define a different UART port B as UART 1. This allows a single program to be uniquely on specified kits.

<RTOS> is NOT applicable here and left out

<Network\_Stack> is NOT applicable here and left out

<MCU-WLAN Interface Bus> is NOT applicable here and left out

<instructions> tell what to do with the application. The list of all instructions that can be provided in the Make Target line are listed in <WICED-SDK installation directory>/ 43xxx\_Wi-Fi/Makefile

The first Make Target, “clean”, is common to any kit and configuration.

Please create “Make Targets” for your kit and the configurations that you will use.

Add up to Six OTA Creation Make Target lines for the configurations you plan to support:

|  |  |  |
| --- | --- | --- |
| Kit | UART Connection | OTA File Creation Make Target line |
| Quicksilver | Thru USB | snip.ota2\_extract-Quicksilver\_EVL\_UART\_SWAP |
| Quicksilver | Direct to J6 | snip.ota2\_extract-Quicksilver\_EVL |
| CYW943907 | Thru USB | snip.ota2\_extract-CYW943907AEVAL1F\_UART\_SWAP |
| CYW943907 |  | snip.ota2\_extract-CYW943907AEVAL1F |
| CYW954907 | Thru USB | snip.ota2\_extract-CYW954907AEVAL1F\_UART\_SWAP |
| CYW954907 |  | snip.ota2\_extract-CYW954907AEVAL1F |

Add up to Six AT\_CMD code compile Make Target lines for the configurations you plan to support:

|  |  |  |
| --- | --- | --- |
| Kit | UART Connection | AT\_CMD code compile and kit program Make Target line |
| Quicksilver | Thru USB | test.at\_cmd-Quicksilver\_EVL\_UART\_SWAP ota2\_image download run |
| Quicksilver | Direct to J6 | test.at\_cmd-Quicksilver\_EVL ota2\_image download run |
| CYW943907 | Thru USB | test.at\_cmd-CYW943907AEVAL1F\_UART\_SWAP ota2\_image download run |
| CYW943907 |  | test.at\_cmd-CYW943907AEVAL1F ota2\_image download run |
| CYW954907 | Thru USB | test.at\_cmd-CYW954907AEVAL1F\_UART\_SWAP ota2\_image download run |
| CYW954907 |  | test.at\_cmd-CYW954907AEVAL1F ota2\_image download run |

Following is further explanation on the Make Targets…

**Clean**

Start with a clean slate by running a “clean”.

Double click the “clean” Make Target.

Note: The clean will delete the “build” folder and all its contents. The build folder shows up in the “Project Explorer” section of your WICED with output files from each Make. Cleaning the build folder ensures the subsequent actions create new files.

Graphical user interface, text, application

Description automatically generated

Create a Make Target in the “Make Target” window of WICED by expanding the 43xxx\_Wi-Fi directory then right-clicking one of the existing Make Targets.

A computer screen capture

Description automatically generated with low confidence

* Click “New…” to create a new Make Target
* Note: What ever was highlighted in the Make Target window will show up in the “Create Make Target” window as the “Target name”. Replace the “Target name” with one of the “Make Target” lines that you intend to add.

Graphical user interface, text, application, email

Description automatically generated

* Into the “Target name:” field, type the following exactly with NO spaces before or after

**snip.ota2\_extract-Quicksilver\_EVL**

* + The AT\_CMD example utilizes an Over The Air (OTA) update platform, which requires a boot code image, which the extract command creates.
* Click the [ OK ] button
* Double click on the newly created Make Target, “snip.ota2\_extract-Quicksilver\_EVL” in the Make Target window.
* Progress will be displayed in the Console window.
* “Build Finished” indicates building of the image completed successfully

Following are the OTA Creation Make File formats for each configuration

|  |  |  |
| --- | --- | --- |
| Kit | UART Connection | OTA File Creation Make File |
| Quicksilver | Thru USB | snip.ota2\_extract-Quicksilver\_EVL\_UART\_SWAP |
| Quicksilver | Direct to J6 | snip.ota2\_extract-Quicksilver\_EVL |
| CYW943907 | Thru USB | snip.ota2\_extract-CYW943907AEVAL1F\_UART\_SWAP |
| CYW943907 |  | snip.ota2\_extract-CYW943907AEVAL1F |
| CYW954907 | Thru USB | snip.ota2\_extract-CYW954907AEVAL1F\_UART\_SWAP |
| CYW954907 |  | snip.ota2\_extract-CYW954907AEVAL1F |

