Introduction to IMP Library

Agenda

- **Introduction to the IMP**
- Introduction to IMPGUI
- Introduction to IMP Library
- Capture IMP code using the switch GUI
- Build IMP code
- Run IMP code from IMP
- Create EEPROM image
- Burn EEPROM image
- Verify EEPROM image by using Switch GUI

IMP -- Integrated management Processors

- Z80 Compatible
  - 200 MHz
  - 60K RAM
- Integrated NIC for sending and receiving frames from/to switch
- Access to switch registers is memory mapped
- CPU and IMP can access switch registers simultaneously
- Boot options
  - Fast boot from EEPROM
  - Code can also be managed from host CPU.
- Replaces Legacy Register loader

IMP usage cases

- Switch initialization
- State machines
  - Example automatically Flush ATU entries for a port when link goes down
- Active LED management
- Managing external clause 45 PHYs or Clause 22 PHYs
- Bridge loop detection

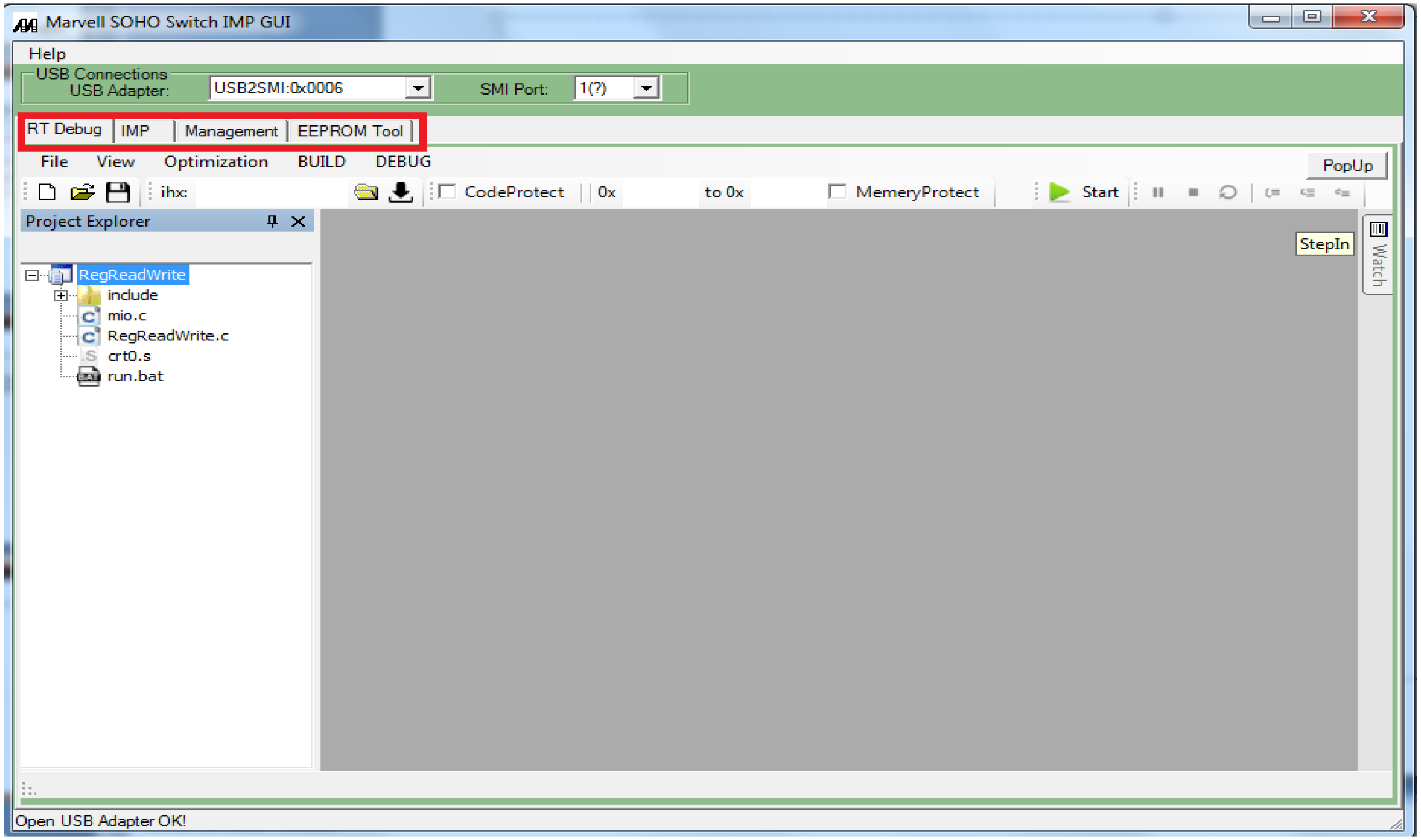
….

Agenda

- Introduction to the IMP
- **Introduction to IMPGUI**
- Introduction to IMP Library
- Capture IMP code using the switch GUI
- Build IMP code
- Run IMP code from IMP
- Create EEPROM image
- Burn EEPROM image
- Verify EEPROM image by using Switch GUI

Introduction of IMPGUI

- IMPGUI (Integrated Micro Processor GUI) is a debug tool for Peridot and Topaz internal Z80 processor.



