Securing IoT Devices using Arm TrustZone®

Class 4: Designing and Debugging a Secure Boot Solution

November 29, 2018

Jacob Beningo

DesignNews

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS

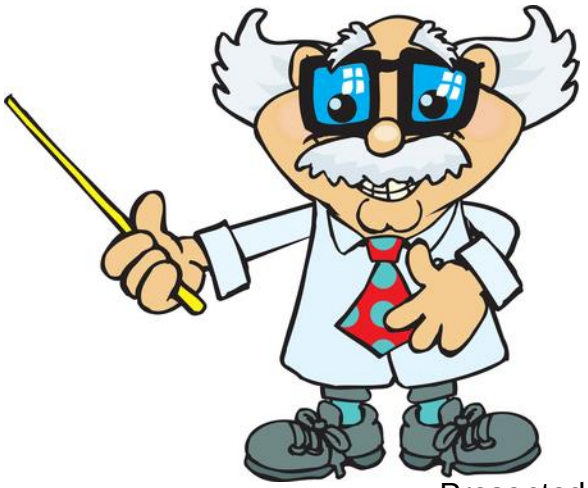
Course Overview

Topics:

- Understanding Embedded System Security
- Introduction to Arm TrustZone®
- Creating your First TrustZone Application
- **Designing and Debugging a Secure Boot Solution**
- Securing a RTOS Application with TrustZone

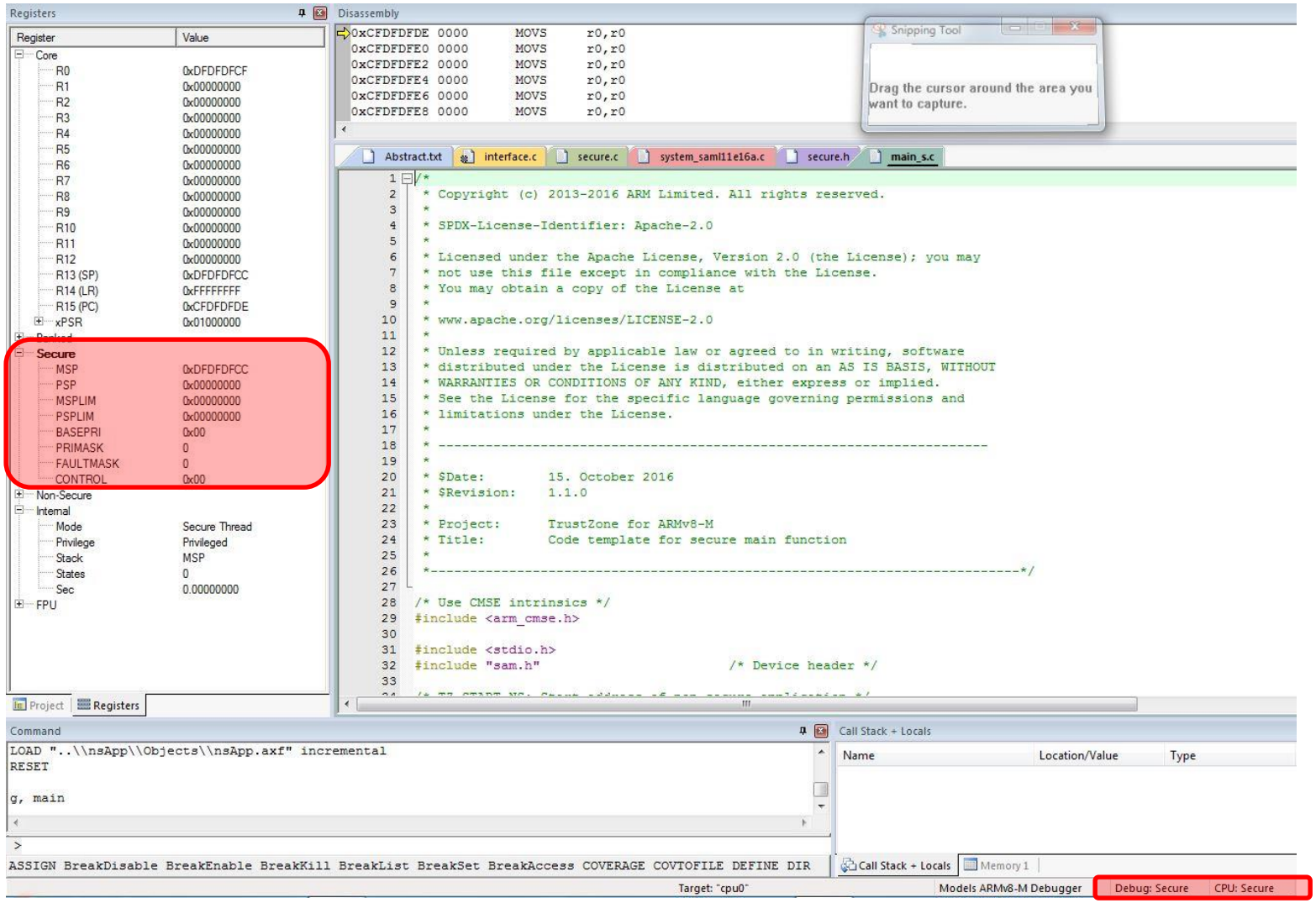
Session Overview

- Continuing FVP Debugging
- Chain of Trust
- Demo Project Setup
- Trusted Execution Environment



