

XE166 Family

AP16171

Using an OLED Display with the Infineon XE166 Family Products

Application Note

V1.0 2009-09

Edition 2009-09

**Published by
Infineon Technologies AG
81726 Munich, Germany**

**© 2010 Infineon Technologies AG
All Rights Reserved.**

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (www.infineon.com).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Device1

Revision History: V1.0, 2009-09

Previous Version: none

Page	Subjects (major changes since last revision)

We Listen to Your Comments

Is there any information in this document that you feel is wrong, unclear or missing?
 Your feedback will help us to continuously improve the quality of this document.
 Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com



Table of Contents

1	Introduction	5
2	Description.....	6
2.1	Which XE166 Resources Are used?	6
2.2	DAvE Settings	6
2.2.1	Universal Serial Interface Channel (USIC)1, Channel 0	7
2.2.2	Real-Time Clock (RTC).....	10
2.2.3	IO Ports	13
2.2.4	Analog Digital Converter (ADC) 1 – Channel 0.....	16
2.3	Functions	20
2.3.1	Interrupt Service Routine of ADC	20
2.3.2	Interrupt Service Routine of RTC	20
2.3.3	Modify the timer of display standby and 5ms-flashlight	20
2.3.4	Text	21
2.3.5	Basic Screen	22
2.3.6	Buttons / Switches.....	22
2.3.7	Show the Potentiometer.....	23
2.3.8	Application Example.....	23
2.3.9	Design Examples for the Infoarea.....	24

1 Introduction

This Application Note provides a simple example of using an OLED-Display with the XE166 family of products from Infineon Technologies.

All the hardware settings necessary for using the OLED-Display in this example are in a DAVE¹ project. It includes simple functions for the switches and potentiometer in combination with the display. The XE166 Real-Time Clock (RTC) is used to create standby functions, switching the display off in a similar way to a screensaver, and another LED is used to show the display state.

The actual hardware required for this project can be purchased from the Hitex e-shop (www.ehitex.de), including the UConnect XE164 USB-Stick and the UConnect OLED Extension Board.

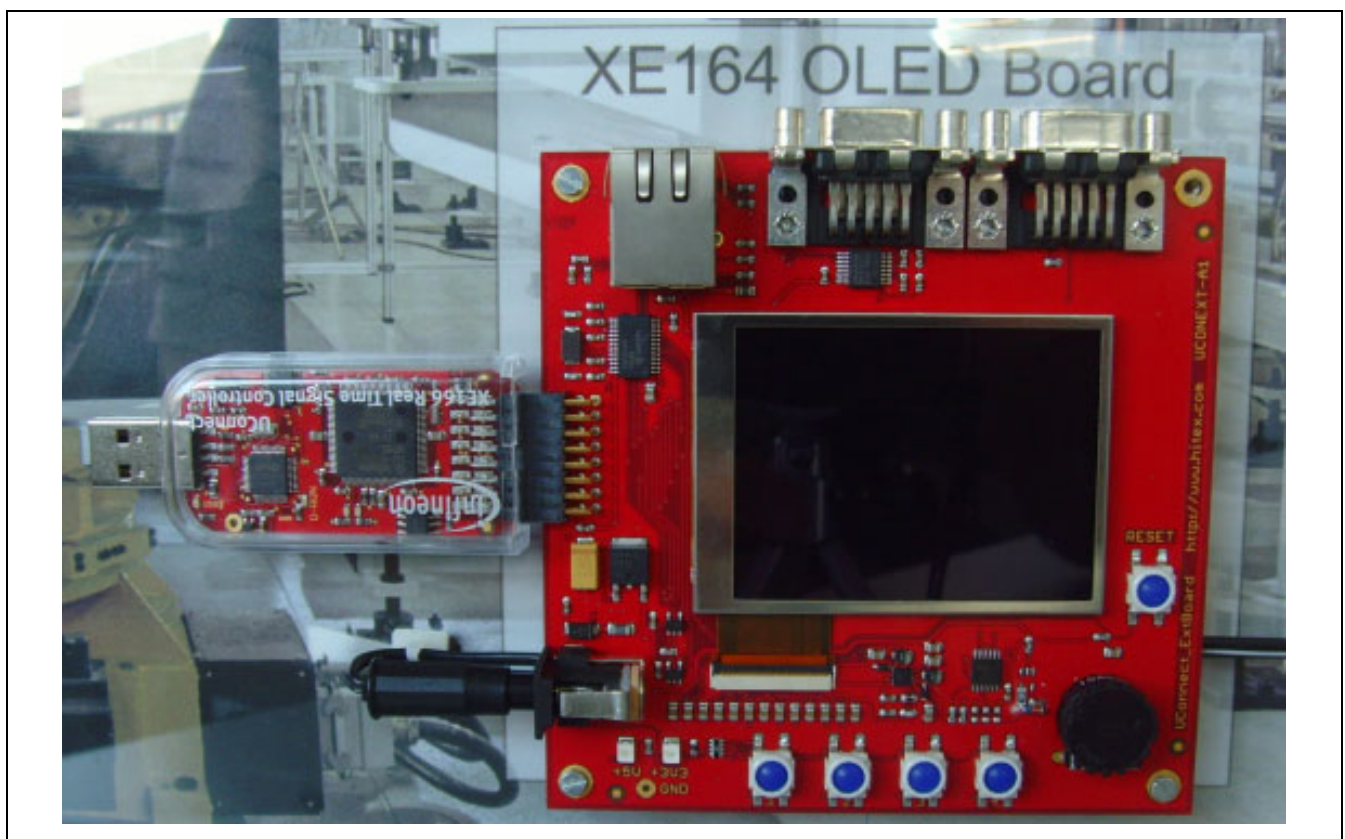


Figure 1 Hitex UConnect XE164 Combined with UCONEXT-A1

¹ DAVE stands for Digital Application Virtual Engineer and is Infineon Technologies code generator for the 8, 16 and 32 bit range of Microcontrollers. It provides initialization, configuration and driver code to ease programming for beginners and experts alike.

2 Description

The following diagram shows the simple design that we will achieve in this OLED example. The screen display has a title bar (OLED Example), four areas in contact to the switches (SW1, SW2, SW3 and SW4), a bar to give the value of the potentiometer, and an area for information:

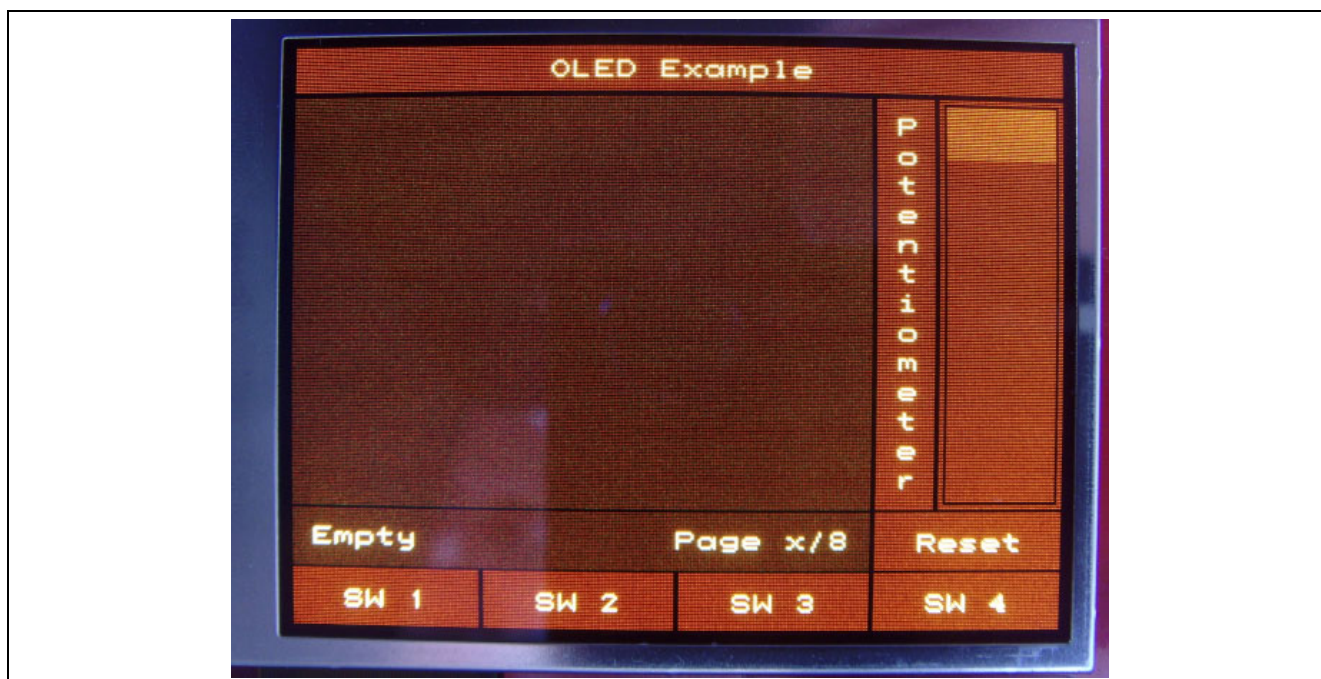


Figure 2 Example Screenshot

With a switch selected the background colour changes to indicate its selection, although no further functions are integrated.

Two different options for the potentiometer are provided, with display either as a bar graph or as a scrollbar.

A screensaver is implemented through a function to switch the display off after one minute if no events have occurred on the switches or potentiometer. In this screen saving mode the screen is black, but the user LED 0 blinks every five milliseconds. To re-activate the display one of the switches needs to be pushed. The default timeout period of one minute can be modified as required.

2.1 Which XE166 Resources Are used?

Because this is only a small application, relatively few parts of the XE166 microcontroller are actually used. Those that are used include:

- IO (for the switches, the user LEDs, and those that are required for other peripherals such as USIC)
- Real-Time Clock (RTC)
- Analog Digital Converter (ADC) 0
- Universal Serial Interface Channel (USIC) 1 - Channel 0
- Several Interrupts relating to the items listed above

2.2 DAVE Settings

Note that this document only gives descriptions and screenshots for those DAVE windows that actually require settings to be changed.

2.2.1 Universal Serial Interface Channel (USIC)1, Channel 0

The Universal Serial Interface Channel (USIC) 1 must be configured first, because the OLED is connected via SPI (Serial Peripheral Interface). Use the information in the screenshots which follow to set the USIC1 module.

