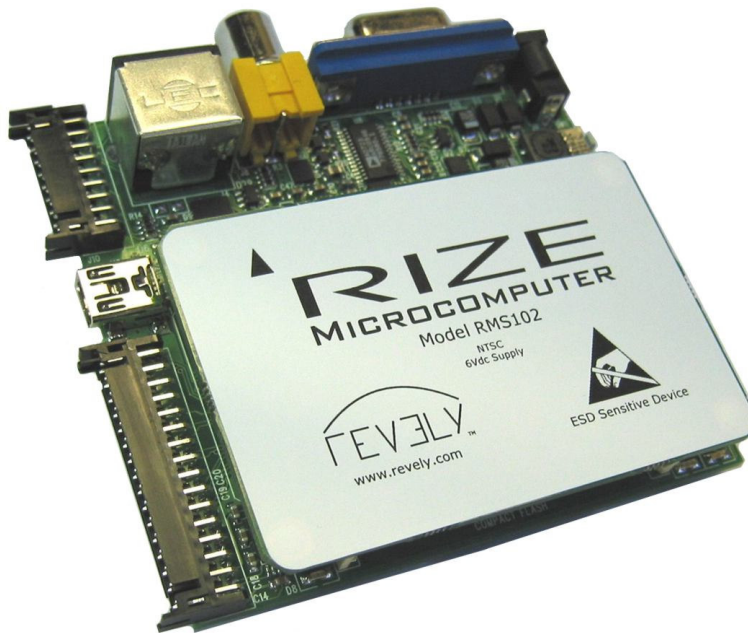




RIZE Microcomputer

Model RMS102

Technical Manual



RIZE RMS102 Technical Manual

© 2006 Revely Microsystems

Revision 1.00

Important Information

RMS102 is sensitive to ESD (Electro-Static Discharge).
Handle using ESD safety precautions.

Observe electrical specifications carefully.

RMS102 is not intended for use in critical equipment intended
to sustain or protect life or prevent injury.

All software is provided on an as-is basis.



RIZE™ is a trademark of Revely Microsystems
CompactFlash is a registered trademark of SanDisk Inc.

Table of Contents

OVERVIEW.....	4
SYSTEM ELECTRICAL SPECIFICATIONS	4
Overview	4
Parameter Table	4
I/O EXPANSION CONNECTOR.....	5
Overview	5
Electrical.....	5
SYSTEM CONNECTOR.....	6
Overview	6
i.MX JTAG port.....	6
Other Useful Signals.....	6
Electrical.....	6
MEMORY MAP	7
Overview	7
Memory Space Diagram	7
PERIPHERAL I/O CONTROLLER	8
Overview	8
Registers	8
I/O Controller Config Register.....	8
VIDEO.....	8
Overview	8
Video Formats.....	9
VGA Output.....	9
TV Output	9
Color Modes.....	9
12-bit Color.....	9
8-bit Paletted Color	10
COMPACTFLASH.....	10
Overview	10
Access	10
Interface Signals.....	11

Overview

RMS102 is a unique computer platform which combines a wide range of peripherals in a small low-power format. This document details RMS102's hardware capabilities.

The core of RMS102 is a Freescale i.MXL device containing an ARM920T processor. t RMS102 hardware peripherals are implemented in the i.MXL device and in a range digital and analog circuits.

RMS102 consists of two circuit boards. The processor board (also called the MXM) contains the i.MX processor, Flash, SDRAM and support components. The platform board has all the I/O connectors, memory slots and other peripheral circuits. There should be no reason to separate the two cricuit boards, however if they are disconnected make certain that they are reinserted with the correct orientation!

