

# Improve Time to Market Easily by Customizing Your Power Management Integrated Circuit

Franco Contadini, Staff Engineer

## Abstract

Power management integrated circuits (PMICs) are attractive to reduce the complexity and size of end applications. However, customization of the powerup setting is needed due to the default start-up sequence and output voltages not being aligned with the application's requirements. Most often a nonvolatile memory (NVM) is not available to store these settings. A low power microcontroller is a good solution with the ability and tools to program the PMIC control registers at power up without requiring firmware development. This article will address the challenge of achieving integration using a toolchain that simplifies PMIC customization and minimizes development time since no firmware development is required.

#### Introduction

Handheld devices, smart cameras, and other portable devices require highly integrated and powerful devices to reduce size and cost. The MAX77714 is an example of a highly integrated PMIC including several buck converters, low dropout (LDO) regulators, general purpose input outputs (GPIOs), and a real-time clock (RTC) that can be used to generate all supply rails needed from the application.

A flexible sequencer allows customized power-up and power-down sequences for target applications.

This PMIC is available with a couple of  $V_{out}$ /sequencer options (defined from the full part number) that don't match the application requirement. To increase flexibility, an alternative solution could use a low power microcontroller that wakes up at power on, programs device registers, and then sleeps to save power.

A medium-volume application requiring high integration needs a simple solution to use off-the-shelf devices, customizing its operation for their needs.

A hardware designer's dream is to have a tool that automatically generates firmware code, allowing them to avoid involving software resources in the early development stage, while still optimizing resource usage.

## MAX28200

The MAX28200 is a low power, low size (1.7 mm  $\times$  1.8 mm) RISC microcontroller with several peripherals. For this application, we use two GPIO pins as a bit bang I<sup>2</sup>C main.

The supply voltage is up to 3.6 V, so an always-on LDO regulator is used to power the microcontroller if the input voltage exceeds 3.6 V. The device can be programmed using the I<sup>2</sup>C node interface.

## MAX77714

The MAX77714 (Figure 1) is a highly integrated PMIC that includes four buck converters, nine LDO regulators, eight GPIOs, RTC, and a flexible power sequencer (FPS). Bucks and LDO regulators power up with default sequence and voltages, following voltage rails, can be modified using an I<sup>2</sup>C embedded interface. However, in most cases, it is important that the default sequence and voltage match the requirement of a device connected to the PMIC that could be damaged if voltages are above its maximum rating.

Power-up and power-down sequences are also important since devices such as field programmable gate arrays (FPGAs) require that voltages are enabled and disabled in the correct sequence to operate correctly. Figure 2 shows the default power-up sequence. There are eight time slots, and voltages can be enabled in any of these time slots. The device can be powered with a voltage of up to 5.5 V.



Figure 1. A MAX77714 simplified block diagram.



Figure 2. A MAX77714EWC+ default power-up sequence.

# **Device Interfacing**

P0.0 and P0.1 are used to generate I<sup>2</sup>C transactions toward PMIC, P0.2 monitors the status of GPI04, and P0.3 selects, using an SPDT switch, which is the main of the I<sup>2</sup>C (the MAX28200 or host micro). The analog switch is optional, needed only if the host micro needs to control the PMIC following power-up (see Figure 3).



Figure 3. The MAX28200 and MAX77714 interfacing with the MAX14689 and the ADP166.

At power-on reset (POR) as shown in Figure 4:

1. P0.3 is asserted low to select the device as the  $I^2C$  main for the PMIC.

2. The following desired data registers are written.

3. P0.3 is asserted high to select the host micro as the I<sup>2</sup>C main and the device enters sleep mode.

4. PO.2, which is connected to GPIO4, is used to wake up the device and reprogram registers in case of power down.

Note that GPIO4 must be asserted high at the end of the configuration.

Once the PMIC is enabled using ENO and EN1 signals, it will power up with a new sequence/voltage configuration.



Figure 4. A MAX28200 program flow.

The MAX77714 power-up and power-down sequence has been defined using the MAX77714 GUI available here.

Register values can be exported in an Excel file using the GUI Import/Export tab.

Before showing how to use the GUI EVKIT it is necessary to know which registers are important for customized startup.

# MAX77714 Register Details

#### **Bucks**

Each buck instance has three associated registers (SDx\_CNFG1, SDx\_CNFG2, and SDx\_CNFG3) that define start-up voltage, ramp rate, and operation mode.

