Week 7 Lab

CC2511

Task Description

You are to develop a command-line user interface for the week 5 task (controlling the LED brightness). You must extend your code to perform PWM across all three colour channels, and then provide a textual user interface that allows the user to:

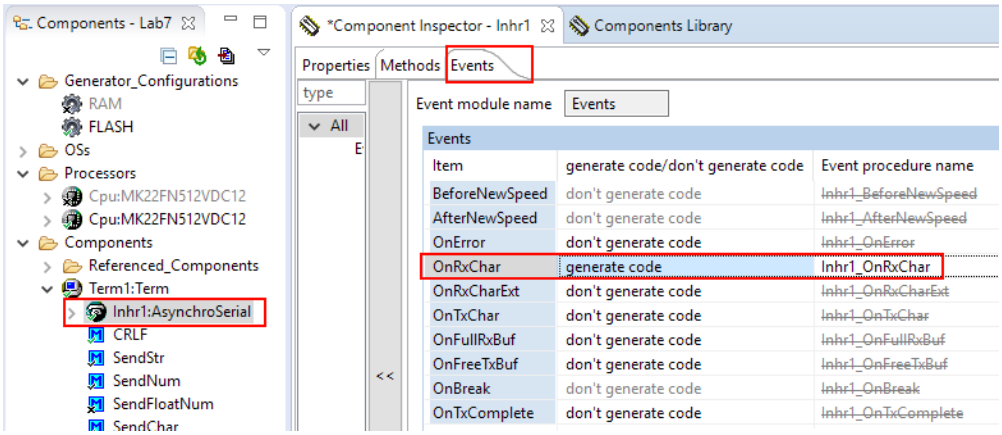
- See the current PWM ratios, and
- Type in new PWM ratios for each channel using commands like "`red 10`" (followed by the Enter key). This command would set the red PWM ratio to `10/255`.

An example implementation is shown below:

NOTE—You do not need to reproduce this visual style. A minimal implementation (suitable to pass the lab) needs only to show the three PWM ratios and allow text commands to be entered to control these.

Hints

- You can copy and paste Processor Expert components between projects.
- The **`Term`** component in Processor Expert provides facilities for moving the cursor to arbitrary coordinates, changing the colour of text, etc. The example shown above used the `SetColor` and `MoveTo` commands of `Term` to draw the yellow boxes.
- Enable the **Interrupt service/event** setting in the `AsynchroSerial` component that is inside `Term`.
- Set **`OnRxChar`** to **generate code** on the Events tab of the `AsynchroSerial` component. Once you generate Processor Expert code, a function `Inh1_OnRxChar` will be created in events.c for you. Write your interrupt handler there.



- Note: AsynchroSerial has the option to maintain its own buffers. You will want to configure a transmit buffer, so that you can transmit multiple characters without waiting for the UART. Processor Expert automatically generates a transmit handler that de-queues characters from the transmit buffer. If the buffer becomes full (because your program generated too many characters too quickly), transmit routines like `Term1_SendStr` will block until the transmit buffer has room to hold all the outgoing characters.
  - If you transmit from an interrupt service routine, you risk locking up your program! The transmit methods spin in a loop waiting from the transmit buffer to clear out, but they are only cleared by the UART interrupt. If another interrupt of the same or higher priority is already running, the transmit buffer will never empty.
- Your main program loop should use the "wait for interrupt" assembly code instruction to enter a low power sleep state.

```
__asm ("wfi");
```

- You should accumulate received characters into a string buffer. Once you detect the enter key has been pressed, add a NULL termination to your buffer and then interpret it with the **`sscanf`** and/or **`strcmp`** functions. Use **`sscanf`** if the command has arguments, and **`strcmp`** if the command is always the same static text.
- The version of `sscanf` included with Kinetis Design Studio does not support the `%hhu` modifier (which would indicate a `uint8` data type). The closest format specifier that is supported is `%hu`, which gives a `uint16` data type. Therefore the variables written to by `sscanf` should be `uint16`. If you use the wrong format specifier then `sscanf` will overwrite other adjacent variables and cause unpredictable behaviour.

Assessment

To finish this lab, demonstrate the following to your tutor:

- A working command-line user interface where arbitrary PWM ratios can be typed in for each colour channel.
- The current value of the three PWM channels displayed on the screen.
- Show your prac tutor your GitHub webpage where your source code is uploaded.

HINT—Your solution to this lab might be a useful in your assignment.

Optional Extension

- Make it possible to erase typing mistakes with backspace. By default on PuTTY, the backspace key sends `0x7f`.
- Correctly handle the case where the user types in a very long command. You will need to ensure that you do not overflow the string buffer.
- Display a visually pleasing interface like the example screenshot above.
- The sequence Ctrl-L is sometimes used in terminal programs to request that the screen be completely redrawn. This is useful if you have disconnected and reconnected PuTTY for example. Ctrl-L will send the character `'\f'`.