PL360 Security Features

Description

The PL360 is a programmable modem for narrow-band Power Line Communication (PLC), able to run any PLC protocol in the frequency band below 500 kHz.

This device has been designed to comply with FCC, ARIB, KN60 and CENELEC EN50065 regulations matching requirements of Internet of Things and Smart Energy applications. It supports state-of-the-art narrow-band PLC standards such as ITU G.9903 (G3-PLC), ITU G.9904 (PRIME) as well as any other narrow-band PLC protocols, at the same time being a future-proof platform able to support the evolution of these standards.

The PL360 has been conceived to be driven by external Microchip host devices, thus providing an additional level of flexibility on the host side. The Microchip host device loads the proper PLC-protocol firmware before modem operation and controls the PL360 modem.

Security Features

- Cryptographic Engine and Secure Boot
  - AES 128, 192, 256 supported
  - Secure boot: supports AES-128 CMAC for authentication, AES-128 CBC for decryption
  - One-time programmable fuses programming control for decryption and authentication 128-bit keys

Table of Contents

Description....1

Security Features.... 1

1. PL360 Security Features.... 3

2. Hardware Requirements for Fuse Programming....4

3. How to Program the Fuses....5
   - 3.1. OTP Fuses Registers....5
   - 3.2. Interface and Protocol.... 5
   - 3.3. PHY Fuses Tool Embedded Application....7

4. How to Encrypt and Sign the PL360 Binary.... 10
   - 4.1. Introduction....10
   - 4.2. Encrypt and Sign Script.... 11

5. How to Include Secure Boot on Firmware Application.... 13

6. Appendix A. How to Generate Keys Using Signature Functions....15

7. Revision History....16
   - 7.1. Rev A - 07/2018.... 16

The Microchip Web Site.... 17

Customer Change Notification Service....17

Customer Support.... 17

Microchip Devices Code Protection Feature.... 17

Legal Notice....18

Trademarks.... 18

Quality Management System Certified by DNV....19

Worldwide Sales and Service....20

1. PL360 Security Features

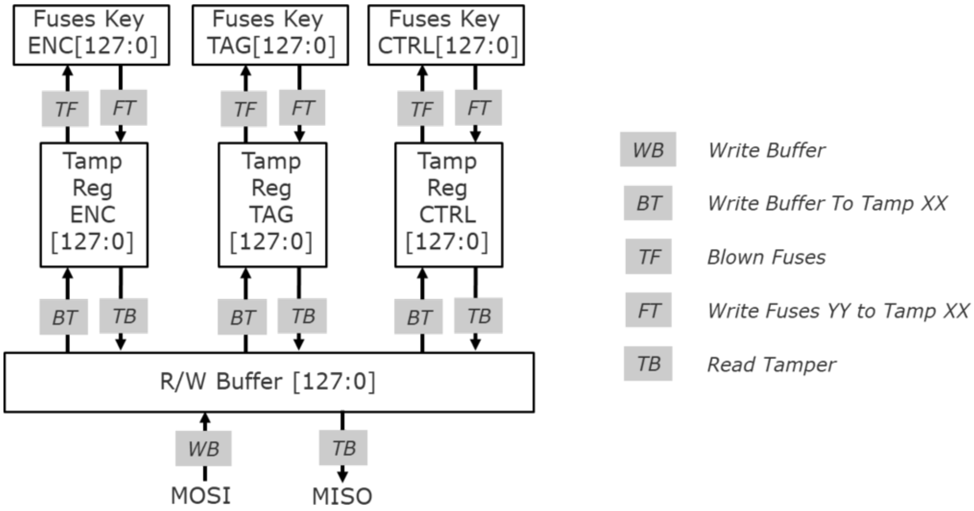
The PL360 contains a Secure Boot mechanism that guarantees the secure transfer of the firmware binary. To achieve this advanced functionality, the PL360 includes the following blocks:

- AES-CMAC: allows signature verification of binaries loaded on PL360
- AES-CBC: allows decryption of binaries loaded on PL360
- OTP Fuses block:
  - KEY_TAG: 128b signature key used to verify signature of binary
  - KEY_ENC: 128b encryption key used to decrypt binary
  - CONTROL_FUSES: 128b Control Fuse key for configuration

PL360 contains a One-Time-Programmable (OTP) fuses block which is a series of nonvolatile memory registers that can only be programmed once and where data is stored permanently.

The figure below shows the structure of the registers and data transfers for fuses and their control logic.

Figure 1-1. Fuse Controller Structure



To take advantage of the security capabilities of PL360, the following processes are involved:

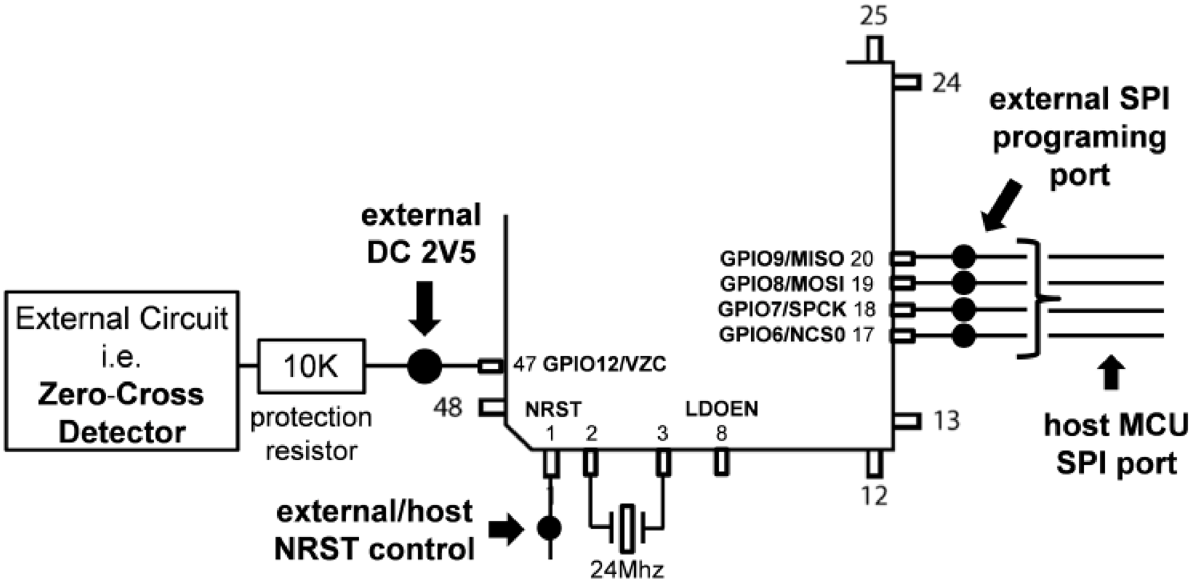
- Program the KEY_ TAG, KEY_ENC and CONTROL_FUSES fuses blocks
- Sign and encrypt the corresponding PL360 binary with KEY_ TAG and KEY_ENC
- Include Secure Boot capabilities in the host application

The following sections describe each one of those processes in detail.

2. Hardware Requirements for Fuse Programming

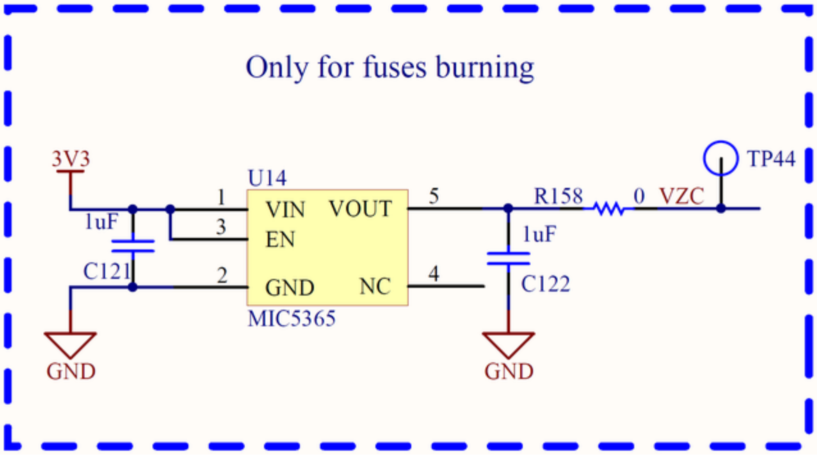
To be able to permanently modify the control or key fuses, an external supply of DC 2.5V ±10% 50 mA must be connected to the VZC pin. If the programming is done in system, an adequate protection of the VZC circuitry should be implemented by means of a 10K resistor, as shown in the figure below:

Figure 2-1. Hardware Requirements for Fuse Programming



The PL360MB-EK evaluation kit is prepared for fuses programming. To generate the voltage of 2.5V to the VZC pin in the PL360MB board, it is necessary to mount the components shown in the figure below:

Figure 2-2. Fuses Programming in PL360MB-EK



The fuse programming commands are sent through SPI. In case of using an external fuses programming controller, the SPI port pins of the host MCU must be set in High Impedance mode as to not interfere in the connection between PL360 and the external fuses programming controller.

3. How to Program the Fuses

3.1 OTP Fuses Registers

PL360 contains an OTP fuses block which is a series of nonvolatile memory registers that can only be programmed once and where data is stored permanently. In this OTP fuses block, there are several 128-bit registers related to security.

KEY_TAG and KEY_ENC are 128-bit registers used by the AES128-CMAC and AES128-CBC modules inside the PL360 Bootloader to store the keys when Secure Boot is enabled in order to verify the signature and decrypt the loaded binary.

CONTROL_FUSES is a 128-bit fuse register that controls the security features of PL360. Only some of the bits are user-configurable as shown on the next table:

