MAX11300/MAX11301 PIXI Configuration Software User Guide

Rev 0, 1/15

Table of Contents

List of Figures

1. Overview

This document explains how a Windows®-based design tool can be used to create a configuration bitstream for the MAX11300 and MAX11301. **The MAX11300 and MAX11301 are functionally similar from a configuration design, with the difference being the type of serial interface (MAX11300 = SPI, MAX11301 = I^2C). Within this document the MAX11300 is often used to refer to both part numbers and is also referred to as PIXI™. Refer to the MAX11300 and MAX11301 IC data sheets for detailed information regarding the operation of the IC.**

The MAX11300 and MAX11301 ICs offer 20 ports, all of which can be individually configured in a number of different modes including ADC, DAC, GPI, GPO, or analog switch terminal. MAX11300 Configuration Software provides easy configuration for the MAX11300 and MAX11301 PIXI devices and all possible configuration modes. Each port can be configured through simple drag-and-drop functionality to select the mode required. The software tool generates the bitstream that users download to the target PIXI in order to configure its functionality within their end application. All configurations implemented through this software can be saved for later use.

2. Required Equipment

- PC with Windows OS (this software has been tested with Windows XP®, Windows 7, and Windows 8 and 8.1 running .net v4)

3. Procedure

The MAX11300 Configuration Software is not only a stand-alone tool, but also included within the MAX11300 EV KIT Software and the Munich GUI, which support the hardware platforms allowing easy evaluation and prototyping with MAX11300. Follow these steps to download and install the software:

- **MAX11300 Configuration Software**
  a) Go to **www.maximintegrated.com/evkitsoftware** to download the latest version of the **MAX11300 Configuration Software**, **MAX11300ConfigurationSetupV1.1.zip**. Save the Configuration Software to a temporary folder and uncompress the ZIP file.
  b) Install the MAX11300 Configuration Software on your computer by running the **MAX11300ConfigurationSetupV1.1.exe** program inside the temporary folder. The program files are copied to your PC and icons are created in the Windows **Start | Programs** menu.
- **MAX11300 EV KIT Software**
  a) Go to **www.maximintegrated.com/evkitsoftware** to download the latest version of the **MAX11300 EV KIT Software**, **MAX11300EVKitSetupV1.1.zip**. Save the MAX11300 EV KIT software to a temporary folder and uncompress the ZIP file.
  b) During the installation process the user can select which components to install between the Configuration Software and the EV KIT GUI.
  c) Install the EV KIT software and USB driver on your computer by running the **MAX11300EVKitSetupV1.1.exe** program inside the temporary folder. The program files are copied to your PC and icons are created in the Windows **Start | Programs** menu. During software installation, some versions of Windows may show a warning message indicating that this software is from an unknown publisher. This is not an error condition and it is safe to proceed with installation. Administrator privileges are required to install the USB device driver on Windows. **The software requires .NET Framework v4**. If this framework is not detected during installation, the installer will launch **dotNetFx40_Full_setup.exe** to install it. Internet access may be required to install the .NET Framework v4 if it is not already installed.
- **Munich GUI 2.02**
  a) Go to **www.maximintegrated.com/evkitsoftware** to download the latest version of the **Munich GUI 2.02** software, **Munich_GUISetupV2.02.zip**. Save the Munich GUI 2.02 software to a temporary folder and uncompress the ZIP file.
  b) During the installation process the user can select which components to install between the Configuration Software and the Munich GUI.
  c) Install the Munich GUI 2.02 software and USB driver on your computer by running the **Munich_GUISetupV2.02.exe** program inside the temporary folder. The program files are copied to your PC and icons are created in the Windows **Start | Programs** menu.

Start the MAX11300 Configuration Software by opening its icon in the Windows **Start | Programs** menu. The MAX11300 Configuration Software main window appears, as shown in Figure 1. Drag and drop **Components** into the **Device**, wire them up, and then use **File | Generate Registers** to export the configuration to **Max11300Register.csv**.

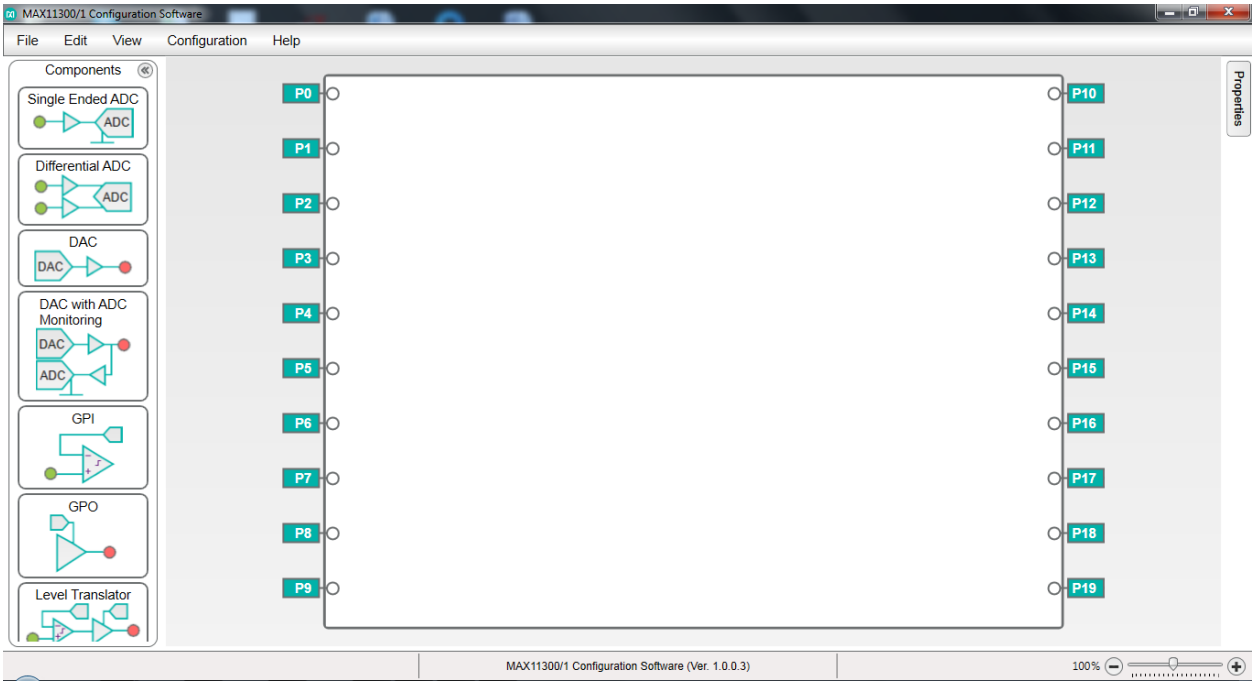

Figure 1. MAX11300 Configuration Software

4. Software Functions

Figure 2 shows the main view for the software.

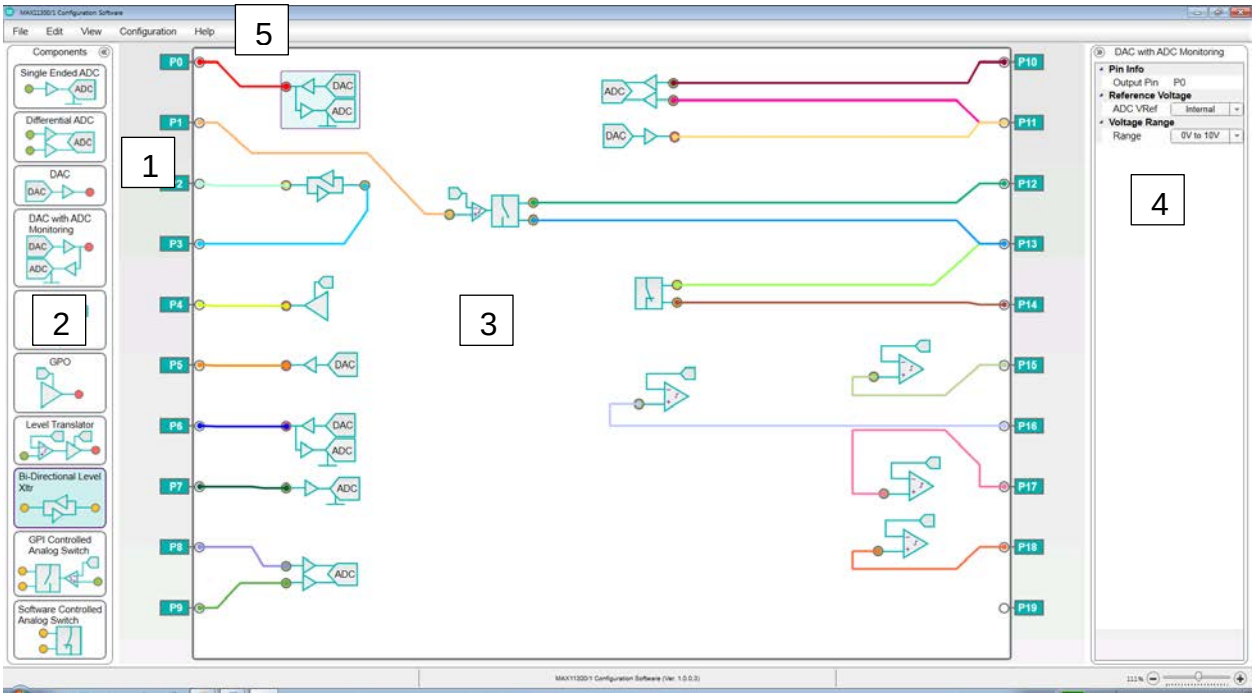

Figure 2. MAX11300 Configuration Software Main View

1. **Ports:** 20 user-configurable I/Os.
2. **Components:** The range of modes that are used to define the functionality each I/O port.
3. **PIXI Device Area:** Used to place components and to connect them to I/O ports.
4. **Properties Pallet:** Details for each I/O port for component specifications such as input or output, and voltage range.
5. **Menu:** Drop-down menus for user options.

5. Starting a Design

Before starting a design it is recommended to use the **General Parameter Configuration** screen (from the **Configuration** menu) to input the external voltages for I/O ports and voltage references (Figure 3). For details of applicable values, refer to the MAX11300 IC data sheet. The **Voltage**-related options are only for validation of the external power supply connected to the board.

FIELD	FUNCTION
AVSS_IO	Analog -V Supply for I/O Ports
AVDD_IO	Analog +V Supply for I/O Ports
DVDD	+V Digital Supply
AVDD	+V Analog Supply
DAC_REF	DAC External Voltage Reference
ADC_EXT_REF	ADC External Voltage Reference
ADC_INT_REF	ADC Internal Voltage Reference

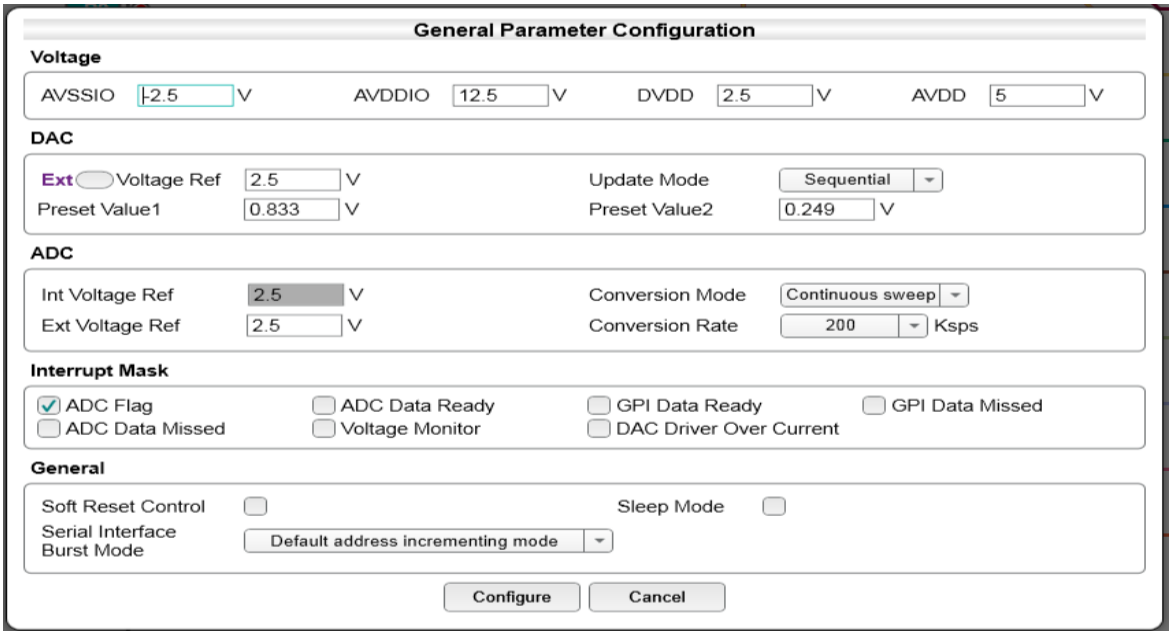

Figure 3. Configuration of External Voltages

6. PIXI (MAX11300/MAX11301) Device Area

The **Device Area** allows connection and configuration of PIXI components to any of the 20 available ports. The software provides simple drag-and-drop function to place the components within the boundary of PIXI device and connect wires to the required ports, as shown in Figure 4. To create a "wire," simply position the cursor over the port (cursor will change from "arrow" to "finger") and then drag the wire to the PIXI component with the left mouse button held.

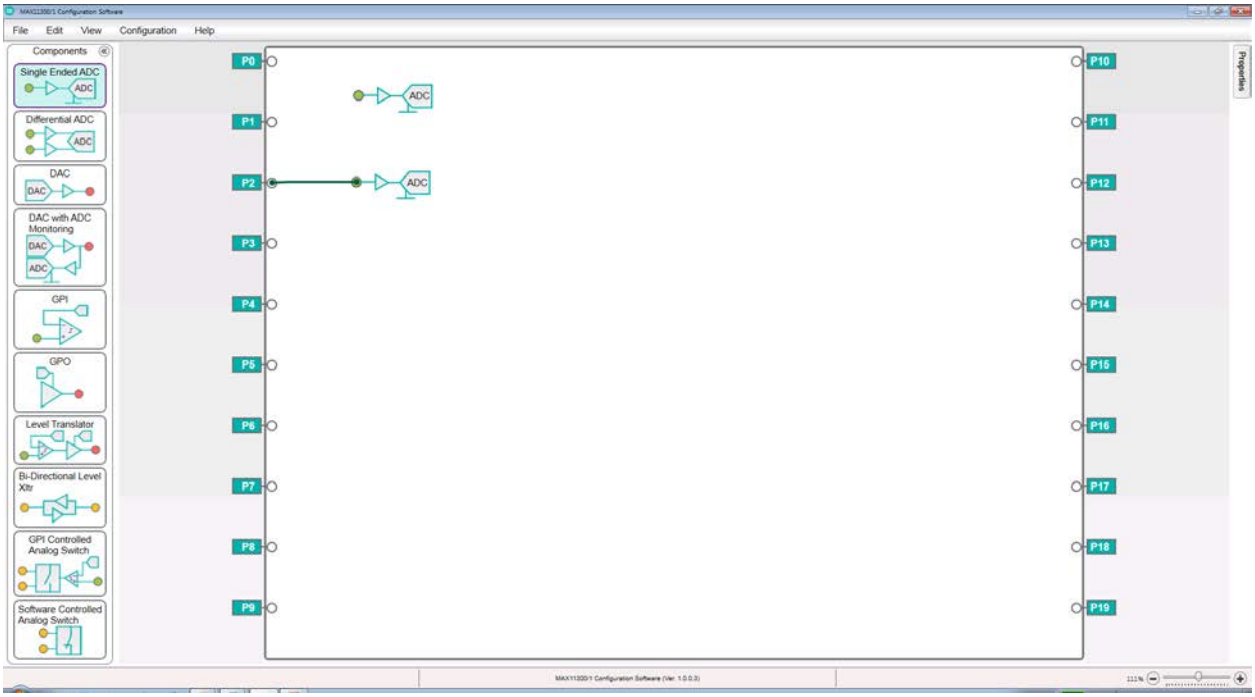

Figure 4. MAX11300 Configuration Software Design Example

The software provides user interface features that include functions like flip, copy, paste, and view properties of components on the device area. These functions are available by right-clicking on the component as shown in Figure 5.

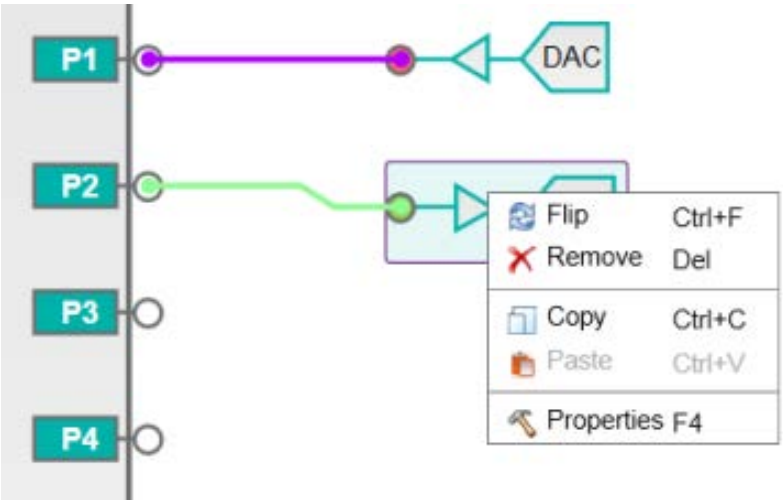

Figure 5. MAX11300 Component Function

Upon selecting **Properties** a new set of data is presented in the **PIXI Components' Properties Pallet**.

Right-clicking on a wire brings up the option to remove the connection, as shown in Figure 6.

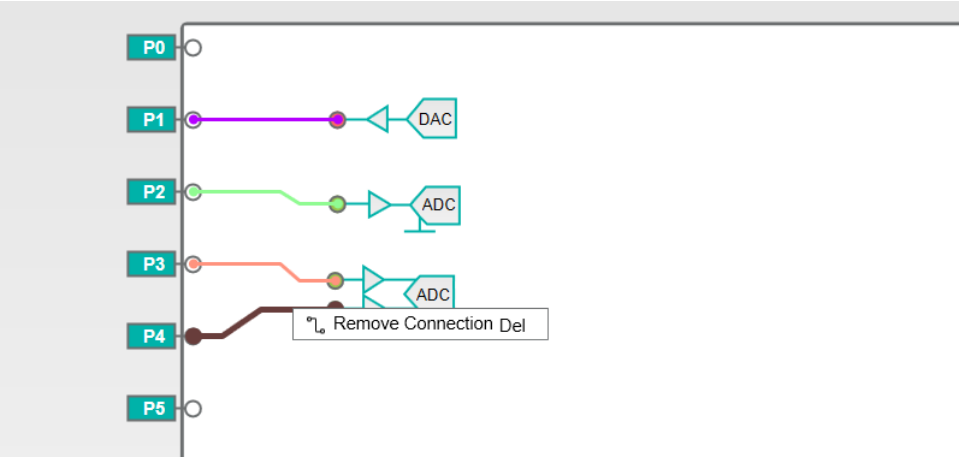

Figure 6. Removing a Connection

Note: An error is generated if a component is placed but it is not connected to any MAX11300 port.

