Configuring the ADC with MCUXpresso Config Tools

Author : Gert van Biljon

email : gertvb@gmail.com

https://community.nxp.com/t5/LPC-Microcontrollers/LPC55S69-Configtool-ADC/td-p/1168980

Herewith just a short overview of the process that I used to create the “LPC55S69_Configtool_ADC” MCUXpresso project for the OKDO E1 board which uses the lpc55s69 MCU.

My aim is to simply get the ADC running by setting it up using the SDK Wizard and MCUXpresso Config Tools, instead of having to shove values into registers all over the show.

When I debug the project, I read 32768 from all the inputs, but if I first run the lpcxpresso55s69_lpadc_polling sample, I read the correct values.

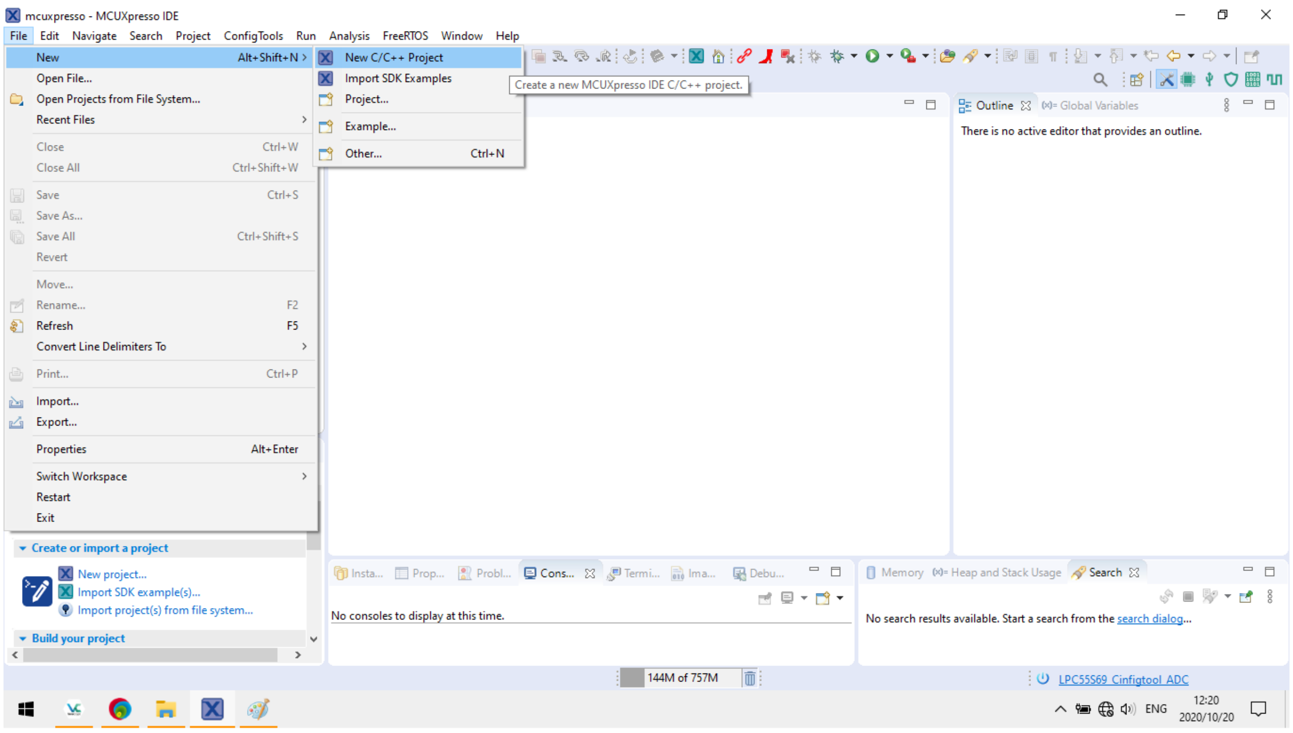
This clearly shows that something is not correctly configured using the SDK wizard and MCUXpresso Config Tools

Note that the OKDO E1 board does not have the 32MHz crystal, and I use the 12MHz free running oscillator through PLL1 as clock source and PLL0 as clock source for the ADC. (Refer to Mark from EmbeddedPro’s excellent YouTube video on doing it)

To start off with:

