



Компэлфест 2017

4-Я ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ СОВРЕМЕННОЙ ЭЛЕКТРОНИКИ

7 – 10 ФЕВРАЛЯ



## Реализация емкостных сенсоров на базе технологии CapTivate от Texas Instruments

Руководство по выполнению практической работы на отладочном наборе [MSP-CAPT-FR2633](https://www.ti.com/capivate)



## Содержание

Введение .....	3
1. Аппаратная платформа и программное обеспечение	
1.1. Аппаратная платформа.....	4
1.1.1. Плата программатора CAPTIVATE-PGMR.....	4
1.1.2. Плата контроллера CAPTIVATE-FR2633 .....	5
1.1.3. Платы сенсоров .....	6
1.1.4. Плата изолятора .....	8
1.2. Программное обеспечение .....	8
2. Работа с демо-проектом CapTivate	
2.1. Загрузка демонстрационного проекта в Captivate Design Center .....	9
2.2. Генерация исходного кода и проекта для IDE .....	11
2.3. Импорт проекта в CodeComposerStudio .....	12
2.4. Загрузка программы в микроконтроллер .....	16
2.5. Отображение данных сенсора в реальном времени .....	18
2.5.1. Отображение работы сенсоров в рабочем поле проекта .....	19
2.5.2. Отображение параметров в свойствах объекта .....	19
3. Реализация нового проекта емкостной кнопки в режиме SelfCapacitance	
3.1. Аппаратная часть .....	21
3.2. Создание нового проекта в Captivate Design Center .....	22
3.3. Отображение параметров сенсора в реальном времени .....	31
3.4. Настройка параметров измерений .....	34
3.5. Добавление callback-функции для кнопки .....	37
4. Реализация слайдера	
4.1. Добавление слайдера в проект .....	38
4.2. Калибровка слайдера .....	42
4.3. Добавление callback-функции для слайдера.....	43
Заключение .....	44

## Введение

Целью данного руководства является помощь в освоении технологии CapTivate от Texas Instruments, которая предназначена для реализации емкостных сенсоров управления.

Технология CapTivate реализована в микроконтроллерах

[MSP430FR2532](#)/[MSP430FR2533](#)/[MSP430FR2632](#)/[MSP430FR2633](#) и представляет из себя отдельный периферийный модуль в микроконтроллере, а также специальное программное обеспечение по настройке параметров сенсоров.

В данном руководстве описаны три практические работы по реализации и настройке программного обеспечения для емкостных сенсоров, выполненных на базе отладочных плат из отладочного набора [MSP-CAPT-FR2633](#).

Более подробная информация о технологии CapTivate и более продвинутых параметрах настройки сенсоров доступна в [центре знаний CapTivate](#).

# 1. Аппаратная платформа и программное обеспечение

## 1.1. Аппаратная платформа

Практические работы выполняются на базе отладочного набора [MSP-CAPT-FR2633](http://www.ti.com/processors/microprocessors/msp430-captivate). Данный набор состоит из нескольких отдельных функционально законченных плат.

### 1.1.1. Плата программатора CAPTIVATE-PGMR

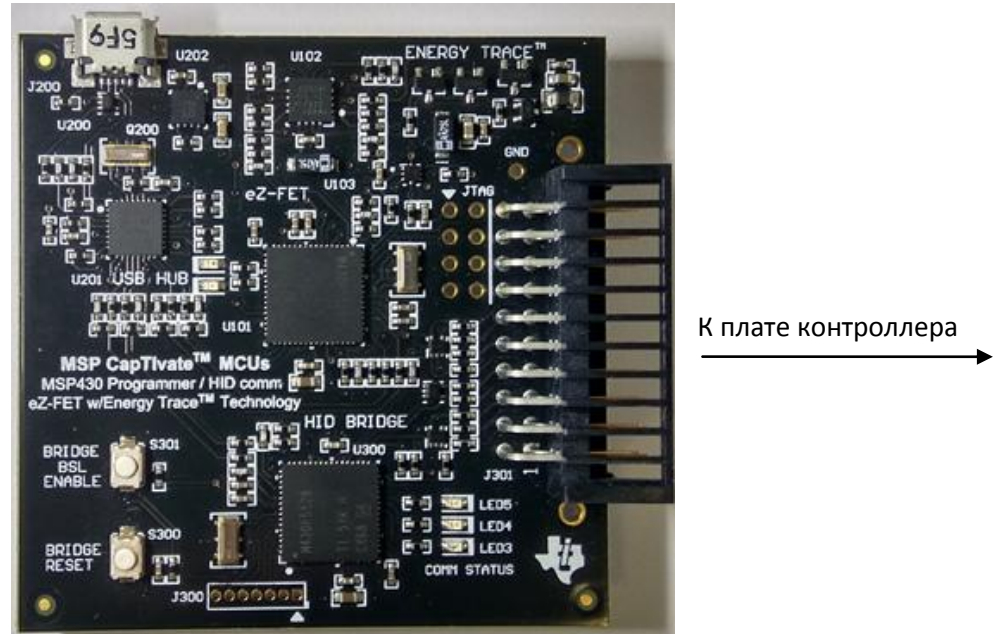


Рис.1 – плата программатора CAPTIVATE-PGMR

Плата CAPTIVATE-PGMR (рис.1) выполняет функции программатора-отладчика и содержит три основных узла:

- 1) Программатор eZFET™ с технологией EnergyTrace™
  - программирование микроконтроллера(МК) по интерфейсу Spy-Bi-Wire
  - может быть использован с любым МК MSP430
  - два отдельных вывода питания +3.3V для целевого МК:
    - от EnergyTrace™
    - от отдельного LDO
- 2) 20-выводный разъем питания/программирования/интерфейса обмена данными
  - программирование МК по интерфейсу Spy-Bi-Wire
  - обмен данными с целевым МК по UART и I2C
  - 2 линии питания +3.3V
  - +5V от USB
- 3) Последовательный мост USB HID
  - Реализует интерфейс между приложением разработки CapTivate™ Design Center и целевым МК
  - Поддерживает UART и I2C
  - HID – не требует драйверов
  - Поддержка скоростей до 250 кБод
  - Простое изменение прошивки моста через BSL

### 1.1.2. Плата контроллера CAPTIVATE-FR2633

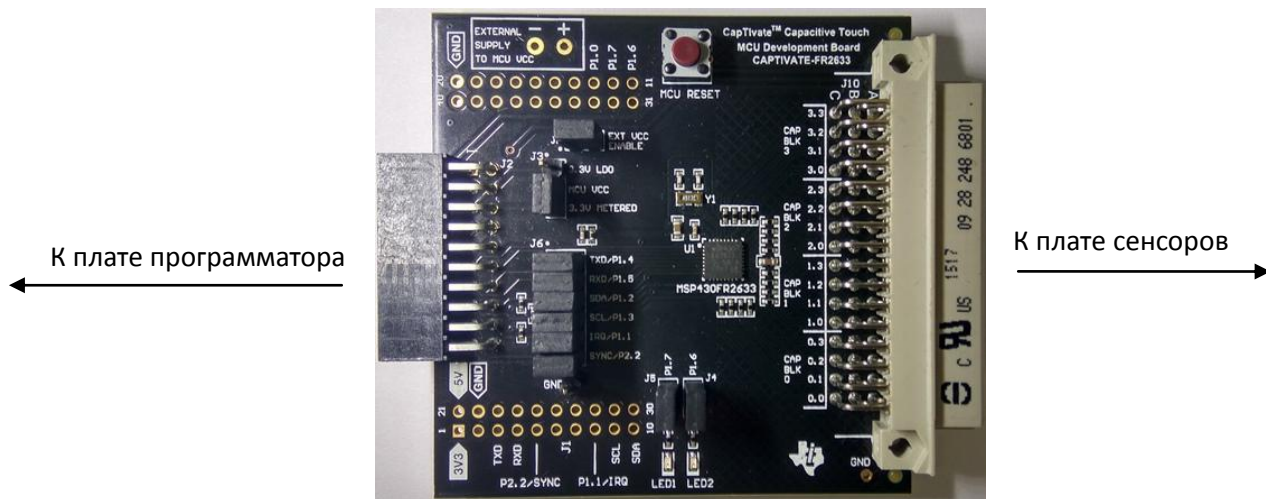


Рис.2 – Плата контроллера CAPTIVATE-FR2633

Плата контроллера (рис.2) содержит целевой микроконтроллер [MSP430FR2633](#) с поддержкой технологии емкостных сенсоров CapTivate. Данная плата устанавливается между платой программатора и одной из трех плат сенсоров.

Для подключения к плате программатора и плате сенсоров на плате установлены 2 разъема. Также на плате реализовано посадочное место для 40-выводного разъема формата BoosterPack для подключения к популярным отладочным платам от Texas Instruments типа [LaunchPad](#). Для реализации автономного питания платы, а также для изоляции от платы программатора реализовано отдельное посадочное место для установки разъема питания.

Плата контроллера CAPTIVATE-FR2633 содержит:

- 1) МК [MSP430FR2633](#) с технологией емкостных сенсоров CapTivate™
  - 15.5кБайт памяти FRAM, 4KB памяти RAM
  - 12кБайт памяти ROM (загрузчик BSL, программная библиотека CapTivate™, HID-интерфейс, библиотека драйверов MSP430, драйвера I2C и UART)
  - 16 выводов CapTivate™
- 2) 48-выводной разъем для подключения к плате сенсоров
  - 16 выводов CapTivate™ I/O
  - 5 GPIO
  - I2C
  - +3.3V LDO
  - +5V (VBUS)
  - Поддержка устройств CapTivate™ следующего поколения с 32 выводами CapTivate™ I/O
- 3) 20-выводный разъем питания/программирования/интерфейса обмена данными
  - Программирование МК по интерфейсу Spy-Bi-Wire
  - Обмен данными с целевым МК по UART и I2C
  - 2 линии питания +3.3V
  - +5V от USB
- 4) Посадочное место под 40-выводной разъем BoosterPack
  - Питание, UART, I2C и 3 GPIO
  - Джампер J3 выбора питания VCC
  - Выбор источника питания от платы программатора:
    - +3.3V EnergyTrace™ для измерений потребления энергии

- +3.3V LDO-регулятор предоставляет до 250mA для целевого МК и платы сенсоров
- Внешнее или батарейное питание

### 1.1.3. Платы сенсоров CAPTIVATE-BSWP, CAPTIVATE-PHONE, CAPTIVATE-PROXIMITY

В отладочный набор [MSP-CAPT-FR2633](#) входят три платы сенсоров, каждая из которых демонстрирует определенный набор преимуществ технологии CapTivate.

Плата **CAPTIVATE-BSWP** (рис. 3) содержит несколько емкостных кнопок, слайдер, колесо и датчик приближения. Все сенсоры на данной плате выполнены для измерений в режиме собственной емкости (self-capacitance).

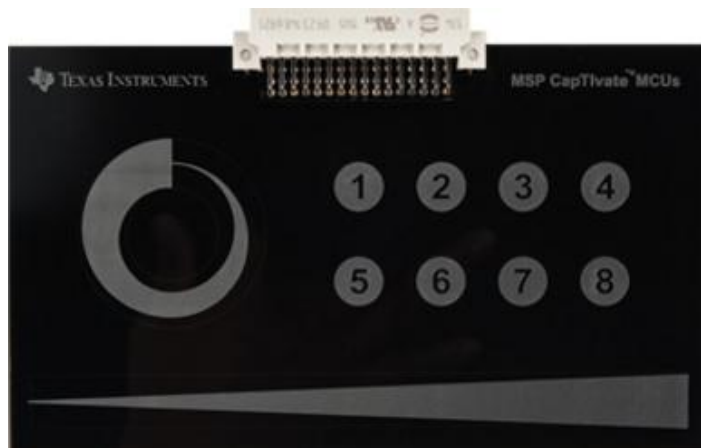


Рис.3 – Плата сенсоров CAPTIVATE-BSWP

Плата CAPTIVATE-BSWP выполняет следующие функции:

1. Демонстрирует принципы разработки низкопотребляющего устройства с батарейным питанием
2. Демонстрирует работу слайдера и колеса в режиме self-capacitance и их разрешение
3. Выступает рекомендуемым примером трассировки печатной платы для слайдера и колеса в режиме self-capacitance
4. Использует все 16 выводов CapTivate™ контроллера [MSP430FR2633](#)
5. Демонстрирует использование функции пробуждения-по-приближению (wake-on-proximity) реализованной в CapTivate™ для снижения потребляемой энергии
6. Демонстрирует использование функции автоматического распределения выводов в среде разработки CapTivate™ Design Center

Плата **CAPTIVATE-PHONE** (рис.4) содержит 17 кнопок, 2 слайдера, колесо и датчик приближения, который также является датчиком защиты от ложных срабатываний. Дополнительно на плате установлен драйвер линейного актуатора [DRV2605L](#) для реализации тактильной обратной связи в виде вибрации.



Рис.4 – Плата сенсоров CAPTIVATE-PHONE

Плата CAPTIVATE-PHONE выполняет следующие функции:

1. Демонстрирует реализацию матричной клавиатуры в режиме измерения взаимной емкости (mutual capacitance) с использованием малого числа выводов МК (7 выводов для 12 кнопок)
2. Выступает рекомендуемым примером трассировки печатной платы для слайдера и колеса в режиме измерения взаимной емкости (mutual capacitance)
3. Демонстрирует одновременное использование режимов измерения собственной и взаимной емкости в одном проекте
4. Демонстрирует использование датчика приближения в качестве датчика защиты от ложных срабатываний при попадании воды или случайном нажатии ладонью
5. Выступает примером реального устройства (панель настольного телефона)

Плата **CAPTIVATE-PROXIMITY** (рис. 5) демонстрирует реализацию панели датчиков приближения с четырьмя электродами и шестью светодиодными индикаторами: 4 для обнаружения приближения и 2 для распознавания жестов.



Рис.5 - Плата сенсоров CAPTIVATE-PROXIMITY

Данная плата выполняет следующие функции:



1. Демонстрирует реализацию управления жестами с повышенной помехоустойчивостью
2. Служит примером трассировки экранирующего электрода CapTivate™ для уменьшения паразитной емкости
3. Демонстрирует использование выводов CapTivate™ IO для управления светодиодами

#### 1.1.4. Плата изолятора CAPTIVATE-ISO

Плата CAPTIVATE-ISO (рис.6) позволяет реализовать работу интерфейса программирования контроллера, а также работу интерфейсов I2C и UART в случае, когда целевой МК питается от внешнего источника или батарейки.

Два четырехканальных изолятора [ISO1541](#) и [ISO1540](#) и один трехканальный - [ISO7131](#).

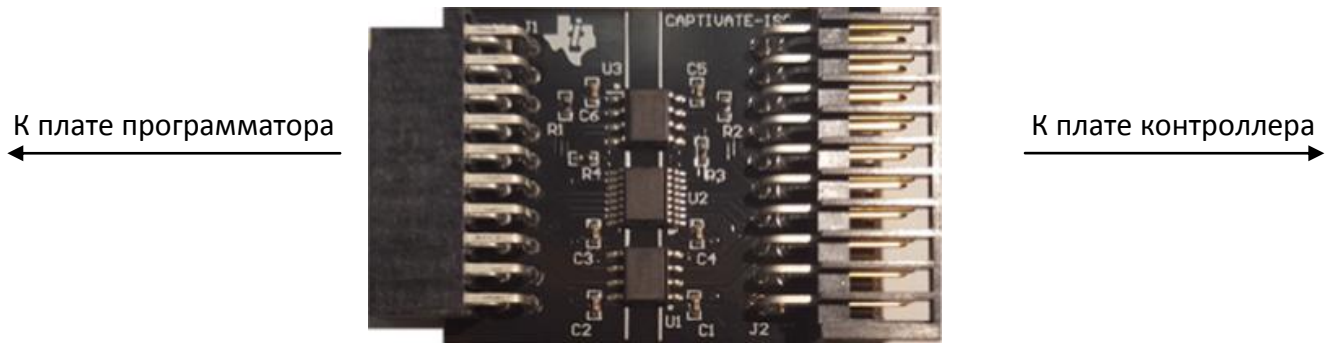


Рис.6 – Плата изолятора CAPTIVATE-ISO

## 1.2. Программное обеспечение

Для разработки сенсорных приложений с использованием технологии CapTivate нам потребуется две основных программы:

1. Для реализации скелета проекта, автоматической генерации кода инициализации периферии CapTivate™ и настройки параметров сенсоров в режиме реального времени используется приложение [MSP CapTivate Design Center GUI](#).
2. Для компиляции проекта, добавления функций пользователя и использования остальной периферии МК, а также для отладки проекта используется одна из двух возможных сред разработки: [CodeComposerStudio](#) или [IAR EmbeddedWorkbench](#)

Установку обоих приложений рекомендуется выполнить в директорию C:\ti\.

Приложения Code Composer Studio и CapTivate Design Center можно найти на сайте ti.com.



## 2. Работа с демо-проектом CapTivate

### 2.1. Загрузка демонстрационного проекта в Captivate Design Center

Знакомство со средой разработки программ для емкостных сенсоров начнем с загрузки демонстрационного проекта в плату контроллеров и проверки ее работы.

