Atmel

TRAINING MANUAL

Single-slope PWM Implementation using SAM D21 TCC in a Touch-based Application

AN-9232

Prerequisites

- Hardware Prerequisites
 - Atmel[®] SAM D21 Xplained Pro Evaluation Kit
 - Atmel QT1 Xplained Pro Kit
 - USB Micro Cable (TypeA/Micro-B)

Software Prerequisites

- Atmel Studio 6.2 or higher
- Atmel Software Framework 3.17.0 or higher
- Atmel QTouch[®] Composer 5.3 or higher
- Atmel QTouch Library 5.3 or higher

Documentation Prerequisites

- Atmel-42271-Introduction-to-QTouch-Design-Parameters-using-SAM-D21-Xplained-Pro_Training-Manual_AN-7846 Training Manual
- Estimated completion time: 60min.

Introduction

The goal of this hands-on is to:

- Create a Touch project using QTouch Project Builder
- Use QTouch Analyzer to test the project
- Implement a PWM signal using the SAM D21 Timer/Counters for Control peripheral (TCC) to drive the brightness of a LED
- Modify the LED brightness in relation to the Touch Delta value of a Button sensor by updating the PWM duty cycle



Table of Contents

Pre	requis	ites	1				
Intro	oductio	on	1				
Icor	n Key	Identifiers	3				
1.	Train	ing Module Architecture	4				
	1.1	Atmel Studio Extension (.vsix)	4				
	1.2	Atmel Training Executable (.exe)	4				
2.	Introduction5						
3.	Assig Builde	nment 1: Create a Touch project using QTouch Project er	6				
	3.1	Project Creation	6				
	3.2	Conclusion	18				
4.	Assig	nment 2: Use QTouch Analyzer to Test the Project	.19				
	4.1	Connect the Kit to QTouch Analyzer	19				
	4.2	Test the Proximity Effect of the QTouch Application	23				
	4.3	Conclusion	24				
5.	Assig Timei	nment 3: Implement a PWM Signal using the SAM D21 r/Counters for Control Application Peripheral (TCC)	.25				
	5.1	Introduction to Timer/Counters for Control Application (TCC)	25				
	5.2	TCC Peripheral Initialization Using ASF Quick Start Guide	27				
	5.3	Conclusion	34				
6.	Assig	nment 4: Use QTouch Button to Control LED Brightness	335				
	6.1	Reading and using Button Touch Delta Value	35				
	6.2	Using PWM to Control LED Brightness	37				
	6.3	Conclusion	41				
7.	Conc	lusion	.42				
8.	Revision History43						



Icon Key Identifiers

i	INFO	Delivers contextual information about a specific topic.
<u>\</u> _	TIPS	Highlights useful tips and techniques.
	TO DO	Highlights objectives to be completed.
	RESULT	Highlights the expected result of an assignment step.
	WARNING	Indicates important information.
•	EXECUTE	Highlights actions to be executed out of the target when necessary.



1. Training Module Architecture

This training material can be retrieved through different Atmel deliveries:

- As an Atmel Studio Extension (.vsix file) usually found on the Atmel Gallery web site (http://gallery.atmel.com/) or using the Atmel Studio Extension manager
- As an Atmel Training Executable (.exe file) usually provided during Atmel Training sessions

Depending on the delivery type, the different resources needed by this training material (hands-on documentation, datasheets, application notes, software, and tools) will be found on different locations.

1.1 Atmel Studio Extension (.vsix)

Once the extension installed, you can open and create the different projects using "*New Example Project from ASF...*" in Atmel Studio.



The projects installed from an extension are usually under "Atmel Training > Atmel Corp. Extension Name".

There are different projects which can be available depending on the extension:

- Hands-on Documentation: contains the documentation as required resources
- Hands-on Assignment: contains the initial project that may be required to start
- Hands-on Solution: contains the final application which is a solution for this hands-on



Each time a reference is made to some resources in the following pages, the user must refer to the Hands-on Documentation project folder.

1.2 Atmel Training Executable (.exe)

Depending where the executable has been installed, you will find the following architecture which is composed by two main folders:

- AN-XXXX Hands-on: contains the initial project that may be required to start and a solution
- Resources: contains required resources (datasheets, software, and tools...)



Unless a specific location is specified, each time a reference is made to some resources in the following pages, the user must refer to this **Resources** folder.



2. Introduction

This hands-on will focus on the development of a simple Touch application that uses a Button sensor from the Atmel QT1 Xplained Pro Self Capacitance extension board to modify the brightness of a LED.

The LED brightness is controlled using a PWM signal that is generated by one of the Atmel SAM D21 Timer/Counters for Control (TCC) peripherals.

The SAM D21 embeds three different 24-bit Timer/Counters for Control peripherals which provide extended functions compared to the standard Timer/Counters (TC):

- Up to four compare channels with optional complementary outputs
- Advanced waveform signal generation with dead time management
- Deterministic fault protection, fast decay, and configurable dead-time between complementary outputs
- Dithering that increase resolution with up to five bits and reduce quantization error

The hands-on will be divided in the following assignments:

• Create a new Touch project based on BUTTON 1 Touch sensor using QTouch Project Builder from Atmel QTouch Composer:



- Use QTouch Analyzer to provide graphical representation and demonstration of real time touch data. This will allow testing the newly-created Touch project.
- Implement a PWM signal using the TCC0 peripheral to control the LED8 brightness
 - Atmel Software Framework (ASF) TCC Quick Start Guide will be used to implement it easily
 - A Waveform Output pin (WO6) from the SAM D21 MCU will be used to output the PWM signal directly to the LED8
- Modify the LED brightness in relation to the Touch Delta value of BUTTON 1 by updating the PWM duty cycle
 - Atmel QTouch library API will be used to get the delta value of BUTTON 1 sensor



3. Assignment 1: Create a Touch project using QTouch Project Builder

3.1 **Project Creation**

🔡 то ро

O Create a new QTouch Project.

- Open Atmel Studio 6.2
- Click on the "QT" icon to open the QTouch Start Page:
- i info

QTouch StartPage is also accessible through the Tools menu:

2 α

Tool	s Window Help		
>	Command Prompt		sensor_state 🔹
4	Device Programming	Ctrl+Shift+P	🏧 🖪 🚽 🖾 🚟 🛛 🛶 🖉
2	Add target		
	QTouchComposer	•	CT QTouch StartPage
	Code Snippets Manager	Ctrl+K, Ctrl+B	QTouch Analyzer

Select "New QTouch UserBoard Project..."



• Ensure "GCC QTouch Executable Project" is selected (default)*





• Fill-in New Project fields according to following use cases:

Atmel Training Executable Case

- Name: Hands-On Assignment
- Location: "AN-9232_SAMD21-XPRO_QTouch_TCC\assignments" (relative path in the ATMEL_TRAINING installation folder)
- Solution name: Hands-On Assignment
- Click OK

Atmel Extension Case (downloaded from Atmel Gallery or Studio Extension Manager)

- **Name**: Hands-On Assignment
- Location: existing Hands-on Documentation solution path
- Solution name: Hands-On Documentation
- Click OK

New Project	with Liferency		? ×	
Recent Templates	Sort by: Default		Search Installed Templates	
Installed Templates C/C++	GCC C ASF Board Project	C/C++	Type: C/C++ Creates an AVR 8-bit or AVR/ARM 32-bit	
Assembler Atmel Studio Solution	GCC C Executable Project	C/C++	QTouch C project	
	GCC C QTouch Executable Project	C/C++		
	GCC C Static Library Project	C/C++		
	GCC C++ Executable Project	C/C++		
	GCC C++ Static Library Project	C/C++		
Name: Hands-on As	signment			
Location: C:\Users\fboyer\Documents\Atmel Studio\6.2				
Solution name: Hands-on Do	ocumentation		Create directory for solution	
			OK Cancel	



The QTouch Project Builder wizard will now guide through the steps involved in creating a QTouch project.