1. Open MCUXpresso IDE and close all the open projects

2. Select File -> New -> New C/C++ project

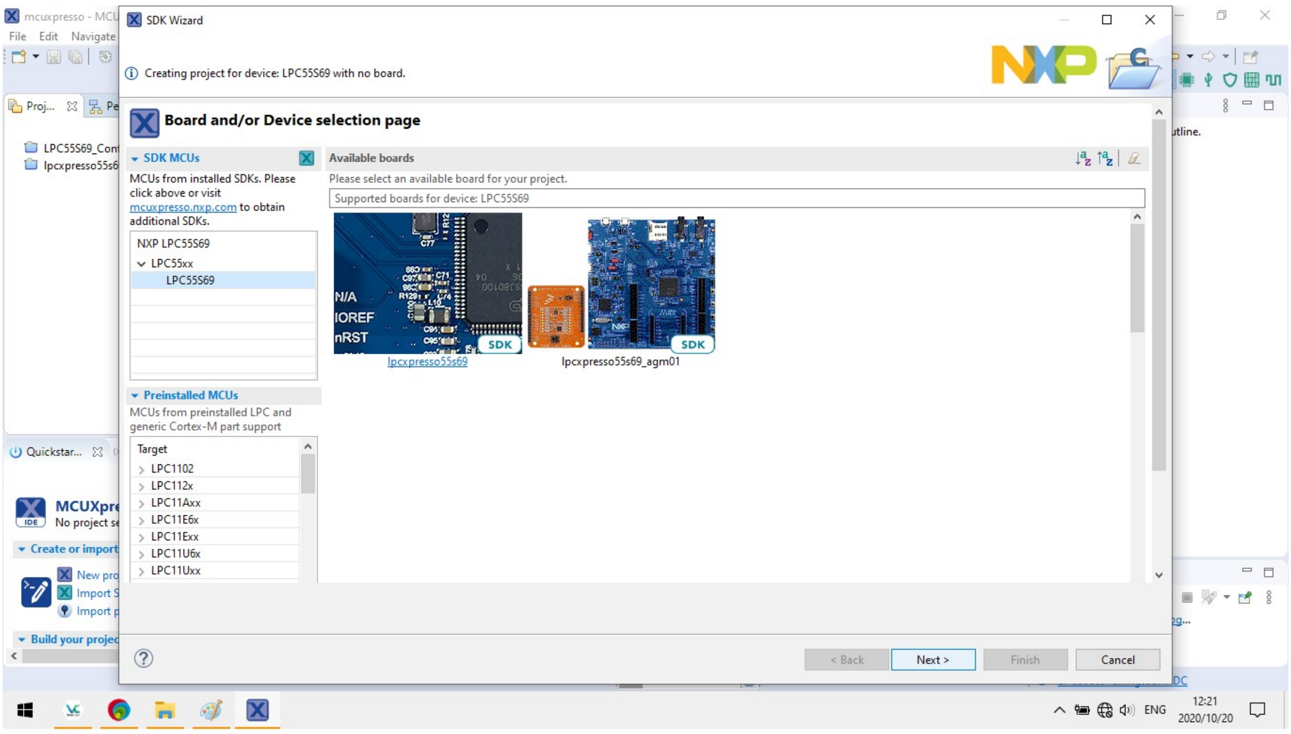


There is no active editor that provides an outline.

No consoles to display at this time.

No search results available. Start a search from the search dialog...

3. Choose ONLY the LPC55S69 processor! And NOT one of the lpcxpresso dev board, and press next



SDK Wizard

Creating project for device: LPC55S69 with no board.

Board and/or Device selection page

SDK MCUs

MCUs from installed SDKs. Please click above or visit mcuxpresso.nxp.com to obtain additional SDKs.

NXP LPC55S69
LPC55xx
LPC55S69

Available boards

Please select an available board for your project.

Supported boards for device: LPC55S69

lpcxpresso55s69
lpcxpresso55s69_agm01

Preinstalled MCUs

MCUs from preinstalled LPC and generic Cortex-M part support

Target
LPC1102
LPC112x
LPC11Axx
LPC11E6x
LPC11Exx
LPC11U6x
LPC11Uxx

< Back | Next > | Finish | Cancel

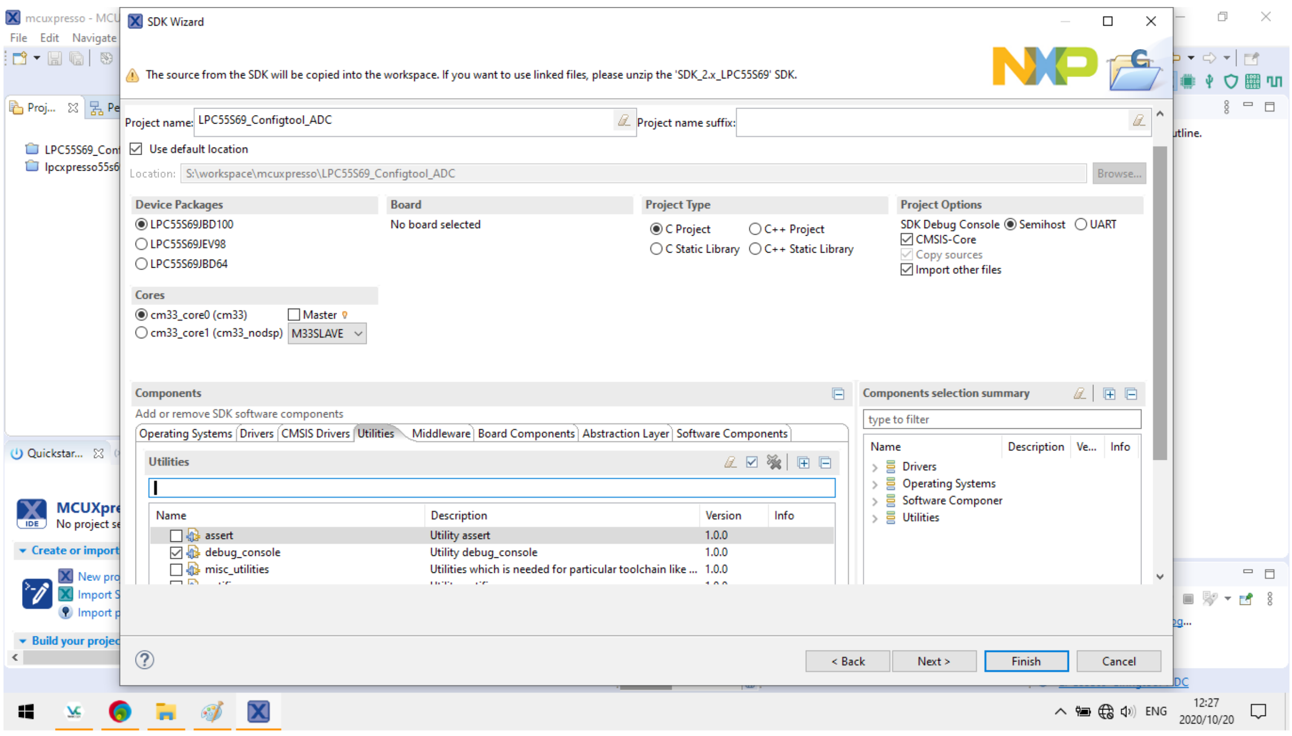
4. On the SDK wizard screen, give the project a name and select the following components:

Operating System Tab -> Bare metal

Drivers -> Gpio an Lpadc

Utilities -> Debug Console

-> Finish



SDK Wizard

The source from the SDK will be copied into the workspace. If you want to use linked files, please unzip the 'SDK_2.x_LPC55S69' SDK.

Project name: LPC55S69_Configtool_ADC
Project name suffix:

[✓] Use default location

Location: S:\workspace\mcuxpresso\LPC55S69_Configtool_ADC

Device Packages
- [● LPC55S69JBD100]
- LPC55S69JEV98
- LPC55S69JBD64

Board
No board selected

Project Type
- [● C Project]
- C++ Project
- C Static Library
- C++ Static Library

Project Options
- SDK Debug Console [● Semihost] UART
- [✓] CMSIS-Core
- [✓] Copy sources
- [✓] Import other files

Cores
- [● cm33_core0 (cm33)]
- cm33_core1 (cm33_nodsp)
- Master
- M33SLAVE

Components
Add or remove SDK software components

Operating Systems | Drivers | CMSIS Drivers | Utilities | Middleware | Board Components | Abstraction Layer | Software Components

Utilities

Name	Description	Version	Info
assert	Utility assert	1.0.0	
[✓] debug_console	Utility debug_console	1.0.0	
misc_utilities	Utilities which is needed for particular toolchain like ...	1.0.0	

