# AN11161

## Using the SCT in LPCXpresso, Keil, and IAR

Rev. 1 — 19 April 2012 Application note

**Document information**

| Info | Content |
|---|---|
| **Keywords** | LPC1850FET256, LPC1850FET180, LPC1850FBD208, LPC1830FET256, LPC1830FET180, LPC1830FET100, LPC1830FBD144, LPC1820FET100, LPC1820FBD144, LPC1820FBD100, LPC1810FET100, LPC1810FBD144, LPC4350FET256, LPC4350FET180, LPC4350FBD208, LPC4330FET256, LPC4330FET180, LPC4330FET100, LPC4330FBD144, LPC4320FET100, LPC4320FBD144, LPC4320FBD100, LPC4310FET100, LPC4310FBD144 SCT, LPCXpresso, Keil, IAR, LPC1800, LPC4300 |
| **Abstract** | This document details the procedure for taking SCT code generated in the LPCXpresso IDE, and incorporating it into the Keil and IAR IDEs. |

### Revision history

| Rev | Date | Description |
|---|---|---|
| 1 | 20120419 | Initial version. |

# Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

# 1. Introduction

## 1.1 SCT overview

The State Configurable Time (SCT) is a general purpose peripheral containing the following features:

- Two 16-bit counters or one 32-bit counter
- Counter(s) clocked by bus clock or selected input
- Up counter(s) or up-down counter(s)
- State variable allows sequencing across multiple counter cycles
- Events can be defined by a counter match condition, an input (or output) condition, a combination of a match and/or and input/output condition in a specified state
- Events control outputs, interrupts, and DMA requests
- Selected event(s) can limit, halt, start, or stop a counter
- Supports:
  - 8 inputs
  - 16 outputs
  - 16 match/capture registers
  - 16 events
  - 32 states

The State Configurable Timer (SCT) allows a wide variety of timing, counting, output modulation, and input capture operations.

The most basic user-programmable option is whether an SCT operates as two 16-bit counters or a unified 32-bit counter. In the two-counter case, in addition to the counter value the following operational elements are independent for each half:

- State variable
- Limit, halt, stop, and start conditions
- Values of Match/Capture registers, plus reload or capture control values
- In the two-counter case, the following operational elements are global to the SCT, but events, outputs, interrupts, and DMA requests can use match conditions from either counter:
- Clock selection
- Inputs
- Events
- Outputs
- Interrupts
- DMA requests

Fig 1 shows the SCT block diagram.

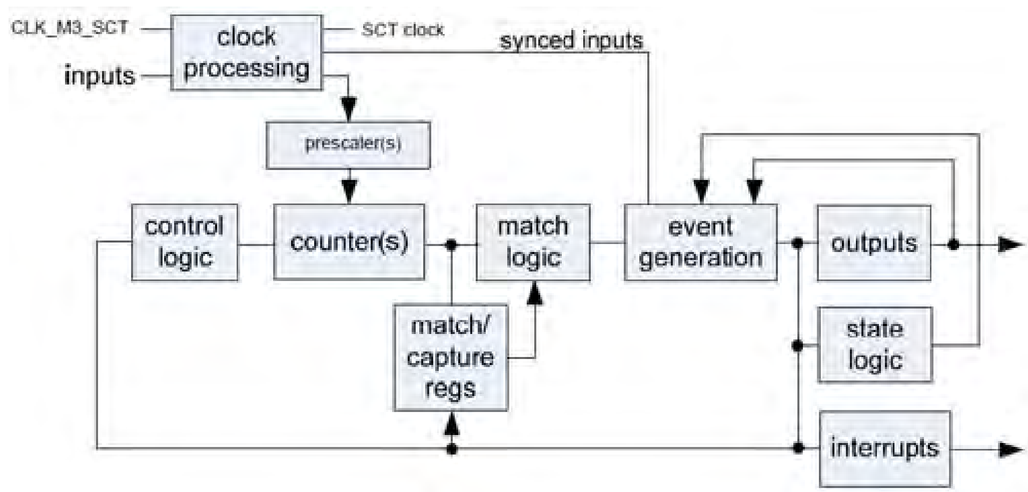



**Fig 1. SCT block diagram**

## 1.2 SCT general description

The State Configurable Timer (SCT) allows a wide variety of timing, counting, output modulation, and input capture operations.

