NXP Semiconductors User's Guide

Document Number: WPR1500BUCKMPUG Rev. 0, 09/2016

WPR1500-BUCK MP Receiver V2.1 Reference Design User's Guide

1 Introduction

This document describes how to use the WPR1500-BUCK medium power receiver reference board designed by NXP Semiconductors. It supports 15 W, is compliant with the WPC Qi V1.2.2 specifications, and has the capability to support future standards. It is a low-cost reference solution that can be easily customized using the FreeMASTER tool.



Figure 1. WPR1500 reference board – BUCK

Contents

1.	Introduction	1
2.	System Features	2
3.	Package Checklist	2
4.	System Block Diagram	2
5.	Hardware Description	3
6.	Getting Started	4
7.	References	31
8.	Revision History	31



2 System Features

The WPR1500 medium power receiver reference board has the following features:

- Reference design compliant with medium power WPC Qi V1.2.2 specifications
- Input power of 3.5 V 20 V AC in peak from the transmitter through the receiver coil
- Output power of 15 W (5 V @ 3 A)
- Support of FSK communication signals from medium power transmitter
- Hardware protection of rectifier voltage, output voltage and output current
- Small PCB size (40 mm × 40 mm)
- FreeMASTER tool to enable customization and calibration

3 Package Checklist

Name	Count
WPR1500-BUCK board	1
WPR1500 debug board	1
10-pin SWD debug cable	1

Table 1. Package checklist

4 System Block Diagram

The WPR1500 medium power receiver wireless charging system is shown in Figure 2.



Figure 2. Wireless charging system overview

For the WPC Qi information, visit www.wirelesspowerconsortium.com/developers/

5 Hardware Description

5.1 Reference board block diagram





5.2 Modules explanation



Figure 4. WPR1500-BUCK board modules overview

WPR1500-BUCK MP Receiver V2.1 Reference Design User's Guide, Rev. 0, 09/2016

• Controller

The NXP WPR15xx MCU is the central controller of the WPR1500 receiver board. The chip is a higher integrated receiver controller MCU for wireless power transfer applications. The WPR15xx is an ARM[®] Cortex[®] M0+ core ASSP with NXP's UHV technology. It includes the FSK and CNC models that allow easy development for bi-directional communication architecture between the transmitter and receiver. The PGA model handles small signal, which eases the solution for foreign object detection. The USB / adapter switcher sets the priority between wired and wireless charging. The following modules are used in this application:

The following modules are used in this application:

- CNC controls the communication and provides AC protection
- High voltage input PMC module with three power modes: Run, Wait, Stop
- Programmable gain amplifier (PGA) with differential input and output
- FSK demodulation timer (FSKDT)
- WDOG with independent clock source
- Rectifier

The rectifier uses self-driven sync type. It has the following characteristics:

- Input voltage : 3.5 V 20 V AC peak
- Output voltage: 3.5 V 20 V DC
- Communication
 - The ASK differential bi-phase signal is modulated by switching the modulation capacitor
 - The FSK signal is demodulated by the CNC and FSKDT modules
- USB switch
 - The CNC module automatically detects the input voltage from wired power plugs, and switches on wired power when the input voltage is in the range of 4.5 V 5.5 V
- DC–DC converter
 - Input voltage : 5 V 21 V DC
 - Output voltage / current: 5 V DC / 3 A.

6 Getting Started

6.1 System development environment

The WPR1500 receiver board supports debugging using the IDE (IAR, Keil MDK or CodeWarrior) and FreeMASTER tools. Figure 5 shows setup of the debug connection. The debugger and the debug board are placed between the PC and the receiver board.

To download an image onto the WPR1516 chip, connect a debugger (J-Link or P&E-Multilink FX) to the SWD port of a debug board, and connect the debug board to the receiver board using a 10-pin cable. The micro interface on the debug board must be connected to the USB to get power.

To monitor the working status of the WPR1500 receiver board, short jumper J110 on the debug board, and connect the micro interface on the debug board to the PC through a micro USB cable.











For details on the J-Link debugger, visit nxp.com and search for "J-Link".

