Qualcomm Developer Project  
**QCA4020 Secure Boot**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribution:** |  | | |
| **Email address** | [yanggh0703@thundersoft.com](mailto:yanggh0703@thundersoft.com)  [hanfd0702@thundersoft.com](mailto:hanfd0702@thundersoft.com)  [zhouxiang0419@thundersoft.com](mailto:zhouxiang0419@thundersoft.com) | | |
| **Project Title**\* |  | | |
| **Description**\*  *High level description of the project* ***(75 words or less)*** | This demo shows how to enable secure boot function based on Home Automation demo. The secure boot is to establishes a trusted platform for running authenticated applications, and only authorized software can executes. | | |
| **Images**  *Upload up to 5 images of your project*  *Please submit/send the original JPEG/PNG files for all images included in the document* | **qca4020\_devkit\_130x130.png**  qca4020_devkit_130x130  [alt tag: “**Secure Boot using the QCA4020 development board.**”]  **servo\_motor\_130x130.png**  servor_motor_130x130  [alt tag: “**Secure Boot using the servo motor.**”]  zigbee\_lamp\_130x130.png  Zigbee_Lamp_130x130  [alt tag: “**Secure Boot using the zigbee lamp.**”]  overall\_project\_130x130.png  overall_project_130x130  [alt tag: “**Secure Boot’s assembly diagram.**”] | | |
| **Objective**   * *What inspired you to create this project?* * *What is your desired outcome?* | The main objective of this demo is to enable secure boot on the Home Automation demo. This is done by one-time programmable fuses on the QCA4020 chipset. For the code to be executed, it must be signed by the trusted entity identified in the hardware fuses. | | |
| **Operation System**\*  (Android, Linux, Windows 10 IoT Core) | Android   Linux  RTOS | | Windows 10 IoT Core   Ubuntu Core |
| **Cloud Services/Platform**  AT&T M2X, AWS IoT, IBM Bluemix, IBM Watson IoT, Such as Microsoft Azure IoT) | Amazon AWS IoT   AT&T M2x   IBM Bluemix | | IBM Watson IoT  Microsoft Azure IoT  Google Cloud Platform |
| **Skill Level Required**  (Beginner, Intermediate, Advanced) | Advanced   Beginner | | Intermediate |
| **Areas of Focus**  (e.g., IoT, smart cities, smart home, robotics, hardware, gaming, healthcare, automotive, digital signage, etc.) | 3D Printing & Modeling   Alexa Voice Service   Bluetooth   Computer Vision   Digital Signage   Education   Embedded    Gaming | | Healthcare   IoT   Robotics  Security   Sensors   Smart Cities   Smart Home   Toys  ZigBee |
| **Materials Required / Parts List / Tools** | Part Name | Link | |
| QCA4020 board | https://www.qualcomm.com/products/qca4020 | |
| Sengled bulb | https://www.amazon.com/Element-Plus-Sengled-2700-6500KSmartThings/dp/B01MQVYNFL/ref=sr\_1\_8?s=hi&ie=UTF8&qid=1523987884&sr=1-8&keywords=sengled+smart+bulb | |
| Lamp holder | https://item.m.jd.com/product/31912759280.html?ShareTm=pRYzt%2BY0bcuAYJHKosVgO%2B6g4QMPvCTgHHayaxLuN934yGW0ixCXpnTHCubwkoJjBMTrBiX5fdVZLMg0SLHZpL2ChVLi%2FgNTr2xnl%2Fn0itiT3EK87e6%2B2TteNvaxqWP5KzoZFR8WWiHUeN%2FdSAyycbow9%2FTdxahwR9r1DAg5hmY%3D&ad\_od=share&utm\_source=androidapp&utm\_medium=appshare&utm\_campaign=t\_335139774&utm\_term=Wxfriends | |
| Adaptor | unspecified | |
| Mobile phone | Android O | |
| Transformer(220v->110v) | unspecified | |
| **Source Code / Source Examples / Application Executable**  *Link to open source / shareable code repository* | Description | Link | |
| Source Code | <https://github.com/canyudeguang/Secure_Boot> | |
|  |  | |
| **Additional Resources**  *List related links or resources such as websites, videos, presentations, or other materials* | Resource Title | Link or File Name (and provide file) | |
| Video of “Zigbee lamp” |  | |
|  |  | |
| **Build / Assembly Instructions** | Parts used The items are the same as Home Automation demo.  Deploying the project  1.Download code from the [github repository](https://github.com/canyudeguang/Secure_Boot).  2.Compile the code and flash the image as described in “[QCLI\_demo with QCA4020 Development kit](https://developer.qualcomm.com/project/qclidemo-w-qca4020-dev-kit)” project.  3.Install the application which is the same as Home Automation demo.  How does it work?  Enable secure boot function on the QCA4020 development kit and only authorized software can executes. Enable secure boot via OTP programmer To set one-time programmable fuses, we need to use OTP programmer which is provided in the QCA4020 SDK.  (1) Edit the configuration file located at \target\quartz\mfg\OTP\tools\otp\_config.xml.  <otp\_descriptor>  <firmware\_region\_write\_disable>0</firmware\_region\_write\_disable>  <model\_id>0x0</model\_id>  <pk\_hash>0110203040506070809001102030405060708090011020304050607080900102</pk\_hash>  <!