System Electrical Specifications

Overview

RMS102 operates from a 5-6Vdc power supply. There are three internal power supply circuits to generate the following voltages:

- 1.8Vdc for ARM Core (not accessible to user)
- 3Vdc for ARM Peripherals and I/O, Memory, most RMS102 circuitry and peripherals
- 5Vdc for PS/2 devices

The 5Vdc supply uses a LDO linear regulator and is used only for the PS/2 port. When operating RMS102 from a supply less than 5.6V, the PS/2 supply rail will be 0.6V less than the supply voltage. Revely has found that mice and keyboards accommodate the low voltage satisfactorily as it's still within their operating tolerances.

The 3V rail is generated by a switching power supply. Up to 400mA is available for powering CompactFlash cards and user interface circuits.

Precautions

The power supply voltage should never exceed 8.0Vdc as RMS102 does not have over-voltage protection. RMS102 is reverse polarity protected.

Parameter Table

Parameter	Minimum	Typical	Maximum
DC Input Voltage	5.0Vdc	6.0Vdc	8.0Vdc
DC Input Current Fcpu=192MHz	60mA @ 6V	220mA @ 6V	550mA @6V
PS/2 Port Output Voltage	-	5.0Vdc	-
PS/2 Port Output Current	-	70mA	200mA
Peripheral Supply Voltage	3.05V	3.15V	3.25V
Peripheral Supply Current			400mA

I/O Expansion Connector

Overview

The I/O expansion connector has a range of serial interfaces available for user applications:

- I²C Bus
- RS232 Level UART (Uart 2)
- Logic Level UART (Uart 1)
- SPI Interface
- SSI Interface for Audio CODEC

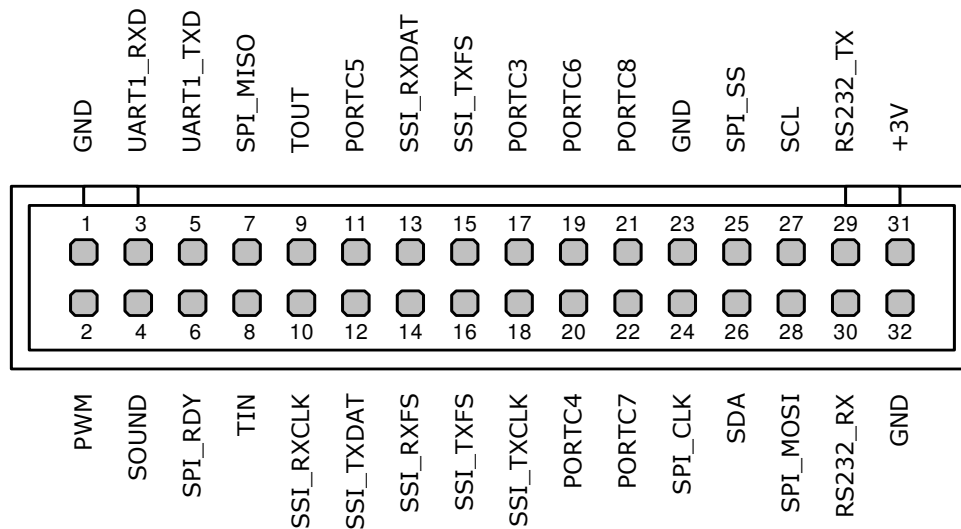
Other signals are:

- PWM wave-modulated audio channel
- Six general purpose I/O lines
- General purpose timer in/out
- PWM output
- Power and Ground

To make interfacing simple, internal RMS102 circuits use none of the above signals. The user is free to modify the peripherals settings. The rizeLib library contains some functions for accessing expansion peripherals. For complete peripheral information refer to Freescale i.MXL technical documentation.

Electrical

All signals are 3V CMOS logic. They are not +5V tolerant but can be appropriately interfaced to 5V logic. Signals do not have special ESD or electrical over-stress protection.



Pin View of RMS102 I/O Expansion Port

System Connector

Overview

The system connector contains the i.MX JTAG port and other signals used mostly for factory configuration.

i.MX JTAG port

The on-board Embedded Cross-Connect circuit normally controls the i.MX JTAG signals. At power-on, if a USB cable is not connected, the Embedded Cross-Connect will release the JTAG signals. This allows external debug devices to assume control of the i.MX JTAG port.

