PDA Hardware Specification

CONFIDENTIAL

Publication date: January 1999

Sony Computer Entertainment America
919 E. Hillsdale Blvd., 2nd floor
Foster City, CA 94404

Sony Computer Entertainment Europe
Waverley House
7-12 Noel Street
London W1V 4HH, England

Table of Contents

About This Manual

This manual is the 2.4 release of the PDA Hardware Specification documentation.

The product specifications contained in this document may be changed without prior notice.

Changes Since Last Release

- In Appendix A, additional notes have been added to the “Function” section of IFStaticControl.

Related Documentation

This manual should be read in conjunction with the PDA Kernel Specification document.

Developer Reference Series

This manual is part of the *Developer Reference Series*, a series of technical reference volumes covering all aspects of PlayStation development. The complete series is listed below:

Manual	Description
PlayStation Hardware	Describes the PlayStation hardware architecture and overviews its subsystems.
PlayStation Operating System	Describes the PlayStation operating system and related programming fundamentals.
Run-Time Library Overview	Describes the structure and purpose of the run-time libraries provided for PlayStation software development.
Run-Time Library Reference	Defines all available PlayStation run-time library functions, macros and structures.
Inline Programming Reference	Describes in-line programming using DMPSX, GTE inline macro and GTE register information.
SDevTC Development Environment	Describes the SDevTC (formerly "Psy-Q") Development Environment for PlayStation software development.
3D Graphics Tools	Describes how to use the PlayStation 3D Graphics Tools, including the animation and material editors.
Sprite Editor	Describes the Sprite Editor tool for creating sprite data and background picture components.
Sound Artist Tool	Provides installation and operation instructions for the DTL-H800 Sound Artist Board and explains how to use the Sound Artist Tool software.
File Formats	Describes all native PlayStation data formats.
Data Conversion Utilities	Describes all available PlayStation data conversion utilities, including both stand-alone and plug-in programs.
CD Emulator	Provides installation and operation instructions for the CD Emulator subsystem and related software.
CD-ROM Generator	Describes how to use the CD-ROM Generator software to write CD-R discs.

Performance Analyzer User Guide	Provides general instructions for using the Performance Analyzer software.
Performance Analyzer Technical Reference	Describes how to measure software performance and interpret the results using the Performance Analyzer.
DTL-H2000 Installation and Operation	Provides installation and operation instructions for the DTL-H2000 Development System.
DTL-H2500/2700 Installation and Operation	Provides installation and operation instructions for the DTL-H2500/H2700 Development Systems.

Typographic Conventions

Certain Typographic Conventions are used through out this manual to clarify the meaning of the text. The following conventions apply to all narrative text except for structure and function descriptions:

Convention	*Meaning*
`courier`	Indicates literal program code.
Bold	Indicates a document, chapter or section title.

The following conventions apply within structure and function descriptions only:

Convention	*Meaning*
Medium Bold	Denotes structure or function types and names.
Italic	Denotes function arguments and structure members.

Developer Support

Sony Computer Entertainment America (SCEA)

SCEA developer support is available to licensees in North America only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In North America	**In North America**
Attn: Developer Tools Coordinator Sony Computer Entertainment America 919 East Hillsdale Blvd., 2nd floor Foster City, CA 94404 Tel: (650) 655-8000	E-mail: DevTech_Support@playstation.sony.com Web: http://www.scea.sony.com/dev Developer Support Hotline: (650) 655-8181 (Call Monday through Friday, 8 a.m. to 5 p.m., PST/PDT)

Sony Computer Entertainment Europe (SCEE)

SCEE developer support is available to licensees in Europe only. You may obtain developer support or additional copies of this documentation by contacting the following addresses:

Order Information	Developer Support
In Europe	**In Europe**
Attn: Production Coordinator Sony Computer Entertainment Europe Waverley House 7-12 Noel Street London W1V 4HH Tel: +44 (0) 171 447 1600	E-mail: dev_support@playstation.co.uk Web: https://www-s.playstation.co.uk Developer Support Hotline: +44 (0) 171 447 1680 (Call Monday through Friday, 9 a.m. to 6 p.m., GMT or BST/BDT)

Notational conventions

Names of registers are enclosed within curly braces { }.

(Example) Register Abc -> {Abc}

Names of bit fields are enclosed within angle brackets < >.

(Example) Bit field Xyz -> <Xyz>

A specific bit field in a specific register is expressed as the register name and bit field connected by a period.

(Example) Bit field Xyz in register Abc -> {Abc}.<Xyz>

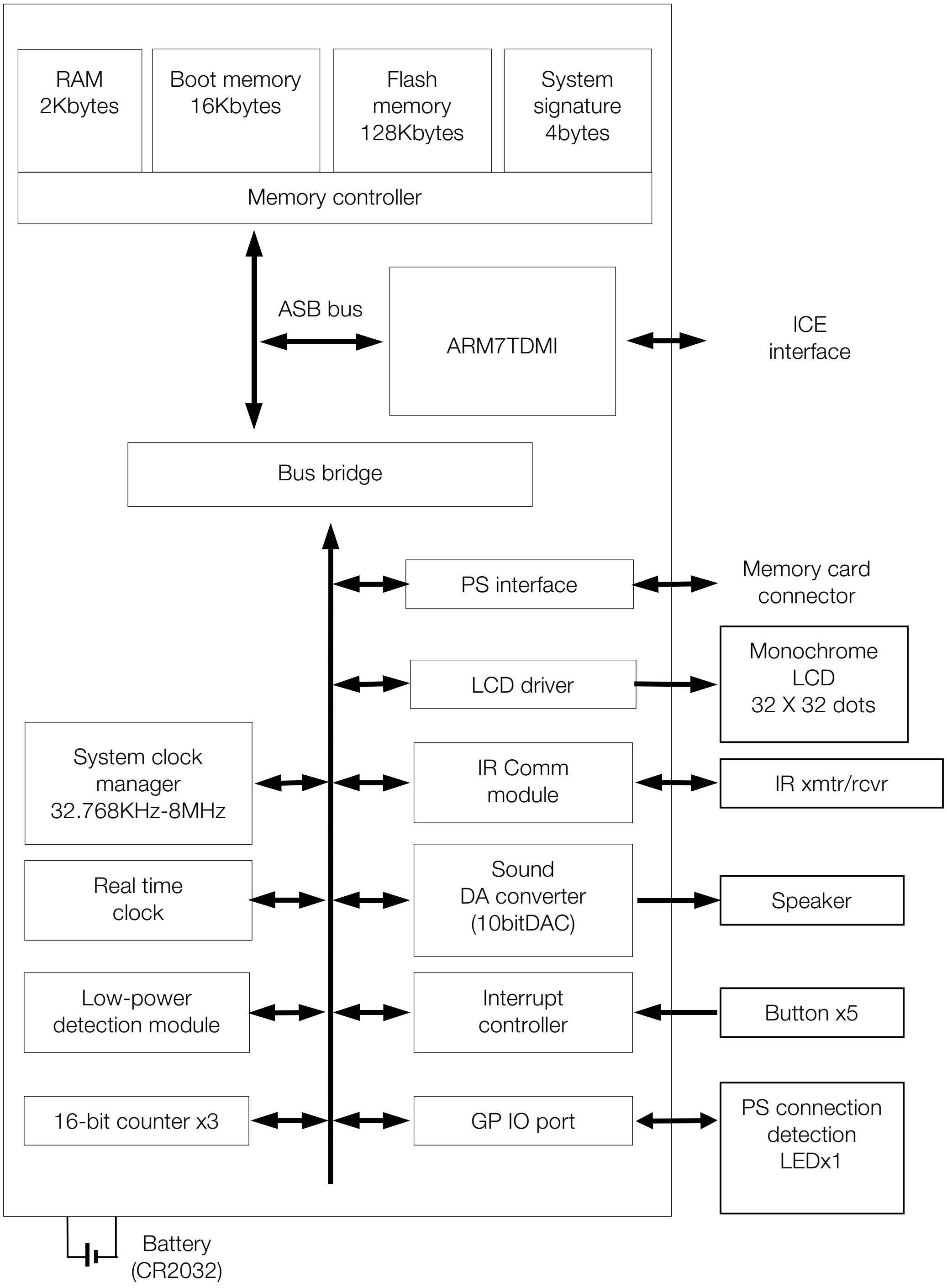
Register numbers are enclosed within square brackets [ ].