Presented by:

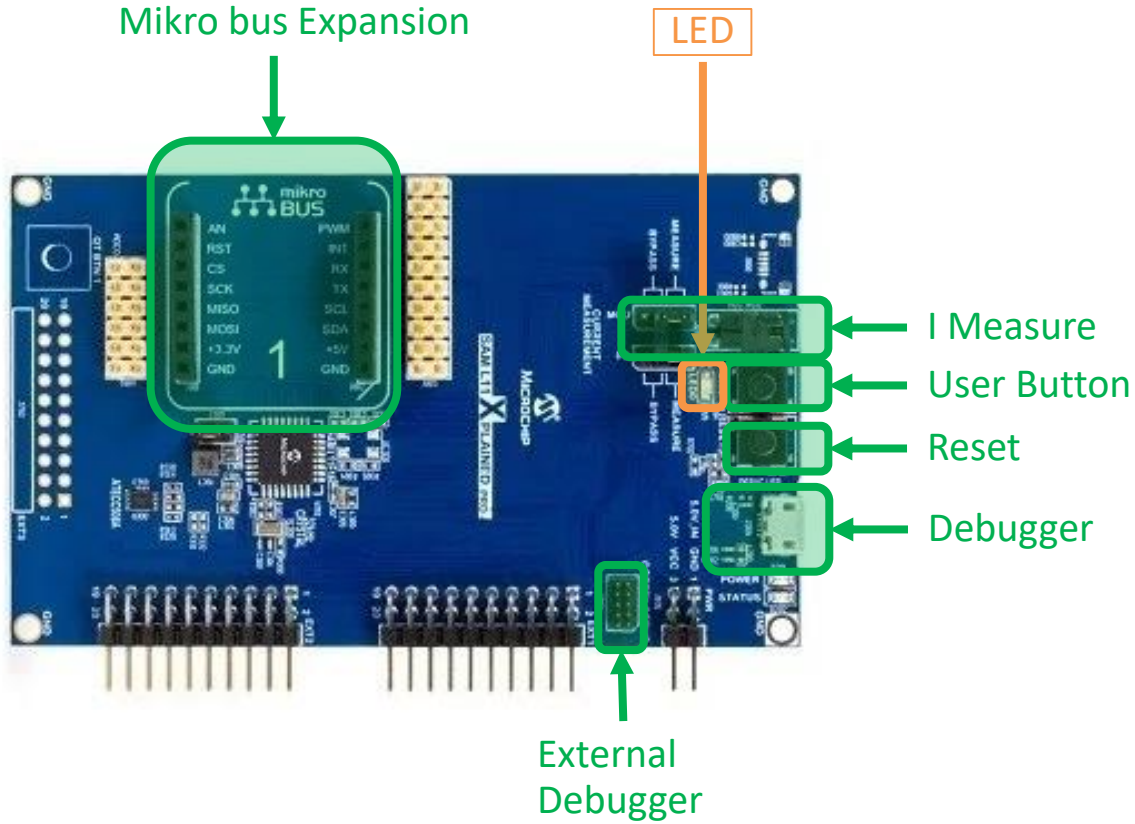
Debugging


Registers
Register
Value
Core
R0 0xDFDFDFCF
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0xDFDFDFCC
R14 (LR) 0xFFFFFFFF
R15 (PC) 0xCFDFDFDE
xPSR 0x01000000
Secure
MSP 0xDFDFDFCC
PSP 0x00000000
MSPLIM 0x00000000
PSPLIM 0x00000000
BASEPRI 0x00
PRIMASK 0
FAULTMASK 0
CONTROL 0x00
Non-Secure
Internal
Mode Secure Thread
Privilege Privileged
Stack MSP
States 0
Sec 0.00000000
FPU
Project
Registers
Disassembly
0xCFDFDFDE 0000 MOVS r0,r0
0xCFDFDFE0 0000 MOVS r0,r0
0xCFDFDFE2 0000 MOVS r0,r0
0xCFDFDFE4 0000 MOVS r0,r0
0xCFDFDFE6 0000 MOVS r0,r0
0xCFDFDFE8 0000 MOVS r0,r0
Snipping Tool
Drag the cursor around the area you want to capture.
Abstract.txt
interface.c
secure.c
system_saml11e16a.c
secure.h
main_s.c
/*
* Copyright (c) 2013-2016 ARM Limited. All rights reserved.
*
* SPDX-License-Identifier: Apache-2.0
*
* Licensed under the Apache License, Version 2.0 (the License); you may
* not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an AS IS BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*
* --
*
* $Date: 15. October 2016
* $Revision: 1.1.0
*
* Project: TrustZone for ARMv8-M
* Title: Code template for secure main function
*
---/
/* Use CMSE intrinsics */
#include <arm_cmse.h>
#include <stdio.h>
#include "sam.h" /* Device header */
Command
LOAD "..\\nsApp\\Objects\\nsApp.axf" incremental
RESET
g, main
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR
Call Stack + Locals
Name
Location/Value
Type
Call Stack + Locals
Memory 1
Target: "cpu0"
Models ARMv8-M Debugger
Debug: Secure
CPU: Secure