6.2 Downloading and debugging firmware

6.2.1 Connecting the J-Link debugger

Connect the J-Link debugger to the SWD port of the debug board, and then connect the debug board to the receiver board using a 10-pin cable.

NOTE

The micro interface on the debug board must be connected to the USB to receive power. Ensure that the jumper J110 is connected before the download.

The J-Link connection is shown in Figure 7.



Figure 7. J-Link connection

When the J-Link is plugged into the PC, it can be found in Windows Device Manager, as shown in Figure 8 and Figure 9.



Figure 8. J-Link debugger plugged in



Figure 9. P&E Multilink debugger plugged in

WPR1500-BUCK MP Receiver V2.1 Reference Design User's Guide, Rev. 0, 09/2016

6.2.2 Downloading an existing WPR1500 project using IAR 7.30 or later versions

To download an existing WPR1500 project using IAR, perform the following steps:

1. Set up the IAR embedded workbench.

The IAR embedded workbench tool is required. Since the receiver driver library is already included in the lib folder of the wireless charger application project, you can open the application project and build the applications directly whenever the wpr_lib.a is ready.

The configurations in MWPR1516_config.h are set for the BUCK board by default. To build this project for the LDO board, change the macro definition BOARD_OUT_TYPE from #define BOARD_OUT_TYPE OUT_TYPE_BUCK to #define BOARD_OUT_TYPE OUT_TYPE_LDO. Then rebuild all the file in the project.

The demo applications workspace files are located in:

<software_package>/build/wpr1500/iar/WPR1500_REF/WPR1500_REF.eww

2. Build a project.

Click on the "Rebuild All" button:



Figure 10. Building a project

When the build is complete, the IAR displays this information in the build window:

B_PFLASH
•
×
Errors 0, Warnings 0

Figure 11. Successfull build of a project

3. Ensure that the debugger is configured properly in the project options. Select the flash loader to support downloading of the binary to the internal flash:

Category: General Options Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver XDS 100/200/ICDI	Setup Download Attach to runnir Verfy download Suppress down V Use flash loade Voveride del	Images ing target load r(s) auit boar	Extra Options	Multicore	F	actory Settings
	Edt		rng vlasnioader	(Treescale)		Cancel

Figure 12. Flash loader configuration

When using J-Link as a debugger, select J-link/J-Trace:

le Edit View I	Project Tools Window	Help	
3 📽 🖬 🕼 8	5 X B B O O	• 4 2 2 2 0 0 0 0 0 0 0	8 🐨 PH 🗩 🛃
ACH NEVE DELAS		adc.c prx_hal.c WPR1500_REF_frdm.map WPR1500_REF.c	• 01
Files	Options for node "WPRI3 Category. Ceneral Options Runtime Checking C/C+ c Compiler Assembler Output Converter Custom Build Build Actions Unker Collogram Argel Argel Argel Coll Server IAR ROM-monitor I-Staffard Scholmentor I-Staffard I-St	OO_REF_findm* Exclose Settings Factory Settings Setup Deventioned Images Edites Other View macro file(s) Use macro file(s) Device desception file Oventio default STOLUT_DIRS-CONFIG: debugger/Freescale::MWPR1516.	' -457 'linear fitti
Log	Third-Party Driver XDS 100/200/8CDI	OK Cancel	J
•			

Figure 13. J-Link debugger configuration for driver

When using P&E Multilink as a debugger, select the PE micro:

Options for node "WPF	1500_REF_frdm"
Options for node "WPR1 Category: General Options Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver XDS 100/200/ICDI	1500_REF_frdm"
	Device description file Override default STOOLKIT_DIRS\CONFIG\debugger\Freescale\MWPR1516.(OK Cancel

Figure 14. P&E Multilink debugger configuration for driver

The P&E Multilink is configured in the PE micro settings, as shown in Figure 15.

			Factory Setting
General Options Runtime Checking C/C++ Compiler Assembler Output Converter	Setup P&E Hardware interface :	type	Communication
Build Actions	USB Multilink (FX)	USB	
Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR. RQM-monitor I-jet/JTAGjet	Reset delay ms JTAG/SWD speed 5000 kHz	Interface JTAG @ SWD	Device 1 Serial port COM1 TCP/IP Auto scan network
J-Link/J-Trace TI Stellaris Macraigor	Show settings dialog		10.0.0.1

Figure 15. Debugger configuration for PE micro

4. Download the project.

After the application is built successfully, click the "Download and Debug" button to download the application to the target device.