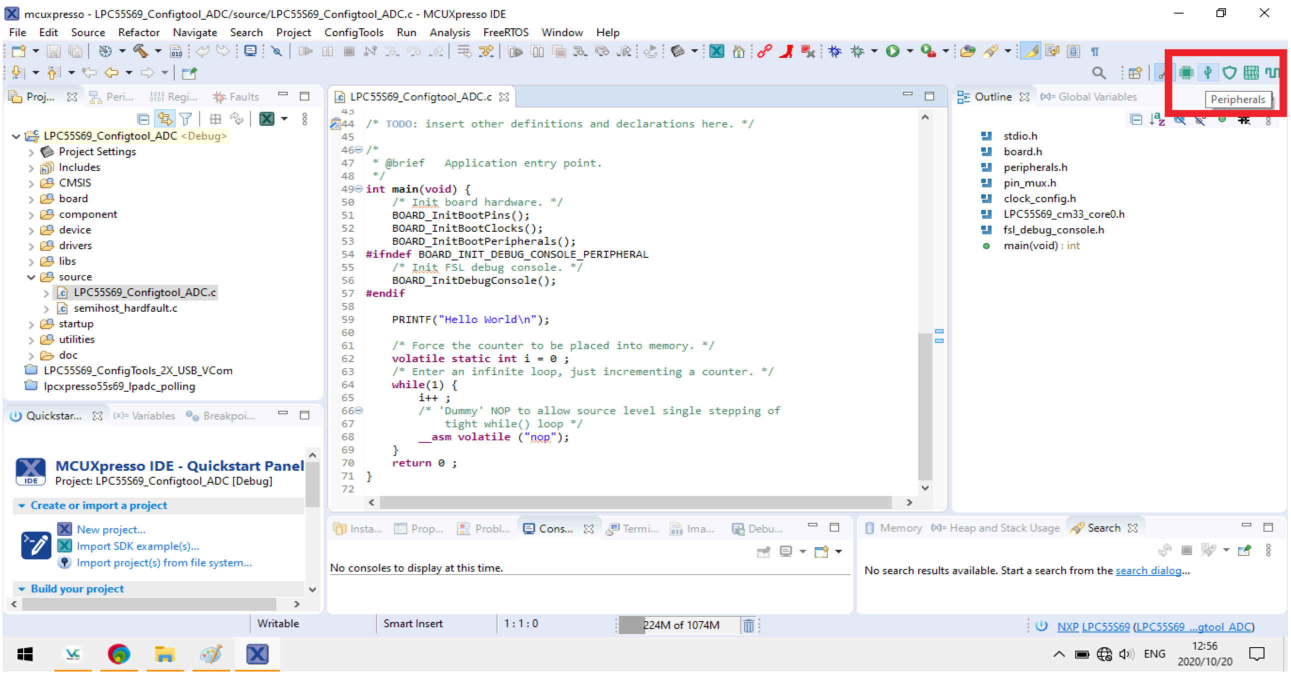
Components selection summary

type to filter

Name	Description	Ve...	Info
Drivers			
Operating Systems			
Software Componer			
Utilities			

5. the SDK wizard will disappear into the wild blue yonder for a while, and then return you to the code editor in MCUXpresso

Select the Peripherals button in the top right . . .


mcuxpresso - LPC55S69_Configtool_ADC/source/LPC55S69_Configtool_ADC.c - MCUXpresso IDE

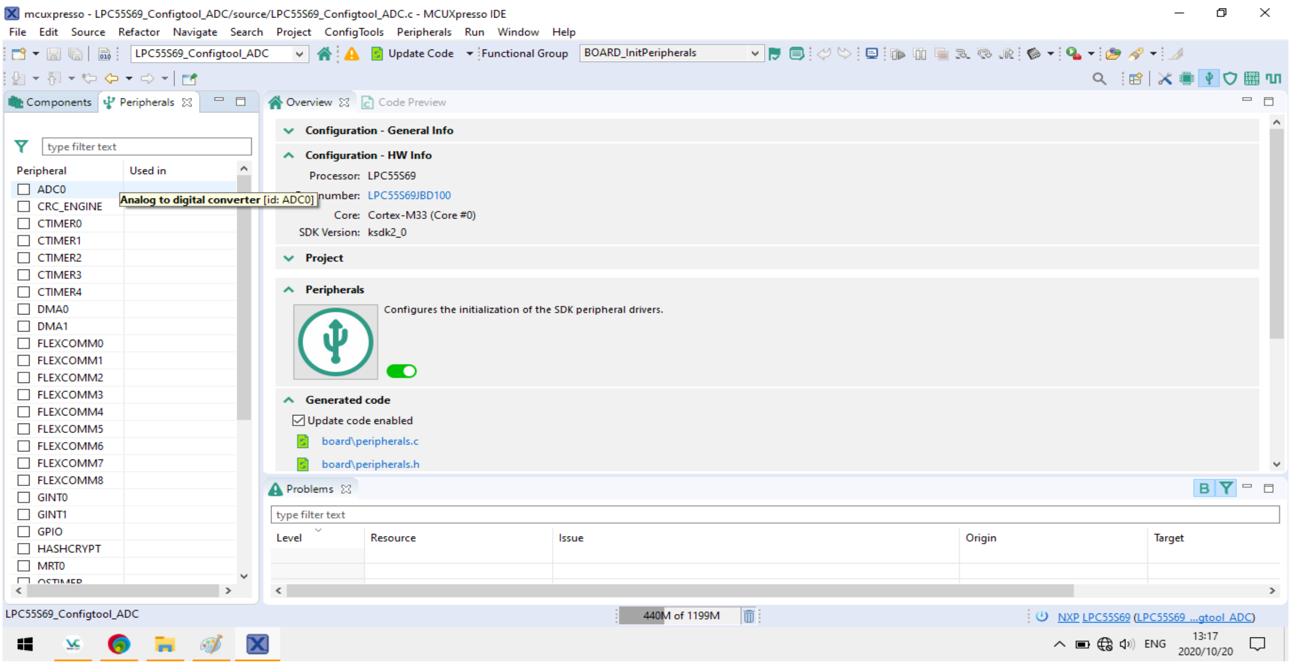
```
/* TODO: insert other definitions and declarations here. */

/*
 * @brief   Application entry point.
 */
int main(void) {
    /* Init board hardware. */
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
    /* Init FSL debug console. */
    BOARD_InitDebugConsole();
#endif

    PRINTF("Hello World\n");

    /* Force the counter to be placed into memory. */
    volatile static int i = 0 ;
    /* Enter an infinite loop, just incrementing a counter. */
    while(1) {
        i++ ;
        /* 'Dummy' NOP to allow source level single stepping of
            tight while() loop */
        __asm volatile ("nop");
    }
    return 0 ;
}
```

No consoles to display at this time.

No search results available. Start a search from the search dialog...