• Click "Next" in the Getting Started page (if shown):



The Kit design Page provides an option to add, delete or move QTouch Sensors such as button, wheel and slider. It also provides options to setup physical and firmware properties of the board and sensors.



• Add a button in the *Kit Design* page by clicking on the "Button" icon, click "Add" and "Next"



The Technology Selection Guide recommends which acquisition technology should be used based on the number of sensors previously selected in Kit Design page and answers from the below questionnaire.

Rouch Project Builder	
Guide for Selecting Touch Technology - Page 3 of 8	
	Touch Technology At Glance
Will your application expose touch sensors to moisture or water droplets like in the case Washing Machine or Cooktop appliance ? Yes Ves No Will your application hardware design involve LCD or other potential sources of radiated noise directly behind the touch sensors ?	QTouch Technology The QTouch® devices are charging a sense electrode of unknown capacitance to a known potential. The electrode is typically a copper area on a printed circuit board. The resulting charge is transferred into a measurement circuit QMatrix Technology QMatrix devices are touch sensor ICs that detect touch using a scanned passive matrix of leastinge and canned passive
With the Configuration provided, we suggest you to select technology as in next page QTouch	Electrodes are typically areas of copper on a printed circuit board but can also be areas of clear conductive indium tin oxide (ITO) on a glass or plastic touch screen. A single QMatrix device can drive a large number of keys, enabling a very low cost- per-key to be achieved
Help < Back Next >	Finish Cancel

• We will consider for this hands-on that our application hardware design is NOT exposed to moisture or water droplets or any specific sources of radiated noises

So, select "No" as the answer to Question 1 and 2. You will see that the proposed technology will become QTouch (i.e. Self Capacitance Technology):



i info

For more information about designing the touch sensor, refer to *Buttons, Sliders and Wheels Touch Sensor Design Guide* available at www.atmel.com/images/doc10752.pdf.

Click "Next" in the Guide for Selecting Touch Technology page



Device Selection Page provides the option to view the number of QTouch sensors such as Button, Wheel and Slider that was selected in the Kit Design Page.

Touch sensors can be based on two sub-technologies, **QTouch/Self Capacitance** and **QMatrix/Mutual Capacitance**.

• We can verify that **QTouch/Self Capacitance** has been correctly selected following our previous choice in the *Guide for Selecting Touch Technology* page

Select Device - Page 4 of 8							
Sensors	Button: 1	Wheel : 0	Slider: 0				
Technology	QTouch/Self	Capacitance 🔘 QI	Matrix/Mutual Capacita	nce			

The device list displays device details such as Device Name, Variant, Application/Boot Memory, Data Memory, EEPROM and supported technology. Data listing will vary based on device family, number of selected sensors and technology as only devices supporting all choices made will be shown. Additionally user has the option to search for a particular device.

- Type "SAMD21" in the search box to make a first filtering
- Select ATSAMD21J18A in the *Select Device* page and click "Next". This is the device on the SAM D21 Xplained Pro kit.