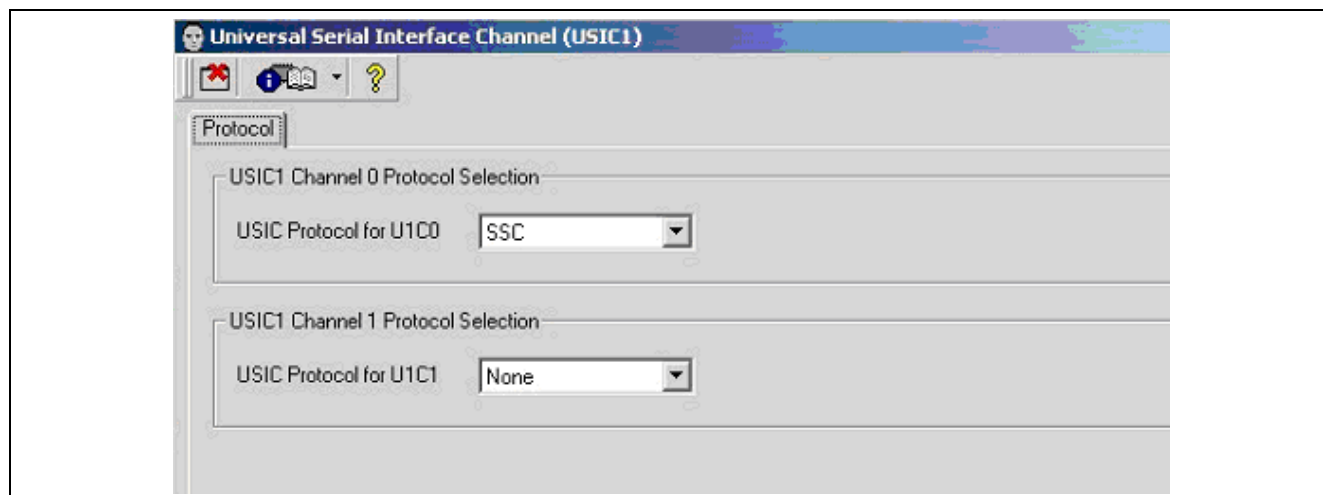


Figure 3 General USIC1 Settings

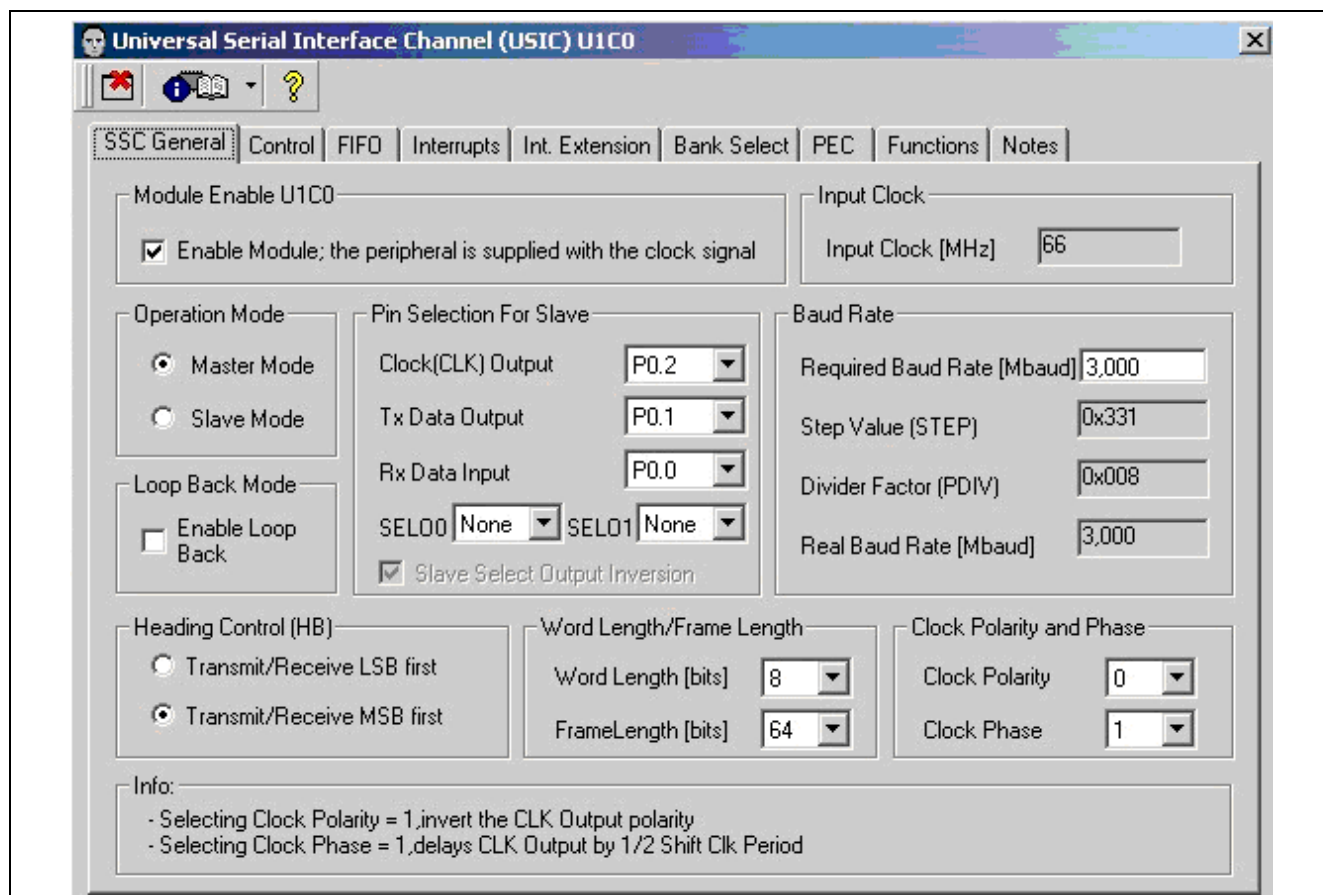


Figure 4 U1C0 Synchronous Serial Channel (SSC) Settings

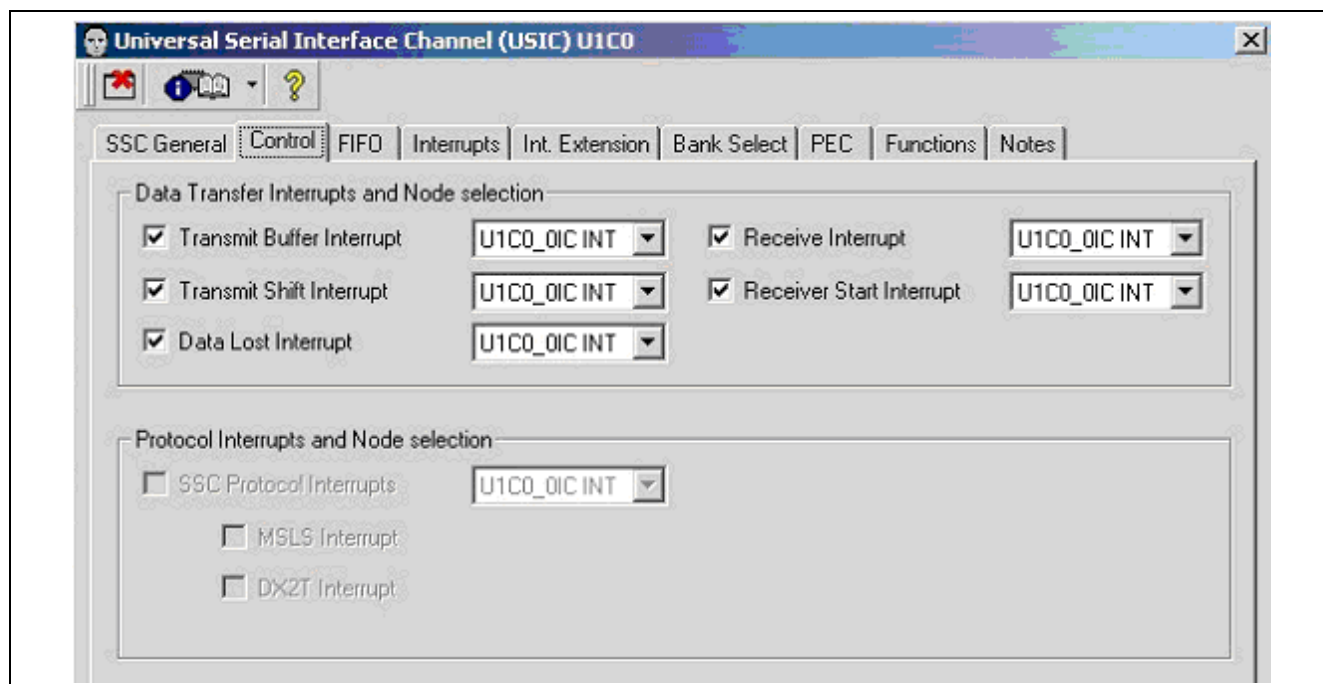


Figure 5 U1C0 Control Settings

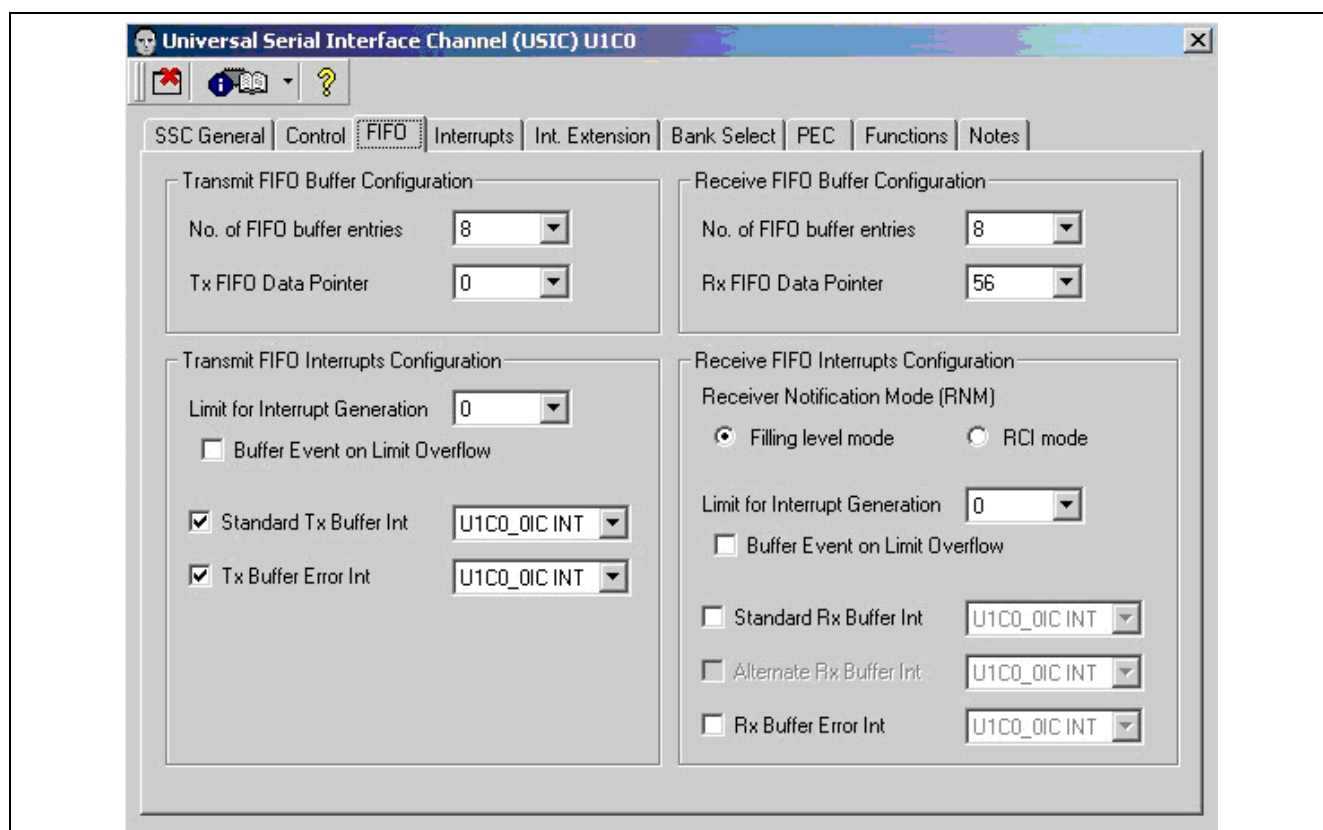


Figure 6 U1C0 FIFO Settings

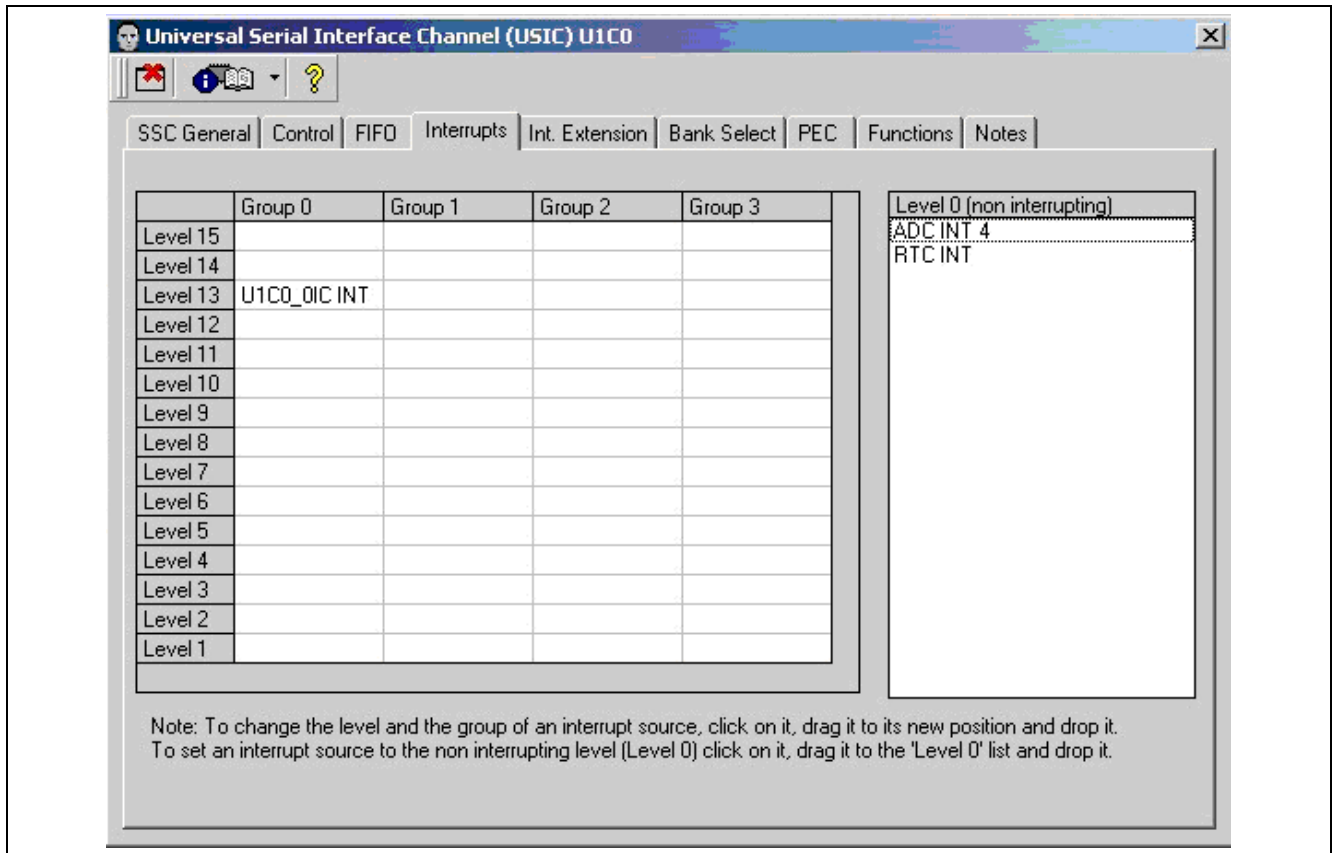


Figure 7 U1C0 Interrupt Settings

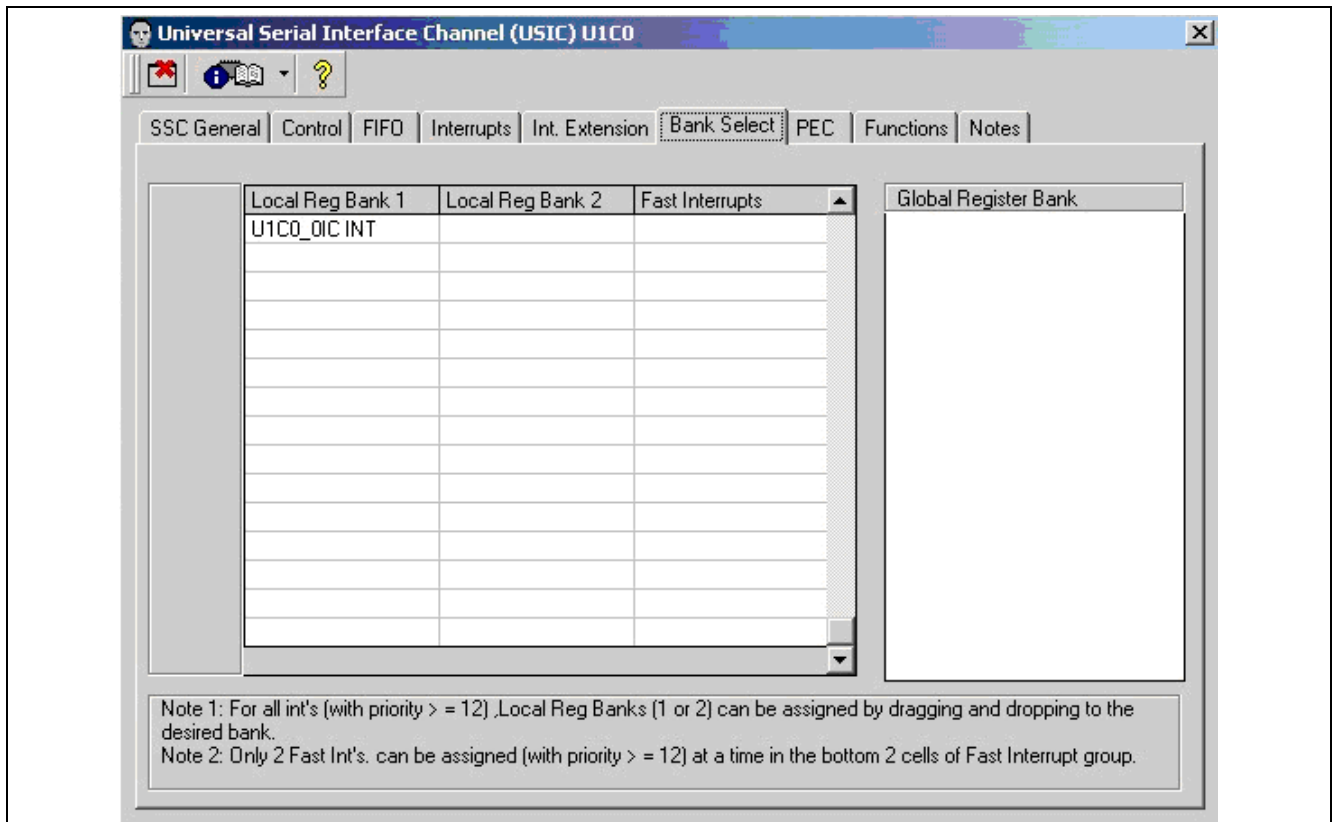


Figure 8 U1C0 Bank Select Settings

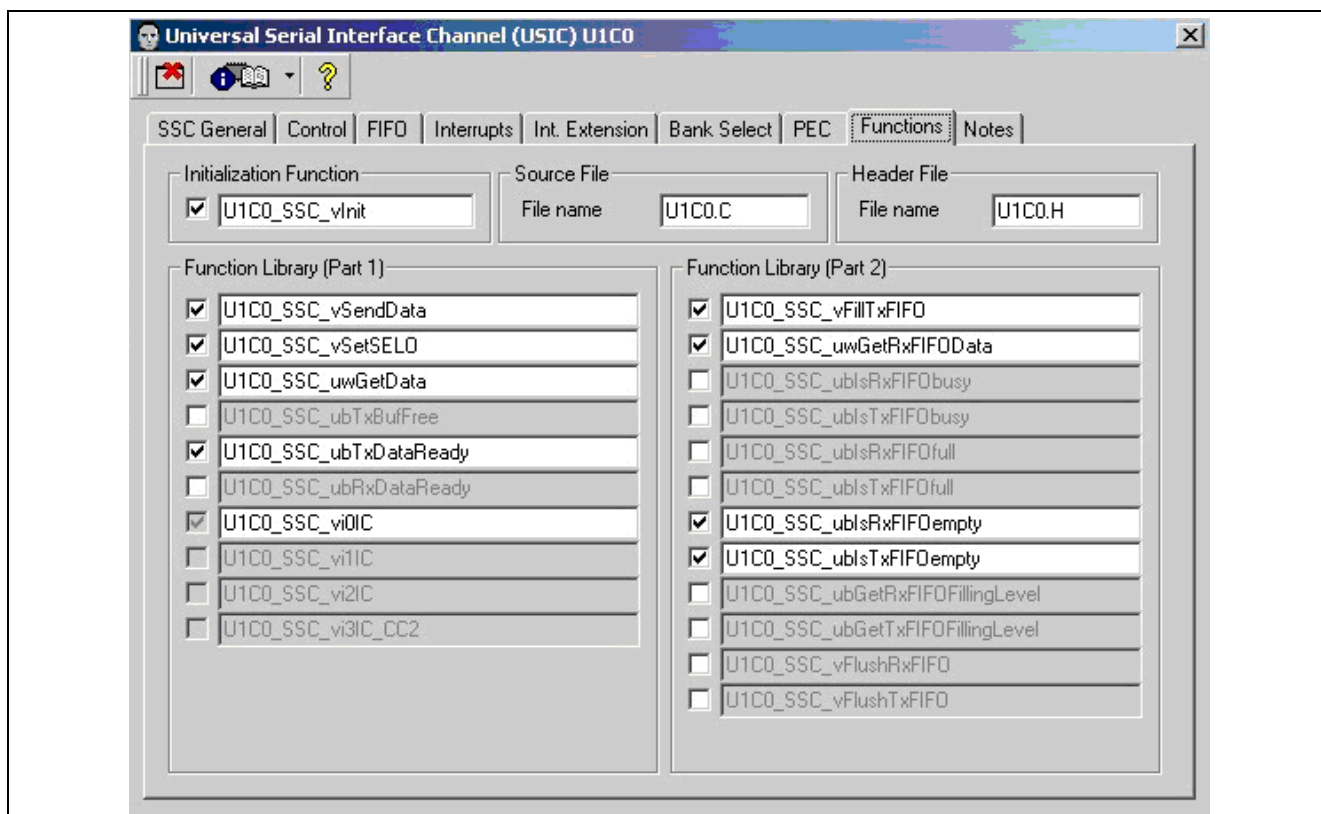


Figure 9 U1C0 Functions

2.2.2 Real-Time Clock (RTC)

The XE166 Real-Time Clock (RTC) is used in the background to time the standby function of the OLED.

The same RTC register is used both for setting the timing of the blinking LED and as the stopwatch for switching off the display. Timer values can be modified by software during run-time. The default setting during initialization is one minute.