Marvell SOHO Switch IMP GUI

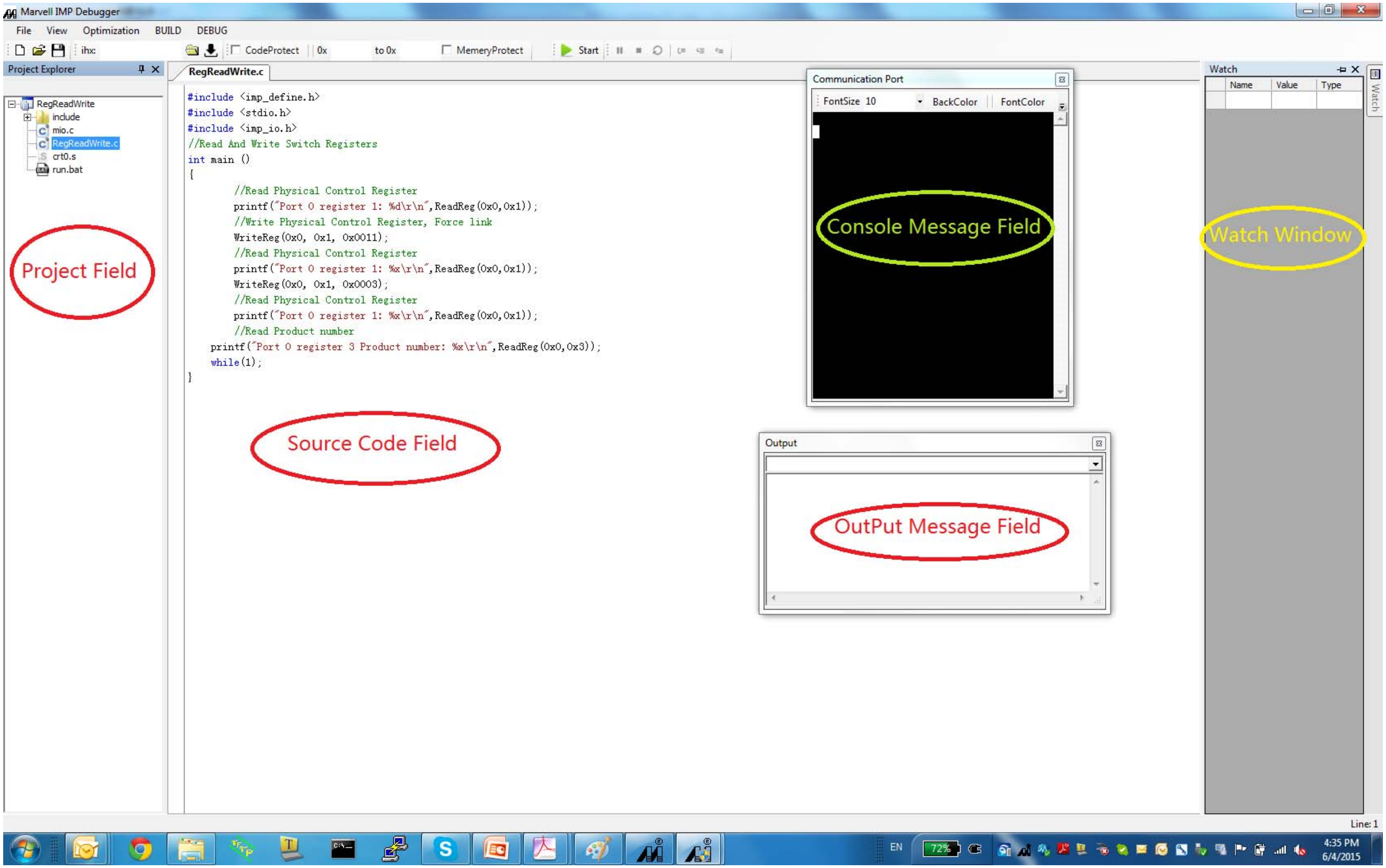
USB Connections

USB Adapter: USB2SMI:0x0006

SMI Port: 1(?)

Open USB Adapter OK!

IMPGUI RT Debug



```
#include <imp_define.h>
#include <stdio.h>
#include <imp_io.h>
//Read And Write Switch Registers
int main ()
{
        //Read Physical Control Register
        printf("Port 0 register 1: %d\r\n",ReadReg(0x0,0x1));
        //Write Physical Control Register, Force link
        WriteReg(0x0, 0x1, 0x0011);
        //Read Physical Control Register
        printf("Port 0 register 1: %x\r\n",ReadReg(0x0,0x1));
        WriteReg(0x0, 0x1, 0x0003);
        //Read Physical Control Register
        printf("Port 0 register 1: %x\r\n",ReadReg(0x0,0x1));
        //Read Product number
    printf("Port 0 register 3 Product number: %x\r\n",ReadReg(0x0,0x3));
    while(1);
}
```

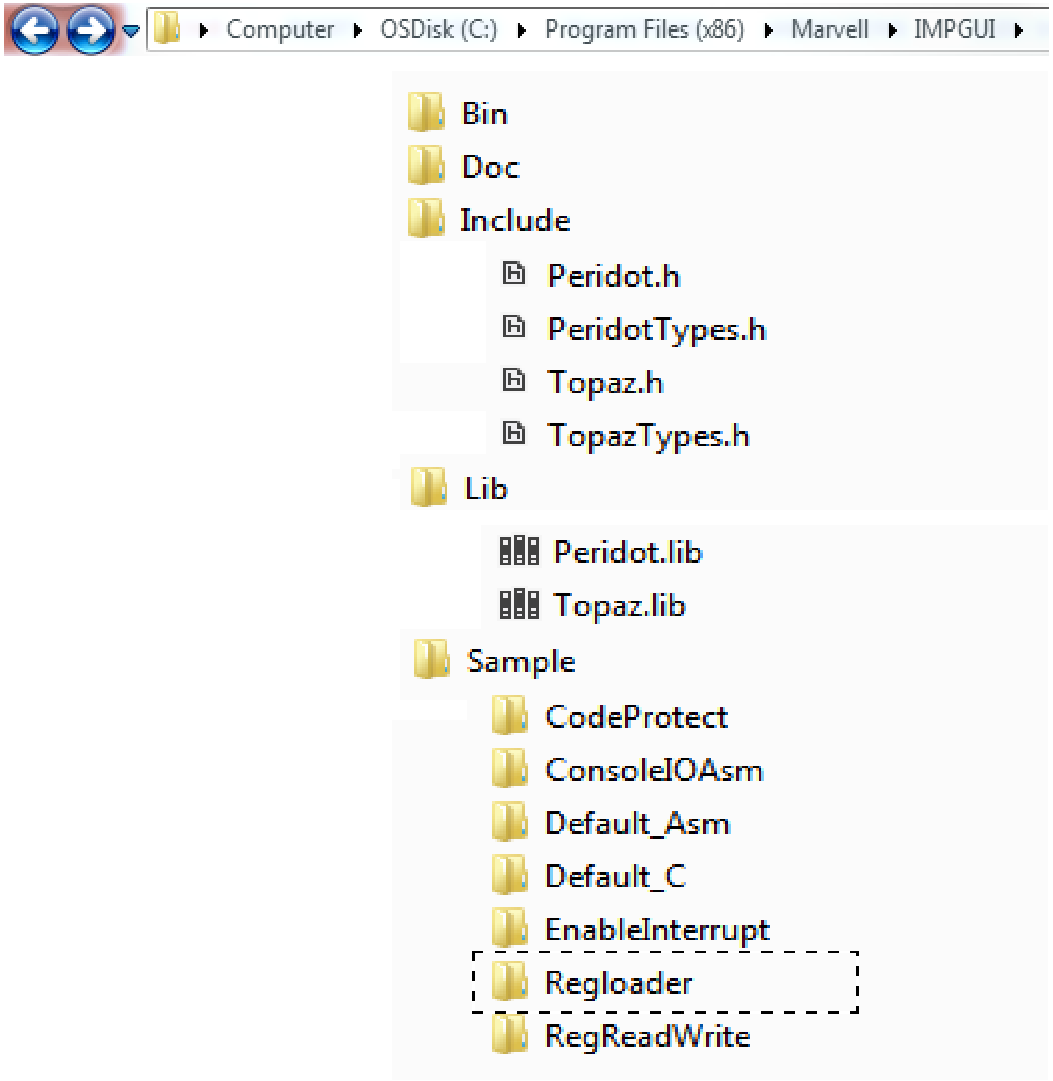
Agenda

- Introduction to the IMP
- Introduction to IMPGUI
- **Introduction to IMP Library**
- Capture IMP code using the switch GUI
- Build IMP code
- Run IMP code from IMP
- Create EEPROM image
- Burn EEPROM image
- Verify EEPROM image by using Switch GUI

IMP Library

- IMP library provide the basic switch supporting APIs.
- Module supports: ATU, VTU, IRL, TCAM, Common Register Access, IO, Reset
- Built with same compiler the GUI uses
- The library is included with the IMPGUI after reversion **1.10.0.**

IMPGUI Library



Computer ▸ OSDisk (C:) ▸ Program Files (x86) ▸ Marvell ▸ IMPGUI ▸

```
Bin
Doc
Include
    Peridot.h
    PeridotTypes.h
    Topaz.h
    TopazTypes.h
Lib
    Peridot.lib
    Topaz.lib
Sample
    CodeProtect
    ConsoleIOAsm
    Default_Asm
    Default_C
    EnableInterrupt
    Regloader
    RegReadWrite
```