File Edit Source Refactor Navigate Search Project ConfigTools Run Analysis FreeRTOS Window Help
Peripherals
LPC55S69_Configtool_ADC <Debug>
Project Settings
Includes
CMSIS
board
component
device
drivers
libs
source
LPC55S69_Configtool_ADC.c
semihost_hardfault.c
startup
utilities
doc
LPC55S69_ConfigTools_2X_USB_VCom
lpcxpresso55s69_lpadc_polling
MCUXpresso IDE - Quickstart Panel
Project: LPC55S69_Configtool_ADC [Debug]
Create or import a project
New project...
Import SDK example(s)...
Import project(s) from file system...
Build your project
stdio.h
board.h
peripherals.h
pin_mux.h
clock_config.h
LPC55S69_cm33_core0.h
fsl_debug_console.h
main(void) : int
Writable
Smart Insert
1 : 1 : 0
224M of 1074M
NXP LPC55S69 (LPC55S69 ...gtool_ADC)
ENG
12:56
2020/10/20

Which will drop you into the peripherals configuration screen,


mcuxpresso - LPC55S69_Configtool_ADC/source/LPC55S69_Configtool_ADC.c - MCUXpresso IDE

Configuration - General Info

Configuration - HW Info

Processor: LPC55S69
Part number: LPC55S69JBD100
Core: Cortex-M33 (Core #0)
SDK Version: ksdk2_0

Project

Peripherals

Configures the initialization of the SDK peripheral drivers.

Generated code

Update code enabled

board\peripherals.c

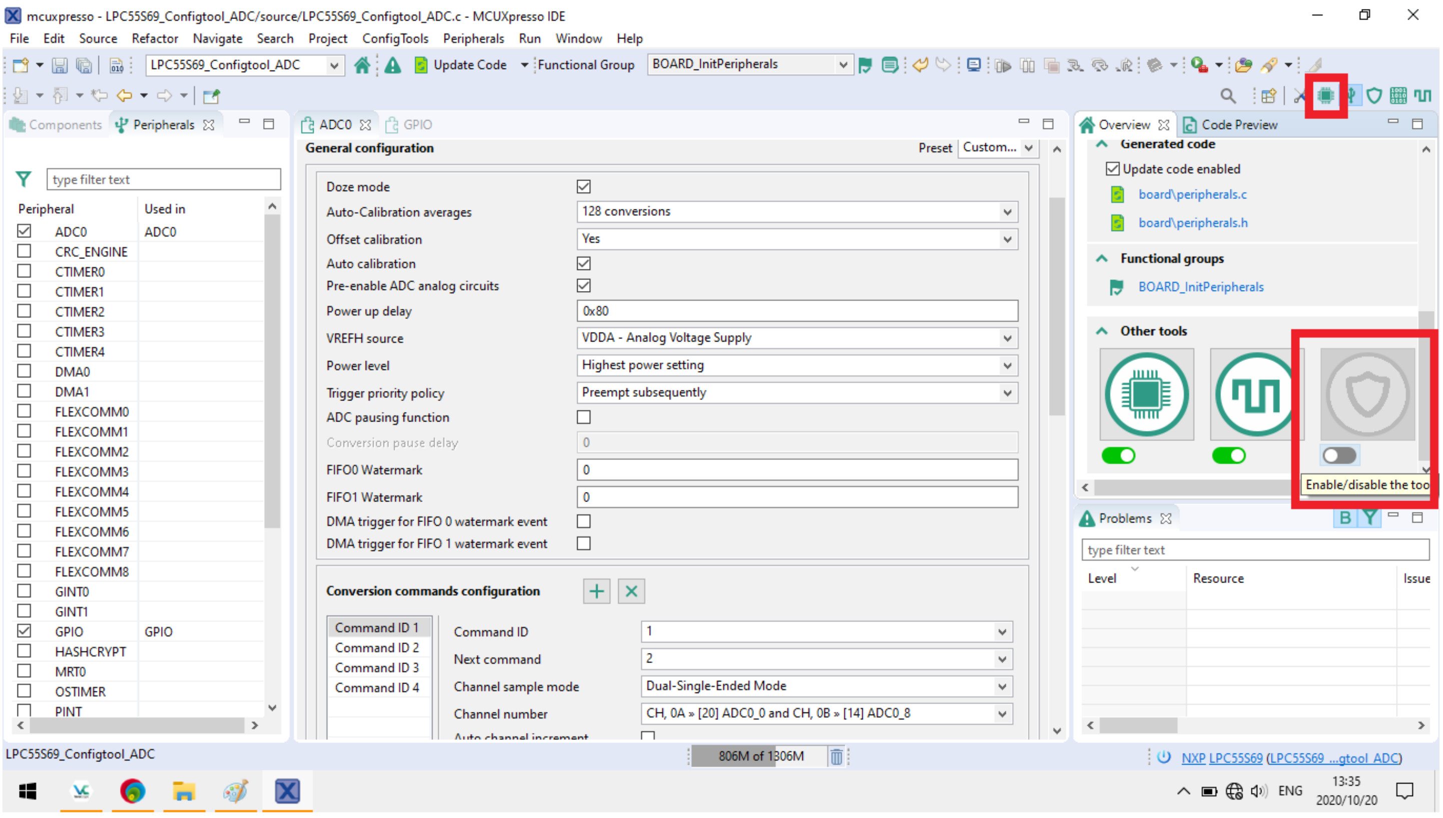
board\peripherals.h

Peripheral	Used in
ADC0	
CRC_ENGINE	
CTIMER0	
CTIMER1	
CTIMER2	
CTIMER3	
CTIMER4	
DMA0	
DMA1	
FLEXCOMM0	
FLEXCOMM1	
FLEXCOMM2	
FLEXCOMM3	
FLEXCOMM4	
FLEXCOMM5	
FLEXCOMM6	
FLEXCOMM7	
FLEXCOMM8	
GINT0	
GINT1	
GPIO	
HASHCRYPT	
MRT0	