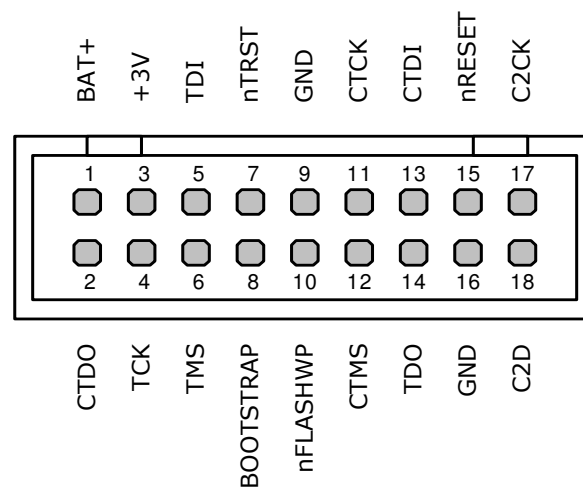
Other Useful Signals

Users may connect to these signals (other signals should be left unconnected). In normal operation no connections are required.

Signal	Use
BAT+	External power input. BAT+ and the main DC input are connected through common cathode diodes. The higher of the two voltages will supply current to RMS102.
+3V	Can be used to power external circuits
BOOTSTRAP	Connect to GND, then reset CPU to run i.MX on-chip bootstrap program. Revely's SC102 cable includes a jumper for this purpose.
nFLASHWP	Sectors in the 2MB bootflash can be locked down for security. Pull this pin low to over-ride lock-down.
nRESET	i.MX reset input. Drive with open-drain type driver.

Electrical

All signals are 3V CMOS logic. They are not +5V tolerant but can be appropriately interfaced to 5V logic. Signals do not have special ESD or electrical over-stress protection.



Pin View of RMS102 System Port

Memory Map

Overview

The i.MX ARM9 CPU has a 32 bit address space. RMS102 uses address lines A0-A23 and several chip selects for memory and external peripheral accesses.

Chip Select Assignments:

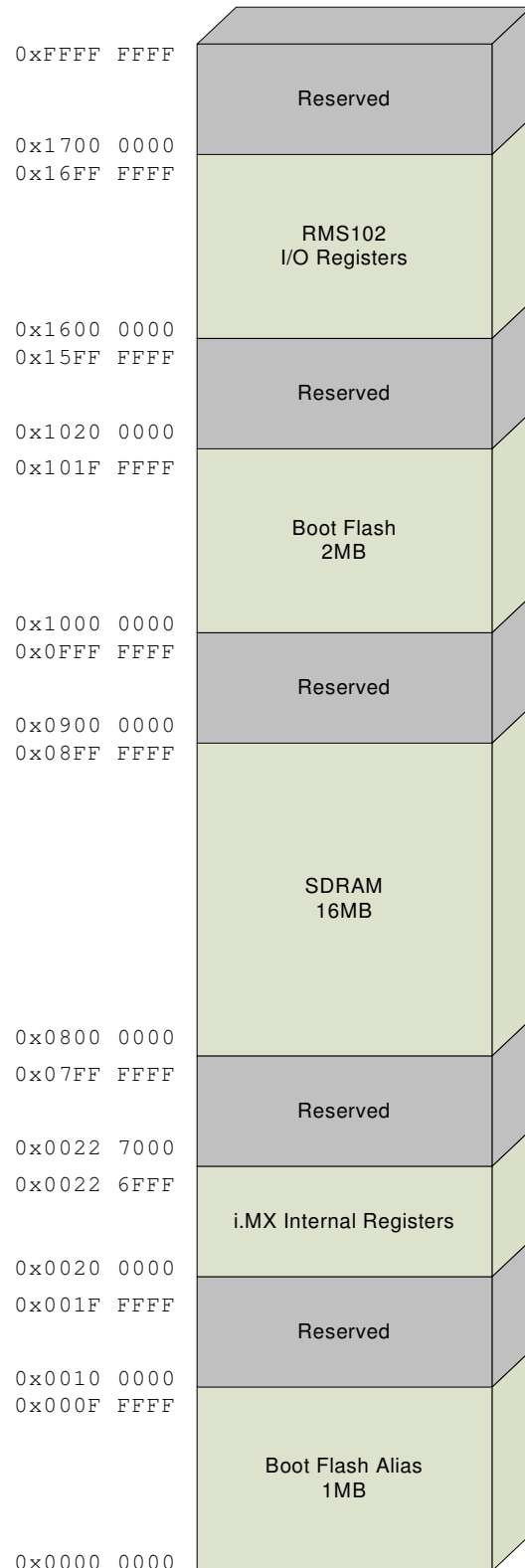
- nCS0 enables Boot Flash accesses
- nCS2/nDCS0 enables SDRAM accesses
- nCS5 enables peripheral I/O accesses