The most basic user-programmable option is whether a SCT operates as two 16-bit counters or a unified 32-bit counter. In the two-counter case, in addition to the counter value the following operational elements are independent for each half:

- State variable
- Limit, halt, stop, and start conditions
- Values of Match/Capture registers, plus reload or capture control values

In the two-counter case, the following operational elements are global to the SCT, but events, outputs, interrupts, and DMA requests can use match conditions from either counter:

- Clock selection
- Inputs
- Events
- Outputs
- Interrupts
- DMA requests

### 1.2.1 Using the SCT in Keil and IAR tools

LPCXpresso incorporates a graphical tool for designing state machines and can generate the code required to implement the SCT. This application note will describe a process to use the code generated by LPCXpresso to add SCT functionality into Keil and IAR IDEs.

# 2. SCT project description

## 2.1 Project description

This application note assumes that the reader is familiar with the basic functionality of the SCT peripheral. For those readers needing a better understanding of the SCT peripheral, they should read the LPC18XX or LPC43XX User Manuals.

The example we are going to use is a modified version of the Blinky project found in the Getting Started with Red State manual included with LPCXpresso/Red Suite v4.1 and newer versions.

The examples included in this app note are based on LPC1850 Rev A silicon assembled on a Hitex Rev A4 evaluation board.

The state machine example toggles three RGB LEDs sequentially, with the direction determined by an input signal. The outputs of the SCT, named LED_GREEN, LED_RED, and LED_BLUE are used to drive the LEDs for visual confirmation of the state machine functionality.  An input named DOWN is used to change the direction of the output sequence, and input RESET is used to force all outputs low, turning off the LEDs. The RESET input signal is configured as active low, with an internal pull-up.

The state diagram for the project is shown in Fig 2.