Chain of Trust

Image Source:
http://www.rgbstock.com/cache1nuP23/users/c/co/cobrasoft/300/meZ96je.jpg

Demo Project Setup



Demo Project Setup

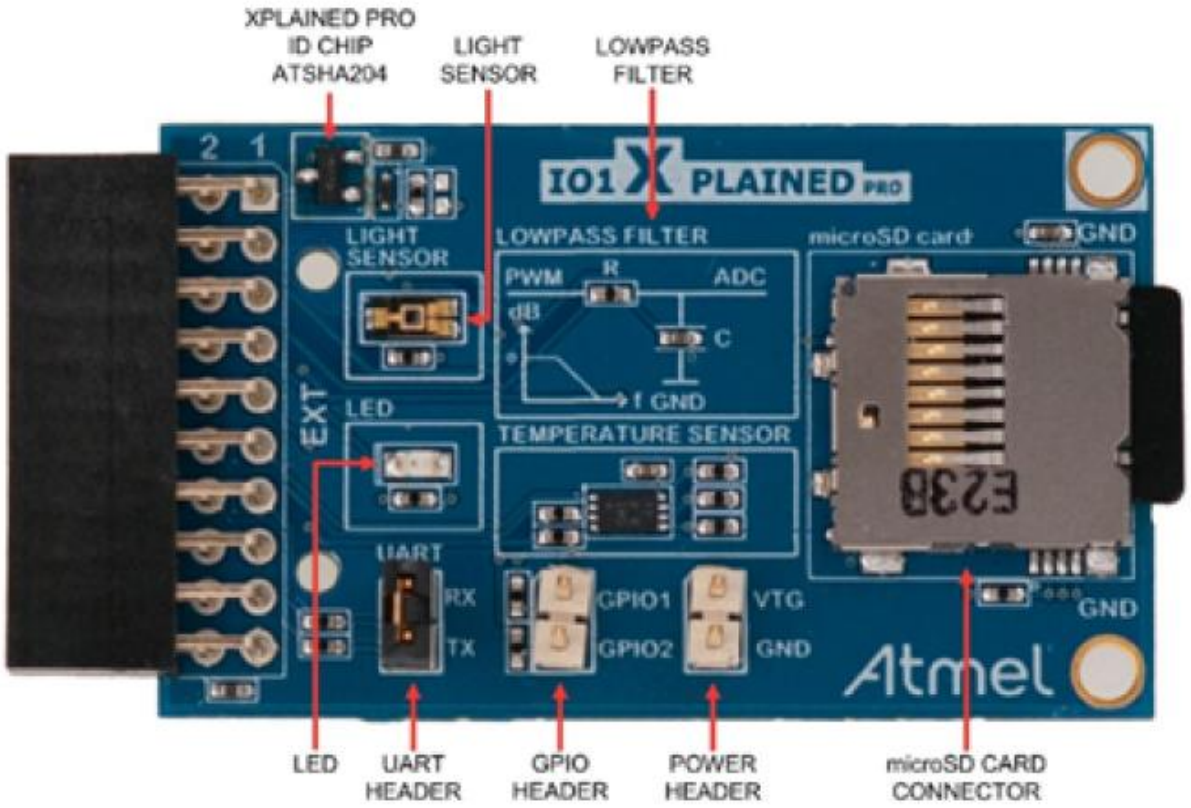


Image Source: http://microchip.com

Trusted Execution Environment

Demonstration Software:

- Customer A IP protection (Temp. sensor library)
- Customer A API's to Customer B
- Customer B App development (Temp. sensor application)

Trusted Execution Environment

Documents

- SAML10L11 Xplained Pro Kit User Guide
- Data Gateway Interface User's Guide
- Atmel Embedded Debugger User Guide
- Low Power Weather Station Demo_SAML10_L11
- Low Power SleepWalking Demo_SAML10_L11
- SAML10_L11 Xplained Pro Board CN
- Trusted Execution Environment Demo_L11
- SAM L11 Xplained Pro Design Documentation

AppNotes

- AN_2698 - Secure Bootloader_SAML11
- AN2699 - UART Bootloader_SAML10L11
- AN5365 - SAM L11 Security Reference Guide Application Note
- AN2722 - Getting Started With SAM L10 / L11 Xplained Pro Application Note

MICROCHIP

Trusted Execution Environment

SAM L11 Trusted Execution Environment Demonstration

Introduction

This document describes the SAM L11 Low-power TrustZone demonstration. It covers following demonstration application aspects:

- Application Requirement
- How to build and load the application on a SAM L11 target device
- Technical Solution description, and key SAM L11 features used to build the demonstration

Software requirements:

- Atmel Studio 7 (build 1912 or later)
- SAM L11 DFP version 1.0.81
- Tera Term : https://osdn.net/projects/ttssh2/releases/