(Example) Abc[0..2] -> Abc0, Abc1, Abc2

System architecture

A block diagram of the system is shown below.

Figure 1: System block diagram



Device registers

All devices are managed as memory-mapped "device registers". For more information on the individual device registers, please refer to the following chapters and to Appendix A.

Access

Each device register is 32-bits wide and has an address that is aligned on a 32-bit boundary. Device registers must be accessed by word (32 bits) and cannot be accessed by byte (8 bits) or halfword (16 bits) except for {FLASHWriteSgn0} and {FLASHWriteSgn1} which are accessed on a halfword (16-bit) boundary.

The memory controller device registers ({REGRemap}, {FLASHREMAPStatus}, {FLASHACTIVEBlocks}, {FLASHVIRTUALAddrX[X=0..15]}) can be accessed in 1 bus clock cycle. All other device register accesses require 2 bus clock cycles.

Bit fields

When multiple devices are associated with a single device register, the device register is divided up into an appropriate number of bit fields.

Undefined bit fields are read as zeros.

Addresses and bit fields

Appendix B describes the memory addresses and structure of bit fields for the individual device registers.

Reset sequence

Reset is performed by turning the power off, then on, or by pressing the reset switch. The reset sequence proceeds according to the following steps:

1) If there is an instruction being executed, it is cancelled.
2) All devices enter stand-by mode.
3) All interrupt requests from interrupt sources are masked.
4) The system clock is set to 32.768 KHz.
5) The memory controller is set to "immediately after reset" mode.
6) The program counter is cleared and instruction execution begins.

Processor core

The system uses the ARM7TDMI processor core. Please refer to the "ARM Architecture Reference Manual" for specifications and for a description of the ARM7TDMI instruction set.

Endianness

The ARM7TDMI processor core operates only in little-endian mode. In little-endian mode, the byte with the lowest address in a word (32 bits) contains the lowest-order byte, and word data is expressed as the address of the lowest-order byte.

Embedded ICE

The ARM7TDMI processor core contains the "ARM Standard Embedded In Circuit Emulation" feature. The on-chip ICE module is connected to the host system through a JTAG port.

Memory

The system is equipped with four types of memory: static RAM, boot memory, flash memory, and system signature.

Static RAM

Static RAM is implemented as a 32-bit device with a total capacity of 2 KBytes. Read accesses (including fetching of instructions) and write accesses of 8, 16, and 32 bits can be performed without the need to insert wait cycles.

Boot memory

Boot memory is implemented as a 16-bit device with a total capacity of 16 Kbytes. 16-bit read accesses (including fetching of instructions) can be performed with no wait cycles if the system clock speed is 4 MHz or less. If the system clock speed is 8 MHz, one wait cycle is required. Wait cycle management is performed in software. 32-bit accesses require twice as many cycles as 16-bit accesses.

Flash memory

Flash memory is a special kind of read-only memory that can be programmed and erased through software. Flash memory is a 16-bit device like boot memory.

The total capacity of flash memory is 128 KBytes. Flash memory is organized as a collection of 128 byte regions known as sectors. Erasing and writing of data is performed in sectors. The memory can also be mapped by the memory controller into 8 KByte blocks in virtual address space.

Read accesses

Read accesses (including fetching of instructions) are performed in a manner similar to that of boot memory.

Write accesses

Flash memory is comprised of 1024 128-byte sectors. Each sector contains a continuous 128 byte memory space with a starting address aligned on a 128-byte boundary. A description of the data write sequence is given below. Data reads/instruction fetches from the flash memory device must not be performed during the data write sequence.

1) Set {FLASHDataController}.<LOADPAGE> and {FLASHDataController}.<ENPRG> to 1.
2) Write halfword data 0xFFAA to address 0x080055AA.
3) Write halfword data 0xFF55 to address 0x08002A54.
4) Write halfword data 0xFFA0 to address 0x080055AA.
5) Write up to 64 halfwords to the target sector.
   The actual writes to the device take place approximately 100 microseconds after the final data write. The completion of the write operation can be determined by checking {FLASHDataController}.<BUSY>. (The write operation takes approximately 20 msec.)
6) Set {FLASHDataController}.<LOADPAGE> and {FLASHDataController}.<ENPRG> to 0.

System signature

The system signature is implemented as 4 bytes of read-only memory and is used to store an ID that is unique to each unit. The system signature can be programmed via software.