Sensors Button : 1 Wheel : 0 Slider : 0 Device Information Technology QTouch/Self Capacitance QMatrix/Mutual Capacitance Device Name: ATSAMD21 Device Family All SAMD21 Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix Vcc 0/0 ATSAMD21E15A 32 32 4096 0 ✓ ✓ Touch QTouch (bytes) 0 O ✓ Touch As ATSAMD21E17A 32 128 16384 0 ✓ ✓ OTouch Max Wheels/Sliders 5 Max Channels 16 ATSAMD21G16A 48 64 8192 0 ✓ ✓ Data Memory used Ode Memory use										8	Page 4 of	- P	elect Device
Sensors Button : 1 Wheel : 0 Slider : 0 Technology QTouch/Self Capacitance QMatrix/Mutual Capacitance Device Name: ATSAMD21 Device Family All SAMD21 SAMD2 Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix ATSAMD21E15A 32 32 4096 Speed 0 Vcc 0/0 Ports AB TSAMD21E16A 32 64 8192 0 QTouch Library Informatic ATSAMD21616A 48 64 8192 0 Max Channels 16 Code Memory used 0 Max Channels 16 TSAMD2117A 48 128 16384 0 Image: Code Memory used		formation	Device Inf										
Technology QTouch/Self Capacitance QMatrix/Mutual Capacitance Family SAMD Device Family All All SAMD21 SAMD21 SAMD21E15A 32 32 4096 	1J18A	ATSAMD21J	Device Name:				0	Slider : (Wheel: 0	n: 1	Butto		Sensors
Device Family All SAMD21 Speed 0 Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix Ag ATSAMD21E15A 32 32 4096 0 ✓ ✓ ATSAMD21E16A 32 64 8192 0 ✓ ✓ ATSAMD21E17A 32 128 16384 0 ✓ ✓ ATSAMD21616A 86 64 8192 0 ✓ ✓ ATSAMD21616A 86 64 8192 ✓ ✓ Max Wheels/Sliders 5 ATSAMD21616A 48 64 8192 ✓ ✓ ✓ Max Wheels/Sliders 5 ATSAMD2117A 48 128 16384 ✓ ✓ ✓ Max Channels 16 Code Memory used ✓ ✓ ✓ ✓ ✓ ✓ ✓ ATSAMD2117A 64 256 32768 ✓ ✓ Max Channels 16 Code Memory used ✓ ✓ ✓ ✓ <td< th=""><th></th><th>SAMD</th><th>Family</th><th></th><th></th><th>itance</th><th>l Cana</th><th>OMatrix/Mutua</th><th>nacitance (</th><th>ouch/Self Can</th><th>OT</th><th></th><th>Technology</th></td<>		SAMD	Family			itance	l Cana	OMatrix/Mutua	nacitance (ouch/Self Can	OT		Technology
Device Family All ✓ SAMD21 Speed 0 Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix AB ATSAMD21E15A 32 32 4096 0 ✓ ✓ AB ATSAMD21E16A 32 64 8192 0 ✓ ✓ QTouch Library Informatic ATSAMD21E17A 32 128 16384 0 ✓ ✓ QTouch Library Informatic ATSAMD21616A 48 64 8192 0 ✓ ✓ Max Wheels/Sliders 5 ATSAMD21616A 48 64 8192 0 ✓ ✓ Max Wheels/Sliders 5 ATSAMD21617A 48 16384 0 ✓ ✓ Data Memory used O Ode Max Wheels/Sliders 5 ATSAMD2117A 64 128 16384 0 ✓ ✓ Data Memory used Ode Max Wheels/Sliders 8 ATSAMD2117A 64 256 32768 ✓ ✓ Max Wheels/Sliders 8 A		0	EEPROM (bytes)			numee	ii cupe	Civia ci ix, ivia caa	pacitance	ouen/oen eap	୍ୟୁ		icennoiog)
Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix Ports A,B ATSAMD21E15A 32 32 4096 0 ✓ ✓ ATSAMD21E16A 32 64 8192 0 ✓ ✓ ATSAMD21E17A 32 128 16384 0 ✓ ✓ ✓ ATSAMD21E17A 32 256 32768 0 ✓ ✓ Max Wheels/Sliders 5 ATSAMD21G16A 48 64 8192 0 ✓ ✓ Max Wheels/Sliders 5 ATSAMD21G16A 48 256 32768 0 ✓ ✓ Max Wheels/Sliders 5 MASAMD21117A 64 128 16384 0 ✓ ✓ Data Memory used Ø ATSAMD21117A 64 128 16384 0 ✓ ✓ Data Memory used Ø ATSAMD21117A 64 128 16384 0 ✓ ✓ Data Memory used Ø ATSAMD21117A 64 256		0	Speed	×	1	SAMD2]	•	All		Device Family
Name Variant App/Boot Memory(KB) Data Memory(Bytes) EEPROM(Bytes) QTouch QMatrix Ports A,B ATSAMD21E15A 32 32 4096 0 ✓ <td></td> <td>0/0</td> <td>Vcc</td> <td></td> <td>_</td> <td></td> <td>_</td> <td></td> <td>)</td> <td></td> <td></td> <td>-</td> <td>-</td>		0/0	Vcc		_		_)			-	-
ATSAMD21E15A 32 32 4096 0 V V ATSAMD21E16A 32 64 8192 0 V V ATSAMD21E17A 32 128 16384 0 V V ATSAMD21E18A 32 256 32768 0 V V ATSAMD21G16A 48 64 8192 0 V V ATSAMD21G17A 48 128 16384 0 V V ATSAMD21G17A 48 256 32768 0 V V ATSAMD21J17A 64 128 16384 0 V V ATSAMD21J17A 64 128 16384 0 V V ATSAMD21J17A 64 256 32768 0 V V ATSAMD21J17A 64 256 32768 0 V V ATSAMD21J18A 64 256 0 V ATSAMD21VISA V ATSAMD21VIS		A,B	Ports	QMatrix	QTouch	OM(Bytes)) EEP	a Memory(Bytes)	nory(KB) Da	p./Boot Mem	ariant Ap	Va	lame
ATSAMD21E16A 32 64 8192 0 V V QTouch Library Informatic ATSAMD21E17A 32 128 16384 0 V V QTouch Library Informatic ATSAMD21E18A 32 256 32768 0 V Max Wheels/Sliders 5 ATSAMD21G16A 48 64 8192 0 V Max Channels 16 ATSAMD21G17A 48 128 16384 0 V Data Memory used Data Memory used Data Memory used Max Channels 256 ATSAMD21J18A 64 256 32768 0 V Max Wheels/Sliders 8 Max Channels 256 ATSAMD21J18A 64 256 32768 0 V Max Wheels/Sliders 8 Max Channels 256 Code Memory used 0 V V Max Wheels/Sliders 8 Max Channels 256 Code Memory used 0 V V Max Wheels/Sliders 8 Max Channels 256 Code Memory used 0 V V Max Wheels/Sliders 8 Max Channels 256 Code Memory used 0 V V Matemo			Datasheets	~	~		0	5	40		2 32	32	TSAMD21E15A
ATSAMD21E17A 32 128 16384 0 V OTouch ATSAMD21E18A 32 256 32768 0 V Max Wheels/Sliders 5 ATSAMD21G16A 48 64 8192 0 V V Max Wheels/Sliders 5 ATSAMD21G17A 48 128 16384 0 V V Max Channels 16 ATSAMD21G18A 48 256 32768 0 V V Data Memory used Data Memory used Data Memory used Max X ATSAMD21J18A 64 256 32768 0 V V Max Wheels/Sliders 8 ATSAMD21J18A 64 256 32768 0 V V Max Wheels/Sliders 8 ATSAMD21J18A 64 256 32768 0 V V Max Channels 256 Code Memory used 0 V V Max Channels 256 Code Memory used Data Memory used Data Memory used OTouch Ubrary Help Atmel Studio Supported To V V V	on	ry Information	OTouch Librar	~	~		0	2	81		2 64	32	TSAMD21E16A
ATSAMD21E18A 32 256 32768 0 V V Max Wheels/Sliders 5 ATSAMD21G16A 48 64 8192 0 V V Max Wheels/Sliders 5 ATSAMD21G17A 48 128 16384 0 V V Max Wheels/Sliders 5 ATSAMD21G17A 48 256 32768 0 V V Data Memory used Data Memory used <td></td> <td>ry mornator</td> <td>OTouch</td> <td>~</td> <td>~</td> <td></td> <td>0</td> <td>34</td> <td>16</td> <td>3</td> <td>2 128</td> <td>32</td> <td>TSAMD21E17A</td>		ry mornator	OTouch	~	~		0	34	16	3	2 128	32	TSAMD21E17A
ATSAMD21G16A 48 64 8192 0 V V ATSAMD21G17A 48 128 16384 0 V V ATSAMD21G17A 48 256 32768 0 V V ATSAMD21117A 64 64 8192 0 V V ATSAMD21117A 64 128 16384 0 V V ATSAMD21117A 64 128 16384 0 V V ATSAMD21117A 64 128 16384 0 V V ATSAMD21117A 64 256 32768 0 V V ATSAMD21118A 64 256 32768 0 V V ATSAMD21118A 64 256 32768 0 V V ATSAMD21118A 64 256 32768 0 V V Max Channels 256 Code Memory used Data Memory used Data Memory used Data Memory used Data Memory used V OTouch Library Help Atmel Studio Supported To X XPRO-EDBG J-Link Atm-Pro J-Link Ultra		5	Max Wheels/Sliders	~	~		0	8	32	5	2 256	32	TSAMD21E18A
ATSAMD21G17A 48 128 16384 0 V Code Memory used ATSAMD21G18A 48 256 32768 0 V Data Memory used ATSAMD21J16A 64 64 8192 0 V Data Memory used ATSAMD21J17A 64 128 16384 0 V V ATSAMD21J17A 64 128 16384 0 V V ATSAMD21J18A 64 256 32768 0 V V Max Channels 256 Code Memory used Otata Memory		16	Max Channels	~	~		0	2	81		3 64	48	TSAMD21G16A
ATSAMD21G18A 48 256 32768 0 V Data Memory used ATSAMD21J18A 64 64 8192 0 V Data Memory used ATSAMD21J18A 64 256 32768 0 V Max Wheels/Sliders 8 Max Channels 256 Code Memory used Data Memory used	2%		Code Memory used	•			0	34	16	3	3 128	48	TSAMD21G17A
ATSAMD21116A 64 64 8192 0 V V Data Meniory Osed Max Wheels/Sliders 8 Max Channels 256 Code Memory used Data Memory Used Da			Data Memory used		¥		0	8	32)	3 256	48	TSAMD21G18A
ATSAMD21118A 64 256 32768 0 ATSAMD21118A 64 256 32768 0 Max Wheels/Sliders 8 Max Channels 256 Code Memory used Data Memory used OTouch Library Help Atmel Studio Supported To X XPRO-EDBG J-Link ARM-Pro J-Link ARM-Pro			OMatrix		* 		0	2	81		4 64	64	ISAMD21J16A
Max Channels 256 Code Memory used Data Memory used <u>OTouch Library Help</u> Atmel Studio Supported To X XPRO-EDBG J-Link ARM-Pro J-Link ARM-Pro J-Link Ultra		8	Max Wheels/Sliders	ž			0		16		1 109	64	
Code Memory used Data Memory used OTouch Library Help Atmel Studio Supported To XPRO-EDBG J-Link ARM-Pro J-Link ARM-Pro J-Link Ultra		256	Max Channels		•		0	8	32)	4 250	64	ISAMD21J18A
Data Memory used <u>OTouch Library Help</u> Atmel Studio Supported To <u>X XPRO-EDBG</u> <i>J_Link</i> ARM-Pro J_Link Ultra	2.00		Code Memory used										
QTouch Library Help Atmel Studio Supported To X PRO-EDBG J-Link J-Link ARM-Pro		·	Data Mamony used										
Atmel Studio Supported To XPRO-EDBG J-Link J-Link ARM-Pro J-Link Ultra	///		OTouch Library Help										
XPRO-EDBG J-Link J-Link ARM-Pro J-Link Ultra	ools	upported Too	Atmel Studio Su										
J-Link J-Link ARM-Pro J-Link Ultra			X XPRO-EDBG										
J-Link ARM-Pro			J-Link										
J-Link Oltra			L Link ARM Pro										
a <u>J-Link Ultra</u>			J-LINK ARWI-FTO										
			J-Link Ultra										
			# 044405										