Figure 16. Download and Debug button

The programming of a project is shown in Figure 17.



Figure 17. Programming a project

WPR1500-BUCK MP Receiver V2.1 Reference Design User's Guide, Rev. 0, 09/2016

5. After the application is downloaded onto the target device, the debugger stops executing at the start of the main() function:



Figure 18. Stopping at main() when debugging

6.2.3 Downloading an existing WPR1500 project with KEIL µVision5

To download an existing WPR1500 project with KEIL, perform the following steps:

1. Set up the KEIL μ Vision MDK.

Because the receiver driver library is already included in the library folder and platform folder of the wireless charger application project, you can open the application project and build the applications directly whenever the wpr_lib.lib is ready.

The configurations in MWPR1516_config.h are set for the BUCK board by default. To build this project for the LDO board, change the macro definition BOARD_OUT_TYPE from #define BOARD_OUT_TYPE OUT_TYPE_BUCK to #define BOARD_OUT_TYPE OUT_TYPE_LDO. Then rebuild all the file in the project.

The demo applications workspace files are located in:

<software_package>/build/wpr1500/keil/WPR1500_REF/WPR1500_REF.uvprojx

2. Build a project.

Click the "Rebuild" button.

😨 C:\Users\B50383\Project\FSL_MPRX_WPR1500_v1.1_RC4\build\wpr1500\keil\WPR1500_REF\WPR1500_REF.uvprojx - 續ision				
File Edit View Project Flash Deb	ug Peripherals Tools SVCS Window Help			
🗄 🖻 🛃 🥥 🕉 🛍 🛍 🤊 (> 🔄 → 隆 熟 熟 課 課 //: //:: 🖄 lout_P 🕢 💽 🗟 🥐 🔍 🐠 💿 🔗 🍓 🔳 🗨 🔦			
🔮 🛅 🗃 🥔 🛄 🙀 🛛 WPR1500_	REF 🕞 🔊 🛔 🔁 💠 🐡 🏟			
Project	WPR1500 REF addr			
Pro Rebuild all target files	100 woid PTT Tack (woid):			
	101 - /***********************************			
🕀 🛄 common	102 * Global functions			
🕀 🛄 cpu	103 _************************************			
🕀 🦳 drivers	104 extern void AD IN Init(void);			
🕀 🦳 freemaster	105			
🕀 🧰 config	106 /************************************			
🕀 🧰 project	107 int main (void)			
🖃 🦢 wpr_lib	108 🗗 {			
prx_lib.h	109 //uint8_t u8Ch;			
wpr_lib.lib	110 // RTC_ConfigType sRTCConfig;			
🗄 🛵 Output	<pre>111 // RTC_ConfigType *pRTCConfig = &sRTCConfig</pre>			
WPR1500_REF.map	112 PIT_ConfigType sPITConfig1;			
	113 PIT_ConfigType *pPIT_Config1 = &sPITConfig1			
	114 115 FE ItemInfoTupe sEE ItemInfo = $(0x^35, 0x^65, 0x^6f, 0x$			
	116 //svsinit():			
	117 //printf("\nRunning the WPR1500 REF project.\n");			
	118			
	119 /* PMC initial */			
E Pro € Bo {} Fu U, Te	<)			
Build Output	Ф 🖬			
	▼			
	•			
📰 Build Output 🛛 🕅 Browser				
Rebuild all target files	J-LINK / J-TRACE Cortex L:17 C			

Figure 19. Building a project

When the build is complete, KEIL displays this information in the build window as follows.





WPR1500-BUCK MP Receiver V2.1 Reference Design User's Guide, Rev. 0, 09/2016

Ensure that the debugger configuration is correct in the project options.
 The flash loader must be selected to support downloading the binary to the internal Flash.