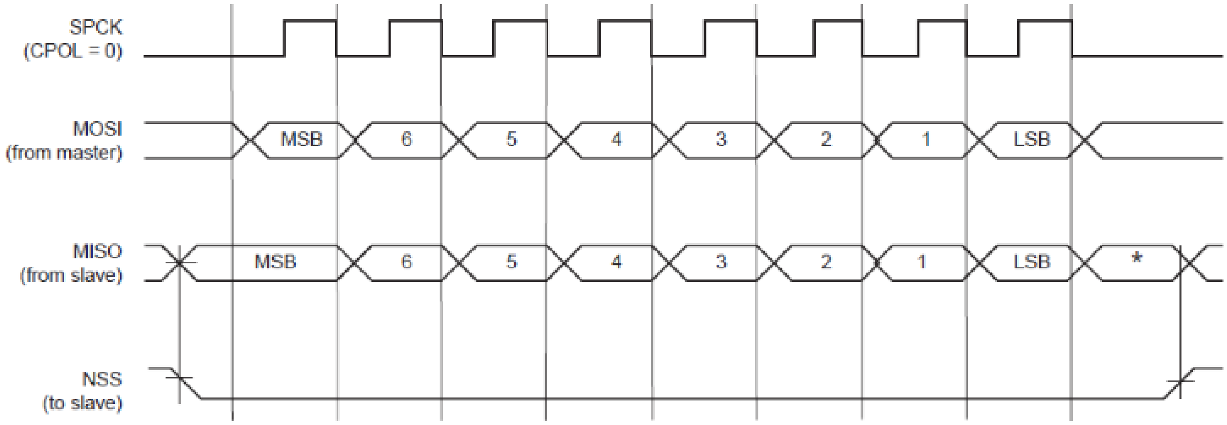
Table 3-1. Fuse bits

Fuse Bit	Name	Description
0	ENCRNOTPLAIN	If it is set, Secure mode is active. Only encrypted and signed binaries with the correct keys will be accepted by the bootloader
1	READ_AES_KEY	If it is set, KEY_ENC and KEY_TAG can’t be read
2	WRITE_AES_KEY	If it is set, KEY_ENC and KEY_TAG can’t be written
5	READ_CONTROL	If it is set, CONTROL_FUSES can’t be read
6	WRITE_CONTROL	If it is set, CONTROL_FUSES can’t be written
7	READ_RAM	If it is set, RAM memory can’t be read
10	FORCE_IVNBINC	If it is set, initialization vector and number of blocks must be used in the calculation of the signature
16	DBG_DISABLE	If it is set, JTAG debug is disabled

3.2 Interface and Protocol

The bootloader interface with host controller is a SPI bus which works in SPI Mode 0 (CPHA=1 and CPOL=0). The basic data transfer is:

Figure 3-1. Typical Bootloader SPI Frame



The bootloader requires a specific SPI frame format in order to interact with PL360 Bootloader to send commands and receive data. The SPI frame format used is shown in the following table:

Table 3-2. Bootloader SPI Frame Format

Address	Command	Data
32 bits	16 bits	n words of 32 bits

The PL360 bootloader implements the following SPI commands related to the security features of PL360:

Command	Description	Addr(31:0)	DATA(n*32-1:0)
0x0007	Write 128 bits fuses value to the Buffer register	0x00000000	0xDDDDDDDD
0x0008	Write Buffer register to the Tamper register for KEY_ENC_FUSES	0x00000000	0x00000000
0x0009	Write Buffer register to the Tamper register for KEY_TAG_FUSES	0x00000000	0x00000000
0x000B	Write Buffer register to the Tamper register for CONTROL_FUSES	0x00000000	0x00000000
0x000C	Blow desired fuses	0x00000000	0x00000000
0x000D	Write KEY_ENC_FUSES to the corresponding Tamper register	0x00000000	0x00000000
0x000E	Write KEY_TAG_FUSES to the corresponding Tamper register	0x00000000	0x00000000
0x0010	Write CONTROL_FUSES to the corresponding Tamper register	0x00000000	0x00000000
0x0011	Read Tamper register	0x00000000	0x000…000
0x0012	Read bootloader status	0x00000000	0x00000000

To write a fuse box, the data must be written in advance in the Buffer register (CMD=0x0007) and then, the content of the buffer written on the Tamper registers KEY_ENC_BOX, KEY_TAG_BOX or

CONTROL_BOX by means of the corresponding command (CMD=0x0008, CMD=0x0009 and CMD=0x000B respectively).

As the last step, to blow the desired fuses with the values in the corresponding Tamper register, the command (CMD=0x000C) must be sent. It’s necessary to check the bootloader status (CMD=0x0012) to know when this writing process finishes. If the writing process is active, bit 0 of the answer is ‘1’; in any other case, all data of the answer is ‘0’.

To activate the features controlled by CONTROL_FUSES, it is necessary to write the CONTROL_FUSES values to the corresponding Tamper register (CMD=0x0010).

To read the fuses registers KEY_ENC, KEY_TAG or CONTROL_FUSES, it is necessary to write their content to the Tamper register using the corresponding command (CMD=0x000D, CMD=0x000E and CMD=0x0010 respectively). After that, it is possible to read the Tamper register (CMD=0x0011) with the desired fuses register value.[1]

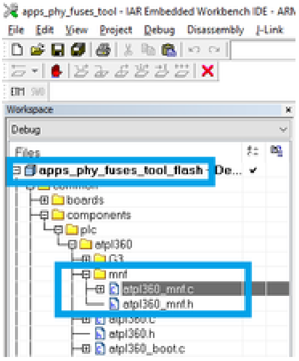
Note:

1. More information about this process can be found in the PL360 Datasheet.

3.3 PHY Fuses Tool Embedded Application

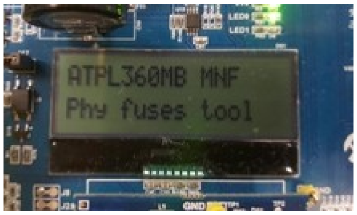
Microchip provides a firmware project example for the PL360 Evaluation Kit named “PHY Fuses Tool” in folder *“thirdparty\PROTOCOL\phy\atpl360\apps\phy_fuses_tool”* that could be used for fuse programming. The folder *“common\components\plc\atpl360\mnf”* contains the functions related to fuse programming.