Self capacitance method uses a single sense electrode, denoted by a Y line.

Self capacitance touch button sensor is formed using a single Y line channel, while a touch rotor or slider sensor can be formed using three Y line channels.



The Y-line numbering in the QT1 Xplained Pro User Guide does not correspond to the Y-line numbering in the SAM D21 PTC module as the QT1 Xplained Pro is a generic touch board (not dedicated to the SAM D21).

 Flip the SAM D21 Xplained Pro and QT1 Xplained Pro boards and determine which SAM D21 I/O pins that are connected to BUTTON 1



Look at next page to determine the right Y-line from QT1 Xplained PRO extension board.





Y-line for BUTTON 1 can be retrieved from the Atmel QT1 Xplained Pro User Guide (http://www.atmel.com/Images/Atmel-42193-QT1-Xplained-Pro_User-Guide.pdf):

Pin on EXT	Function	Description
1	ID	Communication line to ID chip.
2	GND	Ground
3	Y_1	Y-line 1 for Slider
4	Y_2	Y-line 2 for Slider
5	Y_3	Y-line 3 for Slider
6	Y_4	Y-line 4 for Wheel
7	LED_0	Slider, LED 0 (Yellow)
8	LED_1	Slider, LED 1 (Yellow)
9	Y_5	Y-line 5 for Wheel
10	Y_6	Y-line 6 for Wheel
11	Not Connected	
12	Not Connected	
13	Not Connected	
14	Not Connected	
15	LED_2	Slider, LED 2 (Yellow)
16	Y_7	Y-line 7 for Button 2
17	LED 3	Slider, LED 3 (Yellow)
18	Y_8	Y-line 8 for Button 1
19	GND	Ground
20	VCC	Target supply voltage

Table 3-1. QT1 Xplained Pro SC Extension Header 1



• Use the Assign X-Y Line and Gain to the Channel page from QTouch Project Builder to select the correct Y Line:

	Y Line	QT1 Xplained PRO Y Line	SAM D21 Xplained PRO Y Line – I/O
	Y-line	Y_8	Y[5] – PA07
🐥 QT	ouch Project Builder	rate i cogle la Bodiger la sel	
Assi	gn Sensing pin for SAMD20 Por	t Pin - Page 5 of 7	
	Senso	r Channel Sensir	ig Pin Gain
		Suttonu U Y5-PA7	
	Gain is applied	on a per-channel basis to allow a sca	ing-up of the touch sensitivity on contact.
		Reset	
	Help	l	< Back Next > Finish Cancel



We also need to select on this page the Gain for each channel.



TIPS

Gain setting is applied on a per-channel basis to allow superior sensitivity upon contact. Recommended gain settings depend on the sensor design and touch panel thickness.

The increase in the gain value is usually required by a thick front panel but also depends on the sensor design (size of the sensor and coupling between X and Y for mutual capacitance sensors).

Thicker front panels (3mm and more) can result in reduced sensitivity upon touch. Likewise, smaller size sensors or sensors surrounded by ground layer or sensors with ground on the bottom side typically require an increased gain setting.

 In this application, we need to increase the sensor sensitivity to get the highest proximity effect possible

A QTouch Project Builder	Colony, realize 1 are	and the design	and a sector of specific	and the second se		
Assign Sensing pin for SAMD	Assign Sensing pin for SAMD20 Port Pin - Page 5 of 7					
	Sensor	Channel	Sensing Pin	Gain		
	Button0	0	Y5-PA7 ▼			
Gain is	s applied on a per-cha	annel basis to a	Illow a scaling-up of t Reset	he touch sensitivity on conta	ict.	
Help			< Back	Next > Finish	Cancel	

As a consequence, set Gain to 32 and click "Next".



• Check the "Enable QDebug Interface" checkbox, which allows Live streaming of Touch data with QTouch Analyzer

i INFO	QDebug is the Touch data protocol used by QTouch Analyzer to communicate with the SAM D21 through the Atmel D ata G ateway Interface (DGI).
i info	The Atmel Embedded Debugger (EDBG) offers a Data Gateway Interface (DGI) for streaming data to a host PC. This is meant as an aid in debugging and demonstration of features in the application running on the target device. DGI consists of multiple interfaces for data streaming. The supported interfaces are SPI Interface, USART Interface, TWI Interface, and GPIO Interface.
-	

- **TIPS** SAM D21 XPRO Kit Data Gateway Interface supports SPI or I²C.
 - Assign the right port(s) and pins to the SPI interface, these can be found in the SAM D21 Xplained Pro User Guide on the Advanced Options page (http://www.atmel.com/Images/Atmel-42220-SAMD21-Xplained-Pro_User-Guide.pdf):

Table 3-2. DGI Interface Connections when Using SPI

Pin on SAM D21	Function
PB31	SERCOM5 PAD[1] SPI SS (Slave select) (SAM D21 is Master)
PB16	SERCOM5 PAD[0] SPI MISO (Master In, Slave Out)
PB22	SERCOM5 PAD[2] SPI MOSI (Master Out, Slave In)
PB23	SERCOM5 PAD[3] SPI SCK (Clock Out)

Advanced Options - Page (5 of 7		
Enable QDebug Interface (A Select Debug Interface)	llows Live str	eaming of Touch	data to QTouch Analyzer)
Name	Port	Pin]
SPI_BB_SS	в 👻	31 🔹	
SPI_BB_SCK	В •	23 🔹	
SPI_BB_MOSI	В •	22 🔹	
SPI_BB_MISO	B •	16 🔹	
Power Analyzer Disal	ble - (Ti	his option is avail	able only for selected devices.)