Memory chip selects are self-explanatory. The SC102 CD contains the source for configuring the registers that control the timing of these signals.

Memory Space Diagram

The entire ARM physical memory map is shown in the diagram to the right. Note that the bottom 1MB section aliases to either the Boot Flash or the i.MX's internal Bootstrap ROM, depending on the state of the BOOTSTRAP signal during reset.

The ARM920 has an MMU which can be configured to map these physical addresses to virtual addresses as required. In Revely examples, the MMU is either bypassed, or configured to directly map virtual addresses into physical memory.



Peripheral I/O Controller

Overview

RMS102's peripheral I/O controller implements the following functions:

- CompactFlash Timing
- CompactFlash Data bus buffering
- Dual PS/2 ports
- Video vertical and horizontal sync signals
- LED control
- Sound signal multiplexing

The peripheral I/O controller is accessed as 16-bit wide registers using nCS5 at 0x16000000. For most registers only D0..D7 (low byte) is used.

Registers

Address	Register	Notes
0x16000000	Compact Flash Data	D0..D7 read/write
0x16000002	I/O Controller Config	D0..D7 write only
0x16000004	PS/2 Port 1 Data	D0..D15 read/write
0x16000006	PS/2 Port 2 Data	D0..D15 read/write

I/O Controller Config Register

Bit	7	6	5	4	3	2	1	0
Name	MUTE	CF RESET	CF Register	Force Clock	Force Data	LED3	LED2	LED1
R/W	w	w	w	w	w	w	w	w

Name	Description
MUTE	Set to disable digital signal to internal piezo speaker and enable signal to SOUND pin on I/O Expansion port.
CF RESET	Set to hold Compact Flash reset signal active
CF Register	Set to select CF main memory, clear to select attribute memory
Force Clock	Set to force PS/2 port clock signal low
Force Data	Set to force PS/2 port data signal low
LED1..3	Set to turn on respective LED

There are two versions of the peripheral I/O controller, one for NTSC and one for PAL. To determine which version, read bit 8 of Compact Flash Data register. A '1' indicates PAL, a '0' indicates NTSC.

Video

Overview

RMS102 generates video signals in 4096 colors at up to 640x480 VGA resolution. Generating video signals requires a large number of timing settings and several clocks and PLLs. Fortunately Rizelib functions do all this for you. All you need to do is choose the mode. Three timing modes are supported in software and hardware.

- VGA 640x480 60Hz (25.175MHz pixel clock)
- NTSC 640x200 60Hz non-interlaced (25.175MHz pixel clock)
- PAL 640x200 50Hz non-interlaced (25.175MHz pixel clock)

In addition to the 640 pixels of display information in each line, i.MX is configured to add an additional 8 pixels before and after each line. These pixels are always set to black. Therefore while the line resolution is 640 pixels, the memory pitch is $640 + 8 + 8 = 656$. Assuming 8-bit color mode, you should add 656 to a pixel's address to access the pixel direct below it.