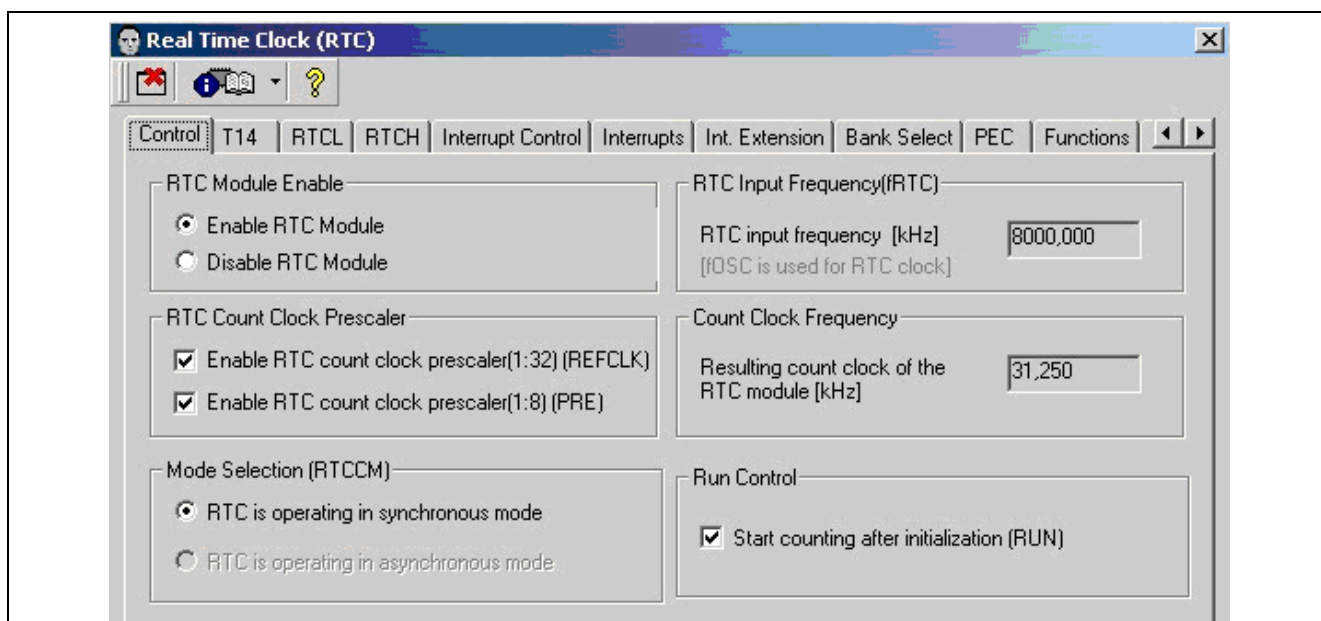


Figure 10 RTC Control Settings

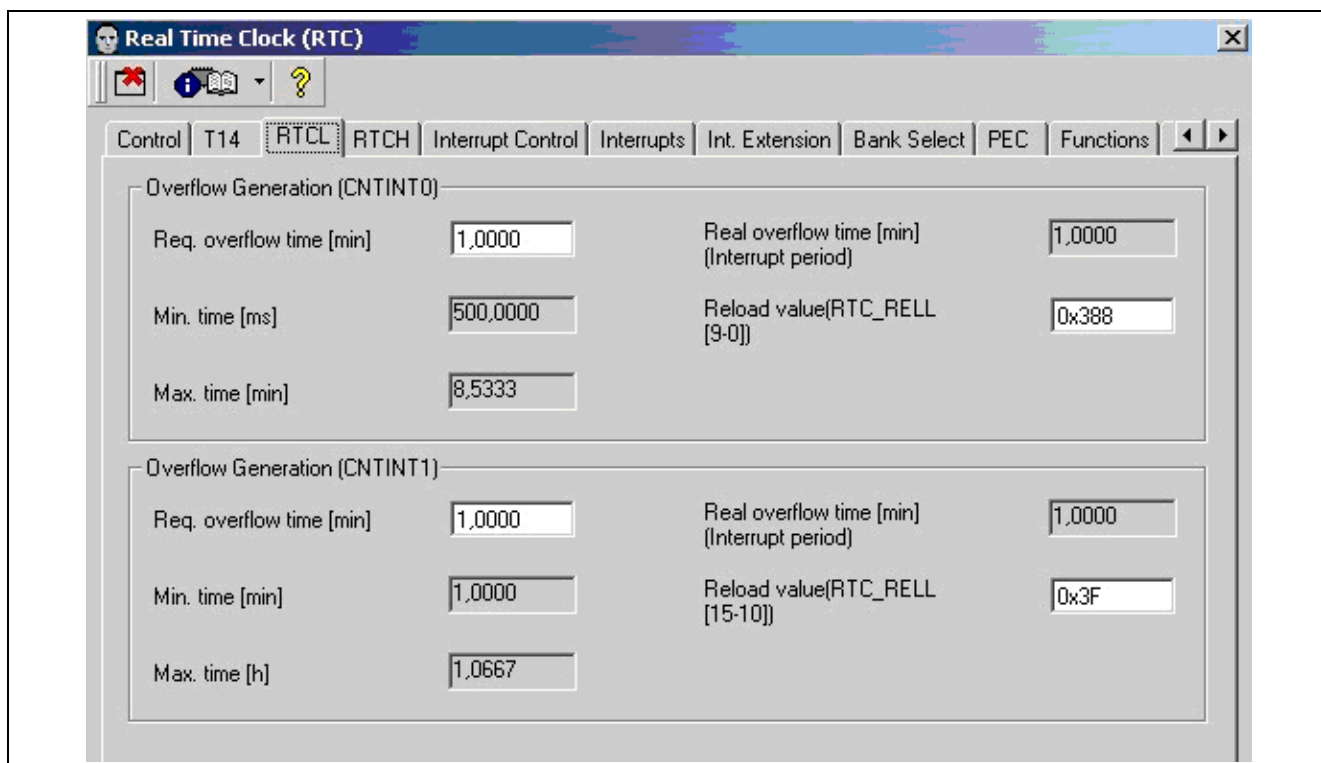


Figure 11 RTC Low Register Settings

The figure above shows the timing settings of the CNT INT 0 (Overflow Generation), configured in this example for 1 minute. If there have not been any switch events after that time, the display switches off. After switching off, software resets the value to five milliseconds for the LED to flash on/off. When there is a switch event the display switches on again, and the timer will be reset to 1 minute. To increase or decrease the screensaver time, modify this setting.

Note: If changing the value, the software values must also be changed! See the section "Modifying the timer display standby and 5ms-flashlight".