Запускаем среду разработки CapTivate Design Center любым удобным способом:

- с помощью ярлыка (рис.7)



Рис.7 – Ярлык CapTivate Design Center

- или через меню «Пуск» (рис.8)

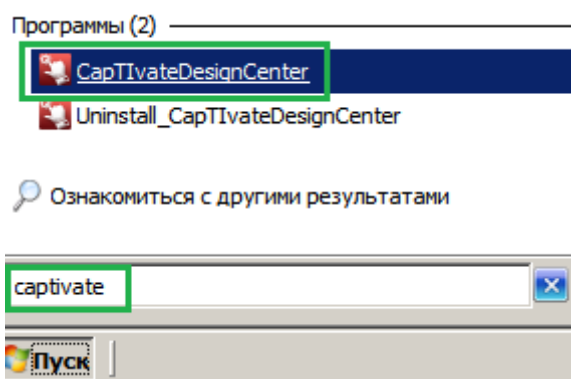


Рис.8 – Меню «Пуск»

Для открытия демонстрационного проекта в главном меню программы выбираем пункт File – Project Open (рис.9)

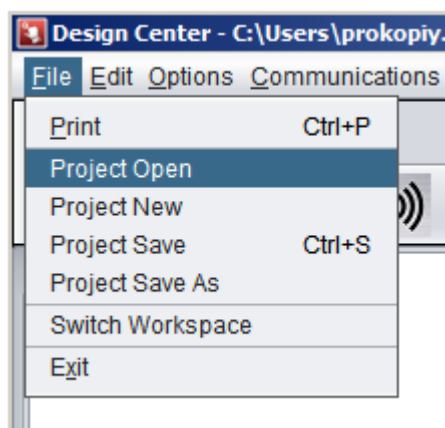


Рис.9 – Открытие проекта в CaptivateDesignCenter

По умолчанию Captivate Design Center сохраняет все проекты в своем рабочем пространстве, которое располагается в директории:

C:\Users\<имя пользователя>\CapTivateDesignCenter\_1\_30\_10\_00\CapTivateDesignCenterWorkspace\

Изменить расположение рабочего пространства можно через пункт меню File-Switch Workspace

Выбираем проект CAPTIVATE-PHONE(рис.10 - 1) и открываем его клавишей Select(рис.10 - 2)

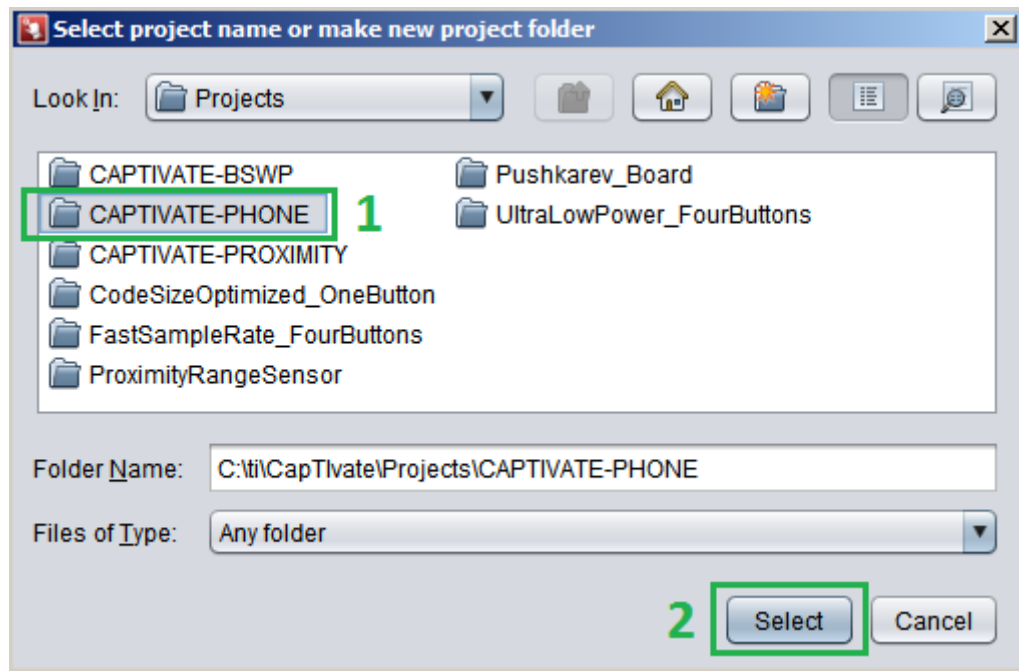


Рис. 10 – Открытие проекта в CaptivateDesignCenter

В главном окне программы появится сам проект, включающий описание проекта(рис.11 - 1) микроконтроллер(рис.11 - 2) и реализованные на плате CAPTIVATE-PHONE сенсоры (рис.11 - 3)

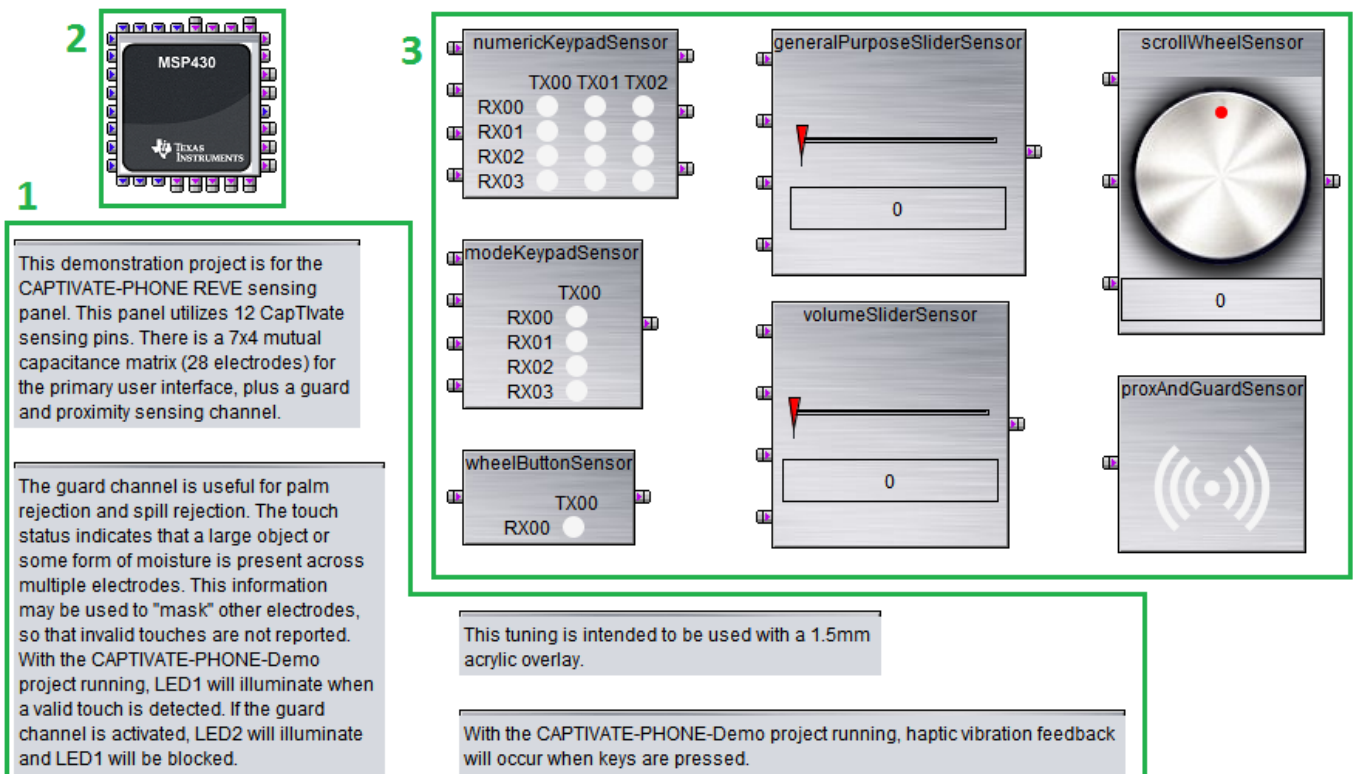


Рис.11 – Вид проекта в рабочей области captivateDesignCenter

## 2.2. Генерация исходного кода и проекта для IDE

Двойным нажатием левой клавишей мышки на символ микроконтроллера входим в его настройки (рис.12).



Рис.12 – Символ микроконтроллера в рабочем поле CaptivateDesignCenter

Здесь представлены все настройки проекта от подключения сенсоров к выводам микроконтроллера до настроек самого процесса измерения емкости сенсоров (рис.13). Основные настройки будут рассмотрены позже в работе по созданию своего проекта с нуля. Сгенерируем код и проект под среду разработки (IAREmbedded Workbench и CodeComposerStudio) нажатием Generate Source Code.

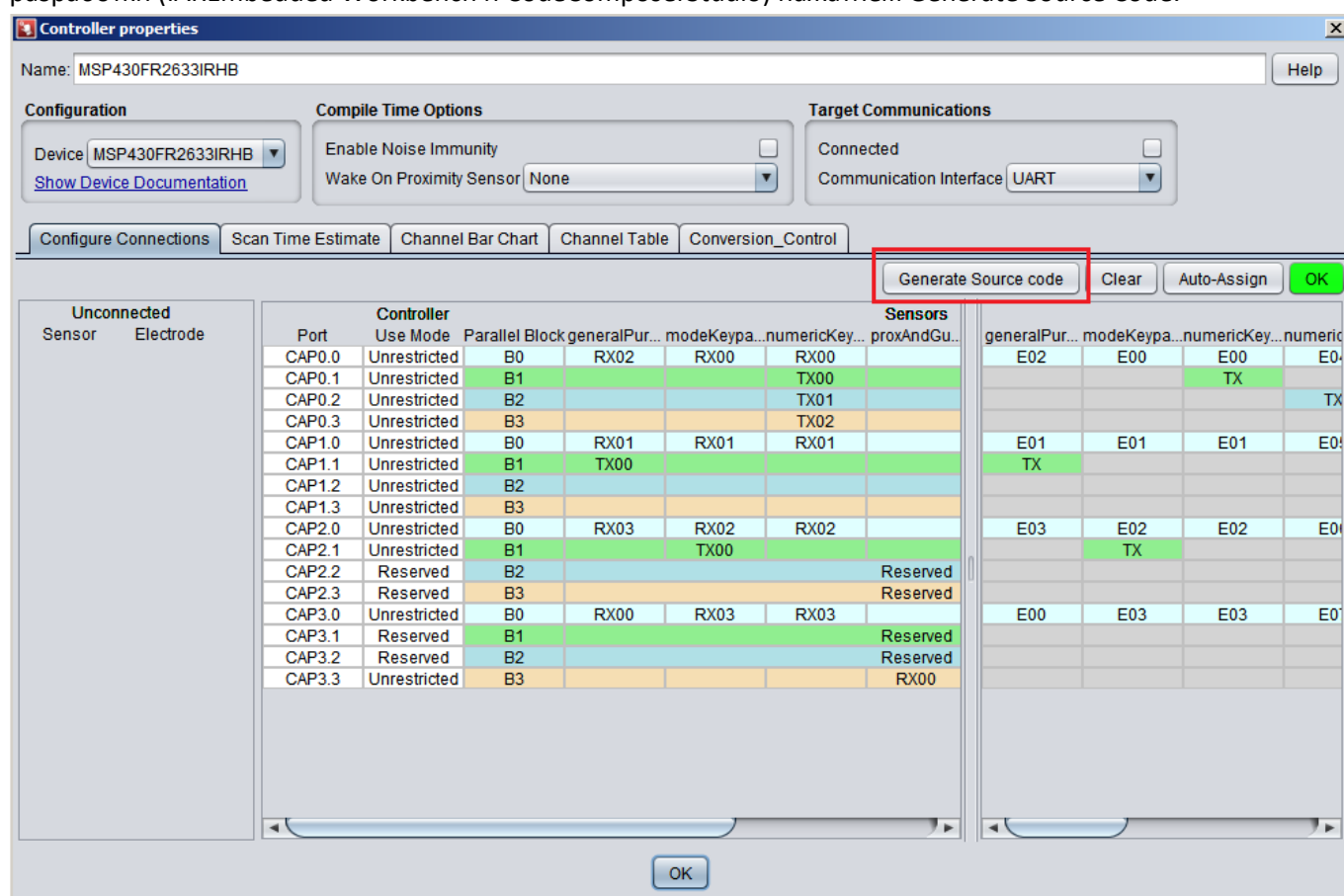


Рис.13 – Меню свойств микроконтроллера с выделенной кнопкой генерации кода

При сохранении проекта в диалоговом окне (рис.14) можно выбрать создание нового проекта или обновление уже существующего. Выбираем пункт Create New Project.

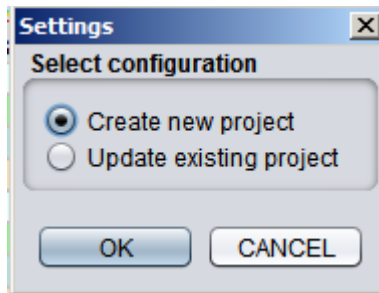


Рис.14 – Диалоговое окно сохранения проекта

Далее необходимо указать путь сохранения проекта (рекомендуется указать путь C:\ti\Captivate\Projects) и нажать OK (рис.15).

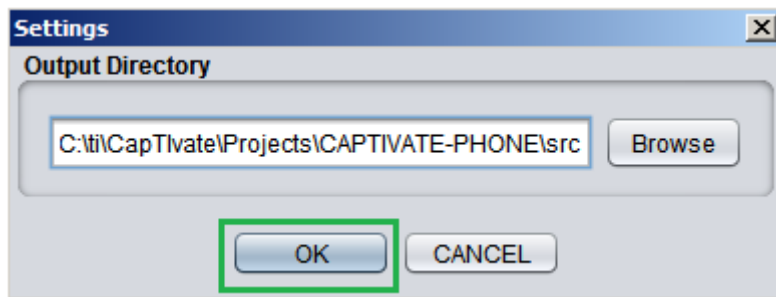
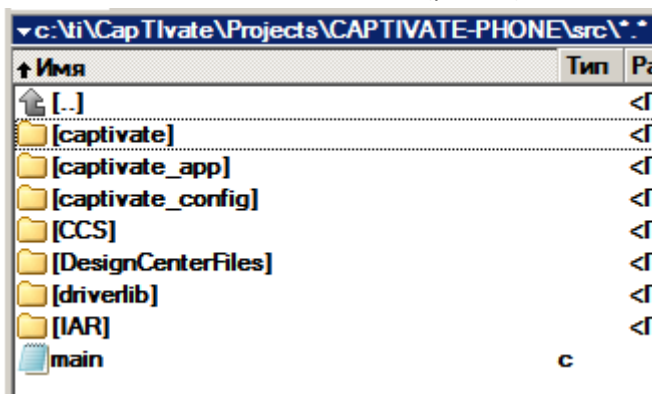


Рис.15 – Диалоговое окно пути сохранения проекта

После этого в указанной папке появятся файлы с исходным кодом, а также проекты для сред разработки CodeComposerStudio и IAREmbeddedWorkbench (рис.16).



CCS- каталог с проектом для CodeComposerStudio  
IAR- каталог с проектом для IAR Embedded Workbench

Рис.16 – Содержимое папки проекта Captivate после экспорта

### 2.3. Импорт проекта в CodeComposerStudio

Запускаем CodeComposerStudio любым удобным способом:

- с помощью ярлыка (рис.17)



Рис.17 – Ярлык CodeComposerStudio

- или через меню «Пуск» (рис.18)

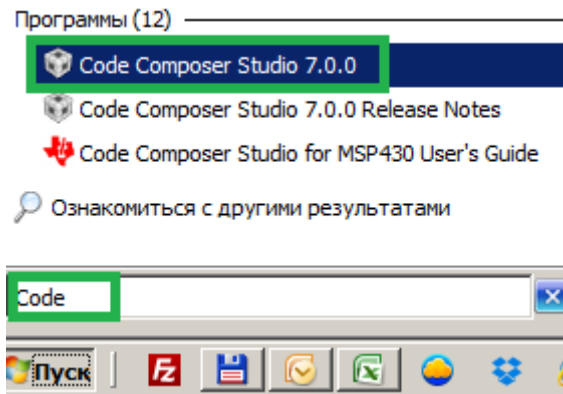


Рис.18 – Меню пуск

Затем импортируем проект через пункт главного меню Project – Import CCS Projects ... (рис.19)

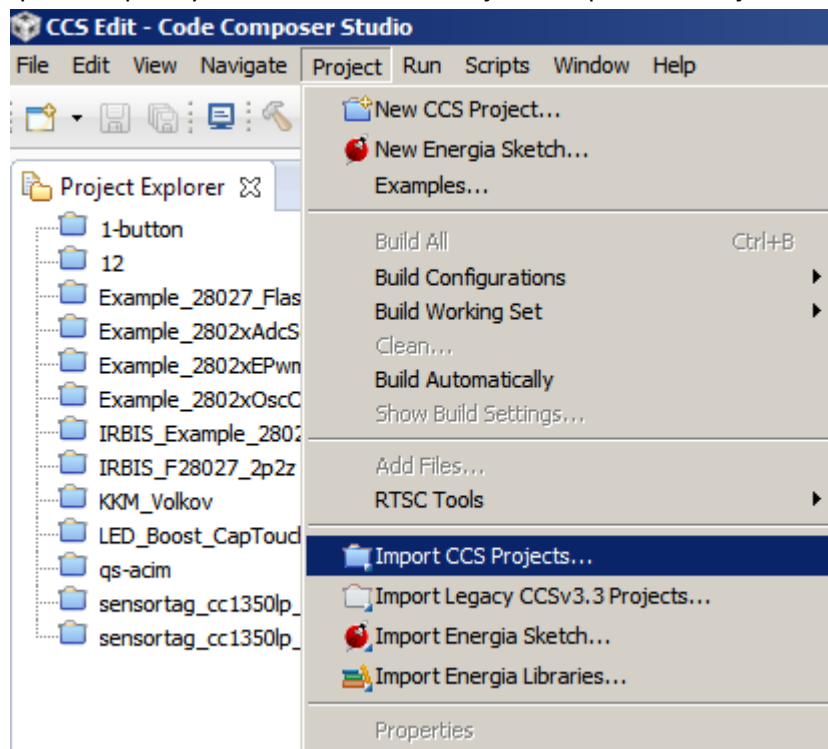


Рис.19 – Импорт проекта в CodeComposerStudio

В появившемся меню необходимо указать директорию, в которой содержится ранее сохраненный проект. Укажем директорию, используя клавишу Browse (рис.20):

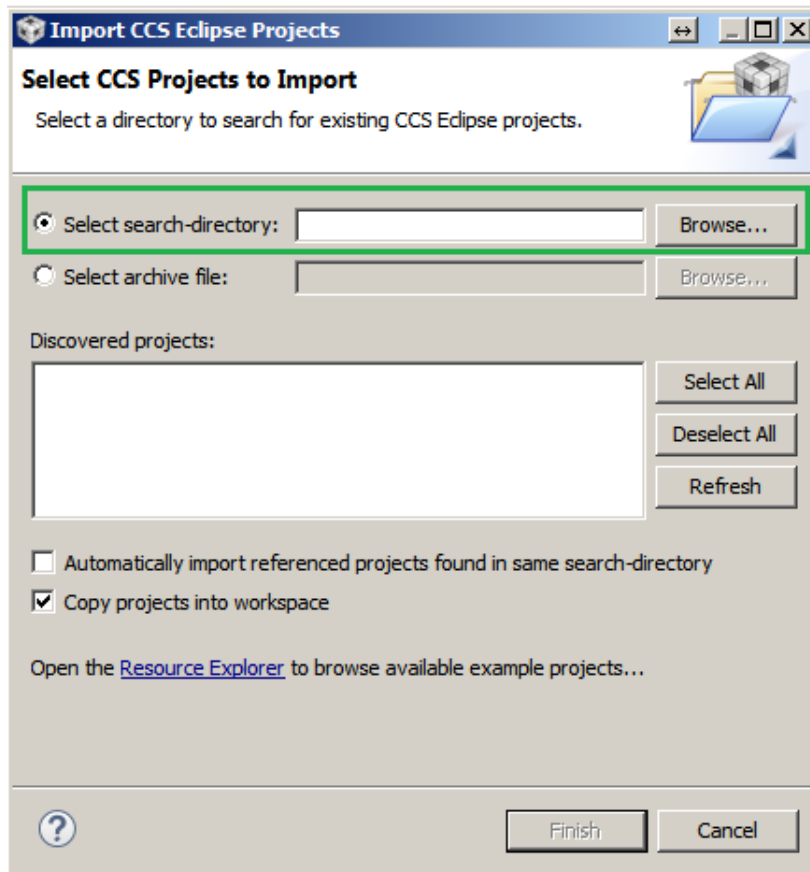


Рис.20 – Выбор директории с проектом в CodeComposerStudio

Указываем папку CCS в директории src, в которую ранее сохранили проект из Captivate Design Center и жмем ОК (рис.21):

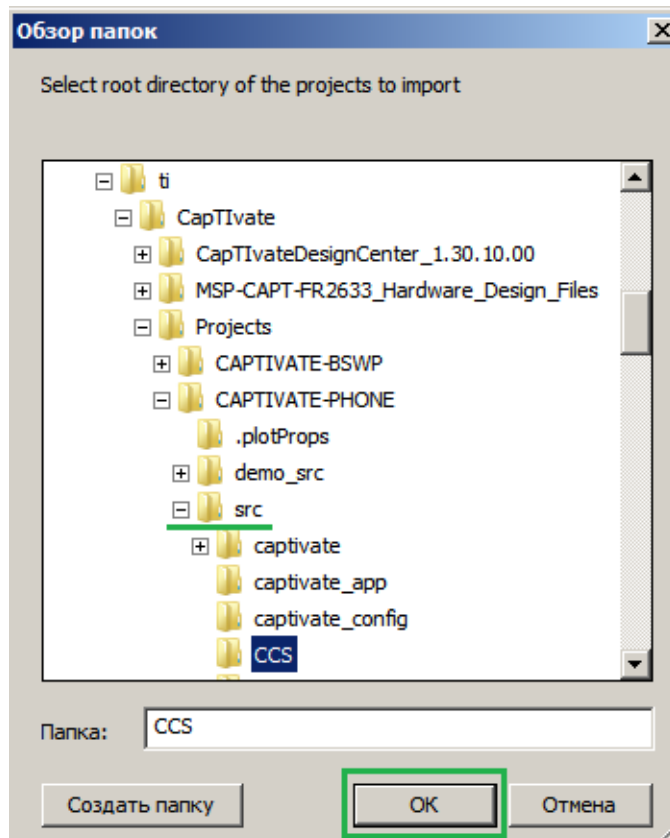


Рис.21 – Выбор директории с проектом в CodeComposerStudio

В окне Discovered Projects появится проект CAPTIVATE-PHONE (рис.22). Убедившись, что напротив названия проекта установлена галочка, необходимо нажать клавишу Finish, после чего импорт проекта будет завершен.