Figure 3-2. PHY Fuses Tool Project Tree


apps_phy_fuses_tool - IAR Embedded Workbench IDE - ARM
File Edit View Project Debug Disassembly J-Link
Workspace
Debug
Files
apps_phy_fuses_tool_flash - De...
common
boards
components
plc
atpl360
G3
mnf
atpl360_mnf.c
atpl360_mnf.h
atpl360.c
atpl360.h
atpl360_boot.c

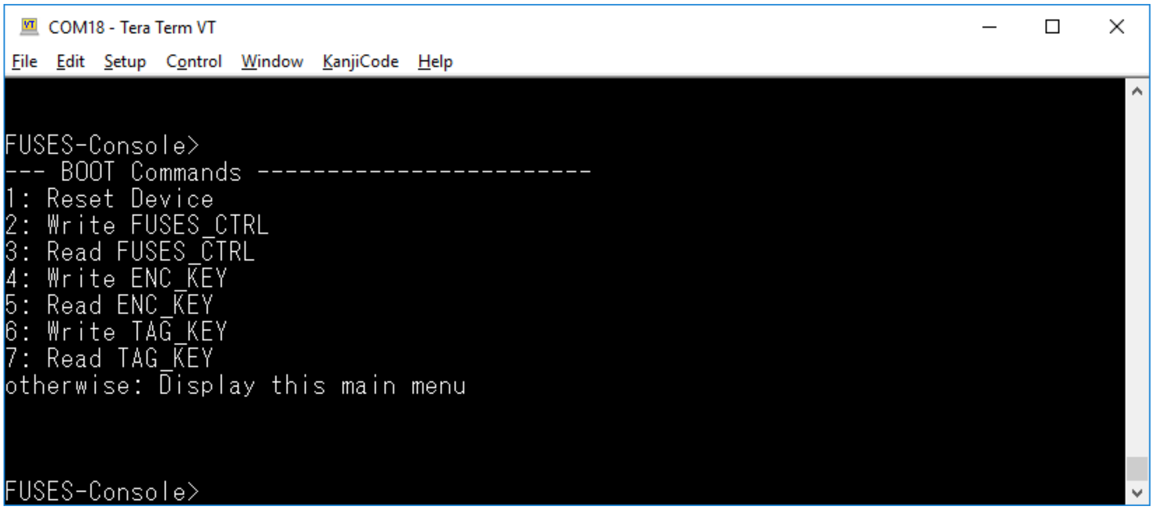
When the firmware runs in the PL360MB-EK, the LCD shows the message “Phy fuses tool”.

Figure 3-3. LCD Content for PHY Fuses Tool Firmware on PL360MB Evaluation Kit



Then, connect the board’s USB-Serial port to a computer running a serial terminal application configured as 115200-8-N-1, no flow control. That will provide access to the embedded console menu, displayed after reset or when pressing Enter.

Figure 3-4. Fuses Console Menu for PHY Fuses Tool

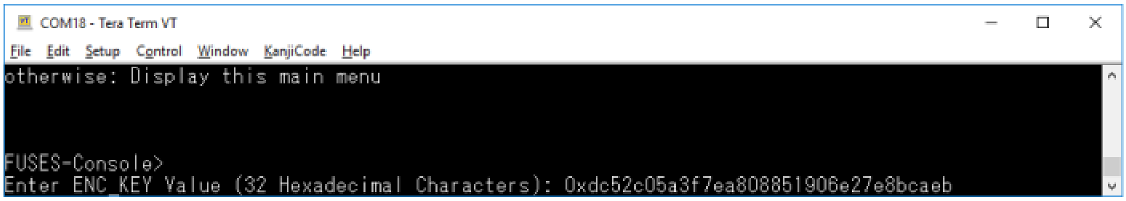


```
FUSES-Console>
--- BOOT Commands -----------------------
1: Reset Device
2: Write FUSES_CTRL
3: Read FUSES_CTRL
4: Write ENC_KEY
5: Read ENC_KEY
6: Write TAG_KEY
7: Read TAG_KEY
otherwise: Display this main menu


FUSES-Console>
```

For information about how to generate the 128-bit keys see 6. Appendix A. How to Generate Keys Using Signature Functions.

To program the ENC_KEY, select option 4 in console menu. ENC_KEY must be in hexadecimal format (32 hexadecimal characters).

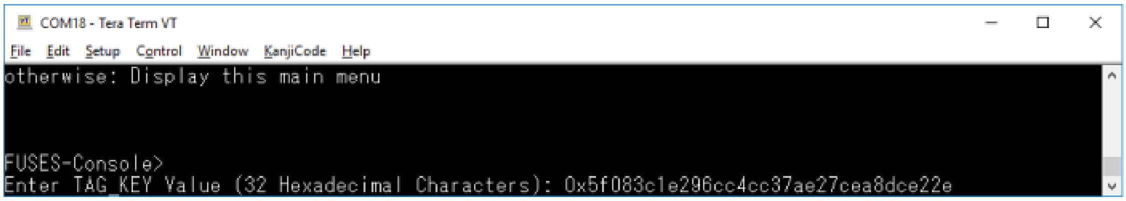
Figure 3-5. ENC_KEY Fuse Key Programming



```
otherwise: Display this main menu


FUSES-Console>
Enter ENC_KEY Value (32 Hexadecimal Characters): 0xdc52c05a3f7ea808851906e27e8bcaeb
```

To program the TAG_KEY, select option 6 in console menu. TAG_KEY must be in hexadecimal format (32 hexadecimal characters).