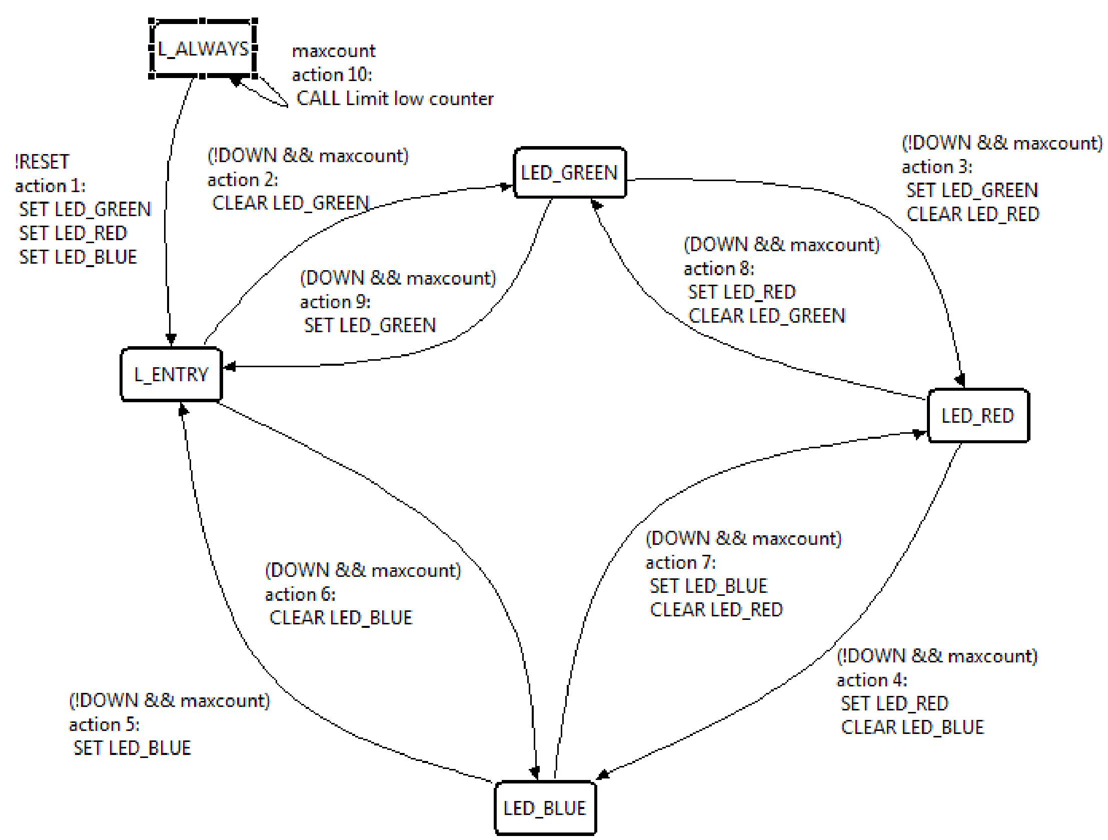



**Fig 2. RGB blinky state machine**

# 3. LPCXpresso / Red State Tool

## 3.1 Importing the example into LPCXpresso

The zip file accompanying this application note contains a folder containing the LPCXpresso project. The project is imported into the LPCXpresso IDE by selecting “Import existing projects” in the QuickStart panel, as shown in Fig 3.

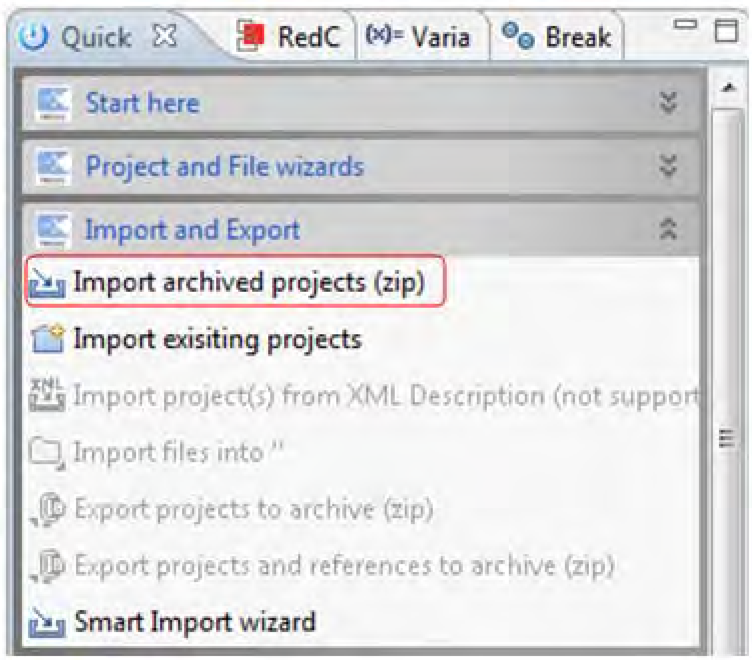



**Fig 3. Importing project into LPCXpresso**

After selecting “Import existing projects”, press the Browse button and then select the archived project named “Hitex_SCT_Blinky.zip”. This file contains two files as shown in Fig 4. After importing the zip file, press the Finish button.