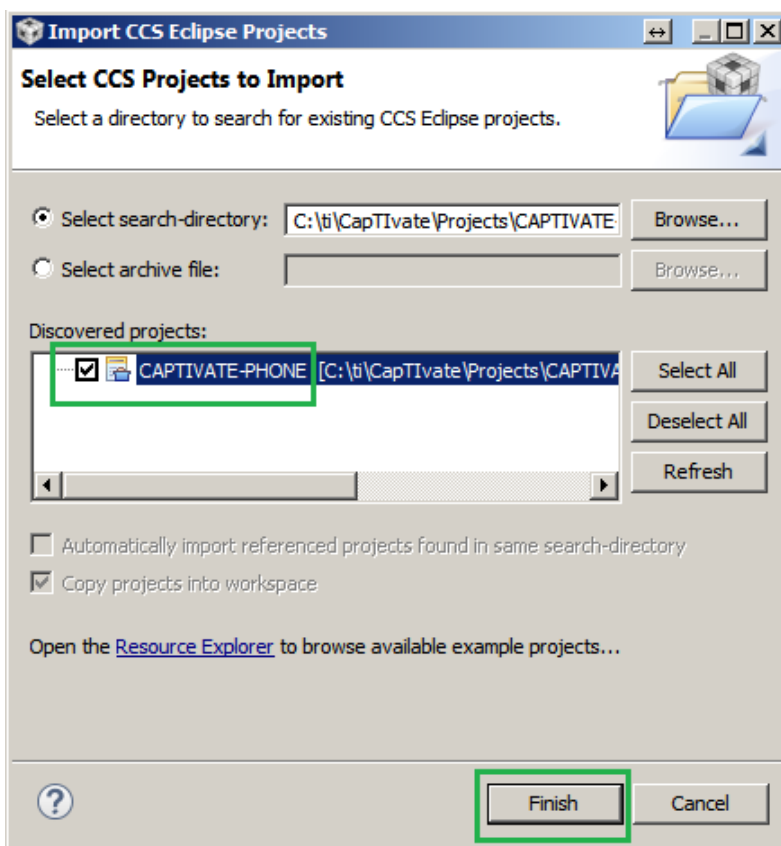


Рис.22 – Завершающий этап добавления проекта

Проект должен появиться в окне Project Explorer, которое находится в левой части экрана (рис.23).

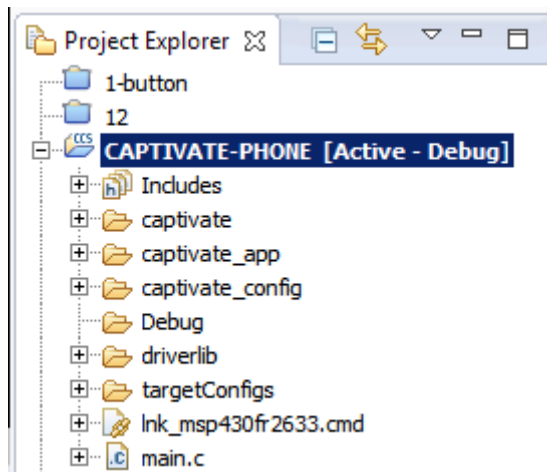


Рис.23 – Проект в окне ProjectExplorer

Если у Вас отсутствует окно ProjectExplorer, убедитесь что Вы находитесь в перспективе CCS Edit в правом верхнем углу окна программы (рис.24)



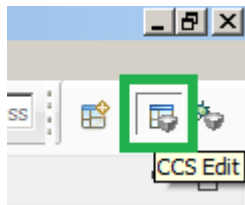


Рис.24 – Кнопки переключения режимов отображения Edit/Debug

А затем включите отображение окна Project Explorer через меню View-ProjectExplorer (рис.25).

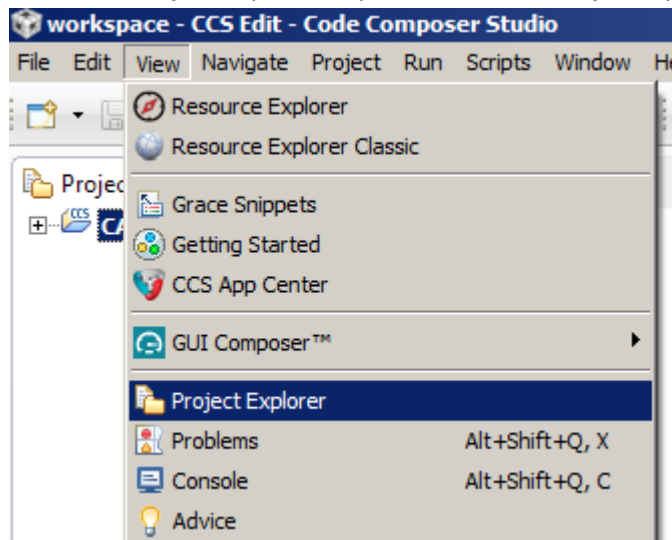


Рис.25 – Включение окна ProjectExplorer

#### 2.4. Загрузка программы в микроконтроллер

Для загрузки программы в микроконтроллер [MSP430FR2633](#) необходимо соединить несколько плат из отладочного набора [MSP-CAPT-FR2633](#) в следующем порядке (рис.26):

CAPTIVATE-PGMR(1) – CAPTIVATE-FR2633(2) – CAPTIVATE-PHONE(3)

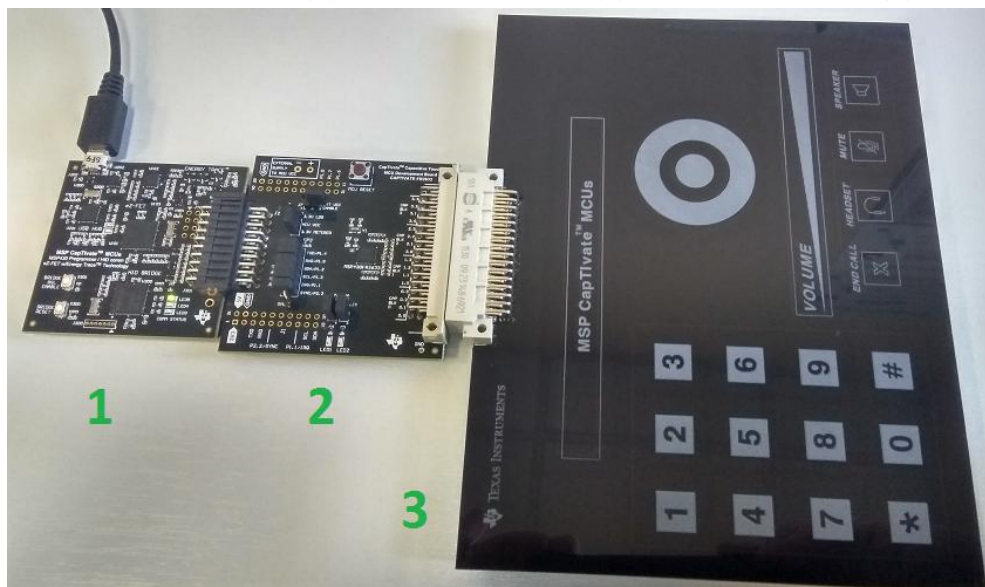


Рис.26 – Конфигурация отладочного набора для выполнения первой работы

Затем необходимо подключить отладочную плату к ПК через USB-разъем в плате CAPTIVATE-PGMR(1).

После автоматической установки драйверов (загрузку драйверов из центра обновления Windows рекомендуется пропустить), устройство готово к работе.

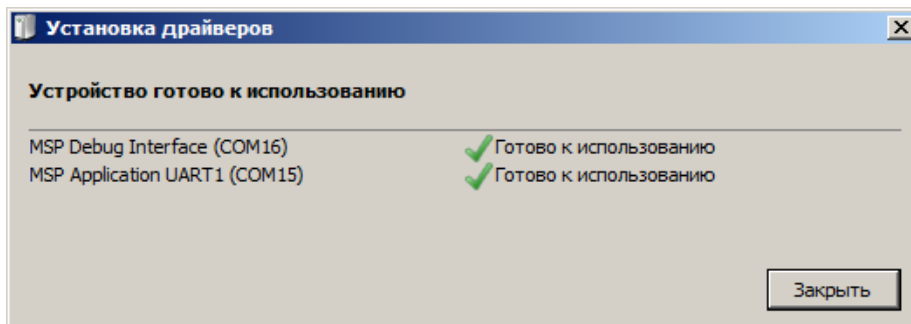


Рис.27 – Сообщение системы об удачной установке драйверов

Для запуска компиляции программы с последующей загрузкой её в контроллер и запуска процесса отладки необходимо нажать на кнопку Debug ( с изображением зеленого жука) на панели инструментов Code Composer Studio (рис.28).

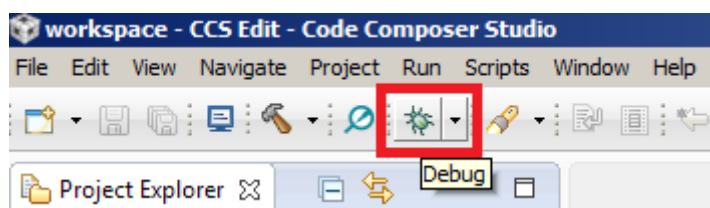


Рис.28 – Кнопка Debug в CodeComposerStudio

В случае успешной компиляции и загрузки программы, программный счетчик микроконтроллера должен остановить выполнение кода на первой строке функции main(), о чем будет свидетельствовать синяя стрелка напротив этой строки (рис.29).

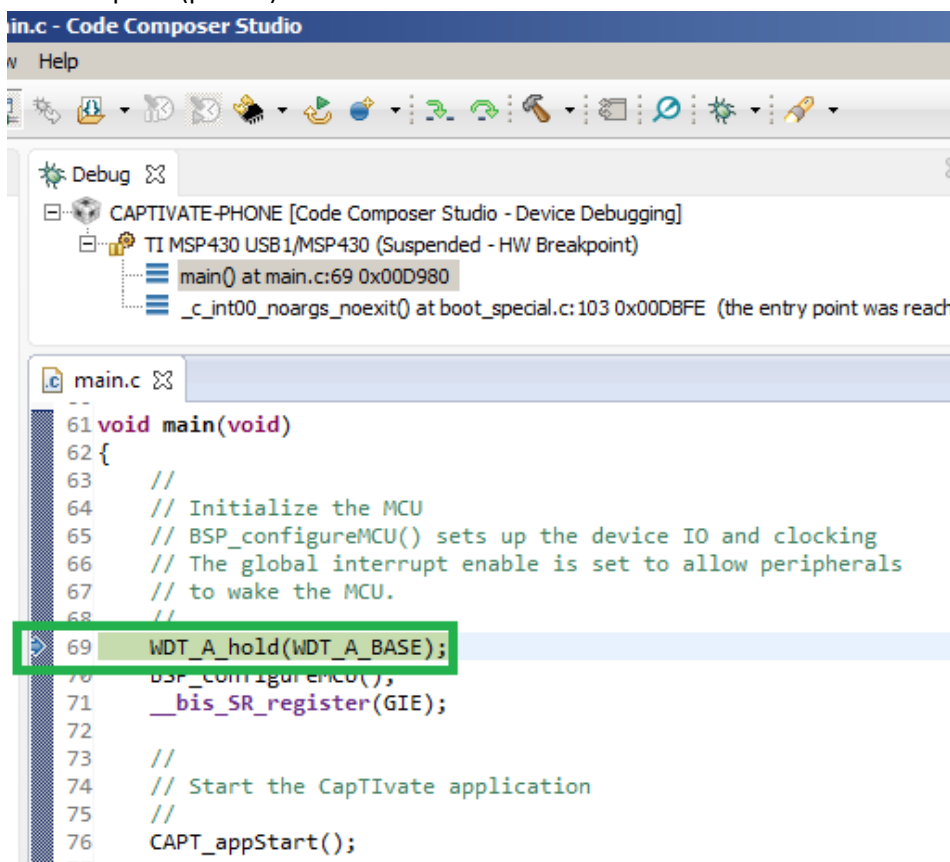


Рис.29 – Автоматическая остановка выполнения кода на первой строке функции main()

Завершаем процесс отладки нажатием на кнопку Terminate (с изображением красного квадрата) (рис.30)

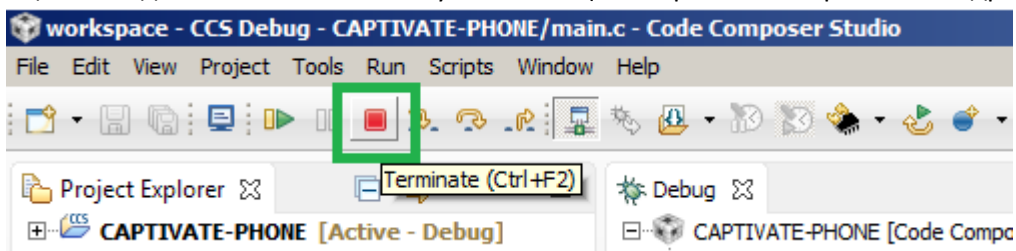


Рис.30 – Кнопка Terminate в CodeComposerStudio

Сейчас программа загружена в [MSP430FR2633](#) и теперь можно вернуться в CapTivate Design Center для наблюдения данных от емкостных сенсоров в реальном времени.

## 2.5. Отображение данных сенсора в реальном времени

В среде CapTivate Design Center для подключения к контроллеру и получения потоковых данных от емкостных сенсоров необходимо выбрать пункт меню Communications – Connect (рис.31).

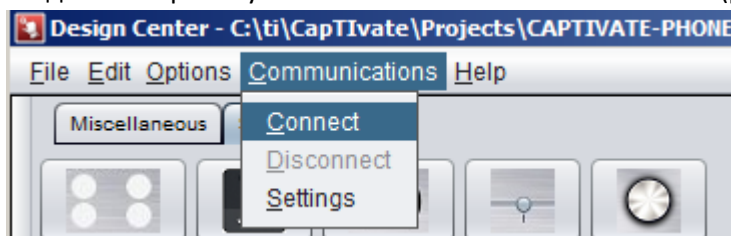


Рис.31 – Подключение к плате в CapTivateDesignCenter

Об успешном подключении будет свидетельствовать надпись HID Connected в левом нижнем углу приложения (рис.32).

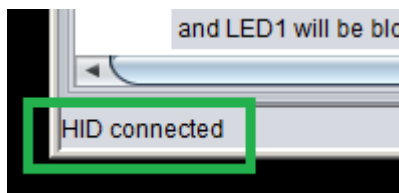


Рис.32 – Индикатор успешного подключения к плате

Затем необходимо сбросить контроллер нажатием на клавишу MCU RESET на плате контроллера, чтобы запустить корректную работу HID-устройства:



Рис.33 – Кнопка RESET на плате контроллера

Просматривать состояние и параметры сенсоров можно двумя способами.

### 2.5.1. Отображение работы сенсоров в рабочем поле проекта

Наблюдение за состоянием сенсоров в рабочем поле проекта является первичным инструментом проверки работы сенсоров. Если аппаратная и программная части проекта реализованы без ошибок, то каждый сенсор при взаимодействии с ним изменяет свое отображение в рабочем поле проекта.

Например, кнопки меняют свой цвет на оранжевый при приближении к ним пальца, а при нажатии на них – на зеленый (рис.34).

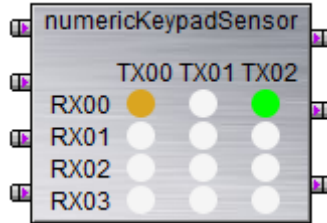
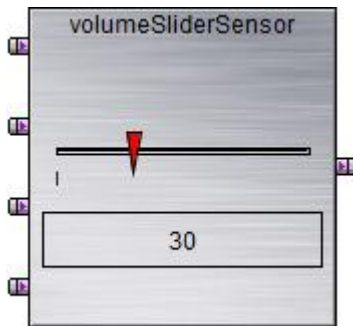
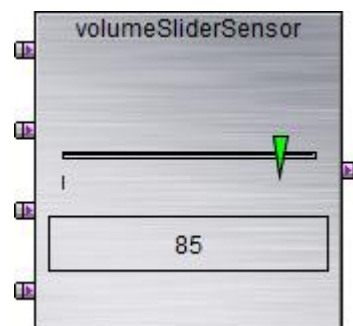


Рис.34 - Кнопка 1 детектирует приближение пальца, кнопка 3 нажата

При касании к слайдеру цвет движка изменяется с красного на зеленый, и при изменении положения пальца на нем отображает изменения в реальном времени (рис.35).



Нет касания к слайдеру



Есть касание к слайдеру, и изменено его значение

Рис.35 – Изменение значения слайдера

Проверьте отображение изменения состояния в рабочем окне всех сенсоров, находящихся на плате.

При реализации своих плат емкостных сенсоров, отличных от отладочных, требуется не просто убедиться в работоспособности сенсора, но и настроить параметры его срабатывания. Для настройки сенсоров и получения более точных данных используется второй способ отображения данных.

### 2.5.2. Отображение параметров в свойствах объекта

Более подробная информация о параметрах сенсора доступна для каждого объекта в его свойствах.