Video Formats

VGA Output

The VGA format should be compatible with all VGA monitors, both CRT and LCD. There may be minor scaling artifacts when using LCD panels, as it's highly unlikely the panel is being run in its native resolution.

TV Output

RMS102 is factory assembled for either NTSC or PAL TV out. The user cannot change in software or the TV standard. An Analog Devices AD723 video encoder converts the analog RGB information into the appropriate composite video signals. The encoder does not perform scaling, so the i.MX must be programmed with the correct timing.

A control signal, mapped to GPIO controls whether the mode is standard or phase alternating "PAL-M" or combination N as used in South America (note that this is presently implemented by not tested).

640x200 resolution was selected for the example code because it simplifies transitioning to/from VGA graphics formats. Depending on the TV, the left and right portions of the display may not be visible, especially on small CRT TVs.

The TV output is non-interlaced – that is, the same image is output for both odd and even fields.

Other TV resolutions are possible by reprogramming the i.MX LCD controller and PLL.

Color Modes

RMS102's video circuit incorporates a 12-bit DAC (4 bits each for Red, Green and Blue) to support 8 bit paletted color and 12 bit true-color. 8-bit color uses a 256x12bit color look-up table to generate a palette of 256 colors.

12-bit Color

In true color mode, the color look-up table is bypassed and the i.MX HCLK system clock is operated at 4x the pixel clock ($25.175\text{MHz} \times 4 = 100.7\text{MHz}$). Each pixel on the display requires 16 bits. A complete VGA screen requires $656 \times 480 \times 2 \text{ bytes} = 615\text{KB}$ of RAM.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sig	R3	R2	R1	R0	-	G3	G2	G1	G0	-	-	B3	B2	B1	B0	-
Color	Red					Green						Blue				

12-bit color is best for displaying bitmaps and pictures, although it uses more memory and bus bandwidth than 8-bit mode.

8-bit Paletted Color

8-bit paletted mode uses the i.MXL's color look-up table to create a palette of 256 12-bit colors. The advantages of 8-bit mode are lower RAM requirements, less bus loading and the ability to do clever image effects by manipulating the color palette (The plasma.c demo code is an excellent example of this).

The draw backs are lack of color depth and a requirement that the i.MX HCLK be run at 5x the pixel clock ($25.175\text{MHz} \times 5 = 125.875\text{MHz}$). To achieve this operating speed, the i.MX bus must be over-clocked by approximately 25%,. The CPU core can remain within specification at 200MHz as it has a separate PLL to generate its clock. Revely has found no issues with operating in this mode within a normal range of room temperatures. As an added bonus, the extra bus speed boosts overall system performance.

The RIZE software library contains functions for configuring and adjusting the palette.

CompactFlash

Overview

RMS102 incorporates a Type II CompactFlash slot for memory and peripheral cards such as 802.11 type cards. The Peripheral I/O Controller chip handles retiming of ARM bus signals to meet the CompactFlash standard, as well as buffering of the data bus.

Access

Data to/from the CompactFlash slot is accessed at 0x16000000 in the memory map. Because most data transfers take place at a single CF card address, the address bus to the card is controlled by i.MX GPIO Port A to simplify timing and address buffering. For example, to read data from the CF card once it is configured, the following steps are used:

1. Set PORTA to CF COMMAND register address
2. Write READ command to 0x16000000
3. Set PORTA to CF DATA register address
4. Read a byte from 0x16000000 and store
5. Repeat step #4 once for each byte to be read

Once the command is sent and the CF data register selected, continuous read/writes can be conducted using polling or a DMA operation.

RMS102 uses Memory-mode control signals (nOE and nWE) to access CF cards. IO mode (nIOR and nIOW) signals are connected by not used.

Both fixed timing and DTACK terminated memory accesses are supported.