7. MAX11300 Components

The MAX11300 device allows each port to be configured as a variety of functional components such as ADC, DAC, GPI. GPO, or level translator (Figure 7). Each component has a set of properties and default values for sub parameters.

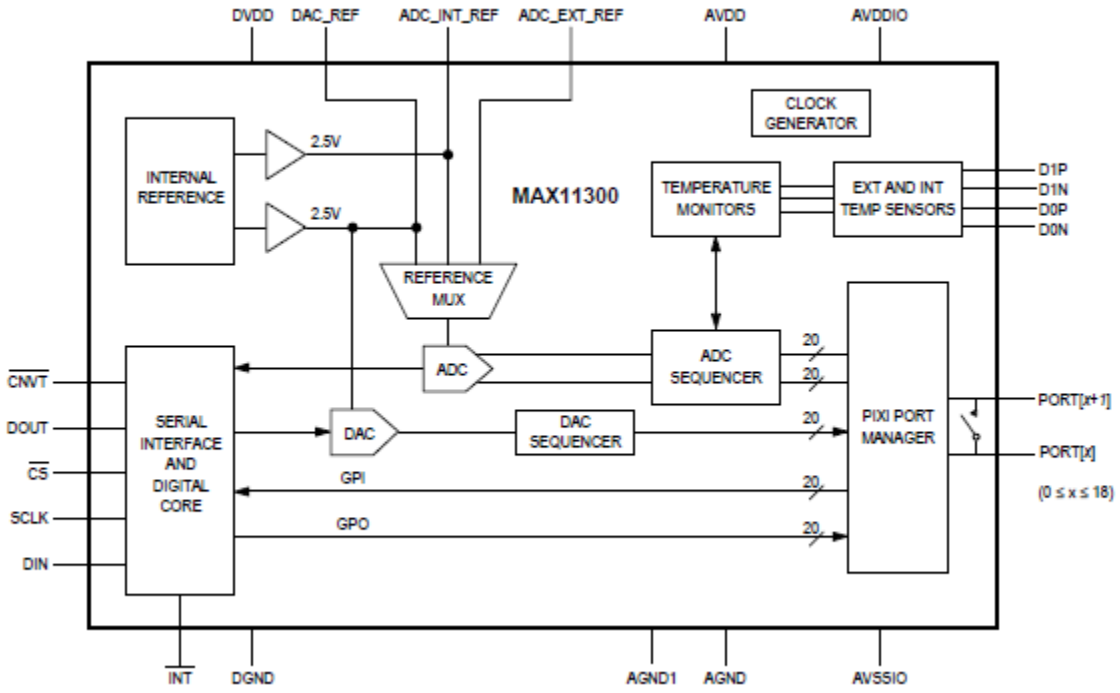

Figure 7. MAX11300 Functional Diagram

7.1 ADC with Single-Ended Input

Figure 8 represents a single-ended ADC. The readings are taken with reference to ground. The input can be connected to any MAX11300 port.

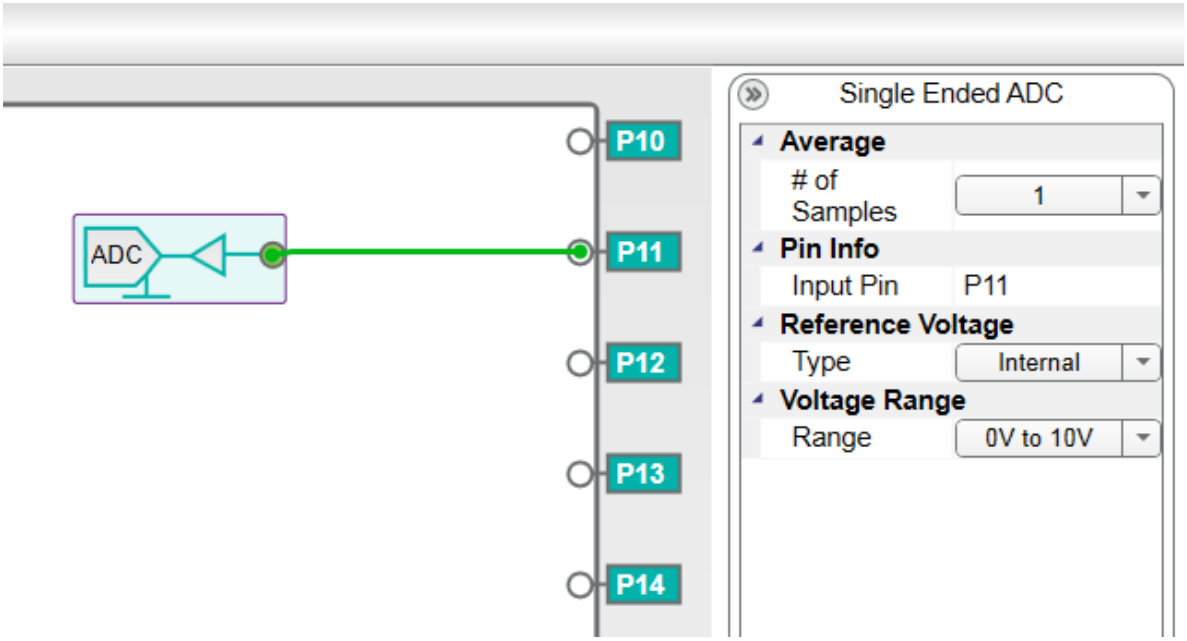

Figure 8. ADC with Single-Ended Input

NAME	DESCRIPTION	DEFAULT	RANGE
Average – # of Samples	Number of measurement average	1	1, 2, 4, 8, 16, 32, 64, 128
Pin Info – Input Pin	Port# the input is connected to	None	PORT0 to PORT19
Reference Voltage – Type	Is internal or external reference used	Internal	Internal / External
Voltage Range – Range	V_{REF} = 2.5V	0V to 10V	0V to 10V -5V to 5V -10V to 0V 0 to 2.5V

7.2 ADC with Differential Inputs

Figure 9 represents a differential ADC with two inputs. The two inputs can be connected to any I/O port. Note that the upper port is always the positive ADC input.

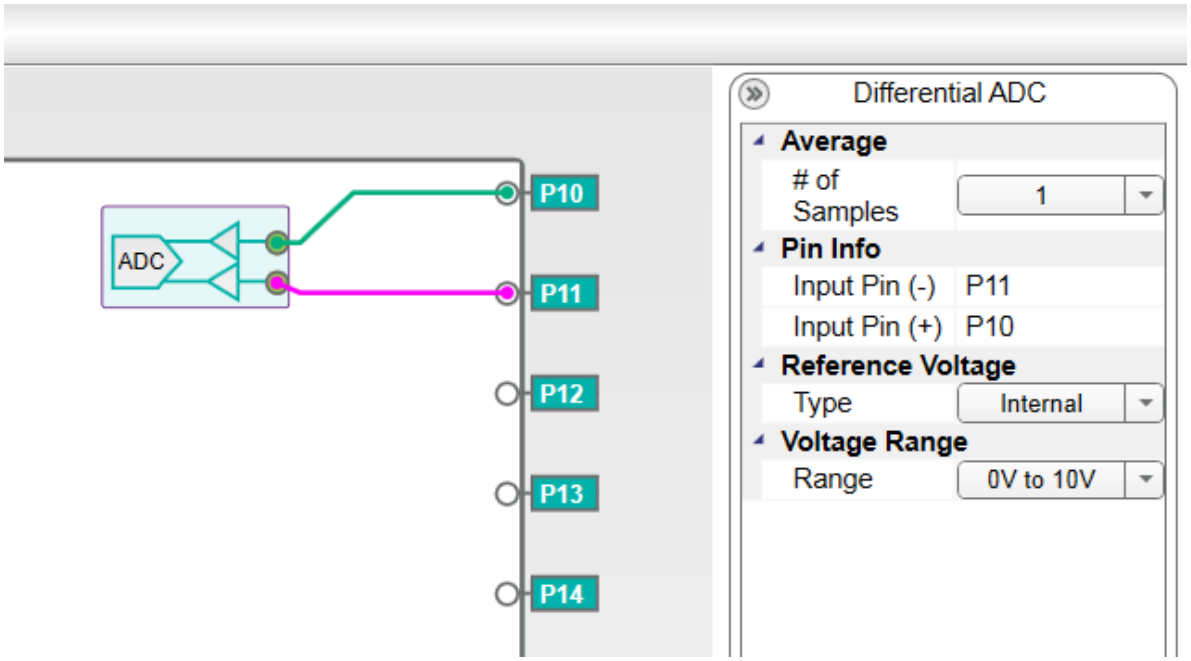

Figure 9. ADC with Differential Input

NAME	DESCRIPTION	DEFAULT	RANGE
Average – # of Samples	Number of measurement average	1	1, 2, 4, 8, 16, 32, 64, 128
Pin Info – Input Pin (-)	Port# the first input is connected to	None	PORT0 to PORT19
Pin Info – Input Pin (+)	Port# the second input is connected to	None	PORT0 to PORT19
Reference Voltage – Type	Is internal or external reference used	Internal	Internal / External
Voltage Range – Range	V_{REF} = 2.5V	0V to 10V	0V to 10V -5V to 5V -10V to 0V 0 to 2.5V

7.3 ADC with Pseudo-Differential Input Set by DAC

In pseudo-differential mode (Figure 10), one port produces the voltage applied to the negative input of the ADC while another port forms the positive input. The negative port (lower port on the differential ADC component symbol) is tied to a DAC port, with more than one differential ADC tied to the same DAC.

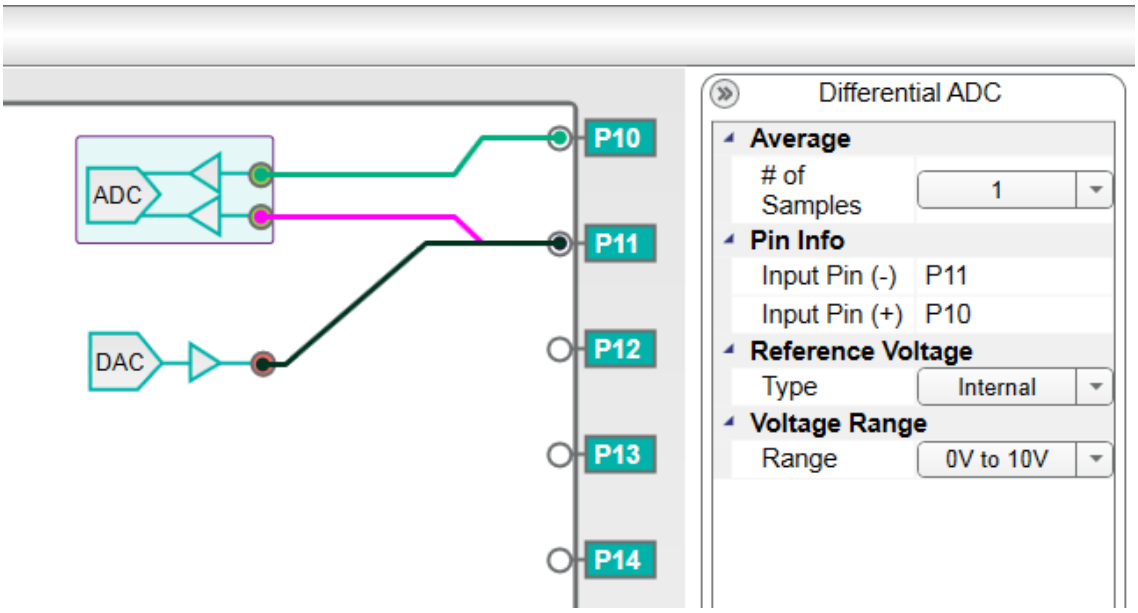

Figure 10. ADC with Pseudo-Differential Input Set by DAC

7.4 DAC

Figure 11 represents a DAC with voltage output. It can be used to set any voltage to any port, between the selected voltage ranges.

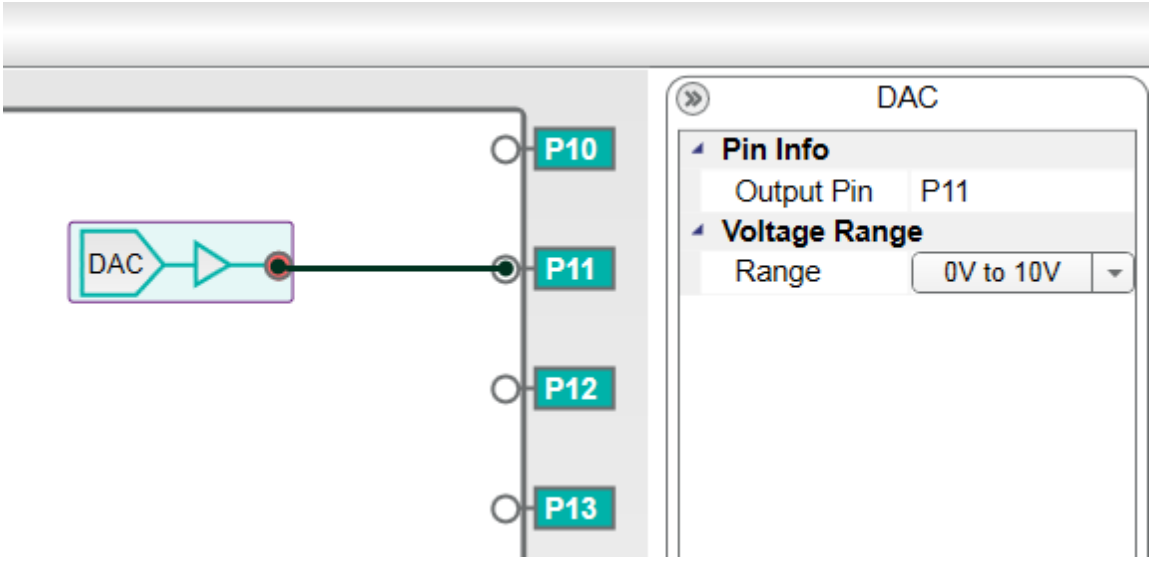

Figure 11. DAC Output

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Output Pin	Port# the output is connected to	None	PORT0 to PORT19
Voltage Range – Range	V_{REF} = 2.5V	0V to 10V	0V to 10V -5V to 5V -10V to 0V

7.5 DAC with ADC Monitoring

Figure 12 represents a DAC output with an ADC readback. This object allows a DAC to be monitored by a dedicated ADC. The input can be connected to any port.

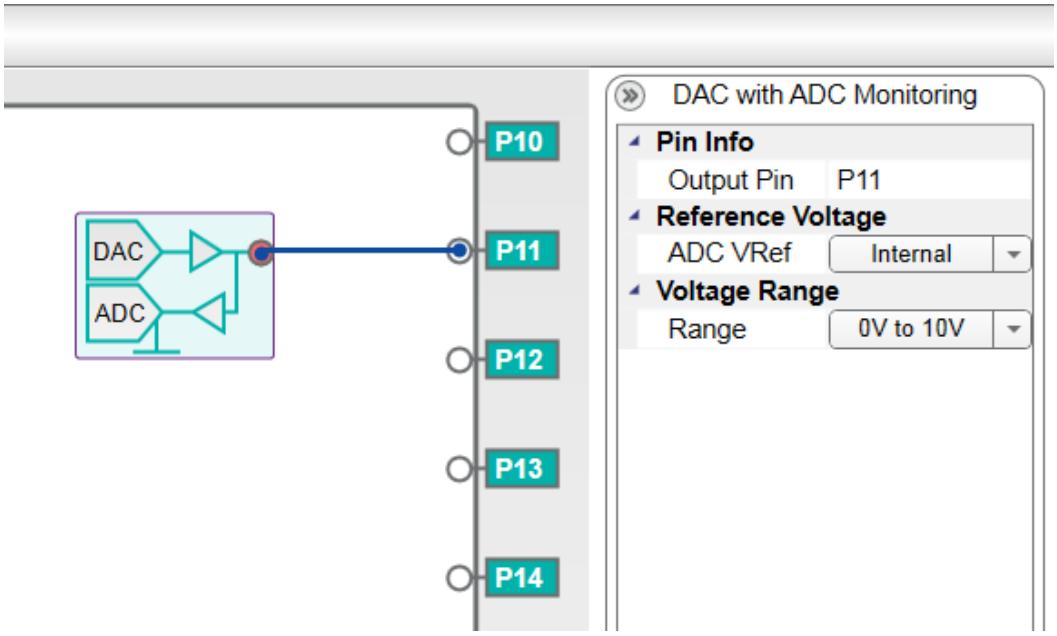

Figure 12. DAC with ADC Monitoring

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Output Pin	Port# the output is connected to	None	PORT0 to PORT19
Reference Voltage – ADC VRef	Internal or external reference for ADC	Internal	Internal / External
Voltage Range – Range	V_{REF} = 2.5V	0V to 10V	0V to 10V -5V to 5V -10V to 0V

7.6 GPI Mode

Figure 13 represents a general-purpose input (GPI) block. It consists of a comparator with one of the comparison line driven by a DAC. It can be connected to any PIXI I/O port.

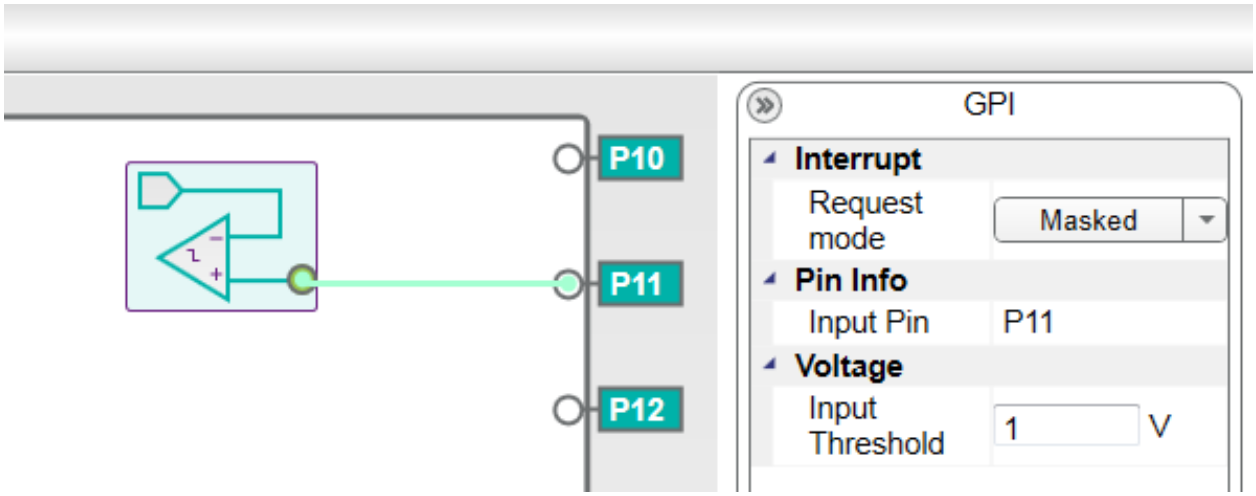

Figure 13. GPI Mode

NAME	DESCRIPTION	DEFAULT	RANGE
Interrupt – Request mode	How will the interrupt be generated	Masked (no interrupt generated)	Masked, Rising Edge, Falling Edge, Either Edge
Pin Info – Input Pin	Port# the input is connected to	None	PORT0 to PORT19
Voltage – Input Threshold	Threshold for the comparator (V_{REF} = 2.5V)	0V	0 to 2.5V

7.7 GPO Mode

Figure 14 represents a general-purpose output (GPO) block. It can be set to 0 or 1 with the high level set by the DAC.