Чтобы перейти в меню свойств сенсора необходимо нажать два раза левой клавишей мыши на нем в рабочем поле.

В свойствах объекта параметры сенсора в реальном времени можно наблюдать в трех вкладках:

- Channel Bar Chart – отображение данных по каждому элементу сенсора в виде столбцов (рис.36)
- Channel Oscilloscope Plot – отображение в виде графиков (рис.37)
- Channel Table – отображение численных данных в табличном виде (рис.38)

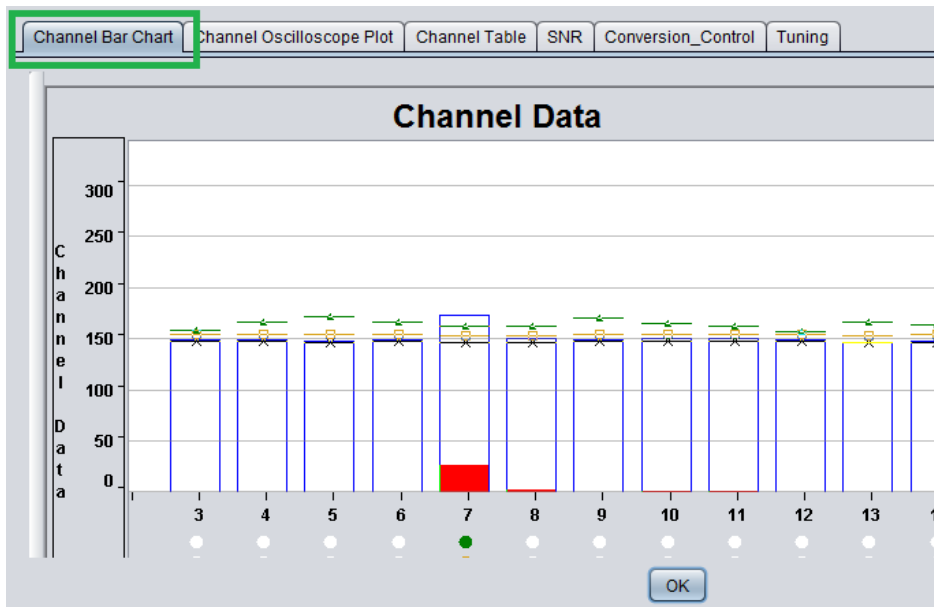


Рис.36 - Отображение данных в виде столбцов

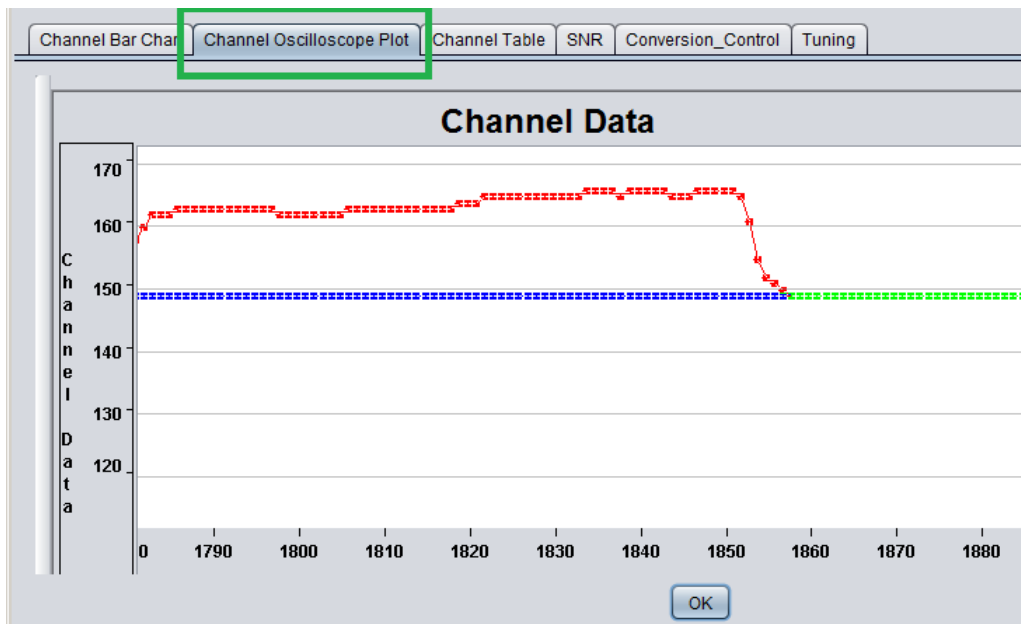


Рис.37 - Отображение данных в графическом виде

Channel	Sensor	Element	TX	RX	Time C...	Port	Parallel...	LTA	Count	Delta	Prox Th...	Touch ...	Prox St...	Touch ...
3	numeric...	E00	TX00	RX00	numeric...	CAP0.0	B0	149	164	15	156	159	●	●
4	numeric...	E01	TX00	RX01	numeric...	CAP1.0	B0	149	154	5	156	168	●	●
5	numeric...	E02	TX00	RX02	numeric...	CAP2.0	B0	148	150	2	155	173	●	●
6	numeric...	E03	TX00	RX03	numeric...	CAP3.0	B0	149	151	2	156	167	●	●
7	numeric...	E04	TX01	RX00	numeric...	CAP0.0	B0	147	149	2	154	164	●	●
8	numeric...	E05	TX01	RX01	numeric...	CAP1.0	B0	147	148	1	154	164	●	●
9	numeric...	E06	TX01	RX02	numeric...	CAP2.0	B0	149	150	1	156	171	●	●
10	numeric...	E07	TX01	RX03	numeric...	CAP3.0	B0	149	150	1	156	166	●	●
11	numeric...	E08	TX02	RX00	numeric...	CAP0.0	B0	149	150	1	156	164	●	●
12	numeric...	E09	TX02	RX01	numeric...	CAP1.0	B0	149	149	0	156	158	●	●
13	numeric...	E10	TX02	RX02	numeric...	CAP2.0	B0	147	147	0	154	167	●	●
14	numeric...	E11	TX02	RX03	numeric...	CAP3.0	B0	148	149	1	155	165	●	●

Рис.38 - Отображение данных в табличном виде

Выберите один любой сенсор на плате и исследуйте изменение его параметров при различных режимах отображения: во вкладке Channel Bar Chart, Channel Oscilloscope Plot, Channel Table.

Определите значения порогов детектирования приближения и нажатия.

### 3. Реализация нового проекта емкостной кнопки в режиме SelfCapacitance

#### 3.1. Аппаратная часть

Для данной работы используется плата сенсоров CAPTIVATE-BSWP (название платы можно узнать по маркировке на её оборотной стороне). В этой работе мы реализуем программу для обработки емкостного сенсора, расположенного на плате под номером 1 и проведем настройку его параметров (рис.39).



Рис.39 – Конфигурация отладочного набора для выполнения второй работы

### 3.2. Создание нового проекта в Captivate Design Center

Для создания нового проекта в программе Captivate Design Center необходимо выбрать пункт главного меню File - Project New (рис.40)

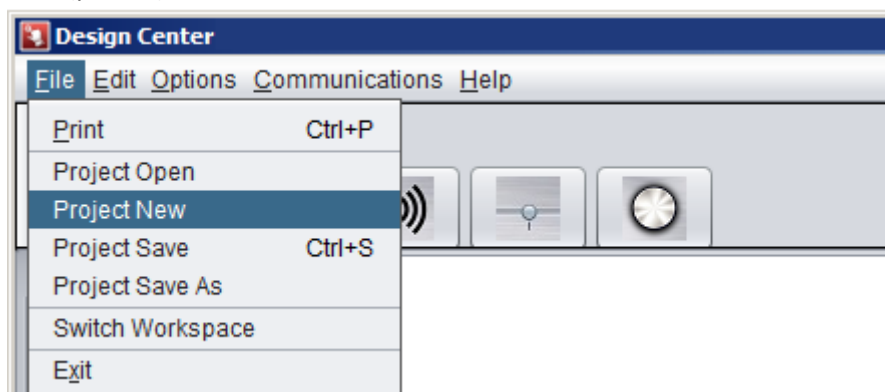


Рис.40 – Создание нового проекта в CaptivateDesignCenter

Затем в появившемся окне задать имя нового проекта

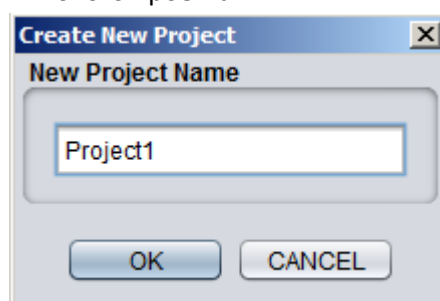


Рис.41 – Имя нового проекта

После этого необходимо добавить микроконтроллер, для которого мы планируем реализовать проект. Для добавления контроллера необходимо навести курсор нажать на символ микросхемы на панели инструментов,

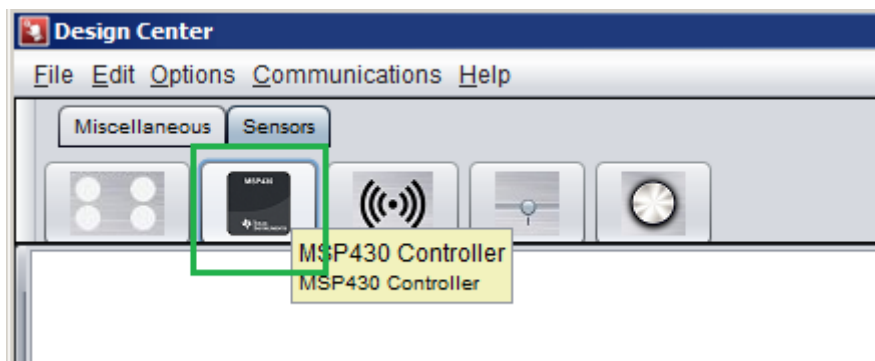


Рис.42 – Добавление микроконтроллера в проект

а затем переместить курсор на рабочее поле и однократным нажатием левой клавиши мыши установить компонент.



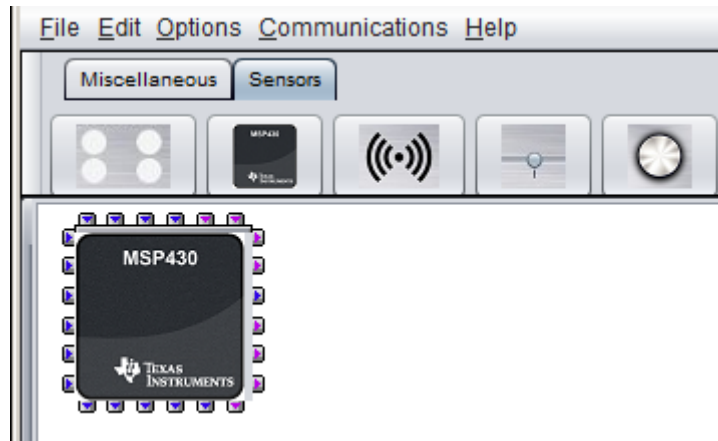


Рис.43 – Установка контроллера в рабочее поле

Т.к. данную работу мы выполняем на готовой аппаратной части, то нам необходимо уточнить номер вывода микроконтроллера, к которому подключен сенсор клавиши “1”. Т.к. к отладочному набору [MSP-CAPT-FR2633](#) прилагается полный комплект [технической документации](#), то это можно сделать по принципиальной схеме или топологии печатной платы сенсоров и платы контроллера (рис.44, рис.45).

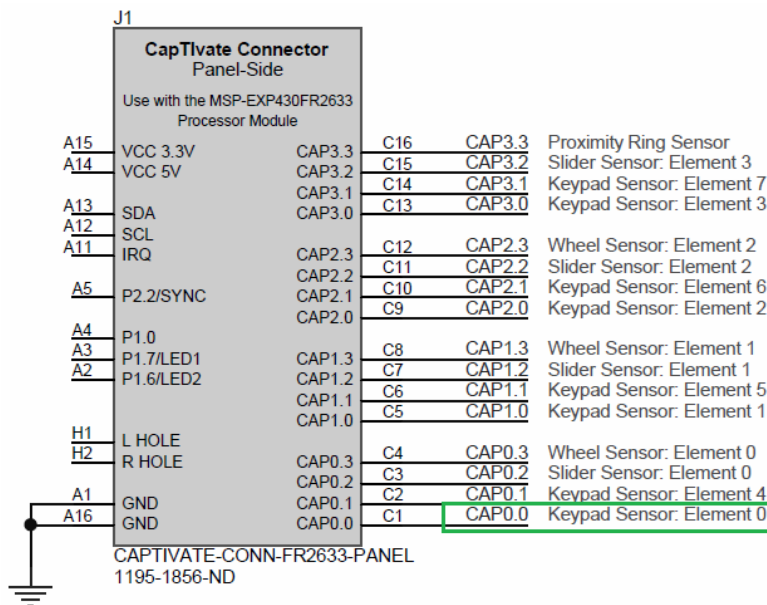


Рис.44 - Фрагмент принципиальной схемы для платы CAPTIVATE-BSWP

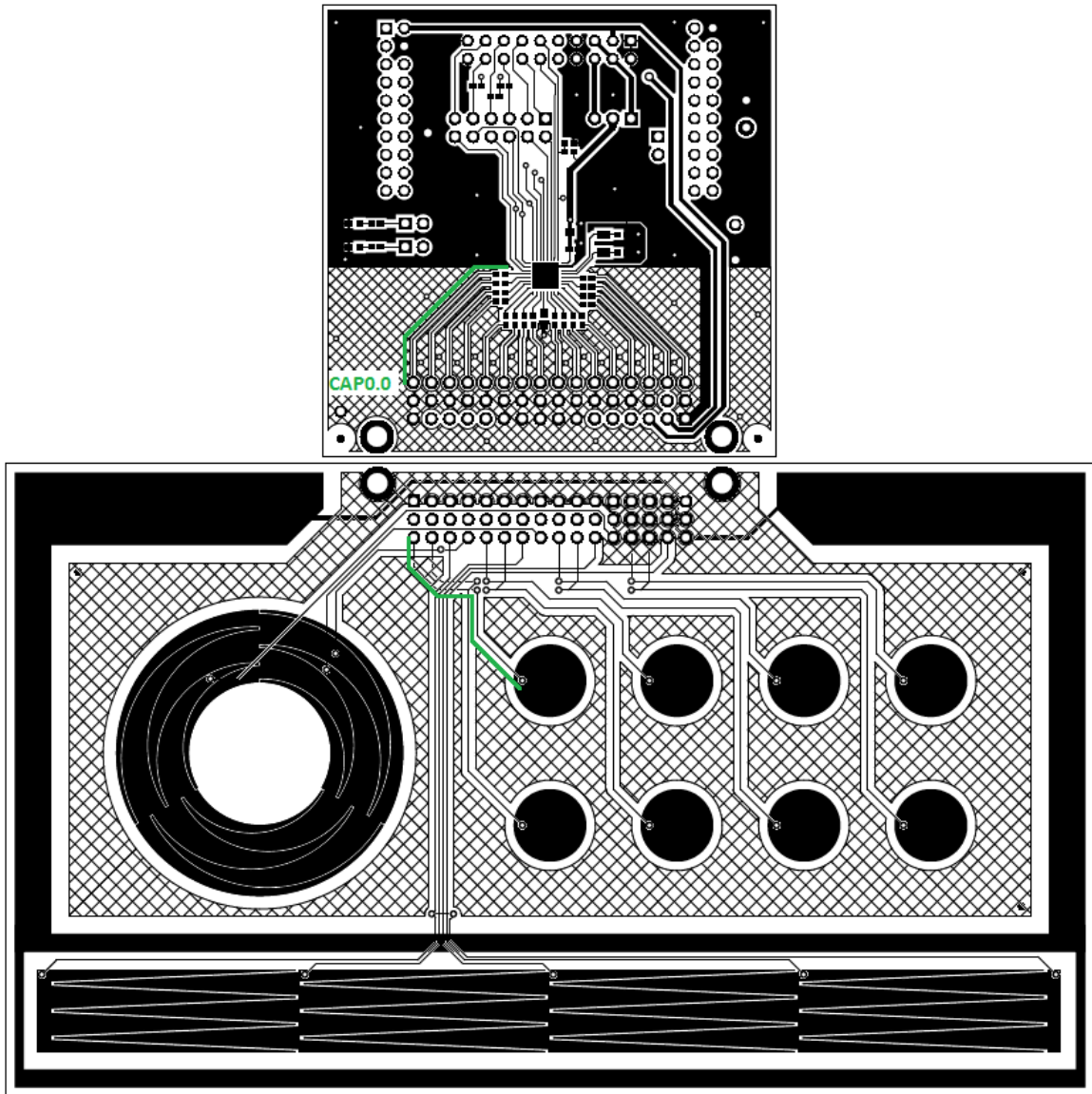


Рис.45 - Топология печатных плат CAPTIVATE-FR2633 и CAPTIVATE-BSWP

Двойным нажатием левой клавиши мышки по изображению контроллера в рабочем поле открываем настройки микроконтроллера.