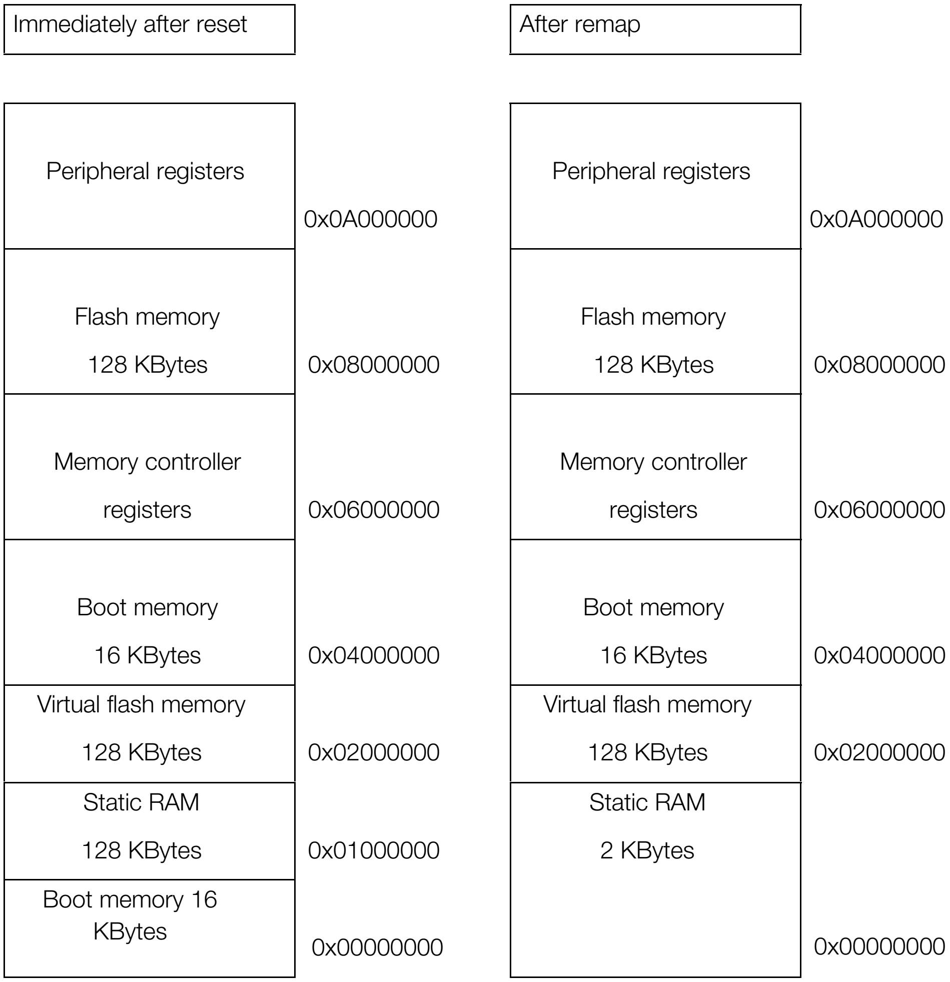
Access to this memory is carried out via a system call.

Memory map

The memory map has 2 modes. These are known as "immediately after reset" and "after remap". Immediately after a reset, the memory map goes into "immediately after reset" mode. The memory map goes into "after remap" mode when a 1 is written to {REGRemap}.<GENREM>. Mode transition in the opposite direction is not supported.

When the memory map is in the "after remap" mode, the interrupt vectors are in static RAM, thus allowing the user to use vector hooks, re-definitions, and dynamic rewrites.

Figure 2: Memory Map



Virtual flash memory

The 128 KBytes of flash memory are divided into 8 KByte contiguous memory regions. Each region is referred to as a "block" and is aligned on an 8 KByte boundary. Blocks are identified by "block numbers" assigned starting with the lower addresses. Mapping to virtual flash memory regions can be performed by specifying a base address (in 8 KByte units) using {FLASHVIRTUALAddrX[X=0..15]}. The block to be mapped to the virtual flash memory region can be selected using {FLASHACTIVEBlocks}.

Regardless of whether virtual flash memory regions are mapped, the flash memory is always mapped to physical addresses.

Mapping procedure

Mapping is performed using the following procedure:

1) When a block is to be mapped, {FLASHVIRTUALAddrX[X=0..15]}. {VIRTUALADDR} is set to the base address of the block.
2) The block number of the block to be mapped to the virtual flash memory region is set in {FLASHACTIVEBlocks}.
3) {REGRemap}.<FLASHVIR> is set to 1.
4) Confirm that {REGRemap}.<FLASHVIR> is set to 1.

If an improper setting was made in (1) above, mapping to the virtual flash memory region is not performed and {REGRemap}.<FLASHVIR> will be set to 0. If this happens, the incorrect setting(s) must be analyzed from the {FLASHREMAPStatus} bits and the operations above must be repeated.

Mapping to the virtual flash memory region can be disabled by setting {REGRemap}.<FLASHVIR> to 0.

If a normal memory access is attempted to an unmapped virtual flash memory region, a "processor exception (data fetch based on access type)" will be generated.

Map example

The following is an example of a sample mapping operation:

Objective:

Blocks 13 and 8 are to be placed continuously, and in order, at the lowermost positions of the virtual flash memory space.

Procedure:

1) Set {FLASHVIRTUALAddr13} to 0x0.
2) Set {FLASHVIRTUALAddr8} to 0x1.
3) Set {FLASHACTIVEBlocks} to 0x2100.
4) Set {REGRemap} to 0x2.
5) Check to see if {REGRemap}.<FLASHVIR> is 1.

Power supply

System power can be obtained from either the battery or the PS interface. Switching between these two power sources is performed automatically, and is triggered by a power supply voltage detector that is part of the PS interface. The power output from the PS interface is used as a double-edged interrupt source.

Power supply switching

When the PS interface is not the active power source and an internal battery is present, power will be taken from the battery. If the battery is providing power, and the PS interface also begins supplying power, the power source will automatically switch from the battery to the PS interface. Once the PS interface becomes the active power source, it will remain the active power source regardless of the state of the battery. However, if the PS interface itself stops providing power while it is the active power source, the power source will automatically switch to the battery.

If there is no internal battery present and the PS interface begins providing power, the PS interface will become the active power source and the system will start up.

Detecting the active power source

The power supply voltage from the PS interface is used as a double-edged interrupt source (source number 11). This signal is available through read-only general-purpose IO port PIO4 and can be used to determine the active power source or a power source switch, either by polling the interrupt or the device register of the interrupt controller.

Power supply switch

The system does not have a power supply switch.

Low-voltage management

When the power supply voltage starts to drop, writes to flash memory are inhibited. If the voltage continues to drop, the processor core may enter an unpredictable state and will not return to a deterministic state until it is reset, even if the power supply voltage is restored. If the processor is in an unpredictable state and the PS interface begins providing power, the flash memory may be overwritten by the processor.

Interrupt controller

The ARM7TDMI processor core supports two interrupt vectors: FIQ (address 0x1C) and IRQ (address 0x18). In this system, FIQ is connected to the Synchronous Serial Communications module and to counter channel 2. IRQ is connected to all other interrupt sources (total of 12 types). Two independent interrupt controllers are provided for the interrupt vectors.

Interrupt sources

Interrupt sources are assigned consecutive "source numbers" unrelated to the vector to which they are connected.

Interrupt sources are divided into three types based on the interrupt conditions which generate the request. The interrupt types are "positive edge", "double-edged", and "level". These are fixed for each source and cannot be changed.

Positive-edge sources make one interrupt request each time the interrupt request condition is satisfied. Double-edged sources make an initial request when the interrupt request condition is satisfied and a subsequent request when the interrupt request condition is no longer valid. Level sources continuously generate interrupt requests as long as the interrupt request condition is satisfied.

The device registers of the interrupt controllers operate independently for each interrupt. The interrupt source to be handled is specified by setting the appropriate bit(s) which correspond to the source.