IMPGUI Library API list

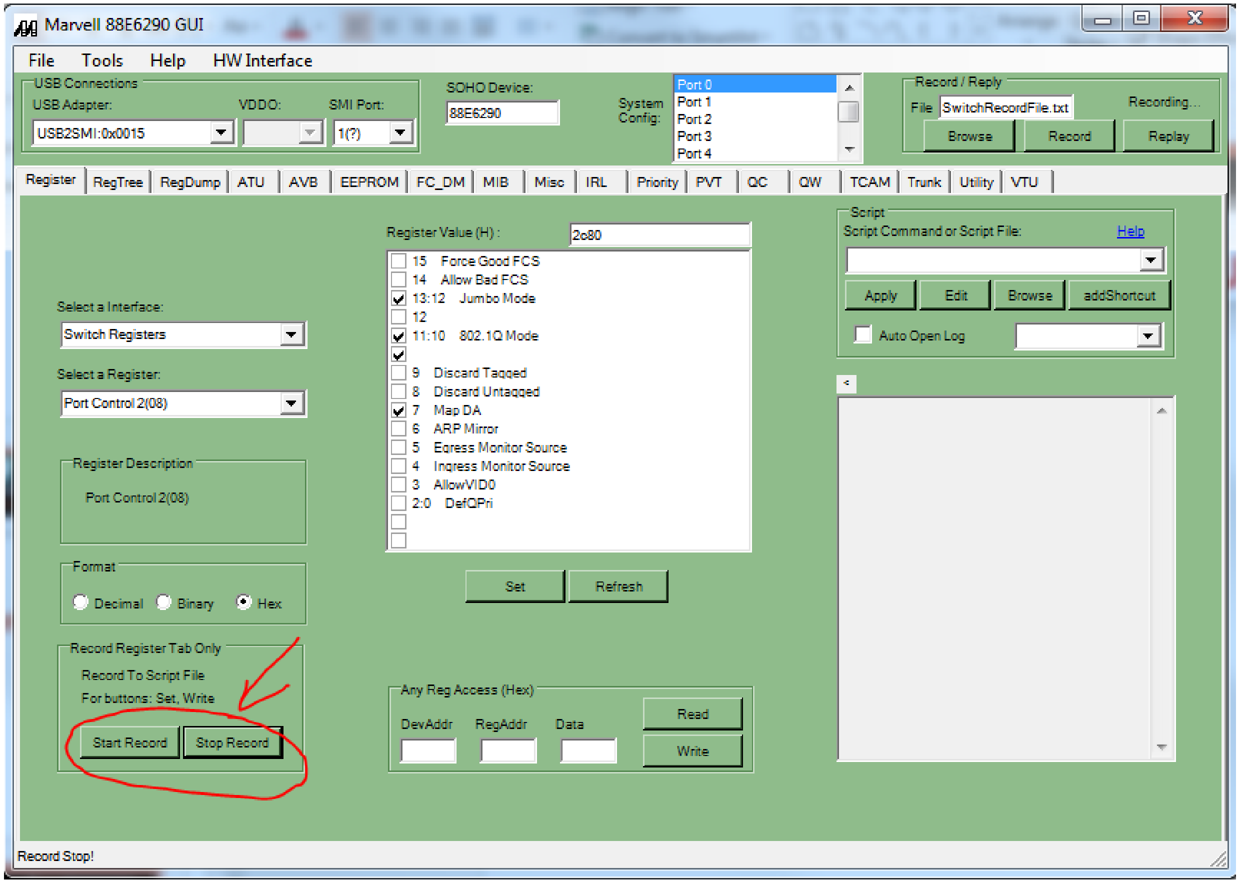
- ATU
  - atuAddEntry, atuFlush, atuGetEntryNext
- VTU
  - vtuAddEntry, vtuFlush, vtuGetEntryNext
- TCAM
  - TCAMOP, tcamLoadEntry, TCAMFlushAll, tcamReadTCAMData, tcamGetNextTCAMData, tcamLoadEgrEntry
- IRL
  - irlInitialize, irlWrite, irlRead

- Common Register Access
  - ReadReg, WriteReg, PortRead, PortWrite, G1Read, G1Write
  - G2Read, G2Write, C45RegWrite, C45RegRead,
  - C22RegWrite, C22RegRead, ExtC22Read, ExtC22Write
  - IntC22Read, IntC22Write
- IO
  - FastCopy, memcpy_fast, eeprom2ram
- Reset
  - ResetSwitch

Build and run image from IMPGUI

MARVELL

Generate IMP code with the SwitchGUI



Marvell 88E6290 GUI

USB Connections
USB Adapter: USB2SMI:0x0015
VDDO:
SMI Port: 1(?)

SOHO Device: 88E6290

System Config: Port 0, Port 1, Port 2, Port 3, Port 4

Record / Reply
File: SwitchRecordFile.txt
Recording...

Register Value (H) : 2c80

□ 15 Force Good FCS
□ 14 Allow Bad FCS
☑ 13:12 Jumbo Mode
□ 12
☑ 11:10 802.1Q Mode
☑
□ 9 Discard Tagged
□ 8 Discard Untagged
☑ 7 Map DA
□ 6 ARP Mirror
□ 5 Egress Monitor Source
□ 4 Ingress Monitor Source
□ 3 AllowVID0
□ 2:0 DefQPri
□
□

Select a Interface: Switch Registers

Select a Register: Port Control 2(08)

Register Description: Port Control 2(08)

Format: ○ Decimal ○ Binary ● Hex

Record Register Tab Only
Record To Script File
For buttons: Set, Write

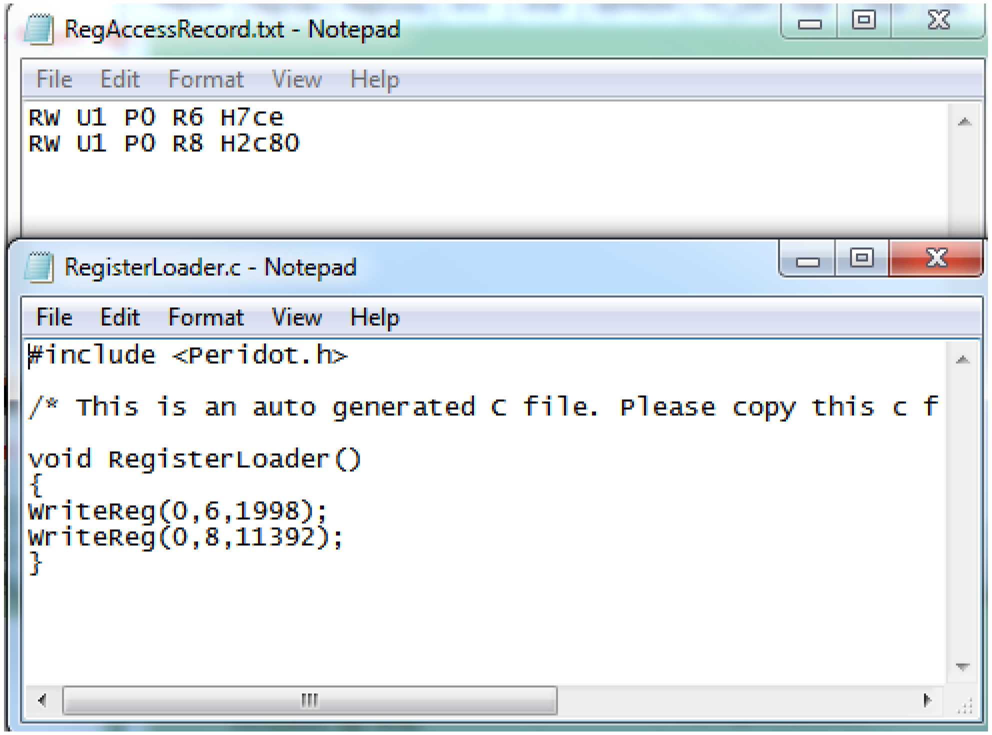
Any Reg Access (Hex)
DevAddr: RegAddr: Data:

Script
Script Command or Script File:
□ Auto Open Log

Record Stop!

Using switchGUI “start Record” “stop Record” button to produce the IMP write code.

