USING TEXTPAD AND COLOR SYNTAX HIGHLIGHTING

Application Note **v1.0**

INTRODUCTION

If you intend to use Textpad to edit and develop *smart* BASIC source code, then the goal of this document includes the following:

- To explain how to configure Textpad's color syntax highlighting for *smart* BASIC code.

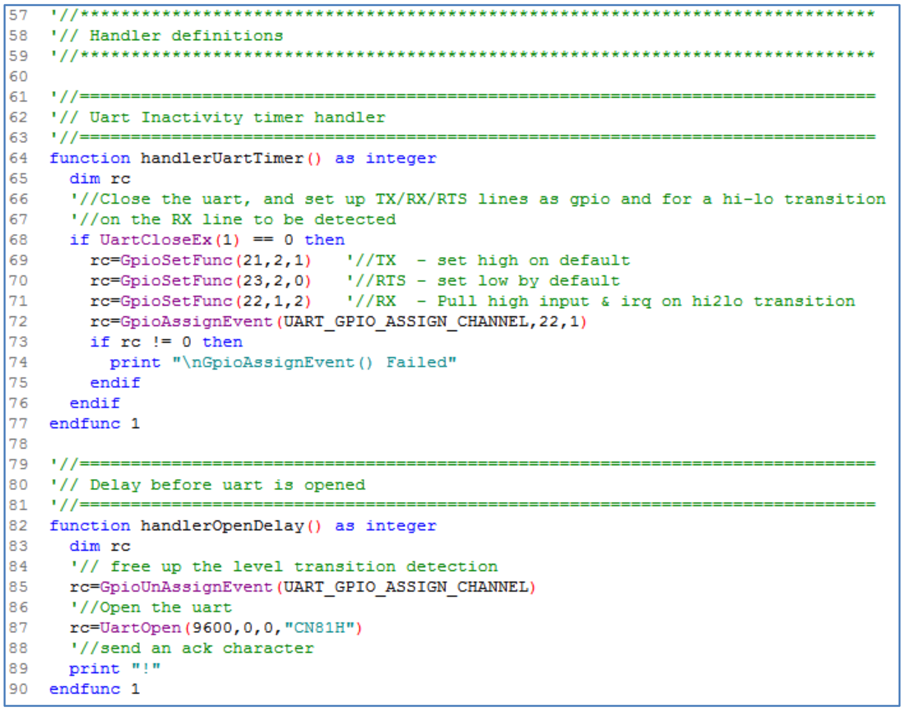
OVERVIEW

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 has been written.
smart BASIC has a BASIC flavour and in addition, for each implementation, there are platform-specific extension functions.

Textpad can be configured using a .syn file 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(textpad).syn** which can be used to configure Textpad to display *smart* BASIC source code. For example, all *smart* BASIC functions and routines are named in the .syn file so that when writing code, if the *color* of the function does not change as it is supposed to, it is a good indication that you have misspelled the function.

A typical look and feel of smartBASIC source fragment in Textpad when configured appropriately 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.



```
'//*******************************************************************************
'// Handler definitions
'//*******************************************************************************

'//===============================================================================
'// Uart Inactivity timer handler
'//===============================================================================
function handlerUartTimer() as integer
  dim rc
  '//Close the uart, and set up TX/RX/RTS lines as gpio and for a hi-lo transition
  '//on the RX line to be detected
  if UartCloseEx(1) == 0 then
    rc=GpioSetFunc(21,2,1)   '//TX  - set high on default
    rc=GpioSetFunc(23,2,0)   '//RTS - set low by default
    rc=GpioSetFunc(22,1,2)   '//RX  - Pull high input & irq on hi2lo transition
    rc=GpioAssignEvent(UART_GPIO_ASSIGN_CHANNEL,22,1)
    if rc != 0 then
      print "\nGpioAssignEvent() Failed"
    endif
  endif
endfunc 1

'//===============================================================================
'// Delay before uart is opened
'//===============================================================================
function handlerOpenDelay() as integer
  dim rc
  '// free up the level transition detection
  rc=GpioUnAssignEvent(UART_GPIO_ASSIGN_CHANNEL)
  '//Open the uart
  rc=UartOpen(9600,0,0,"CN81H")
  '//send an ack character
  print "!"
endfunc 1
```

Figure 1: Standard smartBASIC source code with color syntax highlighting

Requirements

- Windows PC (XP or newer) with Texpad version 7.0 or newer installed. You can download a trial version from the following URL: http://www.textpad.com/download/index.html#downloads.

Textpad Setup

Note: Textpad only has to be configured one time for color syntax highlighting. The configuration remains intact each time Textpad is used.

To configure Textpad for color syntax highlighting, follow these steps:

1. If necessary, install Textpad (version 7.0) and save it in the following folder: **C:\Progam Files\Textpad 7**.
2. From the BL600 firmware upgrade zip file, copy the smartBASIC(Textpad).syn file to the following folder: **C:\Progam Files\Textpad 7\system**.

   Note: You must perform this action each time a new firmware zip is released by Laird. Each newly-released firmware may contain new extension functions which will require highlighting.

3. Open Textpad and close all current Textpad documents including the empty document that Textpad initiates.
4. From the menu, select **Configure > Preferences**.
5. From the Preferences dialog box in left side list box, click **Document Classes**.
6. In the User Defined Classes list box, add the word *smartBASIC*.
7. In the far left list box, click the bordered **+** beside the Document Classes item to expand it and display a sublist.
8. From that sublist, click **smartBASIC**.
9. In the Files in class smartBASIC list box, add the following lines:
   - *.sb
   - *.sblib
10. Click the bordered **+** beside the smartBASIC to expand the far left list box and click **Syntax**.
11. Select the **Enable syntax highlighting** checkbox.
12. In the Syntax definition file dropdown box, enter or select the file **smartBASIC(Textpad).syn** from the folder where you saved it in step 2.
13. In the list box select **Colors** and then from the '**Item:**' list box select '**Keywords 1**' or any item that you wish to change the color for and do so using dropdown boxes on the right.
14. Select **OK**.
15. Close and restart Textpad.

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.

Further Information

Further information relating to *smart*BASIC and the BL600 module can be found at the Laird website at http://www.lairdtech.com/Products/Embedded-Wireless-Solutions/Bluetooth-Radio-Modules/BL600-Series