A description of the interrupt sources is given below:

Table 1: List of Interrupt Sources

Source number	Symbol	Vector	Type	Source name
0	EXT0IRQ	IRQ	Level	Enter button
1	EXT1IRQ	IRQ	Level	Right button
2	EXT2IRQ	IRQ	Level	Left button
3	EXT3IRQ	IRQ	Level	Down button
4	EXT4IRQ	IRQ	Level	Up button
5	EXT5IRQ	IRQ	Level	Unused
6	SPIIRQ	FIQ	Positive edge	Synchronous Serial Communications module
7	TC0IRQ	IRQ	Positive edge	Counter 0
8	TC1IRQ	IRQ	Positive edge	Counter 1
9	RTCIRQ	IRQ	Positive edge	Real time clock
10	VOLIRQ	IRQ	Positive edge	Low-power detection module
11	BATIRQ	IRQ	Double-edged	PS interface power supply line
12	IFIRQ	IRQ	Double-edged	Infrared Communications module
13	TC2FIQ	FIQ	Positive edge	Counter 2

Interrupt priority

In the following discussion, the phrase "during interrupt processing" refers to the interval beginning with the jump to the interrupt vector and ending with the return to the original point prior to the start of interrupt processing.

Vectors

During IRQ interrupt processing, IRQ interrupts are automatically masked. During FIQ interrupt processing, both IRQ interrupts and FIQ interrupts are automatically masked.

Interrupt sources

For interrupts that are connected to a single interrupt vector, there is no prioritization of interrupt sources.

Mechanism

Each interrupt source is associated with two bit-registers: a "latch register" and a "mask register".

The latch register is set by an interrupt request from the source and is cleared by writing a 1 to the bit corresponding to the interrupt source in {INTSourceClear}. The value of the latch register can be obtained by reading {INTStatus}.

The mask register is set by writing a 1 to the bit corresponding to the interrupt source in {INTEnableSet} and is cleared by writing a 1 to the corresponding bit in {INTEnableReset}. The value of the mask register can be obtained by reading {INTEnableSet}.

The contents of the latch register and mask register are ANDed together for each interrupt source and the results are ORed together for each interrupt vector to generate interrupt request signals for the processor core.

The status of the interrupt request before it is latched is known as the "interrupt request status." It can be obtained by reading INTRawStatus.

Host interface

The interface with the host system consists of the Synchronous Serial Communications module, which handles the system as a slave station, and the host system's power detection feature (see the "Power Supply" section for more information).

System Clock Control module

The system clock frequency can be dynamically changed from within a program in order to reduce power consumption. A stand-by mode is also available to suspend all hardware activity except for the liquid crystal display and the RTC.

System clock frequency

The frequency of the system clock can be set to one of nine possible values. The frequency is set by writing a bit pattern to {PMFrequency}.<FREQ> that represents the desired value. The table below shows clock values, corresponding bit patterns, actual clock frequency, and power consumption.

Table 2: System clock frequency

Clock Value	Bit Pattern	Actual Frequency	Power Consumption *
32.768 KHz	0000	32.768 KHz	0.363 mA
62.5 KHz	0001	63.488 KHz	0.772 mA
125 KHz	0010	126.976 KHz	0.998 mA
250 KHz	0011	253.952 KHz	1.26 mA
500 KHz	0100	507.904 KHz	1.6 mA
1 MHz	0101	1.015808 MHz	2 mA
2 MHz	0110	1.998848 MHz	3.1 mA
4 MHz	0111	3.9977MHz	5.4 mA
8 MHz	1xxx	7.9977 MHz	10.8 mA

* With power supply voltage at 3.0 V, LED off, Infrared Communications module and sound DA converter in stand-by mode.

Since the system operation can become unstable if the system clock frequency is switched while in ARM mode, the system clock frequency must be switched in ARM Thumb mode.

Latencies when the system clock is updated

When the system clock is running at 32 KHz and the frequency is changed, the latency between writing to {PMFrequency}.<FREQ> and the update of the clock is 32 cycles (approximately 1 msec). A few additional cycles are required when the system clock is initially at a value other than 32 KHz. When the system clock is updated, it is necessary to confirm that {PMFrequency}.<LOCK> is set to 1, then the following operations must be performed.

Reset

The system clock will be set to 32.768 KHz immediately after a reset.

Stand-by mode

When {PMStandby}.<STDBY> is set to 1, the system will enter stand-by mode. In stand-by mode, all hardware activity with the exception of the liquid crystal display and the RTC is suspended. In stand-by mode, the contents of static RAM and registers are saved. The LCD has a separate feature to reduce power consumption. The RTC cannot be stopped.

The system transitions from stand-by mode to active mode when the Enter button is pressed (source number 0), or as a result of an interrupt request from the RTC (source number 9), low power detection (source number 0), and the PS interface power supply (source number 11).This process is known as "wake-up". Immediately after wake-up, the system clock momentarily runs at 32.768 KHz then changes to the clock value that was active before stand-by mode was entered.

However, when switching to stand-by mode when IntRawStatus is high, an interrupt will not be generated when wake-up has been performed with the RTC and low power detection factors. Therefore, when detecting wake-up using these two factors, both IntRawStatus types must be controlled immediately after switching to stand-by mode processing.

Low-voltage Detection module

The system is equipped with a low-voltage detection module for detecting drops in power supply voltage.

Mechanism

The Low-voltage Detection module determines when the power supply voltage drops close to the minimum voltage required to write to the flash memory. A positive-edge interrupt request (source number 10) is generated approximately 1 msec after the low voltage condition is detected.

Power conservation features

The Low-voltage Detection module provides a mechanism to suspend low-voltage detection and a periodic test mode to reduce power consumption.

Suspend function

The "generate reference voltage" function, which serves as the reference point for low-voltage detection, can be suspended by setting {LVDControl}.<BGSTBY> to 1. "Voltage comparison", which performs the actual detection operation, can be suspended by setting {LVDControl}.<COMPSTBY> to 1. Note that suspending "reference voltage" generation prevents the LCD from operating.

Periodic Test mode

If {LVDControl}.<AUTCNT> is set to 1, the Low-voltage Detection module will use the RTC to perform low-voltage detection once per second. In other words, "voltage comparison" is performed periodically when

Periodic Test mode is enabled. In Periodic Test mode, bit fields other than {LVDControl}.<AUTCNT> are ignored.

Reset

Immediately after a reset, both "generate reference voltage" and "voltage comparison" operations are suspended.

Liquid crystal display features

The system provides a monochrome, two-level liquid crystal display with 32 dots each in the horizontal and vertical directions.

Video memory (VRAM)

The video memory (hereinafter referred to as VRAM) is mapped to memory space as 32 device registers {BackPlanen[n=0..31]} (read/write, 32-bit access only).

The 32 pixels in the top row of the screen correspond to {BackPlane0} and the 32 pixels in the bottom row of the screen correspond to {BackPlane31}. The leftmost pixels on the screen correspond to the lowest-order bits in {BackPlanen[n=0..31]}, and the rightmost pixels on the screen correspond to the highest-order bits in {BackPlanen[n=0..31]}.