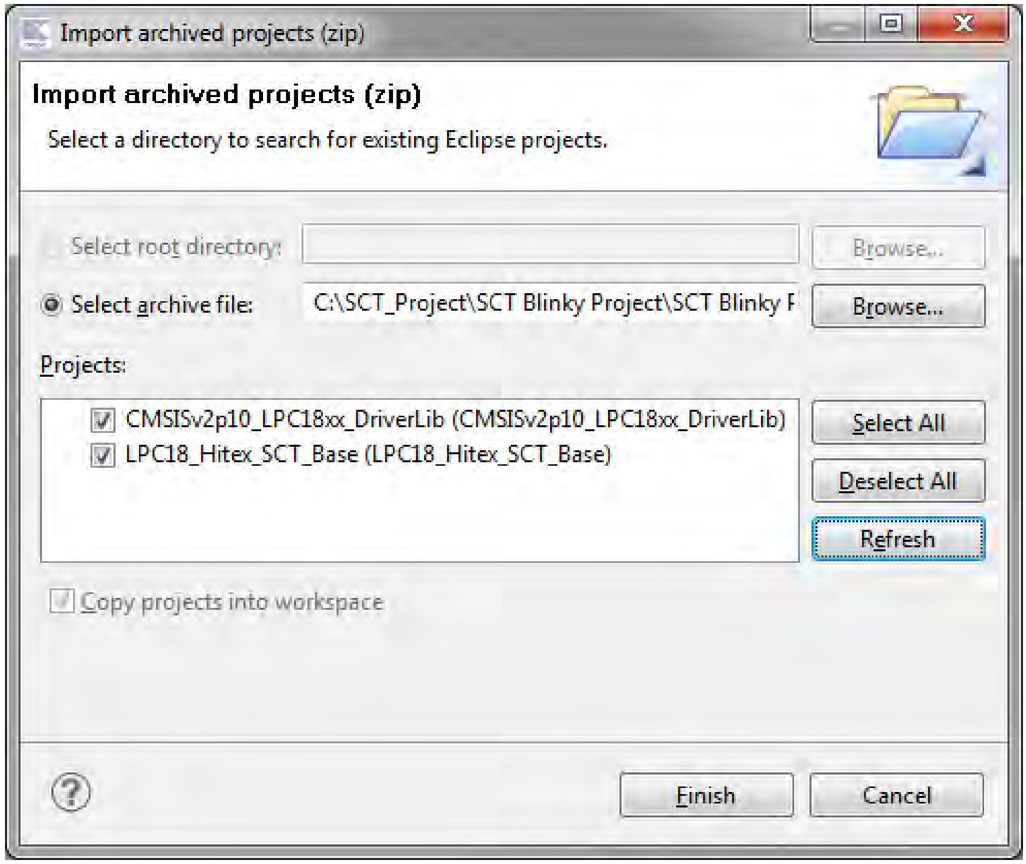
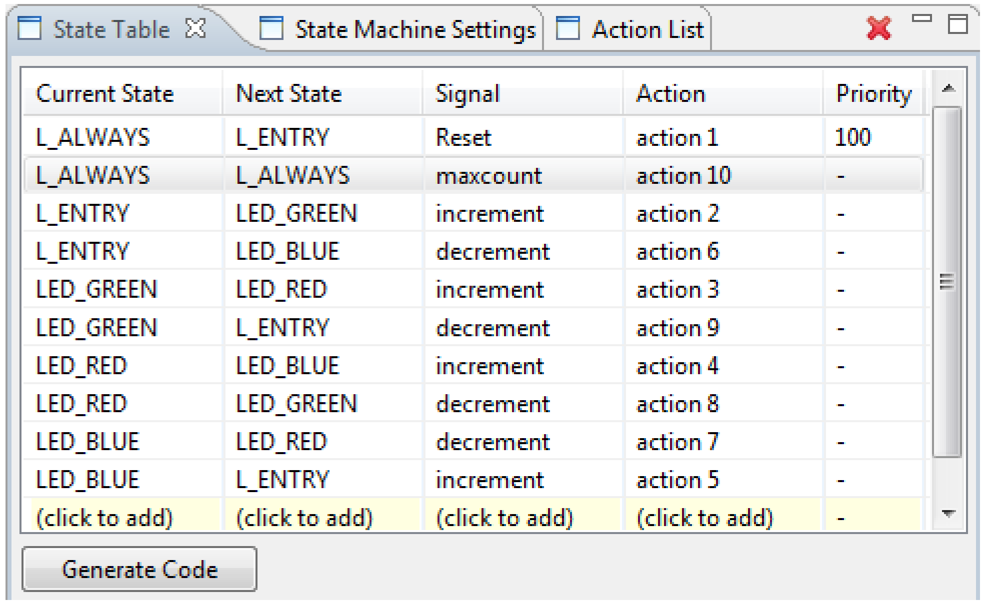