Recommend creating those Make Files that you might use so they exist in your Make Target folder. When you’re ready to program a kit, it’s best to re-run the sequence by clicking on the set corresponding to your kit and UART Connection in sequence: Clean / snip.ota2… / test.at\_cmd-…

## 2.5 Compile and load AT\_CMD program

* Into the “Target name:” field, type the following exactly with NO spaces before or after

**test.at\_cmd-Quicksilver\_EVL ota2\_image download run**

* Click the [ OK ] button
* Double click on the newly created Make Target, “test.at\_cmd-Quicksilver\_EVL download run” in the Make Target window.
* Build progress will be displayed in the Console window.
* “Build Finished” indicates building and downloading of the application completed successfully

Note: If the build fails, indicated by light red or orange shaded lines and the word “error” or “failed”, see the troubleshooting section of these instructions.

Following are the

|  |  |  |
| --- | --- | --- |
| Kit | UART Connection | AT\_CMD code compile and kit program Make File |
| Quicksilver | Thru USB | test.at\_cmd-Quicksilver\_EVL\_UART\_SWAP ota2\_image download run |
| Quicksilver | Direct to J6 | test.at\_cmd-Quicksilver\_EVL ota2\_image download run |
| CYW943907 | Thru USB | test.at\_cmd-CYW943907AEVAL1F\_UART\_SWAP ota2\_image download run |
| CYW943907 |  | test.at\_cmd-CYW943907AEVAL1F ota2\_image download run |
| CYW954907 | Thru USB | test.at\_cmd-CYW954907AEVAL1F\_UART\_SWAP ota2\_image download run |
| CYW954907 |  | test.at\_cmd-CYW954907AEVAL1F ota2\_image download run |

Recommend creating those Make Files that you might use so they exist in your Make Target folder. When you’re ready to program a kit, it’s best to re-run the sequence by clicking on the set corresponding to your kit and UART Connection in sequence: Clean / snip.ota2… / test.at\_cmd-…

## 3.1 Setup virtual uart connection:

A screenshot of a computer

Description automatically generated with medium confidence

When using a UART Connection thru USB, your windows operating system will need a Virtual UART port.

To evaluate AT command software without a UART interface, the USB port of Quicksilver through the FTDI USB to Serial chip, can be used by following these steps:

Enable Virtual Comm Port (VCP) for the USB connection on your PC

In your PC’s Windows Search menu; Open Device Manager 🡪 Expand Universal Serial Bus controllers à Right Click on WICED USB Serial Port B then select Properties (or double click WICED USB Serial Port B) à Select tab Details (or Advanced) then check “Load VCP”

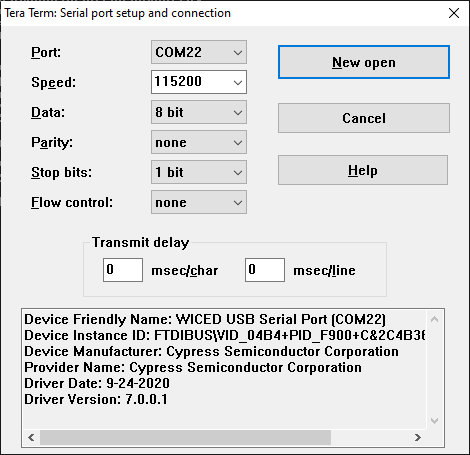
## 3.2 SETUP terminal emulator:

Many Terminal programs exist including TeraTerm, Putty, and YAT

The following example uses TeraTerm – any terminal program will work.

Note: YAT offers an output buffer, which lets you fix typos before submitting a command. With TeraTerm, I copy the command from the document then right click it into the terminal window.

* Open a terminal program.
* Create a Serial Port Connection: 115200 baud, 8-bit, no parity, 1 stop bit, no flow control



* Enable CR+LF in your terminal program connected to the USB to UART
  + If using TeraTerm, click the “Setup” dropdown menu then select “Terminal”

Graphical user interface, application

Description automatically generated

## 3.3 TEST an at command:

* In the terminal window type

**AT+WSCAN** followed by Enter

* You should see text flow in both terminal windows similar to what is pictured below.

Text

Description automatically generated