Output imp source code



RegAccessRecord.txt - Notepad
File Edit Format View Help

```
RW U1 P0 R6 H7ce
RW U1 P0 R8 H2c80
```

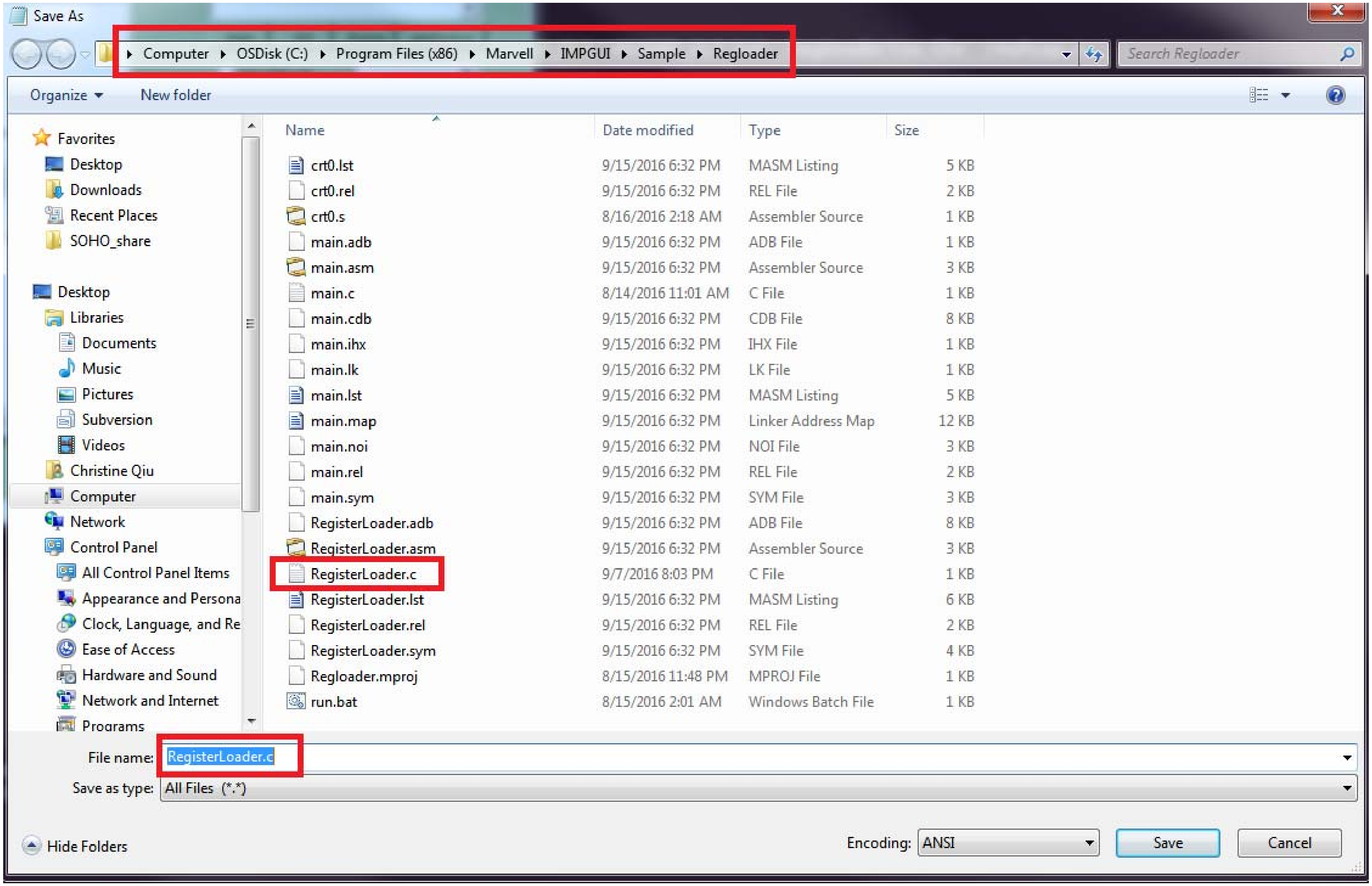
RegisterLoader.c - Notepad
File Edit Format View Help

```
#include <Peridot.h>

/* This is an auto generated C file. Please copy this c f

void RegisterLoader()
{
WriteReg(0,6,1998);
WriteReg(0,8,11392);
}
```

Save the output file “RegisterLoader.c” into the “Regloader” IMP project



Computer ▸ OSDisk (C:) ▸ Program Files (x86) ▸ Marvell ▸ IMPGUI ▸ Sample ▸ Regloader

Name	Date modified	Type	Size
crt0.lst	9/15/2016 6:32 PM	MASM Listing	5 KB
crt0.rel	9/15/2016 6:32 PM	REL File	2 KB
crt0.s	8/16/2016 2:18 AM	Assembler Source	1 KB
main.adb	9/15/2016 6:32 PM	ADB File	1 KB
main.asm	9/15/2016 6:32 PM	Assembler Source	3 KB
main.c	8/14/2016 11:01 AM	C File	1 KB
main.cdb	9/15/2016 6:32 PM	CDB File	8 KB
main.ihx	9/15/2016 6:32 PM	IHX File	1 KB
main.lk	9/15/2016 6:32 PM	LK File	1 KB
main.lst	9/15/2016 6:32 PM	MASM Listing	5 KB
main.map	9/15/2016 6:32 PM	Linker Address Map	12 KB
main.noi	9/15/2016 6:32 PM	NOI File	3 KB
main.rel	9/15/2016 6:32 PM	REL File	2 KB
main.sym	9/15/2016 6:32 PM	SYM File	3 KB
RegisterLoader.adb	9/15/2016 6:32 PM	ADB File	8 KB
RegisterLoader.asm	9/15/2016 6:32 PM	Assembler Source	3 KB
RegisterLoader.c	9/7/2016 8:03 PM	C File	1 KB
RegisterLoader.lst	9/15/2016 6:32 PM	MASM Listing	6 KB
RegisterLoader.rel	9/15/2016 6:32 PM	REL File	2 KB
RegisterLoader.sym	9/15/2016 6:32 PM	SYM File	4 KB
Regloader.mproj	8/15/2016 11:48 PM	MPROJ File	1 KB
run.bat	8/15/2016 2:01 AM	Windows Batch File	1 KB

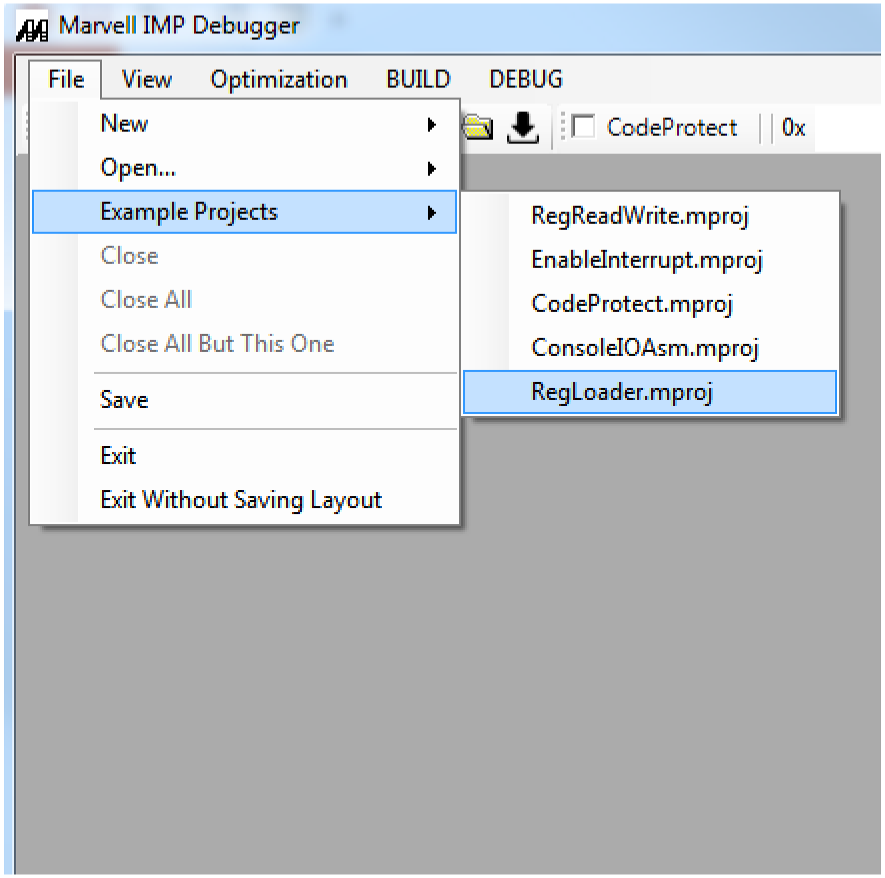
File name: RegisterLoader.c

Save as type: All Files (*.*)