VRAM access can be performed regardless of display state. No synchronization or arbitration is provided between VRAM reads performed by hardware and VRAM writes performed by software.

Frame rate

The frequency with which VRAM data is read by the hardware for display purposes is referred to as the "frame rate". A "frame" is the smallest unit of operation during a read. For one frame, signals corresponding to the VRAM data are sent to the liquid crystal device twice. Each transfer is referred to as a "field". The timing and state of the frame and field transfers are not available.

Three different frame rates--16, 32, 64 Hz--can be set using {LCDControl}.<FR>.

Inverted display

When {LCDControl}.<ROT> is set to 0, the display will appear correctly oriented with the PS interface connector positioned down and the Infrared Communications module transmitter/receiver positioned up. When {LCDControl}.<ROT> is set to 1, the liquid crystal display will be rotated 180 degrees.

When {LCDControl}.<ROT> is 0, the display is referred to as being in "normal display mode". When {LCDControl}.<ROT> is 1, the display is in "inverted display" mode.

Power conservation features

The liquid crystal display will stay in active mode even if the entire system is put in stand-by mode. To reduce power consumed by the liquid crystal display, independent features allow the display to be inhibited, the power to the liquid crystal display device to be suspended, and the generation of the reference voltage to the liquid crystal display device to be stopped. Also, the area of the display region can be reduced by one-half or one-quarter. These features are described below.

Suspending power

Power to the liquid crystal display device can be suspended by setting {LDCControl}.<CPEN> to 0.

Stopping the generation of the reference voltage

Power to the liquid crystal display device can be suspended by setting {LVDControl}.<BGSTBY> to 1.

Reducing the display area

The display screen is divided into four sub-regions along the vertical axis. To reduce power consumption, the display can be restricted to a single sub-region or two adjacent sub-regions. Each sub-region is referred to as a "segment." Each segment is assigned a "segment number" from 1 to 4, starting with the top of the screen.

Full-screen, 1/2-screen, and 1/4-screen display can be selected by setting the corresponding bit pattern in {LDCControl}.<DISMOD> as shown below.

Table 3: Liquid crystal display regions

Bit pattern	Display segment
000	1+2+3+4
001	1
010	2
011	3
100	4
101	1+2
110	2+3
111	3+4

Reset

The display and power are turned off immediately after a reset. The generation of the "reference voltage", which is also used by the Low-voltage module, is suspended.

A reset or the operation of turning the display on/off will not put the VRAM into a deterministic state.

Turning the display on and off

The procedure for turning the display on is as follows:

1) Set {LCDControl}.<DISON> to 0. Enter the bit pattern which specifies the desired frame rate into {LCDControl}.<FR> and set {LCDControl}.<CPEN> to 1.
2) Set {LCDControl}.<DISON> to 1 and enter appropriate values into the other bit fields.

The procedure for turning the display off is as follows:

1) Set {LCDControl}.<DISON> to 0.
2) If a frame is being displayed, the display will be turned off after completion of the operation.

Sound

The system provides PCM-based sound features.

Mechanism

{DACData}.<DACV> maps the digital input of the DA converter into an analog signal for the speaker. The value contained in this bit field is interpreted as a 10-bit two's complement signed integer and is used to control the current flowing to the speaker. If the integer value of {DACData}.<DACV> is 0, no current will flow to the speaker. If the value is 0x1FF/0x200, the maximum current will flow to the speaker. If {DACData}.<DACV> is negative, current will flow in the opposite direction.

Power conservation features

Independent stand-by and active modes are provided for the DA converter and the speaker in order to reduce power consumption. The contents of {DACData} are saved during stand-by mode. DA converter mode can be selected through {DACControl}.<STDBY>. Speaker mode can be selected through general-purpose IO port 5.

Stand-by mode can also be used to implement muting.

Reset

The DA converter will enter stand-by mode immediately after a reset. {DACData}.<DACV> will be set to 0.

Button switches

The 5 button switches act as independent level-based interrupt sources.

Table 4: Interrupt source assignments for buttons

Interrupt source number	Button name
0	Enter
1	Right-arrow
2	Left-arrow
3	Down-arrow
4	Up-arrow

The Infrared Communications module

The system is equipped with a bit-level infrared receiver/transmitter which provides half-duplex communication.

Subcarrier modulation

Subcarrier modulation makes communication less susceptible to noise and increases the possible communication distance. Subcarrier modulation can be enabled from within a program. If the system clock is 1 MHz or greater, the subcarrier frequency will be 40 KHz. 40 KHz subcarrier modulation is generally used in infrared "remote control" devices. Subcarrier modulation will improve noise tolerance but with the trade-off that the data transfer rate will be reduced. This should be contrasted with the IrDA protocol, which does not use subcarrier modulation.

To receive a subcarrier-modulated infrared signal, enable the subcarrier removal filter from general-purpose IO port 6.

Receiver

The receiver acts as a double-edged interrupt source. Interrupt requests can be triggered by both dark-to-light and light-to-dark state transitions. Receiver polling can be performed by accessing the device register of the interrupt controller.

The Infrared Communications module receiver is equipped with a filter for eliminating noise. To enable this filter, set {IFStaticControl}.<BFLT> to 0. If the filter is enabled, signals which are at or above 4 cycles of the filter frequency will be received. The filter frequency is fixed at 40 KHz when the system clock is 1 MHz or higher. If the system clock is less than 1 MHz, the filter frequency is 1/25th that of the system clock.

Signals cannot be received during transmission.

Transmitter

The transmitter is enabled when {IFDynamicControl}.<T_EN> is set to 1.

If {IFStaticControl}.<BGEN> is set to 0, transmission is performed with subcarrier modulation. When the system clock is 1 MHz or higher, subcarrier modulation is fixed at 40 KHz. If the system clock is less than 1 MHz, the subcarrier frequency is 1/25 that of the system clock.

Table 5: System clock and subcarrier frequency

System clock (nominal)	System clock (actual)	Subcarrier frequency (actual)
1 MHz	1.015808 MHz	40.6 KHz
2 MHz	1.998848 MHz	39.97 KHz
4 MHz	3.997700 MHz	39.97 KHz
8 MHz	7.995392 MHz	39.97 KHz

Signals cannot be received during transmission.

Transmit signal strength can be selected as subcarrier modulation mode (low output) or IrDA mode (high output) using general-purpose IO port 6.

Power conservation features

The Infrared Communications module provides independent stand-by and active modes to reduce power consumption. The Infrared Communications module is set to stand-by mode when {IFStaticControl}.<STDBY> is set to 1. When this bit is set to 0, active mode is enabled.

Reset

Immediately after a reset, the Infrared Communications module will enter stand-by mode.

Real time clock

The real time clock (hereinafter referred to as the RTC) is a device that counts real time in one-second units and is independent from the system clock.

Counters

Two sets of 32-bit BCD registers are provided for year/month/date and day/hours/minutes/seconds. Hours are expressed in 24-hour time.

It is recommended that applications wait for {INTStatus}.<RTCIRQ> to become 1 before reading counter values.