Figure 3-6. TAG_KEY Fuse Key Programming



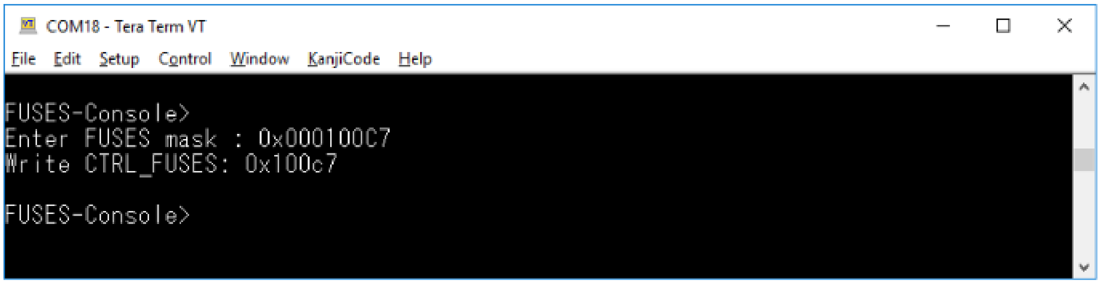
```
otherwise: Display this main menu


FUSES-Console>
Enter TAG_KEY Value (32 Hexadecimal Characters): 0x5f083c1e296cc4cc37ae27cea8dce22e
```

Tip: Press ‘ESC’ in order to cancel the programming operation.

To program CONTROL_FUSES, select option 2 in console menu. CONTROL_FUSES must be in hexadecimal format (8 hexadecimal characters).

Important: Since only fuses bits from 0 to 17 are user configurable, CONTROL_FUSES mask is 8 hexadecimal characters.

Figure 3-7. CONTROL_FUSES Fuse Programming



```
FUSES-Console>
Enter FUSES mask : 0x000100C7
Write CTRL_FUSES: 0x100c7

FUSES-Console>
```

4. How to Encrypt and Sign the PL360 Binary

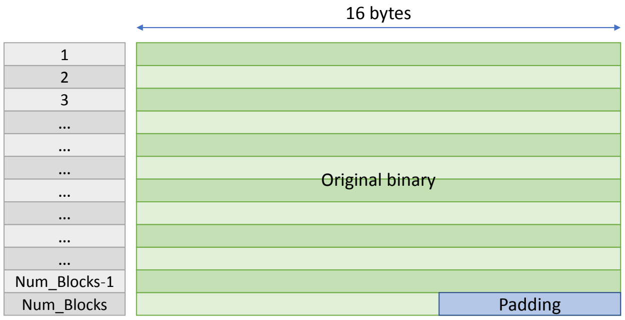
4.1 Introduction

An encrypted and signed binary is called “secured binary”. The process to get a secured binary is:

1. Encrypt the PL360 binary getting an “encrypted binary”
2. Sign the “encrypted binary”

To generate the “encrypted binary”, the original binary must be multiple of 16 bytes. In case of a non-multiple of 16 bytes binary, it must be padded. Padding consists of filling the binary with a '1' bit and all the necessary '0' bits up to reach a total size of the binary multiple of 16 bytes. Regarding the signature validation, the binary length must be specified as the number of 16-bytes blocks.

Figure 4-1. Padding of Non-Encrypted Binary



Binary to encrypt = Original binary + Padding

Padding = 0b10000000….0000

Binary to encrypt = N * 16 bytes

Depending on FORCE_IVNBINC fuse, the signature is calculated over the “encrypted binary” (if the fuse is not set) or over “encrypted binary + Initialization Vector + Number of blocks” (if the fuse is set).

Figure 4-2. Structure of Data Signed with FORCE_IVBINC=0

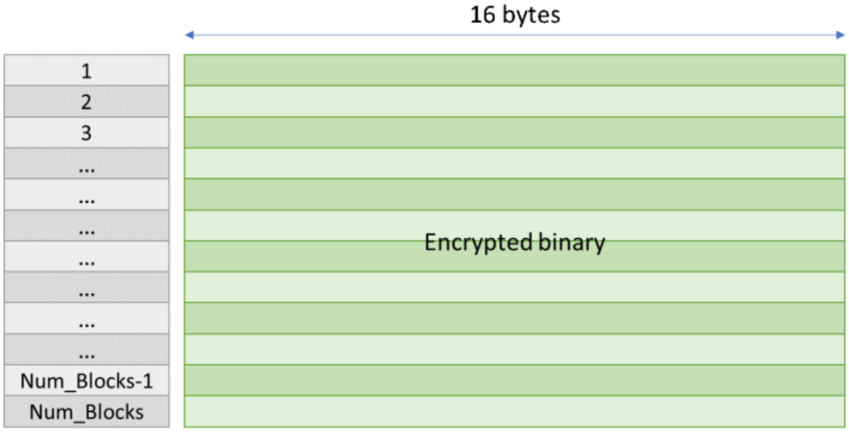
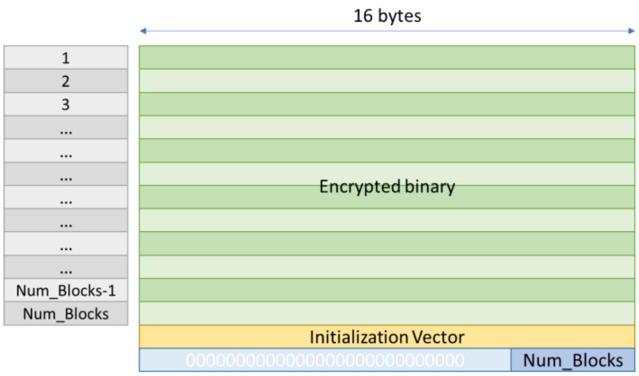


Figure 4-3. Structure of Data Signed with FORCE_IVBINC=1



Num_Blocks is stored in the last 2 bytes of the last 16-bytes block, the rest of the 16-bytes block is filled with 0's