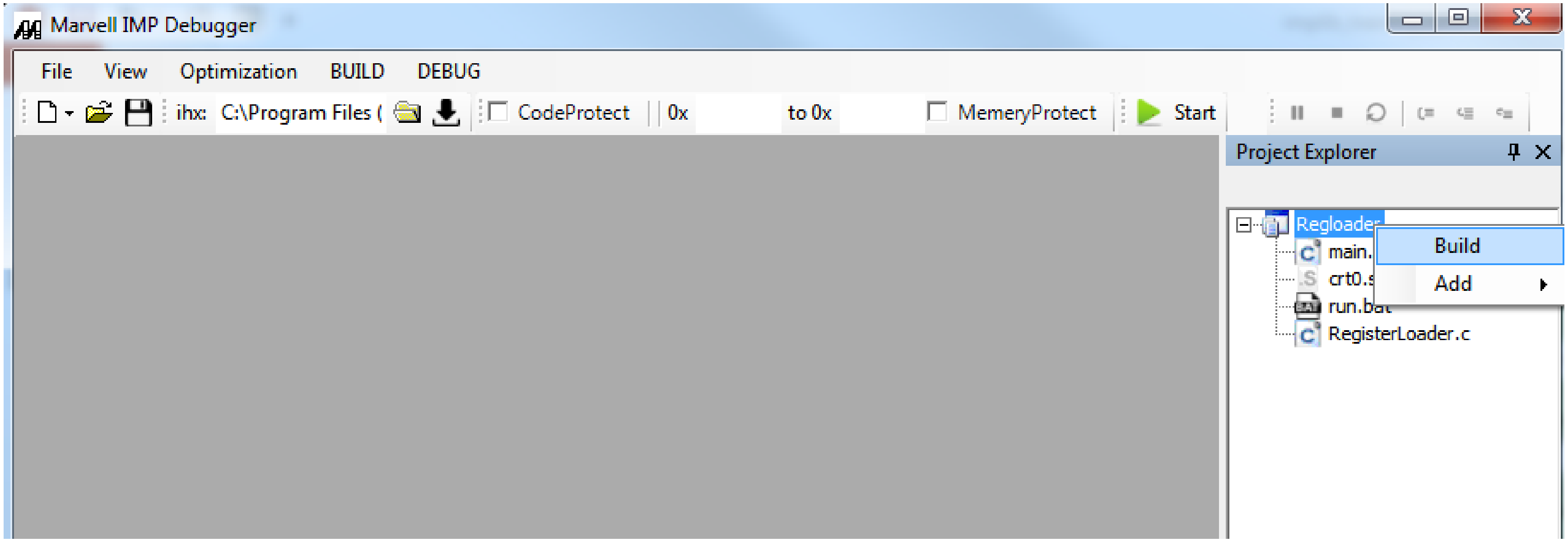
Encoding: ANSI

MARVELL

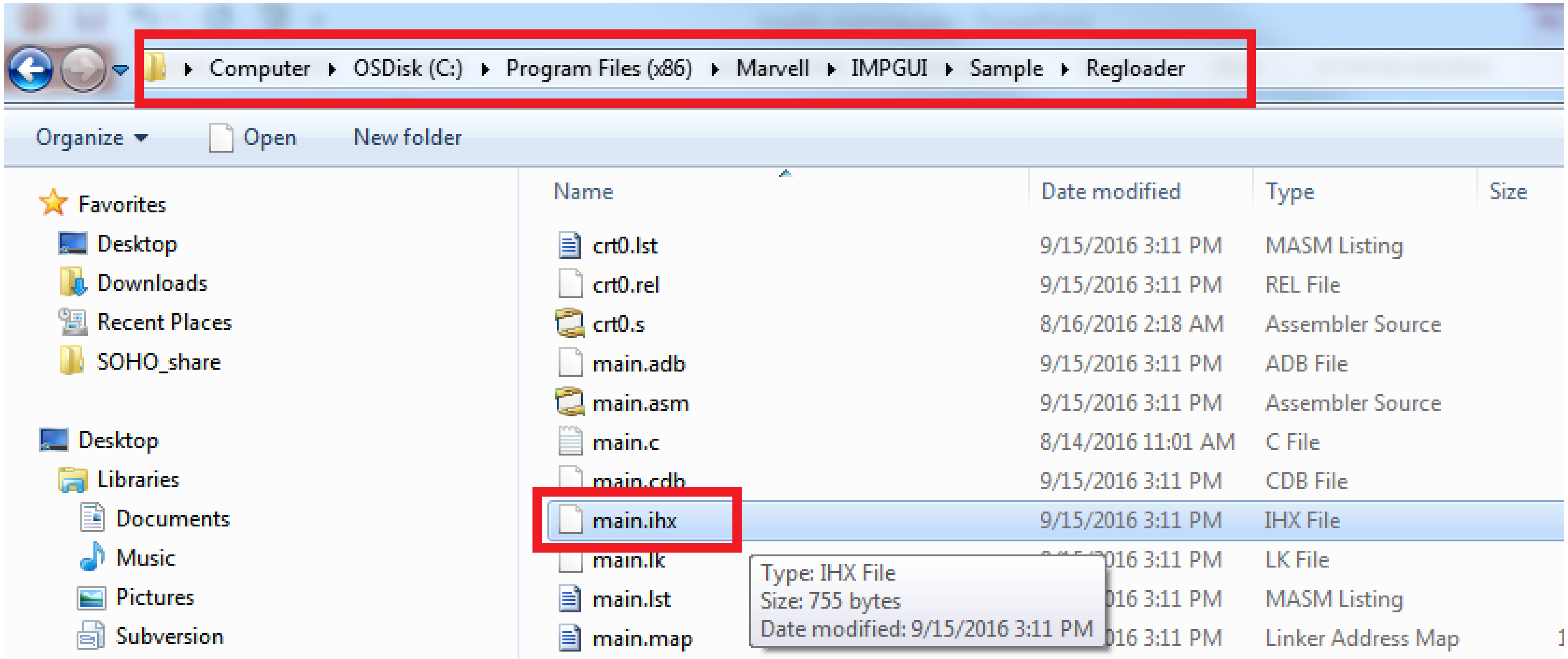
Open “RegLoader” project in IMPGUI



Build project



Output file is main.ihx



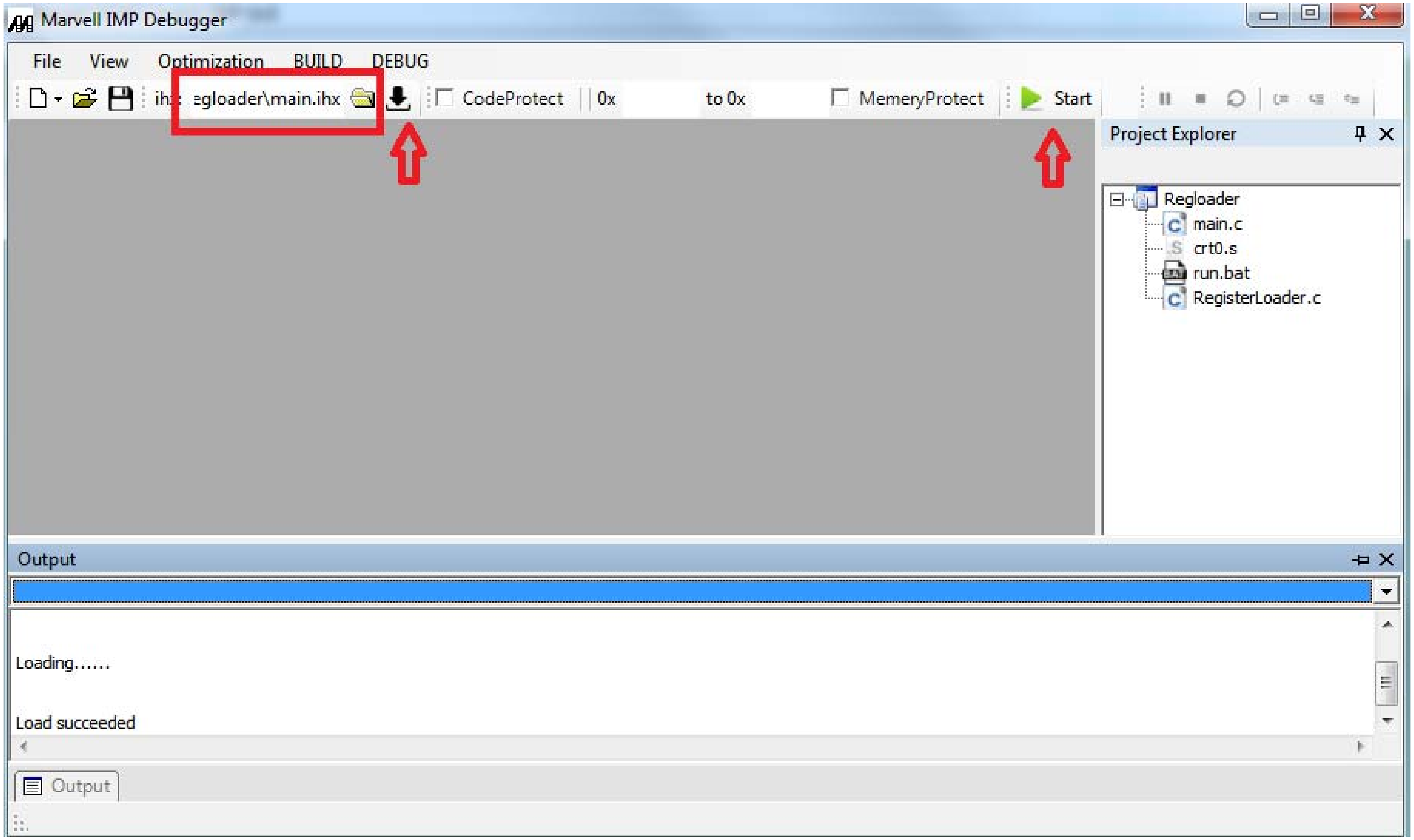
Computer ▸ OSDisk (C:) ▸ Program Files (x86) ▸ Marvell ▸ IMPGUI ▸ Sample ▸ Regloader

Name	Date modified	Type	Size
crt0.lst	9/15/2016 3:11 PM	MASM Listing	
crt0.rel	9/15/2016 3:11 PM	REL File	
crt0.s	8/16/2016 2:18 AM	Assembler Source	
main.adb	9/15/2016 3:11 PM	ADB File	
main.asm	9/15/2016 3:11 PM	Assembler Source	