C Use Simulator Setting	s C Use PEMicro Dahuggar Settings
Limit Speed to Real-Time	Altera Blaster Cortex Debugger
Load Application at Startup Run to main() Initialization File:	Stellaris ICDI Signum Systems JTAGjet J-LINK /J-TRACE Cortex Initializatic ULINK Pro Cortex Debugger NULink Debugger SiLabs UDA Debugger
Restore Debug Session Settings Breakpoints Toolbox Watch Windows & Performance Analyzer Memory Display System Viewer	Restore ST-Unix Debugger CMSIS-DAP Debugger PreFast Models Debugger Watch Windows Memory Display System Viewer
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DARMCM1.DLL -pCM0+	Dialog DLL: Parameter: TARMCM1.DLL pCM0+

Figure 21. Flash loader configurations

If using the J-Link as debugger, choose J-link/J-Trace.

Liek / Trees Adapter		-		
SN: 913277973			Device Name	Move
Device: J-Link Lite-FSL	SWD	⊙ 0x0BC11477	ARM CoreSight SW-DP	Up
HW : V1.00 dll : V4.94j				Down
FW : J-Link Lite-FSL V1 compiled Ji	- · ·		10.000 F	
Port: Max Clock:	Autor O Marco	natic Detection		
Connect & Reset Options	Add		e Options Download O ache Code	ptions
Connect: Normal Reset: Norma Reset: Norma Reset after Connect	"	Ca	sche Memory 🛛 🔽 Downloa	id to Flash

Figure 22. Debugger configurations for J-Link

If using P&E Multilink as the debugger, choose the PE micro.

P&E Connection Manager - v1.28.00.00					
Please select connection interface, port, and settings for connection.					
Connection port and Interface Type Interface: USB Multilink, USB Multilink FX, or Embedded OSJtag					
Port: USB1 : Multilink Universal Rev B (PE5656514)					
Interface Detected : Firmware Version :					
Target CPU Information					
CPU: wpr1516m2					
BDM Communication Speed					
PC Parallel Port wait states : IO_DELAY_CNT = 0					
Debug Shift Speed = (0) : Multilink = 1.00Mhz , Multilink FX = 25.00Mhz , OSJTAG = 0.25Mhz ▼ BDM_SPEED = 0					
Use SWD reduced pin protocol for communications					
-MCH Internal Bus Executional (Ear programming)					

Figure 23. Debugger configurations for PE micro

4. Download the project.

After the application is built successfully, click the "Download" button to download the application to the target device.



Figure 24. Download button

Program a project.

🔀 C-USers/B50383/Project/FSL_MPRX_WPR1500_v1.1_RC4\build\wpr1500\keil/WPR1500_REF.WPR1500_REF.wprojx - 碘ision				
File Edit View Project Flash Del	oug Peripherals Tools SVCS Window Help			
🗋 🖻 🛃 🥥 🐇 ち 🛍 🔊 🥬	이 (슈 아) 😰 魏 殷 谭 菲 /// /// 👹 Jout_P 💽 🗟 🌮 🔍 (이 이 이 요) 💽 🔍			
🔗 🖻 🖓 😓 🦉 WPRISOLREF 💽 🔊 🛔 🗟 🚸 🕎 🎰				
Project R 🛛 WPR1500_REF.c] adc.c				
🖃 🍄 Project: WPR1500_REF	112 PIT ConfigType sPITConfig1;			
😑 🚂 WPR1500_REF	113 PIT_ConfigType *pPIT_Config1 = &sPITConfig1			
🗄 🛄 common	114			
🗈 🛄 cpu	115 EE_ItemInfoType sEE_ItemInfo = {0xA5, {0x65, 0xff, 0xff, 0xff,			
🕀 🛄 drivers	116 //sysinit();			
🕀 🛄 freemaster	<pre>117 //printf("\nRunning the WPR1500_REF project.\n");</pre>			
🕀 🛄 config	118			
🕀 🦾 project	119 /* PMC initial */			
🖃 🦾 wpr_lib	120 PMC_Init();			
prx_lib.h	121 / FIGEMASTER INICIAL /			
wpr_lib.lib	122 FIT FREEDERFERTER FILWARE > 0			
🗆 🚜 Output	124 #endif			
WPR1500_REF.map	125 -			
	126 /* configure RTC to 1Hz interrupt frequency */// Note: RTC is n			
	127 // DETCConfig-2016ModuloValue = 10.			
E Pro (♂Bo {} Fu U→Te	•			
Build Output				
Target info:				
Device: MUDD1516yyy				
VTarget = 1.735V				
State of Pins:				
TCK: 0, TDI: 1, TDO: 1, TMS	: 0, TRES: 1, TRST: 0			
Software-Breakpoints: 2				
Watchpoints: 2				
JTAG speed: 2000 kHz				
4				
Build Output				
Erase: 00001C00H	* JLink Info: FPUnit: 2 code (BP) slots and 0 literal s J-LINK / J-TRACE Cortex			