Необходимо настроить 3 параметра (рис.46):

- 1) модель микроконтроллера – [MSP430FR2633IRHB](#)
- 2) режим работы вывода CAP0.0 – Self Only
- 3) интерфейс обмена данными с хостом для настройки в реальном времени - UART

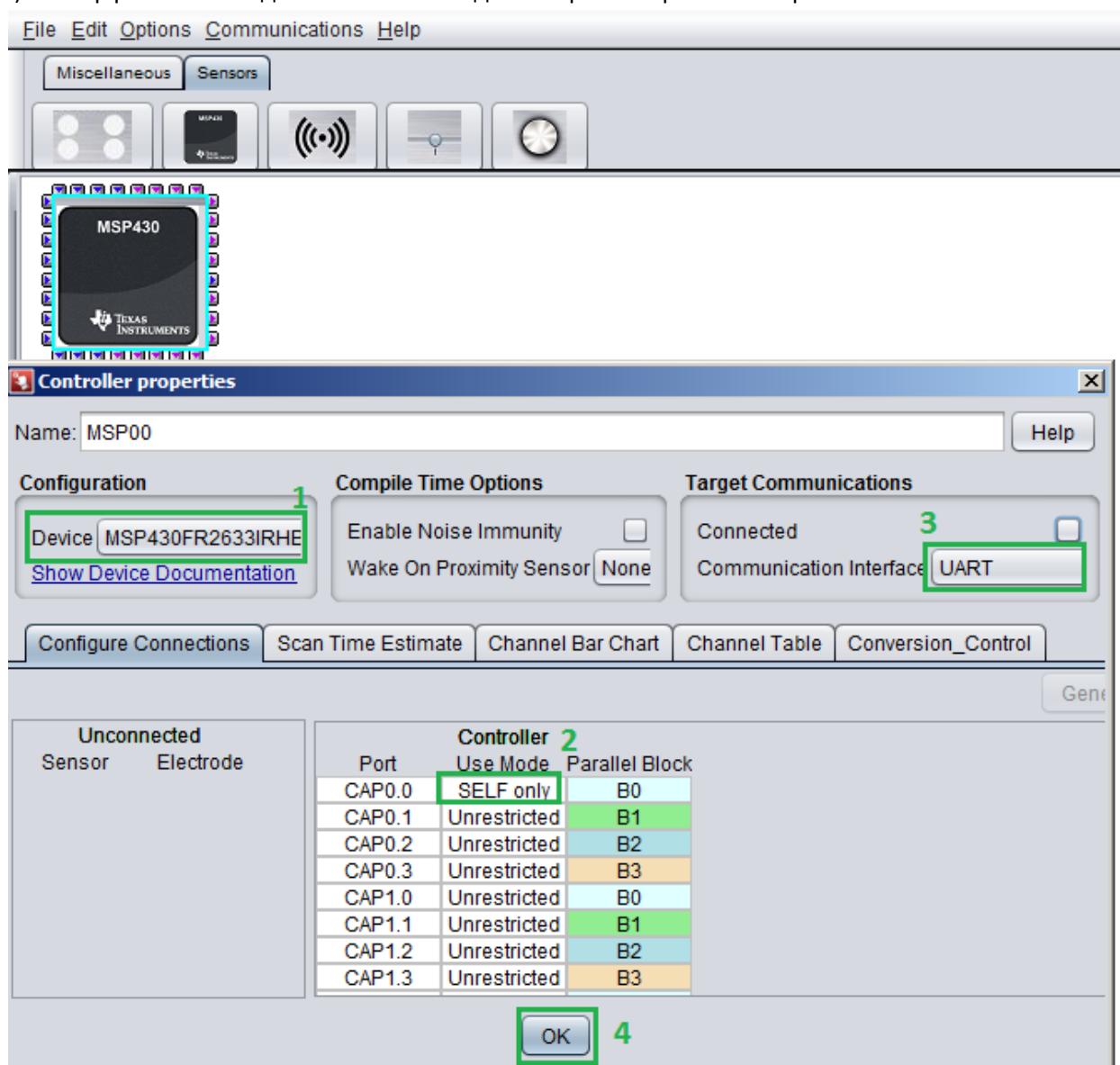


Рис.46 – Настройка параметров контроллера

Теперь в проект необходимо добавить сам сенсор. Для этого нажимаем на символ группы кнопок в панели инструментов и устанавливаем его в рабочее поле проекта (рис.47).

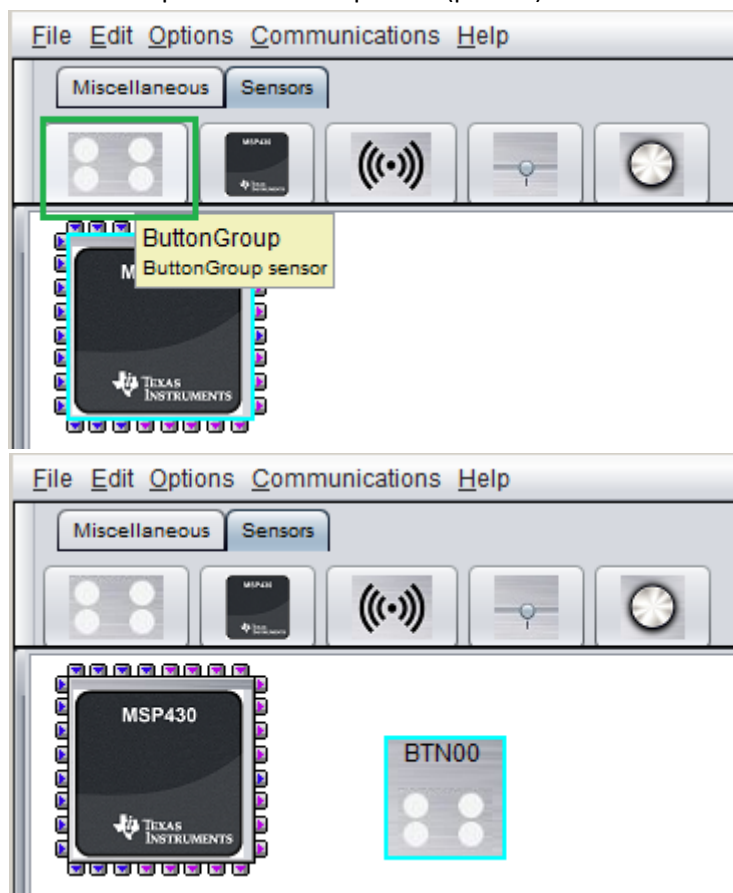


Рис.47 – Добавление сенсора в проект

Двойным щелчком по символу группы кнопок в рабочем поле переходим в настройки сенсора.

Для сенсора нам необходимо настроить 3 параметра (рис.48):

- 1) имя элемента – BTN01
- 2) режим работы сенсора – SELF
- 3) количество элементов - 1

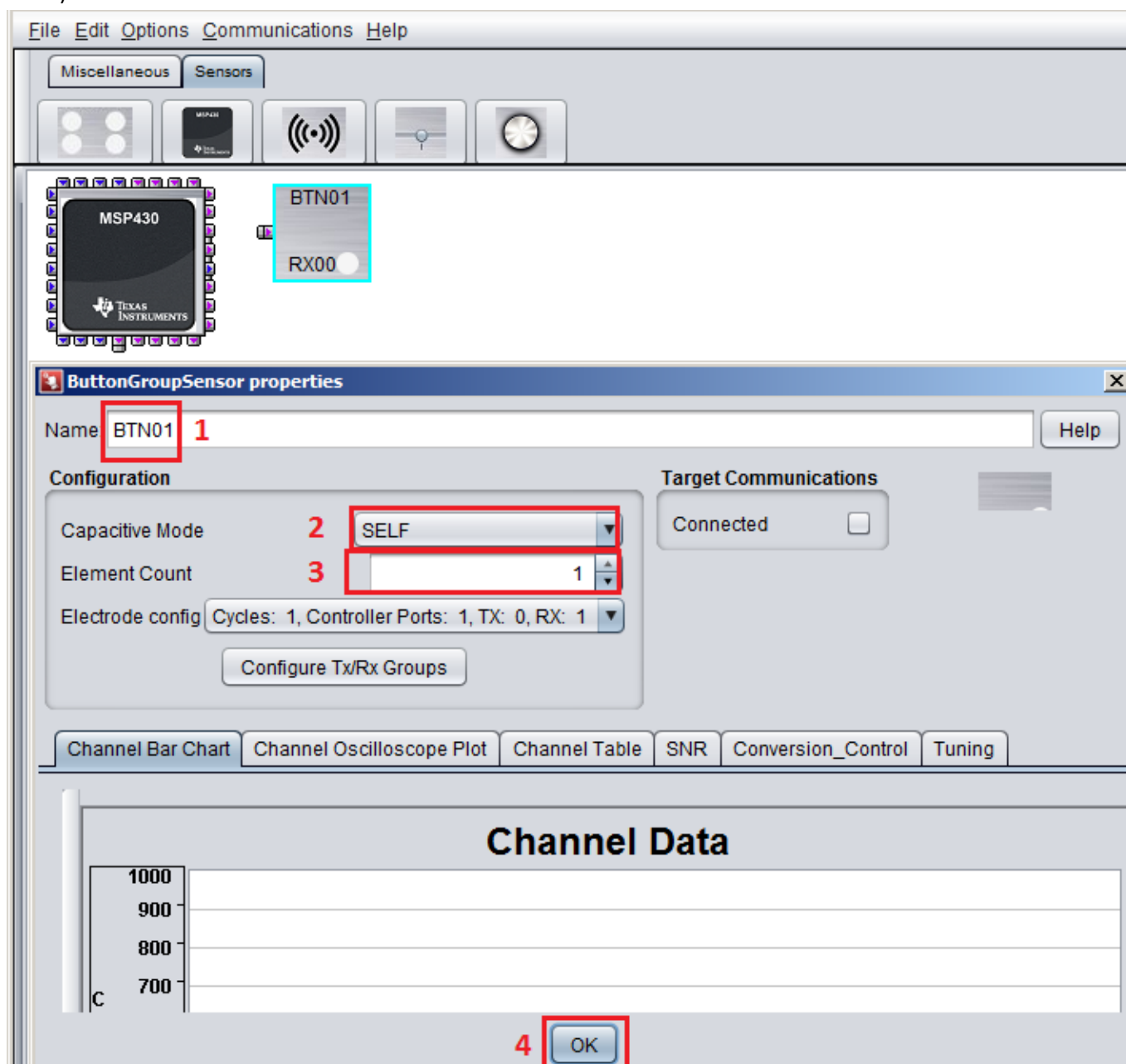


Рис.48 – Настройка параметров сенсора

После этого возвращаемся в настройки контроллера и устанавливаем подключение вывода CAP0.0 к сенсору BTN01. При правильной настройке в правой части окна свойств контроллера должен загореться зеленый индикатор «OK»(рис.49 - 3).

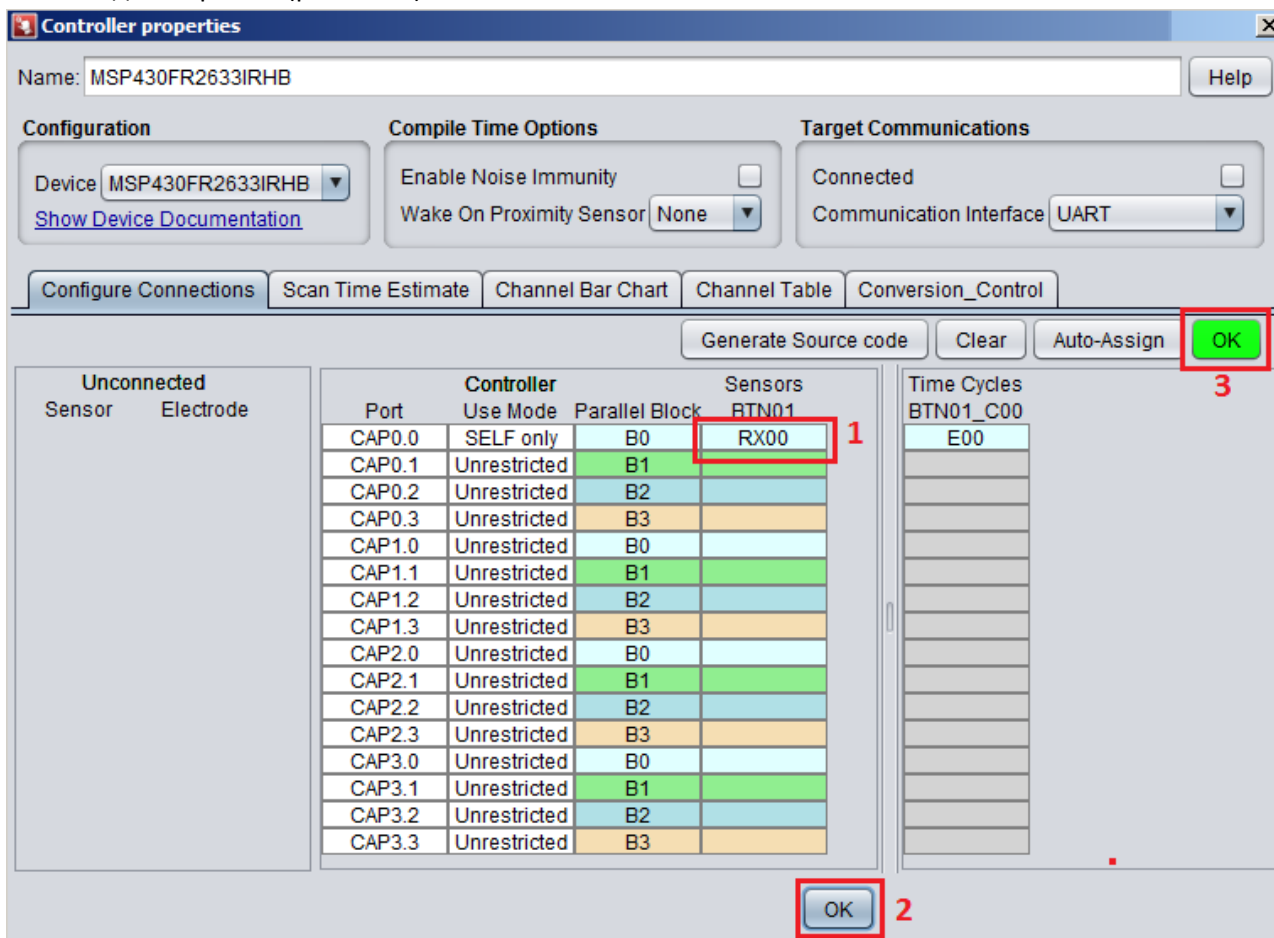


Рис.49 – Настройка подключения контроллера к сенсору

Правильность настройки можно также проверить визуально, включив отображение линий связи в меню настроек Options – Display – Show connection fly lines

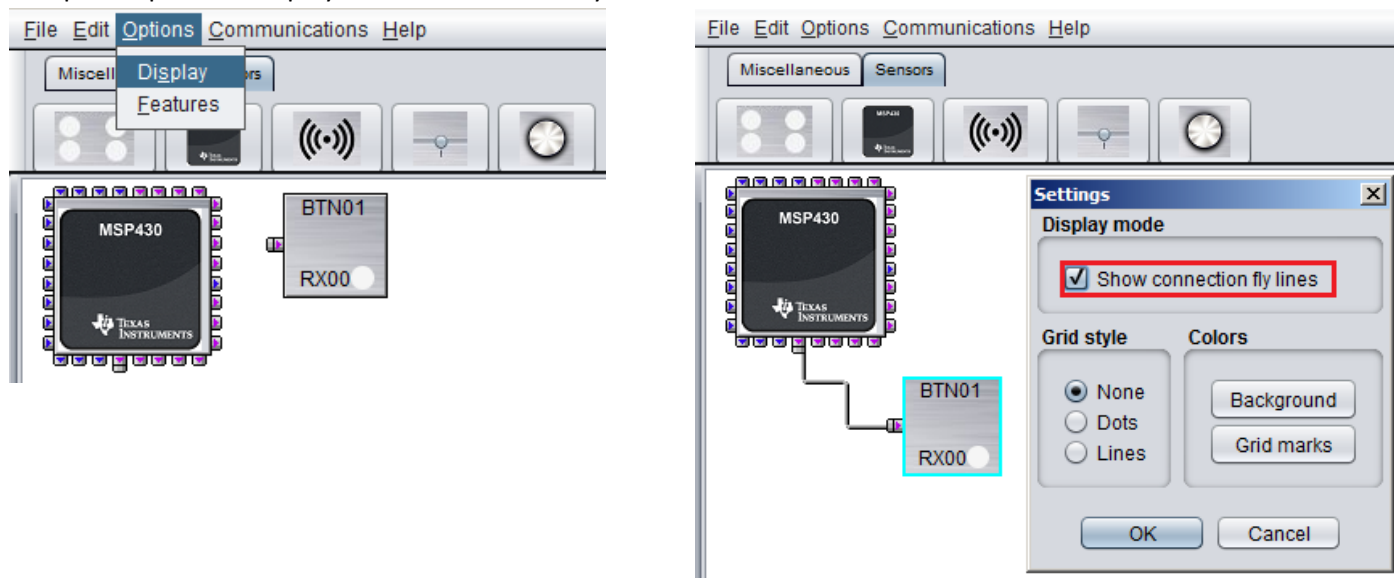


Рис.50 – Отображение линий связи

После установки и настройки компонентов необходимо сгенерировать файлы с исходным кодом и проект для среды разработки приложений.

Для этого необходимо в настройках микроконтроллера нажать клавишу Generate Source Code (рис.51).

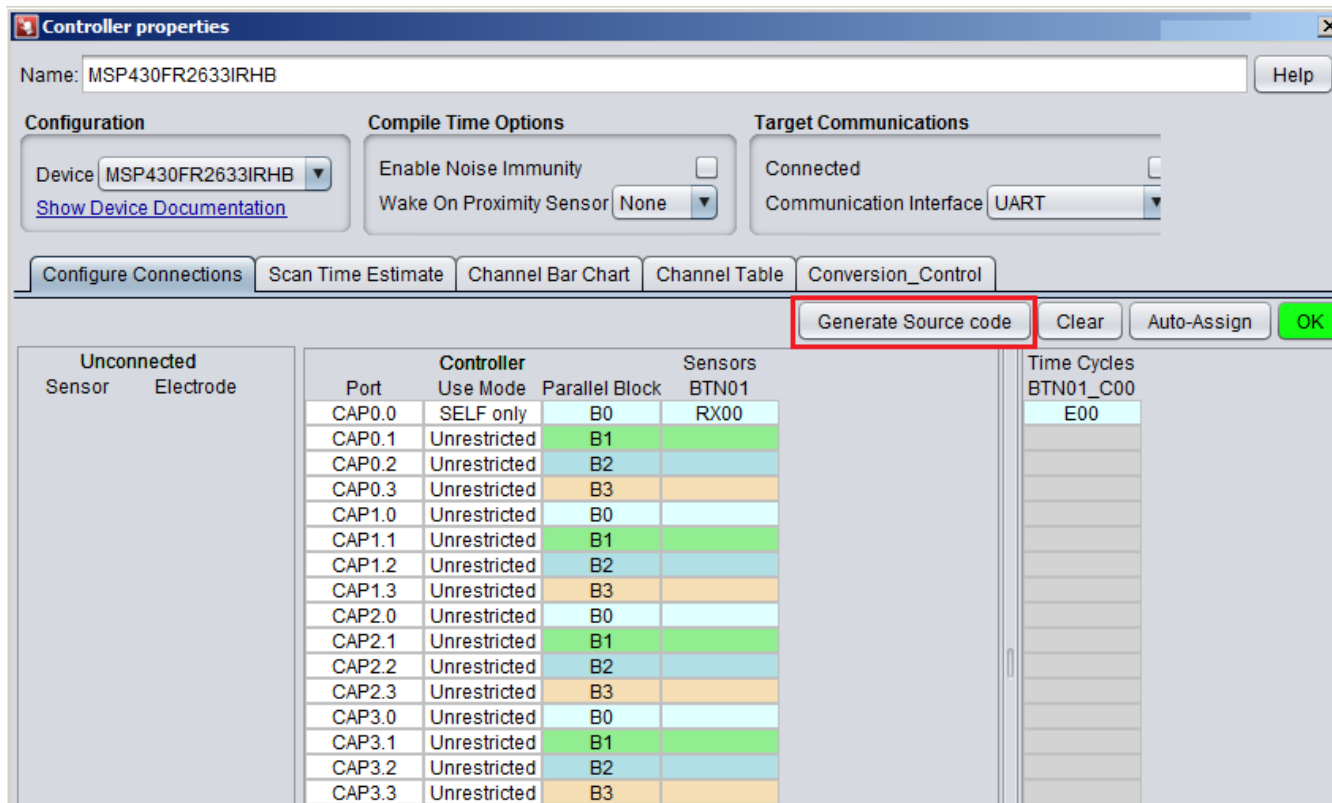


Рис.51 – Экспорт проекта из CodeComposerStudio

Затем, для создания нового проекта выбрать вариант Create New Project (рис.52).