**Fig 4. Import archived projects (zip)**

## 3.2 Running the SCT project in LPCXpresso

The SCT diagram is found in the blinky.rsm file. After completing the diagram, the code is generated by pressing the Generate Code button, as shown in Fig 5. **Note that any changes made to the state machine require you to regenerate the code by pressing the Generate Code button. Building the project does not make any changes to the SCT specific files.**




| Current State | Next State | Signal | Action | Priority |
|---|---|---|---|---|
| L_ALWAYS | L_ENTRY | Reset | action 1 | 100 |
| L_ALWAYS | L_ALWAYS | maxcount | action 10 | - |
| L_ENTRY | LED_GREEN | increment | action 2 | - |
| L_ENTRY | LED_BLUE | decrement | action 6 | - |
| LED_GREEN | LED_RED | increment | action 3 | - |
| LED_GREEN | L_ENTRY | decrement | action 9 | - |
| LED_RED | LED_BLUE | increment | action 4 | - |
| LED_RED | LED_GREEN | decrement | action 8 | - |
| LED_BLUE | LED_RED | decrement | action 7 | - |
| LED_BLUE | L_ENTRY | increment | action 5 | - |
| (click to add) | (click to add) | (click to add) | (click to add) | - |

**Fig 5. Generate code button**

The sct_main.c file shows you where the inputs and outputs have been routed.

For the Hitex board:

```
* Signal   Port   Signal     Connector
* ----------------------------------------------------------------
* CTIN0    PD_13  DOWN       X19-1 or JP29-1
* CTIN1    PD_10  RESET      X19-2 or JP34-1
*
* CTOUT2   PE_6   LED_GREEN   -
* CTOUT3   PE_5   LED_RED     -
* CTOUT4   PE_8   LED_BLUE    -
* ----------------------------------------------------------------
```

The SCT input/output mapping may also be found with the LPCXpresso Red State installation. This document is titled “SCT_PortMapping.pdf”.

## 3.3 Completed Red State project

After completing the Red State example project, you will see the files shown in Fig 6. Four of the files from this project will need to be copied to your new Keil or IAR project:

- sct_main.c
- sct_fsm.c
- sct_fsm.h
- sct_user.h

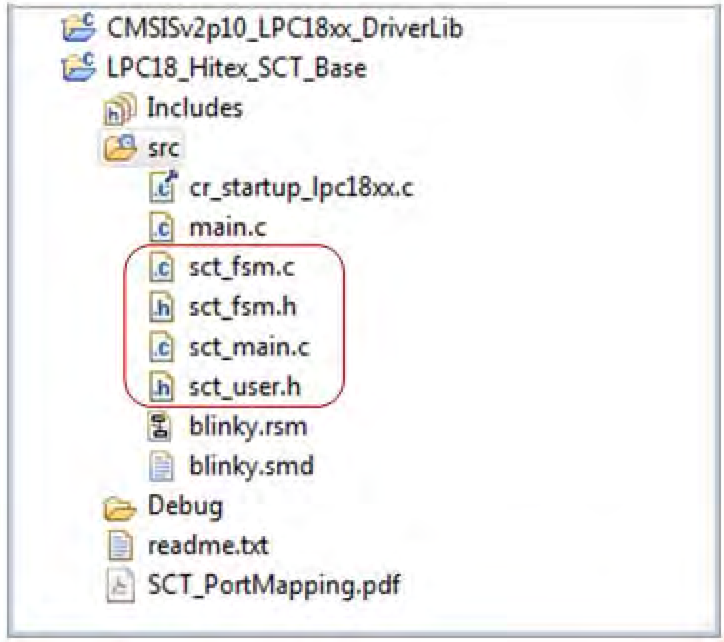



**Fig 6. Files generated by Red State**

You should always keep a copy of the LPCXpresso project, since any changes you want to make to the SCT operation will require you to generate the code in LPCXpresso again.