Figure 25. Programming a project

5. After the application is downloaded to the target device, click "Start/Stop Debug Session" to debug the code. The debugger stops executing at the start of the main() function.

🕎 C:\Users\B50383\Project\FSL_MPRX_WPR1500_v1.1_RC4\build\wpr1500\keil\WPR1500_REF\WPR1500_REF.uvprojx - 礦ision						
File Edit View	Project Flash	Debug Periph	erals Tools SVCS Window Help			
🗅 💕 🛃 Ø	X 🗅 🖺	n (e 🔶 🔿	隆 豫 豫 譯 譯 //E //E 🖄 Tout.P			
🏦 🖹 🗟 🖓 🤻	} {} {} {} {} {} {} {} {} {} {} {} {} {}	🔶 🔽 🔯 I	R 📰 🖧 👰 • 🔲 • 📴 • 🔜 • 🔜 • 🔛 • 🙀 •			
Registers	д 🔀	Disassembly				
Register	Value	108: (
Core		109:	//uint8_t u8Ch;			
R0	0x000012CB	110: /	/ RTC_ConfigType sRTCConfig;			
R1	0x200005A0		/ Ric Configive *pricconfig = &sricconfig			
R2	0x00000000					
R3	0x0000385B	WPR15	00_REF.c D adc.c D startup_MWPR1516.s			
R4	0x00003898	103	***************************************			
RG	0x00000001	104	extern void AD IN Init(void);			
R7	0x00000000	105				
R8	0×00000000	106	/************************************			
R9	0x200001A8	107	int main (woid)			
R10	0x00000000	N 108 -				
R11	0×00000000	100				
R12	0x200001F0	110	// DEC Carfigure aperconfigu			
R13 (SP)	0x200005A0	110	// RTC_ConfigType skrcconfig;			
B15 (PC)	0x000037C1	111	// RTC_ChildType processing = &srtcchild			
	0x61000000	112	PIT_ConfigType spirconfigI;			
+ Banked		113	<pre>PIT_ConfigType *pPIT_Config1 = &sPITConfig1</pre>			
+ System		114				
E Internal		115	EE_ItemInfoType sEE_ItemInfo = {0xA5, {0x65, 0xff, 0xf			
Mode	Thread	116	//sysinit();			
Stack	MSP	117	<pre>//printf("\nRunning the WPR1500_REF project.\n");</pre>			
		118				
		119	/* PMC initial */			
		120	<pre>PMC_Init();</pre>			
Project Regis	ters	4				
and respect parallely for						

Figure 26. Stopping at main() when running debugging

6.2.4 Downloading an existing WPR1500 project with CodeWarrior

To download an existing WPR1500 project with CodeWarrior, perform the following steps:

1. Set up the CodeWarrior.

As the WPR1516 chip is a new series chip for CodeWarrior, install CodeWarrior V10.6.4 with update 5 or later. To install update 5 package successfully, check for an update to let CodeWarrior install some essential components.

Because the receiver driver library is already included in the lib folder and platform folder of the wireless charger application project, you can open the application project and build the applications directly whenever the libwpr_lib.a is ready.

The configurations in MWPR1516_config.h are set for the BUCK board by default. To build this project for the LDO board, change the macro definition BOARD_OUT_TYPE from #define BOARD_OUT_TYPE OUT_TYPE_BUCK to #define BOARD_OUT_TYPE OUT_TYPE_LDO. Then rebuild all the file in the project.