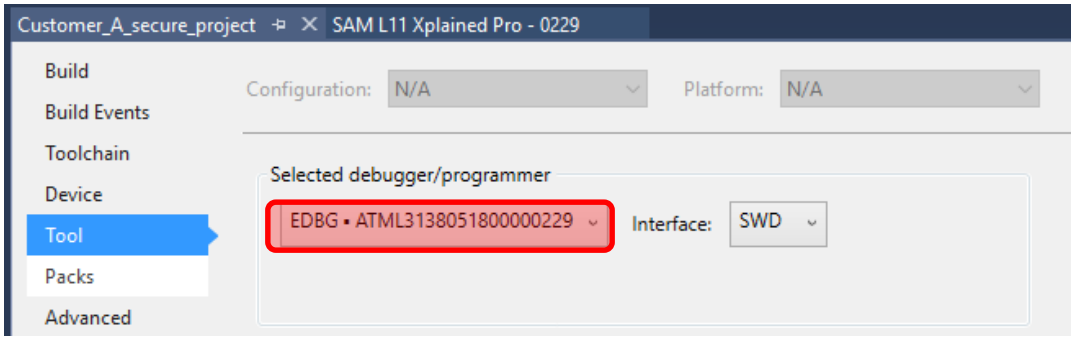
Hardware requirements:

- 1 x Microchip SAM L11 Xplained Pro
- 1 x I/O1 Xplained Pro

Trusted Execution Environment

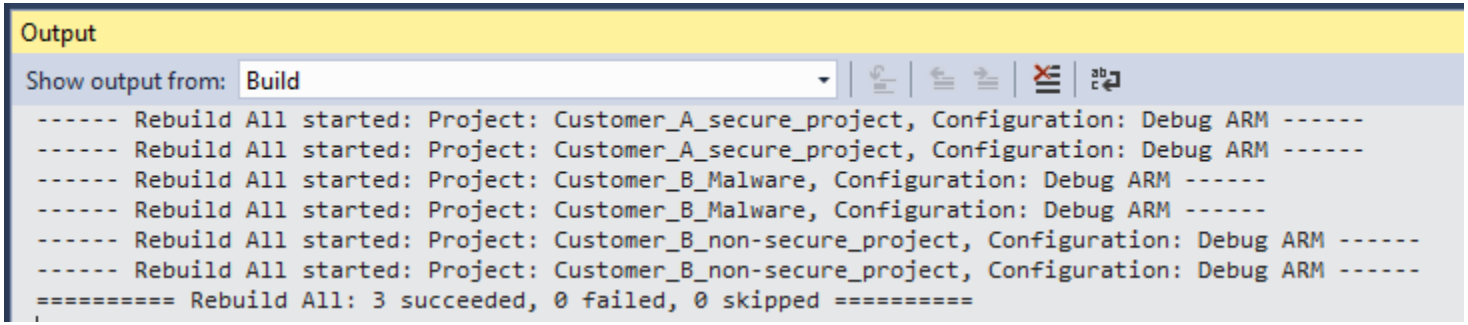
Run the demonstration:

- Open the project / solution
- Select the debugger



- Rebuild the project
- Deploy it on the target board

Trusted Execution Environment



Open a terminal that is set to:

- 115200 bps

Press the board reset button

Trusted Execution Environment

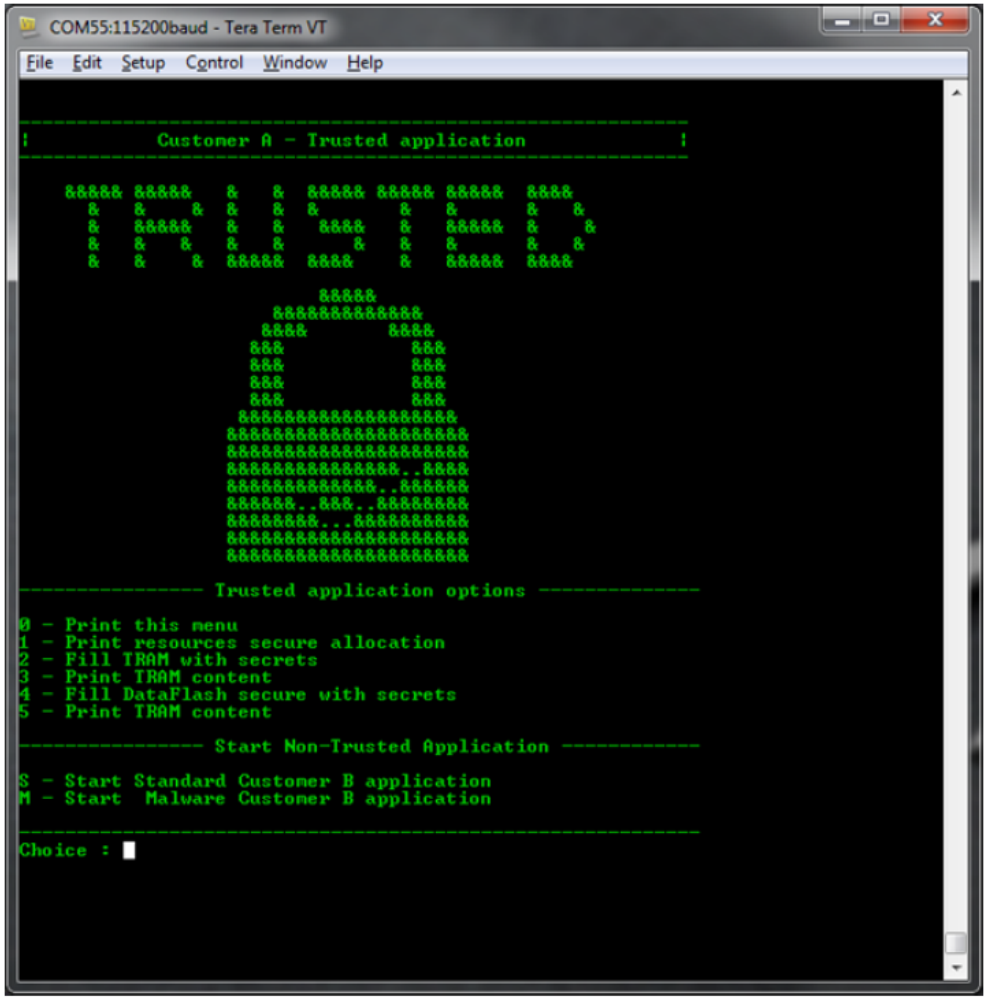