CNFG1 set output voltage, 7 bits to 8 bits field as follows:

VSD0 =  $0.26 \text{ V} + ((\text{SDOVOUT}[6:0] - 1) \times 10 \text{ mV}) \rightarrow \text{range from } 0.26 \text{ V to } 1.56 \text{ V}$ 

VSD1 = 0.26 V + ((SD1VOUT[6:0] - 1) × 10 mV) → range from 0.26 V to 1.56 V

 $VSD2 = 0.6 V + (SD2VOUT[7:0] \times 6.25 mV) \rightarrow range from 0.6 V to 2.194 V$ 

VSD3 = 0.6 V + (SD3VOUT[7:0] × 12.5 mV) → range from 0.6 V to 3.7875 V

CNFG2 and CNFG3 registers are used to select ramp rate (2.5 mV/µs or 10 mV/µs), enable/disable active discharge, enable/disable skip mode, select response time, and overvoltage threshold.

#### **LDO Regulators**

Each LDO regulator instance has two associated registers (LDO\_CNFG1\_Lx, and LDO\_CNFG2\_Lx).

CNFG1 set output voltage:

0.8 V to 2.375 V for LD00 and LD01. VLD0 = 0.8 V + (VOUT\_LD0[5:0] × 25 mV) 0.8 V to 3.95 V for LD02, LD03, LD05, LD06, LD07, LD08 VLD0 = 0.8 V + (VOUT\_LD0[5:0] × 50 mV) 0.4 V to 1.275 V for LD04 VLD0 = 0.4 V + (VOUT\_LD0[5:0] × 12.5 mV)

CNFG2 enable/disable overvoltage clamp, enable/disable auto low power mode, enable/disable active discharge, select fast/slow startup.

#### **GPIOs**

Each GPI0 instance has one associated register CNFG\_GPI0x, which selects the direction GP0/GPI, output type (open drain/push-pull), and output drive level in case of GP0, and input drive level, interrupt behavior, and debounce configuration in case of GPI.

#### FPS

A sequence is defined from xxxFPS registers, LD00FPS to LD08FPS for linear regulators, SD0FPS to SD3FPS for buck regulators, GPI00FPS, GPI01FPS, GPI02FPS, GPI07FPS for GPI0s, and RSTI0FPS for reset.

FPS[2:0] set power-down slot (SLOTO to SLOT7)

FPS[5:3] set power-up slot (SLOTO to SLOT7)

FPS[7:6] set assigned sequencer (FPSO, FPS1)

FPSO is enabled from ENO and FPS1 is enabled from EN1

The size of the power-up and power-down slot can be customized using the MSTR\_PU\_PD register:

MSTR\_PU\_PD[2:0] for power-down slot and MSTR\_PU\_PD[6:4] for power-up slot

Possible step sizes are: 31  $\mu$ s, 63  $\mu$ s, 127  $\mu$ s, 253  $\mu$ s, 508  $\mu$ s, 984  $\mu$ s, 1936  $\mu$ s, 3904  $\mu$ s, default value for MAX77714EWC+ is 3904  $\mu$ s.

## **GUI in Action**

In Figure 5, using the GUI output voltage of SDO has been modified to 0.7 V.

SD0 Output Voltage	0x2D = 0.700 V		Read
PWM Mode Enable	Om 0 = Disabled	ĺ	Write
Converter Active-Discharge Disable	Om 0 = Active-Discharge Enabled		
FSR Active-Discharge Enable	1 = Active-Discharge Enabled		
Start-Up Ramp Rate Selection	0 = 2.5mV/µs		
Over-Voltage Threshold	1 = 116.6%		
Pulse Rejection Period	0x3 = Slow	~	
Threshold Hysteresis	0x1 = 10%	~	
Falling Threshold	0x1 = 80%	~	

Figure 5. GUI used to modify SDO voltage.

SDO power-up and power-down sequence has been modified to Slot 2 (Figure 6).

SD0FPS		
Power Down Sequence Slot Number	0x2 = Slot 2 ~	Read
Power Up Sequence Slot Number	0x2 = Slot 2 ~	Write
Regulator Elexible Power Sequencer Source	0x0 = FPS0 ~	

Figure 6. GUI used to modify SDO slot number.