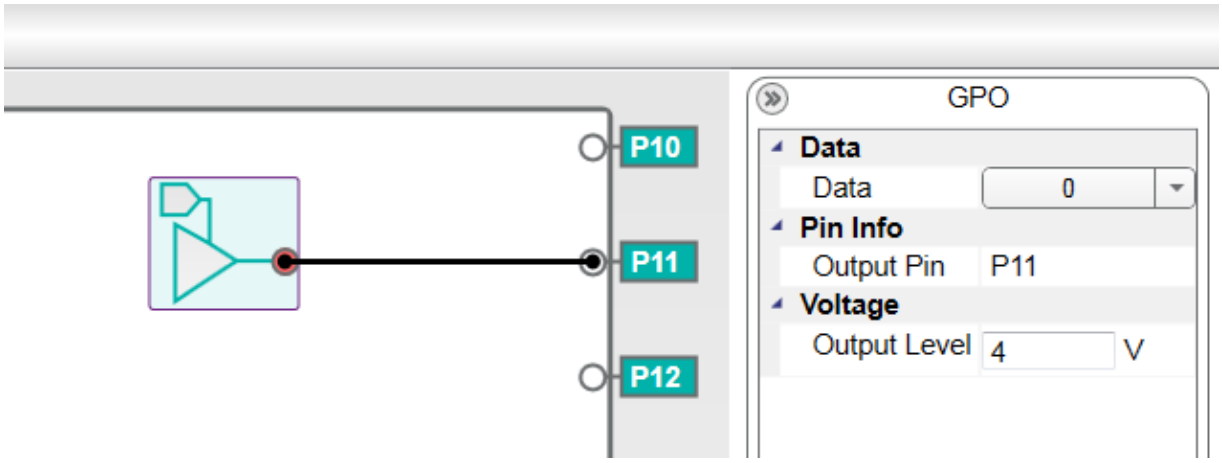

Figure 14. GPO Mode

NAME	DESCRIPTION	DEFAULT	RANGE
Data – Data	This is the logic value of the output	0 (LO)	0 / 1 (HI / LO)
Pin Info – Output Pin	Port# the output is connected to	None	PORT0 to PORT19
Voltage – Output Level	Voltage for the HI level (= 4x DAC value)	0V	0 to 2.5V

7.8 Unidirectional Level Translator

Figure 15 represents a level translator that is equivalent to a GPI connected to GPO back to back. The input side uses the DAC to set the comparison level, while the output side uses a DAC to set the HI level. The input and output ports can be connected to any port. Another feature it allows is the ability to invert the result of the comparison.

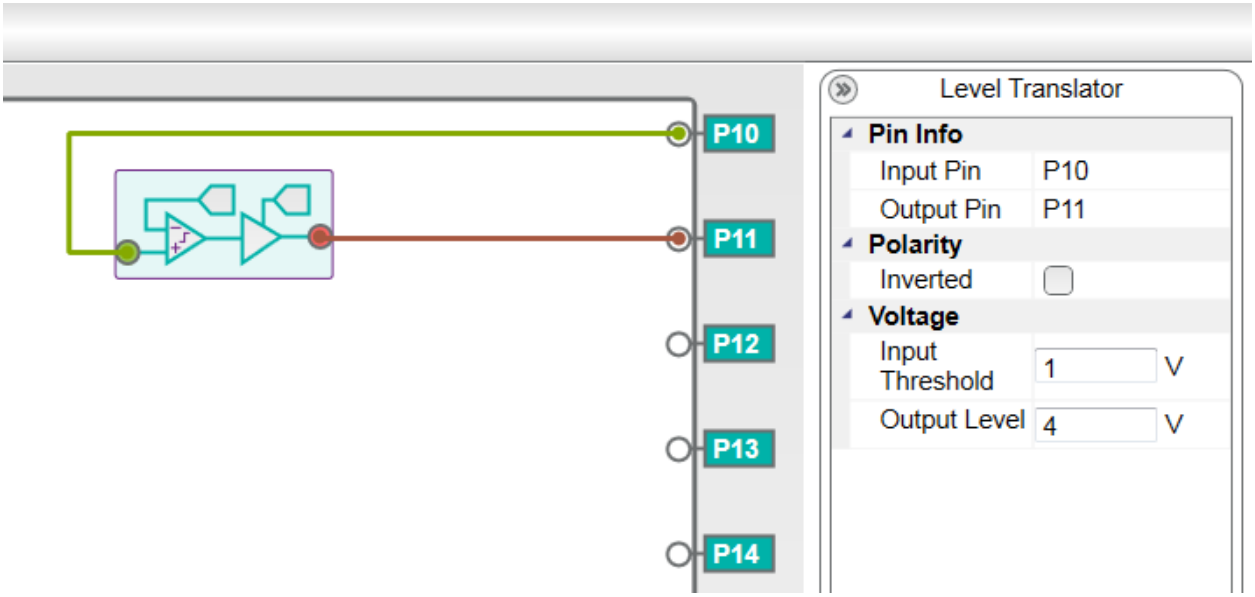

Figure 15. Unidirectional Level Translator

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Input Pin	Port# the input is connected to	none	PORT0 to PORT19
Pin Info – Output Pin	Port# the output is connected to	none	PORT0 to PORT19
Polarity	Flip the control logic	Non-Inverted	Non-Inverted / Inverted
Voltage – Input Threshold	Threshold for the comparator (V_{REF} = 2.5V)	0V	0 to 2.5V
Voltage – Output Level	Voltage for the HI level	0V	0 to 10V

7.9 Bidirectional Level Translator

Figure 16 represents a bidirectional level translator that allows level translation in both directions (equivalent to two GPI with open-collector outputs). This is ideal for interfacing buses with two different levels. Each port acts like an I/O port that can compare the voltage being applied and the ability to actively drive low only on the other side. This component mode requires adjacent ports.

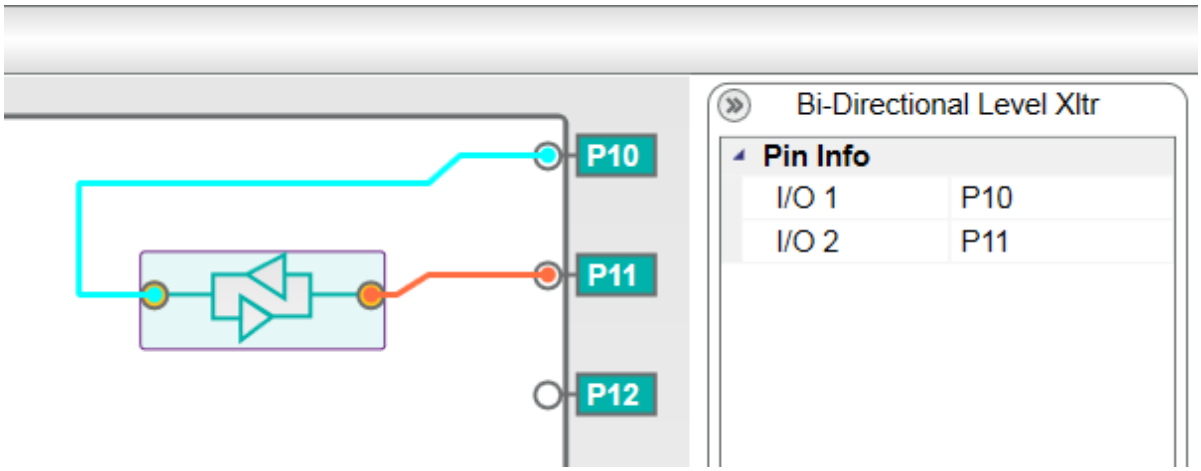

Figure 16. Bidirectional Level Translation

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Input Pin	Port# the input is connected to	None	PORT0 to PORT19
Pin Info – Output Pin	Port# the output is connected to	None	PORT0 to PORT19

7.10 GPI Controlled Analog Switch

Two adjacent PIXI ports can form a 60Ω analog switch that is controlled by two different schemes. Figure 17 represents an analog switch that is dynamically controlled by any other PIXI port configured as GPI. The switch component connects two adjacent ports, using a third port as a control line for the switch. The DAC on the control line sets the level that will trigger the opening or closing of the switch. The switch connections have to be adjacent ports with connection not allowed between PORT0 and PORT19. The control line does not have this restriction. The control line logic can also be flipped by just setting the inversion bit.

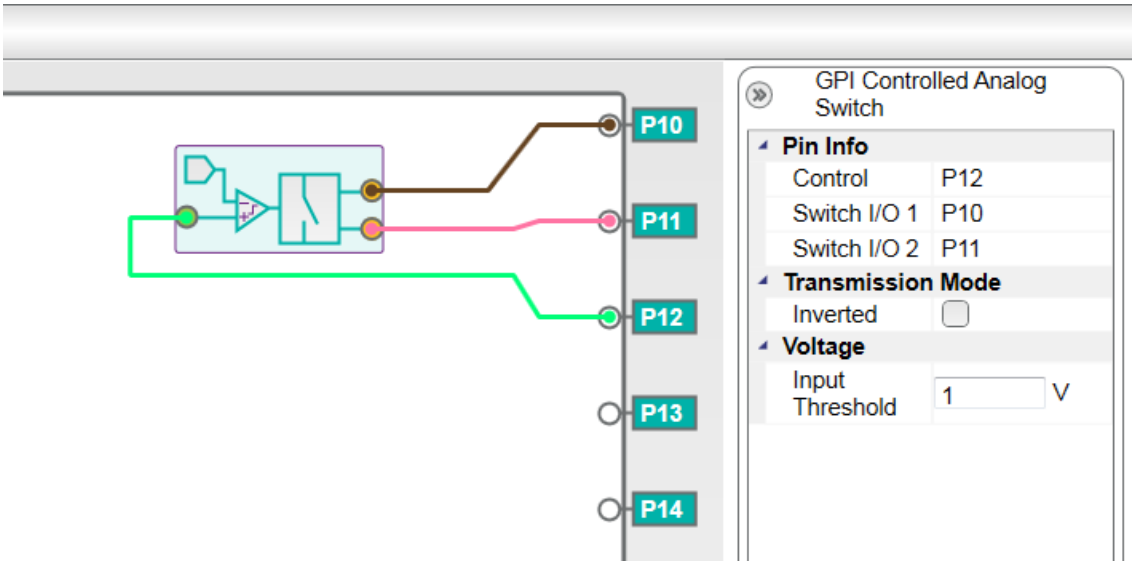

Figure 17. Externally Controlled Analog Switch

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Control	Port# for the control line input	None	PORT0 to PORT19
Pin Info – Switch I/O 1	Port# for one side of switch	None	PORT0 to PORT19
Pin Info – Switch I/O 2	Port# for other side of switch	None	PORT0 to PORT19
Transmission Mode– Inverted	Flip the control logic	Non-Inverted	Non-Inverted / Inverted
Voltage – Input Threshold	Threshold for the comparator (V_{REF} = 2.5V)	0V	0 to 2.5V

7.11 Software Controlled Analog Switch

The switch (Figure 18) is programmed to be permanently “ON” by configuring the corresponding PIXI port. This is similar to the Externally Controlled Analog Switch, except it is not controlled by a PIXI port. Use is restricted to adjacent ports. To turn the switch “OFF” the PIXI port needs to be reprogrammed to high-impedance (HI-Z) mode.

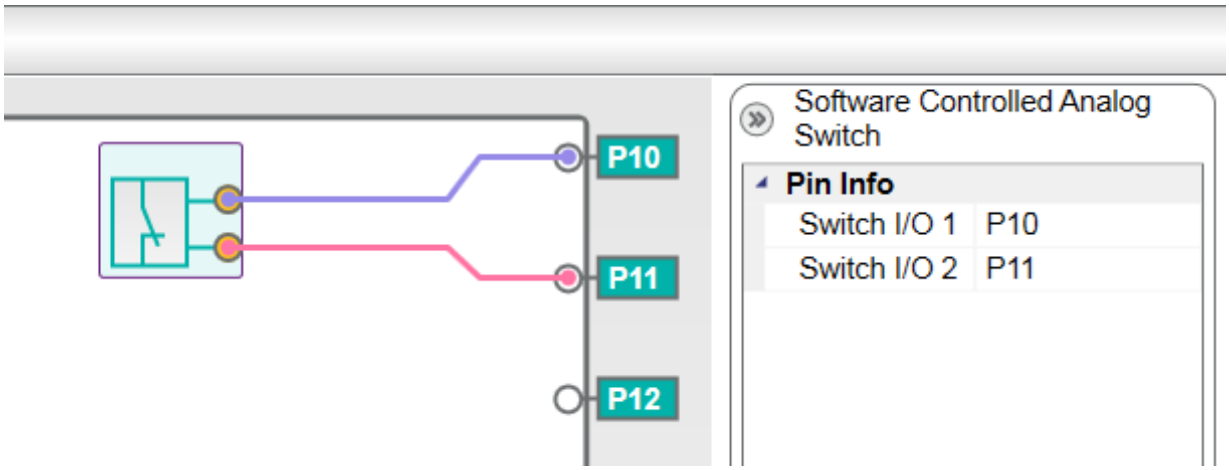

Figure 18. Internally Controlled Analog Switch

NAME	DESCRIPTION	DEFAULT	RANGE
Pin Info – Switch I/O 1	Port# for one side of switch	None	PORT0 to PORT19
Pin Info – Switch I/O 2	Port# for one side of switch	None	PORT0 to PORT19

8. PIXI Component Properties Pallet

The software provides easy configuration of PIXI components through the **Property Pallet**. Each PIXI component has a set of **Properties** that could be different from other components. The **Property Pallet** shows properties of the selected component only.

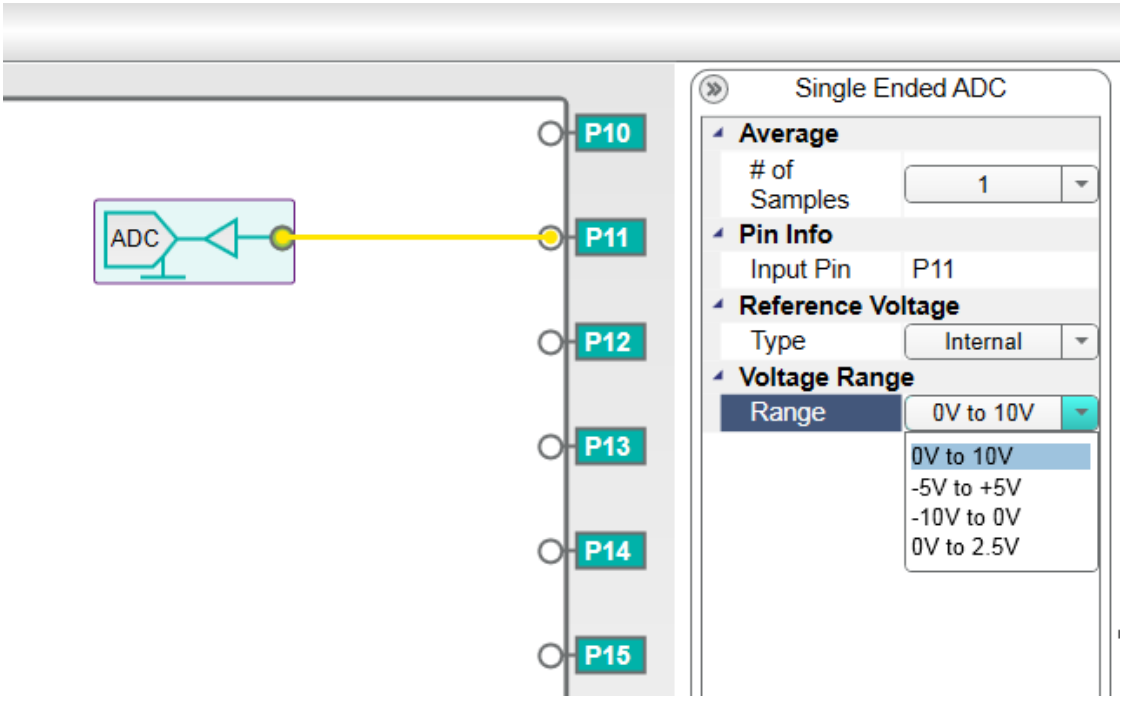

Figure 19. Properties Pallet

All the settings in properties pallet, for each component, are reflected in the register map generated by the software.

Note: The range of analog components such as ADC or DAC mainly depends on the AVDD_IO and AVSS_IO power supply. If different components have different ranges, then +12.5V and -12.5V supply is recommended on AVDD_IO and AVSS_IO, respectively.

9. Software Main Menu

Figure 20 shows the PIXI Configuration Software Main Menu contents.

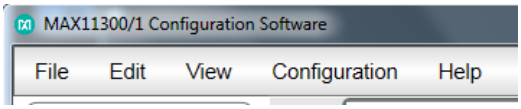

Figure 20. Configuration Software Menu

9.1 File Menu

The **File** menu (Figure 21) provides options to make new, open already saved, or save current PIXI configuration. It also provides option to generate the register map through the configuration.