Image Source: http://microchip.com

Trusted Execution Environment

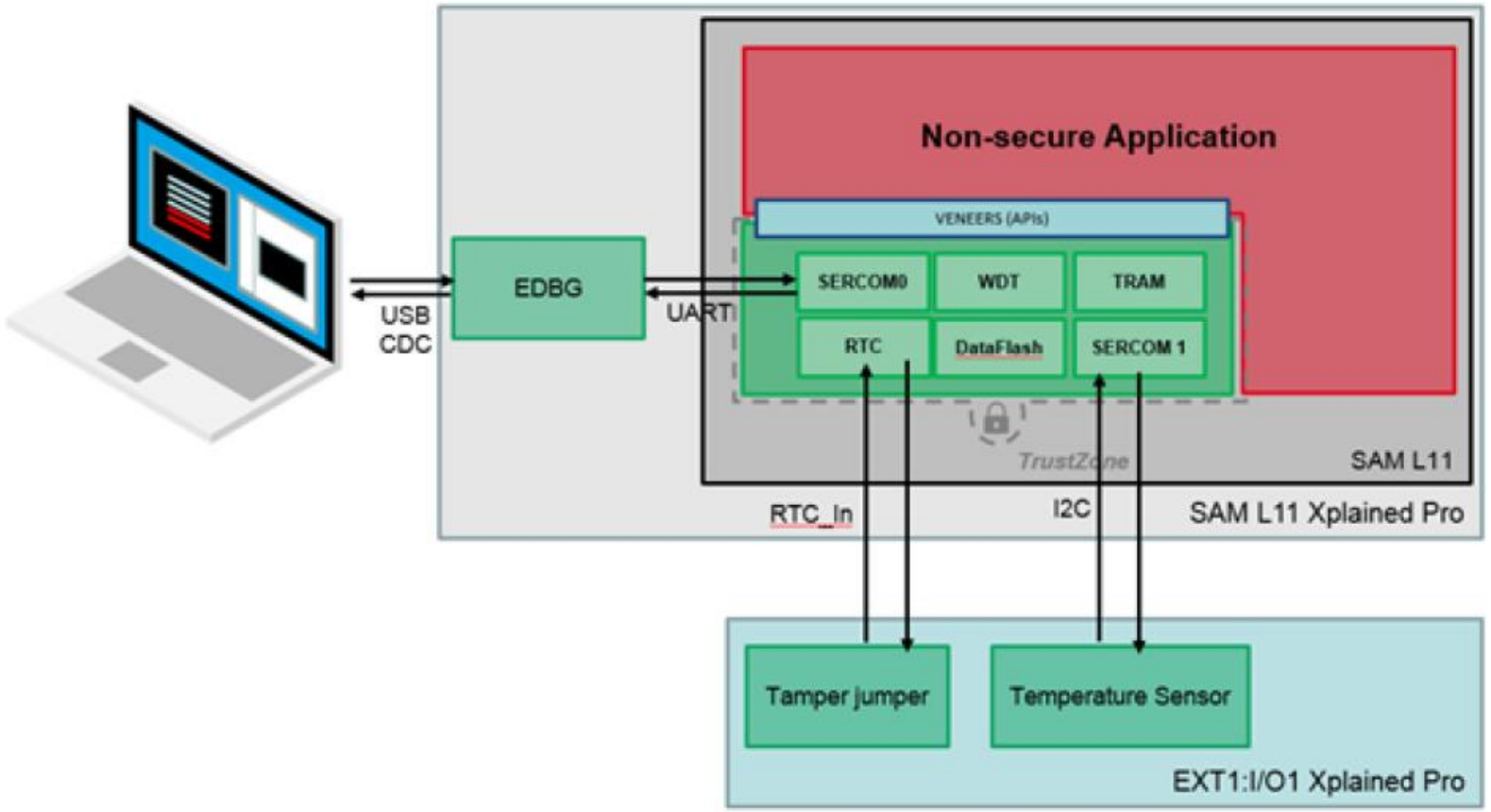
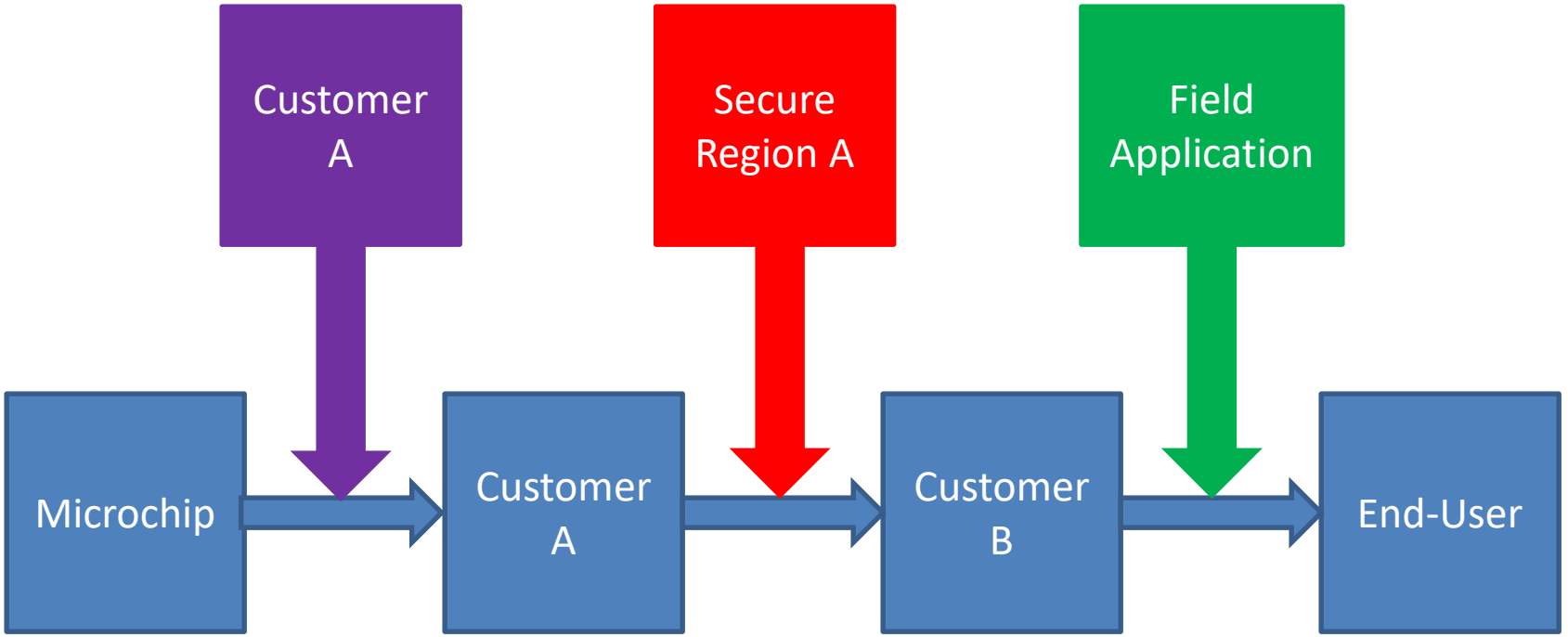