4.2 Encrypt and Sign Script

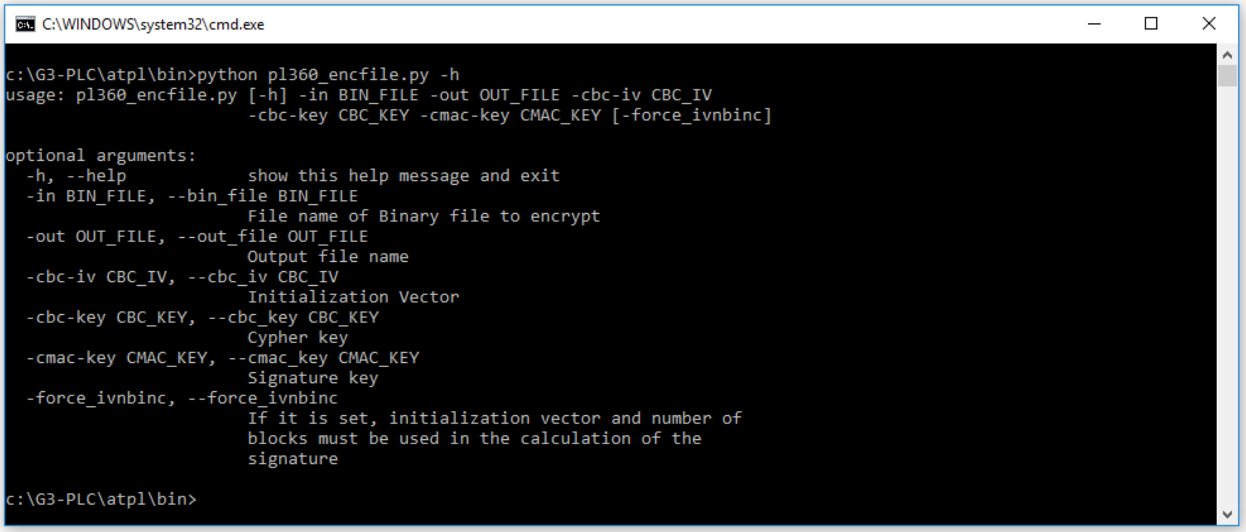
Microchip provides an example Python™ script to encrypt and calculate the signature of firmware binaries. You can find the Python script (pl360_encfile.py) in the folder *"atpl\bin\"*. In the same folder, it is also included a windows batch file (pl360_encfile.bat) which launches the Python script with the suitable parameters.

To use it, Python version 2.7 or 3 must be installed on the computer including the "cryptography" package.

Tip: To install the cryptography library, run "pip install cryptography" at the Python terminal.

Tip: Launch the script with arguments "-h" or "—help" to see the required parameters.

Figure 4-4. Python pl360_encfile.py Help Menu



```
c:\G3-PLC\atpl\bin>python pl360_encfile.py -h
usage: pl360_encfile.py [-h] -in BIN_FILE -out OUT_FILE -cbc-iv CBC_IV
                        -cbc-key CBC_KEY -cmac-key CMAC_KEY [-force_ivnbinc]

optional arguments:
  -h, --help            show this help message and exit
  -in BIN_FILE, --bin_file BIN_FILE
                        File name of Binary file to encrypt
  -out OUT_FILE, --out_file OUT_FILE
                        Output file name
  -cbc-iv CBC_IV, --cbc_iv CBC_IV
                        Initialization Vector
  -cbc-key CBC_KEY, --cbc_key CBC_KEY
                        Cypher key
  -cmac-key CMAC_KEY, --cmac_key CMAC_KEY
                        Signature key
  -force_ivnbinc, --force_ivnbinc
                        If it is set, initialization vector and number of
                        blocks must be used in the calculation of the
                        signature

c:\G3-PLC\atpl\bin>
```

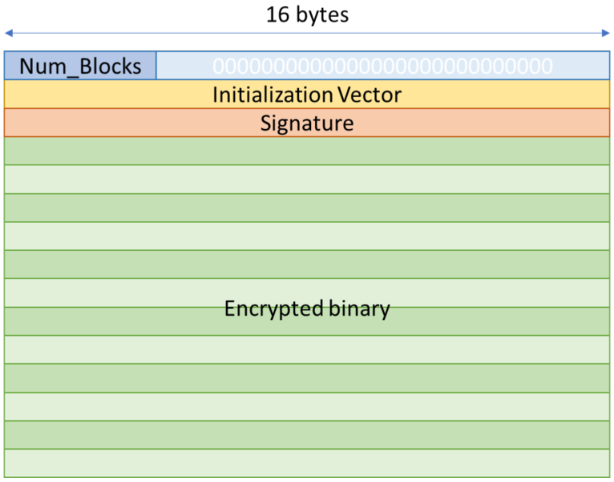
The script requires the following parameters:

- Binary file to encrypt
- Output encrypted binary file
- KEY_ENC key programmed in the PL360 fuses (in the script, it is called CBC_KEY)
- KEY_TAG key programmed in the PL360 fuses (in the script, it is called CMAC_KEY)
- Initialization vector used for encryption/decryption (in the script, it is called CBC_IV)
- Boolean variable to indicate if signature is calculated over “encrypted binary + Initialization Vector + Number of blocks” (True) or not (False)

Important: ENC_KEY, TAG_KEYand IV_KEY must be in hexadecimal format characters.