# 4. Completing the SCT project in the Keil IDE

## 4.1 Minimal Keil project

Fig 7 shows a minimal project that you should have in your project before you add the SCT files. This minimal project can be created using any of the example projects found in the CMSIS-Compliant Standard Peripheral Firmware Driver Library. This library can be found at: http://www.lpcware.com/file_filter/nxpf/lpc18xx

Your project may look different, depending what other modules you have included in the project.

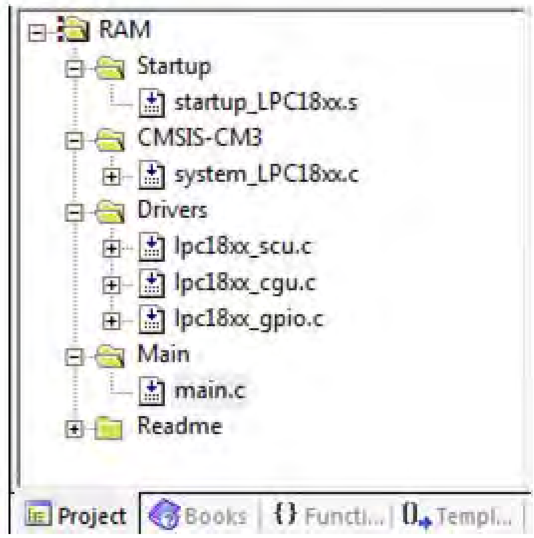



**Fig 7. Minimal Keil project**

## 4.2 Keil project with SCT files included

After copying the four files into the Keil project directory, we added a new group called SCT to the Project Explorer so that it should now look similar to Fig 8. The sct_fsm.c and sct_main.c files can be added to this group.

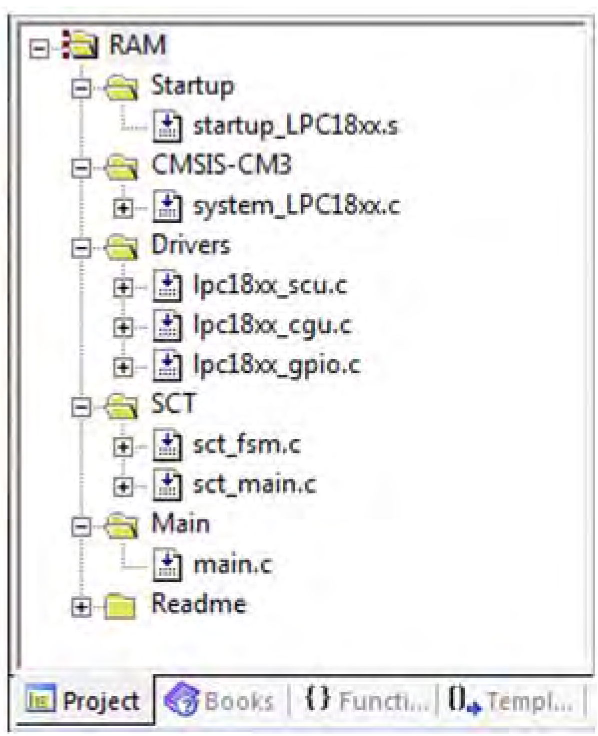



**Fig 8. Keil project with SCT files added**

### 4.3 Completing the Keil project

At this point, you should be able to build and run the project.

## 5. Completing the SCT project in the IAR IDE

### 5.1 Minimal IAR project

Fig 9 shows a minimal project that you should have in your Workspace before you add the SCT files. This minimal project can be created using any of the example projects found in the CMSIS-Compliant Standard Peripheral Firmware Driver Library. This library can be found at http://www.lpcware.com/file_filter/nxpf/lpc18xx

Your project may look different, depending what other modules you have included in the project.