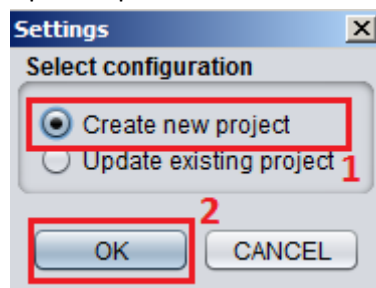


Рис.52 – Диалоговое окно создания проекта



Указать путь сохранения нового проекта (рекомендуется использовать директорию C:\ti\workspace\<имя проекта>) (рис.53).

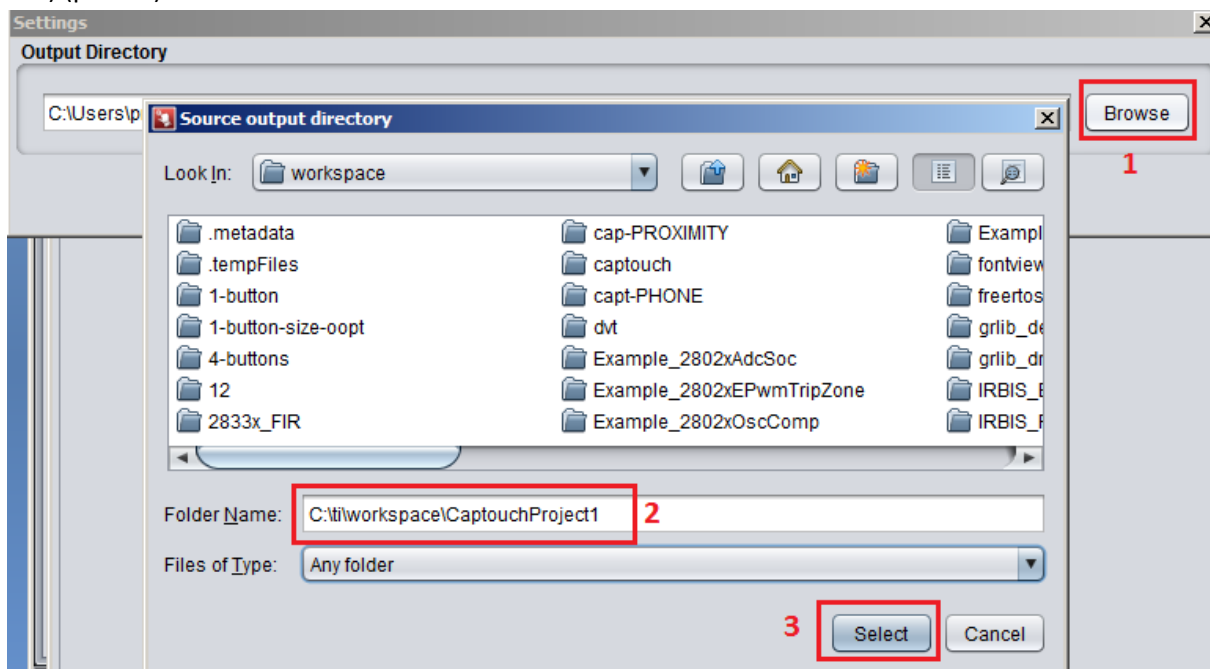


Рис.53 – Выбор директории проекта

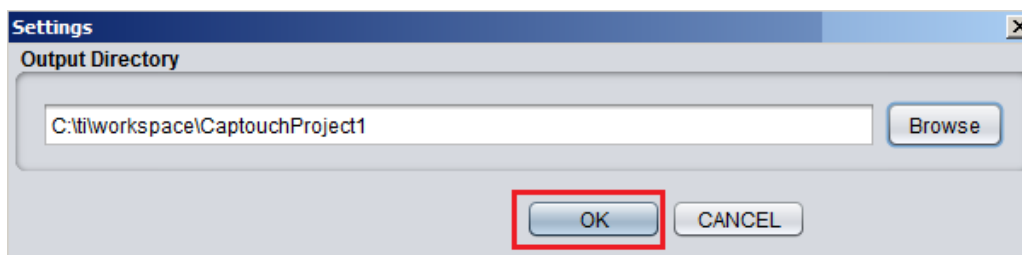


Рис.54 – Сохранение проекта

После этого в указанной папке будут созданы файлы с исходным кодом и проекты для сред CodeComposerStudio и IAREmbeddedWorkbench.

Затем переходим в CodeComposer studio и импортируем проект аналогично пункту 2.3 (Импорт проекта в CodeComposerStudio) и загрузить программу в контроллер аналогично пункту 2.4 (Загрузка программы в микроконтроллер).

После загрузки программы в контроллер запустите ее на выполнение с помощью кнопки Resume.

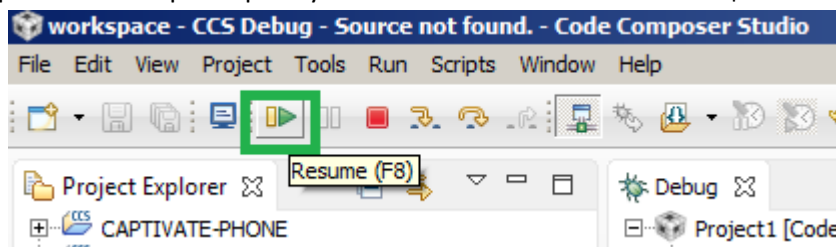


Рис.55 – Кнопка Resume в CodeComposerStudio

Отследите момент включения светодиода LED2 на плате контроллера. Всегда ли он включается только при прикосновении к нему?

### 3.3. Отображение параметров сенсора в реальном времени

Возвращаемся в среду Captivate Design Center и подключаемся к контроллеру для настройки сенсора в реальном времени (рис.56).

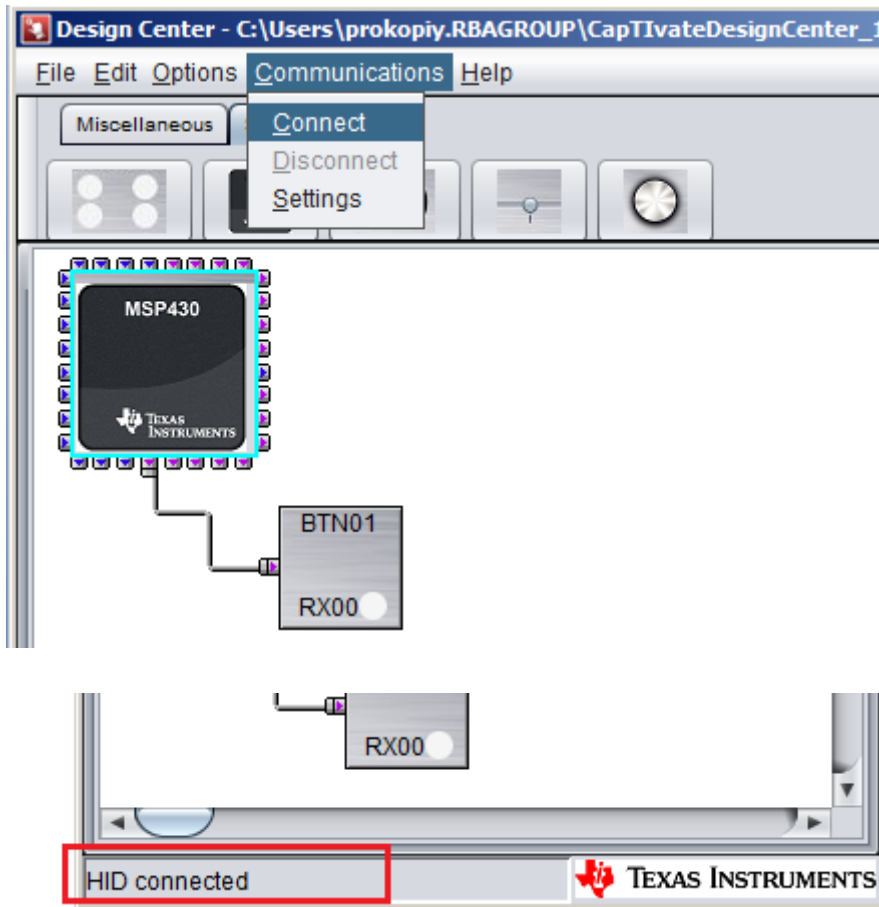


Рис.56 – Подключение к контроллеру в Captivate Design Center

(может потребоваться сброс микроконтроллера кнопкой MCU RESET на плате контроллера).

Войдем в настройки сенсора двойным нажатием по элементу BTN01 и перейдем во вкладку Channel Bar Chart.

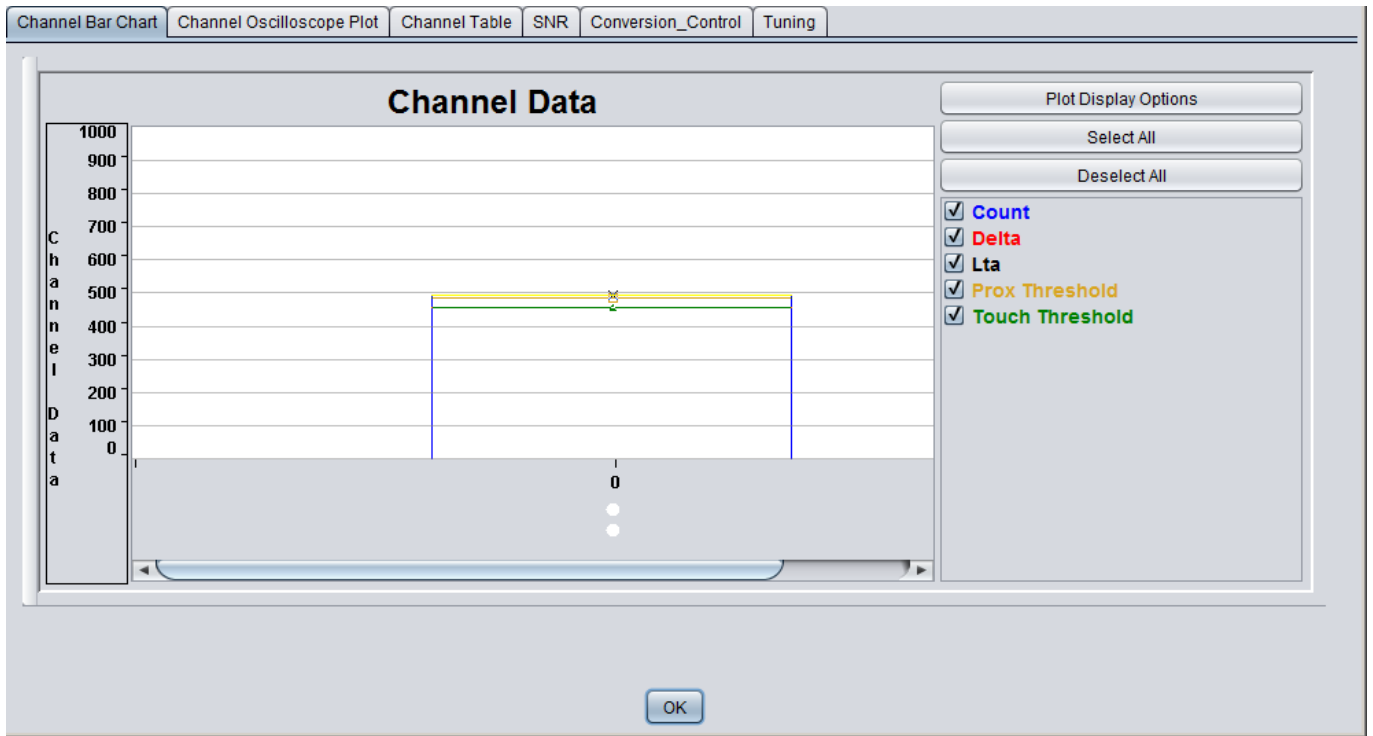
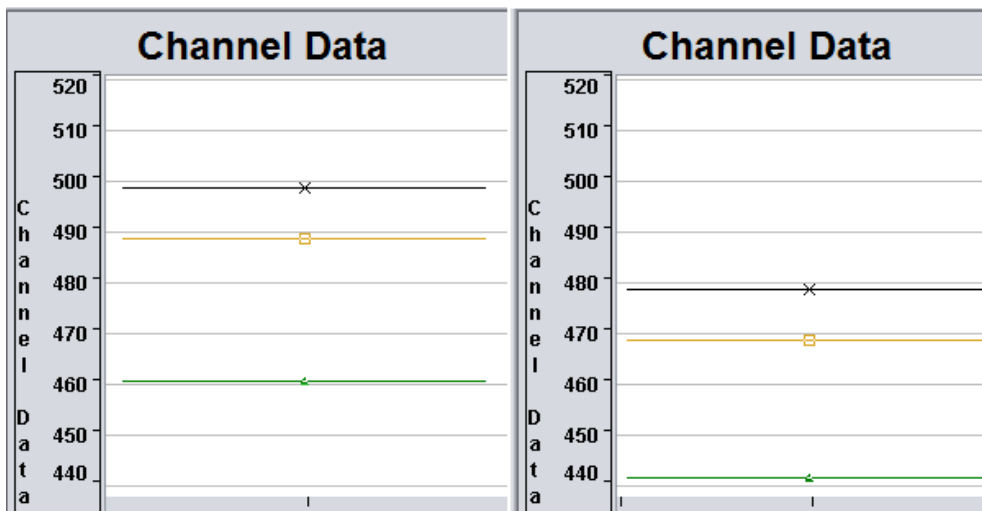


Рис.57 – Вкладка Channel Bar Chart в свойствах сенсора BTN01

Измените масштаб отображения столбца, как показано на рисунке 58. Сделать это можно с помощью выделения мышкой желаемой для увеличения области.

Попробуйте приближать палец к сенсору (клавиша 1) сбоку по поверхности платы с небольшой скоростью (примерно 1мм за 5 секунд). При этом засекуте начальное положение параметра LTA (Long Time Average) на графике ChannelBarChart. Обратите внимание, что пороги Proximity и Touch снижаются синхронно со значением LTA.



Proximity – желтый, Touch – зеленый

Рис.58 – Снижение уровня LTA(черный) синхронно с Proximity и Touch

Пороги Proximity и Touch(рис.59 - 2) настраиваются во вкладке Tuning(рис.59 - 1). Также можно настроить коэффициент фильтрации самого значения LTA (рис.59 - 3).

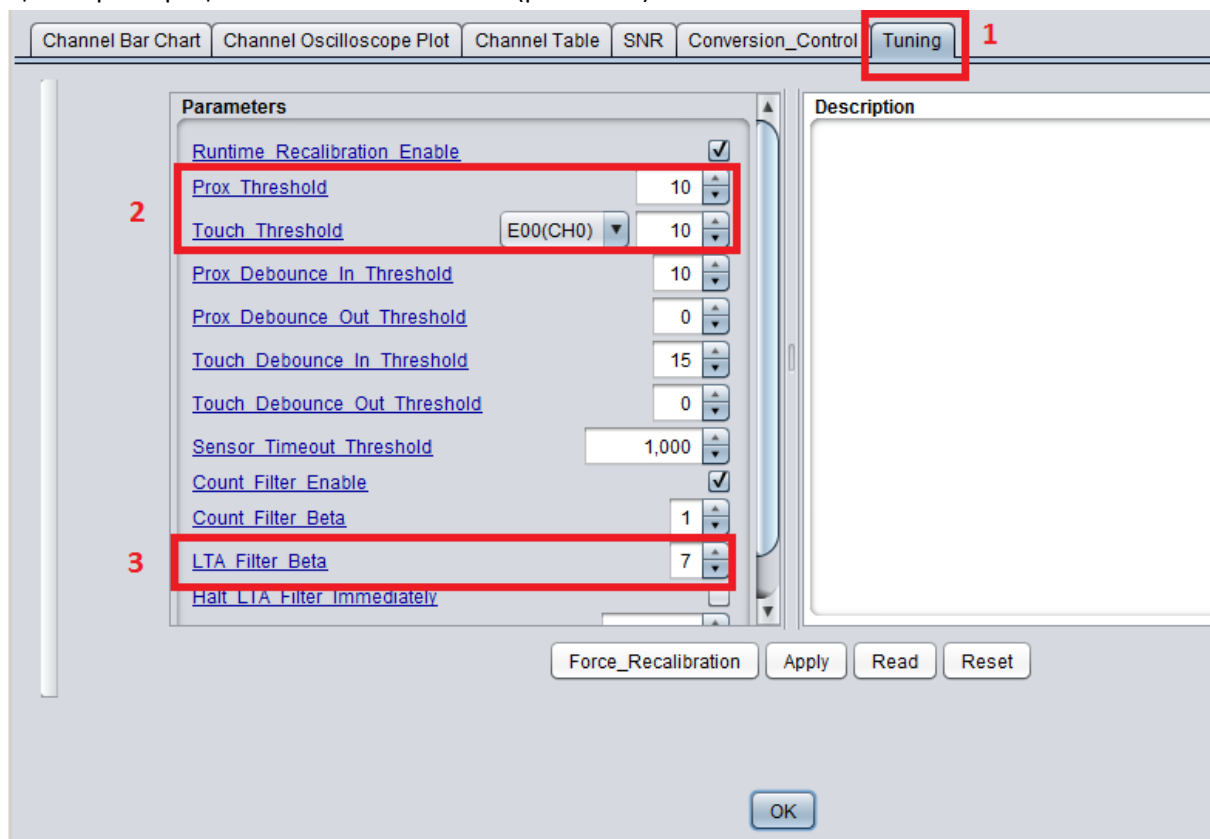


Рис.59 –Параметры настройки порогов срабатывания сенсора

Для получения точных числовых значений параметров перейдем к табличному виду во вкладке Channel Table.

При свободном сенсоре значение Count (количество переносов заряда) равно 486 (рис.60). При нажатом сенсоре – 416 (рис.61)(значения, полученные при выполнении данной работы могут отличаться от приведенных).