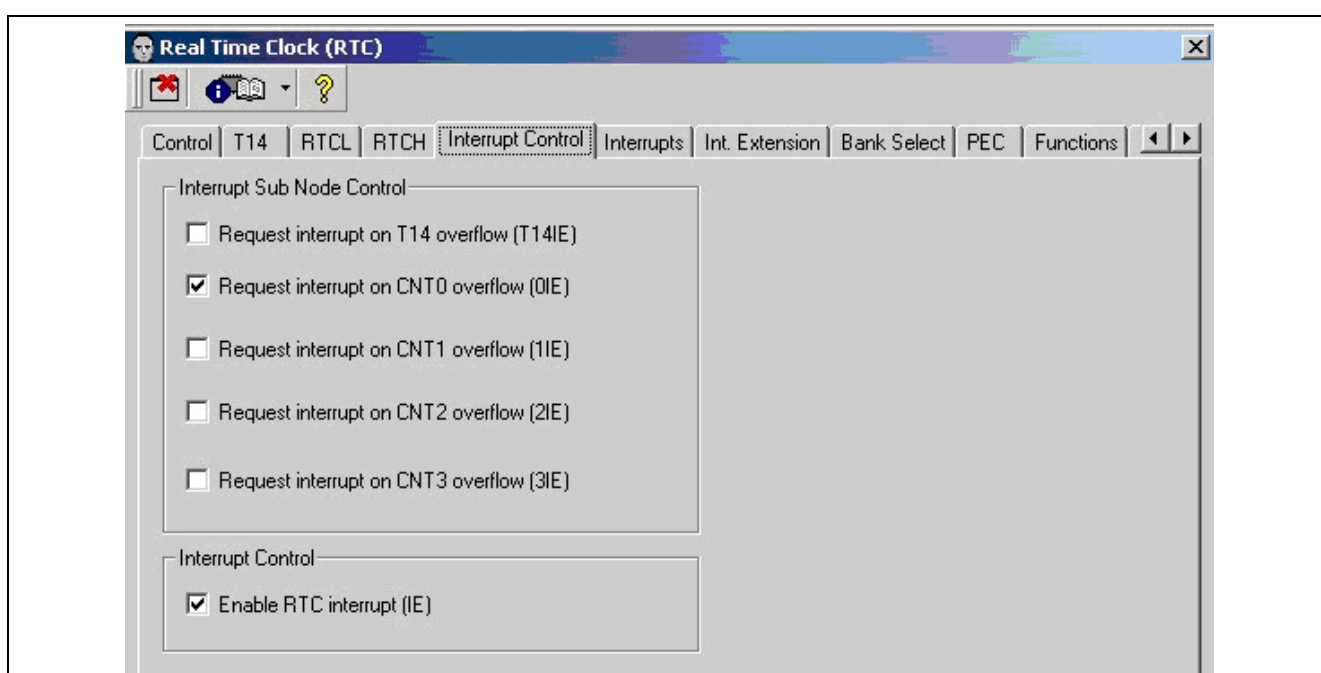


Figure 12 RTC Interrupt Control Settings

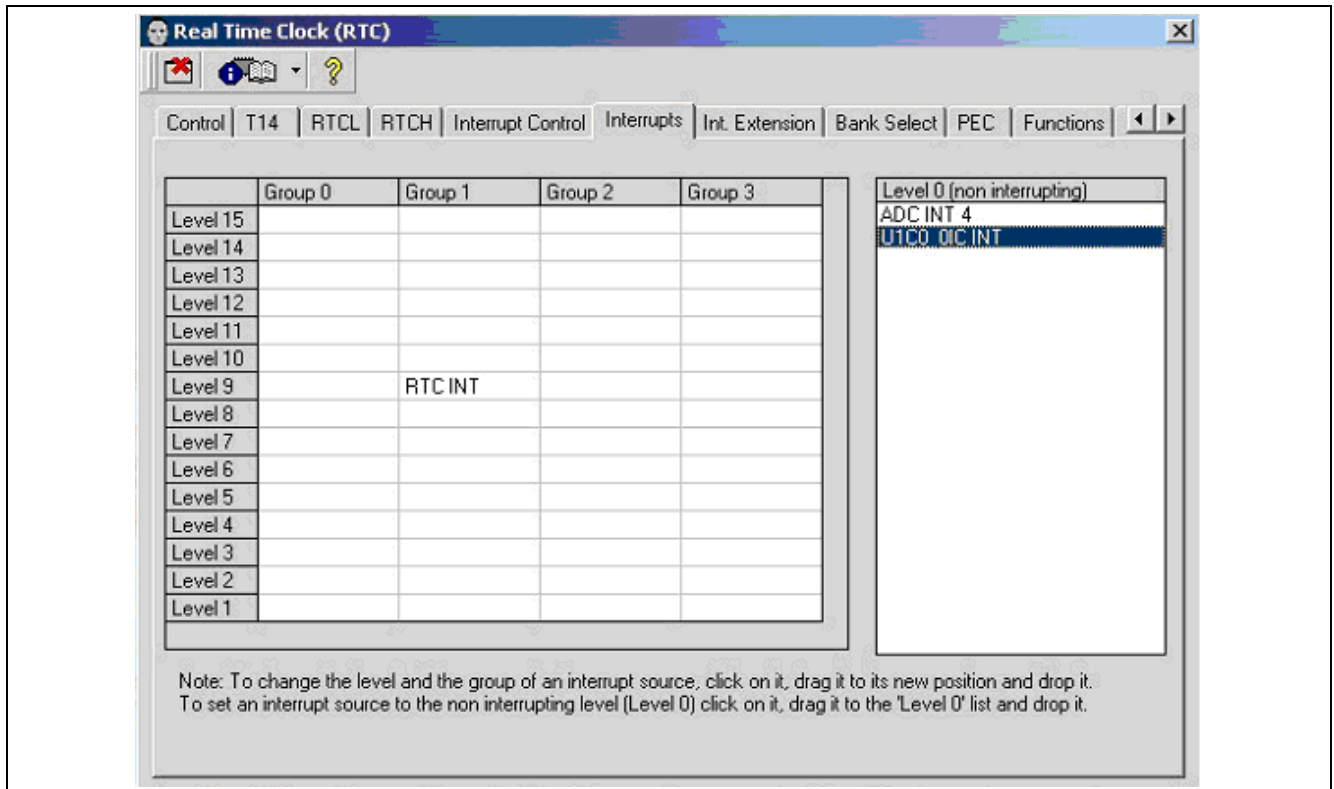


Figure 13 RTC Interrupt Settings

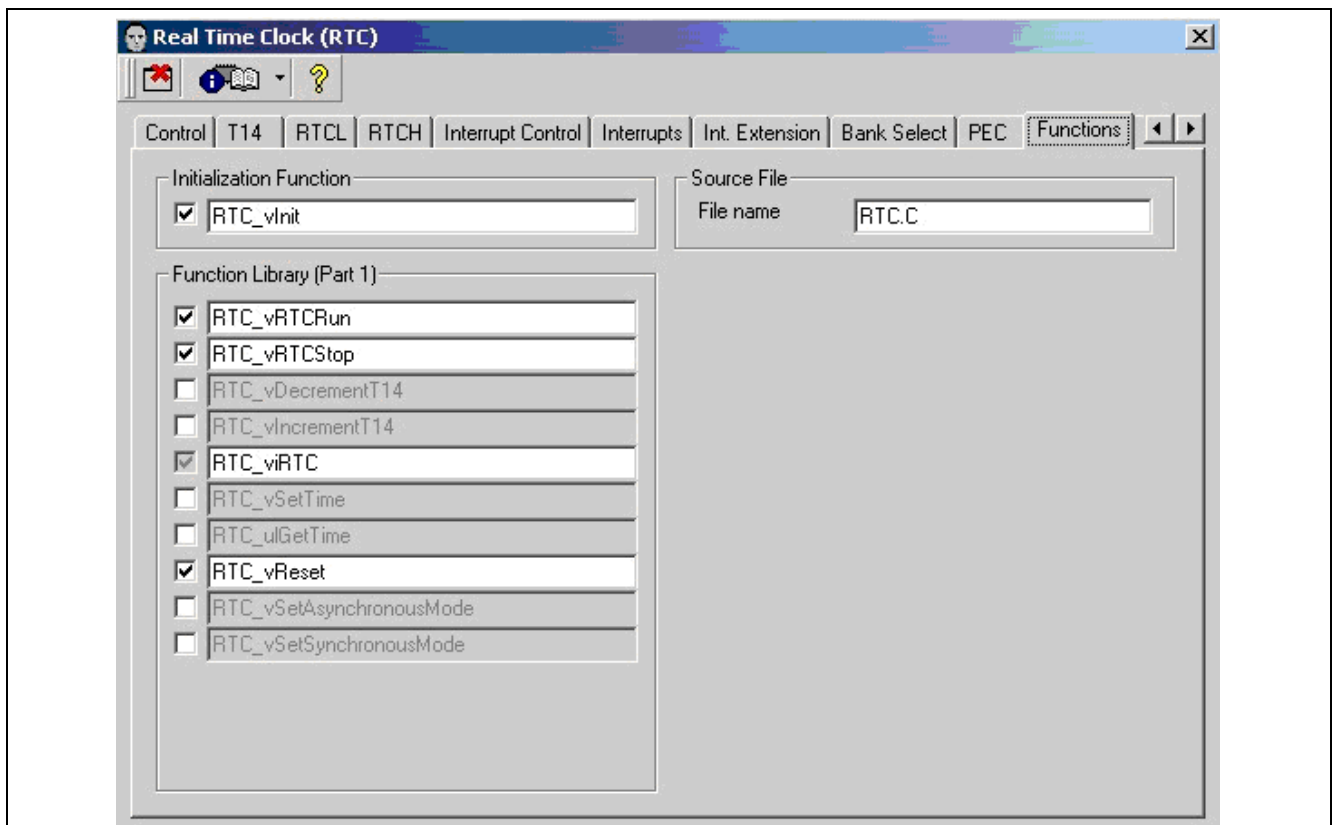


Figure 14 RTC Functions

2.2.3 IO Ports

Most of the ports are set automatically, dependent on other configuration settings. However the port pins 2.7 and 2.8 have to be set as output for the user LED's, while port pins 0.5, 5.0, 5.8 and 5.9 are used for the switches. The potentiometer uses port pin 15.0.