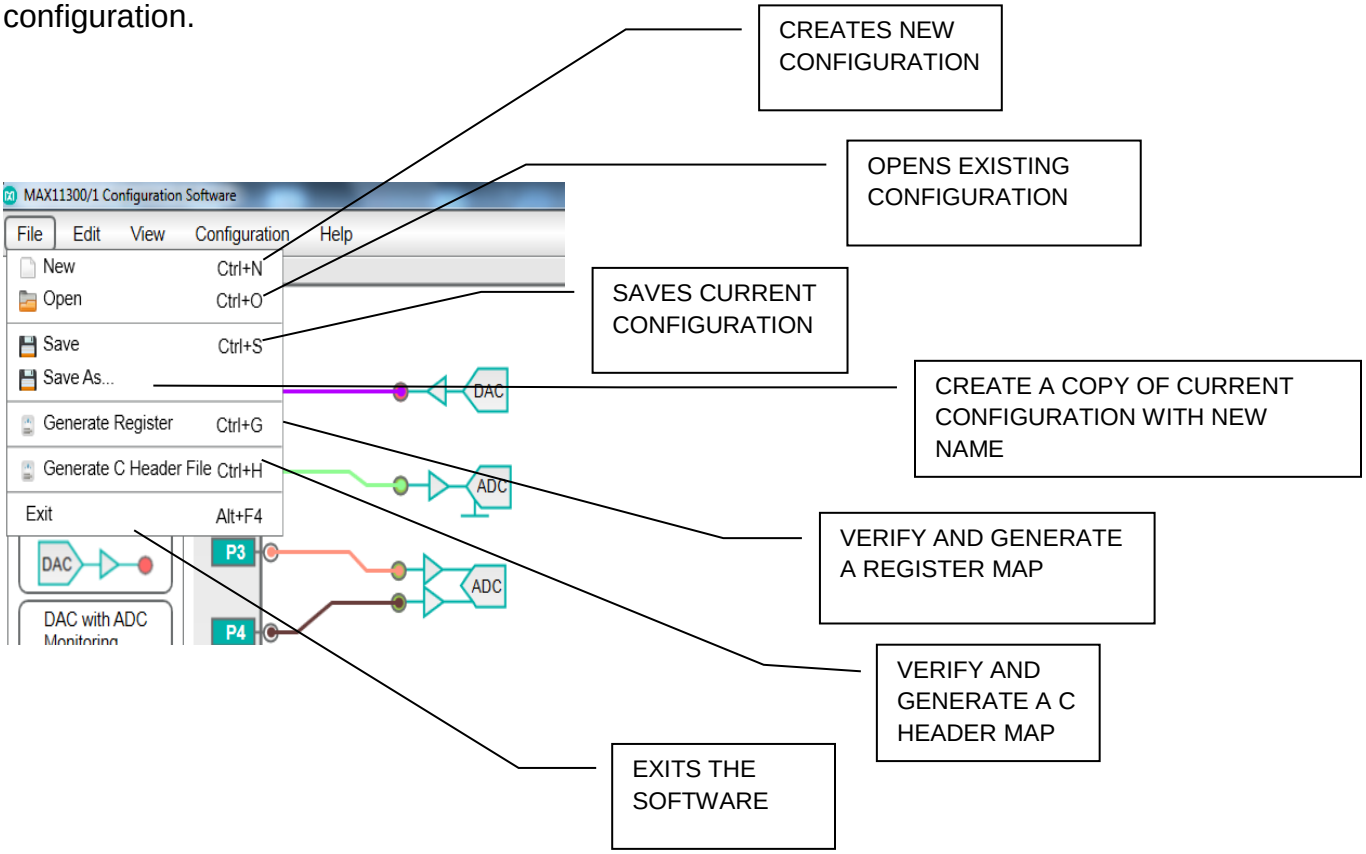

Figure 21. File Options

9.2 Edit Menu

The **Edit** menu (Figure 22) provides option to undo, redo any changes, and copy and paste any components in device area.

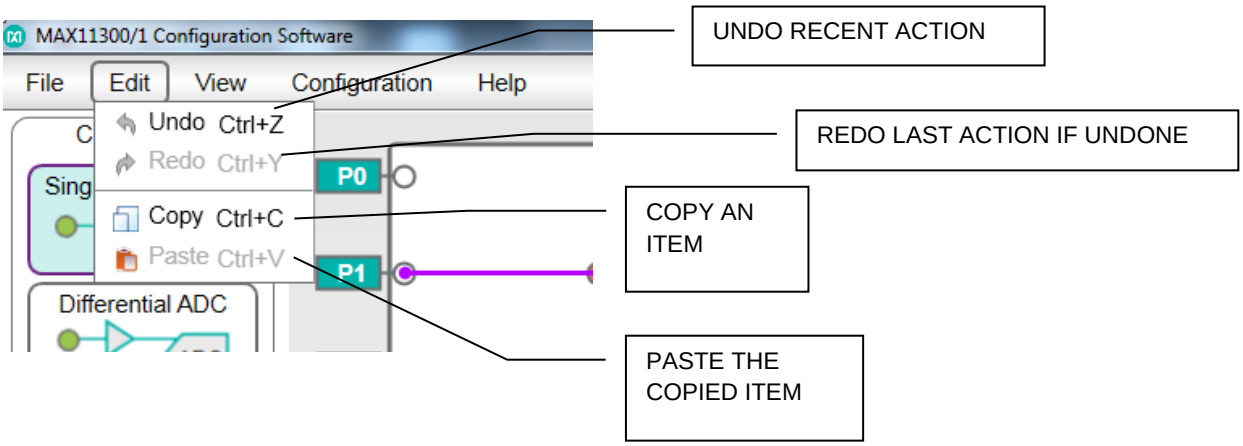

Figure 22. Edit Options

9.3 View Menu

The **View** menu (Figure 23) has options related to device area. These options include zoom functions, fit to screen, and full-screen view function.

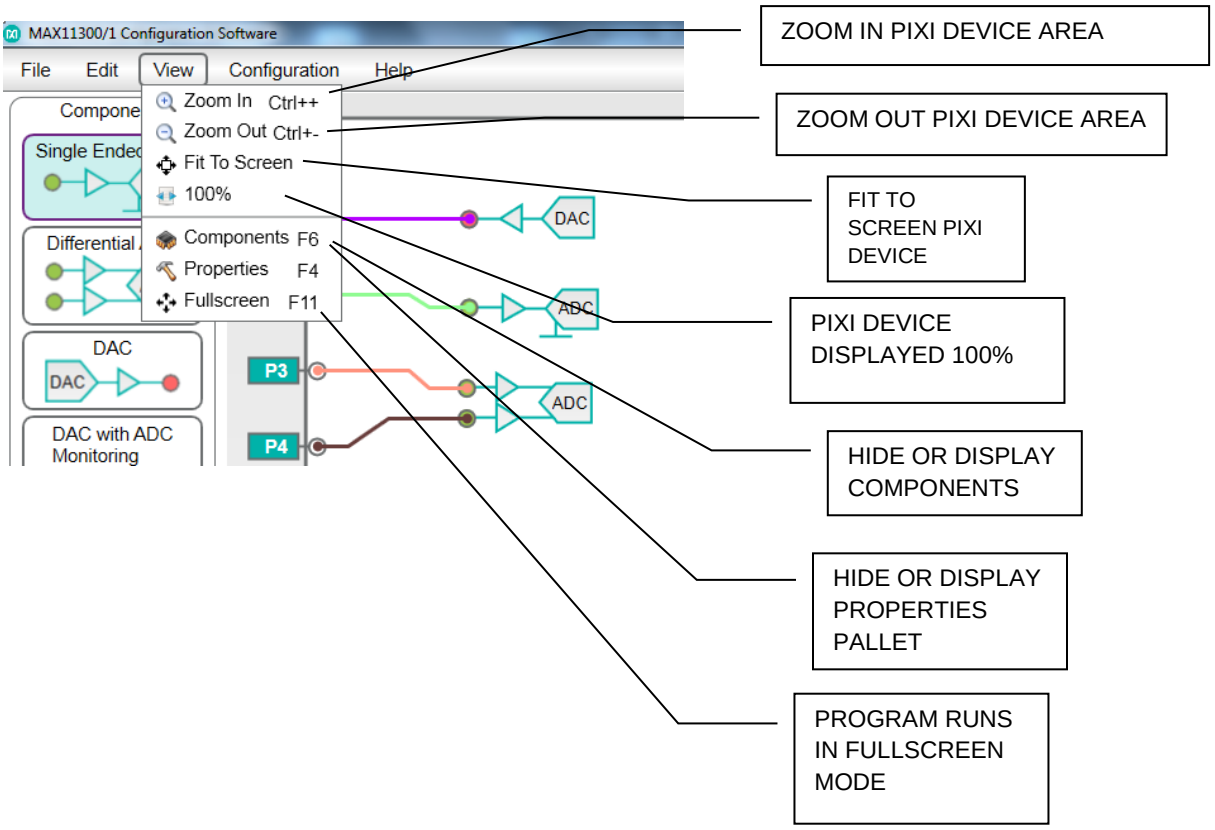

Figure 23. View Options

10. Configuration Menu

The **General Parameter Configuration** (Figure 24) and **Temperature Sensor Configuration** (Figure 25) menu provides option to configure MAX11300 device functions. These options include temperature sensor configuration, ADC and DAC reference and conversion modes adjustment, MAX11300 interrupt-related functions, and MAX11300 general functions including power mode and interface selection.

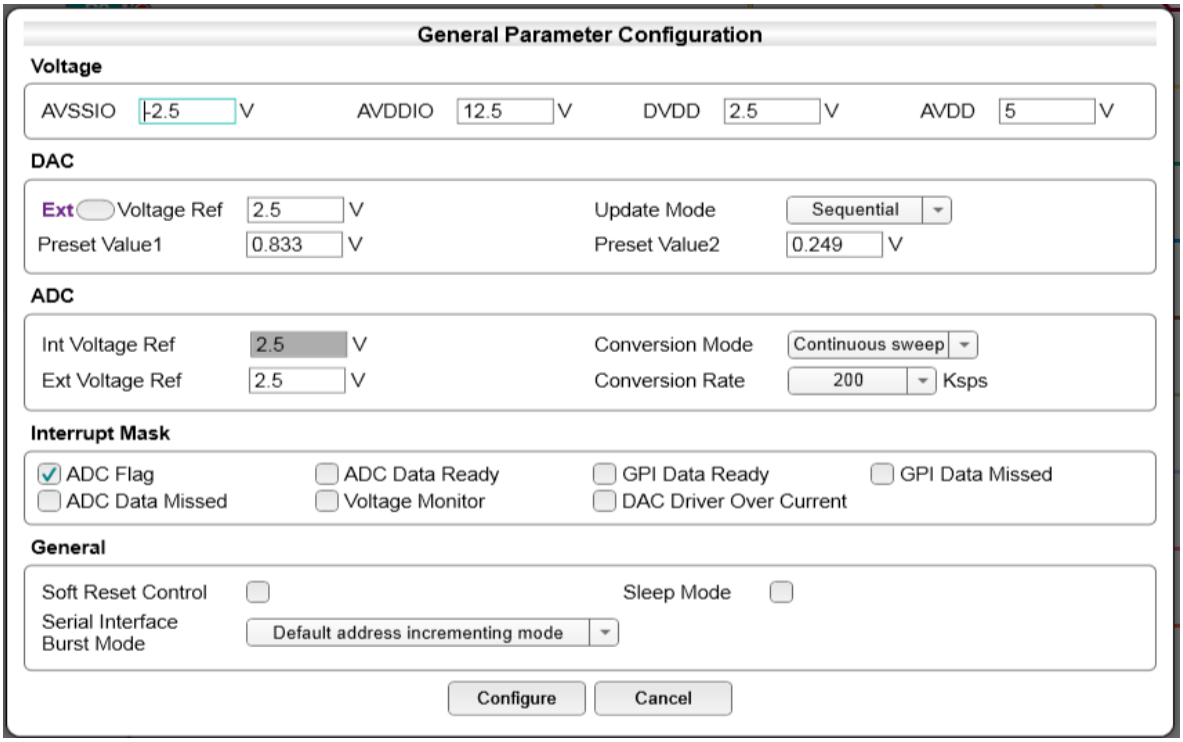

Figure 24. General Parameter Configuration Options

The **Voltage**-related options are only for validation of the external power supply connected to the board.

Through the **ADC** and **DAC** option pallet, voltage reference and conversion modes can be selected. Internal reference is selected by default for ADC and DAC. DAC can be configured for four update modes, whereas ADC can be configured for four conversion modes and four conversion rates.

Interrupt Mask provides the option to enable/disable all MAX11300 interrupts.

Through the **General** setting area, MAX11300 operation mode and serial interface options can be selected.

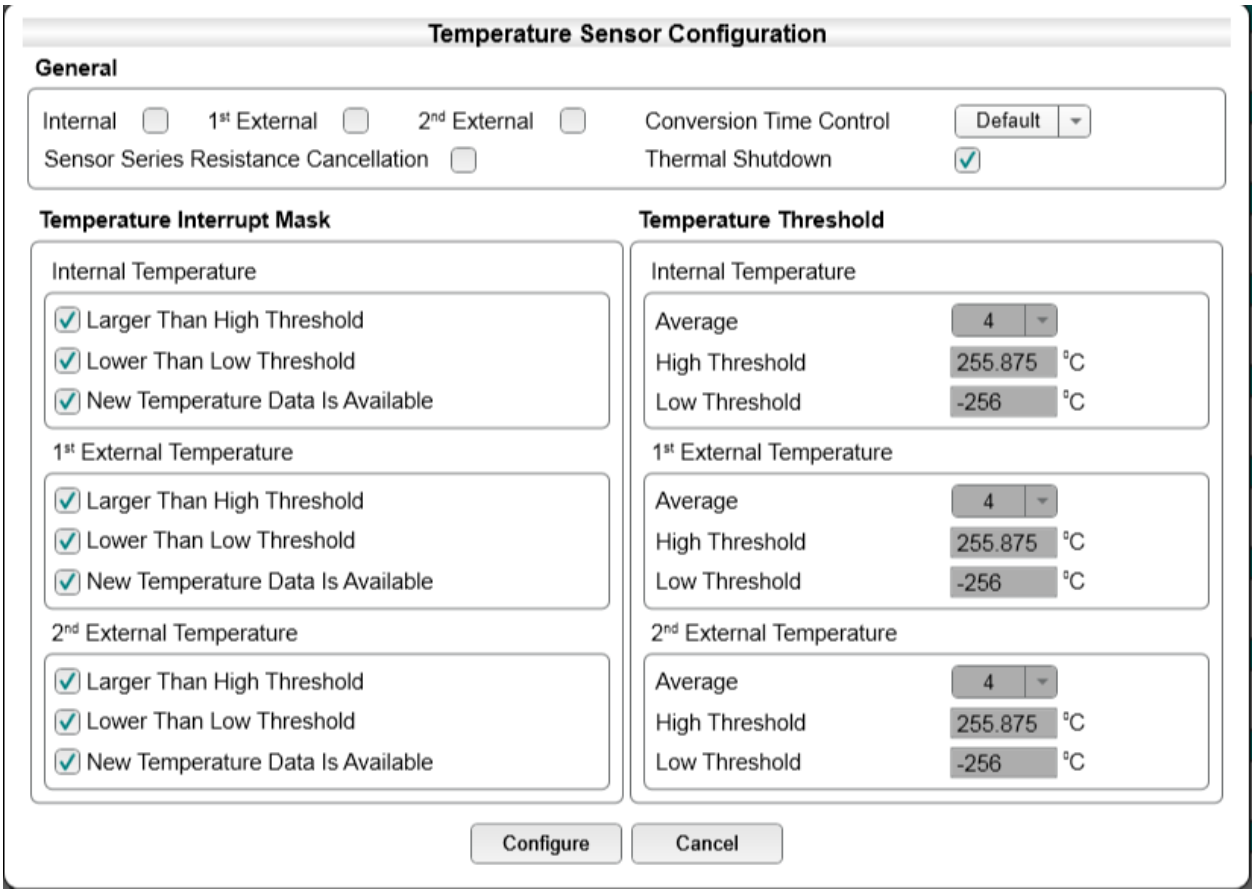

Figure 25. Temperature Sensor Configuration Options

Through the **General** setting area, internal temperature sensor and external temperature can be enabled and the threshold can be changed in **Temperate Threshold** section. **Temperature Interrupt Mask** provides the option to enable/disable all temperature interrupts.

Through **Conversion Time Control** option, the conversion time can be extended.

Through **Sensor Series Resistance Cancellation** option, MAX11300 can eliminate temperature reading error for parasitic series resistances up to10Ω.

Through the **Thermal Shutdown** option, when the temperature meets the shutdown requirement, the device can be reset, bringing all channels to high-impedance mode and setting all registers to their default value.

When all settings are made and the Configure button is pressed, the user selects **Generate Register** in the **File** menu to allow the software to generate a register map with the updated values in hex format. This register map becomes the bitstream that is programmed into the IC device in the end equipment to configure it for the required mode of operation. The user can also select **Generate C Header File** in the **File** menu to allow the software to generate a global file including C function declarations and components definitions to be shared in a module. By including the header file in the program with C preprocessing directive #include, the user can make sure they are using the same definitions for all of the program components.

10.1 Device Zoom Function

The software provides feature to zoom in and out MAX11300 device area through the drag button, as shown in Figure 26.

Figure 26. Device Zoom Function

This could also be achieved through mouse wheel or through **View** options in the **Main** menu.

11. Register Map

Figure 27 is an example of a register map generated as an output from the PIXI Configuration Software.