If a counter read collides with a counter update, the accuracy of the read cannot be guaranteed. Therefore, it is recommended that reads be performed repeatedly until two consecutive reads provide the same result.

Counter settings

To set the RTC, first set {RTCControl}.<PRGSEL> to 1 which will place the RTC in programming mode. In this mode, at 4KHz intervals {INTStatus}.<RTCIRQ> become 1 and an interrupt is generated. Setting {RTCInc}.<INCCNT> to 1 twice causes the counter specified by {RTCControl}.<CNTLSEL> to be incremented by 1. (Incrementation is performed at 2KHz cycles.)

Counters

The system is equipped with 3 channels of 16-bit counters. Each counter is an interrupt source and can automatically restore its initial setting.

Mechanism

Each counter counts down as a function of whether the system clock is pre-scaled by 2, 32, or 512. The current values of the counters are stored in device register{TIMERnValue[n=0..2]}. This register is read-enabled.

Each counter is linked to an interrupt controller as an independent interrupt source. An interrupt request is generated when the value of a counter becomes 0.

Pre-scaling values

Pre-scaling can be selected by entering the following values into {TIMERControl}.<CLKDIV>.

Table 6: Counter pre-scaling values

Bit pattern	Pre-scaling value
00	2
01	32
10	512
11	(undefined)

Initial values

The initial values of the counters are set by writing to device registers {TIMERnLoad[n=0..2]}. These values are saved until a reset is performed or until the settings are changed. Writes to {TIMERnLoad[n=0..2]} are enabled only when the counters are stopped.

The counters begin counting down when {TIMERnControl[n=0..2]}.<ENTIMER> is set to 1. When a counter value reaches 0, an interrupt request is generated. Then the initial values from {TIMERnLoad[n=0..2]} are copied to {TIMERnValue[n=0..2]} and the counter resumes its countdown.

General-purpose IO ports

The system provides 7 channels of general-purpose IO ports (PIO0 - PIO6).

Bi-directional ports

The four general-purpose IO ports from PIO0 to PIO3 can be set to either input or output by setting {PIOControl}. Input and output cannot be performed simultaneously.

Access

High Level output to a general-purpose IO port can be performed by setting the corresponding bit in {PIOSetOutputData} to 1.

Low Level output to a general-purpose IO port can be performed by setting the corresponding bit in {PIOClearOutputData} to 1.

Input from a general-purpose IO port can be performed by reading a corresponding bit from {PIOReadInputData}.

Table 7: Devices connected to general-purpose IO ports

Corresponding bit number	Port name	Property	Connected device
0	PIO0	Bi-directional	None
1	PIO0	Bi-directional	LED
2	PIO2	Bi-directional	None
3	PIO3	Bi-directional	None
4	PIO4	Input	PS interface (power-supply line)
5	PIO5	Output	Speaker (stand-by/active switch)
6	PIO6	Output	Infrared Communications module (sub-carrier optical filter switch)

LED

An LED is connected to bi-directional port PIO1. When PIO0 is in output mode, it is used as the on switch for the LED.

PIO4

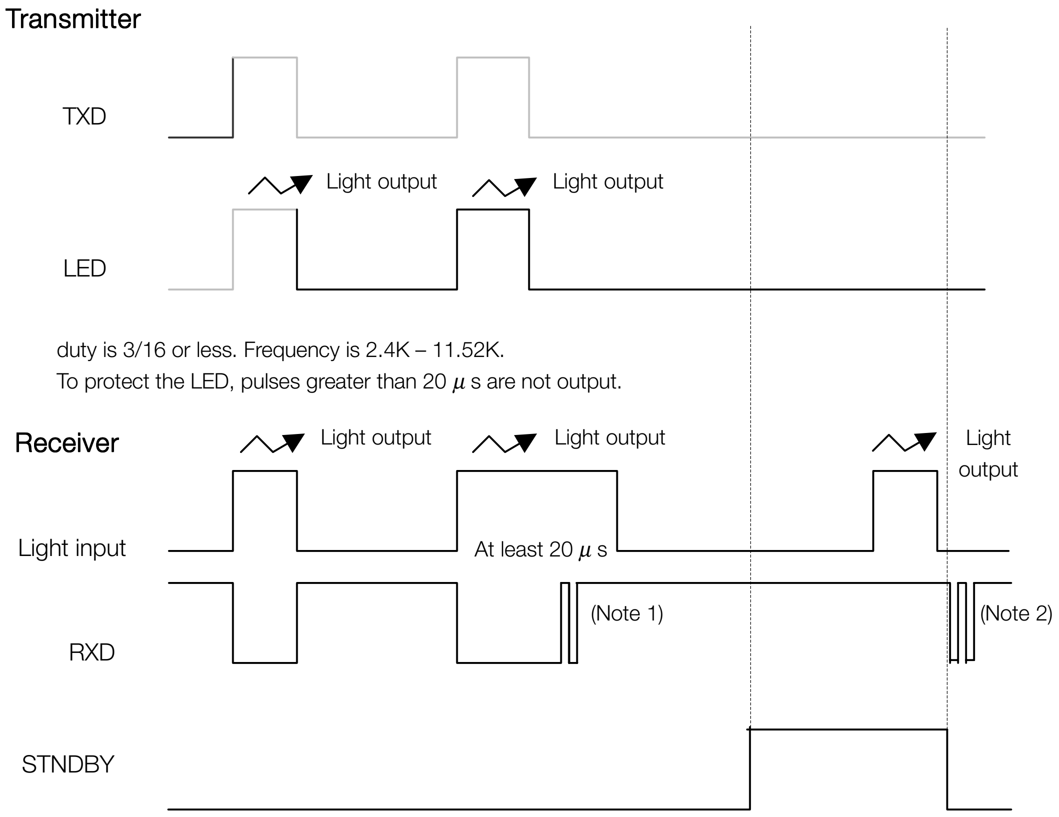
The read-only general-purpose IO port PIO4 is connected to the power supply line in the PS interface and acts as a level-type interrupt source (source number 11).

Hardware supplement

Infrared Communications module

The Infrared Communications module has an IrDA mode and a remote control mode. The following is an overview of the transmit/receive waveforms used in the two modes.

IrDA mode



duty is 3/16 or less. Frequency is 2.4K – 11.52K.
To protect the LED, pulses greater than 20 μ s are not output.

(Note 1)

The oscillation shown above is generated when a light input pulse of 20 μ s or longer is received with an emission strength of minimum signal sensitivity.