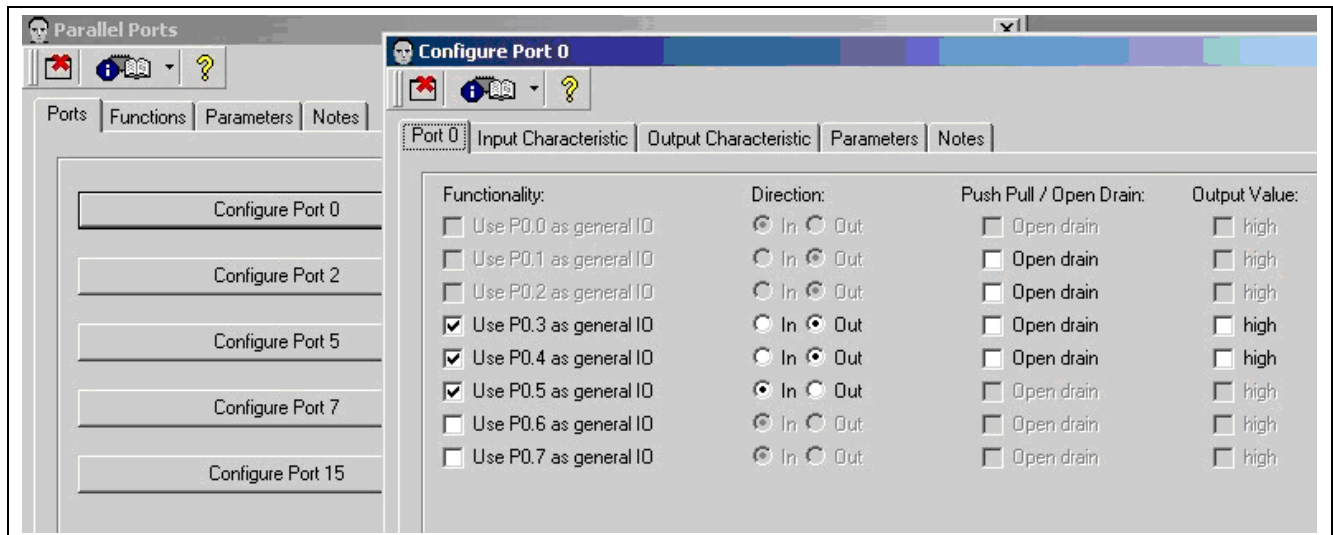


Figure 15 Settings Port 0

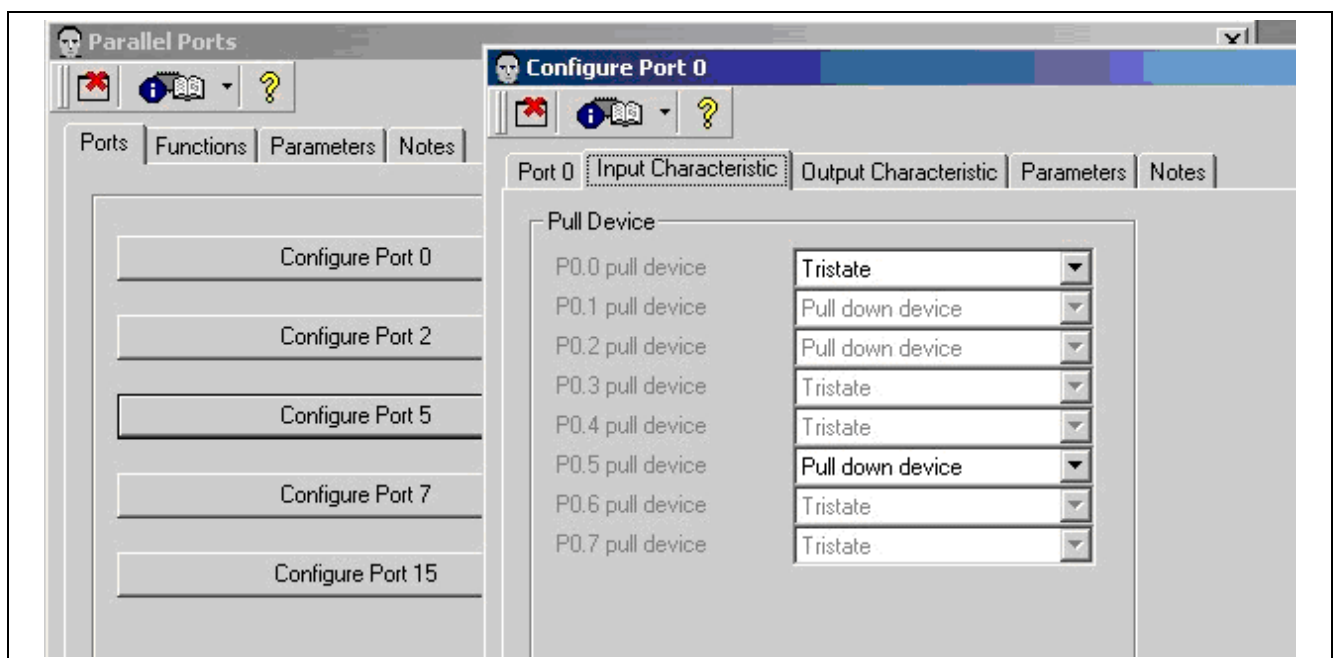


Figure 16 Input Characteristics Port 0

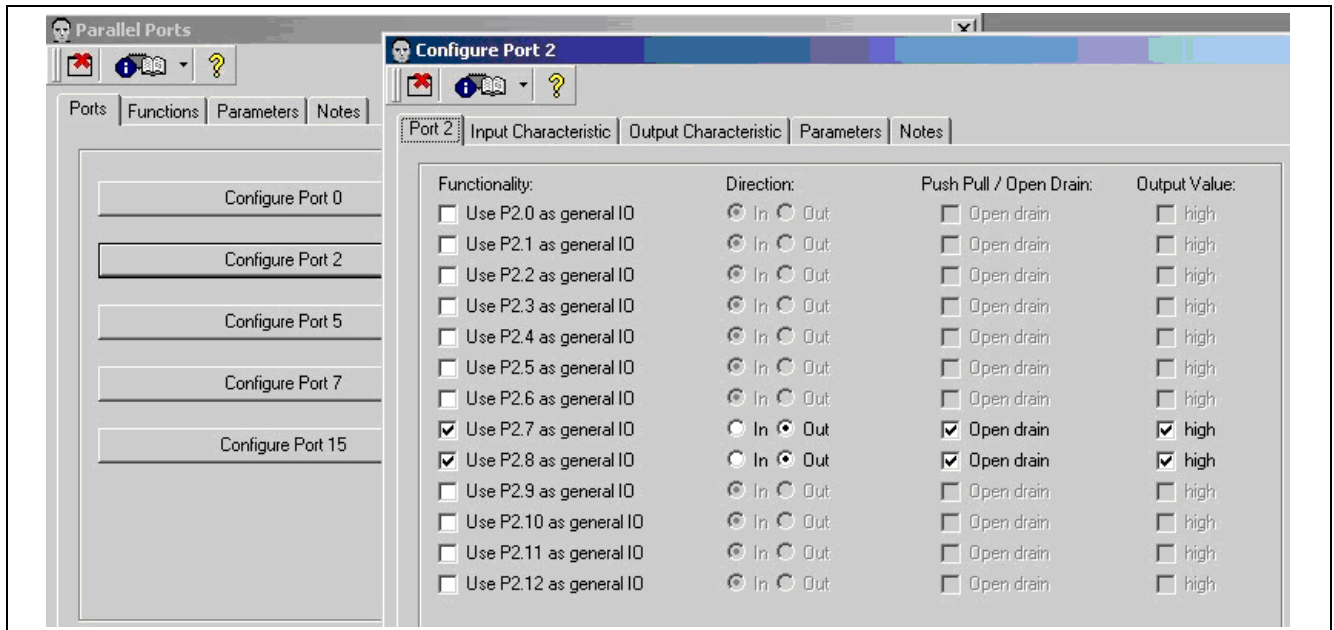


Figure 17 Settings Port 2

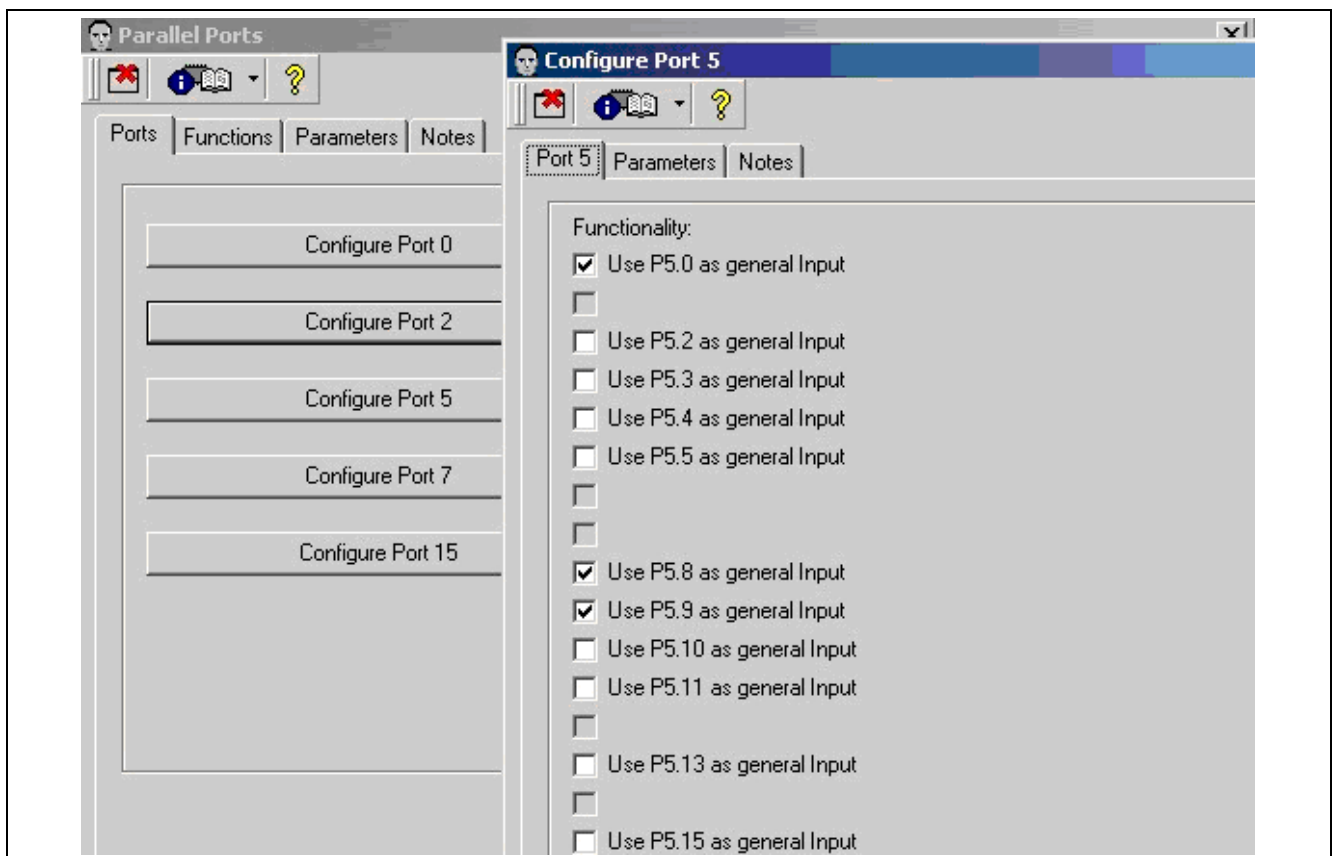


Figure 18 Settings Port 5

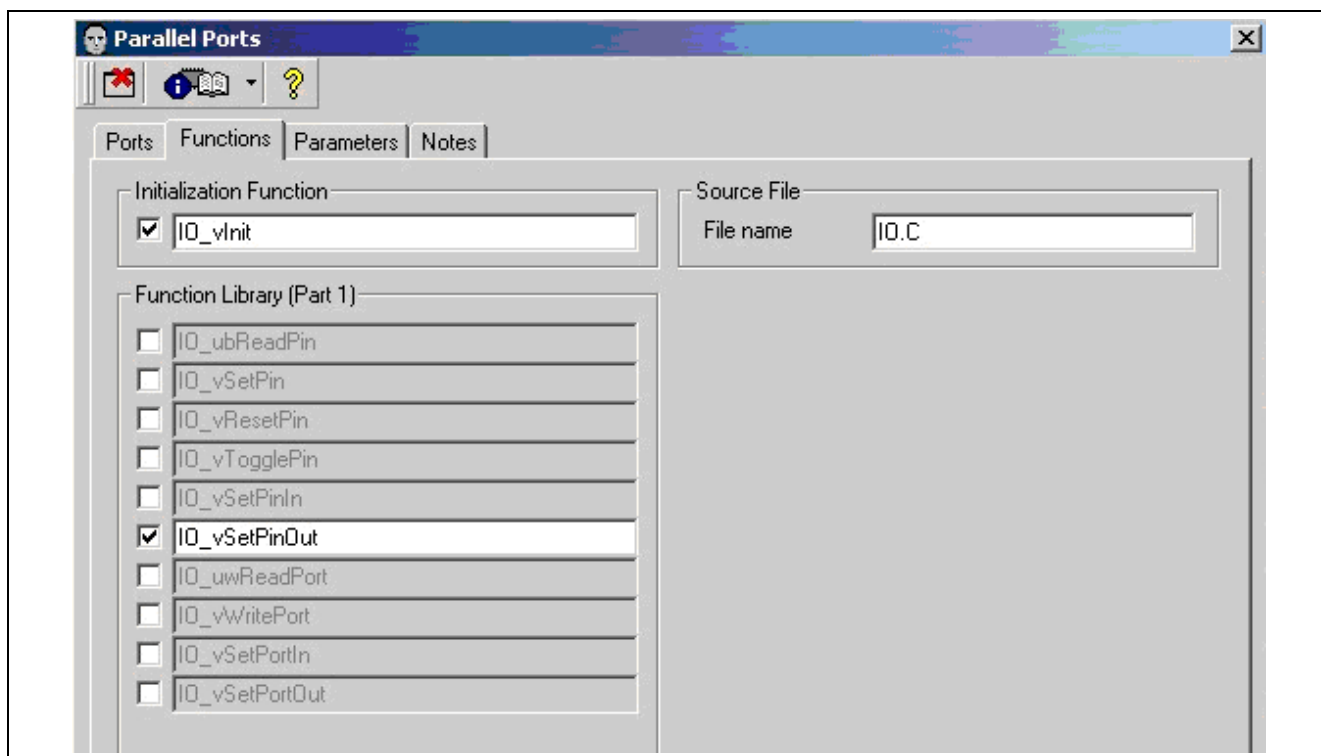


Figure 19 IO Port Functions

2.2.4 Analog Digital Converter (ADC) 1 – Channel 0

The Analog Digital Converter (ADC), Channel 0 is used for the potentiometer, and is connected to port pin 15.0.