The following table extracted from the QTouch Library PTC User Guide (http://www.atmel.com/Images/Atmel-42195-Qtouch-Library-Peripheral-Touch-Controller_User-Guide.pdf), indicates the expected **Resting Signal** value (also called **Reference**) for a given combination of gain setting and filter level setting.

Expected Resting Signal Value for FILTER LEVEL and GAIN Combination	GAIN_1=0	GAIN_2=1	GAIN_4=2	GAIN_8=3	GAIN_16=4	GAIN_32=5
FILER_LEVEL_1 = 0	512	512	512	512	512	512
FILER_LEVEL_2 = 1	512	1024	1024	1024	1024	1024
$FILER_LEVEL_4 = 2$	512	1024	2048	2048	2048	2048
FILER_LEVEL_8 = 3	512	1024	2048	4096	4096	4096
FILER_LEVEL_16 = 4	512	1024	2048	4096	8192	8192
FILER LEVEL 32 = 5	512	1024	2048	4096	8192	16384
$FILER_LEVEL_64 = 6$	512	1024	2048	4096	8192	16384

By increasing both the Gain and the Filter Level, we will increase the Reference value and so scale up the Touch Delta value, which will allow improving the proximity effect.



(Touch) Delta represents the difference between the Signal and the Reference.

• On the on the Advanced Options page, update Filter Level to 64:

A QTouch Project Builder		X
Advanced Options - Page	5 of 7	
Enable QDebug Interface (A	Illows Live streaming of Touch data to QTouch Analyzer)	
Select Debug Interface	SW implemented SPI 🔹	
Name	Port Pin	
SPI_BB_SS	B • 31 •	
SPI_BB_SCK	B v 23 v	
SPI_BB_MOSI	B • 22 •	
SPI_BB_MISO	B • 16 •	
Tuning Parameters	Enabling noise counter measure sets up the PTC for pre-scaler auto tuning along with frequency hop acquisition mode providing best noise performance. Disabling noise counter measure sets up the PTC for series resistor auto tuning along with frequency none acquisition mode resulting in least power consumption.	
Filter Level 64	The filter Level setting controls the number of samples taken to resolve each acquisition. A higher filter level setting provides improved signal to noise ratio under noisy conditions, while increasing the task likes de task l	
Auto OS Disabl	Auto oversample controls the automatic oversampling of sensor channels when unstable signals are detected with the default setting of 'Filter level'. Enabling Auto oversample results in 'Filter level' x 'Auto Oversample' number of samples taken on the corresponding sensor channel when an unstable signal is observed. In a case where 'Filter level' is set to 4 and 'Auto Oversample' is set to 8, 4 oversamples are taken with stable signal values and 16 oversamples are taken when unstable signal is detected.	
Advanced Tuning Param	eters	
Help	< Back Next > Finish Ca	ncel



• Review the Summary page which provides a summary of the settings provided by the previous set of wizard pages. Then, click on Finish and wait for your Touch project to be generated.

QTouch Project Builder	- T + F	r.1033,	12.10		and the local division of	
Summary - Page 7 of 7				im U prida tj US		Contraction of the second seco
Device Information						*
Device Name		ATSAI	MD21J18A			
Device Variant		64				
Technology		QTouc	h			
Sensor Information						
Number Of Buttons		1				
Number Of Wheels		0				
Number Of Sliders		0				
Channel Information						
Total Channels Consumed		1				
Pin Configuration						
Available Ports		A,B				
Total Pins Used		6				
Name of the sensor	Channel Number	Sensin	gPair		PortPin	
Button0	1	¥5			PA7	
Debug Interface- SAMD20SpiBi	tBanged		Port Pin			
SPI_BB_SS			PB31			
SPI_BB_SCK			PB23			
SPI_BB_MOSI			PB22			
SPI_BB_MISO			PB16			
Power Analyzer			Disable			
Memory type	Total	Used			Free	
Data Memory	32768	277			32491	
Code Memory	262144	9664			252480	
Library Information						
Tool Chain Name		GCC				
Default Library		libsamd21_qtouc	h_gcc.a			
Other Options						
Frequency Mode		NONE				
Filter Level		Six				
Auto OS		Zero				
Auto Tuning		RSEL				
PreScalar		One				
Series Resistor		Zero				
Frequency Hops		One,Two,Three				
						-
Help			< Back	Next >	Finish	Cancel





You have created a Touch Project with a Self Capacitance technology Touch Button.

🖗 Hands-on Assignment - AtmelStudio	
Eile Edit View VAssistX ASF Project Build Debug Iools Window Help	
[1] ・ 田 田 ・ 🍅 👦 🥔 茶 🏎 (ウ - (マ - (コ - 12) + 13) 図 🔍 (図 OK ク) Mi Debug 🔹 (図 get_selfcap 👘) 🔍 😁 🎒 🔬 🖉	<u>▲ □ • ↓]</u> 詳 詳 <u>□</u> 일 □ <i>₽</i> ♀ ♀ ♀ ♀ ↓ ↓
] 😧 🗁 🎜 🏠 😚 😘 🔬 🖆 🚽 🕅 💷 今 川 🕨 🐼 💷 💭 🖬 T Hex 🚭 + 🚽 🕄 🖓 💷 🖾 🚽 💮 🗰 ATSAMD2LIBA 🁔	No Tool 🖕
main.c 🗙 SAM D21 Xplained Pro - 0205 QTouch Start Page Start Page 🗸 🗸	Solution Explorer 🔹 후 부 🗙
🔶 main.c 🔹 🐨	
<pre>maint</pre>	 Solution 'Hands-on Assignment' (1 project) Hands-on Assignment (1 project) Hands-on Assignment. Dependencies Ubraries Thands-on Assignment.qtdgn main.c
touch time.time to measure touch = 1;	
100 %	
Output	- ↓ ×
Show output from: XDK Packaging 🔹 🖓 🖓 🖓 🦃 🧏	
	b.
Ready	Col 57 Ch 57 INS

3.2 Conclusion

In the first assignment, you have learnt:

- How to create a Touch Project using QTouch Composer Project Builder
- How easy it is to use QTouch Composer to design a Touch project on the Atmel SAM D21 MCUs



4. Assignment 2: Use QTouch Analyzer to Test the Project

4.1 Connect the Kit to QTouch Analyzer

- **TO DO** Build the QTouch project and Program the SAM D21 Xplained Pro.
 - Connect the QT1 Xplained Pro Self Capacitance board to the SAM D21 Xplained Pro board:



- Connect the SAM D21 Xplained Pro board to your PC using DEBUG USB connector
- Build the solution and ensure you get no errors:



•Þ 💵

Program the application by clicking on the Start Without Debugging icon:



• You will be asked the first time to select your debug tool:

Hands-on Assignn	ent 🗙 SAM D21 Xplained Pro - 0205 QTouch Start Page Start Page main.c
Build	Configuration: N/A Platform: N/A
Build Events	
Toolchain	Selected debugger/programmer
Device	▼ Interface: ▼
Tool	Programming settings Atmel Studio
Advanced	Erase entire chip 🔻
	Debug settings
	Continue

• Select your EDBG and SWD (Serial Wire Debug) as Interface:

Selected debugger/programmer		
EDBG • ATML2130021800001447 👻	Interface:	SWD 🗸

• Set SWD clock to 8MHz to speed up programming:

	~ S	WD Clock
		8.00MHz
	г	he clock frequency should not exceed target CPU speed * 10.
•	Click aga	ain on the Start Without Debugging icon:
	INFO	You may be asked to upgrade your EDBG firmware. If so, click on Upgrade:
		Firmware Upgrade
		XPRO-EDBG firmware upgrade started
		On Tool On Disk Firmware Version 1.12 1.18
		Waiting for bootloader usb enumeration
	WARNIN	G Upgrade operation may take a few minutes, please <u>wait</u> for the operation to complete.
	RESULT	Your QTouch project is now updated and programmed on the SAM D21 Xplained Pro.