Interface Signals

Pin	Signal	Pin	Signal
1	GND	26	
2	D3	27	
3	D4	28	
4	D5	29	
5	D6	30	
6	D7	31	
7	nCE1	32	+3V
8	A10	33	GND
9	nOE	34	IOR
10	A9	35	IOW
11	A8	36	nCFWE
12	A7	37	READY
13	+3V	38	+3V
14	A6	39	GND
15	A5	40	
16	A4	41	RESET
17	A3	42	DTACK
18	A2	43	
19	A1	44	nCFREG
20	A0	45	
21	D0	46	
22	D1	47	
23	D2	48	
24		49	
25		50	GND

Audio

Overview

RMS102's internal piezo speaker generates a range of tones by using the i.MX's Pulse-Width Modulator (PWM) module in Tone mode. In this mode, the module is configured to generate a fixed duty-cycle square wave at the required frequency. Once the frequency is loaded it will continue without CPU overhead until either stopped or modified.

The PWM module also supports wave-synthesis. In this mode a stream of audio wave data should be feed into the duty-cycle register. Modify the I/O controller's Config register to disable the piezo speaker and route the PWM signal to the I/O connector. A low-pass filter and amplifier are all that is required to complete the wave-synthesiser. Refer to i.MX Reference Manual for more information.

Appendix A – Revely ASCII Table

The software examples for the RMS101 computer use a modified ASCII table. In order to provide a unique code for special keys (ie. Function and Arrow keys) some traditional (but seldom used) ASCII codes have been used. This makes application access to special keys much easier. With this system there is no need to use special function calls – getch() works for all keys.

Actually, there are a few special keys currently unimplemented (Left and Right Windows, Volume etc), but everything else is there. Thanks to the PS/2 scan code system; the keyboard to ASCII function source-code can be modified further to meet new requirements.

Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII
0	00	NUL	43	2B	+	86	56	V
1	01	Up Arrow	44	2C	,	87	57	W
2	02	Down Arrow	45	2D	-	88	58	X
3	03	Left Arrow	46	2E	.	89	59	Y
4	04	Right Arrow	47	2F	/	90	5A	Z
5	05	Pg Up	48	30	0	91	5B	[
6	06	Home	49	31	1	92	5C	\
7	07	-	50	32	2	93	5D]
8	08	-	51	33	3	94	5E	^
9	09	Tab	52	34	4	95	5F	_
10	0A	LF	53	35	5	96	60	`
11	0B	End	54	36	6	97	61	a
12	0C	Pg Down	55	37	7	98	62	b
13	0D	CR	56	38	8	99	63	c
14	0E	Insert	57	39	9	100	64	d
15	0F	F1	58	3A	:	101	65	e
16	10	F2	59	3B	;	102	66	f
17	11	F3	60	3C	<	103	67	g
18	12	F4	61	3D	=	104	68	h
19	13	F5	62	3E	>	105	69	i
20	14	F6	63	3F	?	106	6A	j
21	15	F7	64	40	@	107	6B	k
22	16	F8	65	41	A	108	6C	l
23	17	F9	66	42	B	109	6D	m
24	18	F10	67	43	C	110	6E	n
25	19	-	68	44	D	111	6F	o
26	1A	-	69	45	E	112	70	p
27	1B	Escape	70	46	F	113	71	q
28	1C	-	71	47	G	114	72	r
29	1D	-	72	48	H	115	73	s
30	1E	-	73	49	I	116	74	t
31	1F	Num Blnk 5	74	4A	J	117	75	u
32	20	SP	75	4B	K	118	76	v
33	21	!	76	4C	L	119	77	w
34	22	"	77	4D	M	120	78	x
35	23	#	78	4E	N	121	79	y
36	24	\$	79	4F	O	122	7A	z
37	25	%	80	50	P	123	7B	{
38	26	&	81	51	Q	124	7C	
39	27	`	82	52	R	125	7D	}
40	28	(83	53	S	126	7E	~
41	29)	84	54	T	127	7F	DEL
42	2A	*	85	55	U			