Using the parameters as inputs, the script will encrypt and sign the PL360 firmware, generating a new binary inside the “PL360_ENCFILE” folder. The structure of this secured binary obtained as output file is the following:

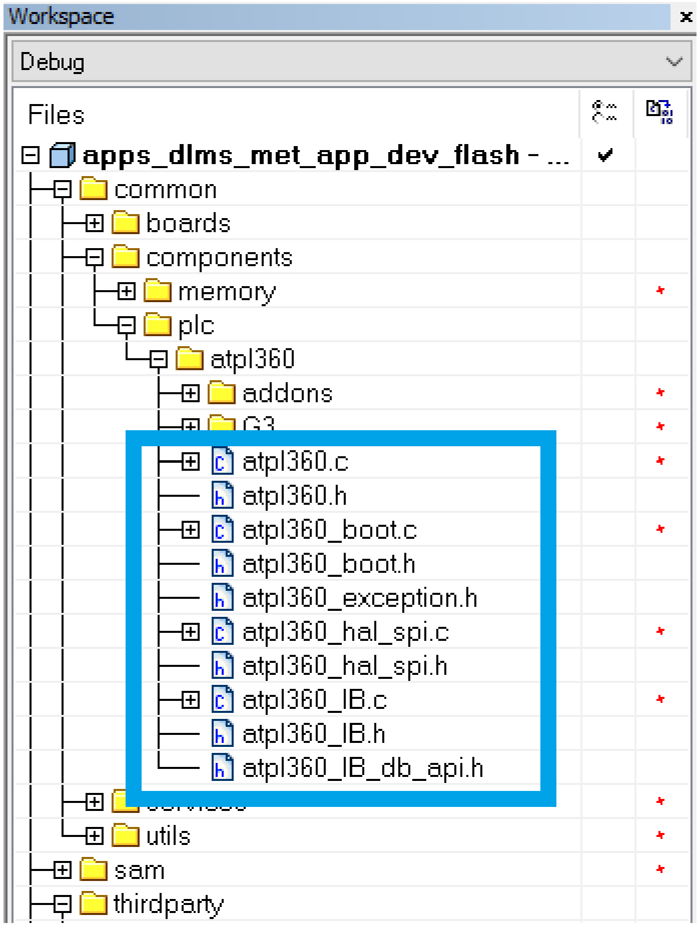
Figure 4-5. Structure of the Secured Binary



5. How to Include Secure Boot on Firmware Application

PL360 projects include all the functions to implement “normal” and “secure” boot.

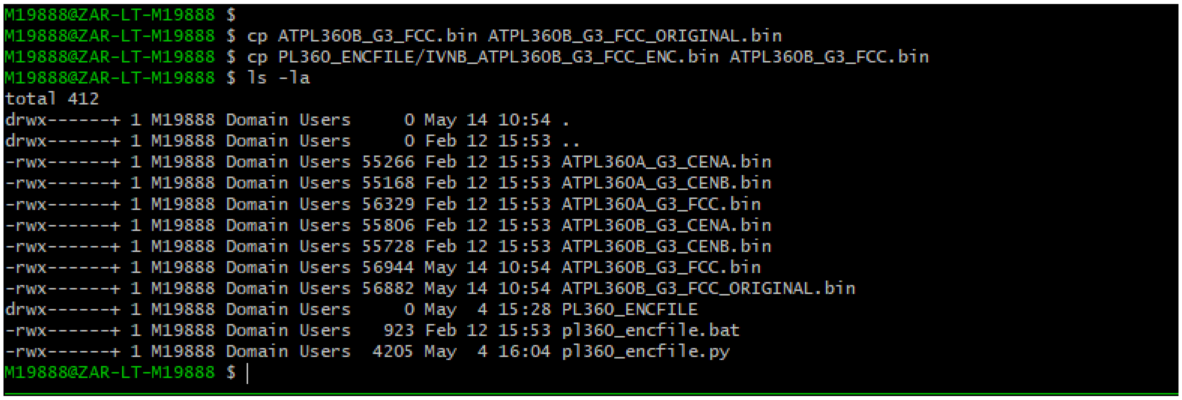
Figure 5-1. PL360 Common Components Tree



Once PL360 binary has been encrypted and signed, the way to work with it in the firmware application project is the following:

1. Copy secured binary in “atpl\bin\” folder with the name of the original PL360 binary file

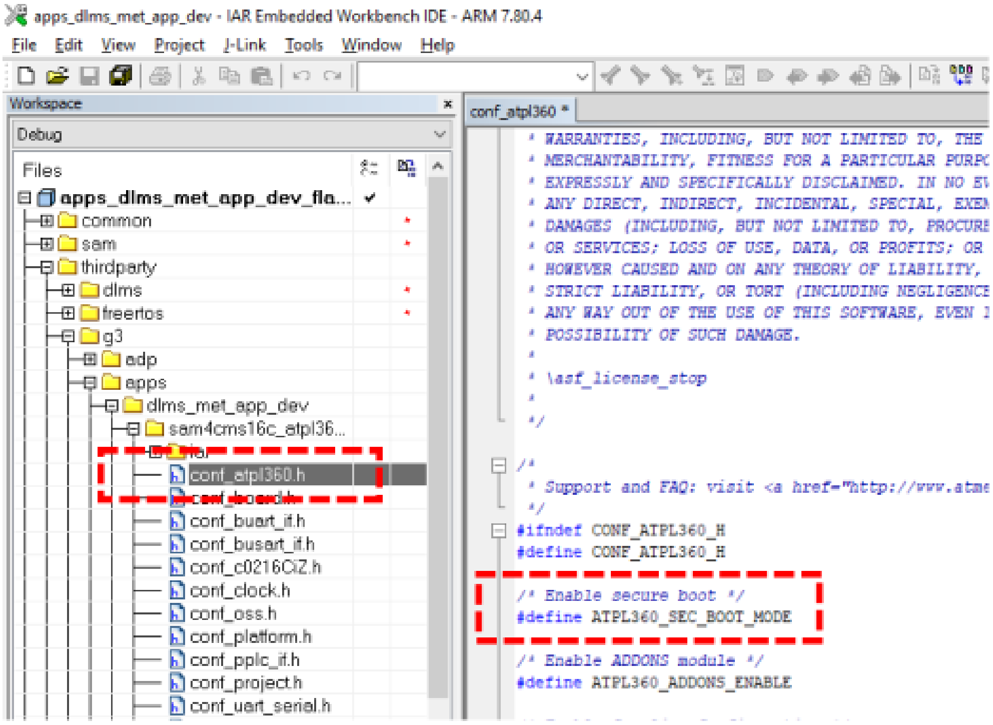
Figure 5-2. Original PL360 Binary Substitution with Secured Version



```
M19888@ZAR-LT-M19888 $
M19888@ZAR-LT-M19888 $ cp ATPL360B_G3_FCC.bin ATPL360B_G3_FCC_ORIGINAL.bin
M19888@ZAR-LT-M19888 $ cp PL360_ENCFILE/IVNB_ATPL360B_G3_FCC_ENC.bin ATPL360B_G3_FCC.bin
M19888@ZAR-LT-M19888 $ ls -la
total 412
drwx------+ 1 M19888 Domain Users     0 May 14 10:54 .
drwx------+ 1 M19888 Domain Users     0 Feb 12 15:53 ..
-rwx------+ 1 M19888 Domain Users 55266 Feb 12 15:53 ATPL360A_G3_CENA.bin
-rwx------+ 1 M19888 Domain Users 55168 Feb 12 15:53 ATPL360A_G3_CENB.bin
-rwx------+ 1 M19888 Domain Users 56329 Feb 12 15:53 ATPL360A_G3_FCC.bin
-rwx------+ 1 M19888 Domain Users 55806 Feb 12 15:53 ATPL360B_G3_CENA.bin
-rwx------+ 1 M19888 Domain Users 55728 Feb 12 15:53 ATPL360B_G3_CENB.bin
-rwx------+ 1 M19888 Domain Users 56944 May 14 10:54 ATPL360B_G3_FCC.bin
-rwx------+ 1 M19888 Domain Users 56882 May 14 10:54 ATPL360B_G3_FCC_ORIGINAL.bin
drwx------+ 1 M19888 Domain Users     0 May  4 15:28 PL360_ENCFILE
-rwx------+ 1 M19888 Domain Users   923 Feb 12 15:53 pl360_encfile.bat
-rwx------+ 1 M19888 Domain Users  4205 May  4 16:04 pl360_encfile.py
M19888@ZAR-LT-M19888 $ |
```

2. Define “ATPL360_SEC_BOOT_MODE” on “conf_atpl360.h” file

Figure 5-3. Firmware Definitions to Work with Secured PL360



```
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPC
 * EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EV
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEN
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCURE
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN I
 * POSSIBILITY OF SUCH DAMAGE.
 *
 * \asf_license_stop
 *
 */

/*
 * Support and FAQ: visit <a href="http://www.atme
 */
#ifndef CONF_ATPL360_H
#define CONF_ATPL360_H

/* Enable secure boot */
#define ATPL360_SEC_BOOT_MODE

/* Enable ADDONS module */
#define ATPL360_ADDONS_ENABLE
```

3. Compile and link the application project
4. Program the application binary in the host microcontroller
5. Run the binary in the host microcontroller

6. Appendix A. How to Generate Keys Using Signature Functions

The keys used in secured bootloader (ENC_KEY, TAG_KEY and IV_KEY) can be any combination of 128 bits (32 hexadecimal characters).

An easy way to generate customized and easily replicable keys is to use a Master Key file, calculate its MD5 and SHA256 hashes and use them as keys (as SHA256 is 256 bits long, 2 128-bit keys are obtained from it). MD5 and SHA256 hashes can be obtained from Windows® or Linux® terminal, or even from web-based tools. With this method, it is not necessary to store the keys in any place, just the Master Key file.

In the following example, a text file containing the word "Microchip" is used to obtain the ENC_KEY using MD5 and TAG_KEY & IV_KEY using SHA256.

- Linux terminal

```
$ echo -n "Microchip" > Microchip.txt

$ sha256sum -t Microchip.txt | cut -c 1-32
230fe79bc18ef87cd51096e450c8c27f

$ sha256sum -t Microchip.txt | cut -c 33-64
10456fd785df2af08397a2eb69647888

$ md5sum -t Microchip.txt | cut -c 1-32
f7962ad60e8c8ec26481358db76f7f6a
```

- Windows Powershell (version 3.0 or above)

```
PS C:\> "Microchip" | Out-File -encoding ascii Microchip.txt -NoNewline

PS C:\> (Get-FileHash .\Microchip.txt -Algorithm SHA256).hash.SubString(0,32)
230FE79BC18EF87CD51096E450C8C27F

PS C:\> (Get-FileHash .\Microchip.txt -Algorithm SHA256).hash.SubString(32,32)
10456FD785DF2AF08397A2EB69647888

PS C:\> (Get-FileHash .\Microchip.txt -Algorithm MD5).hash
F7962AD60E8C8EC26481358DB76F7F6A
```

7. Revision History

7.1 Rev A - 07/2018

Document	Initial document release.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-3328-6

Quality Management System Certified by DNV

ISO/TS 16949
Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra’anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820