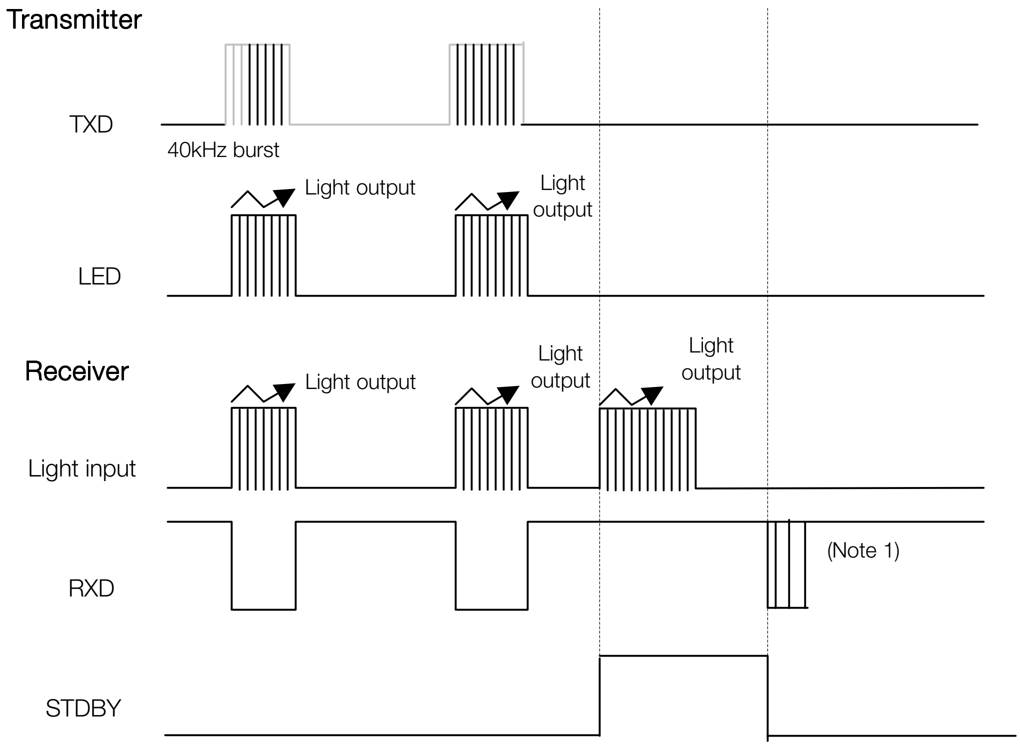
(Note 2)

Normally occurs within 500 μ s after returning from standby.
During this time, noise is generated and should be ignored.

(Note 3)

If light is received spontaneously within 100 μ s after light generation, improper operation will result.
During this time, RXD should be ignored.

Remote-control mode



(Note 1)

Normally occurs within 20 ms after returning from standby
During this time, noise is generated and the signal should be ignored.

(Note 2)

Because the remote control receiver section is unusually sensitive in sunlight, as well as in an environment of 1000Lx or more, such as when the module's upper surface is under fluorescent light, a maximum of 300 μ s of random noise will be generated. For the sake of software, it is best to avoid this situation.

(Note 3)

Improper operation occurs with an inverter fluorescent light source having a frequency of around 40 KHz, shortening the attainable distance.

(Note 4)

Under an environment that exceeds 6000Lx in outdoors, etc. the attainable distance is shortened.

Other properties

Table 8: Infrared Communications module properties

	Transmitter	Receiver	PDA-to-PDA attainable distance
IrDA mode	Transmission wavelength: 850 nm	40 KHz carrier demodulation	20 cm (initial battery)
Remote-control mode	Transmission wavelength 950 nm	Baseband	5 m average (initial battery, fluorescent lamp 6001x)

Liquid crystal display

Hardware properties

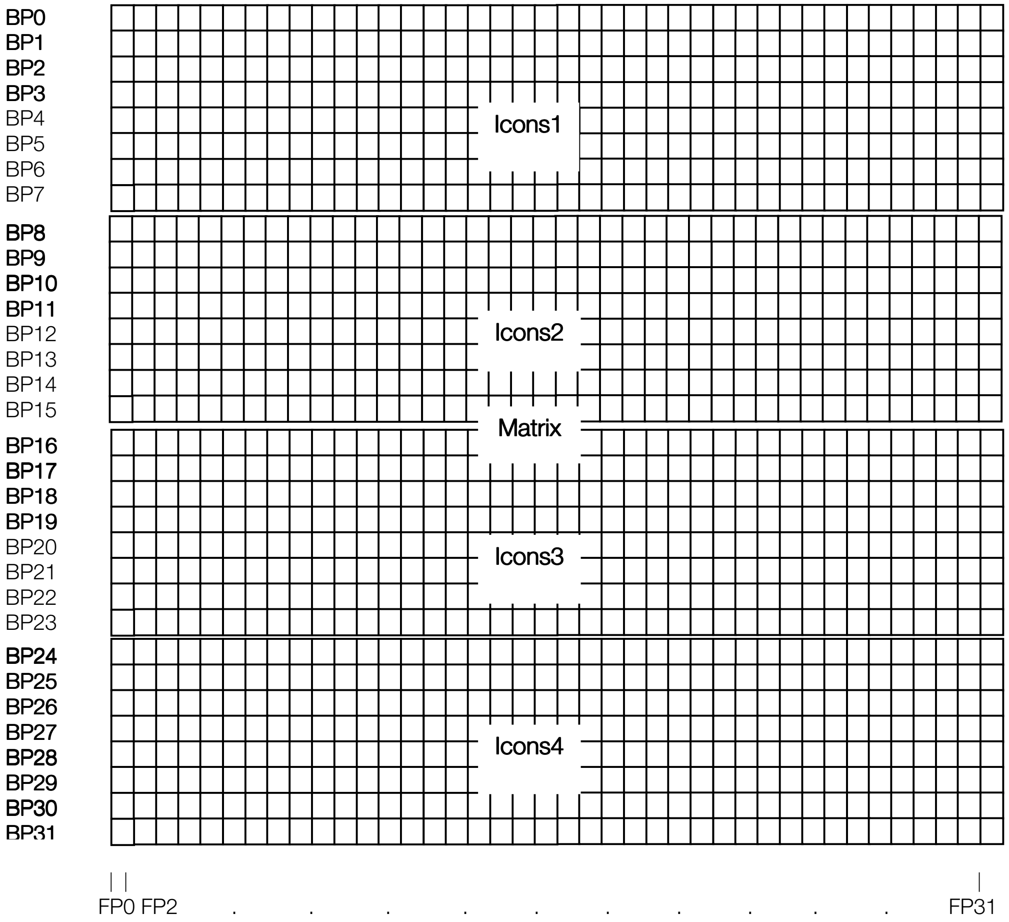
Table 9: LCD specifications

Format	STN
Total speed	190 ms
Viewing angle	+/- 30 degrees
Operating temperature range	-5 degrees C to 45 degrees C