Image Source: http://microchip.com

Trusted Execution Environment



Trusted Execution Environment

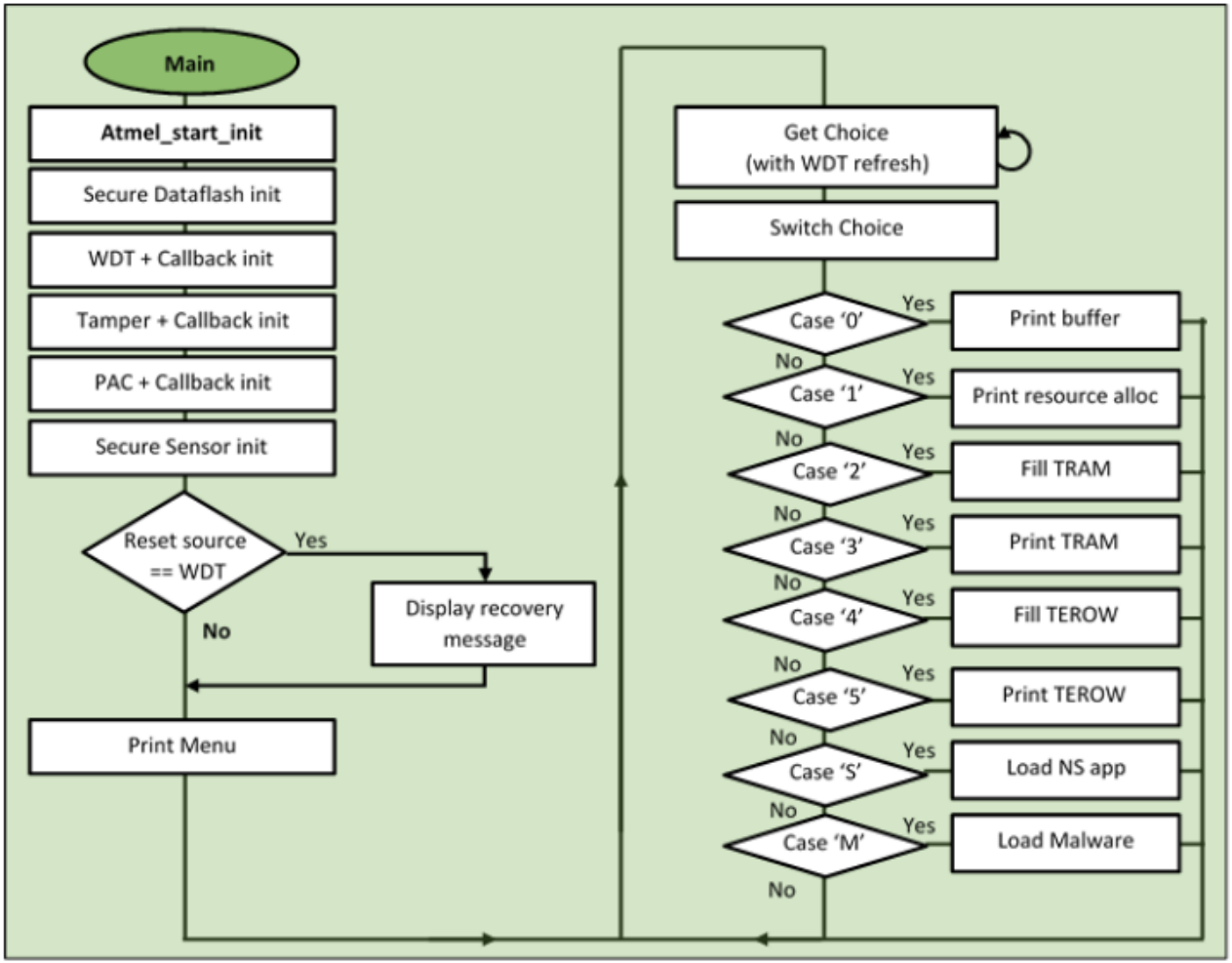
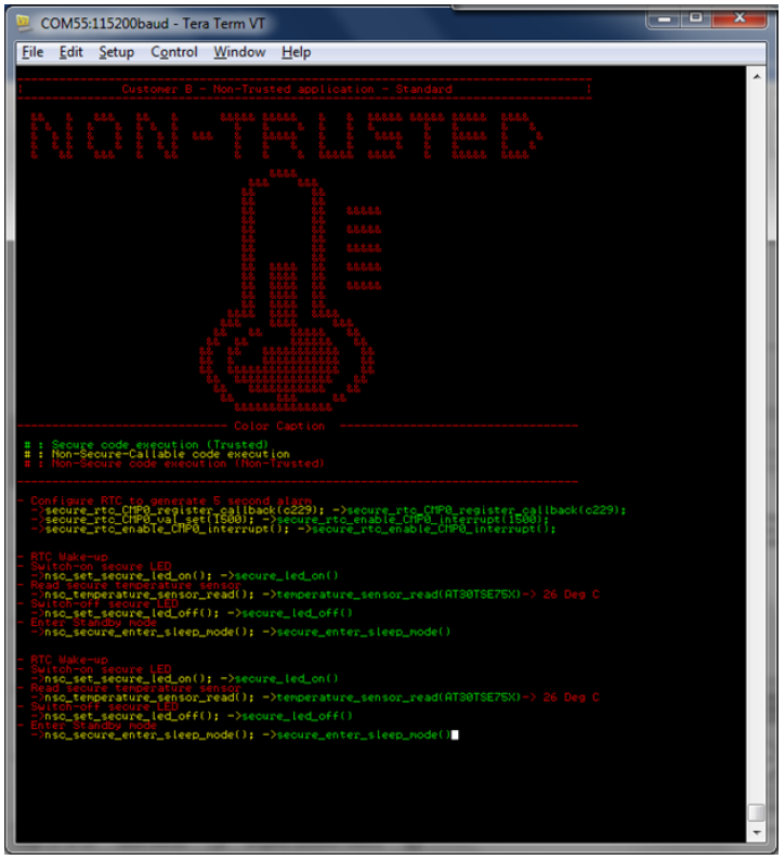