GPI04 has been configured as a push-pull output, with a drive level high (Figure 7).

CNFG_GPIO4			
Output Drive	1 = Push Pull	1	Read
GPIO Direction	0 - GPO		Write
Input Drive Level	0 = Low		
Output Drive Level	1 = High		
Interrupt Configuration (Valid Only for GPI)	0x0 = Mask Interrupt	~	
Dehauses Times for Bath Edges (Jolid Only for CBI)	0x0 = No Debource		

Figure 7. GUI used to configure GPI04.

Finally, modified register values have been exported to Excel files using the Import/Export Tab (Figure 8).

🙆 Regis	ster Explorer									-	×
Read -	Write • Reset •	Сору	Paste	Deselect	Select Sho	w Bits Setting	ps •				
All	Import/Export	Search	1								
Clear	New	Load	Save	Relo	ad View	C:MAXIMMAX	7714.csv				
TUA N	X77714			Slave	Register	Name	Hex	Meaning	Description		
T.F	PMIC			PMIC	0xA4	SD0FPS	0x12				
	BUCK			PMIC	0x40	SD0_CNFG1	0x2D				
	- CLOGIC - GPIO			PMIC	0x74	CNFG_GPI04	0x09				

Figure 8. GUI used to export register values.

Below resulting .csv file (Figure 9).

Slave	Address	Name	Hex
0x38	0xA4	SDOFPS	0x12
0x38	0x40	SD0_CNFG1	0x2D
0x38	0x74	CNFG GPIO4	0x09

Figure 9. The format of the MAX77714 .csv file.

Note that if additional devices connected to the I<sup>2</sup>C have to be programmed during power-on you can append additional commands to the .csv file using the same format.

### Creating I<sup>2</sup>C Main Firmware from a .csv File

The MAX28200 GUI is available here and can be used to program the device with the following steps:

1. Build a .hex file from a .csv file (Figure 10).

2. Programming device (Figure 11 and Figure 12).

MAX28200 16-bit Micro EV Kit	
File Tools Options Help	
Build Boot Loader Scripting Error Log	
Source File (.asm or .csv)	
C:\MAXIM\MAX77714.csv	
Ordent File	
Output File	
C:\MAXIM\MAX77714.hex	
Build Clean Build Status	
Processing C: MAXIM/MAX77714.asm	^

Figure 10. Build a .hex file from the original .csv file.

MAX28200 16-bit Micro EV Kit	-	×
File Tools Options Help		
Open Intel Hex File		
Load Script Exit		
DS9400 not detected, searching for DS9481P-300		 
DS9481P-300 Entering 12t Mode DS9481P-300 detected!		

Figure 11. A programming device with the selected script.

-	-		10-01	It IVIN	10 E	V KIL														1
Fil	e lo	ols	0	ptior	IS	Hel	p													
Bui	ild	Boo	Lo	ader	S	crip	ting	E	ror l	og										
Pre	ogram/	Veri	fy																	
Int	tel He	ex 1	File	e Co	onve	erte	ed t	to 1	AX2	282	00 1	for	nat							
BC	Addr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			
10	0000	01	0B	69	OC	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
10	0050	3A	DA	3A	DA	00	2B	01	29	00	40	OA	DA	3A	DA	3A	DA			
0	0060	8D	8C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF			
0	0200	05	6D	FF	4D	OD	8C	97	80	27	OB	10	6D	80	97	01	2C			
0	0210	FD	4D	2A	DA	F5	3D	OD	8C	80	87	06	2C	F1	3D	17	80			
0	0220	EF	3D	97	80	80	87	FA	6C	OD	8C	EA	3D	87	80	E8	3D			
0	0230	97	80	E6	3D	07	80	E4	3D	17	80	E2	3D	80	97	OD	80			
0	0240	07	80	DE	3D	97	80	DC	3D	87	80	DA	3D	OD	8C	00	4B			
0	0250	07	80	87	80	D5	3D	D7	3D	80	87	D2	3D	17	80	DO	3D			
0	0260	07	80	00	4B	87	80	OD	8C	02	2C	07	80	02	0C	87	80			
0	0270	3A	DA	C6	3D	02	2C	80	87	02	OC	80	87	2A	DA	03	2C			
0	0280	C2	3D	17	80	BD	3D	OD	80	08	7D	8A	8A	5A	SA	EC	3D			
0	0290	02	2C	FC	5D	DC	3D	OD	80	9F	8A	00	2B	OA	89	BO	3D			
0	02A0	AF	3D	C3	3D	11	2C	89	8A	EF	3D	02	6C	C9	3D	F7	OC			
0	02B0	9F	8A	EA	3D	03	6C	C4	3D	AF	8A	Fl	0C	9F	8A	E4	3D			
0	02C0	03	6C	BE	3D	AF	8A	EB	OC	OD	80	E6	3D	00	2C	<b>B</b> 8	3D			
0	02D0	OD	80	10	4E	00	6B	00	70	00	4B	03	00	03	00	00	2B			
0	02E0	77	88	3A	DA	3A	DA	3A	DA	00	4B	B7	80	27	80	00	4B			
0	02F0	27	80	87	D8	00	40	00	2B	04	20	00	2B	00	29	03	2B			
0	0300	2E	3F	30	7F	00	4F	04	20	37	80	86	3D	DE	3D	00	2B			
0	0310	9F	F9	4F	8A	03	4A	OA	CF	09	78	F8	70	<b>B</b> 7	80	80	A7			
0	0320	FE	6C	3A	DA	00	6B	C7	FO	3A	DA	3A	DA	E8	0C	38	A4			
0	0330	12	FF	38	40	2D	FF	38	74	09	FF	FF	FF	FF	FF	FF	FF			
	File:	MAX	777	14.h	ex		1	DS	9481	1P-3	00 (1	W	2.3		Î		P	ort: COI	<b>V</b> 17	