Analog to digital converter [id: ADC0]

6. Select the Gpio and select the ADC and populate the values, as well as a few ADC conversion commands and an ADC trigger

Also deselect the TEE functionality


mcuxpresso - LPC55S69_Configtool_ADC/source/LPC55S69_Configtool_ADC.c - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project ConfigTools Peripherals Run Window Help

LPC55S69_Configtool_ADC | Update Code | Functional Group | BOARD_InitPeripherals

Components | Peripherals

Peripheral	Used in
☑ ADC0	ADC0
☐ CRC_ENGINE	
☐ CTIMER0	
☐ CTIMER1	
☐ CTIMER2	
☐ CTIMER3	
☐ CTIMER4	
☐ DMA0	
☐ DMA1	
☐ FLEXCOMM0	
☐ FLEXCOMM1	
☐ FLEXCOMM2	
☐ FLEXCOMM3	
☐ FLEXCOMM4	
☐ FLEXCOMM5	
☐ FLEXCOMM6	
☐ FLEXCOMM7	
☐ FLEXCOMM8	
☐ GINT0	
☐ GINT1	
☑ GPIO	GPIO
☐ HASHCRYPT	
☐ MRT0	
☐ OSTIMER	
☐ PINT	

ADC0 | GPIO

General configuration — Preset: Custom...

Doze mode	☑
Auto-Calibration averages	128 conversions
Offset calibration	Yes
Auto calibration	☑
Pre-enable ADC analog circuits	☑
Power up delay	0x80
VREFH source	VDDA - Analog Voltage Supply
Power level	Highest power setting
Trigger priority policy	Preempt subsequently
ADC pausing function	☐
Conversion pause delay	0
FIFO0 Watermark	0
FIFO1 Watermark	0
DMA trigger for FIFO 0 watermark event	☐
DMA trigger for FIFO 1 watermark event	☐

Conversion commands configuration

Command ID 1 / Command ID 2 / Command ID 3 / Command ID 4

Command ID	1
Next command	2
Channel sample mode	Dual-Single-Ended Mode
Channel number	CH, 0A » [20] ADC0_0 and CH, 0B » [14] ADC0_8
Auto channel increment	☐