The demo applications workspace folder is located in: <software package>/build/wpr1500/cw/WPR1500 REF

2. Build a project.

Click the Clean Project button to clean project, and then click the Build Project button.

10	C/C++ - WPR1500_REF/Sources/p		New	F		3
File	File Edit Source Refactor Sear □ → □ □ ●		Go Into		ndow Help	
			Open in New Window			
			Index	÷	Quick Access	g
8	😼 CodeWarrior Projects 🛛		Build Configurations	÷	WPR1500_REF.c 🐹 🗖 🗖	8
	🗄 🖧 🕒 😫 🔎		Make Targets	÷	nitial */	
	File Name		Build Project		();	9
	WPR1500_REF : FLAS		Clean Project		ASTER initial */	AY B
	5 5 FLASH	B	Сору	Ctrl+C	ER FIRMWARE > 0	
	Project_Headers	Ē	Paste	Ctrl+V	it();	
	Project_Settings	×	Delete			
	Sources Sources Sources		Move		JPPORT	
	👂 🗁 config		Rename		are PIT channel 0, enable interrupt and time	
	b 🗁 opu	ы	Import		ig->u32LoadValue = 0x02D0;//720;//Syst	
	Inversion inv	4	Export		fig->bFreeze = FALSE; fig->bModuleDis = FALSE; /*!< enab	
	👂 🔂 project		Add Files		fig->bInterruptEn = TRUE;	
	WDR1500 REE ELAS WDR1500 REE ELAS	Ş	Refresh	F5	<pre>fig->bChainMode = FALSE;</pre>	
		1	Close Project		Fig->bTimerEn = FALSE;	
			Run As	×	(PIT CHANNELO, pPIT Config);	
			Debug As	•	allback(PIT_CHANNELO, PIT_ChO_Task);	
			Profile As	Þ		
			Convert To		ire DIT channel 1 enable interrupt and time	
			Edit Linked Resources Locations		Fig->u32LoadValue = 0x1770;//0x5Dc0; /*	
			leam		fig->bFreeze = FALSE;	

Figure 27. Build the project

When the build is complete, CodeWarrior displays this information in the Console window as follows.

📳 Problems 🗐 Console 🐹 🔗 Search 📋 Memory 🖉
수 수 🔄 🔜 🗟 보 - 📬 -
CDT Build Console [WPR1500_REF]
'Invoking: ARM Ltd Windows GNU Create Listing'
"C:/Freescale/CW MCU
v10.6/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-e
abi-objdump" -h -S WPR1500_REF.elf >"WPR1500_REF.lst"
'Invoking: ARM Ltd Windows GNU Print Size'
"C:/Freescale/CW MCU
v10.6/Cross_Tools/arm-none-eabi-gcc-4_7_3/bin/arm-none-e
abi-size"format=berkeley WPR1500_REF.elf
text data bss dec hex filename
14552 212 1940 16704 4140 WPR1500_REF.elf
'Finished building: WPR1500 REF.siz'
• •

Figure 28. Build the project successfully

3. Select **Debug Configurations** to select debugger setting. Two supported debugger settings are already configured in the project.



Figure 29. Flash loader configurations

When using the J-Link as the debugger, select **WPR1500_REF_FLASH_Segger Jlink_Trace**. When using the P&E Multilink as the debugger, select **WPR1500_REF_FLASH_PnE U-MultiLink.** If you want to change some parameters, click the **Edit** button in the Target setting subpage. The configuration interface is shown as follows.

Debug Configurations							
Create, manage, and run configurations Debug or run an application to a target.							
	Name: WPR1500_REF_FLASH_Segger J-Link_Trace						
type filter text	🗈 Main 🚳 Arguments 🏇 Debugger 🤯 Source 🌆 Environment 🔲 Common 💣 Trace and Profile						
C Codewarnor WPR1500_REF FLASH WPR1500_REF FLASH	Debug session type Choose a predefined debug session type or custom type for maximum flexibility						
C WPR1500_REF_FLASH_OpenSDA	Ownload O Connect Attach Custom						
Launch Group	▼ C/C++ application						
	Project: WPR1500_REF Browse						
	Application: FLASH/WPR1500_REF.elf Search Project Browse Variables						
	▼ Build (if required) before launching						
	Build (if required) before launching						
	Build configuration:						
	Select configuration using 'C/C++ Application' Disable sute build						
	Obsable auto build Obsable auto build Obsable auto build Obsable auto build Obsable auto build						
Filter matched 6 of 6 items	▼ Target settings						
Filter by Project	Connection: WPR1500 REF FLASH Segger J-Link Trace T Edit New						
₩PR1500 REF		•					
	Apply Revert						
?	Debug Close						