- Press RESET button on the SAM D21 Xplained Pro
- Launch QTouch Analyzer by clicking on the following icon:



QTouch Analyzer is also accessible through the Tools menu:

Tool	s Window Help		
>	Command Prompt		sensor_state 🔹
\$	Device Programming	Ctrl+Shift+P	🔯 🔄 🚽 🕮 🚟 👗 🚽 📟 A
7	Add target		
	QTouchComposer	•	CT QTouch StartPage
	Code Snippets Manager	Ctrl+K, Ctrl+B	QTouch Analyzer

• A kit named QDEBUG_DGI will be listed in the combo box. Select the kit and click on Connect:

QTouch Analyze	r × SAM	D21 Xplair	ned Pro - I	0205 Q
QDEBUG_DGI -	00205	Con	nect 🕨	Start Readi
Virtual Kit				
	No Kita	Connecte	d	

• DGI settings dialog box will be displayed as shown in the figure, click OK:

秦 DGI Settings	
Interface	▼ Iq2
Character Length	EightBits 👻
SPI Mode	RisingEdgeRead_SCKLow +
	Ok Cancel



	QTouch Analyzer × SAM D21 Xplained Pro - 0205 QTouch
	QDEBUG_DGI - 00205
	Virtual Kit Trac
Click on S	art Reading: QDEBUG_DGI - 00205 🔹 🔁 ReConnect 🕨 Start Reading
RESULT	QTouch Analyzer is now connected to your kit and displays both the sensor signal and reference values.
🚺 TIPS	Click on Stop Reading to prevent QTouch Analyzer from collecting touch data over a long period as this might slow down the computer.





4.2 Test the Proximity Effect of the QTouch Application



• Click on checkbox "Check to enable user scale" and set Y scale from 0 to 50:



- Unselect Signals and References and Deltas in the Trace View
- Select Deltas and Detect Threshold in the Trace View



• Approach BUTTON 1 on your kit, which is displayed as Button0 on QTouch Analyzer and check that a detect touch can occur BEFORE touching the sensor (i.e. the sensor Touch Delta value becomes higher than its Detect Threshold).



4.3 Conclusion

In this assignment, you have learnt:

• How to use QTouch Analyzer to test a Touch application



5. Assignment 3: Implement a PWM Signal using the SAM D21 Timer/Counters for Control Application Peripheral (TCC)

In this assignment, a Pulse-Width Modulation (PWM) signal will be generated using the SAM D21 Timer/Counters for Control Application (TCC) to light the LED8 of QT1 Xplained Pro extension board:



5.1 Introduction to Timer/Counters for Control Application (TCC)

The TCC will be used in Single-Slope PWM mode to generate the required PWM signal and output it on a dedicated Waveform Output pin (WO[x]).

For single-slope PWM generation mode, the period time is controlled by the Period register (PER), while the Compare/Capture registers (CCx) control the duty cycle of the waveform.

When up-counting (default configuration), WO[x] pin is:

- Set at start or compare match between the counter (COUNT) and period (PER) registers' values
- Cleared on compare match between the counter (COUNT) and Compare/Capture (CCx) registers' values





Determine which Waveform Output pin to use.

• By flipping the boards, you can check that the LED8 of the QT1 Xplained Pro extension board is connected to the SAM D21 Xplained Pro board pin PB12:





• By looking now at the SAM D21 product datasheet, you can verify that this is the Waveform Output pin 6 (WO[6]) from TCC0 which is multiplexed with PB12

Table 5-1.	PORT Function	Multiplexing	(Continued)
			(

Pin			I/O Pin	Supply	Туре	A	в					c	D	E	F	G	н
SAMD21E	SAMD21G	SAMD21J				EIC	REF	ADC	AC	PTC	DAC	SERCOM	SERCOM-ALT	тс/тсс	тсс	сом	AC/GCLK
	19	23	PB10	VDDIO		EXTINT[10]							SERCOM4 /PAD[2]	TC5/WO[0]	TCC0/ WO[4]	12S/ MCK[1]	GCLK_IO[4]
	20	24	PB11	VDDIO		EXTINT[11]							SERCOM4 /PAD[3]	TC5/WO[1]	TCC0/ WO[5]	12S/ SCK[1}	GCLK_IO[5]
		25	PB12	VDDIO	I ² C	EXTINT[12]				X[12]		SERCOM4/ PAD[0]		TC4/WO[0]	TCC0/ WO[6]	12S/ FS[1]	GCLK_IO[6]

i INFO

SAM D21 Product Datasheet can be retrieved at the following link: http://www.atmel.com/Images/Atmel-42181-SAM-D21_Datasheet.pdf.

WO[6] pin must be used to output the PWM signal directly to LED8 using TCC0.



5.2 TCC Peripheral Initialization Using ASF Quick Start Guide

The drivers in ASF have quick start guides as part of their API documentation.

Quick Start guides show and explain, in a step-by-step process, the code and actions needed to set up and use a driver in one or more use cases.

Once a driver such as TCC is added using the ASF Wizard, its Quick Start Guide becomes available in the ASF Explorer view.



To get started on the ASF API documentation as the Quick Start Guides, please go to: http://asf.atmel.com/docs/latest/get_started.html.



Add TCC Support using ASF Wizard.

Click on the ASF Wizard icon or right click on the Hands-on Assignment project > ASF Wizard

10015	W	naow	<u> </u>	leip	-
- E	80	۹	⊳	⊳ 11	C

- From latest ASF version available, select the following component and add it to the Selected Modules (Add >> button):
 - TCC Timer Counter for Control Applications (driver) polled

Available Modules	Selected Modules						
Extensions: Atmel ASF(3.15.0) Show: All Search for modules	Image:						
	Image:						
Sleep manager (service)	• I QT - Atmel QTouch Library (component)						
Image (controller (component)	ibsamd21_qtouch_gcc.a (component)						
Image: Standard serial I/O (stdio) (driver)							
SYSTEM - Clock Management (driver)							
🛛 🔟 SYSTEM - I/O Pin Multiplexer (driver)							
SYSTEM - Interrupt Driver (driver)							
TC - Timer Counter (driver) callback							
TCC - Timer Counter for Control Applications (driver) polled 🔻							
Imit test framework (driver)							
USART - Serial interface (service)							
Device (service) cdc 💌							
▷ III USB Host (service) cdc ▼							

i info

In this application, TCC polled driver version can be used as no interrupt(s) management is required to generate the PWM signal on the waveform output WO[x] pin.