main.c	8/14/2016 11:01 AM	C File	
main.cdb	9/15/2016 3:11 PM	CDB File	
main.ihx	9/15/2016 3:11 PM	IHX File	
main.lk	016 3:11 PM	LK File	
main.lst	016 3:11 PM	MASM Listing	
main.map	016 3:11 PM	Linker Address Map	1

Type: IHX File
Size: 755 bytes
Date modified: 9/15/2016 3:11 PM

Using the switchGUI to verify the main.ihx

- Reset the out and run the output.ihx on IMPGUI

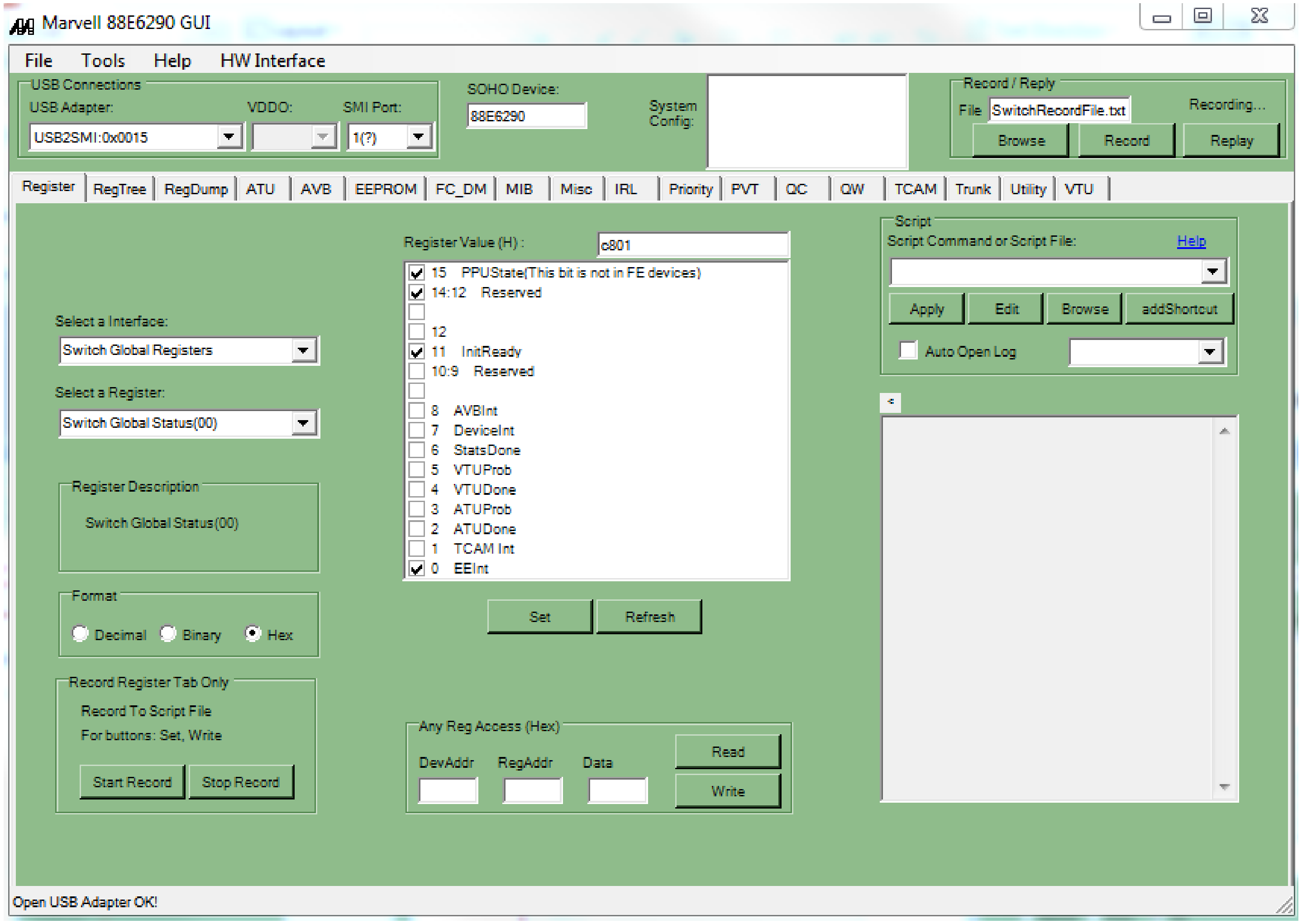


```
Loading......

Load succeeded
```

Using the switchGUI to verify the main.ihx

- Open the switchGUI and check whether the configuration is correct.