Created on	21/10/2014 07:04		
Name	Address	Value	Description
gpo_data_15_to_0	0x0D	0x0000	GPO data for PIXI ports 15 to 0
gpo_data_19_to_16	0x0E	0x0000	GPO data for PIXI ports 19 to 16
device_control	0x10	0x0703	Device main control register
interrupt_mask	0x11	0x6d81	Interrupt mask register
gpi_irqmode_7_to_0	0x12	0x0000	GPI port 0 to 7 mode register
gpi_irqmode_15_to_8	0x13	0x0000	GPI port 8 to 15 mode register
gpi_irqmode_19_to_16	0x14	0x0039	GPI port 16 to 19 mode register
dac_preset_data_1	0x16	0x0155	DAC preset data #1
dac_preset_data_2	0x17	0x0066	DAC preset data #2
tmp_mon_cfg	0x18	0x0000	Temperature monitor configuration
tmp_mon_int_hi_thresh	0x19	0x02a8	Internal temperature monitor high threshold
tmp_mon_int_lo_thresh	0x1A	0x0ec0	Internal temperature monitor low threshold
tmp_mon_ext1_hi_thresh	0x1B	0x039a	1st external temperature monitor high threshold
tmp_mon_ext1_lo_thresh	0x1C	0x0090	1st external temperature monitor low threshold
tmp_mon_ext2_hi_thresh	0x1D	0x07ff	2nd external temperature monitor high threshold
tmp_mon_ext2_lo_thresh	0x1E	0x0800	2nd external temperature monitor low threshold
port_cfg_00	0x20	0x6100	Configuration register for PIXI port 0
port_cfg_01	0x21	0x1000	Configuration register for PIXI port 1
port_cfg_02	0x22	0x2000	Configuration register for PIXI port 2
port_cfg_03	0x23	0x0000	Configuration register for PIXI port 3
port_cfg_04	0x24	0x3000	Configuration register for PIXI port 4
port_cfg_05	0x25	0x5100	Configuration register for PIXI port 5
port_cfg_06	0x26	0x6100	Configuration register for PIXI port 6
port_cfg_07	0x27	0x7400	Configuration register for PIXI port 7
port_cfg_08	0x28	0x8409	Configuration register for PIXI port 8
port_cfg_09	0x29	0x9400	Configuration register for PIXI port 9
port_cfg_10	0x2A	0x810b	Configuration register for PIXI port 10
port_cfg_11	0x2B	0xa100	Configuration register for PIXI port 11

port_cfg_12	0x2C	0xb001	Configuration register for PIXI port 12
port_cfg_13	0x2D	0x0000	Configuration register for PIXI port 13
port_cfg_14	0x2E	0x0000	Configuration register for PIXI port 14
port_cfg_15	0x2F	0x1000	Configuration register for PIXI port 15
port_cfg_16	0x30	0x1000	Configuration register for PIXI port 16
port_cfg_17	0x31	0x1000	Configuration register for PIXI port 17
port_cfg_18	0x32	0x1000	Configuration register for PIXI port 18
port_cfg_19	0x33	0x0000	Configuration register for PIXI port 19
dac_data_port_00	0x60	0x0000	DAC data register for PIXI port 0
dac_data_port_01	0x61	0x0666	DAC data register for PIXI port 1
dac_data_port_02	0x62	0x0000	DAC data register for PIXI port 2
dac_data_port_03	0x63	0x0000	DAC data register for PIXI port 3
dac_data_port_04	0x64	0x0666	DAC data register for PIXI port 4
dac_data_port_05	0x65	0x0000	DAC data register for PIXI port 5
dac_data_port_06	0x66	0x0000	DAC data register for PIXI port 6
dac_data_port_07	0x67	0x0000	DAC data register for PIXI port 7
dac_data_port_08	0x68	0x0000	DAC data register for PIXI port 8
dac_data_port_09	0x69	0x0000	DAC data register for PIXI port 9
dac_data_port_10	0x6A	0x0000	DAC data register for PIXI port 10
dac_data_port_11	0x6B	0x0000	DAC data register for PIXI port 11
dac_data_port_12	0x6C	0x0000	DAC data register for PIXI port 12
dac_data_port_13	0x6D	0x0000	DAC data register for PIXI port 13
dac_data_port_14	0x6E	0x0000	DAC data register for PIXI port 14
dac_data_port_15	0x6F	0x0666	DAC data register for PIXI port 15
dac_data_port_16	0x70	0x0666	DAC data register for PIXI port 16
dac_data_port_17	0x71	0x0666	DAC data register for PIXI port 17
dac_data_port_18	0x72	0x0666	DAC data register for PIXI port 18
dac_data_port_19	0x73	0x0000	DAC data register for PIXI port 19