Click on Apply button

Click on Solution Explorer tab and check that the TCC driver has been successfully added as • shown below:

Solution Explorer	▼ ₽>
🖷 🏠	
Output Files	
b Ibraries	
🔺 🗁 src	
a 🙆 ASF	L L L
common	
common2	
🔺 🧰 sam0	
a 🙆 drivers	
port	
D intc	
b system	
🔺 间 tcc	
p quick_start	
b i quick_start_dma	
d tcc.c	
🖻 tcc.h	
b utils	
b initial party	
b is config	
D QTouch	
🛗 asf.h	
b 🔯 Hands-on Assignment.qtdgn	
🖸 main.c	
🔍 ASF Explorer 🔎 VA View 🛭 😤 VA Outline 🗔 Solution Explorer	
	<u>/</u>

RESULT TCC driver is added to the project.



We will now implement the TCC initialization by using an example code available in the TCC Quick Start Guide.



TCC Quick Start Guide can be accessed by selecting the ASF Explorer tab then double clicking on TCC > Quick Start Guide:

ASF Explorer	•	д	×			
😂 🗄 📾 👒 🧇						
🔺 븚 Hands-on Assignment						
Image:						
Ibsamd21_qtouch_gcc.a						
Image:						
SYSTEM - Core System Driver						
TCC - Timer Counter for Control Applications (Polled APIs)						
API Documentation						
Search Quick Start Guide						
tcc.h						
PORT - GPIO Pin Control						
SAM D21 compiler driver						
SYSTEM - Core System Driver						
🔍 ASF Explorer 🥌 VA View 🛛 👘 VA Outline 🛛 💐 Solution Explorer						

WARNING The ASF API documentation is exclusively available on the web.

Several Quick Start Guides are available depending on the TCC driver characteristics: polled, callback, with DMA support...

i info

The TCC Quick Start Guide which relates to the ASF TCC driver in polled mode is called "*Quick Start Guide for TCC – Basic*":

The following code that will be implemented can be found in the sub section of the Quick Start Guide called Workflow.

That sub section provides a detailed step by step guide to initialize the TCC in a "polled mode" use:

Workflow

- - - tcc enable(&tcc instance);
 - **TO DO** TCC Initialization Implementation.
- In the main.c file, create a new function above main function (in red):

```
static void configure_tcc(void)
{
    int main (void)
    {
        system_init();
    }
}
```

• Create a TCC module instance:

```
struct tcc_module tcc_instance;
static void configure_tcc(void)
{
}
```

• In the configure_tcc function, create a TCC module configuration instance and initialize it with the module's default values using tcc_get_config_defaults function:



```
i info
```

TCC0 definition corresponds to the TCC0 peripheral base address.

The TCC0 peripheral has four different compare/capture channels. Each channel has a dedicated register (CC0 to CC3).

TCC0 has also an output matrix which can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types.



By default, the Output Matrix Channel Pin Routing Configuration is the following:

Compare and Capture registers	Waveform Output pins
CC0	WO[0], WO[4]
CC1	WO[1], WO[5]
CC2	WO[2], WO[6]
CC3	WO[3], WO[7]

📜 TIPS

As a consequence, the Channel 2 Compare and Capture register (CC2) must be used to output the PWM signal on WO[6].



- Update default TCC settings to configure:
 - The Period register (PER) value to get a resolution of 16-bit for the PWM waveform
 - The waveform generation mode to Single-Slope PWM
 - The Channel 2 Compare and Capture register (CC2) value to half the period register value to get a duty cycle of 50%

INFO

The following equation calculates the exact resolution for a single-slope PWM waveform:

$$Resolution = \frac{\log (PER + 1)}{\log (2)}$$

To get a 16-bit resolution, PER must be equal to 65535 \Leftrightarrow 0xFFFF.

- Update default TCC settings to:
 - Enable the PWM waveform output pin WO[6]
 - Specify the WO[6] pin output which is PIN_PB12F_TCC0_WO6
 - Specify the WO[6] peripheral multiplexing port which is MUX_PB12F_TCC0_WO6

```
config_tcc.pins.enable_wave_out_pin[6] = true;
config_tcc.pins.wave_out_pin[6] = PIN_PB12F_TCC0_WO6;
config_tcc.pins.wave_out_pin_mux[6] = MUX_PB12F_TCC0_WO6;
```

• Configure the TCC module with the desired settings:

```
tcc_init(&tcc_instance, TCC0, &config_tcc);
```

• Enable the TCC module to start the timer and begin PWM signal generation:

tcc enable(&tcc instance);



RESULT TCC0 initialization is completed.

```
struct tcc_module tcc_instance;
static void configure_tcc(void)
{
    struct tcc_config config_tcc;
    tcc_get_config_defaults(&config_tcc, TCC0);
    config_tcc.counter.period = 0xFFFF;
    config_tcc.compare.wave_generation =
 TCC_WAVE_GENERATION_SINGLE_SLOPE_PWM;
    config_tcc.compare.match[2] = 0x7FFF;
    config_tcc.pins.enable_wave_out_pin[6] = true;
    config_tcc.pins.wave_out_pin[6] = pIN_PB12F_TCC0_WO6;
    config_tcc.pins.wave_out_pin_mux[6] = MUX_PB12F_TCC0_WO6;
    tcc_init(&tcc_instance, TCC0, &config_tcc);
    tcc_enable(&tcc_instance);
}
```



TO DO Test PWM Signal Implementation.

• Call the configure tcc function in your main function:

```
int main (void)
{
    system_init();
    configure_tcc();
```

i info

In STANDBY sleep mode, all clocks and functions are stopped expect those selected to continue running.

D 🚺

 Disable the entry in STANDBY sleep mode by commenting system_sleep function so that the TCC0 clock and so the waveform output WO[6] are not disabled



- Build the solution and ensure you get no errors:
- Program the application by clicking on the Start Without Debugging icon:
- Press RESET button on the SAM D21 Xplained Pro





5.3 Conclusion

In this assignment, you have learnt:

- How to configure the TCC to generate a PWM signal
- How to configure the TCC to output a PWM signal on a dedicated Waveform Output pin WO[x]



6. Assignment 4: Use QTouch Button to Control LED Brightness

In this assignment, the PWM duty cycle of waveform output WO[6] signal will be configured to control the LED8 brightness in relation to the BUTTON 1 Touch Delta value.

Here are the steps to follow in order to complete this assignment:

- Use QTouch Library API to read BUTTON 1 Touch Delta value
- Update Waveform Output WO[6] duty cycle in relation to BUTTON 1 Touch Delta value

6.1 Reading and using Button Touch Delta Value

We will use the API function called <code>touch_selfcap_sensor_get_delta</code> to retrieve the delta value corresponding to BUTTON 1.

Here is the API function definition for Self Capacitance technology:

Where:

- sensor id is the sensor ID for which delta value is being retrieved
- p delta is the pointer to the delta variable to be updated by the Touch Library



This API function is defined in touch api SAMD.h from QTouch folder.

The ON/OFF state of a button as the slider/rotor position, are touch status parameters.

These parameters must be read by the application only after the measurement done touch flag is set.