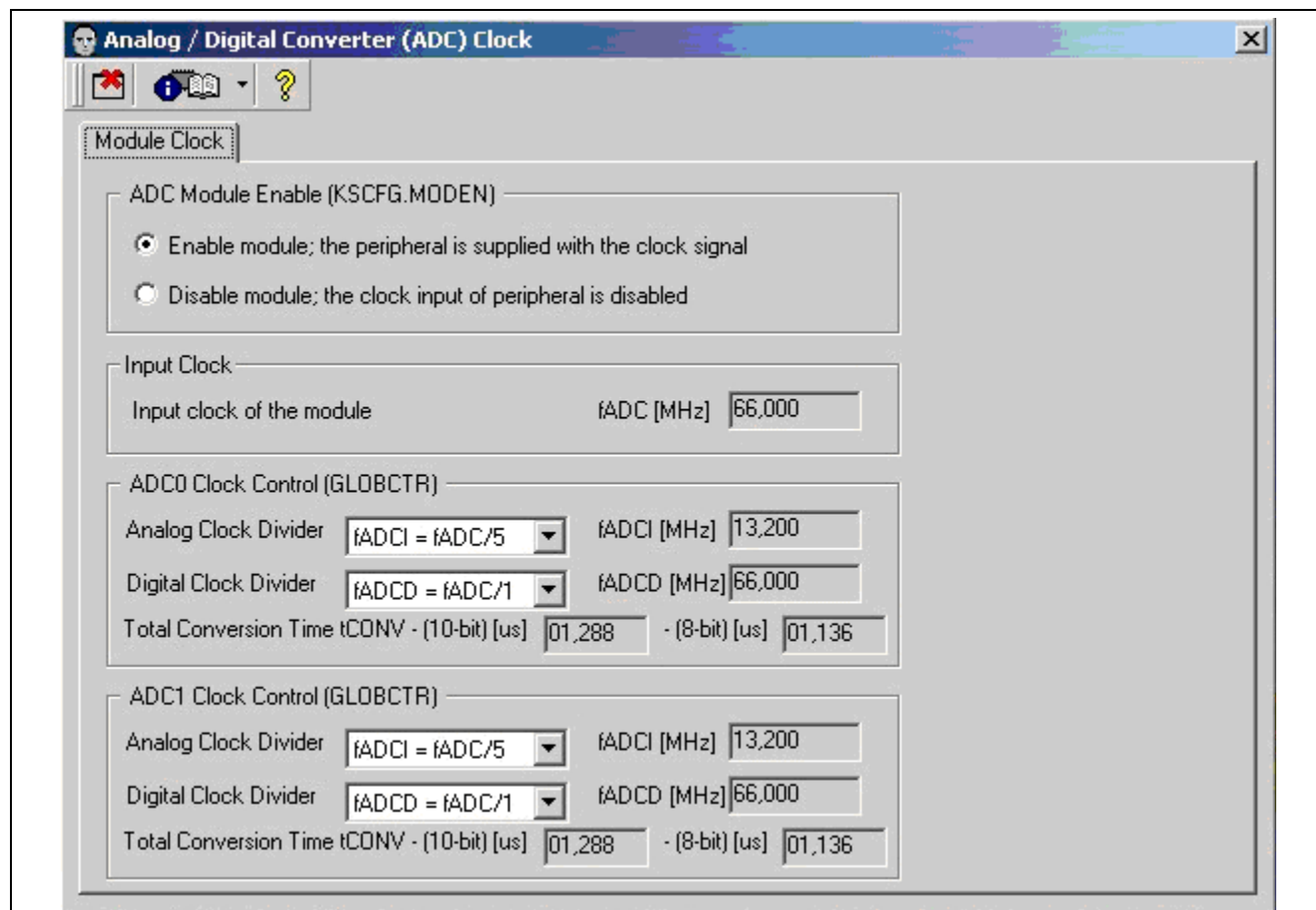


Figure 20 ADC Clock Settings

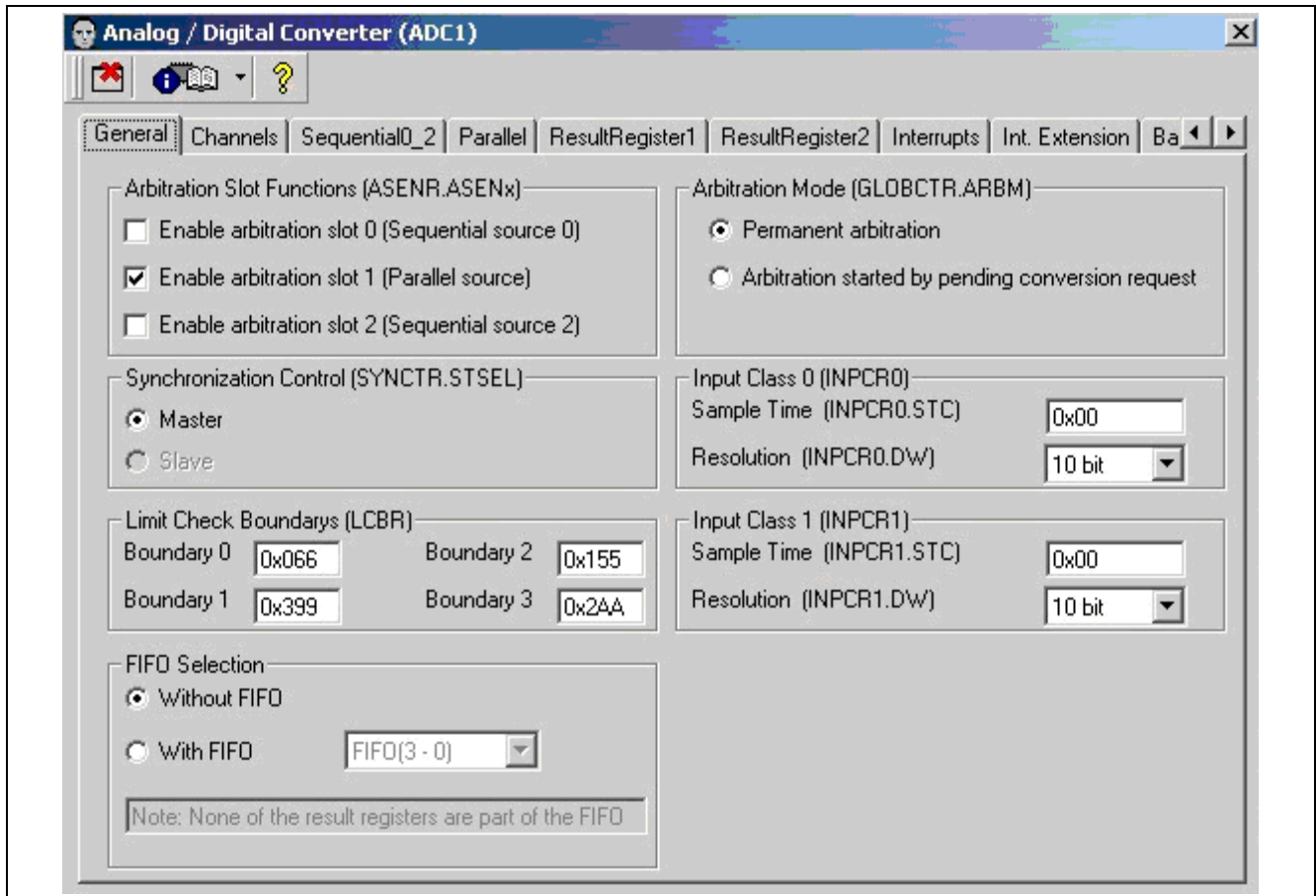


Figure 21 ADC 1 General Settings

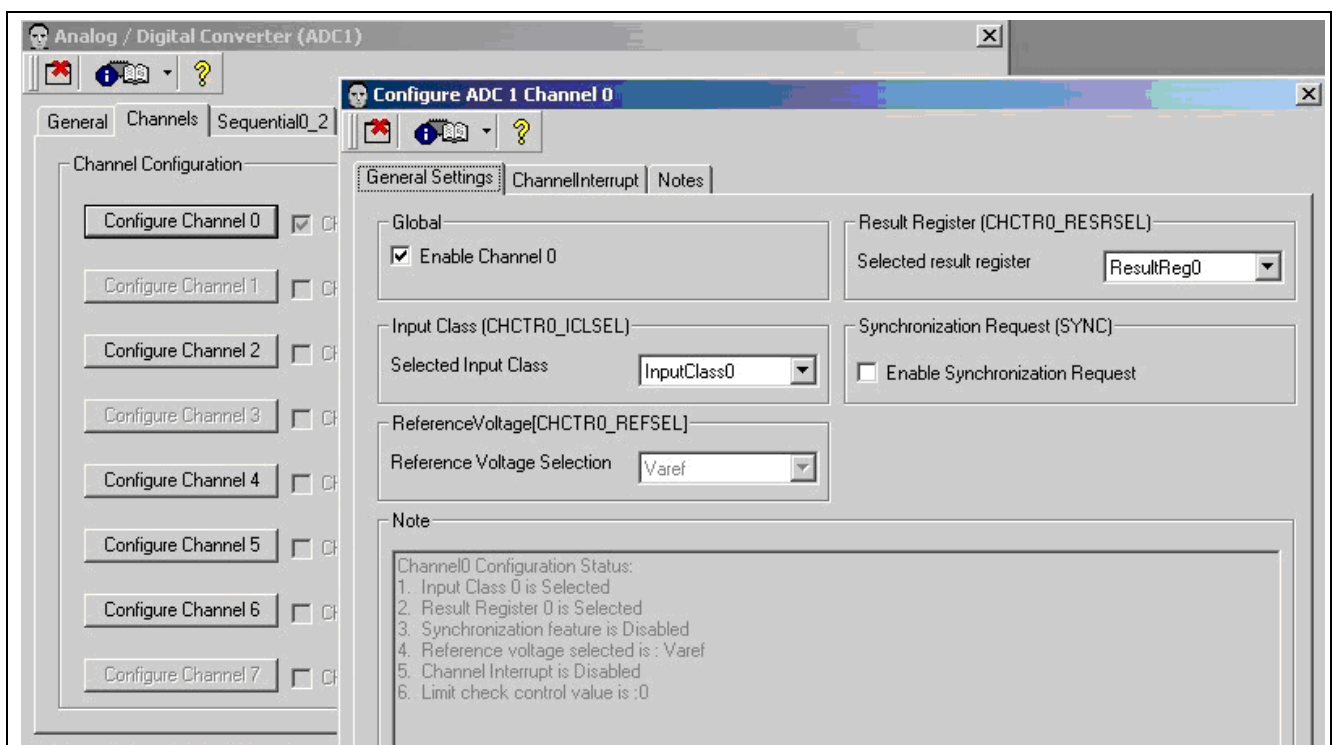


Figure 22 General Settings of ADC 1 - Channel 0

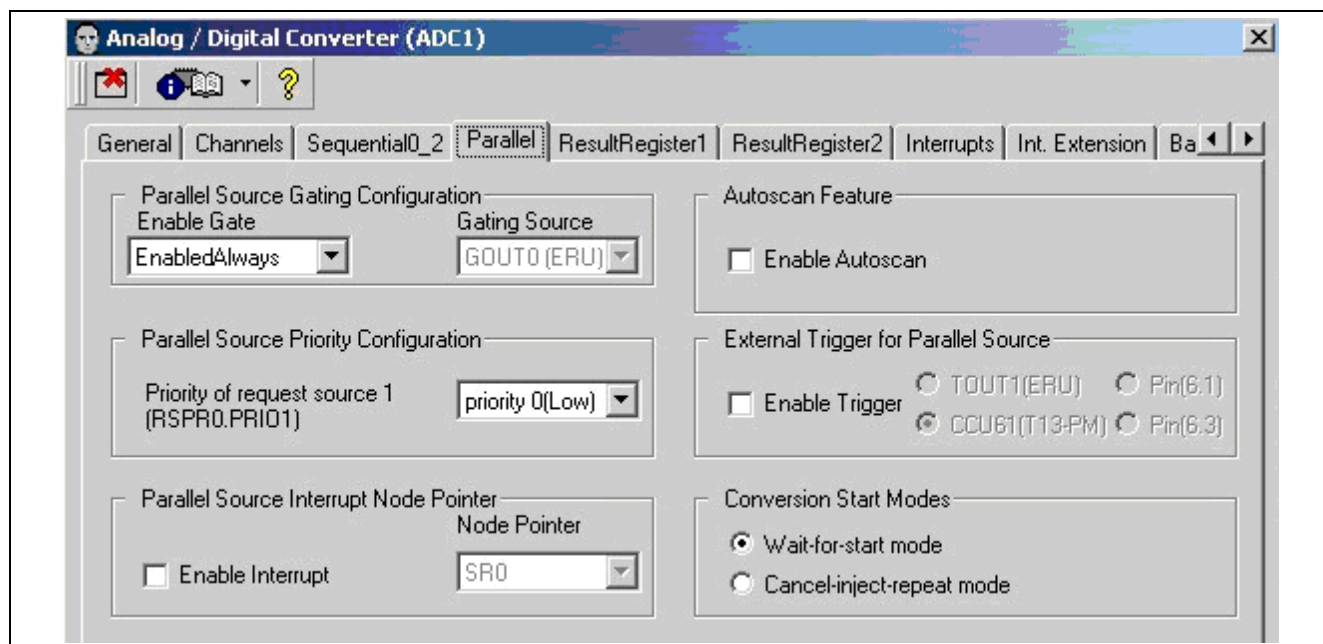


Figure 23 ADC 1 is Configured for Parallel Mode

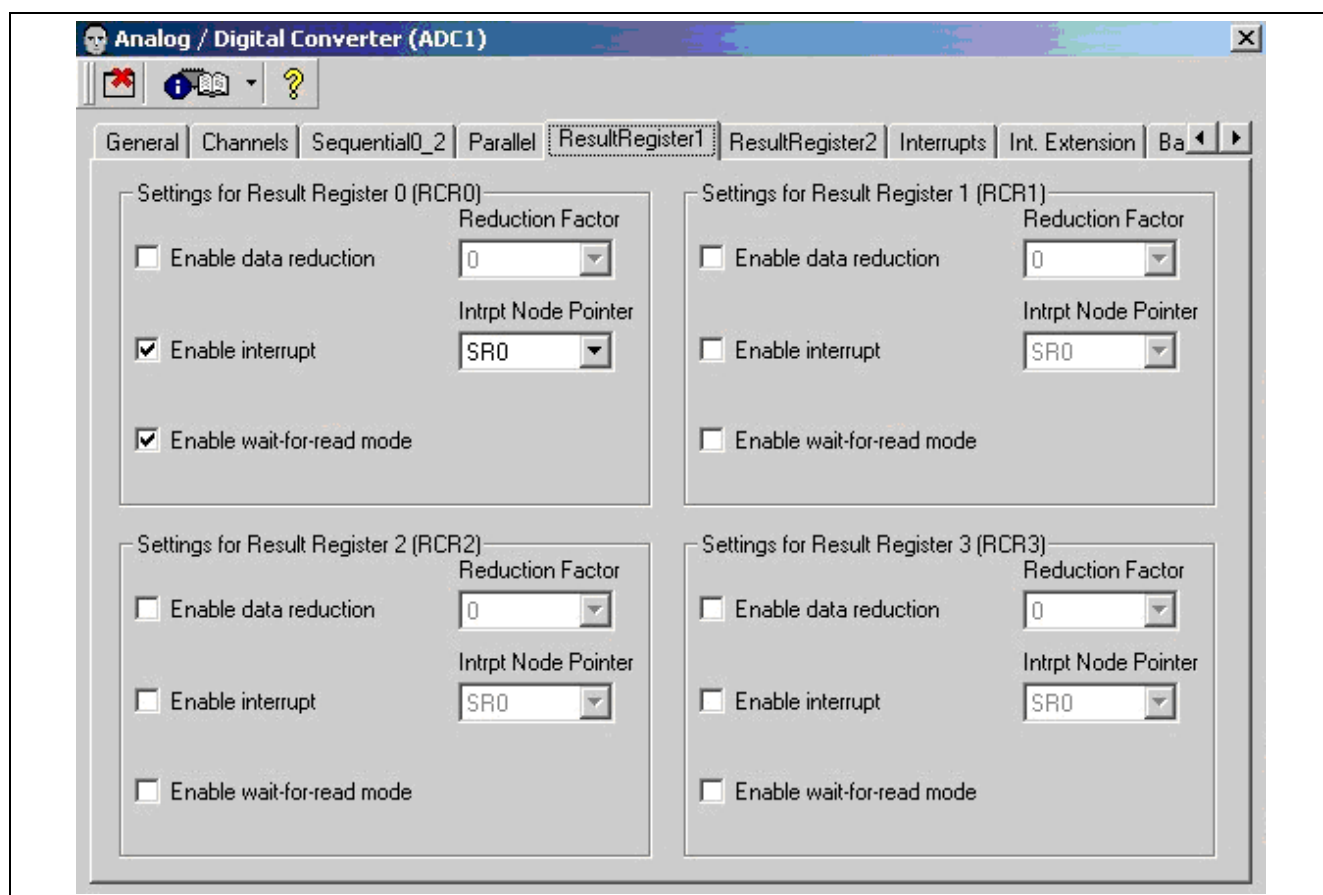


Figure 24 ADC 1, Result Register 1 Settings

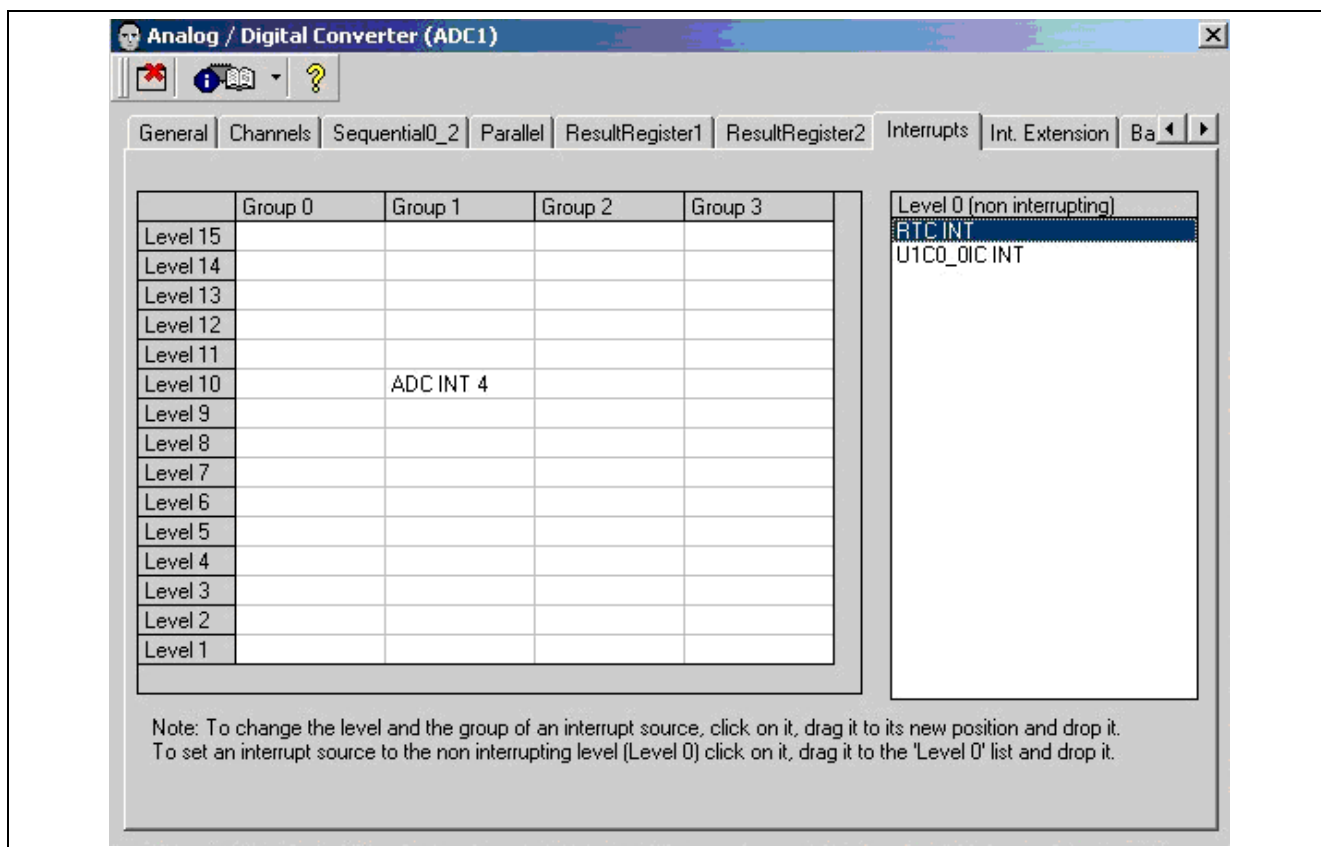


Figure 25 ADC 1 Interrupt Settings

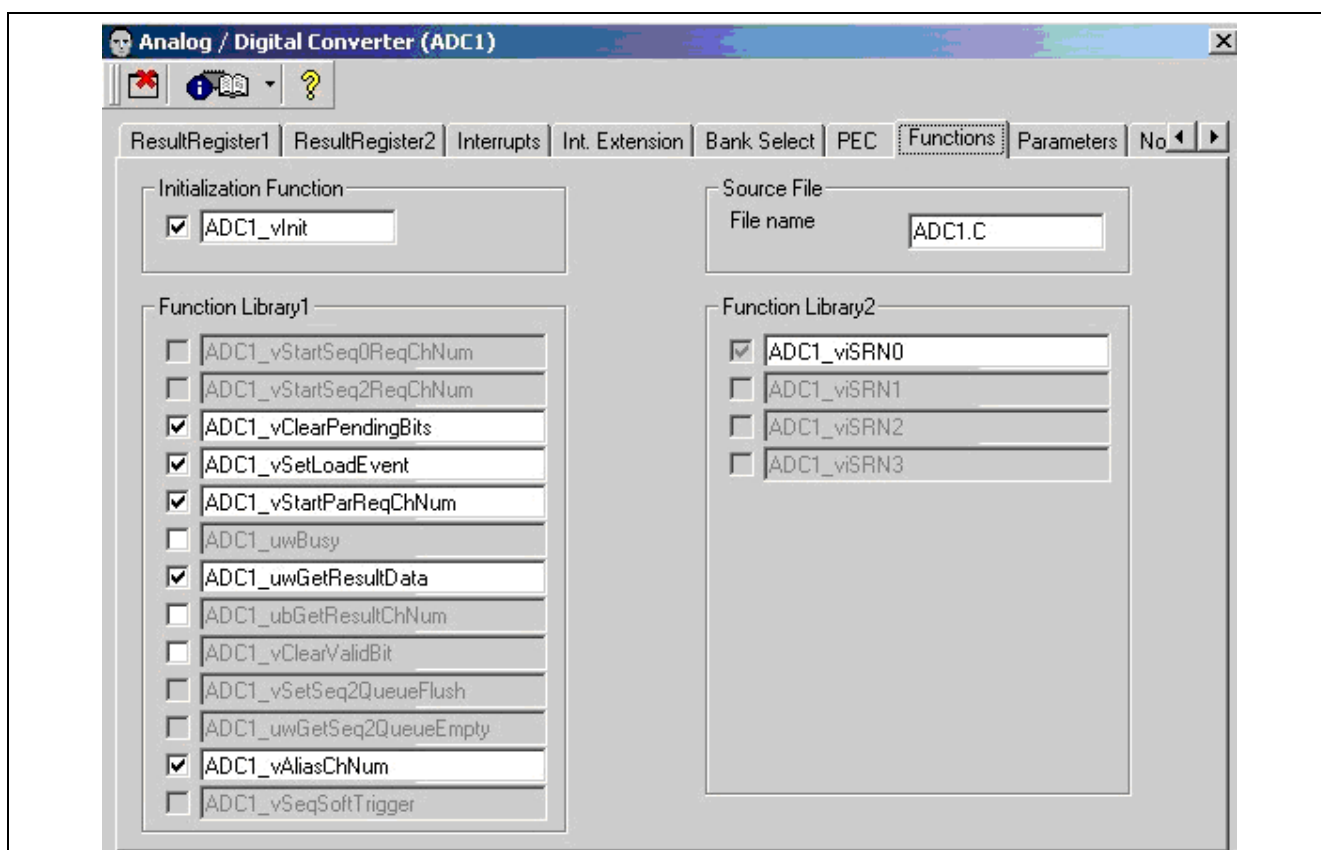


Figure 26 ADC 1 Functions

2.3 Functions

The CD provided with the Hitex UConnect Extension Board, contains a workshop of files that need to be modified to create the OLED display example. The relevant files are:

- Graphic.c
- Graphic.h
- OLED_Ild.c
- OLED_Ild.h
- OLED_font_Ild.h

The workshop includes a library to produce a graph display. Other basic functions provided include one to print text or print a rectangle for example. A description of each function is available in the c file comments.

Note: The OLED and the Ethernet module use some of the same port pins, but the necessary bit is set in the example application during initialization.

The colours used in this example design are the ones already used in the workshop. This 'dark' colour design is useful for protecting the OLED.

The additional files required with the existing workshop project to create the OLED display example are:

- OLED_example.c
- OLED_example.h

The following sections describe the sourcecode and the functions included in these files.

2.3.1 ADC Interrupt Service Routine (ISR)

```
_interrupt(ADC1_SRN0INT) void ADC1_viSRN0(void)
```

Code to round out the ADC result and copy it into a global variable.

2.3.2 RTC Interrupt Service Routine (ISR)

```
_interrupt(RTCINT) void RTC_viRTC(void)
```