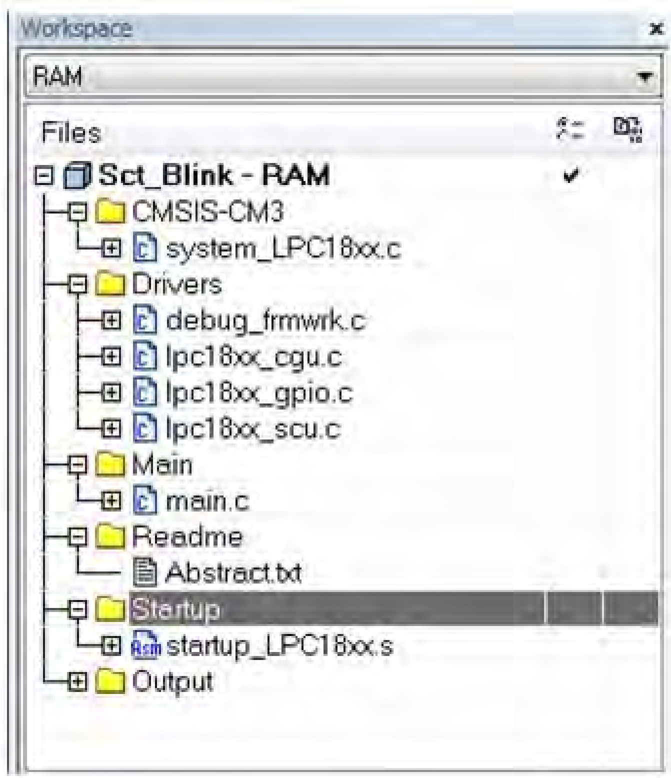



**Fig 9. Minimal IAR project**

### 5.2 IAR Project with SCT files included

After copying the four files into the IAR project directory, we added a new group called SCT to the Workspace so that it should now look similar to Fig 10.

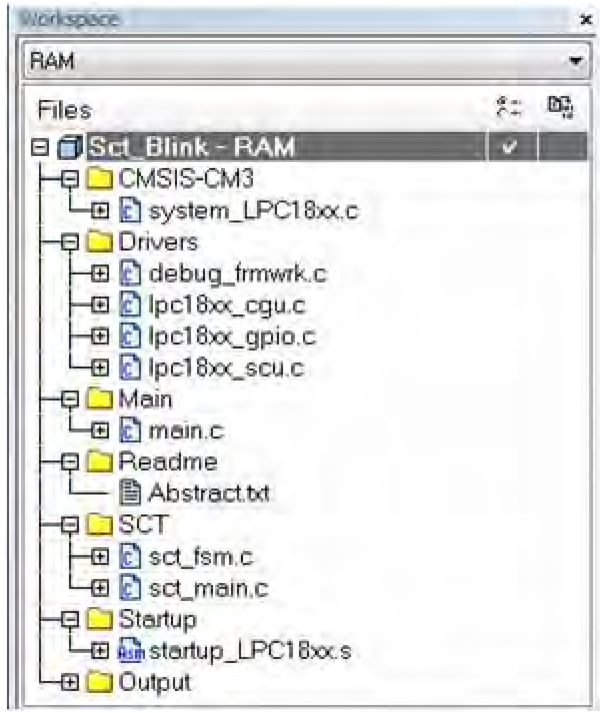



Fig 10. IAR project with SCT files added

## 5.3 Completing the IAR project

After adding the SCT files into the project, you can build and run the code.

# 6. Output waveforms

Fig 11 and Fig 12 show the output from the SCT when the example project is built and run.