Figure 27. Register Map for MAX11300

12. C Header File

Figure 28 is an example of C header file generated by the PIXI Configuration Software.

```
/// Generated by: MAX11300 Configuration Software (Ver. 1.0.0.3) 20/10/2014 11:34
#ifndef _MAX11300_DESIGNVALUE_H_
#define _MAX11300_DESIGNVALUE_H_

/// SPI first byte when writing MAX11300 (7-bit address in bits 0x7E; LSB=0 for write)
#define MAX11300Addr_SPI_Write(RegAddr) ( (RegAddr << 1)     )

/// SPI first byte when reading MAX11300 (7-bit address in bits 0x7E; LSB=1 for read)
#define MAX11300Addr_SPI_Read(RegAddr)  ( (RegAddr << 1) | 1 )

/// MAX11300EVKIT Register Addresses
typedef enum MAX11300RegAddressEnum {

/// 0x00 r/o dev_id Device Identification
dev_id = 0x00,

/// 0x01 r/o interrupt_flag Interrupt flags
interrupt_flag = 0x01,

/// 0x02 r/o adc_status_15_to_0 new ADC data available
adc_status_15_to_0 = 0x02,

/// 0x03 r/o adc_status_19_to_16 new ADC data available
adc_status_19_to_16 = 0x03,

/// 0x04 r/o dac_oi_status_15_to_0 DAC Overcurrent Interrupt
dac_oi_status_15_to_0 = 0x04,

/// 0x05 r/o dac_oi_status_19_to_16 DAC Overcurrent Interrupt
dac_oi_status_19_to_16 = 0x05,

/// 0x06 r/o gpi_status_15_to_0 GPI event ready
 gpi_status_15_to_0 = 0x06,

/// 0x07 r/o gpi_status_19_to_16 GPI event ready
gpi_status_19_to_16 = 0x07,

/// 0x08 r/o tmp_int_data Internal Temeprature
tmp_int_data = 0x08,

/// 0x09 r/o tmp_ext1_data External Temperature D0P/D0N
tmp_ext1_data = 0x09,

/// 0x0a r/o tmp_ext2_data External Temperature D1P/D1N
tmp_ext2_data = 0x0a,

/// 0x0b r/o gpi_data_15_to_0 GPI input ports data
gpi_data_15_to_0 = 0x0b,

/// 0x0c r/o gpi_data_19_to_16 GPI input ports data
gpi_data_19_to_16 = 0x0c,
```

```
/// 0x0d r/w gpo_data_15_to_0 GPO output ports data
gpo_data_15_to_0 = 0x0d,

/// 0x0e r/w gpo_data_19_to_16 GPO output ports data
gpo_data_19_to_16 = 0x0e,

/// 0x0f r/o reserved_0F reserved
reserved_0F = 0x0f,

/// 0x10 r/w device_control Global device control register
device_control = 0x10,

/// 0x11 r/w interrupt_mask interrupt mask (1 = disable interrupt source)
interrupt_mask = 0x11,

/// 0x12 r/w gpi_irqmode_7_to_0 xxxxxx
gpi_irqmode_7_to_0 = 0x12,

/// 0x13 r/w gpi_irqmode_15_to_8 xxxxxx
gpi_irqmode_15_to_8 = 0x13,

/// 0x14 r/w gpi_irqmode_19_to_16 xxxxxx
gpi_irqmode_19_to_16 = 0x14,

/// 0x15 r/w gpi_irqmode_31_to_24 xxxxxx
gpi_irqmode_31_to_24 = 0x15,

/// 0x16 r/w dac_preset_data_1 DAC preset activated by <see cref="device_control"/>
dac_preset_data_1 = 0x16,

/// 0x17 r/w dac_preset_data_2 DAC preset activated by <see cref="device_control"/>
dac_preset_data_2 = 0x17,

/// 0x18 r/w tmp_mon_cfg Temperautre Monitor Configuration
tmp_mon_cfg = 0x18,

/// 0x19 r/w tmp_mon_int_hi_thresh Internal Temeprature Hot Threshold
tmp_mon_int_hi_thresh = 0x19,

/// 0x1a r/w tmp_mon_int_lo_thresh Internal Temeprature Cold Threshold
tmp_mon_int_lo_thresh = 0x1a,

/// 0x1b r/w tmp_mon_ext1_hi_thresh External Temperature D0P/D0N Hot Threshold
tmp_mon_ext1_hi_thresh = 0x1b,

/// 0x1c r/w tmp_mon_ext1_lo_thresh External Temperature D0P/D0N Cold Threshold
tmp_mon_ext1_lo_thresh = 0x1c,

/// 0x1d r/w tmp_mon_ext2_hi_thresh External Temperature D1P/D1N Hot Threshold
tmp_mon_ext2_hi_thresh = 0x1d,

/// 0x1e r/w tmp_mon_ext2_lo_thresh External Temperature D1P/D1N Cold Threshold
tmp_mon_ext2_lo_thresh = 0x1e,

/// 0x1f r/w reserved_1F reserved
reserved_1F = 0x1f,
```

```
/// 0x20 r/w port_cfg_00 PIXI Port 0 configuration register
port_cfg_00 = 0x20,

/// 0x21 r/w port_cfg_01 PIXI Port 1 configuration register
port_cfg_01 = 0x21,

/// 0x22 r/w port_cfg_02 PIXI Port 2 configuration register
port_cfg_02 = 0x22,

/// 0x23 r/w port_cfg_03 PIXI Port 3 configuration register
port_cfg_03 = 0x23,

/// 0x24 r/w port_cfg_04 PIXI Port 4 configuration register
port_cfg_04 = 0x24,

/// 0x25 r/w port_cfg_05 PIXI Port 5 configuration register
port_cfg_05 = 0x25,

/// 0x26 r/w port_cfg_06 PIXI Port 6 configuration register
port_cfg_06 = 0x26,

/// 0x27 r/w port_cfg_07 PIXI Port 7 configuration register
port_cfg_07 = 0x27,

/// 0x28 r/w port_cfg_08 PIXI Port 8 configuration register
port_cfg_08 = 0x28,

/// 0x29 r/w port_cfg_09 PIXI Port 9 configuration register
port_cfg_09 = 0x29,

/// 0x2a r/w port_cfg_10 PIXI Port 10 configuration register
port_cfg_10 = 0x2a,

/// 0x2b r/w port_cfg_11 PIXI Port 11 configuration register
port_cfg_11 = 0x2b,

/// 0x2c r/w port_cfg_12 PIXI Port 12 configuration register
port_cfg_12 = 0x2c,

/// 0x2d r/w port_cfg_13 PIXI Port 13 configuration register
port_cfg_13 = 0x2d,

/// 0x2e r/w port_cfg_14 PIXI Port 14 configuration register
port_cfg_14 = 0x2e,

/// 0x2f r/w port_cfg_15 PIXI Port 15 configuration register
port_cfg_15 = 0x2f,

/// 0x30 r/w port_cfg_16 PIXI Port 16 configuration register
port_cfg_16 = 0x30,

/// 0x31 r/w port_cfg_17 PIXI Port 17 configuration register
port_cfg_17 = 0x31,

/// 0x32 r/w port_cfg_18 PIXI Port 18 configuration register
port_cfg_18 = 0x32,

/// 0x33 r/w port_cfg_19 PIXI Port 19 configuration register
```

```
port_cfg_19 = 0x33,

/// 0x40 r/o adc_data_port_00 PIXI Port 0 Analog to Digital Converter register
adc_data_port_00 = 0x40,

/// 0x41 r/o adc_data_port_01 PIXI Port 1 Analog to Digital Converter register
adc_data_port_01 = 0x41,

/// 0x42 r/o adc_data_port_02 PIXI Port 2 Analog to Digital Converter register
adc_data_port_02 = 0x42,

/// 0x43 r/o adc_data_port_03 PIXI Port 3 Analog to Digital Converter register
adc_data_port_03 = 0x43,

/// 0x44 r/o adc_data_port_04 PIXI Port 4 Analog to Digital Converter register
adc_data_port_04 = 0x44,

/// 0x45 r/o adc_data_port_05 PIXI Port 5 Analog to Digital Converter register
adc_data_port_05 = 0x45,

/// 0x46 r/o adc_data_port_06 PIXI Port 6 Analog to Digital Converter register
adc_data_port_06 = 0x46,

/// 0x47 r/o adc_data_port_07 PIXI Port 7 Analog to Digital Converter register
adc_data_port_07 = 0x47,

/// 0x48 r/o adc_data_port_08 PIXI Port 8 Analog to Digital Converter register
adc_data_port_08 = 0x48,

/// 0x49 r/o adc_data_port_09 PIXI Port 9 Analog to Digital Converter register
adc_data_port_09 = 0x49,

/// 0x4a r/o adc_data_port_10 PIXI Port 10 Analog to Digital Converter register
adc_data_port_10 = 0x4a,

/// 0x4b r/o adc_data_port_11 PIXI Port 11 Analog to Digital Converter register
adc_data_port_11 = 0x4b,

/// 0x4c r/o adc_data_port_12 PIXI Port 12 Analog to Digital Converter register
adc_data_port_12 = 0x4c,

/// 0x4d r/o adc_data_port_13 PIXI Port 13 Analog to Digital Converter register
adc_data_port_13 = 0x4d,

/// 0x4e r/o adc_data_port_14 PIXI Port 14 Analog to Digital Converter register
adc_data_port_14 = 0x4e,

/// 0x4f r/o adc_data_port_15 PIXI Port 15 Analog to Digital Converter register
adc_data_port_15 = 0x4f,

/// 0x50 r/o adc_data_port_16 PIXI Port 16 Analog to Digital Converter register
adc_data_port_16 = 0x50,

/// 0x51 r/o adc_data_port_17 PIXI Port 17 Analog to Digital Converter register
adc_data_port_17 = 0x51,

/// 0x52 r/o adc_data_port_18 PIXI Port 18 Analog to Digital Converter register
adc_data_port_18 = 0x52,
```

```
/// 0x53 r/o adc_data_port_19 PIXI Port 19 Analog to Digital Converter register
adc_data_port_19 = 0x53,

/// 0x60 r/w dac_data_port_00 PIXI Port 0 Digital to Analog Converter register
dac_data_port_00 = 0x60,

/// 0x61 r/w dac_data_port_01 PIXI Port 1 Digital to Analog Converter register
dac_data_port_01 = 0x61,

/// 0x62 r/w dac_data_port_02 PIXI Port 2 Digital to Analog Converter register
dac_data_port_02 = 0x62,

/// 0x63 r/w dac_data_port_03 PIXI Port 3 Digital to Analog Converter register
dac_data_port_03 = 0x63,

/// 0x64 r/w dac_data_port_04 PIXI Port 4 Digital to Analog Converter register
dac_data_port_04 = 0x64,

/// 0x65 r/w dac_data_port_05 PIXI Port 5 Digital to Analog Converter register
dac_data_port_05 = 0x65,

/// 0x66 r/w dac_data_port_06 PIXI Port 6 Digital to Analog Converter register
dac_data_port_06 = 0x66,

/// 0x67 r/w dac_data_port_07 PIXI Port 7 Digital to Analog Converter register
dac_data_port_07 = 0x67,

/// 0x68 r/w dac_data_port_08 PIXI Port 8 Digital to Analog Converter register
dac_data_port_08 = 0x68,

/// 0x69 r/w dac_data_port_09 PIXI Port 9 Digital to Analog Converter register
dac_data_port_09 = 0x69,

/// 0x6a r/w dac_data_port_10 PIXI Port 10 Digital to Analog Converter register
dac_data_port_10 = 0x6a,

/// 0x6b r/w dac_data_port_11 PIXI Port 11 Digital to Analog Converter register
dac_data_port_11 = 0x6b,

/// 0x6c r/w dac_data_port_12 PIXI Port 12 Digital to Analog Converter register
dac_data_port_12 = 0x6c,

/// 0x6d r/w dac_data_port_13 PIXI Port 13 Digital to Analog Converter register
dac_data_port_13 = 0x6d,

/// 0x6e r/w dac_data_port_14 PIXI Port 14 Digital to Analog Converter register
dac_data_port_14 = 0x6e,

/// 0x6f r/w dac_data_port_15 PIXI Port 15 Digital to Analog Converter register
dac_data_port_15 = 0x6f,

/// 0x70 r/w dac_data_port_16 PIXI Port 16 Digital to Analog Converter register
dac_data_port_16 = 0x70,

/// 0x71 r/w dac_data_port_17 PIXI Port 17 Digital to Analog Converter register
dac_data_port_17 = 0x71,
```

```
/// 0x72 r/w dac_data_port_18 PIXI Port 18 Digital to Analog Converter register
dac_data_port_18 = 0x72,

/// 0x73 r/w dac_data_port_19 PIXI Port 19 Digital to Analog Converter register
dac_data_port_19 = 0x73,

} MAX11300RegAddress_t;

/// 0x00 r/o dev_id Device Identification
/// <code>1111xxxxxxxxxxxx</code> PART Part field
/// <code>xxxx11xxxxxxxxxx</code> REV Revision
/// <code>xxxxxx11xxxxxxxx</code> IFMODE Inteface Mode
/// <code>xxxxxxxx11xxxxxx</code> IFSP Inteface Speed
/// <code>xxxxxxxxxx11xxxx</code> NBRPRTS Number of ports
/// <code>xxxxxxxxxxxx11xx</code> RES Resolution
/// <code>xxxxxxxxxxxxxx11</code> VRNG Voltage Range
#define dev_id_PART  0xf000
#define dev_id_REV   0x0c00
#define dev_id_IFMODE       0x0300
#define dev_id_IFSP  0x00c0
#define dev_id_NBRPRTS      0x0030
#define dev_id_RES   0x000c
#define dev_id_VRNG  0x0003

/// 0x01 r/o interrupt_flag Interrupt flags
/// <code>1xxxxxxxxxxxxxxx</code> VMON High Voltage Supply Monitor
/// <code>x1xxxxxxxxxxxxxx</code> TMPEXT2HOT External Temperature D1P/D1N Hot
/// <code>xx1xxxxxxxxxxxxx</code> TMPEXT2COLD External Temperature D1P/D1N Cold
/// <code>xxx1xxxxxxxxxxxx</code> TMPEXT2NEW External Temperature D1P/D1N New
/// <code>xxxx1xxxxxxxxxxx</code> TMPEXT1HOT External Temperature D0P/D0N Hot
/// <code>xxxxx1xxxxxxxxxx</code> TMPEXT1COLD External Temperature D0P/D0N Cold
/// <code>xxxxxx1xxxxxxxxx</code> TMPEXT1NEW External Temperature D0P/D0N New
/// <code>xxxxxxx1xxxxxxxx</code> TMPINTHOT Internal Temeprature Hot
/// <code>xxxxxxxx1xxxxxxx</code> TMPINTCOLD Internal Temeprature Cold
/// <code>xxxxxxxxx1xxxxxx</code> TMPINTNEW Internal Temeprature New
/// <code>xxxxxxxxxx1xxxxx</code> DACOI DAC over-current
/// <code>xxxxxxxxxxx1xxxx</code> GPIDM GPI event missed
/// <code>xxxxxxxxxxxx1xxx</code> GPIDR GPI event ready
/// <code>xxxxxxxxxxxxx1xx</code> ADCDM ADC data missed
/// <code>xxxxxxxxxxxxxx1x</code> ADCDR ADC data ready
/// <code>xxxxxxxxxxxxxxx1</code> ADCFLAG ADC conversion/sweep complete
#define interrupt_flag_VMON       0x8000
#define interrupt_flag_TMPEXT2HOT  0x4000
#define interrupt_flag_TMPEXT2COLD       0x2000
#define interrupt_flag_TMPEXT2NEW  0x1000
#define interrupt_flag_TMPEXT1HOT  0x0800
#define interrupt_flag_TMPEXT1COLD       0x0400
#define interrupt_flag_TMPEXT1NEW  0x0200
#define interrupt_flag_TMPINTHOT   0x0100
#define interrupt_flag_TMPINTCOLD  0x0080
#define interrupt_flag_TMPINTNEW   0x0040
#define interrupt_flag_DACOI       0x0020
#define interrupt_flag_GPIDM       0x0010
#define interrupt_flag_GPIDR       0x0008
#define interrupt_flag_ADCDM       0x0004
#define interrupt_flag_ADCDR       0x0002
#define interrupt_flag_ADCFLAG     0x0001
```

```
/// 0x02 r/o adc_status_15_to_0 new ADC data available
/// <code>1xxxxxxxxxxxxxxx</code> ADCST15 ADCST[15] new <see cref="adc_data_port_15"/>
/// <code>x1xxxxxxxxxxxxxx</code> ADCST14 ADCST[14] new <see cref="adc_data_port_14"/>
/// <code>xx1xxxxxxxxxxxxx</code> ADCST13 ADCST[13] new <see cref="adc_data_port_13"/>
/// <code>xxx1xxxxxxxxxxxx</code> ADCST12 ADCST[12] new <see cref="adc_data_port_12"/>
/// <code>xxxx1xxxxxxxxxxx</code> ADCST11 ADCST[11] new <see cref="adc_data_port_11"/>
/// <code>xxxxx1xxxxxxxxxx</code> ADCST10 ADCST[10] new <see cref="adc_data_port_10"/>
/// <code>xxxxxx1xxxxxxxxx</code> ADCST09 ADCST[9]  new <see cref="adc_data_port_09"/>
/// <code>xxxxxxx1xxxxxxxx</code> ADCST08 ADCST[8]  new <see cref="adc_data_port_08"/>
/// <code>xxxxxxxx1xxxxxxx</code> ADCST07 ADCST[7]  new <see cref="adc_data_port_07"/>
/// <code>xxxxxxxxx1xxxxxx</code> ADCST06 ADCST[6]  new <see cref="adc_data_port_06"/>
/// <code>xxxxxxxxxx1xxxxx</code> ADCST05 ADCST[5]  new <see cref="adc_data_port_05"/>
/// <code>xxxxxxxxxxx1xxxx</code> ADCST04 ADCST[4]  new <see cref="adc_data_port_04"/>
/// <code>xxxxxxxxxxxx1xxx</code> ADCST03 ADCST[3]  new <see cref="adc_data_port_03"/>
/// <code>xxxxxxxxxxxxx1xx</code> ADCST02 ADCST[2]  new <see cref="adc_data_port_02"/>
/// <code>xxxxxxxxxxxxxx1x</code> ADCST01 ADCST[1]  new <see cref="adc_data_port_01"/>
/// <code>xxxxxxxxxxxxxxx1</code> ADCST00 ADCST[0]  new <see cref="adc_data_port_00"/>
#define adc_status_15_to_0_ADCST15       0x8000
#define adc_status_15_to_0_ADCST14       0x4000
#define adc_status_15_to_0_ADCST13       0x2000
#define adc_status_15_to_0_ADCST12       0x1000
#define adc_status_15_to_0_ADCST11       0x0800
#define adc_status_15_to_0_ADCST10       0x0400
#define adc_status_15_to_0_ADCST09       0x0200
#define adc_status_15_to_0_ADCST08       0x0100
#define adc_status_15_to_0_ADCST07       0x0080
#define adc_status_15_to_0_ADCST06       0x0040
#define adc_status_15_to_0_ADCST05       0x0020
#define adc_status_15_to_0_ADCST04       0x0010
#define adc_status_15_to_0_ADCST03       0x0008
#define adc_status_15_to_0_ADCST02       0x0004
#define adc_status_15_to_0_ADCST01       0x0002
#define adc_status_15_to_0_ADCST00       0x0001

/// 0x03 r/o adc_status_19_to_16 new ADC data available
/// <code>1xxxxxxxxxxxxxxx</code> ADCST31 ADCST[31] new <see cref="adc_data_port_31"/>
/// <code>x1xxxxxxxxxxxxxx</code> ADCST30 ADCST[30] new <see cref="adc_data_port_30"/>
/// <code>xx1xxxxxxxxxxxxx</code> ADCST29 ADCST[29] new <see cref="adc_data_port_29"/>
/// <code>xxx1xxxxxxxxxxxx</code> ADCST28 ADCST[28] new <see cref="adc_data_port_28"/>
/// <code>xxxx1xxxxxxxxxxx</code> ADCST27 ADCST[27] new <see cref="adc_data_port_27"/>
/// <code>xxxxx1xxxxxxxxxx</code> ADCST26 ADCST[26] new <see cref="adc_data_port_26"/>
/// <code>xxxxxx1xxxxxxxxx</code> ADCST25 ADCST[25] new <see cref="adc_data_port_25"/>
/// <code>xxxxxxx1xxxxxxxx</code> ADCST24 ADCST[24] new <see cref="adc_data_port_24"/>
/// <code>xxxxxxxx1xxxxxxx</code> ADCST23 ADCST[23] new <see cref="adc_data_port_23"/>
/// <code>xxxxxxxxx1xxxxxx</code> ADCST22 ADCST[22] new <see cref="adc_data_port_22"/>
/// <code>xxxxxxxxxx1xxxxx</code> ADCST21 ADCST[21] new <see cref="adc_data_port_21"/>
/// <code>xxxxxxxxxxx1xxxx</code> ADCST20 ADCST[20] new <see cref="adc_data_port_20"/>
/// <code>xxxxxxxxxxxx1xxx</code> ADCST19 ADCST[19] new <see cref="adc_data_port_19"/>
/// <code>xxxxxxxxxxxxx1xx</code> ADCST18 ADCST[18] new <see cref="adc_data_port_18"/>
/// <code>xxxxxxxxxxxxxx1x</code> ADCST17 ADCST[17] new <see cref="adc_data_port_17"/>
/// <code>xxxxxxxxxxxxxxx1</code> ADCST16 ADCST[16] new <see cref="adc_data_port_16"/>
#define adc_status_19_to_16_ADCST31      0x8000
#define adc_status_19_to_16_ADCST30      0x4000
#define adc_status_19_to_16_ADCST29      0x2000
#define adc_status_19_to_16_ADCST28      0x1000
#define adc_status_19_to_16_ADCST27      0x0800
#define adc_status_19_to_16_ADCST26      0x0400
#define adc_status_19_to_16_ADCST25      0x0200
```

```
#define adc_status_19_to_16_ADCST24      0x0100
#define adc_status_19_to_16_ADCST23      0x0080
#define adc_status_19_to_16_ADCST22      0x0040
#define adc_status_19_to_16_ADCST21      0x0020
#define adc_status_19_to_16_ADCST20      0x0010
#define adc_status_19_to_16_ADCST19      0x0008
#define adc_status_19_to_16_ADCST18      0x0004
#define adc_status_19_to_16_ADCST17      0x0002
#define adc_status_19_to_16_ADCST16      0x0001

/// 0x04 r/o dac_oi_status_15_to_0 DAC Overcurrent Interrupt
/// <code>1xxxxxxxxxxxxxxx</code> DACOIST15 DACOIST[15] new <see cref="dac_data_port_15"/>
/// <code>x1xxxxxxxxxxxxxx</code> DACOIST14 DACOIST[14] new <see cref="dac_data_port_14"/>
/// <code>xx1xxxxxxxxxxxxx</code> DACOIST13 DACOIST[13] new <see cref="dac_data_port_13"/>
/// <code>xxx1xxxxxxxxxxxx</code> DACOIST12 DACOIST[12] new <see cref="dac_data_port_12"/>
/// <code>xxxx1xxxxxxxxxxx</code> DACOIST11 DACOIST[11] new <see cref="dac_data_port_11"/>
/// <code>xxxxx1xxxxxxxxxx</code> DACOIST10 DACOIST[10] new <see cref="dac_data_port_10"/>
/// <code>xxxxxx1xxxxxxxxx</code> DACOIST09 DACOIST[9]  new <see cref="dac_data_port_09"/>
/// <code>xxxxxxx1xxxxxxxx</code> DACOIST08 DACOIST[8]  new <see cref="dac_data_port_08"/>
/// <code>xxxxxxxx1xxxxxxx</code> DACOIST07 DACOIST[7]  new <see cref="dac_data_port_07"/>
/// <code>xxxxxxxxx1xxxxxx</code> DACOIST06 DACOIST[6]  new <see cref="dac_data_port_06"/>
/// <code>xxxxxxxxxx1xxxxx</code> DACOIST05 DACOIST[5]  new <see cref="dac_data_port_05"/>
/// <code>xxxxxxxxxxx1xxxx</code> DACOIST04 DACOIST[4]  new <see cref="dac_data_port_04"/>
/// <code>xxxxxxxxxxxx1xxx</code> DACOIST03 DACOIST[3]  new <see cref="dac_data_port_03"/>
/// <code>xxxxxxxxxxxxx1xx</code> DACOIST02 DACOIST[2]  new <see cref="dac_data_port_02"/>
/// <code>xxxxxxxxxxxxxx1x</code> DACOIST01 DACOIST[1]  new <see cref="dac_data_port_01"/>
/// <code>xxxxxxxxxxxxxxx1</code> DACOIST00 DACOIST[0]  new <see cref="dac_data_port_00"/>
#define dac_oi_status_15_to_0_DACOIST15   0x8000
#define dac_oi_status_15_to_0_DACOIST14   0x4000
#define dac_oi_status_15_to_0_DACOIST13   0x2000
#define dac_oi_status_15_to_0_DACOIST12   0x1000
#define dac_oi_status_15_to_0_DACOIST11   0x0800
#define dac_oi_status_15_to_0_DACOIST10   0x0400
#define dac_oi_status_15_to_0_DACOIST09   0x0200
#define dac_oi_status_15_to_0_DACOIST08   0x0100