Marvell 88E6290 GUI

USB Connections
USB Adapter: USB2SMI:0x0015
VDDO:
SMI Port: 1(?)

SOHO Device: 88E6290

System Config:

Record / Reply
File SwitchRecordFile.txt
Recording...

Select a Interface: Switch Global Registers

Select a Register: Switch Global Status(00)

Register Description
Switch Global Status(00)

Format
○ Decimal ○ Binary ● Hex

Record Register Tab Only
Record To Script File
For buttons: Set, Write

Register Value (H) : c801

☑ 15 PPUState(This bit is not in FE devices)
☑ 14:12 Reserved
□
□ 12
☑ 11 InitReady
□ 10:9 Reserved
□
□ 8 AVBInt
□ 7 DeviceInt
□ 6 StatsDone
□ 5 VTUProb
□ 4 VTUDone
□ 3 ATUProb
□ 2 ATUDone
□ 1 TCAM Int
☑ 0 EEInt

Any Reg Access (Hex)
DevAddr
RegAddr
Data

Script
Script Command or Script File:
□ Auto Open Log

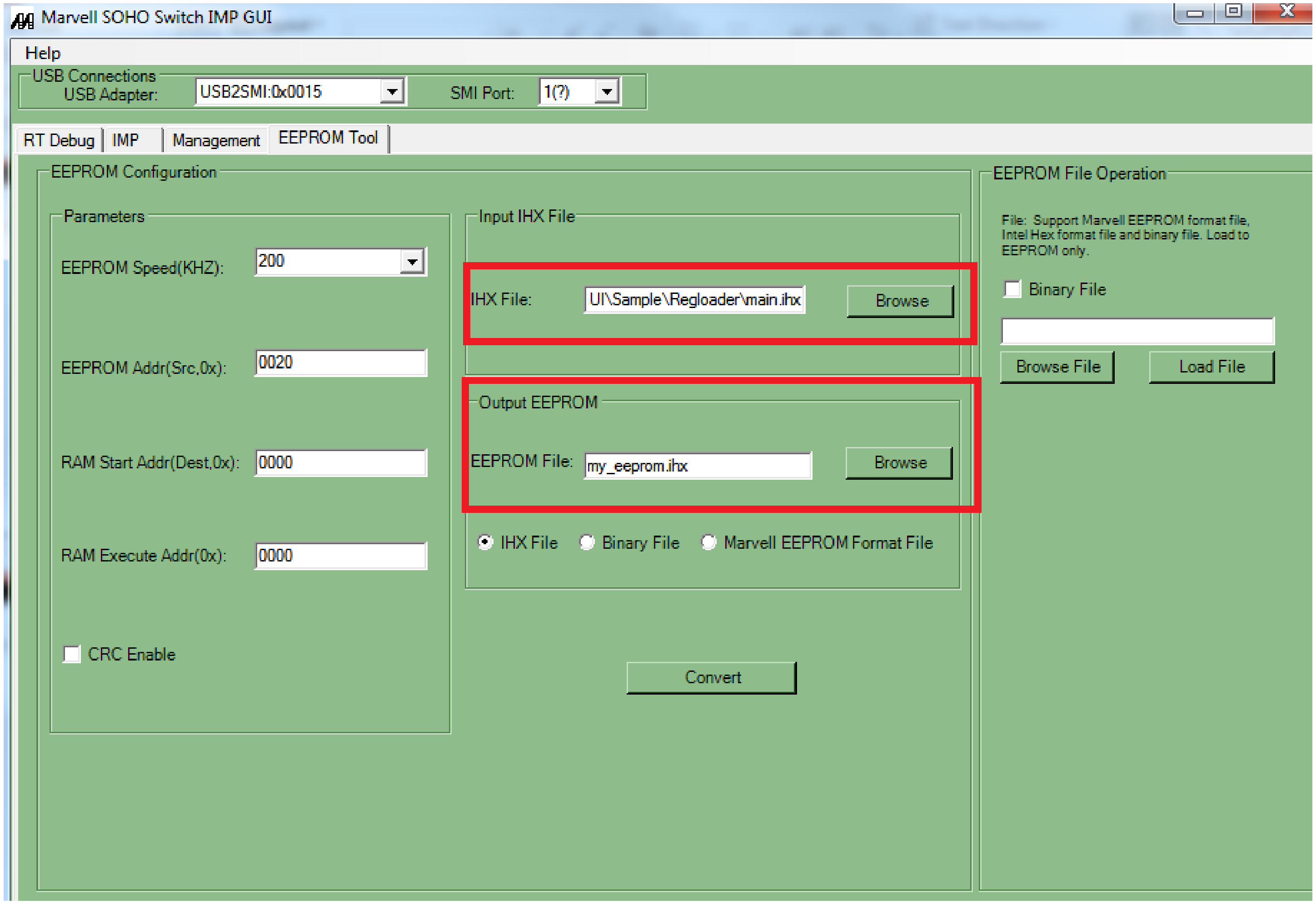
Open USB Adapter OK!

MARVELL

Convert IMP image EEPROM image

Using IMPGUI EEPROM tool to do the conversion

- Input IHX File is the main.ihx which IMPGUI just compile.
- Output Ihx is the EEPROM format IHX file which will add boot header in front of main.ihx.



Marvell SOHO Switch IMP GUI

USB Connections

USB Adapter: USB2SMI:0x0015

SMI Port: 1(?)