This ISR contains the sourcecode to switch-off the display after one minute if there has been no switch or potentiometer action for that length of time. It then toggles the LED once every five milliseconds (to act as a type of flashlight); i.e. RTC_RELL register and RTC_RTCL register are reloaded with the value for 5 milliseconds, replacing the previous 1 minute setting.

2.3.3 Modifying the Timer Display Standby and 5ms-Flashlight

```
RTC_setRTC_RELL(u16 value)
```

The Real-Time Clock Counter 0 is used twice.

When the display is active the timer counts 1 minute cycles. When the display is inactive (i.e. the screensaver/flashlight mode is active), it counts five millisecond cycles.

This function modifies the RTC_RTCL and RTC_RELL register values, the reset value and the timer restart.

Note: There is no filter to modify only the bits of counter 0! The selected register will be completely modified, as well as the value of counter 1!

For centralization, two defines appear in the file RTC.h:

```
RTC_Reload_5ms
```

```
RTC_Reload_1min
```

Modify the timing values as required.

Note: Any timing changes are to made in conjunction with modifying the values in the DAVE project.

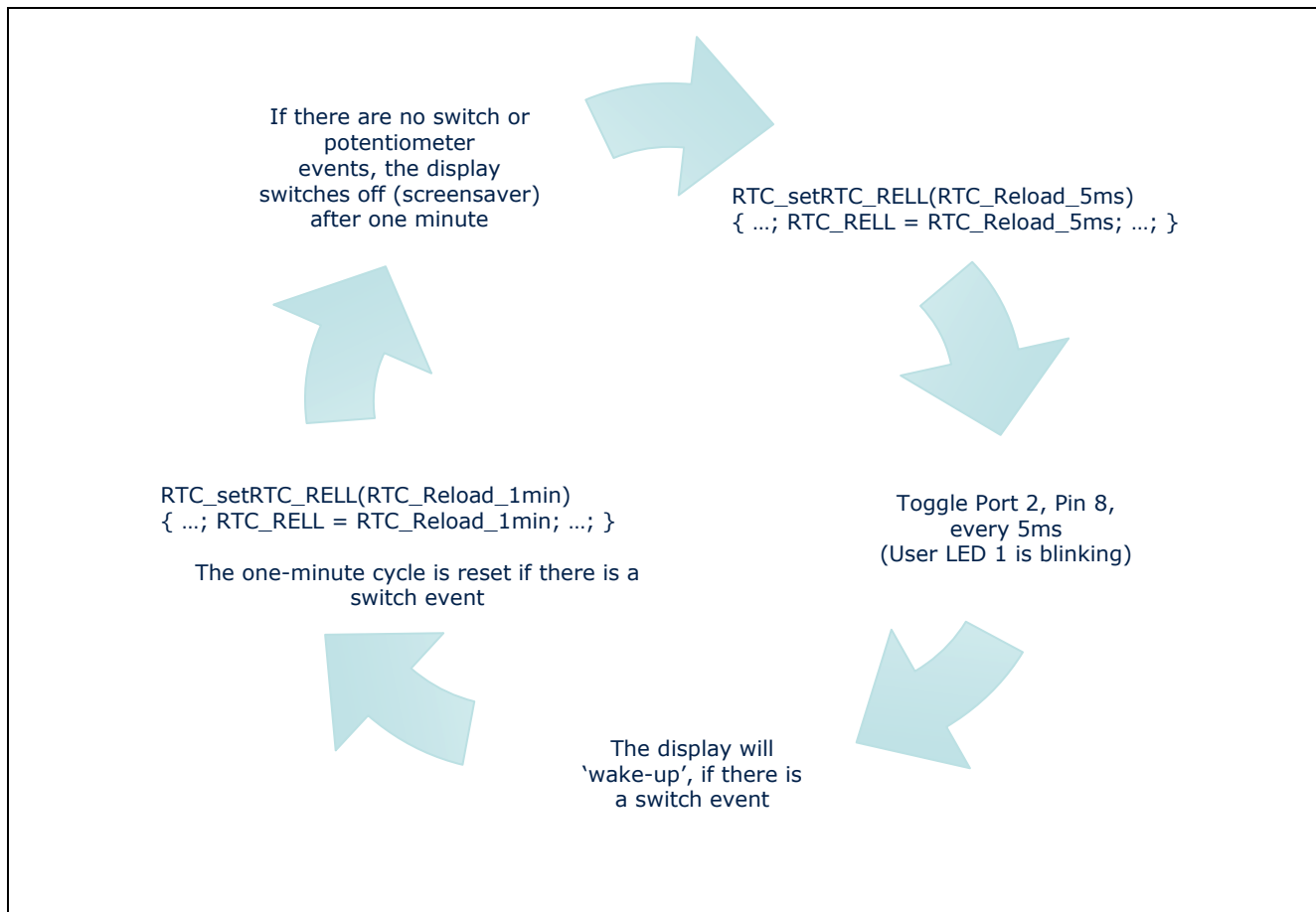


Figure 27 RTC Operation Chart

2.3.4 Text

```
void OLED_vert_text(t_rect rect, char *string, u16 txtColor, u16 bgColor)
```

Use this function to display text vertically. It centers the text vertically and horizontally. As `rect` this function expects the rectangle of the complete vertical bar which should be named, not only the rectangle around the text.

```
void OLED_convertNo2String(sword no, char* resultStr, u8 nSize, sword alignment)
```

Use this function to convert numbers (signed and unsigned integers) to characters.

The number to convert is the first parameter. The second parameter must be a pointer to a character field. This field contains the conversion result.

For a left-aligned result, set `sword alignment` to a negative value. Use zero for center-aligned and a positive value greater than zero for right-alignment.

When the parameter `nSize` is set to a value larger than zero, it limits the number of numeric characters.