Image Source: http://microchip.com

Trusted Execution Environment



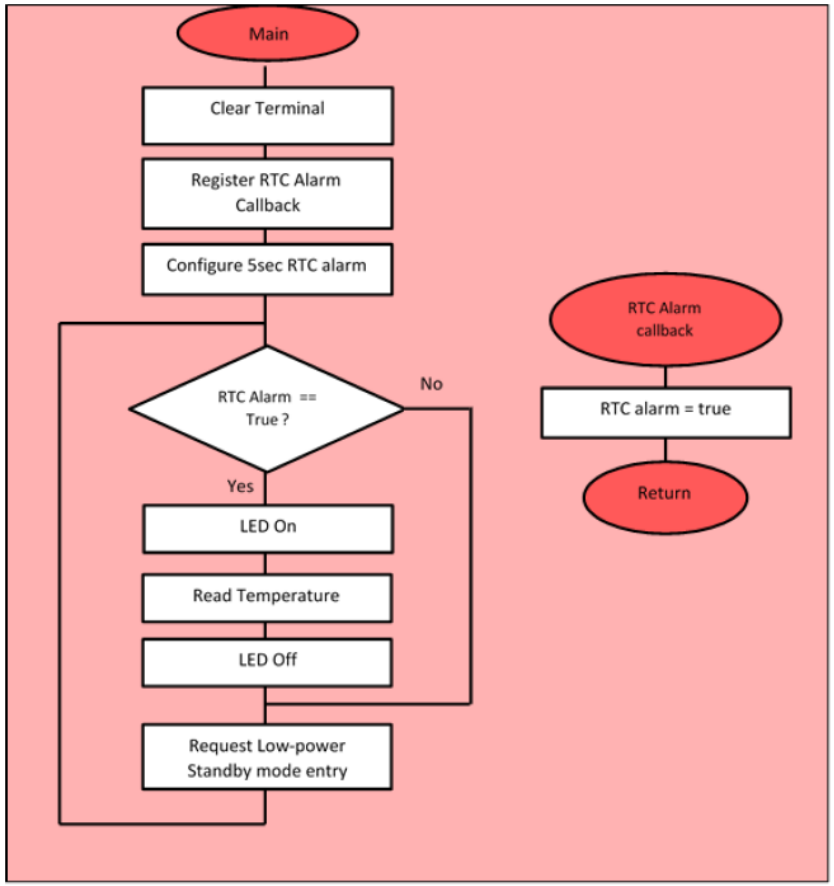


Image Source: http://microchip.com

Trusted Execution Environment

Malicious Examples:

- Jump to secure function
- Dump data from secure flash
- Dump data from secure Data flash
- Dump data from secure RAM
- Dump Trust RAM Memory
- Disable tamper
- Drive secure LED
- Drive secure COM

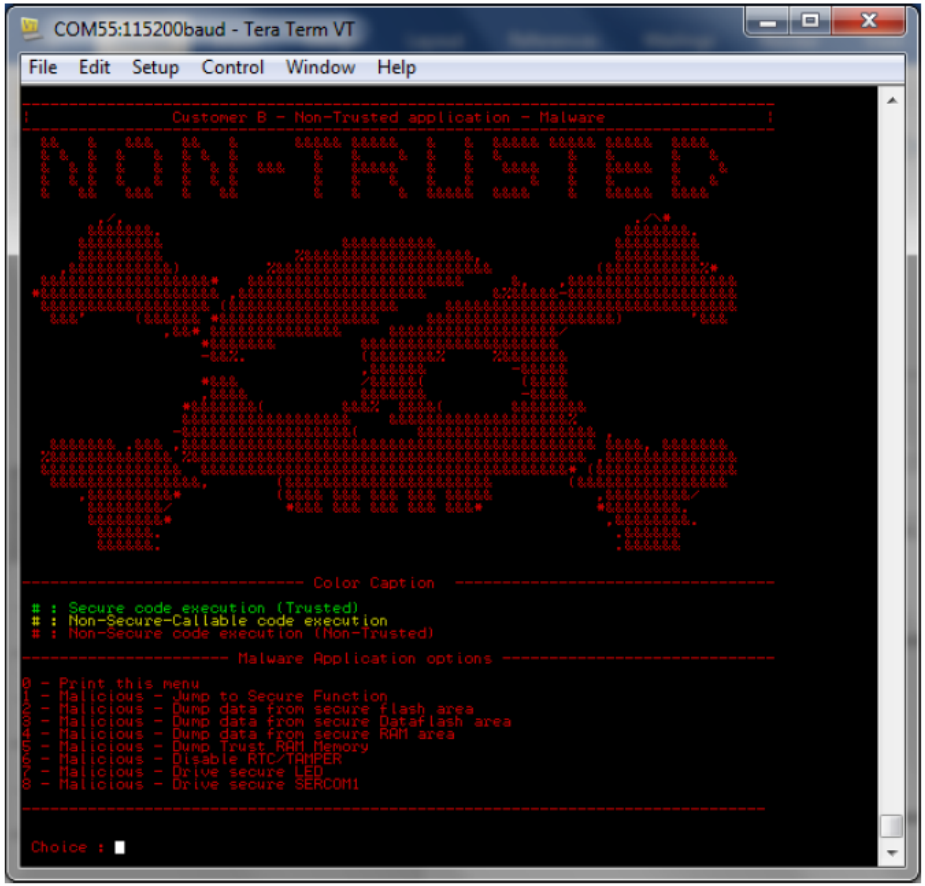


Image Source: http://microchip.com

Trusted Execution Environment

- https://www.youtube.com/watch?v=Mh1dhk5JTO4

Additional Resources

- Download Course Material for
  - C/C++ Doxygen Templates
  - Example source code
  - Blog
  - YouTube Videos
- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Securing IoT Devices using Arm TrustZone