GM8136

IRDA DEVICE

User Guide
Rev.: 1.0
Issue Date: August 2014

REVISION HISTORY

GM8136 IRDA User Guide

Date	Rev.	From	To
Aug. 2014	1.0	-	Original

Printed in Taiwan 2014

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
http://www.grain-media.com

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This chapter contains the following sections:

1.1 General Description

The Infrared Data Association (IRDA) device is used for the remote control. A GPIO pin will be selected as an input signal to the IRDA device. When a button for the remote control is pressed, the IRDA receiver connected to the GM8136 board will generate the interrupt through the selected GPIO pin to the IRDA device. The driver will receive the button code in the interrupt service routine and report it to the user application.

1.2 Suggested Readers

Software designers who need to understand the application and driver are suggested to read this document.

Chapter 2

Programming Model

This chapter contains the following sections:

2.1 Introduction

The register settings are addressed in Section 2.2 through Section 2.7. If the register settings listed in Section 2.2 though Section 2.7 of this document has discrepancies with the contents in the GM8136 data sheet, users should use the register settings shown in the GM8136 data sheet. Registers listed in Table 2-2 should be changed with the remote control features listed in Table 2-1. Registers listed in Table 2-2 are embedded in the source code of the driver.

Users should set bits[20:16] of the param2 register (Offset = 0x0C) as the number of the input GPIOs. Since GPIO 3 is in the GM8136 EVB, to use GPIO 3, the Multi-function Port Setting Register 0, bit[7:6] (Offset = 0x50) of the system control unit should be set to 0x01, which can be achieved through irda_SetPinMux() in the source code of the driver.

Table 2-1. Remote Control Features

Feature	Remote Control Type				
	NEC	RCA	SHARP	Phillips RC-5 Protocol	Nokia NRC17
Carrier frequency (kHz)	28	56	38	36	38
Mode selection	Pulse distance modulation	Pulse distance modulation	Pulse distance modulation	Bi-phase coding (Aka Manchester coding)	Bi-phase coding
Bit time	1.125 or 2.25	1.5 or 2.5	1 or 2	1.778	1
Command length	8	8	5	6	8
Address length	8	4	8	5	4

Table 2-2. Remote Control Registers

Register	Remote Control Type				
	NEC	RCA	SHARP	Phillips RC-5 Protocol	Nokia NRC17
Param0 (0x04)	0400_0032h	0400_0032h	0FC0_001Eh	0110_0000h	0130_0017h
Param1 (0x08)	0070_0190h	0070_0190h	0076_0FC0h	0087_0000h	004B_00F8h
Param2 (0x0C)	0000_1F77h	0000_1F77h	0008_1E77h	0010_1F77h	001F_3F77h
Status (0x10)	0000_0003h	0000_0003h	0000_0003h	0000_0003h	0000_0003h
Control (0x00)	0000_0001h	0000_0011h	0000_0003h	0000_0006h	0000_0009h

2.2 Control Register (Offset = 0x00)

Table 2-3. Control Register (Offset = 0x00)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:5]	-	-	Reserved	-	-
4	MSB_F	R/W	MSB first 1: ir_data[31] is the MSB bit. 0: ir_data[0] is the MSB bit.	0x0	HR/WR
3	bi_phase	R/W	Bi-phase mode selection	0x0	HR/WR
2	iil	R/W	Inverse input level	0x0	HR/WR
1	NSCD	R/W	This bit indicates that no start code will be detected.	0x0	HR/WR
0	enable	R/W	Enable IRDET	0x0	HR/WR

2.3 Param0 Register (Offset = 0x04)

Table 2-4. Param0 Register (Offset = 0x04)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	timeout_th	R/W	Timeout threshold	0x0	HR/WR
[14:0]	high_th0	R/W	Logic-high threshold	0x0	HR/WR

2.4 Param1 Register (Offset = 0x08)

Table 2-5. Param1 Register (Offset = 0x08)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	low_th1	R/W	Logic-low threshold 1	0x0	HR/WR
[15:0]	low_th0	R/W	Logic-low threshold 0	0x0	HR/WR

2.5 Param2 Register (Offset = 0x0C)

Table 2-6. Param2 Register (Offset = 0x0C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:21]	-	-	Reserved	-	-
[20:16]	channel_sel	R/W	This bit selects the GPIO pin to be the IR input. 00000: GPIO[0] 00001: GPIO[1] 00010: GPIO[2] --- 11111: GPIO[31]	0x0	HR/WR
[13:8]	fifo_size	R/W	fifo_size-1	-	-
[7:0]	clk_div	R/W	Clock divider The source of the clock divider is 12 MHz.	0x0	HR/WR

2.6 Status Register (Offset = 0x10)

Table 2-7. Status Register (Offset = 0x10)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:8]	-	-	Reserved	-	-
[7:2]	data_amount	R/W	This field indicates the data amount.	-	-
1	repeat	R/W	Repeat a command	-	-
0	ir_int	R/W	Data arrival interrupt Write 1’b1 to clear	0x0	HR/WR

2.7 IR DATA Register (Offset = 0x14)

Table 2-8. IR DATA Register (Offset = 0x14)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	IRDATA	R/W	IR receive data	0x0	HW/WR

Chapter 3

Interface

This chapter contains the following sections:

3.1 Included File

/arm-linux-3.3/module/include/irda/irda_api.h

3.2 Usage of IRDA Driver

After configuring the Linux kernel, users can refer to "arm-linux-3.3/module/irda" for the source code of the IRDA module and build the irda.ko module. Then, insert this module to the device and create an IRDA device node with the "mdev -s" command, as shown in Figure 3-1.

The IRDA module supports GPIO 1 ~ 3 , users can choose by using "insmod irda.ko gpio_num".

```
# insmod irda.ko
 ************************************
 * Welcome to use irda_dev,  *
 ************************************
 * dev_data_alloc_specific done
 * gpio 3, protocol 0
 * driver_probe done, irq_no 48 0x90CE2000, io_padr 0x91600000 0x832438E0
# mdev -s
```

Figure 3-1. Insert IRDA Module

"arm-linux-3.3/module/irda/test" is the sample code for the IRDA module usage. The table of the button code for the NEC remote control is embedded in irda_test.c.