Overview | Code Preview

Generated code
☑ Update code enabled
board\peripherals.c
board\peripherals.h

Functional groups
BOARD_InitPeripherals

Other tools
Enable/disable the too

Problems — type filter text

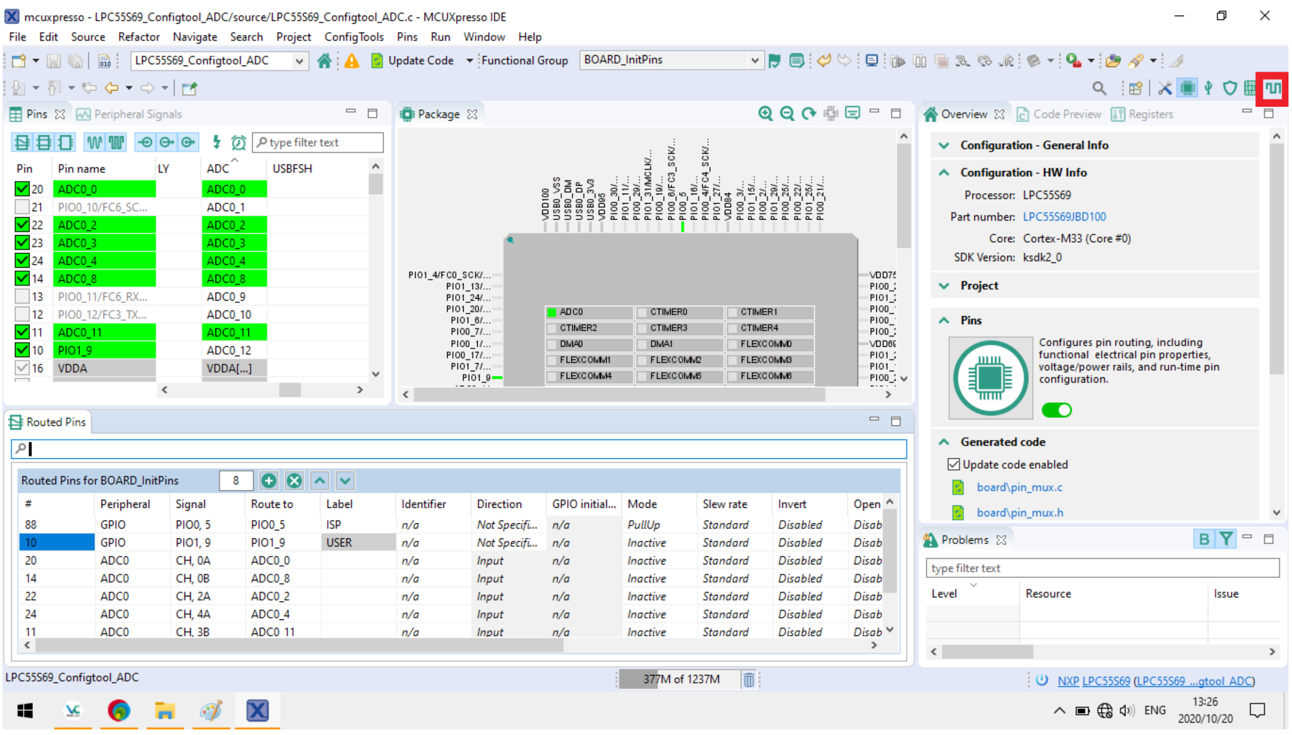
Level	Resource	Issue

LPC55S69_Configtool_ADC | 806M of 1306M | NXP LPC55S69 (LPC55S69 ...gtool ADC)

ENG 13:35 2020/10/20

When finished, select the Pins tool, to go and route input pins to their desired functionality...

7. Route all the inputs to their destinations


mcuxpresso - LPC55S69_Configtool_ADC/source/LPC55S69_Configtool_ADC.c - MCUXpresso IDE

Pin	Pin name	LY	ADC	USBFSH
✓ 20	ADC0_0		ADC0_0	
21	PIO0_10/FC6_SC...		ADC0_1	
✓ 22	ADC0_2		ADC0_2	
✓ 23	ADC0_3		ADC0_3	
✓ 24	ADC0_4		ADC0_4	
✓ 14	ADC0_8		ADC0_8	
13	PIO0_11/FC6_RX...		ADC0_9	
12	PIO0_12/FC3_TX...		ADC0_10	
✓ 11	ADC0_11		ADC0_11	
✓ 10	PIO1_9		ADC0_12	
✓ 16	VDDA		VDDA[...]	

Configuration - General Info

Configuration - HW Info

Processor:	LPC55S69
Part number:	LPC55S69JBD100
Core:	Cortex-M33 (Core #0)
SDK Version:	ksdk2_0

Project

Pins

Configures pin routing, including functional electrical pin properties, voltage/power rails, and run-time pin configuration.

Generated code

Update code enabled

board\pin_mux.c

board\pin_mux.h

Routed Pins

Routed Pins for BOARD_InitPins 8

#	Peripheral	Signal	Route to	Label	Identifier	Direction	GPIO initial...	Mode	Slew rate	Invert	Open
88	GPIO	PIO0, 5	PIO0_5	ISP	n/a	Not Specifi...	n/a	PullUp	Standard	Disabled	Disab
10	GPIO	PIO1, 9	PIO1_9	USER	n/a	Not Specifi...	n/a	Inactive	Standard	Disabled	Disab
20	ADC0	CH, 0A	ADC0_0		n/a	Input	n/a	Inactive	Standard	Disabled	Disab
14	ADC0	CH, 0B	ADC0_8		n/a	Input	n/a	Inactive	Standard	Disabled	Disab
22	ADC0	CH, 2A	ADC0_2		n/a	Input	n/a	Inactive	Standard	Disabled	Disab
24	ADC0	CH, 4A	ADC0_4		n/a	Input	n/a	Inactive	Standard	Disabled	Disab
11	ADC0	CH, 3B	ADC0_11		n/a	Input	n/a	Inactive	Standard	Disabled	Disab