Figure 30. Debug configurations

4. Download and debug the project.

After the application is built successfully, click the Debug button in the above-mentioned configuration dialog. Then the project code is downloading and the debugger stops executing at the start of the main() function.



Figure 31. Stop at main() when running debugging

6.3 Debugging the WPR1500 receiver using FreeMASTER

6.3.1 Connecting the debug cable

The FreeMASTER debug connection is shown in Figure 32. Ensure that the jumper J110 is removed before debugging, and then put the receiver on the transmitter panel.

Figure 32. FreeMASTER debug connection

6.3.2 Setting FreeMASTER project options

1. Set the symbol file for your project.

Select a symbol file from FreeMASTER Project -> Options -> MAP Files, as shown in the following figure. The IAR project is used as an example.

Options		<u> </u>
Comm M	IAP Files Pack Dir HTML Pages Demo Mode Views & Bars	
Default sy	mbol file: .\FLASH_16KB_PFLASH\Exe\WPR1500_REF.out	
File format	Binary ELF with DWARF1 or DWARF2 dbg format.	Del
List of all v	Alid .\FLASH_16KB_PFLASH\Exe\WPR1500_REF.out	New.
symbornic	o.	Del
		View
	Note: The file selected in the list will be used as default symbol file when the project is opened	
	On Load	1
	Let the user select starting symbol file	
	Synchronize variables each time the symbol file loads	
	List errors (variables using undefined symbols)	
	Always Except after project load	
	OK Cancel Apply	Help

Figure 33. Selecting a symbol file

Select a communication interface for FreeMASTER.
 Select Plug-in Module in FreeMASTER Project -> Options -> Comm, as shown in the following figure. Then select FreeMASTER Segger/Jlink Communication Plug-in.

Options	3
Comm MAP Files Pack Dir HTML Pages Demo Mode Views & Bars Communication	1
Direct RS232: Port: COM5 Speed: 57600 Timeouts	
Plug-in Module: FreeMASTER Segger/JLink Communication Plug-in Connect string: Configure	
Save settings to project file Save settings to registry, use it as default.	
Communication state on startup and on project load Copen port at startup Do not open port at startup Store port state on exit, apply it on startup	
Store state to project file, apply upon its load Advanced	
OK Cancel Apply Help	

Figure 34. Communication interface setting

6.3.3 Debugging using FreeMASTER

NXP provides the FreeMASTER GUI tool to monitor the system working status. Open WPR1500_REF.pmp with this tool, start up the RX board, and click the red Start/Stop button. For the FreeMASTER tool, visit nxp.com/Freemaster.

The FreeMASTER GUI can be used to monitor the system state and change some parameters manually if necessary. The user can get the system error code through the oscilloscope in the GUI.

Figure 35. FreeMASTER GUI tool

The control page includes three tabs: Current/PGA Calibration, FOD Calibration, and System Control. The output current calibration function is enabled when the macro ADC_CALIBRATION_FREEMASTER is set to 1.

After getting the calibration result, click the **Store to flash** button to save these parameters in flash. These parameters are used to calibrate the output current every time when the RX board is started. The internal PGA calibration function is disabled by default because of the flash limitation. When this function is enabled, the calibration result can also be saved in flash by default.

The FOD Calibration function can work offline. The user needs to measure some parameters and fulfill them in this page. When calibration is complete, the parameters that can be used in the project code are listed in the bottom area of this page.

The System Control page is used to control the system manually when RX works properly. It can set the expected Vrec voltage, enable or disable output, set the control error packet value, and so on. This function is useful to debug the system in details.