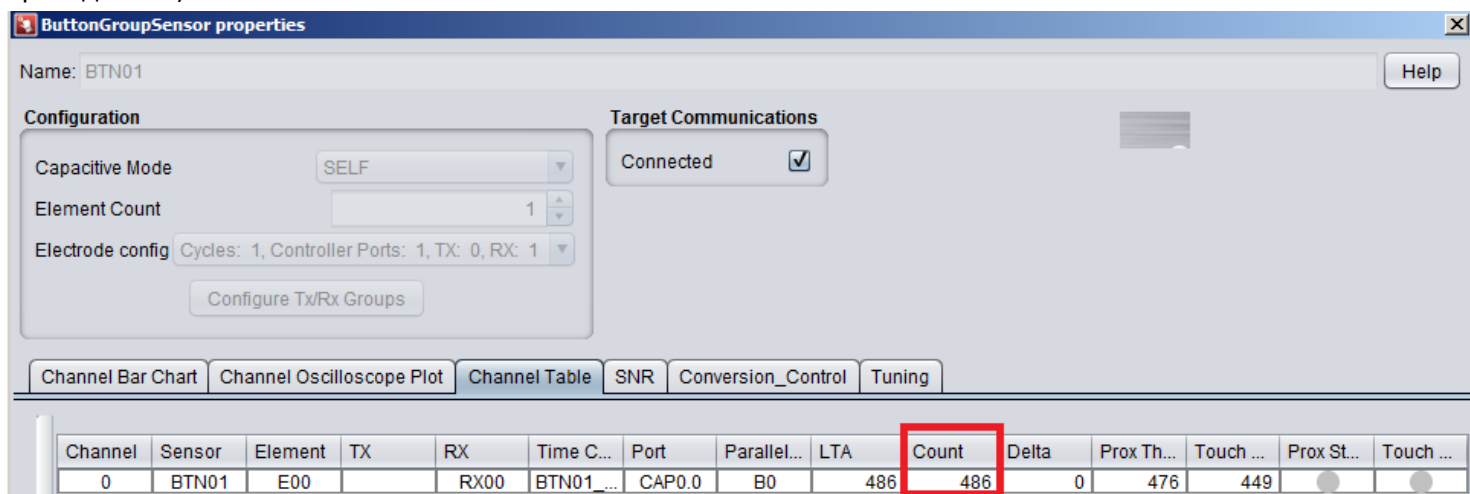


Рис.60 –Емкость сенсора без воздействия - Count<sub>BASE</sub> = 486

Channel	Sensor	Element	TX	RX	Time C...	Port	Parallel...	LTA	Count	Delta	Prox Th...	Touch ...	Prox St...	Touch ...
0	BTN01	E00		RX00	BTN01_...	CAP0.0	B0	486	416	70	476	449		

Рис.61 – Емкость сенсора при воздействии -  $Count_{FINGER} = 416$  ( $Count_{BASE} - 14\%$ )

### 3.4. Настройка параметров измерений

Настройка процесса измерения емкости сенсора выполняется в свойствах сенсора во вкладке Conversion Control (рис.62).

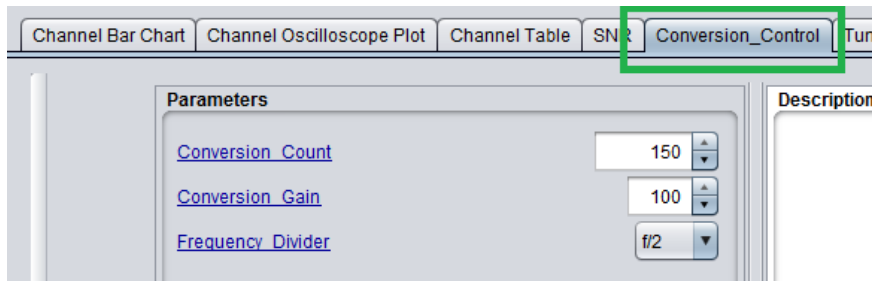


Рис.62 – Окно настройки параметров измерения

Для увеличения разрешения сенсора нужно увеличивать параметр Conversion Count.

Для увеличения чувствительности необходимо также увеличивать Conversion Count или снижать Conversion Gain.

Значение Conversion Count определяет время преобразования, т.к. указывает сколько переносов заряда необходимо выполнить для одного цикла измерения.

Значение Conversion Count должно быть больше или равно значению Conversion Gain.

После ввода новых значений Conversion Count и Conversion Gain в соответствующие строки необходимо их применить с помощью кнопки Apply (рис.63 - 2). Сброс на значения по умолчанию выполняется кнопкой Reset (рис.63 - 1)

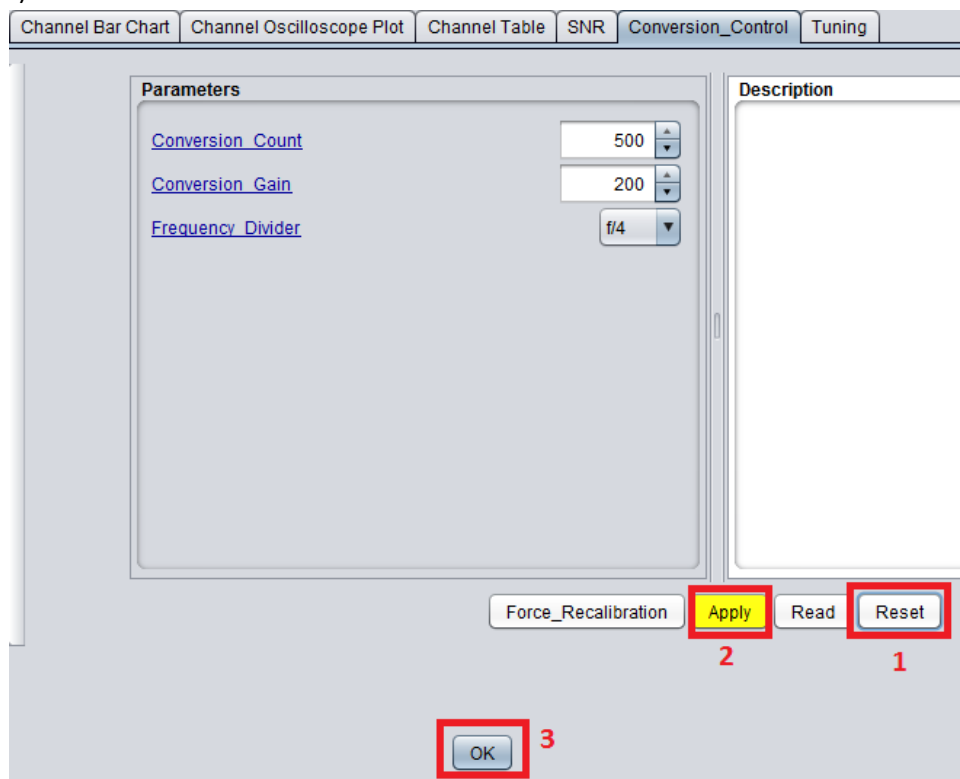


Рис.63 – Настройка параметров измерения

Проанализируйте, как изменятся значения Count для двух случаев:

1) Conversion Count = 100

Conversion Gain = 100

Count<sub>BASE</sub> = \_\_\_\_\_

Count<sub>FINGER</sub> = \_\_\_\_\_ (Count<sub>BASE</sub> - \_\_\_\_\_%)

2) Conversion Count = 1000

Conversion Gain = 100

Frequency Divider = f/128

Count<sub>BASE</sub> = \_\_\_\_\_

Count<sub>FINGER</sub> = \_\_\_\_\_ (Count<sub>BASE</sub> - \_\_\_\_\_%)

После тестов установите значения по умолчанию с помощью клавиши Reset

Убедитесь, что пороги Prox и Touch срабатывают корректно.

После изменений исходный код для микроконтроллера нужно обновить. Для этого заходим в настройки микроконтроллера и генерируем исходный код (рис.64)

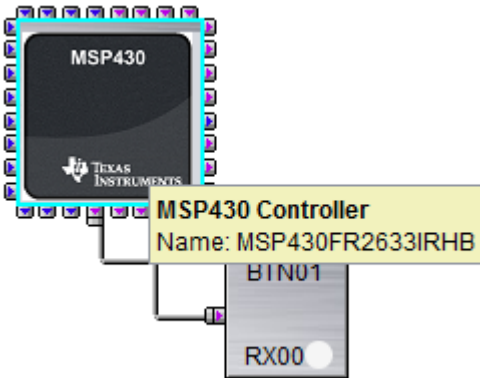


Рис.64 – Экспорт кода из CodeComposerStudio

Только теперь не создаем новый проект, а обновляем существующий:

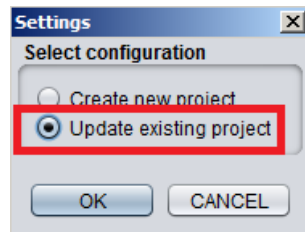


Рис.65 – Обновление проекта

И указываем рабочий каталог текущего проекта (рис.66).

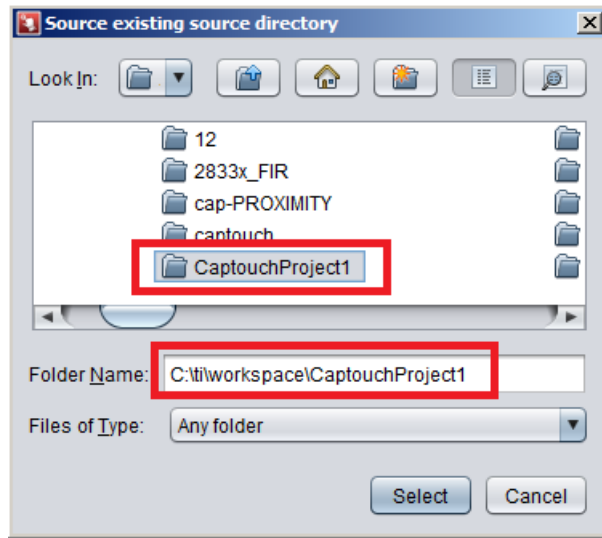


Рис.66 – Выбор каталога с проектом для обновления

Обновляем проект в CodeComposerStudio, выбрав пункт Refresh в контекстном меню проекта в окне Project Explorer (рис.67).

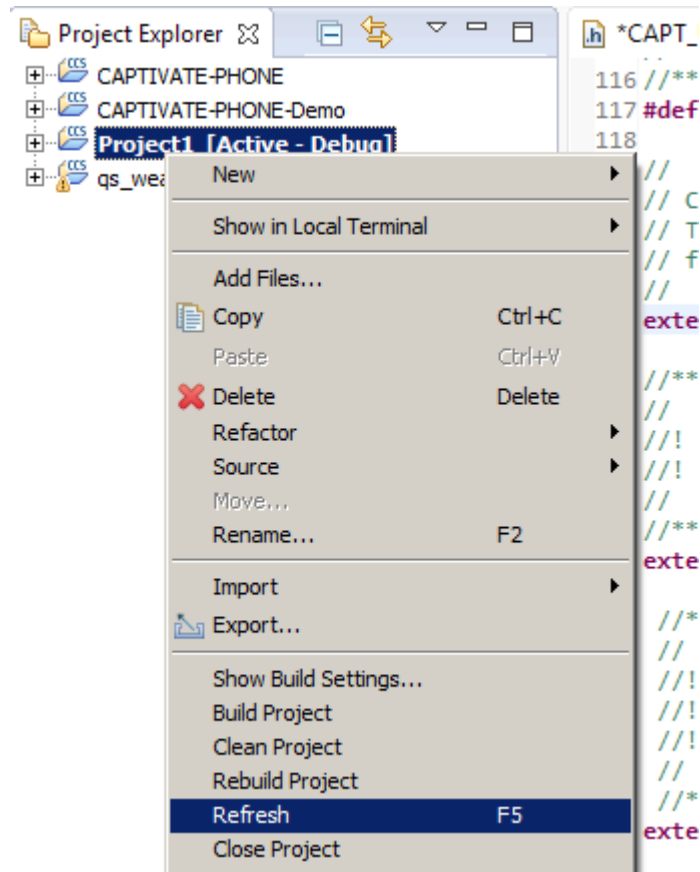


Рис.67 – Обновление проекта в CodeComposerStudio



### 3.5. Добавление callback-функции для кнопки

В файле `captivate_config/CAPT_UserConfig.h` содержатся названия сенсоров, сгенерированные средой Captivate Design Center.

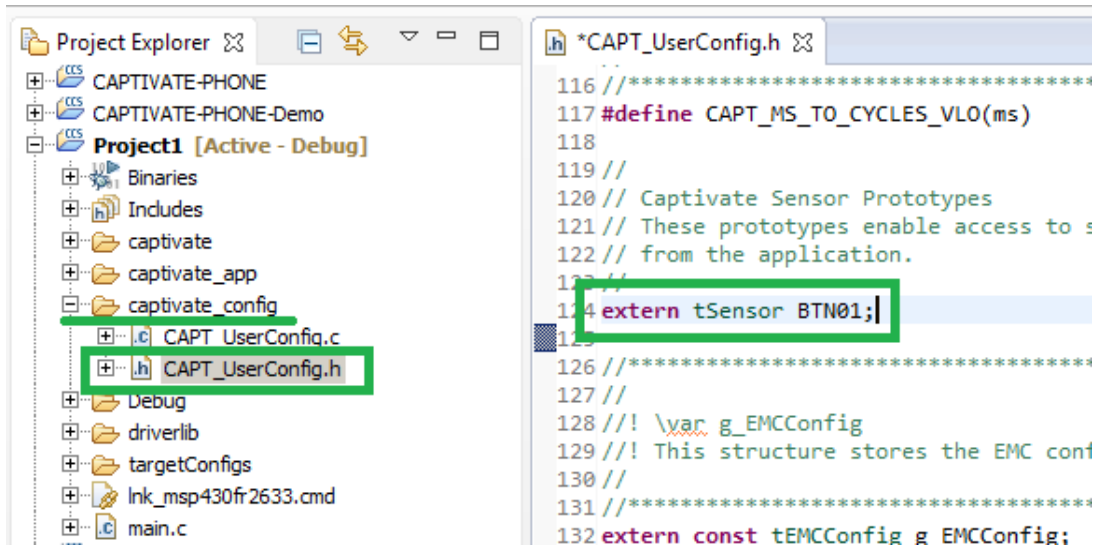


Рис.68 – Расположение файла CAPT\_UserConfig.h

Для обработки событий, связанных с этим датчиком нам необходимо создать callback-функцию. Callback-функция – это общепринятое название для функций, которые используются в приложении пользователя для совместной работы с какой-либо библиотекой. Такие функции вызываются в приложении пользователя при возникновении прерывания или выполнении определенного условия в библиотеке.

В библиотеке Captivate callback-функция вызывается сразу после окончания сканирования и обработки данных от очередного сенсора при условии обнаружения приближения к нему или нажатия. Внутри callback-функции пользователь может обратиться к структуре данных сенсора для определения его текущего состояния. В Captivate callback-функция вызывается только в том случае, если она объявлена в коде пользователя и привязана к конкретному сенсору. После вызова callback-функции выполняется передача данных в приложение Captivate Design Center (рис.69) (если в приложении был включен интерфейс передачи данных для отслеживания параметров сенсоров в реальном времени).



Рис.69 – Вызов callback-функции после измерения сенсора

Добавляем callback-функцию в файле `main.c` перед функцией `main`:

```
void sensor_callback(tSensor* pSensor)  
{  
    if((pSensor->bSensorTouch == true) && (pSensor-> bSensorPrevTouch == false))  
    {  
        LED2_ON;  
    }  
    else if((pSensor->bSensorTouch == false) && (pSensor-> bSensorPrevTouch == true))  
    {  
        LED2_OFF;  
    }  
}
```

И изменяем часть кода в функции main для объявления callback-функции и ее привязки к элементу BTN01:

```
....  
// Start the CapTivate application  
//  
MAP_CAPT_registerCallback(&BTN01, &sensor_callback);  
CAPT_appStart();  
....  
/*  
LED1_ON;  
if(CAPT_appHandler()==true)  
    LED2_ON;  
else  
    LED2_OFF;  
    LED1_OFF;  
*/  
CAPT_appHandler();
```

Компилируем, загружаем приложение в контроллер и запускаем на выполнение и проверяем его работу. Как теперь работает LED2? Что изменилось в его работе?

## 4. Реализация слайдера

### 4.1. Добавление слайдера в проект

Добавляем в рабочее поле новый элемент из панели инструментов – слайдер (рис. 70).

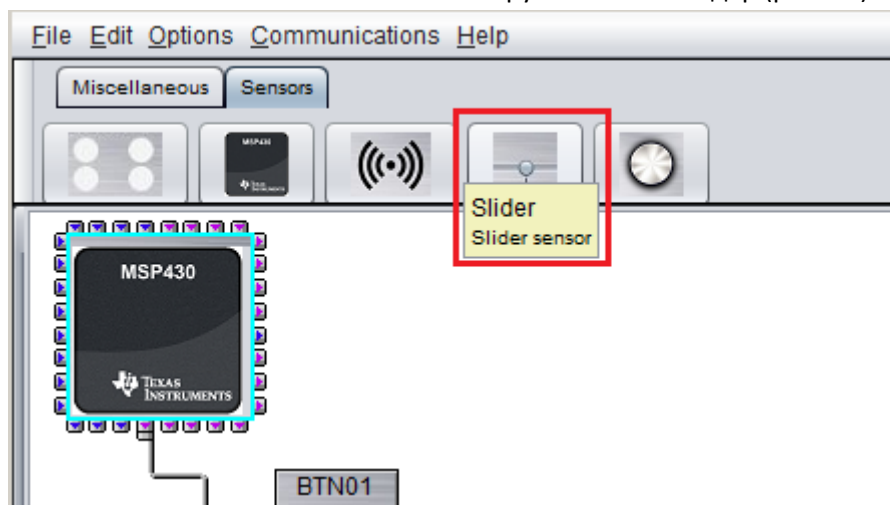


Рис.70 – Символ слайдера на панели инструментов

В свойствах слайдера задаем ему следующие параметры (рис.71):

- 1) имя – SLD00
- 2) режим измерения емкости - SELF
- 3) количество элементов, из которых состоит слайдер - 4

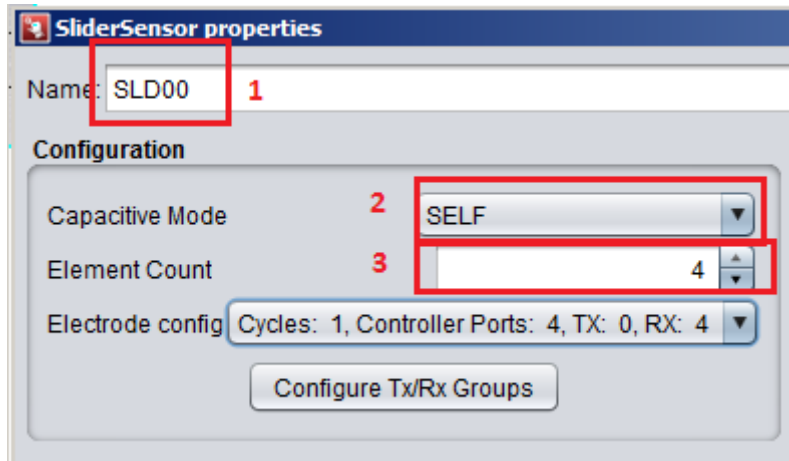


Рис.71 – Параметры слайдера

Из [технической документации](#) к отладочному набору [MSP-CAPT-FR2633](#) мы можем определить, что элементы слайдера подключены к выводам CAP0.2, CAP1.2, CAP2.2, CAP3.2 (рис.72, рис.73).

