

USING NOTEPAD++ AND COLOR SYNTAX HIGHLIGHTING

Application Note

v1.1

INTRODUCTION

If you intend to use Notepad++ to edit and develop smartBASIC source code, the goal of this document is to explain how to configure Notepad++ so that smartBASIC code is displayed with color syntax highlighting.

REQUIREMENTS

- Windows PC (XP or newer) with Notepad++ version 6.4.5 or newer installed. You can install it by downloading it from <http://notepad-plus-plus.org>.

OVERVIEW

When editing and developing source code for programs most modern text editors can display the code with color syntax highlighting.

The color scheme is based on the underlying syntax of the language for which the code was written. smartBASIC has a BASIC flavour and in addition, for each implementation, there are platform specific extension functions.

Notepad++ can be configured directly from the software by using the built in syntax editor which defines the color syntax highlighting for a specifically named programming language. As of September 2013, all BL600 firmware upgrade zip files from Laird contain a file called **smartBASIC(Notepad++).xml** which can be used to configure Notepad++ to display smartBASIC source code.

A typical look and feel of smartBASIC source fragment in Notepad++ (when configured correctly) is shown in Figure 1. For example, **UartCloseEx()** is an extension function displayed in a different color from the token **function** which is part of the core language.

```

57 '*****
58 '// Handler definitions
59 '*****
60
61 '====
62 '// Uart Inactivity timer handler
63 '====
64 function handlerUartTimer() as integer
65     dim rc
66     '//Close the uart, and set up TX/RX/RTS lines as gpio and for a hi-lo transition
67     '//on the RX line to be detected
68     if UartCloseEx(1) == 0 then
69         rc=GpioSetFunc(21,2,1)    '//TX - set high on default
70         rc=GpioSetFunc(23,2,0)    '//RTS - set low by default
71         rc=GpioSetFunc(22,1,2)    '//RX - Pull high input & irq on hi2lo transition
72         rc=GpioAssignEvent(UART_GPIO_ASSIGN_CHANNEL,22,1)
73         if rc != 0 then
74             print "\nGpioAssignEvent() Failed"
75         endif
76     endif
77 endfunc 1
78

```

Figure 1: Standard smartBASIC source code with color syntax highlighting

SMARTBASIC NOTEPAD++ SYNTAX SETUP

To configure Notepad++ for color syntax highlighting for the first time perform the following actions:

- If necessary, install Notepad++ (version 6.4.5 or newer).

Using Notepad++ and Color Syntax Highlighting

Application Note

- From the BL600 firmware upgrade zip file copy the file smartBASIC(Notepad++.xml) to the desktop (it will be deleted in the final step).
- Start Notepad++.
- If Notepad++ has already been configured for *smartBASIC* color syntax highlighting, then *smartBASIC* must be removed. To do this, follow these steps:
 - Select **Language > Define your language**.
 - In the User Language dropdown menu, select *smartBASIC*.
 - Click **Remove**.
 - Close the User Defined Language dialog box.
- Close Notepad++.
- Start Notepad++.
- Close all current (non-empty) Notepad++ documents.
- From the top menu, select **Language > Define your language** to open the User Defined Language vW.X.Y.Z window.
- Click **Import**.
- From the new popup screen, find and select the smartBASIC(Notepad ++).xml file in the folder saved in step 2.
- Click **Open**.
- Close the remaining popup screen and then close Notepad++.
- Reopen Notepad++.
- From the top menu, select **Language > smartBASIC**.
You can now delete the "smartBASIC (Notepad++.xml)" file from your desktop

Loading a .sb or a .sblib file from the sample apps folder in the firmware zip file should now result in the source code being displayed with color syntax highlighting.

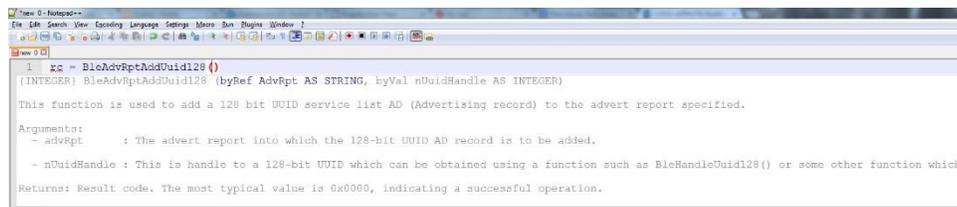
SMARTBASIC NOTEPAD++ SYNTAX EDITING

- Start Notepad++.
- From the top select **Language > Define your language**.
- From the new window, select **User Language > smartBASIC**.

You are now able to edit your syntax as needed.