Problems

type filter text

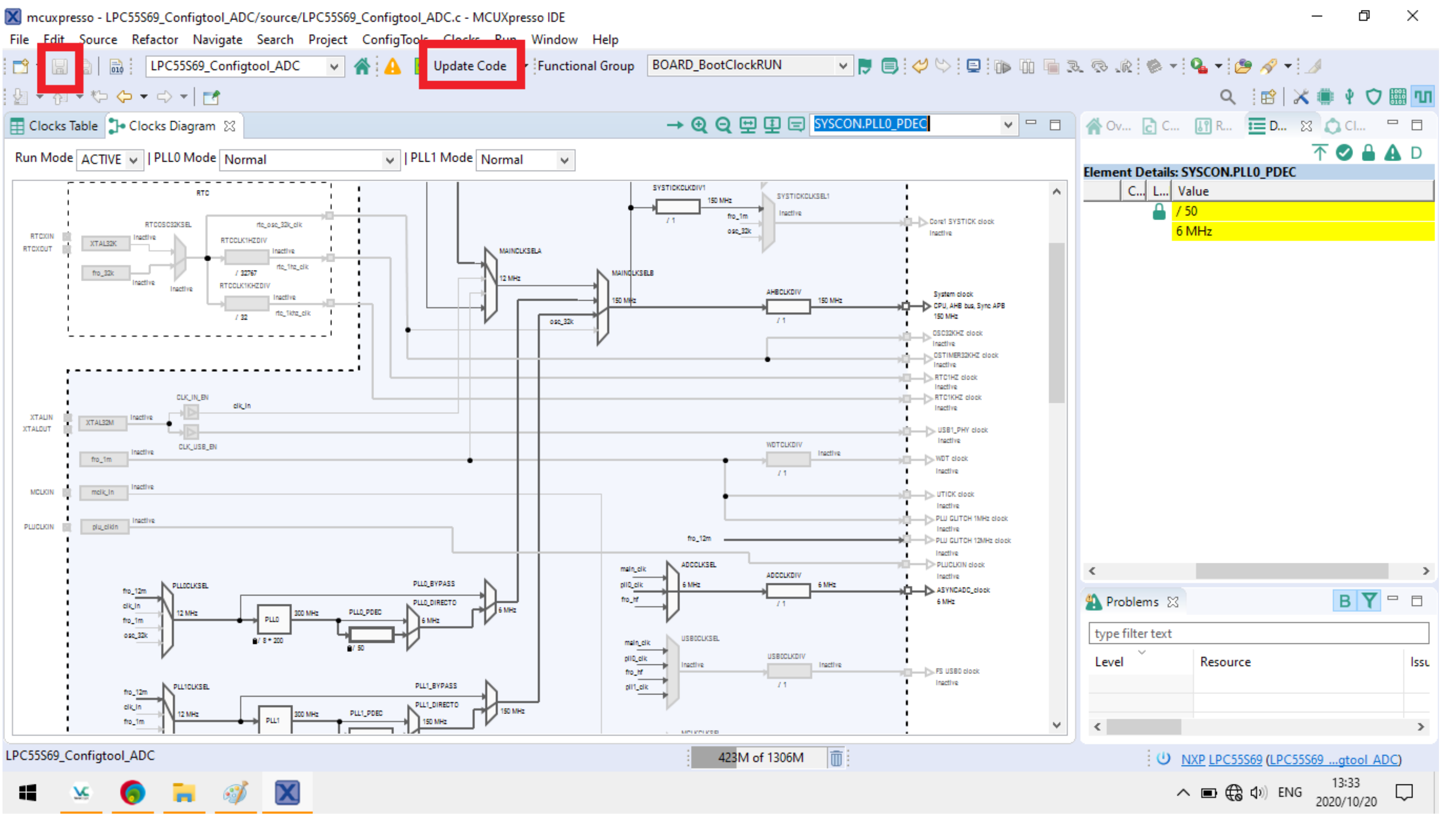
Level	Resource	Issue

LPC55S69_Configtool_ADC   377M of 1237M   NXP LPC55S69 (LPC55S69 ...gtool_ADC)

ENG 13:26 2020/10/20

Select the clocks button top right to set up the clock sources....

8. Configure your clocks


mcuxpresso - LPC55S69_Configtool_ADC/source/LPC55S69_Configtool_ADC.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Clocks Run Window Help
Update Code
Functional Group
BOARD_BootClockRUN
Clocks Table
Clocks Diagram
SYSCON.PLL0_PDEC
Run Mode ACTIVE | PLL0 Mode Normal | PLL1 Mode Normal
Element Details: SYSCON.PLL0_PDEC
/ 50
6 MHz
Problems
LPC55S69_Configtool_ADC
423M of 1306M

Save the configuration, and VERY importantly click on “Update code” which will generate code from your configuration

The Config Tool will now exit and MCUXpresso will take you back to the Code Editor

9. Add the following code to the while(1) loop to read and print the ADC values:

```
int main(void) {
        /* Init board hardware. */
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
    /* Init FSL debug console. */
    BOARD_InitDebugConsole();
#endif

    PRINTF("Hello All! OKDO E1 Board; LPC55S69 MCU; ADC configured with MCUXpresso Config Tools\n");

    lpadc_conv_result_t mLpadcResultConfigStruct;
    uint32_t conv_result_count;

    while(1)
    {
        LPADC_DoSoftwareTrigger(ADC0_PERIPHERAL, 1U); /* 1U is trigger0 mask. */

        conv_result_count = LPADC_GetConvResultCount(ADC0_PERIPHERAL, 0);
        while (conv_result_count < 6)
        {
            conv_result_count = LPADC_GetConvResultCount(ADC0_PERIPHERAL, 0);
        }
        PRINTF("Result fifo0, Count %d;  ", LPADC_GetConvResultCount(ADC0_PERIPHERAL, 0));

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d;  ", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d;  ", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d;  ", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d;  ", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d;  ", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);

        while (!LPADC_GetConvResult(ADC0_PERIPHERAL, &mLpadcResultConfigStruct, 0U))
        {
            __asm volatile ("nop");
        }
        PRINTF("ADC config %d: %5d\r", mLpadcResultConfigStruct.commandIdSource, mLpadcResultConfigStruct.convValue);
    }

    return 0 ;
}
```