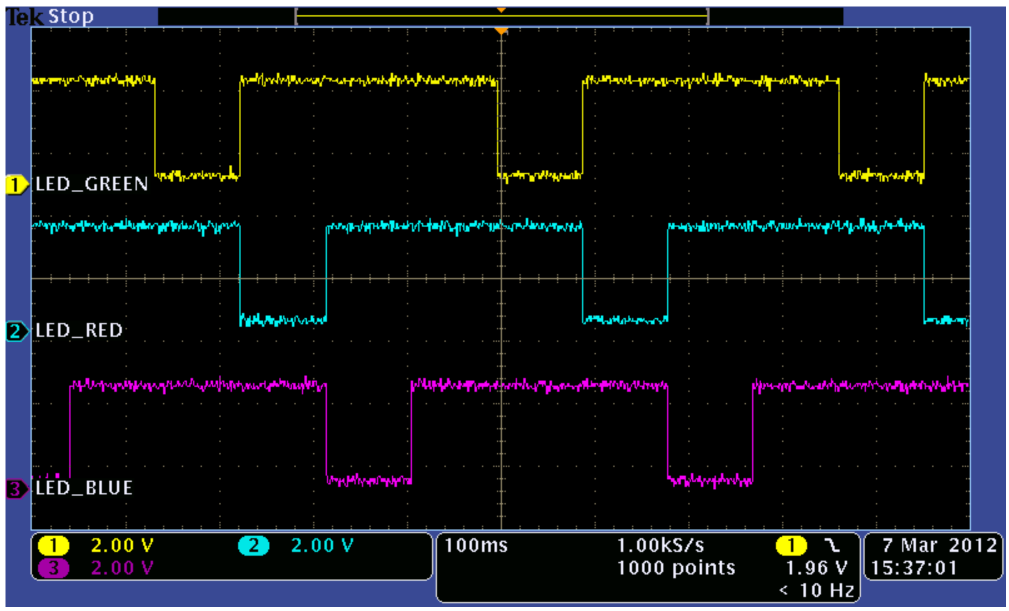



Fig 11. Output waveforms with DOWN = 0

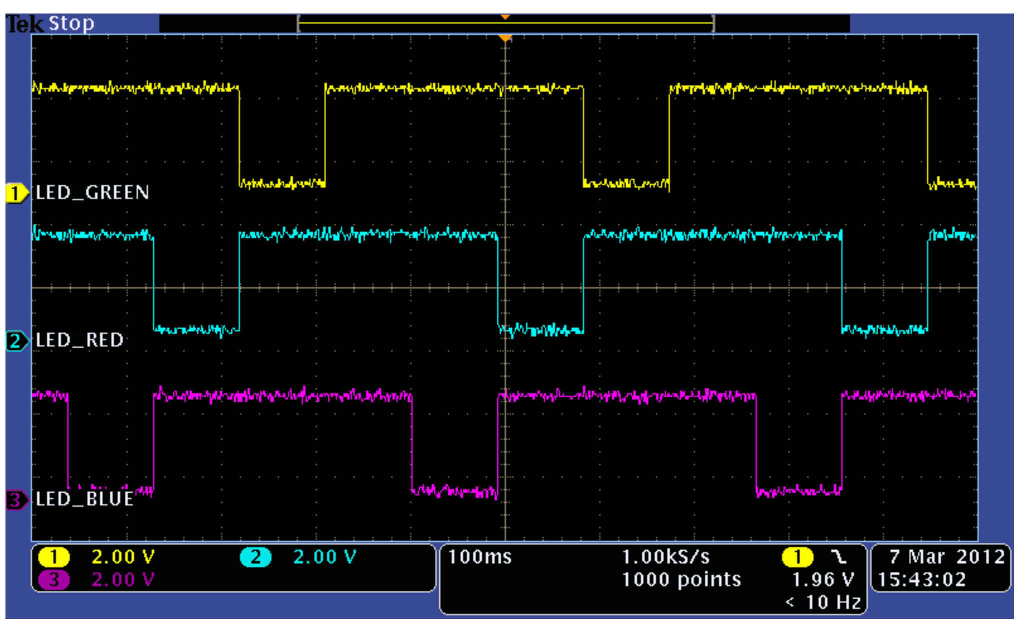



Fig 12. Output waveforms with DOWN = 1

# 7. Legal information

## 7.1 Definitions

**Draft —** The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

## 7.2 Disclaimers

**Limited warranty and liability —** Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes —** NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use —** NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications —** Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Export control —** This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Evaluation products —** This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

## 7.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

## 8. List of figures

## 9. Contents





**Date of release: 19 April 2012**
**Document identifier: AN11161**