MAKING NOTEPAD++ POPUP SMARTBASIC FUNCTION PARAMETERS.

You can also make Notepad++ popup with smartBASIC function parameters as you type giving you valuable hints. This means that when you type in a function when writing a smartBASIC application you'll get a popup like as shown below. So in most cases you wouldn't need to go to the user manual.



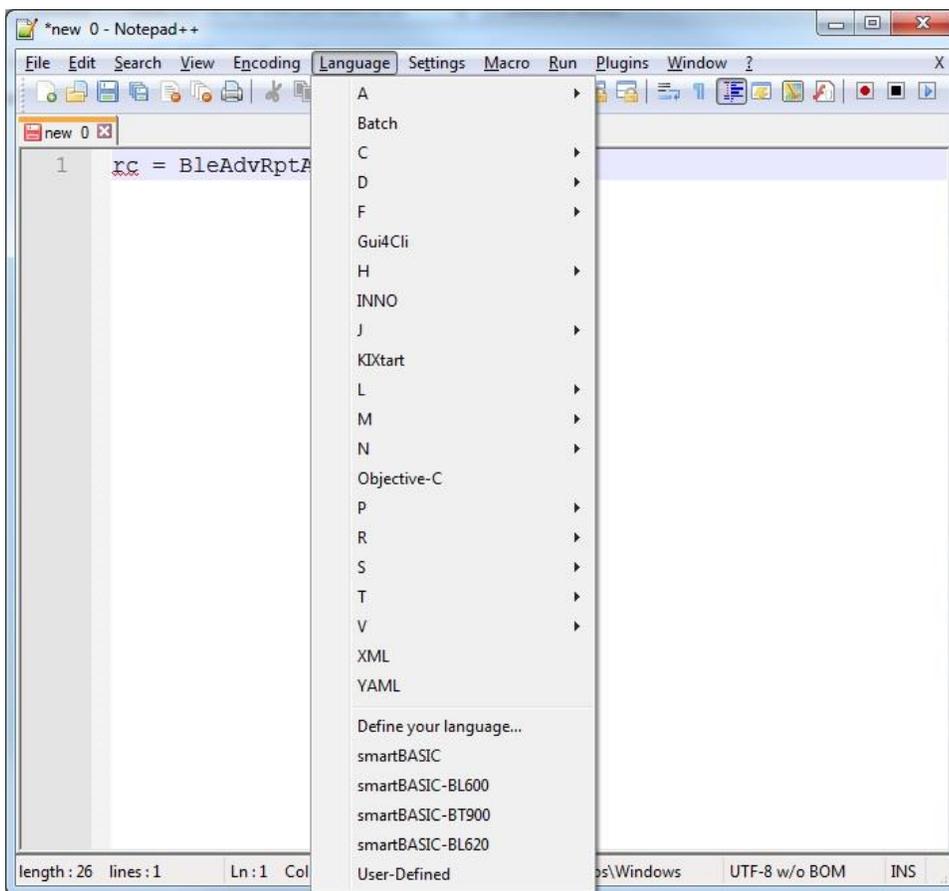
The following steps will guide you through steps required for installing relevant files for the BL600, BT900 and BL620.

In order to use this you must have Colour Syntax Highlighting set up. If you haven't, the instructions for setting that up are attached.

Using Notepad++ and Color Syntax Highlighting

Application Note

1. Next, you need to put the files in “Function Param Hint Files” zip in C:\Program Files (x86)\Notepad++\plugins\APIs
2. In notepad++ go to Language -> Define your language and select smartBASIC from the dropdown menu at the top.
3. Click Save As and name it smartBASIC-BL600.
4. Repeat steps 3 and 4, naming the language as smartBASIC-BT900 and smartBASIC-BL620. (This is because the function parameter files need to have the same filename as the “User Defined Language” in notepad++).
5. Close the User Defined Language dialog and go to Settings -> Preferences -> Auto-Completion.
6. Check “Enable auto-completion on each input”, select “Function completion” and check “Function parameters hint on input”.
7. Close and reopen notepad++.
8. When you reopen Notepad++ you will see three new options under the Language menu. Select the appropriate language for the Laird product you are working with.



9 . Notepad++ is now ready, Happy coding!

Further Information

Further information relating to firmware and the use of UWTerminal is available from the Laird website at <http://www.lairdtech.com/products/bl600-series>.

Using Notepad++ and Color Syntax Highlighting

Application Note

REVISION HISTORY

Revision	Date	Description	Approved By
1.0	24 Sept 2013	Converted from HTML. Initial Release	Jonathan Kaye
1.1	27 Mar 2015	Reordered sections. Fixed web link.	Dave Drogowski