-- <otp\_encryp\_key>0102030405060708090a0b0c0d0e0f00</otp\_encryp\_key> -->  <otp\_profile>development</otp\_profile>  </otp\_descriptor>  (2) Compile OTP  Execute the command "build.bat t 4020 cdb" at  target\quartz\mfg\OTP\build\gcc.  (3) Flash the OTP programmer image to the development board at \target\quartz\mfg\OTP\build\gcc with the command:  flash\_openocd.bat  (4) Ensure that the USB serial cable is connected to the PC and a serial terminal is active.  (5) Flash the OTP programmer image at \target\quartz\mfg\OTP\build\gcc with the python script qflash.py, then I can get the "OTP update success" message appearing on the serial console.  IMG_256 Building signed image After set one-time programmable fuses, only authorized software can executes. So we must sign image.  (1)Modify the script \target\quartz\demo\QCLI\_demo\build\gcc\build.bat  SET SECBOOT=true  ...  IF /I "%BOARD\_VARIANT%" == "CDB" (      python %SectoolsQdnDir%\sectools.py iot -p 4020 -g m4 -i %OUTDIR%\%PROJECT%.elf -k %SectoolsCertsDir%\qpsa\_rootca.key -c %SectoolsCertsDir%\qpsa\_rootca.cer --cfg\_oem\_id=0xffff --cfg\_model\_id=0x0000 -o . -s      python %SectoolsQdnDir%\sectools.py iot -p 4020 -g m0 -i %RootDir%\bin\cortex-m0\threadx\ioe\_ram\_m0\_threadx\_ipt.mbn -k %SectoolsCertsDir%\qpsa\_rootca.key -c %SectoolsCertsDir%\qpsa\_rootca.cer --cfg\_oem\_id=0xffff --cfg\_model\_id=0x0000 -o . -s      python %SectoolsQdnDir%\sectools.py iot -p 4020 -g kf -i %RootDir%\bin\wlan\wlan\_fw\_img.bin -k %SectoolsCertsDir%\qpsa\_rootca.key -c %SectoolsCertsDir%\qpsa\_rootca.cer --cfg\_oem\_id=0xffff --cfg\_model\_id=0x0000 -o . -s  (2)Execute the command “build.bat t 4020 cdb”，then you can get the following files.  IMG_256 Flashing signed image Edit the flash script at \target\quartz\demo\QCLI\_demo\build\gcc\flash\_openocd.bat as follows:  (1) Modify the flash script to select the signed images.  SET M4\_DIR=%ROOTDIR%\quartz\demo\QCLI\_demo\build\gcc\4020\m4  SET M0\_DIR=%ROOTDIR%\quartz\demo\QCLI\_demo\build\gcc\4020\m0  SET WLAN\_DIR=%ROOTDIR%\quartz\demo\QCLI\_demo\build\gcc\4020\kf  SET M4\_IMAGE="%M4\_DIR%\Quartz.elf"  SET M0\_IMAGE="%M0\_DIR%\ioe\_ram\_m0\_threadx\_ipt.mbn"  SET WLAN\_IMAGE="%WLAN\_DIR%\wlan\_fw\_img.bin"  (2) Execute the command “flash\_openocd.bat” to flash the signed images to the development board at \target\quartz\demo\QCLI\_demo\build\gcc.  Now that you understand how controlling the light bulb is implemented in the project, below are some usage instructions to test the project. | | |
| **Usage Instructions** | Power on the QCA4020 via the power button, and then you can control it as Home Automation demo.  Note:  1.You need to read Build / Assembly Instructions carefully when you enable secure boot function.  2.If you cannot run it well, you must check that the OTP programmer and the signed image is in the same version of QCA4020 SDK.There are two version of qca4020 SDK which are qca4020.or\_.1.1\_postcs1 and qca4020-or-2-0\_qca\_oem\_sdk-cdb.  3.If you cannot connect JTAG, you must install FTDI driver for JTAG and OpenOCD plugin.  4.If you cannot flash the image through JTAG, check that if you flashed image with USB before, if it is, please delete the three files which are firmware\_table.bin, generated\_fwd\_table.xml and generated\_partition\_table.xml located at target\quartz\demo\QCLI\_demo\build\gcc\output. | | |
| **Contributor(s) Info**  *Feel free to include headshots!* | Name | Title  Company | |
| Yang  [yanggh0703@thundersoft.com](mailto:yanggh0703@thundersoft.com) | Thundersoft | |
| Jay  [zhouxiang0419@thundersoft.com](mailto:zhouxiang0419@thundersoft.com) | Thundersoft | |
| Scott  [hanfd0702@thundersoft.com](mailto:hanfd0702@thundersoft.com) | Thundersoft | |

*By submitting your content (“Submission”), you are granting Qualcomm a royalty-free, perpetual, non-exclusive, unrestricted, worldwide license to: (a) post, use, copy, sublicense, adapt, transmit, publicly perform or display any such Submission, (b) use, reproduce, modify, adapt, publish, translate, create derivative works from, distribute, perform, play, host, communicate, make available and publish your Submission without restriction and (c) sublicense to third parties the unrestricted right to exercise any of the foregoing rights granted with respect to the Submission. The foregoing grants shall include the right to exploit any ideas, concepts, intellectual property, or proprietary rights in such Submission, including but not limited to rights under copyright, trademark, servicemark or patent laws under any relevant jurisdiction without Qualcomm owing any monies to you whatsoever. You represent and warrant that you own all right, title and interest in and to the Submission, or you have been granted sufficient rights in and to the Submission allowing the foregoing use of such Submission.*