There are four defines to calculate text start position and to indicate whether it should be centered horizontally or vertically. The result is `u16`. The string is necessary to calculate the length.

```
X_CENTER_HORIZ_TEXT(WIDTH, STRING)
```

```
Y_CENTER_HORIZ_TEXT(AREAHEIGHT)
```

```
X_CENTER_VERT_TEXT(AREAWIDTH)
```

```
Y_CENTER_VERT_TEXT(HEIGHT, STRING)
```

2.3.5 Basic Screen

```
u8 OLED_basicScreen(void)
```

This function prints a basic screen onto the OLED. Among the settings for this function, a choice is required between using a bar graph or scrollbar for the potentiometer.

The function can scroll different pages or control a motor connected via the CAN network for example. Depending on the required use, modify the definition at the top of the OLED_example.c file. The corresponding define is called `SHOW_POTI`.

Further options are to print a title bar (OLED Example in the screenshot below) and the names of the switches (SW1, SW2, SW3 and SW4 for example). Refer to the comments in the sourcecode for further information.

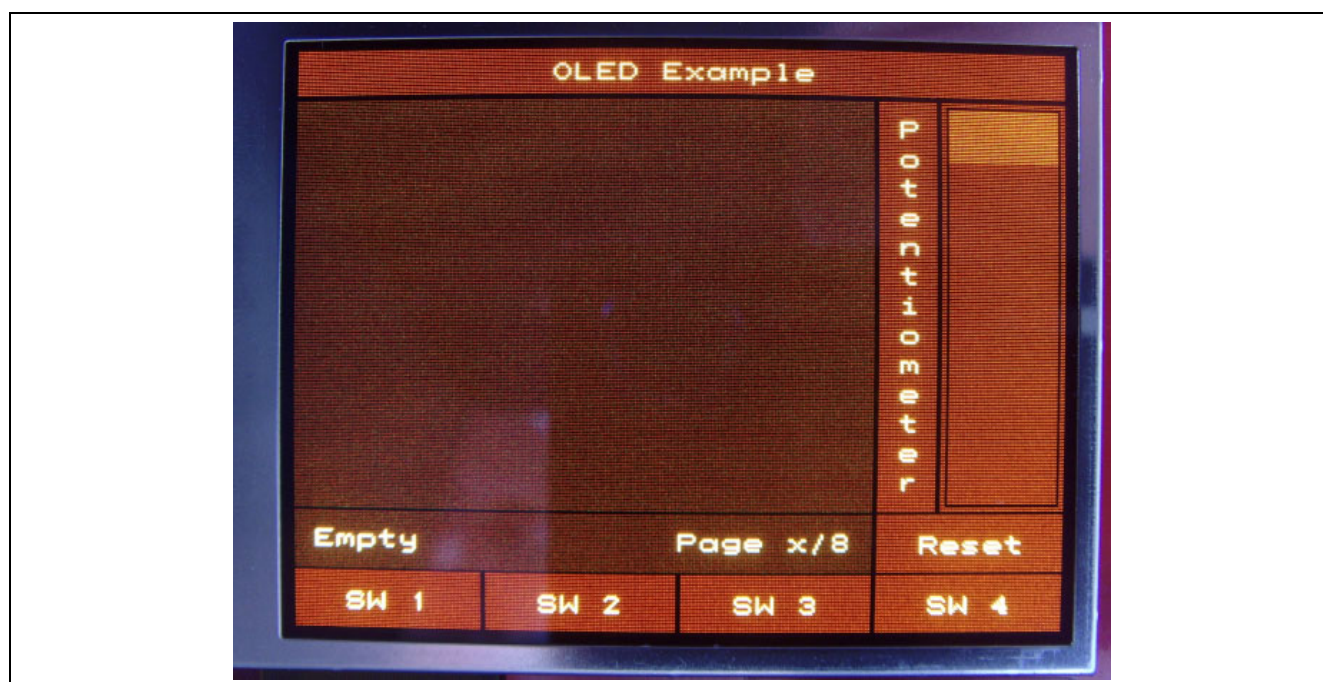


Figure 28 Basic Screen Example

2.3.6 Buttons / Switches

```
static void OLED_Button(t_point startpoint, char *string, u16 bgColor)
```

This function creates buttons. The first parameter is the startpoint; i.e. the start pixel in the OLED design. The second parameter is a string for the name of the button. The third parameter is the background colour.

In the application example the OLED_example.c file includes definitions for the different options such as start points and colours.

```
void OLED_switch(u8 sw, u8 state)
```

A function to print the four extension board switches on to the OLED at the bottom of the display. The content of each switch resides in a global field called `swText[4][11]`. When there is a switch event, this function will cause an output only.

```
void OLED_setSwitchText(char * sw1, char * sw2, char * sw3, char * sw4)
```

A function used to manipulate the text content of the switches. Before exiting it calls the function `OLED_Button()` to print any new text content to the display.

2.3.7 Show the Potentiometer

```
void OLED_FillPotiBar_ScrollBar(u8 pos)
void OLED_FillPotiBar_Bargraph(uword level)
```

There are two options for displaying the potentiometer. It is either used to turn pages or to show the level for steering a motor for example. Therefore the choice is between a scrollbar and a bar graph. Both the scrollbar and the bar graph only print an output to the display when there is a modification to the potentiometer.

Note: The maximum number of pages for a scrollbar or the value of the maximum level for a bar graph must be defined. The defines are called:

```
BARGRAPH_MAX_POS
SCROLLBAR_MAX_POS
```

2.3.8 Application Example

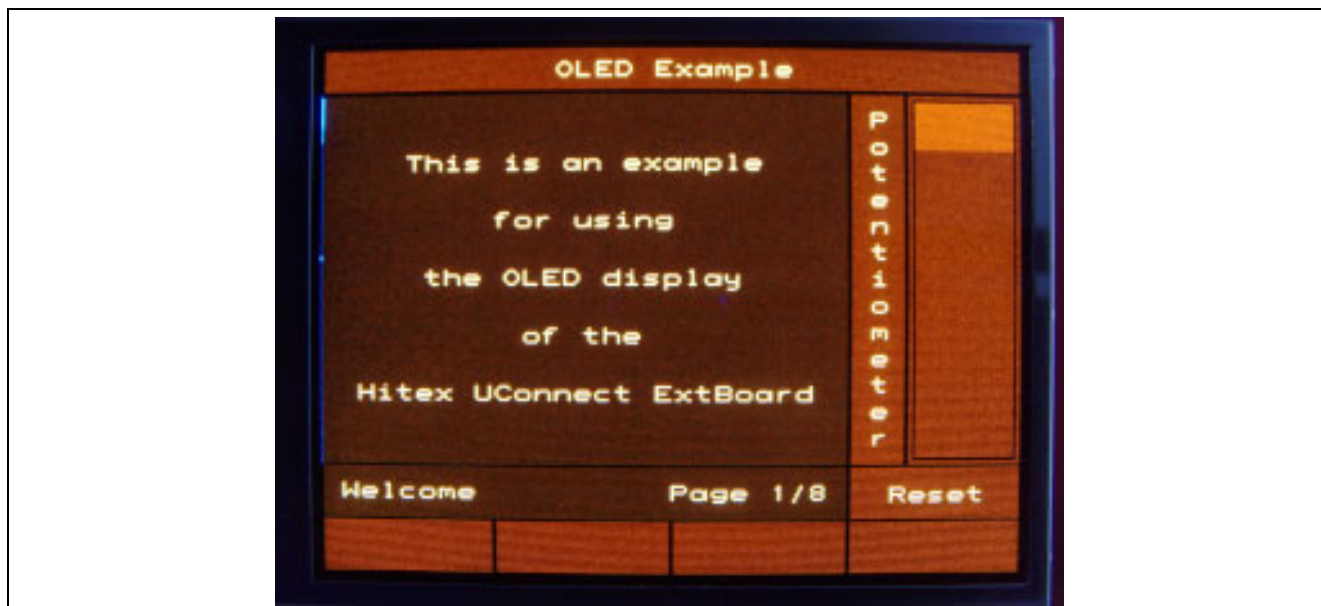


Figure 29 Application Example: OLED Example

```
void OLED_example_Application(void)
```

The application resides completely inside this function. To implement this application execute only this function in `main()`.

At the beginning of the function are the initializations. Alongside the peripheral init functions, the port pin P0.3 is set to switch the SPI for the OLED. It also calls the function to print the basic screen. Before the infinite loop, the User LED 1 is switched on like a power-on signal.

Inside the infinite loop the switches are monitored on every cycle. If there is an event, the button function is called and the `RTC_RTCL` register will reload. This action resets the one-minute timer for switching off the display (screensaver). The same process happens when actuating the potentiometer.

After an action to re-activate the display (push any switch), the display switches on, the User LED 1 is switched on permanently, and the appropriate registers will reload for the one-minute timer.

2.3.9 Design Examples for the Display Information Area

Some page examples for using the Information area are integrated in the workshop example provided. The content is described in the table below. The example shows the complex mathematical function of the “Mandelbrot” set. This means that there is a calculation with a mathematical function, which works with the complex plane. This plane describes the area of the display. At every coordinate a calculation is started in that plane. The result of this calculation is used to determine the colour of the appropriate pixel.

Table 1 Page Description

Page	Title	Content / Function
1	Welcome	A short welcome text.
2	Mandelbrot	Start and restart the “Mandelbrot” picture with reset start values.
3	Change coordinates	Change the x- and y-coordinates of the start pixel for a new picture. The current value of x and y are shown the page title bar. The picture will continue to be drawn with the last parameters set.
4	Zoom	Change the zoom factor from one to eleven. The page title bar shows the current zoom factor. The picture is shown at the last zoom parameter set.
5	New Picture	When the start pixel or the zoom factor have been modified, this page will display a new picture with the new parameters.
x	Empty	This application example is designed for eight pages, but the upper three are empty.

The remainder of this section gives some screen shots of this example application:



Figure 30 OLED Example – Welcome Screen



Figure 31 OLED Example – Apfelmaennchen

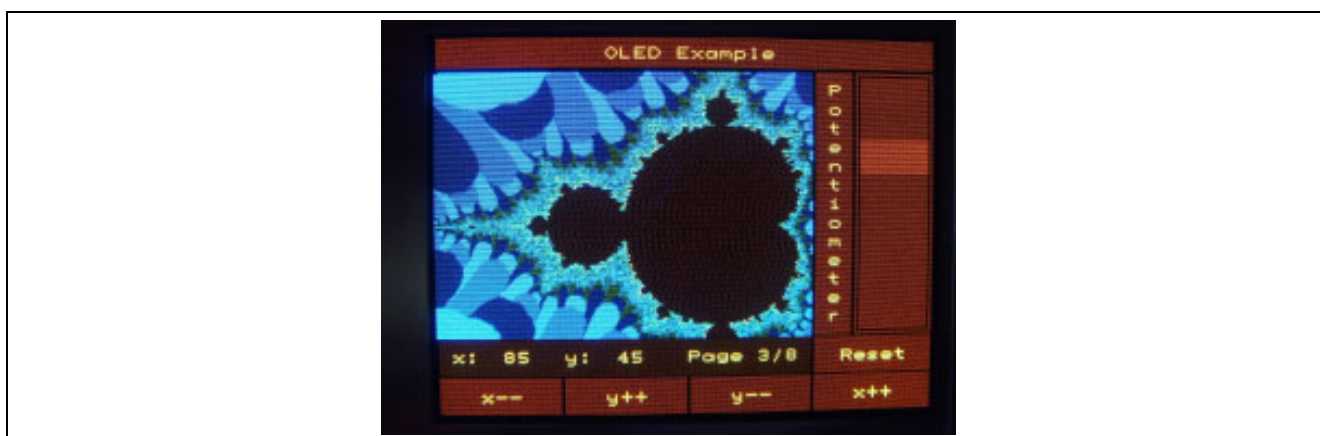


Figure 32 OLED Example – Coordinates



Figure 33 OLED Example – Zoom Factor



Figure 34 OLED Example – New Picture

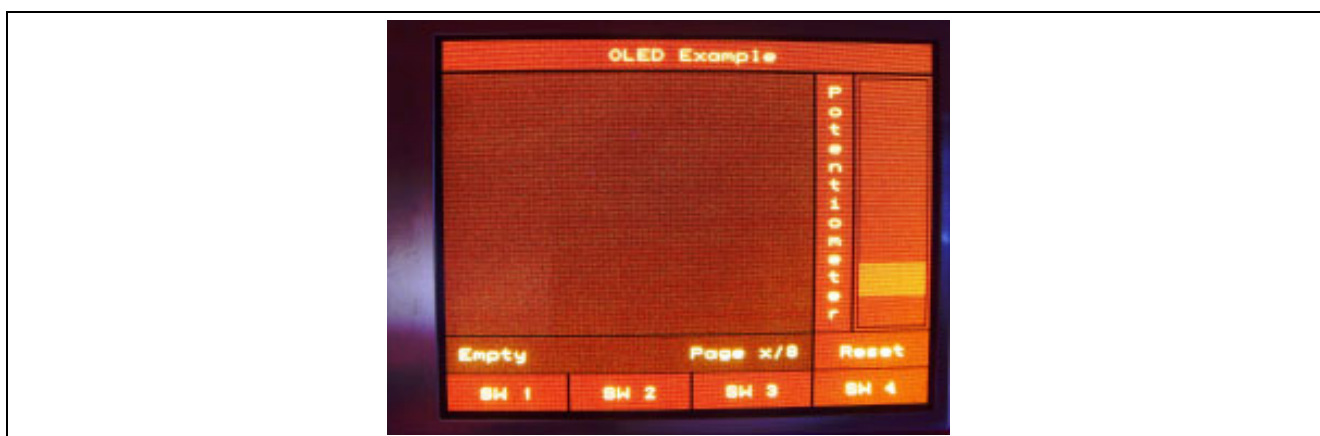


Figure 35 OLED Example - Empty

www.infineon.com