EEPROM Configuration

Parameters

EEPROM Speed(KHZ): 200

EEPROM Addr(Src,0x): 0020

RAM Start Addr(Dest,0x): 0000

RAM Execute Addr(0x): 0000

□ CRC Enable

Input IHX File

IHX File: UI\Sample\Regloader\main.ihx

Output EEPROM

EEPROM File: my_eeprom.ihx

● IHX File ○ Binary File ○ Marvell EEPROM Format File

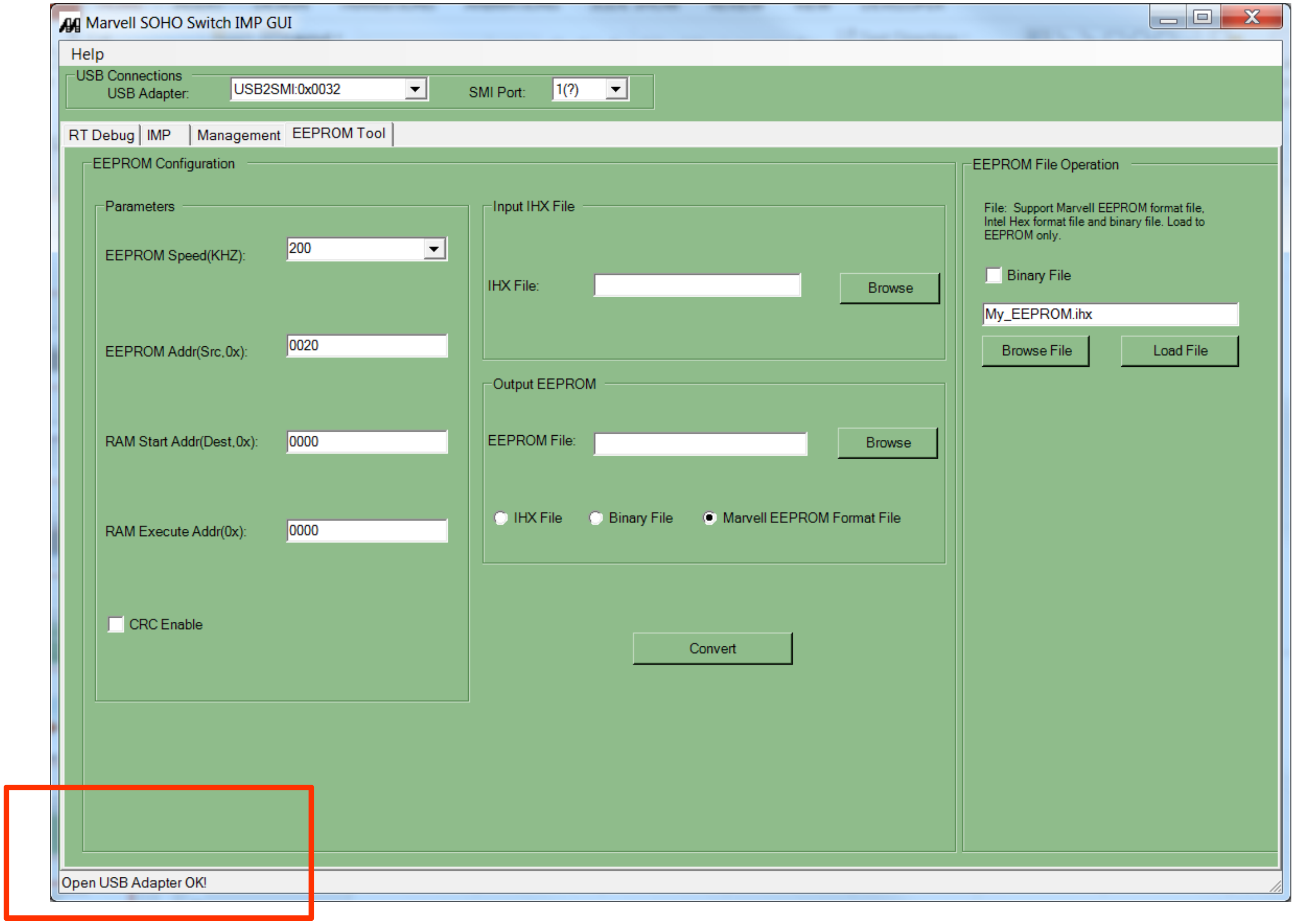
EEPROM File Operation

File: Support Marvell EEPROM format file, Intel Hex format file and binary file. Load to EEPROM only.

□ Binary File

MARVELL

Verify the EEPROM file on the board tab to program My_EEPROM.hix to the EEPROM on the board


Marvell SOHO Switch IMP GUI

USB Connections
USB Adapter: USB2SMI:0x0032
SMI Port: 1(?)

EEPROM Configuration

Parameters
EEPROM Speed(KHZ): 200
EEPROM Addr(Src,0x): 0020
RAM Start Addr(Dest,0x): 0000
RAM Execute Addr(0x): 0000
□ CRC Enable

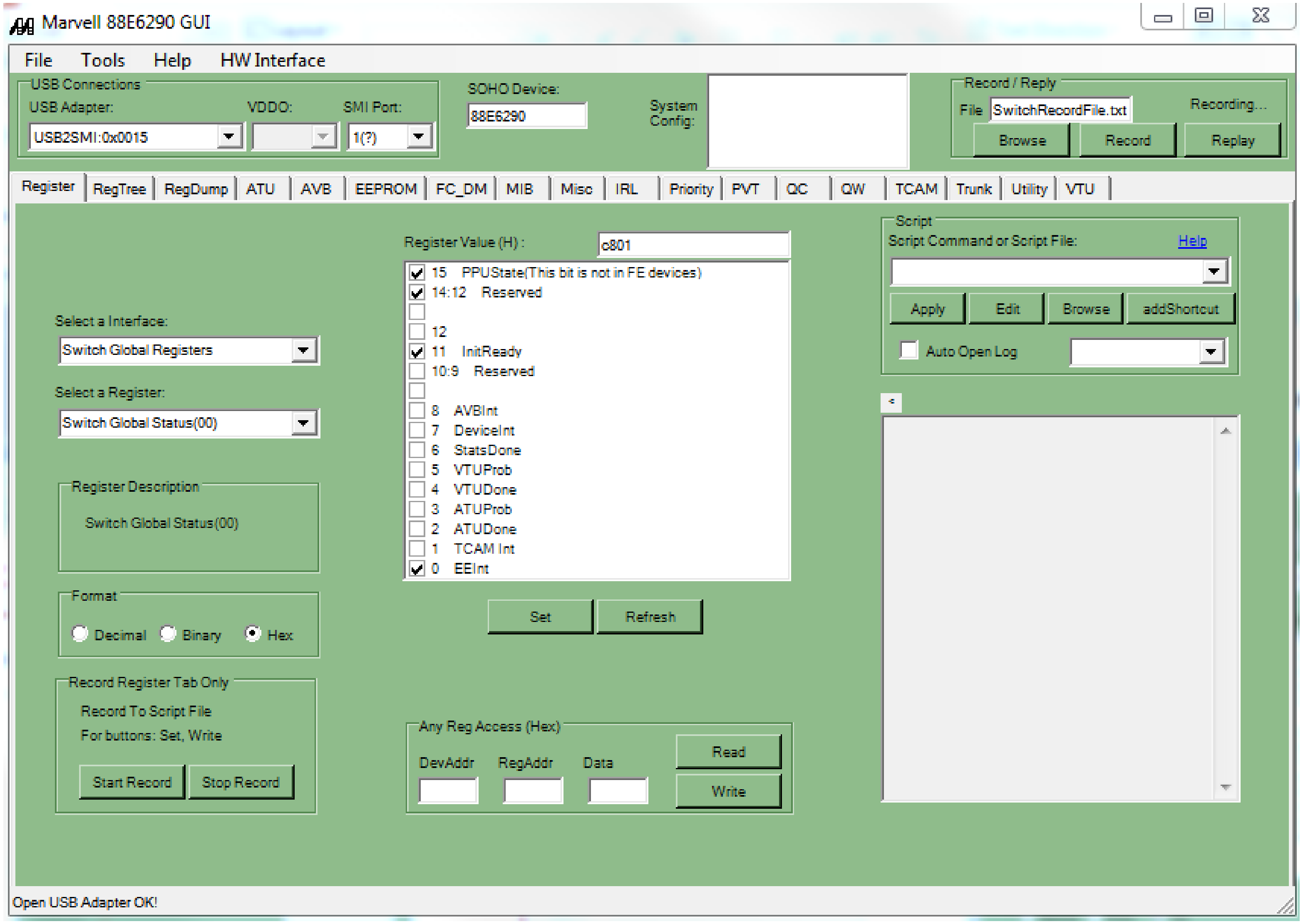
Input IHX File
IHX File:

Output EEPROM
EEPROM File:
○ IHX File ○ Binary File ● Marvell EEPROM Format File

EEPROM File Operation
File: Support Marvell EEPROM format file, Intel Hex format file and binary file. Load to EEPROM only.
□ Binary File
My_EEPROM.ihx

Open USB Adapter OK!

Reset board and verify EEPROM programmed correct registers with switch GUI



USB Connections

USB Adapter: USB2SMI:0x0015

VDDO:

SMI Port: 1(?)

SOHO Device: 88E6290

System Config:

Record / Reply

File SwitchRecordFile.txt

Recording...

Select a Interface: Switch Global Registers

Select a Register: Switch Global Status(00)

Register Description

Switch Global Status(00)

Format

○ Decimal ○ Binary ● Hex

Record Register Tab Only

Record To Script File

For buttons: Set, Write

Register Value (H) : c801

☑ 15 PPUState(This bit is not in FE devices)
☑ 14:12 Reserved
☐
☐ 12
☑ 11 InitReady
☐ 10:9 Reserved
☐
☐ 8 AVBInt
☐ 7 DeviceInt
☐ 6 StatsDone
☐ 5 VTUProb
☐ 4 VTUDone
☐ 3 ATUProb
☐ 2 ATUDone
☐ 1 TCAM Int
☑ 0 EEInt

Any Reg Access (Hex)

DevAddr RegAddr Data

Script

Script Command or Script File:

☐ Auto Open Log

Open USB Adapter OK!

MARVELL