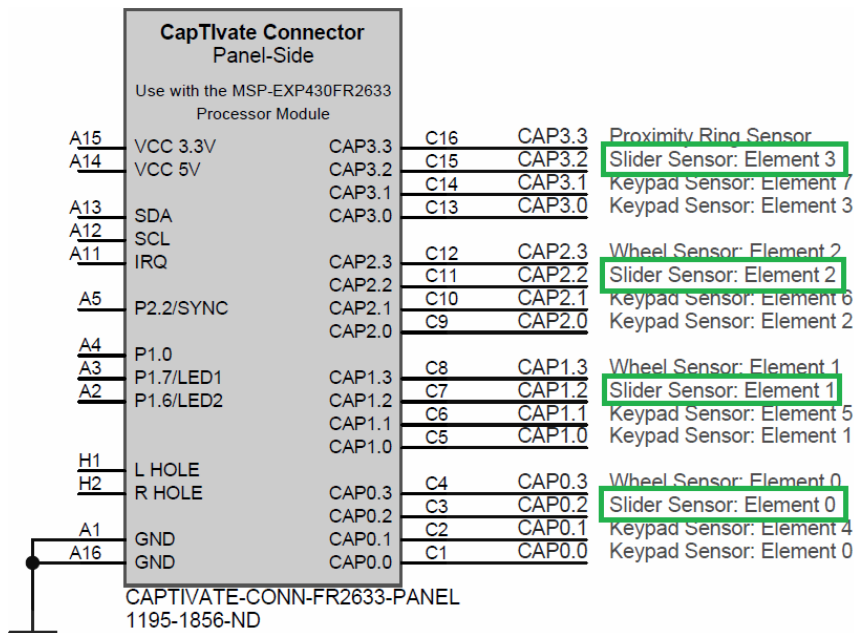


Рис.72 - Фрагмент принципиальной схемы на плату CAPTIVATE-FR2633

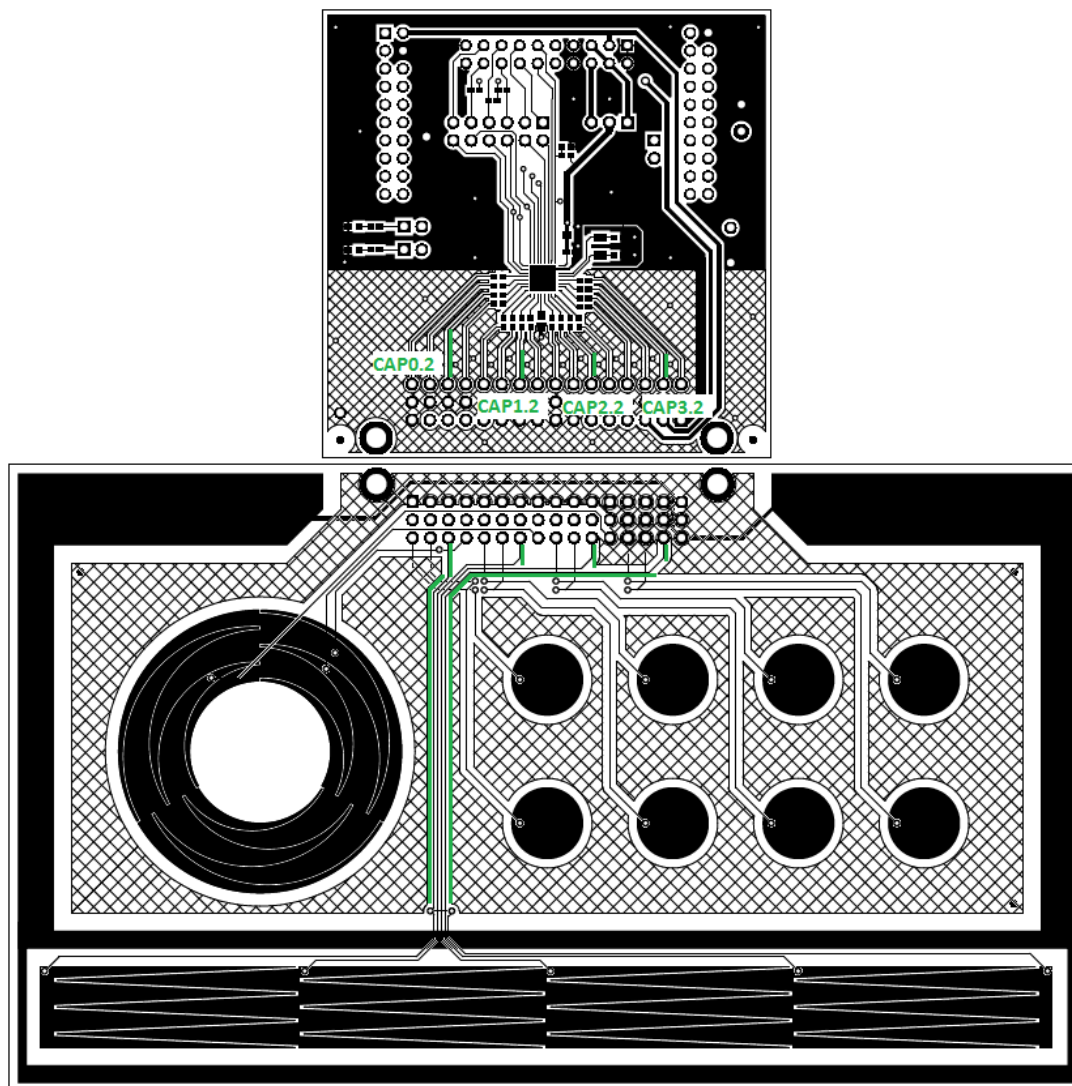


Рис.73 - Топология печатных плат CAPTIVATE-FR2633 и CAPTIVATE-BSWP

В настройках контроллера необходимо перевести выводы в режим измерения SELF only, и подключить выводы контроллера к элементам слайдера. Для корректной работы слайдера рекомендуется соблюдать порядок подключения, приведенный на рисунке 74.

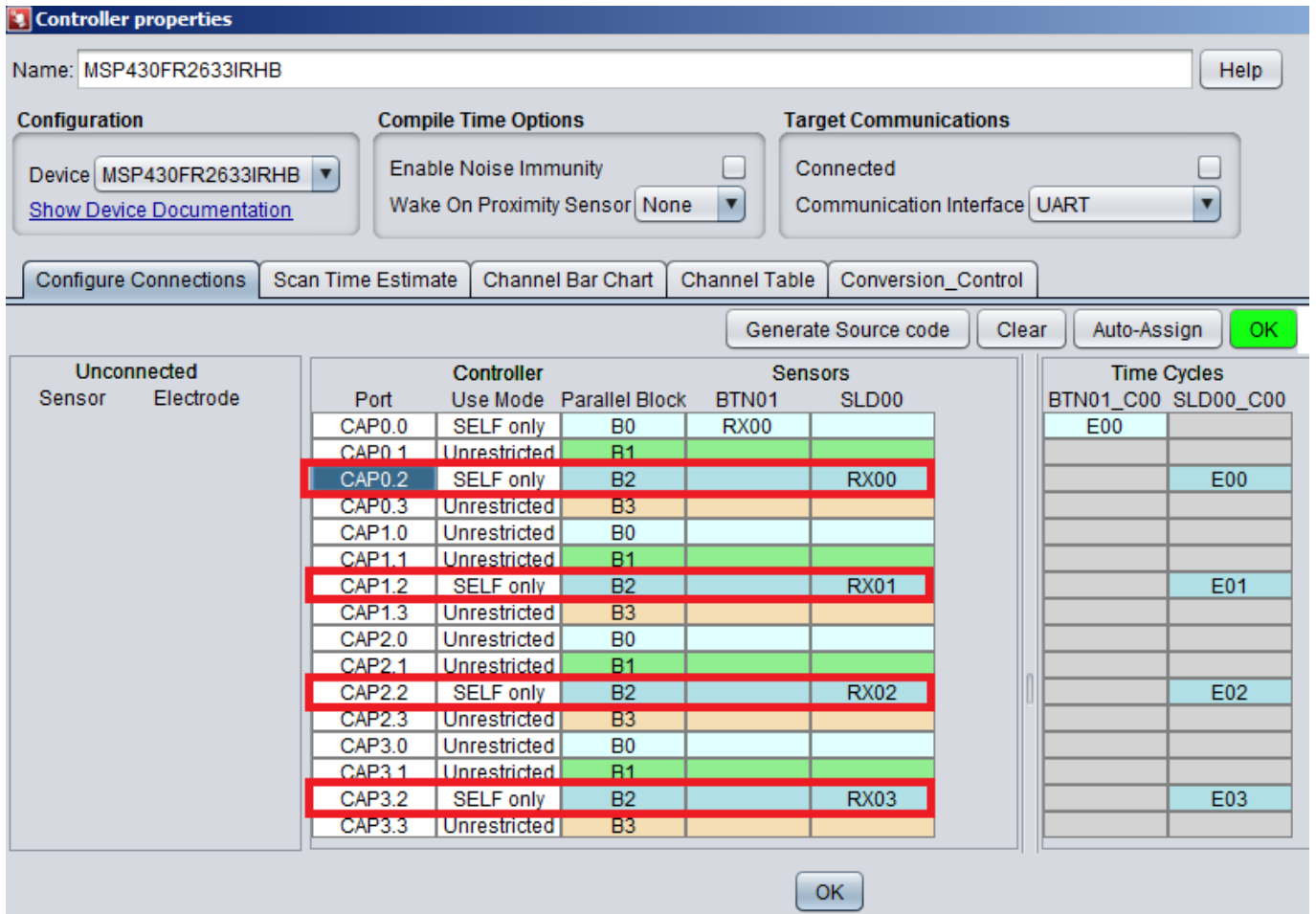


Рис.74 – Подключение элементов слайдера к выводам контроллера

Генерируем код с обновлением предыдущего проекта.

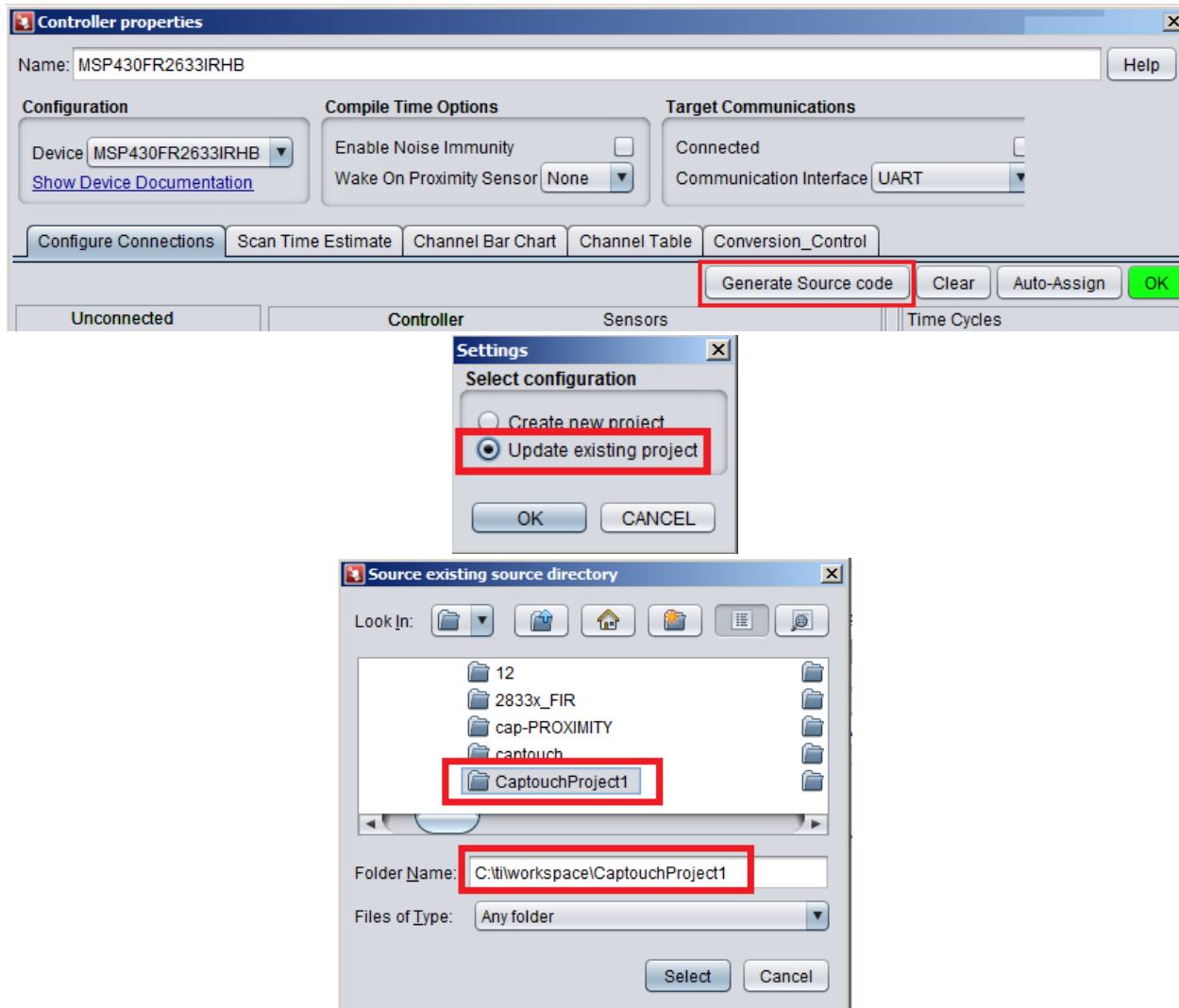


Рис.75 – Экспорт кода с обновлением проекта

Обновляем проект в CodeComposerStudio, компилируем и загружаем в контроллер. Проверяем работу слайдера в Captivate Design Center.

Корректно ли работает слайдер?

#### 4.2. Калибровка слайдера

На краях слайдера значения не соответствуют минимальному и максимальному значению шкалы.



Крайнее левое положение

Крайнее правое положение

Рис.76 – Крайние положения слайдера

Для корректной работы слайдер должен быть откалиброван.

Калибровка проводится путем добавления значений в поля Lower Trim и Upper Trim (рис.77 - 1) в настройках слайдера во вкладке Tuning. После введения нового значения в любое из полей необходимо использовать кнопку Update для загрузки его в контроллер в реальном времени.

Значения для полей Lower Trim и Upper Trim подбираются опытным путем.

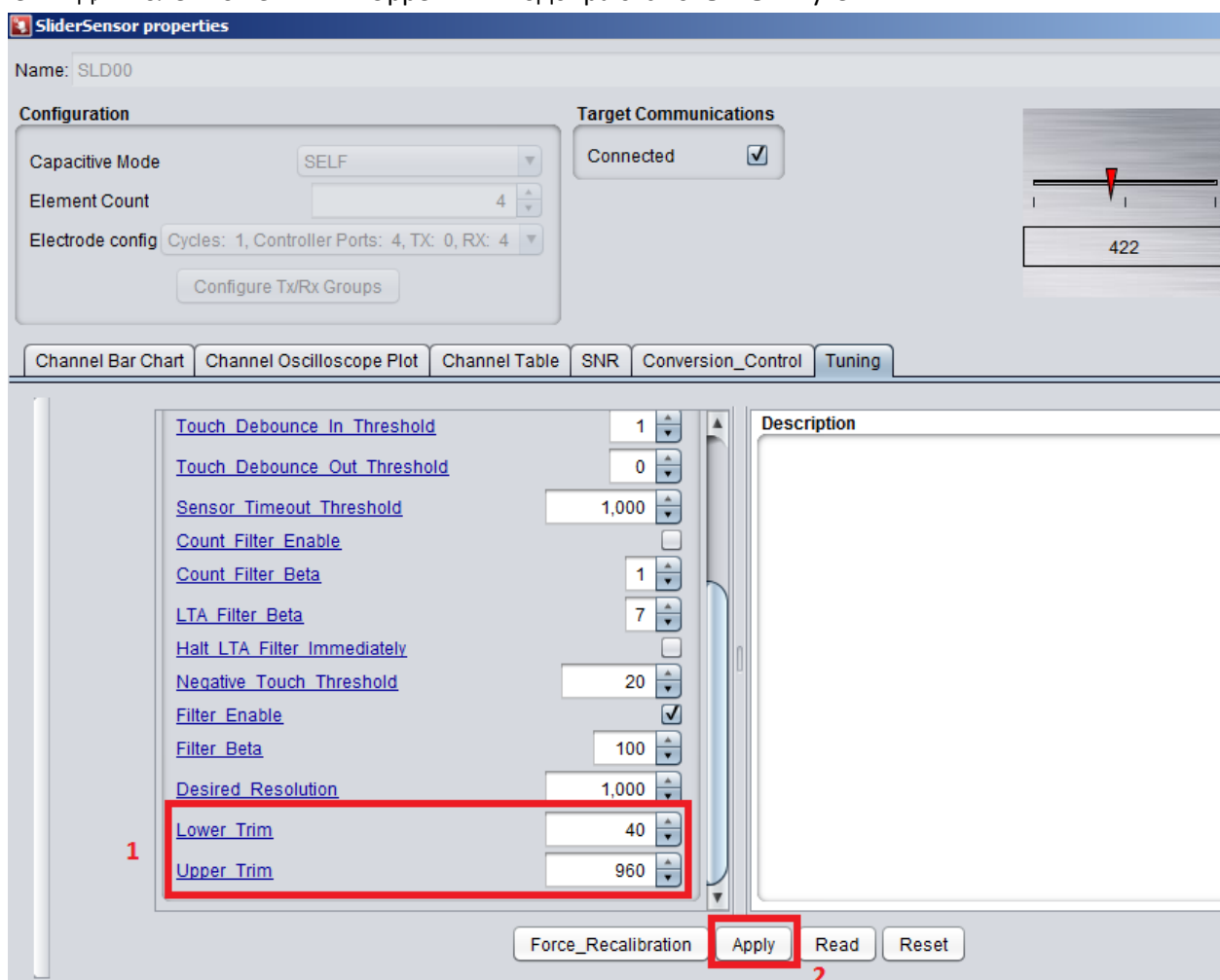


Рис.77 – Вкладка Tuning в свойствах сенсора

### 4.3. Добавление callback-функции для слайдера

Аналогично добавлению обработчика для кнопки добавляем callback-функцию для слайдера.

Добавляем функцию в файле main.c перед функцией main:

```
void slider_callback(tSensor* pSensor)
{
    tSliderSensorParams *pSliderParams;

    if(pSensor->bSensorTouch == true)
    {
        pSliderParams = (tSliderSensorParams*)(pSensor->pSensorParams);
        if(pSliderParams->SliderPosition.ui16Natural >500)
        {
            LED1_ON;
        }
    }
    else
```



```
        {  
            LED1_OFF;  
        }  
    }  
}
```

И изменяем часть кода в функции main, регистрируя callback-функцию и связывая ее с элементом SLD00 :

```
....  
  
//  
// Start the CapTivate application  
//  
MAP_CAPT_registerCallback(&BTN01, &sensor_callback);  
MAP_CAPT_registerCallback(&SLD00, &slider_callback);  
CAPT_appStart();
```

Компилируем и загружаем приложение в контроллер. Запускаем программу на выполнение. Обратите внимание на работу LED1. Как его включение зависит от положения слайдера?

## Заключение

Выполнения данных работ достаточно для получения базовых практических навыков разработки емкостных сенсоров на микроконтроллерах [MSP430FR2532](#) / [MSP430FR2533](#) / [MSP430FR2632](#) / [MSP430FR2633](#) .

Более подробная информация о технологии CapTivate и более продвинутых параметрах настройки сенсоров доступна в [центре знаний CapTivate](#).

Ваши замечания и предложения присылайте на [ti@compel.ru](mailto:ti@compel.ru).

Руководство подготовил Вячеслав Прокопий. Компэл. Февраль, 2017г.