This flag is part of the touch measure data structure called touch measure data t.

i info

A pointer to this data structure is already declared in touch.c and initialized by the QTouch library:

```
/* ! Self capacitance method measured data pointer. */
touch_measure_data_t *p_selfcap_measure_data = NULL;
```

So, the following code can be used to quickly implement a read of the Touch Delta value of a specific sensor:

```
/* ! Start Touch Sensor Measurement. */
touch_sensors_measure();
/* Update touch status once measurement complete flag is set. */
if ((p_selfcap_measure_data->measurement_done_touch == 1u))
{
    p_selfcap_measure_data->measurement_done_touch = 0;
    touch_selfcap_sensor_get_delta(sensor_id, p_delta);
}
```





• Create in main function a touch delta t instance:



• Under touch_sensors_measure function, add the following code to implement the read of the Touch Delta value for BUTTON 1:

```
/* ! Start Touch Sensor Measurement. */
touch_sensors_measure();
/* Update touch status once measurement complete flag is set. */
if ((p_selfcap_measure_data->measurement_done_touch == 1u))
{
    p_selfcap_measure_data->measurement_done_touch = 0;
    touch_selfcap_sensor_get_delta(0, &button1_delta);
}
```



sensor id = 0 as we only have one sensor (BUTTON 1).

RESULT The application is now able to get BUTTON 1 Touch Delta value.



6.2 Using PWM to Control LED Brightness

To control the brightness of an LED, you have to vary the power which is sent to the LED. The more power the LED receives, the brighter it is.

A PWM signal provides the ability to 'simulate' varying levels of power by oscillating the output from the microcontroller in relation to its duty-cycle.



The PWM duty-cycle refers to the total amount of time a pulse is 'on' over the duration of the cycle.

So, LED8 brightness will be controlled by modifying the PWM duty cycle of the Waveform Output pin WO[6].

To do that, the TCC0 period value (PER register) must be modified and aligned to the maximum measured Touch Delta value. This will allow writing directly the measured Touch Delta value in the Channel 2 Compare and Capture register (CC2).

Any Touch Delta value change on the PTC will be then directly seen as a duty-cycle update on the PWM signal:







Get Maximum Measured Touch Delta value.

• Launch QTouch Analyzer by clicking on the following icon:



- Press BUTTON 1 on your kit and check using the Tabular View the maximum Touch Delta value
- **i** INFO It may happen that the delta value goes slightly below 0 (negative values). This behavior has to be considered to correctly implement the duty cycle update later.
- **RESULT** The maximum measured Touch Delta value will vary depending on how the button is pressed but should be in the range of the one measured below:

Tał	oular View							
Bu	Buttons							
	Button Id	Name	State	Delta	Delta RMS	Channel Id	Signal	
	0	Button0	ON	11387	0	0	27286	

We will now implement the update of the PWM duty cycle in relation to BUTTON 1 Touch Delta value.

The ASF API function called tcc_set_compare_value will be used to update the TCC0 Channel 2 Compare and Capture value.

Here is the ASF API function definition:

Where:

- module inst is the pointer to the software module instance
- channel index is the index of the compare channel to write to
- compare is the new compare value to set





 In the configure_tcc function, update PER and CC2 registers to the maximum measured Touch Delta value:

static void configure tcc(void) { struct tcc config config tcc; tcc get config defaults(&config tcc, CONF PWM MODULE); config tcc.counter.period = 12000; config tcc.compare.wave generation = TCC WAVE GENERATION SINGLE SLOPE PWM; config tcc.compare.match[2] = 12000;



The Maximum Touch Delta value is rounded up as different maximum values may be observed depending on the user.

• In the main function, update the following code to have the LED8 brightness controlled by the measured Touch Delta value:





channel_index = 2 as CC2 register is used to generate the PWM signal.



TIPS

Out-of-bounds values are handled.

- Build the solution and ensure you get no errors:
- Program the application by clicking on the Start Without Debugging icon:
- Press RESET button on the SAM D21 Xplained Pro

RESULT LED8 brightness varies when approaching the finger to BUTTON 1 but the proximity effect is not in line with the application expectation as LED8 is ON when no touch is detected and its brightness decreases when approaching the finger.



The application needs to have LED8 switched off when there is no touch or proximity touch.

On the QT1 Xplained extension board, LED8 is turned off when PIN_PB12 is set:



So, the solution is to invert the PWM waveform output signal thanks to the waveform output polarity feature of the SAM D21 TCC (POLx bit):

POLx	Waveform Generation Output Update						
	Set	Clear					
0	Timer/counter matches TOP	Timer/counter matches CCx					
1	Timer/counter matches CCx	Timer/counter matches TOP					

POLx = 0 by default where x is the TCC Channel Number.

If POLx = '1', waveform output pin becomes:

- Cleared at start or on compare match between the counter (COUNT) and period (PER) registers' values (TOP)
- Set on compare match between the counter (COUNT) and Compare/Capture (CCx) registers' values



IINFO



Invert PWM signal by modifying the Waveform Output Polarity.

- In the configure_tcc function, update TCC settings to:
 - Have LED8 switched off when there is no touch or proximity touch
 - Invert the Waveform Output WO[6] polarity (TCC0 Channel 2)



- Build the solution and ensure you get no errors:
- Program the application by clicking on the Start Without Debugging icon:



Press RESET button on the SAM D21 Xplained Pro

RESULT LED8 brightness finally increases when approaching the finger to BUTTON 1.

6.3 Conclusion

In this assignment, you have learnt:

- How to use the QTouch Library API to read a Button Touch Delta value
- How to update a TCC Waveform Output PWM duty cycle in relation to a Button Touch Delta value



7. Conclusion

This hands-on demonstrated the ease of use of the different Atmel QTouch technology tools as well as the Atmel Software Framework APIs and their full integration in the Atmel Studio IDE.

The following topics have been covered:

- Touch project creation using Atmel QTouch Composer
- Touch project analysis using Atmel QTouch Analyzer
- Touch project implementation using the Atmel QTouch Library and the Atmel Software Framework APIs

You have seen how Atmel Studio, the Atmel Software Framework API and the Atmel QTouch tools make it easy to add capacitive touch sensing to your project.



8. Revision History

Doc. Rev.	Date	Comments
42327A	07/2014	Initial document release.



Atmel Enabling Unlimited Possibilities[®]

Atmel Corporation

1600 Technology Drive San Jose, CA 95110 USA Tel: (+1)(408) 441-0311 Fax: (+1)(408) 487-2600 www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F BEA Tower, Millennium City 5 418 Kwun Tong Road Kwun Tong, Kowloon HONG KONG Tel: (+852) 2245-6100 Fax: (+852) 2722-1369

Atmel Munich GmbH

Business Campus Parkring 4 D-85748 Garching b. Munich GERMANY Tel: (+49) 89-31970-0 Fax: (+49) 89-3194621

Atmel Japan G.K.

16F Shin-Osaki Kangyo Bldg. 1-6-4 Osaki, Shinagawa-ku Tokyo 141-0032 JAPAN Tel: (+81)(3) 6417-0300 Fax: (+81)(3) 6417-0370

© 2014 Atmel Corporation. All rights reserved. / Rev.: 42327A-07/2014

Atmel[®], Atmel logo and combinations thereof, Enabling Unlimited Possibilities[®], QTouch[®], and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL EFOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.