WPR1500_REF.pmp - FreeMASTER							
File Edit View Explorer Project	ct Tools Help						
: 🚅 🖬 📵 🔚 🐚 🐏 🍊 🔶 🖷	🔷 🛐 💷 🚧 🦌	n 🖈 🕫 🖸 🖬 🛉	🐇 🔤 🏘 🗄 Tahor	ma	• 8 • I	B / U ● ● ■ = = =	
Project Tree							
圖 WPR1500							
MP Parameters							
			MDDV	ErooM	ACTE	P CIII	
				геем	ASIE	K GUI	
E timing							=
timing		Current/DC	Calibration	EOD Calibrat	ion Suo	tom Control	
E Clamp		Current/FG/	A Calibration	FOD Calibrat	ion sys	tem control	
System Error							
		EOD Calibra	tion for BU	CK board	4		
MOSEET Protection		FUD Calibia		CR DUalt	A		
MOSFET power							
startup& Vrec control		In order to keep the	parameters stable	and record val	lue easily, use	er can use DC power supply to	
startup		provide input power.	Thus Vrec can be	e adjusted easil	ly. The output	current of the DC power supply and	
ault_status		viec, voul and loul i	leed to measure t	by multimeter.			
🔤 🥸 fault							
E United Back power loss			DC nouve	r cupply			
FOD parameters			DC howe	supply			
ime usage							
- Co time			Á (∇			
working Debugging		\frown	¥_	Jrbiter;	modulo	Y	
				BUCKI	nouule		
		(Coil⋒)	Rectifier	DC-D	DC 🔶 Ci	urrent Sensor	
				11 4 4	<u> </u>		
			FSK		PGOOD		
			Vrec				
		CLAMP	ASK			No.	
		CLAWF	ASK	WPR1	516	Vout	
		CLAMP	Comm		<		
						Control Circuit	
		Horo we accure the	t the input current	of MCILlis c.co	netant and d	a pot change with Vroc voltage Ma	
		arellercono	it the input current	UNICO IS a CO	instant and d	o not change with viec voltage. We	
	control p	page oscilloscope					
	Variable	Watch	Mahaa	1.1-2			-
	CtlErr	ror	0 value	DEC	0	Period	
	outpu	ut_enable	1	DEC	0		
	Iout		1154	DEC	0		
	Vout		5084	DEC	0		
	syste	m_error_flag	0	DEC	0		
	vrec_	_control_expected_value	5100	DEC	1000		
	rect_	voltage_average1	5117	DEC	1000		
	keep	control error zero flag	0	DEC	1000		
	vrec	control_manually_flag	0	DEC	1000		
	disabl	le_output_manually_flag	0	DEC	1000		
	contro	oi_error_manual_limit	255	DEC	1000		
Ready						Not connected	Scope Bad

Figure 36. Control page of the FreeMASTER GUI

6.4 Testing

6.4.1 Signals on the board

The main signals on the WPR1500 reference board are shown in Figure 37.

Figure 37. Test points on the WPR1500-BUCK board

Testing points on the WPR1500 BUCK board are as follows:

- TP6: USB input voltage
- TP9: VREC
- TP7 & TP8: GND
- TP12: DC-DC output enable

The following examples show how to measure and debug the board follow.

- 1. Figure 38 shows the input voltage and communication signal from ping phase to power transfer setup.
 - Channel 1: rectifier DC output voltage VREC
 - Channel 2: coil AC input voltage V_Coil
 - Channel 3: communication signal CTX
 - Channel 4: coil AC input current I_Coil

Figure 38. Ping process of the wireless power receiver

2. System response measurement for a 3 A load is shown in Figure 39.

Figure 39. System response for a 3 A load

3. System response measurement for adding the load gradually from 0 A is shown in Figure 40.

Figure 40. System response for adding loading gradually

4. System response measurement for reducing the load from 3 A to 0 A is shown in Figure 41.

Figure 41. System response for reducing the load from 3 A to 0 A

7 References

• NXP wireless charging solution page: nxp.com/wirelesscharging

- NXP FreeMASTER tool page: nxp.com/Freemaster
- WPC page: www.wirelesspowerconsortium.com

8 Revision History

This table summarizes revisions to this document.

Table 2. Revision history

Revision number Date		Substantive changes		
0	09/2016	Initial release		