#define dac_oi_status_15_to_0_DACOIST07   0x0080
#define dac_oi_status_15_to_0_DACOIST06   0x0040
#define dac_oi_status_15_to_0_DACOIST05   0x0020
#define dac_oi_status_15_to_0_DACOIST04   0x0010
#define dac_oi_status_15_to_0_DACOIST03   0x0008
#define dac_oi_status_15_to_0_DACOIST02   0x0004
#define dac_oi_status_15_to_0_DACOIST01   0x0002
#define dac_oi_status_15_to_0_DACOIST00   0x0001

/// 0x05 r/o dac_oi_status_19_to_16 DAC Overcurrent Interrupt
/// <code>1xxxxxxxxxxxxxxx</code> DACOIST31 DACOIST[31] new <see cref="dac_data_port_31"/>
/// <code>x1xxxxxxxxxxxxxx</code> DACOIST30 DACOIST[30] new <see cref="dac_data_port_30"/>
/// <code>xx1xxxxxxxxxxxxx</code> DACOIST29 DACOIST[29] new <see cref="dac_data_port_29"/>
/// <code>xxx1xxxxxxxxxxxx</code> DACOIST28 DACOIST[28] new <see cref="dac_data_port_28"/>
/// <code>xxxx1xxxxxxxxxxx</code> DACOIST27 DACOIST[27] new <see cref="dac_data_port_27"/>
/// <code>xxxxx1xxxxxxxxxx</code> DACOIST26 DACOIST[26] new <see cref="dac_data_port_26"/>
/// <code>xxxxxx1xxxxxxxxx</code> DACOIST25 DACOIST[25] new <see cref="dac_data_port_25"/>
/// <code>xxxxxxx1xxxxxxxx</code> DACOIST24 DACOIST[24] new <see cref="dac_data_port_24"/>
/// <code>xxxxxxxx1xxxxxxx</code> DACOIST23 DACOIST[23] new <see cref="dac_data_port_23"/>
/// <code>xxxxxxxxx1xxxxxx</code> DACOIST22 DACOIST[22] new <see cref="dac_data_port_22"/>
/// <code>xxxxxxxxxx1xxxxx</code> DACOIST21 DACOIST[21] new <see cref="dac_data_port_21"/>
/// <code>xxxxxxxxxxx1xxxx</code> DACOIST20 DACOIST[20] new <see cref="dac_data_port_20"/>
/// <code>xxxxxxxxxxxx1xxx</code> DACOIST19 DACOIST[19] new <see cref="dac_data_port_19"/>
```

```
/// <code>xxxxxxxxxxxxx1xx</code> DACOIST18 DACOIST[18] new <see cref="dac_data_port_18"/>
/// <code>xxxxxxxxxxxxxx1x</code> DACOIST17 DACOIST[17] new <see cref="dac_data_port_17"/>
/// <code>xxxxxxxxxxxxxxx1</code> DACOIST16 DACOIST[16] new <see cref="dac_data_port_16"/>
#define dac_oi_status_19_to_16_DACOIST31  0x8000
#define dac_oi_status_19_to_16_DACOIST30  0x4000
#define dac_oi_status_19_to_16_DACOIST29  0x2000
#define dac_oi_status_19_to_16_DACOIST28  0x1000
#define dac_oi_status_19_to_16_DACOIST27  0x0800
#define dac_oi_status_19_to_16_DACOIST26  0x0400
#define dac_oi_status_19_to_16_DACOIST25  0x0200
#define dac_oi_status_19_to_16_DACOIST24  0x0100
#define dac_oi_status_19_to_16_DACOIST23  0x0080
#define dac_oi_status_19_to_16_DACOIST22  0x0040
#define dac_oi_status_19_to_16_DACOIST21  0x0020
#define dac_oi_status_19_to_16_DACOIST20  0x0010
#define dac_oi_status_19_to_16_DACOIST19  0x0008
#define dac_oi_status_19_to_16_DACOIST18  0x0004
#define dac_oi_status_19_to_16_DACOIST17  0x0002
#define dac_oi_status_19_to_16_DACOIST16  0x0001

/// 0x06 r/o gpi_status_15_to_0 GPI event ready
/// <code>1xxxxxxxxxxxxxxx</code> GPIST15 GPIST[15]
/// <code>x1xxxxxxxxxxxxxx</code> GPIST14 GPIST[14]
/// <code>xx1xxxxxxxxxxxxx</code> GPIST13 GPIST[13]
/// <code>xxx1xxxxxxxxxxxx</code> GPIST12 GPIST[12]
/// <code>xxxx1xxxxxxxxxxx</code> GPIST11 GPIST[11]
/// <code>xxxxx1xxxxxxxxxx</code> GPIST10 GPIST[10]
/// <code>xxxxxx1xxxxxxxxx</code> GPIST09 GPIST[9]
/// <code>xxxxxxx1xxxxxxxx</code> GPIST08 GPIST[8]
/// <code>xxxxxxxx1xxxxxxx</code> GPIST07 GPIST[7]
/// <code>xxxxxxxxx1xxxxxx</code> GPIST06 GPIST[6]
/// <code>xxxxxxxxxx1xxxxx</code> GPIST05 GPIST[5]
/// <code>xxxxxxxxxxx1xxxx</code> GPIST04 GPIST[4]
/// <code>xxxxxxxxxxxx1xxx</code> GPIST03 GPIST[3]
/// <code>xxxxxxxxxxxxx1xx</code> GPIST02 GPIST[2]
/// <code>xxxxxxxxxxxxxx1x</code> GPIST01 GPIST[1]
/// <code>xxxxxxxxxxxxxxx1</code> GPIST00 GPIST[0]
#define gpi_status_15_to_0_GPIST15        0x8000
#define gpi_status_15_to_0_GPIST14        0x4000
#define gpi_status_15_to_0_GPIST13        0x2000
#define gpi_status_15_to_0_GPIST12        0x1000
#define gpi_status_15_to_0_GPIST11        0x0800
#define gpi_status_15_to_0_GPIST10        0x0400
#define gpi_status_15_to_0_GPIST09        0x0200
#define gpi_status_15_to_0_GPIST08        0x0100
#define gpi_status_15_to_0_GPIST07        0x0080
#define gpi_status_15_to_0_GPIST06        0x0040
#define gpi_status_15_to_0_GPIST05        0x0020
#define gpi_status_15_to_0_GPIST04        0x0010
#define gpi_status_15_to_0_GPIST03        0x0008
#define gpi_status_15_to_0_GPIST02        0x0004
#define gpi_status_15_to_0_GPIST01        0x0002
#define gpi_status_15_to_0_GPIST00        0x0001

/// 0x07 r/o gpi_status_19_to_16 GPI event ready
/// <code>1xxxxxxxxxxxxxxx</code> GPIST31 GPIST[31]
/// <code>x1xxxxxxxxxxxxxx</code> GPIST30 GPIST[30]
/// <code>xx1xxxxxxxxxxxxx</code> GPIST29 GPIST[29]
```

```
/// <code>xxx1xxxxxxxxxxxx</code> GPIST28 GPIST[28]
/// <code>xxxx1xxxxxxxxxxx</code> GPIST27 GPIST[27]
/// <code>xxxxx1xxxxxxxxxx</code> GPIST26 GPIST[26]
/// <code>xxxxxx1xxxxxxxxx</code> GPIST25 GPIST[25]
/// <code>xxxxxxx1xxxxxxxx</code> GPIST24 GPIST[24]
/// <code>xxxxxxxx1xxxxxxx</code> GPIST23 GPIST[23]
/// <code>xxxxxxxxx1xxxxxx</code> GPIST22 GPIST[22]
/// <code>xxxxxxxxxx1xxxxx</code> GPIST21 GPIST[21]
/// <code>xxxxxxxxxxx1xxxx</code> GPIST20 GPIST[20]
/// <code>xxxxxxxxxxxx1xxx</code> GPIST19 GPIST[19]
/// <code>xxxxxxxxxxxxx1xx</code> GPIST18 GPIST[18]
/// <code>xxxxxxxxxxxxxx1x</code> GPIST17 GPIST[17]
/// <code>xxxxxxxxxxxxxxx1</code> GPIST16 GPIST[16]
#define gpi_status_19_to_16_GPIST31      0x8000
#define gpi_status_19_to_16_GPIST30      0x4000
#define gpi_status_19_to_16_GPIST29      0x2000
#define gpi_status_19_to_16_GPIST28      0x1000
#define gpi_status_19_to_16_GPIST27      0x0800
#define gpi_status_19_to_16_GPIST26      0x0400
#define gpi_status_19_to_16_GPIST25      0x0200
#define gpi_status_19_to_16_GPIST24      0x0100
#define gpi_status_19_to_16_GPIST23      0x0080
#define gpi_status_19_to_16_GPIST22      0x0040
#define gpi_status_19_to_16_GPIST21      0x0020
#define gpi_status_19_to_16_GPIST20      0x0010
#define gpi_status_19_to_16_GPIST19      0x0008
#define gpi_status_19_to_16_GPIST18      0x0004
#define gpi_status_19_to_16_GPIST17      0x0002
#define gpi_status_19_to_16_GPIST16      0x0001

/// 0x08 r/o tmp_int_data Internal Temeprature
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_int_data_tempcode     0x0fff

/// 0x09 r/o tmp_ext1_data External Temperature D0P/D0N
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_ext1_data_tempcode    0x0fff

/// 0x0a r/o tmp_ext2_data External Temperature D1P/D1N
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_ext2_data_tempcode    0x0fff

/// 0x0b r/o gpi_data_15_to_0 GPI input ports data
/// <code>1xxxxxxxxxxxxxxx</code> GPIDAT15 GPIDAT[15]
/// <code>x1xxxxxxxxxxxxxx</code> GPIDAT14 GPIDAT[14]
/// <code>xx1xxxxxxxxxxxxx</code> GPIDAT13 GPIDAT[13]
/// <code>xxx1xxxxxxxxxxxx</code> GPIDAT12 GPIDAT[12]
/// <code>xxxx1xxxxxxxxxxx</code> GPIDAT11 GPIDAT[11]
/// <code>xxxxx1xxxxxxxxxx</code> GPIDAT10 GPIDAT[10]
/// <code>xxxxxx1xxxxxxxxx</code> GPIDAT09 GPIDAT[9]
/// <code>xxxxxxx1xxxxxxxx</code> GPIDAT08 GPIDAT[8]
/// <code>xxxxxxxx1xxxxxxx</code> GPIDAT07 GPIDAT[7]
/// <code>xxxxxxxxx1xxxxxx</code> GPIDAT06 GPIDAT[6]
/// <code>xxxxxxxxxx1xxxxx</code> GPIDAT05 GPIDAT[5]
/// <code>xxxxxxxxxxx1xxxx</code> GPIDAT04 GPIDAT[4]
```

```
/// <code>xxxxxxxxxxxx1xxx</code> GPIDAT03 GPIDAT[3]
/// <code>xxxxxxxxxxxxx1xx</code> GPIDAT02 GPIDAT[2]
/// <code>xxxxxxxxxxxxxx1x</code> GPIDAT01 GPIDAT[1]
/// <code>xxxxxxxxxxxxxxx1</code> GPIDAT00 GPIDAT[0]
#define gpi_data_15_to_0_GPIDAT15  0x8000
#define gpi_data_15_to_0_GPIDAT14  0x4000
#define gpi_data_15_to_0_GPIDAT13  0x2000
#define gpi_data_15_to_0_GPIDAT12  0x1000
#define gpi_data_15_to_0_GPIDAT11  0x0800
#define gpi_data_15_to_0_GPIDAT10  0x0400
#define gpi_data_15_to_0_GPIDAT09  0x0200
#define gpi_data_15_to_0_GPIDAT08  0x0100
#define gpi_data_15_to_0_GPIDAT07  0x0080
#define gpi_data_15_to_0_GPIDAT06  0x0040
#define gpi_data_15_to_0_GPIDAT05  0x0020
#define gpi_data_15_to_0_GPIDAT04  0x0010
#define gpi_data_15_to_0_GPIDAT03  0x0008
#define gpi_data_15_to_0_GPIDAT02  0x0004
#define gpi_data_15_to_0_GPIDAT01  0x0002
#define gpi_data_15_to_0_GPIDAT00  0x0001

/// 0x0c r/o gpi_data_19_to_16 GPI input ports data
/// <code>1xxxxxxxxxxxxxxx</code> GPIDAT31 GPIDAT[31]
/// <code>x1xxxxxxxxxxxxxx</code> GPIDAT30 GPIDAT[30]
/// <code>xx1xxxxxxxxxxxxx</code> GPIDAT29 GPIDAT[29]
/// <code>xxx1xxxxxxxxxxxx</code> GPIDAT28 GPIDAT[28]
/// <code>xxxx1xxxxxxxxxxx</code> GPIDAT27 GPIDAT[27]
/// <code>xxxxx1xxxxxxxxxx</code> GPIDAT26 GPIDAT[26]
/// <code>xxxxxx1xxxxxxxxx</code> GPIDAT25 GPIDAT[25]
/// <code>xxxxxxx1xxxxxxxx</code> GPIDAT24 GPIDAT[24]
/// <code>xxxxxxxx1xxxxxxx</code> GPIDAT23 GPIDAT[23]
/// <code>xxxxxxxxx1xxxxxx</code> GPIDAT22 GPIDAT[22]
/// <code>xxxxxxxxxx1xxxxx</code> GPIDAT21 GPIDAT[21]
/// <code>xxxxxxxxxxx1xxxx</code> GPIDAT20 GPIDAT[20]
/// <code>xxxxxxxxxxxx1xxx</code> GPIDAT19 GPIDAT[19]
/// <code>xxxxxxxxxxxxx1xx</code> GPIDAT18 GPIDAT[18]
/// <code>xxxxxxxxxxxxxx1x</code> GPIDAT17 GPIDAT[17]
/// <code>xxxxxxxxxxxxxxx1</code> GPIDAT16 GPIDAT[16]
#define gpi_data_19_to_16_GPIDAT31       0x8000
#define gpi_data_19_to_16_GPIDAT30       0x4000
#define gpi_data_19_to_16_GPIDAT29       0x2000
#define gpi_data_19_to_16_GPIDAT28       0x1000
#define gpi_data_19_to_16_GPIDAT27       0x0800
#define gpi_data_19_to_16_GPIDAT26       0x0400
#define gpi_data_19_to_16_GPIDAT25       0x0200
#define gpi_data_19_to_16_GPIDAT24       0x0100
#define gpi_data_19_to_16_GPIDAT23       0x0080
#define gpi_data_19_to_16_GPIDAT22       0x0040
#define gpi_data_19_to_16_GPIDAT21       0x0020
#define gpi_data_19_to_16_GPIDAT20       0x0010
#define gpi_data_19_to_16_GPIDAT19       0x0008
#define gpi_data_19_to_16_GPIDAT18       0x0004
#define gpi_data_19_to_16_GPIDAT17       0x0002
#define gpi_data_19_to_16_GPIDAT16       0x0001

/// 0x0d r/w gpo_data_15_to_0 GPO output ports data
/// <code>1xxxxxxxxxxxxxxx</code> GPODAT15 GPODAT[15]
/// <code>x1xxxxxxxxxxxxxx</code> GPODAT14 GPODAT[14]
```

```
/// <code>xx1xxxxxxxxxxxxx</code> GPODAT13 GPODAT[13]
/// <code>xxx1xxxxxxxxxxxx</code> GPODAT12 GPODAT[12]
/// <code>xxxx1xxxxxxxxxxx</code> GPODAT11 GPODAT[11]
/// <code>xxxxx1xxxxxxxxxx</code> GPODAT10 GPODAT[10]
/// <code>xxxxxx1xxxxxxxxx</code> GPODAT09 GPODAT[9]
/// <code>xxxxxxx1xxxxxxxx</code> GPODAT08 GPODAT[8]
/// <code>xxxxxxxx1xxxxxxx</code> GPODAT07 GPODAT[7]
/// <code>xxxxxxxxx1xxxxxx</code> GPODAT06 GPODAT[6]
/// <code>xxxxxxxxxx1xxxxx</code> GPODAT05 GPODAT[5]
/// <code>xxxxxxxxxxx1xxxx</code> GPODAT04 GPODAT[4]
/// <code>xxxxxxxxxxxx1xxx</code> GPODAT03 GPODAT[3]
/// <code>xxxxxxxxxxxxx1xx</code> GPODAT02 GPODAT[2]
/// <code>xxxxxxxxxxxxxx1x</code> GPODAT01 GPODAT[1]
/// <code>xxxxxxxxxxxxxxx1</code> GPODAT00 GPODAT[0]
#define gpo_data_15_to_0_GPODAT15  0x8000
#define gpo_data_15_to_0_GPODAT14  0x4000
#define gpo_data_15_to_0_GPODAT13  0x2000
#define gpo_data_15_to_0_GPODAT12  0x1000
#define gpo_data_15_to_0_GPODAT11  0x0800
#define gpo_data_15_to_0_GPODAT10  0x0400
#define gpo_data_15_to_0_GPODAT09  0x0200
#define gpo_data_15_to_0_GPODAT08  0x0100
#define gpo_data_15_to_0_GPODAT07  0x0080
#define gpo_data_15_to_0_GPODAT06  0x0040
#define gpo_data_15_to_0_GPODAT05  0x0020
#define gpo_data_15_to_0_GPODAT04  0x0010
#define gpo_data_15_to_0_GPODAT03  0x0008
#define gpo_data_15_to_0_GPODAT02  0x0004
#define gpo_data_15_to_0_GPODAT01  0x0002
#define gpo_data_15_to_0_GPODAT00  0x0001
#define gpo_data_15_to_0_DESIGNVALUE     0x0000

/// 0x0e r/w gpo_data_19_to_16 GPO output ports data
/// <code>1xxxxxxxxxxxxxxx</code> GPODAT31 GPODAT[31]
/// <code>x1xxxxxxxxxxxxxx</code> GPODAT30 GPODAT[30]
/// <code>xx1xxxxxxxxxxxxx</code> GPODAT29 GPODAT[29]
/// <code>xxx1xxxxxxxxxxxx</code> GPODAT28 GPODAT[28]
/// <code>xxxx1xxxxxxxxxxx</code> GPODAT27 GPODAT[27]
/// <code>xxxxx1xxxxxxxxxx</code> GPODAT26 GPODAT[26]
/// <code>xxxxxx1xxxxxxxxx</code> GPODAT25 GPODAT[25]
/// <code>xxxxxxx1xxxxxxxx</code> GPODAT24 GPODAT[24]
/// <code>xxxxxxxx1xxxxxxx</code> GPODAT23 GPODAT[23]
/// <code>xxxxxxxxx1xxxxxx</code> GPODAT22 GPODAT[22]
/// <code>xxxxxxxxxx1xxxxx</code> GPODAT21 GPODAT[21]
/// <code>xxxxxxxxxxx1xxxx</code> GPODAT20 GPODAT[20]
/// <code>xxxxxxxxxxxx1xxx</code> GPODAT19 GPODAT[19]
/// <code>xxxxxxxxxxxxx1xx</code> GPODAT18 GPODAT[18]
/// <code>xxxxxxxxxxxxxx1x</code> GPODAT17 GPODAT[17]
/// <code>xxxxxxxxxxxxxxx1</code> GPODAT16 GPODAT[16]
#define gpo_data_19_to_16_GPODAT31       0x8000
#define gpo_data_19_to_16_GPODAT30       0x4000
#define gpo_data_19_to_16_GPODAT29       0x2000
#define gpo_data_19_to_16_GPODAT28       0x1000
#define gpo_data_19_to_16_GPODAT27       0x0800
#define gpo_data_19_to_16_GPODAT26       0x0400
#define gpo_data_19_to_16_GPODAT25       0x0200
#define gpo_data_19_to_16_GPODAT24       0x0100
#define gpo_data_19_to_16_GPODAT23       0x0080
```

```
#define gpo_data_19_to_16_GPODAT22       0x0040
#define gpo_data_19_to_16_GPODAT21       0x0020
#define gpo_data_19_to_16_GPODAT20       0x0010
#define gpo_data_19_to_16_GPODAT19       0x0008
#define gpo_data_19_to_16_GPODAT18       0x0004
#define gpo_data_19_to_16_GPODAT17       0x0002
#define gpo_data_19_to_16_GPODAT16       0x0001
#define gpo_data_19_to_16_DESIGNVALUE    0x0000

/// 0x0f r/o reserved_0F reserved


/// 0x10 r/w device_control Global device control register
/// <code>1xxxxxxxxxxxxxxx</code> RESET Soft reset command
/// <code>x1xxxxxxxxxxxxxx</code> BRST Burst Mode
/// <code>xx1xxxxxxxxxxxxx</code> LPEN Low Power Enable
/// <code>xxx1xxxxxxxxxxxx</code> RS_CANCEL series resistance cancelation on external
temperature monitors D0P/D0N and D1P/D1N
/// <code>xxxx1xxxxxxxxxxx</code> TMPPER temperature monitor period
/// <code>xxxxx1xxxxxxxxxx</code> TMPCTLEXT1 monitor external temperature D1P/D1N
/// <code>xxxxxx1xxxxxxxxx</code> TMPCTLEXT0 monitor external temperature D0P/D0N
/// <code>xxxxxxx1xxxxxxxx</code> TMPCTLINT monitor internal temperature
/// <code>xxxxxxxx1xxxxxxx</code> THSHDN Thermal Shutdown
/// <code>xxxxxxxxx1xxxxxx</code> DACREF DAC voltage reference
/// <code>xxxxxxxxxx11xxxx</code> ADCCONV ADC conversion rate
/// <code>xxxxxxxxxxxx11xx</code> DACCTL DAC update mode
/// <code>xxxxxxxxxxxxxx11</code> ADCCTL ADC conversion mode
#define device_control_RESET       0x8000
#define device_control_BRST        0x4000
#define device_control_LPEN        0x2000
#define device_control_RS_CANCEL   0x1000
#define device_control_TMPPER      0x0800
#define device_control_TMPCTLEXT1  0x0400
#define device_control_TMPCTLEXT0  0x0200
#define device_control_TMPCTLINT   0x0100
#define device_control_THSHDN      0x0080
#define device_control_DACREF      0x0040
#define device_control_ADCCONV     0x0030
#define device_control_DACCTL      0x000c
#define device_control_ADCCTL      0x0003
#define device_control_DESIGNVALUE       0x0703

/// 0x11 r/w interrupt_mask interrupt mask (1 = disable interrupt source)
/// <code>1xxxxxxxxxxxxxxx</code> VMON High Voltage Supply Monitor
/// <code>x1xxxxxxxxxxxxxx</code> TMPEXT2HOT External Temperature D1P/D1N Hot
/// <code>xx1xxxxxxxxxxxxx</code> TMPEXT2COLD External Temperature D1P/D1N Cold
/// <code>xxx1xxxxxxxxxxxx</code> TMPEXT2NEW External Temperature D1P/D1N New
/// <code>xxxx1xxxxxxxxxxx</code> TMPEXT1HOT External Temperature D0P/D0N Hot
/// <code>xxxxx1xxxxxxxxxx</code> TMPEXT1COLD External Temperature D0P/D0N Cold
/// <code>xxxxxx1xxxxxxxxx</code> TMPEXT1NEW External Temperature D0P/D0N New
/// <code>xxxxxxx1xxxxxxxx</code> TMPINTHOT Internal Temeprature Hot
/// <code>xxxxxxxx1xxxxxxx</code> TMPINTCOLD Internal Temeprature Cold
/// <code>xxxxxxxxx1xxxxxx</code> TMPINTNEW Internal Temeprature New
/// <code>xxxxxxxxxx1xxxxx</code> DACOI DAC over-current
/// <code>xxxxxxxxxxx1xxxx</code> GPIDM GPI event missed
/// <code>xxxxxxxxxxxx1xxx</code> GPIDR GPI event ready
/// <code>xxxxxxxxxxxxx1xx</code> ADCDM ADC data missed
/// <code>xxxxxxxxxxxxxx1x</code> ADCDR ADC data ready
```

```
/// <code>xxxxxxxxxxxxxxx1</code> ADCFLAG ADC conversion/sweep complete
#define interrupt_mask_VMON        0x8000
#define interrupt_mask_TMPEXT2HOT  0x4000
#define interrupt_mask_TMPEXT2COLD        0x2000
#define interrupt_mask_TMPEXT2NEW  0x1000
#define interrupt_mask_TMPEXT1HOT  0x0800
#define interrupt_mask_TMPEXT1COLD        0x0400
#define interrupt_mask_TMPEXT1NEW  0x0200
#define interrupt_mask_TMPINTHOT   0x0100
#define interrupt_mask_TMPINTCOLD  0x0080
#define interrupt_mask_TMPINTNEW   0x0040
#define interrupt_mask_DACOI       0x0020
#define interrupt_mask_GPIDM       0x0010
#define interrupt_mask_GPIDR       0x0008
#define interrupt_mask_ADCDM       0x0004
#define interrupt_mask_ADCDR       0x0002
#define interrupt_mask_ADCFLAG     0x0001
#define interrupt_mask_DESIGNVALUE        0x6d81

/// 0x12 r/w gpi_irqmode_7_to_0 xxxxxx
/// <code>11xxxxxxxxxxxxxx</code> GPIMD07 GPIMD[7]
/// <code>xx11xxxxxxxxxxxx</code> GPIMD06 GPIMD[6]
/// <code>xxxx11xxxxxxxxxx</code> GPIMD05 GPIMD[5]
/// <code>xxxxxx11xxxxxxxx</code> GPIMD04 GPIMD[4]
/// <code>xxxxxxxx11xxxxxx</code> GPIMD03 GPIMD[3]
/// <code>xxxxxxxxxx11xxxx</code> GPIMD02 GPIMD[2]
/// <code>xxxxxxxxxxxx11xx</code> GPIMD01 GPIMD[1]
/// <code>xxxxxxxxxxxxxx11</code> GPIMD00 GPIMD[0]
#define gpi_irqmode_7_to_0_GPIMD07        0xc000
#define gpi_irqmode_7_to_0_GPIMD06        0x3000
#define gpi_irqmode_7_to_0_GPIMD05        0x0c00
#define gpi_irqmode_7_to_0_GPIMD04        0x0300
#define gpi_irqmode_7_to_0_GPIMD03        0x00c0
#define gpi_irqmode_7_to_0_GPIMD02        0x0030
#define gpi_irqmode_7_to_0_GPIMD01        0x000c
#define gpi_irqmode_7_to_0_GPIMD00        0x0003
#define gpi_irqmode_7_to_0_DESIGNVALUE    0x0000

/// 0x13 r/w gpi_irqmode_15_to_8 xxxxxx
/// <code>11xxxxxxxxxxxxxx</code> GPIMD15 GPIMD[15]
/// <code>xx11xxxxxxxxxxxx</code> GPIMD14 GPIMD[14]
/// <code>xxxx11xxxxxxxxxx</code> GPIMD13 GPIMD[13]
/// <code>xxxxxx11xxxxxxxx</code> GPIMD12 GPIMD[12]
/// <code>xxxxxxxx11xxxxxx</code> GPIMD11 GPIMD[11]
/// <code>xxxxxxxxxx11xxxx</code> GPIMD10 GPIMD[10]
/// <code>xxxxxxxxxxxx11xx</code> GPIMD09 GPIMD[9]
/// <code>xxxxxxxxxxxxxx11</code> GPIMD08 GPIMD[8]
#define gpi_irqmode_15_to_8_GPIMD15       0xc000
#define gpi_irqmode_15_to_8_GPIMD14       0x3000
#define gpi_irqmode_15_to_8_GPIMD13       0x0c00