File Tools Options Help

Build Boot Loader Scripting Error Log

Program/Verify

//Program MAX28200 //Verify MAX28200

//Programming Successful!
//Elapsed Time = 0.93 seconds

Figure 12. A programming device with a data pattern.

# Test of the Proposed Solution

Once programmed with the MAX77714.hex at power-up, P0.3 is asserted low to connect the MAX77714 I<sup>2</sup>C lines to the MAX28200, and the SD0 and GPI04 registers are programmed with new values (Figure 13).

12C - C	n SDA/SCL Ø	C) [] [] ( Deta -	A4 🚺 Deta - 12	1 <b>1</b> )((	)     ( Deta - 40	Døta - 2D	<u>)</u> -( <u></u> )	Dota - 74	( Data - 09 ) ()
-0.1 ms	0.0	0.1	0.2		0.3	0.4	0.5	0.6	0.7
KL IDA 20.3									
Packet	Start Time	End Time	Address Bits	Address	Address + R/W	R/W	Address ACK	Data	Data ACK
0	-1.812 µs	237.1 µs	7	1C	38	Write	0	A4 12	0.0
	250.7 µs	489.7 µs	7	1C	38	Write	0	40 2D	0.0
			-						

Figure 13. The MAX77714 SDO and GPIO4 registers are programmed with new values.

Now when the MAX77714 is enabled, asserting ENO high, SDO powers up on Slot 2 at 0.7 V as programmed. Figure 14 shows power-up before register writing (a) and after register writing (b).



Figure 14. SDD power-up before and after registering writing. (a) Power-up before register writing. (b) Power-up after register writing.

# Conclusion

The MAX28200 is an attractive microcontroller solution to provide power-up settings for a PMIC. The tools are available to automate this process without a designer writing a single line of code. This is a very attractive solution for hard-ware designers not familiar with software that can use a microcontroller as a black box without involving a firmware designer in this process. A designer can match their integration target using the off-the-shelf device customized for a specific application.

## About the Author

Franco Contadini has over 35 years' experience in the electronic industry. After 10 years as a board and ASIC designer, he became a field applications engineer supporting industrial, telecom, and medical customers and focusing on power and battery management, signal chains, cryptographic systems, and microcontrollers. Franco has authored several application notes and articles on signal chains and power. He studied electronics at ITIS of Genova.

Engage with the ADI technology experts in our online support community. Ask your tough design questions, browse FAQs, or join a conversation.



SUPPORT COMMUNITY

Visit ez.analog.com



For regional headquarters, sales, and distributors or to contact customer service and technical support, visit analog.com/contact.

Ask our ADI technology experts tough questions, browse FAQs, or join a conversation at the EngineerZone Online Support Community. Visit ez.analog.com. ©2024 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.