#define gpi_irqmode_15_to_8_GPIMD12       0x0300
#define gpi_irqmode_15_to_8_GPIMD11       0x00c0
#define gpi_irqmode_15_to_8_GPIMD10       0x0030
#define gpi_irqmode_15_to_8_GPIMD09       0x000c
#define gpi_irqmode_15_to_8_GPIMD08       0x0003
#define gpi_irqmode_15_to_8_DESIGNVALUE   0x0000

/// 0x14 r/w gpi_irqmode_19_to_16 xxxxxx
```

```
/// <code>11xxxxxxxxxxxxxx</code> GPIMD23 GPIMD[23]
/// <code>xx11xxxxxxxxxxxx</code> GPIMD22 GPIMD[22]
/// <code>xxxx11xxxxxxxxxx</code> GPIMD21 GPIMD[21]
/// <code>xxxxxx11xxxxxxxx</code> GPIMD20 GPIMD[20]
/// <code>xxxxxxxx11xxxxxx</code> GPIMD19 GPIMD[19]
/// <code>xxxxxxxxxx11xxxx</code> GPIMD18 GPIMD[18]
/// <code>xxxxxxxxxxxx11xx</code> GPIMD17 GPIMD[17]
/// <code>xxxxxxxxxxxxxx11</code> GPIMD16 GPIMD[16]
#define gpi_irqmode_19_to_16_GPIMD23      0xc000
#define gpi_irqmode_19_to_16_GPIMD22      0x3000
#define gpi_irqmode_19_to_16_GPIMD21      0x0c00
#define gpi_irqmode_19_to_16_GPIMD20      0x0300
#define gpi_irqmode_19_to_16_GPIMD19      0x00c0
#define gpi_irqmode_19_to_16_GPIMD18      0x0030
#define gpi_irqmode_19_to_16_GPIMD17      0x000c
#define gpi_irqmode_19_to_16_GPIMD16      0x0003
#define gpi_irqmode_19_to_16_DESIGNVALUE  0x0039

/// 0x15 r/w gpi_irqmode_31_to_24 xxxxxx
/// <code>11xxxxxxxxxxxxxx</code> GPIMD31 GPIMD[31]
/// <code>xx11xxxxxxxxxxxx</code> GPIMD30 GPIMD[30]
/// <code>xxxx11xxxxxxxxxx</code> GPIMD29 GPIMD[29]
/// <code>xxxxxx11xxxxxxxx</code> GPIMD28 GPIMD[28]
/// <code>xxxxxxxx11xxxxxx</code> GPIMD27 GPIMD[27]
/// <code>xxxxxxxxxx11xxxx</code> GPIMD26 GPIMD[26]
/// <code>xxxxxxxxxxxx11xx</code> GPIMD25 GPIMD[25]
/// <code>xxxxxxxxxxxxxx11</code> GPIMD24 GPIMD[24]
#define gpi_irqmode_31_to_24_GPIMD31      0xc000
#define gpi_irqmode_31_to_24_GPIMD30      0x3000
#define gpi_irqmode_31_to_24_GPIMD29      0x0c00
#define gpi_irqmode_31_to_24_GPIMD28      0x0300
#define gpi_irqmode_31_to_24_GPIMD27      0x00c0
#define gpi_irqmode_31_to_24_GPIMD26      0x0030
#define gpi_irqmode_31_to_24_GPIMD25      0x000c
#define gpi_irqmode_31_to_24_GPIMD24      0x0003

/// 0x16 r/w dac_preset_data_1 DAC preset activated by <see cref="device_control"/>
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_preset_data_1_daccode  0x0fff
#define dac_preset_data_1_DESIGNVALUE     0x0155

/// 0x17 r/w dac_preset_data_2 DAC preset activated by <see cref="device_control"/>
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_preset_data_2_daccode  0x0fff
#define dac_preset_data_2_DESIGNVALUE     0x0066

/// 0x18 r/w tmp_mon_cfg Temperautre Monitor Configuration
/// <code>xxxxxxxxxx11xxxx</code> TMPEXT2MONCFG average 4, 8, 16, or 32 measurements
/// <code>xxxxxxxxxxxx11xx</code> TMPEXT1MONCFG average 4, 8, 16, or 32 measurements
/// <code>xxxxxxxxxxxxxx11</code> TMPINTMONCFG average 4, 8, 16, or 32 measurements
#define tmp_mon_cfg_TMPEXT2MONCFG  0x0030
#define tmp_mon_cfg_TMPEXT1MONCFG  0x000c
#define tmp_mon_cfg_TMPINTMONCFG   0x0003

/// 0x19 r/w tmp_mon_int_hi_thresh Internal Temeprature Hot Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_int_hi_thresh_tempcode    0x0fff
```

```
#define tmp_mon_int_hi_thresh_DESIGNVALUE       0x02a8

/// 0x1a r/w tmp_mon_int_lo_thresh Internal Temeprature Cold Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_int_lo_thresh_tempcode    0x0fff
#define tmp_mon_int_lo_thresh_DESIGNVALUE       0x0ec0

/// 0x1b r/w tmp_mon_ext1_hi_thresh External Temperature D0P/D0N Hot Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_ext1_hi_thresh_tempcode   0x0fff
#define tmp_mon_ext1_hi_thresh_DESIGNVALUE      0x039a

/// 0x1c r/w tmp_mon_ext1_lo_thresh External Temperature D0P/D0N Cold Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_ext1_lo_thresh_tempcode   0x0fff
#define tmp_mon_ext1_lo_thresh_DESIGNVALUE      0x0090

/// 0x1d r/w tmp_mon_ext2_hi_thresh External Temperature D1P/D1N Hot Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_ext2_hi_thresh_tempcode   0x0fff
#define tmp_mon_ext2_hi_thresh_DESIGNVALUE      0x07ff

/// 0x1e r/w tmp_mon_ext2_lo_thresh External Temperature D1P/D1N Cold Threshold
/// <code>xxxx111111111111</code> tempcode Temperature code, LSB=0.125 degrees C, 12-bit 2's
complement
#define tmp_mon_ext2_lo_thresh_tempcode   0x0fff
#define tmp_mon_ext2_lo_thresh_DESIGNVALUE      0x0800

/// 0x1f r/w reserved_1F reserved


/// 0x20 r/w port_cfg_00 PIXI Port 0 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_00_PortCfgFuncID  0xf000
#define port_cfg_00_funcprm_avrInv       0x0800
#define port_cfg_00_funcprm_range  0x0700
#define port_cfg_00_funcprm_nsamples     0x00e0
#define port_cfg_00_funcprm_port   0x001f
#define port_cfg_00_DESIGNVALUE    0x6100

/// 0x21 r/w port_cfg_01 PIXI Port 1 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_01_PortCfgFuncID  0xf000
#define port_cfg_01_funcprm_avrInv       0x0800
#define port_cfg_01_funcprm_range  0x0700
#define port_cfg_01_funcprm_nsamples     0x00e0
```

```
#define port_cfg_01_funcprm_port   0x001f
#define port_cfg_01_DESIGNVALUE    0x1000

/// 0x22 r/w port_cfg_02 PIXI Port 2 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_02_PortCfgFuncID  0xf000
#define port_cfg_02_funcprm_avrInv        0x0800
#define port_cfg_02_funcprm_range  0x0700
#define port_cfg_02_funcprm_nsamples      0x00e0
#define port_cfg_02_funcprm_port   0x001f
#define port_cfg_02_DESIGNVALUE    0x2000

/// 0x23 r/w port_cfg_03 PIXI Port 3 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_03_PortCfgFuncID  0xf000
#define port_cfg_03_funcprm_avrInv        0x0800
#define port_cfg_03_funcprm_range  0x0700
#define port_cfg_03_funcprm_nsamples      0x00e0
#define port_cfg_03_funcprm_port   0x001f
#define port_cfg_03_DESIGNVALUE    0x0000

/// 0x24 r/w port_cfg_04 PIXI Port 4 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_04_PortCfgFuncID  0xf000
#define port_cfg_04_funcprm_avrInv        0x0800
#define port_cfg_04_funcprm_range  0x0700
#define port_cfg_04_funcprm_nsamples      0x00e0
#define port_cfg_04_funcprm_port   0x001f
#define port_cfg_04_DESIGNVALUE    0x3000

/// 0x25 r/w port_cfg_05 PIXI Port 5 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_05_PortCfgFuncID  0xf000
#define port_cfg_05_funcprm_avrInv        0x0800
#define port_cfg_05_funcprm_range  0x0700
#define port_cfg_05_funcprm_nsamples      0x00e0
#define port_cfg_05_funcprm_port   0x001f
#define port_cfg_05_DESIGNVALUE    0x5100

/// 0x26 r/w port_cfg_06 PIXI Port 6 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
```

```
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_06_PortCfgFuncID  0xf000
#define port_cfg_06_funcprm_avrInv        0x0800
#define port_cfg_06_funcprm_range  0x0700
#define port_cfg_06_funcprm_nsamples      0x00e0
#define port_cfg_06_funcprm_port   0x001f
#define port_cfg_06_DESIGNVALUE    0x6100

/// 0x27 r/w port_cfg_07 PIXI Port 7 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_07_PortCfgFuncID  0xf000
#define port_cfg_07_funcprm_avrInv        0x0800
#define port_cfg_07_funcprm_range  0x0700
#define port_cfg_07_funcprm_nsamples      0x00e0
#define port_cfg_07_funcprm_port   0x001f
#define port_cfg_07_DESIGNVALUE    0x7400

/// 0x28 r/w port_cfg_08 PIXI Port 8 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_08_PortCfgFuncID  0xf000
#define port_cfg_08_funcprm_avrInv        0x0800
#define port_cfg_08_funcprm_range  0x0700
#define port_cfg_08_funcprm_nsamples      0x00e0
#define port_cfg_08_funcprm_port   0x001f
#define port_cfg_08_DESIGNVALUE    0x8409

/// 0x29 r/w port_cfg_09 PIXI Port 9 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_09_PortCfgFuncID  0xf000
#define port_cfg_09_funcprm_avrInv        0x0800
#define port_cfg_09_funcprm_range  0x0700
#define port_cfg_09_funcprm_nsamples      0x00e0
#define port_cfg_09_funcprm_port   0x001f
#define port_cfg_09_DESIGNVALUE    0x9400

/// 0x2a r/w port_cfg_10 PIXI Port 10 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_10_PortCfgFuncID  0xf000
#define port_cfg_10_funcprm_avrInv        0x0800
#define port_cfg_10_funcprm_range  0x0700
```

```
#define port_cfg_10_funcprm_nsamples      0x00e0
#define port_cfg_10_funcprm_port   0x001f
#define port_cfg_10_DESIGNVALUE    0x810b

/// 0x2b r/w port_cfg_11 PIXI Port 11 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_11_PortCfgFuncID  0xf000
#define port_cfg_11_funcprm_avrInv        0x0800
#define port_cfg_11_funcprm_range  0x0700
#define port_cfg_11_funcprm_nsamples      0x00e0
#define port_cfg_11_funcprm_port   0x001f
#define port_cfg_11_DESIGNVALUE    0xa100

/// 0x2c r/w port_cfg_12 PIXI Port 12 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_12_PortCfgFuncID  0xf000
#define port_cfg_12_funcprm_avrInv        0x0800
#define port_cfg_12_funcprm_range  0x0700
#define port_cfg_12_funcprm_nsamples      0x00e0
#define port_cfg_12_funcprm_port   0x001f
#define port_cfg_12_DESIGNVALUE    0xb001

/// 0x2d r/w port_cfg_13 PIXI Port 13 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_13_PortCfgFuncID  0xf000
#define port_cfg_13_funcprm_avrInv        0x0800
#define port_cfg_13_funcprm_range  0x0700
#define port_cfg_13_funcprm_nsamples      0x00e0
#define port_cfg_13_funcprm_port   0x001f
#define port_cfg_13_DESIGNVALUE    0x0000

/// 0x2e r/w port_cfg_14 PIXI Port 14 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_14_PortCfgFuncID  0xf000
#define port_cfg_14_funcprm_avrInv        0x0800
#define port_cfg_14_funcprm_range  0x0700
#define port_cfg_14_funcprm_nsamples      0x00e0
#define port_cfg_14_funcprm_port   0x001f
#define port_cfg_14_DESIGNVALUE    0x0000

/// 0x2f r/w port_cfg_15 PIXI Port 15 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
```

```
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_15_PortCfgFuncID  0xf000
#define port_cfg_15_funcprm_avrInv       0x0800
#define port_cfg_15_funcprm_range  0x0700
#define port_cfg_15_funcprm_nsamples     0x00e0
#define port_cfg_15_funcprm_port   0x001f
#define port_cfg_15_DESIGNVALUE    0x1000

/// 0x30 r/w port_cfg_16 PIXI Port 16 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_16_PortCfgFuncID  0xf000
#define port_cfg_16_funcprm_avrInv       0x0800
#define port_cfg_16_funcprm_range  0x0700
#define port_cfg_16_funcprm_nsamples     0x00e0
#define port_cfg_16_funcprm_port   0x001f
#define port_cfg_16_DESIGNVALUE    0x1000

/// 0x31 r/w port_cfg_17 PIXI Port 17 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_17_PortCfgFuncID  0xf000
#define port_cfg_17_funcprm_avrInv       0x0800
#define port_cfg_17_funcprm_range  0x0700
#define port_cfg_17_funcprm_nsamples     0x00e0
#define port_cfg_17_funcprm_port   0x001f
#define port_cfg_17_DESIGNVALUE    0x1000

/// 0x32 r/w port_cfg_18 PIXI Port 18 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_18_PortCfgFuncID  0xf000
#define port_cfg_18_funcprm_avrInv       0x0800
#define port_cfg_18_funcprm_range  0x0700
#define port_cfg_18_funcprm_nsamples     0x00e0
#define port_cfg_18_funcprm_port   0x001f
#define port_cfg_18_DESIGNVALUE    0x1000

/// 0x33 r/w port_cfg_19 PIXI Port 19 configuration register
/// <code>1111xxxxxxxxxxxx</code> PortCfgFuncID Port function / mode
/// <code>xxxx1xxxxxxxxxxx</code> funcprm_avrInv AVR / INV
/// <code>xxxxx111xxxxxxxx</code> funcprm_range DAC Range / ADC Range
/// <code>xxxxxxxx111xxxxx</code> funcprm_nsamples Number of samples / CAP
/// <code>xxxxxxxxxxx11111</code> funcprm_port Associated port 0..31
#define port_cfg_19_PortCfgFuncID  0xf000
#define port_cfg_19_funcprm_avrInv       0x0800
```

```
#define port_cfg_19_funcprm_range  0x0700
#define port_cfg_19_funcprm_nsamples      0x00e0
#define port_cfg_19_funcprm_port   0x001f
#define port_cfg_19_DESIGNVALUE    0x0000

/// 0x40 r/o adc_data_port_00 PIXI Port 0 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_00_adccode   0x0fff

/// 0x41 r/o adc_data_port_01 PIXI Port 1 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_01_adccode   0x0fff

/// 0x42 r/o adc_data_port_02 PIXI Port 2 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_02_adccode   0x0fff

/// 0x43 r/o adc_data_port_03 PIXI Port 3 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_03_adccode   0x0fff

/// 0x44 r/o adc_data_port_04 PIXI Port 4 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_04_adccode   0x0fff

/// 0x45 r/o adc_data_port_05 PIXI Port 5 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_05_adccode   0x0fff

/// 0x46 r/o adc_data_port_06 PIXI Port 6 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_06_adccode   0x0fff

/// 0x47 r/o adc_data_port_07 PIXI Port 7 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_07_adccode   0x0fff

/// 0x48 r/o adc_data_port_08 PIXI Port 8 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_08_adccode   0x0fff

/// 0x49 r/o adc_data_port_09 PIXI Port 9 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_09_adccode   0x0fff

/// 0x4a r/o adc_data_port_10 PIXI Port 10 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_10_adccode   0x0fff

/// 0x4b r/o adc_data_port_11 PIXI Port 11 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_11_adccode   0x0fff

/// 0x4c r/o adc_data_port_12 PIXI Port 12 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_12_adccode   0x0fff

/// 0x4d r/o adc_data_port_13 PIXI Port 13 Analog to Digital Converter register
```

```
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_13_adccode   0x0fff

/// 0x4e r/o adc_data_port_14 PIXI Port 14 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_14_adccode   0x0fff

/// 0x4f r/o adc_data_port_15 PIXI Port 15 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_15_adccode   0x0fff

/// 0x50 r/o adc_data_port_16 PIXI Port 16 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_16_adccode   0x0fff

/// 0x51 r/o adc_data_port_17 PIXI Port 17 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_17_adccode   0x0fff

/// 0x52 r/o adc_data_port_18 PIXI Port 18 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_18_adccode   0x0fff

/// 0x53 r/o adc_data_port_19 PIXI Port 19 Analog to Digital Converter register
/// <code>xxxx111111111111</code> adccode 12-bit ADC code
#define adc_data_port_19_adccode   0x0fff

/// 0x60 r/w dac_data_port_00 PIXI Port 0 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_00_daccode   0x0fff
#define dac_data_port_00_DESIGNVALUE      0x0000

/// 0x61 r/w dac_data_port_01 PIXI Port 1 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_01_daccode   0x0fff
#define dac_data_port_01_DESIGNVALUE      0x0666

/// 0x62 r/w dac_data_port_02 PIXI Port 2 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_02_daccode   0x0fff
#define dac_data_port_02_DESIGNVALUE      0x0000

/// 0x63 r/w dac_data_port_03 PIXI Port 3 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_03_daccode   0x0fff
#define dac_data_port_03_DESIGNVALUE      0x0000

/// 0x64 r/w dac_data_port_04 PIXI Port 4 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_04_daccode   0x0fff
#define dac_data_port_04_DESIGNVALUE      0x0666

/// 0x65 r/w dac_data_port_05 PIXI Port 5 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_05_daccode   0x0fff
#define dac_data_port_05_DESIGNVALUE      0x0000

/// 0x66 r/w dac_data_port_06 PIXI Port 6 Digital to Analog Converter register
```

```
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_06_daccode   0x0fff
#define dac_data_port_06_DESIGNVALUE      0x0000

/// 0x67 r/w dac_data_port_07 PIXI Port 7 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_07_daccode   0x0fff
#define dac_data_port_07_DESIGNVALUE      0x0000

/// 0x68 r/w dac_data_port_08 PIXI Port 8 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_08_daccode   0x0fff
#define dac_data_port_08_DESIGNVALUE      0x0000

/// 0x69 r/w dac_data_port_09 PIXI Port 9 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_09_daccode   0x0fff
#define dac_data_port_09_DESIGNVALUE      0x0000

/// 0x6a r/w dac_data_port_10 PIXI Port 10 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_10_daccode   0x0fff
#define dac_data_port_10_DESIGNVALUE      0x0000

/// 0x6b r/w dac_data_port_11 PIXI Port 11 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_11_daccode   0x0fff
#define dac_data_port_11_DESIGNVALUE      0x0000

/// 0x6c r/w dac_data_port_12 PIXI Port 12 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_12_daccode   0x0fff
#define dac_data_port_12_DESIGNVALUE      0x0000

/// 0x6d r/w dac_data_port_13 PIXI Port 13 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_13_daccode   0x0fff
#define dac_data_port_13_DESIGNVALUE      0x0000

/// 0x6e r/w dac_data_port_14 PIXI Port 14 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_14_daccode   0x0fff
#define dac_data_port_14_DESIGNVALUE      0x0000

/// 0x6f r/w dac_data_port_15 PIXI Port 15 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_15_daccode   0x0fff
#define dac_data_port_15_DESIGNVALUE      0x0666

/// 0x70 r/w dac_data_port_16 PIXI Port 16 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_16_daccode   0x0fff
#define dac_data_port_16_DESIGNVALUE      0x0666

/// 0x71 r/w dac_data_port_17 PIXI Port 17 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_17_daccode   0x0fff
#define dac_data_port_17_DESIGNVALUE      0x0666
```

```
/// 0x72 r/w dac_data_port_18 PIXI Port 18 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_18_daccode   0x0fff
#define dac_data_port_18_DESIGNVALUE      0x0666

/// 0x73 r/w dac_data_port_19 PIXI Port 19 Digital to Analog Converter register
/// <code>xxxx111111111111</code> daccode 12-bit DAC code
#define dac_data_port_19_daccode   0x0fff
#define dac_data_port_19_DESIGNVALUE      0x0000

#endif /* _MAX11300_DESIGNVALUE_H_ */

// End of file
```

Figure 28. C Header File for MAX11300

13. Trademarks

PIXI is a trademark of Maxim Integrated Products, Inc.

Windows is a registered trademark and registered service mark and Windows XP is a registered trademark of Microsoft Corporation.

14. Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/15	Initial release	—

OCEAN CHIPS
Океан Электроники
Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «**JONHON**», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «**FORSTAR**».

AVIC JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А