* 7.2 volts, 0 degrees = viewpoint at 90 degrees in front of liquid crystal

Video memory

Figure 3: LCD video memory



Current consumption

CPU current consumption

Table 10: CPU current consumption

Activity Mode		Current Consumption(mA)	Activity time (Hour)	Activity time (Day)
Without liquid crystal display	stand-by mode	0.05	1700	70.8
	32KHz	0.36	400	16.7
	62.5KHz	0.72	220	9.2
	125KHz	1.00	140	5.8
	250KHz	1.26	100	4.2
	500KHz	1.60	70	2.9
	1MHz	2.00	52	2.2
	2MHz	3.10	20	0.8
	4MHz	5.40	4	0.2
With liquid crystal display	stand-by mode	0.15	1000	41.7
	32KHz	0.46	330	13.8
	62.5KHz	0.82	180	7.5
	125KHz	1.10	120	5.0
	250KHz	1.36	90	3.8
	500KHz	1.70	65	2.7
	1MHz	2.10	45	1.9
	2MHz	3.20	19	0.8
	4MHz	5.50	4	0.2

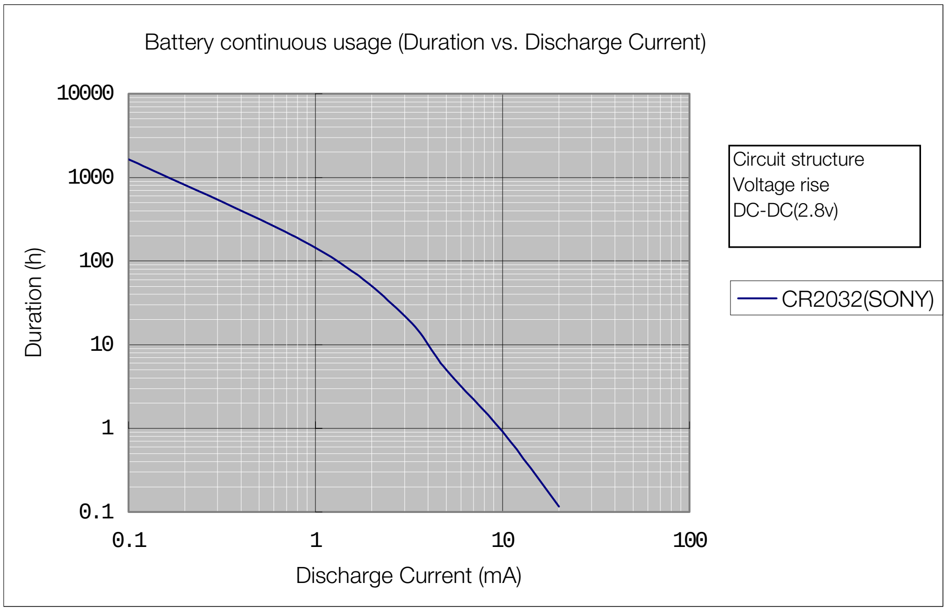
Device current consumption

Table 11: Device current consumption

	TYP	MAX	MAX condition
Audio unit current consumption	10.5 mA	12 mA	Square wave 2.7 Vpp output
LCD current consumption	0.10 mA	0.15 mA	8.5V checkerboard pattern
Transmission LED	4 mA	6 mA	---
Ir remote control transmission	25 mA	40 mA	40 KHz burst
Remote control reception	---	0.5 mA	Idle
IrDA transmission	3 mA	15 mA	3/16 duty
IrDA reception	---	0.5 mA	Idle

Battery life

Figure 4: Battery properties



Note: The graph above is for continuous usage. Intermittent usage will result in longer battery life than what is shown above.

Appendix A: Device Register Details

DACControl

Sound DA Converter Control Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register is used to set the operating mode of the sound DA converter.

Bit definitions:

Bit 0

Name:	STDBY (DAC STanDBY mode)
Meaning:	0: Sound DA converter in stand-by mode.
	1: Sound DA converter active.

DACData

Sound DA Converter Data Register

Access type:

Write

Reset value:

0x00000000

Function:

This register is used to send data to the sound DA converter.

Bit definitions:

Bits 6 .. 15

Name: DACV (DAC Value)

* Two's complement data is used for DACV.

+ maximum value	1FFh
	1FEh
	:
	02h
	01h
Center level	00h
	3FFh
	3FEh
	:
	201h
- maximum value	200h

FLASHACTIVEBlocks

Flash Active Block Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

Each of the bits in this register specifies a physical block that will be mapped to the virtual flash memory region. Writes to this register complete successfully only when mapping is disabled.

Bit definitions:

Bits 0 .. 15

Name: IS0 .. 15 (Defined Active Blocks)

Read values: 0: The block is not mapped to the virtual flash memory region.

1: The block is mapped to the virtual flash memory region.

Write values: 0: Do not map the block to the virtual flash memory region.

1: Map the block to the virtual flash memory region.

FLASHDataController

Flash Control Register

Access type:

Read/Write

Reset value:

0x00000014

Function:

This register contains bits that control flash memory program functions.

Bit definitions:

Bit 0

Name: ENPRG (Enable Flash Programming)

Read values: 0: Programming is disabled.

1: Programming is enabled.

Write values: 0: Disable programming.

1: Enable programming.

Bit 1

Name: LOCK (Lock Bit Protection Mode)

Read values: (reads are not permitted)

Write values: 0: Enable programming.

1: Disable data writes.

Bit 2

Name: BUSY (Flag status of FLASH write)

Function: Flag for detecting completion of flash memory programming.Polling is not necessary.

Read values: 0: Write sequence in progress.

1: No write sequence active.

Write value: 0: (writes not permitted)

Bit 3

Name: STDBY (Flash Standby Mode)

Function: When set, the flash memory device enters stand-by mode, in which the device is temporarily halted to reduce power consumption. A minimum 10 msec start-up time is needed to resume normal operation after leaving stand-by mode.

Read values: 0: Stand-by mode inactive.

1: Stand-by mode active.

Write values: 0: Exit stand-by mode.

1: Enter stand-by mode.

Bit 4

Name: WAIT (WAIT state for Flash access)

Function: Allows 1 wait cycle to be inserted via software for read and write accesses to the flash memory. When flash memory accesses are performed with the system clock operating at 8 MHz, the bit should be set to 1. Otherwise, the bit should be set to 0.

Read values: 0: No wait cycles will be inserted during device accesses.

1: 1 wait cycle will be inserted during device accesses.

Write values: 0: Do not insert wait cycles during device accesses.

1: Insert 1 wait cycle during device accesses.

Bit 5

Name: LOADPAGE (LOAD PAGE)

Read values: 0: Page (sector) writes disabled.

1: Page (sector) writes enabled.

Write values: 0: Disable page (sector) writes.

1: Enable page (sector) writes.

Bit 6

Name: LOADSGN (LOAD system SiGNature)

Read values: 0: System signature writes disabled.

1: System signature writes enabled.

Write values: 0: Disable system signature writes.

1: Enable system signature writes.

FLASHREMAPStatus

Flash Remap Status Register

Access type:

Read

Reset value:

0x00000000

Function:

When the FLASHVIR bit in the REGRemap register is set to 1, this register contains the result of performing an analysis of the contents of registers relating to virtual flash memory region settings. This register is automatically cleared on a read access.

Bit definitions:

Bit 0

Name: MULTIPLE (MULTIPLE blocks defined with the same virtual area)

Read value: 1: Multiple blocks are assigned to the same virtual base address.

Bit 1

Name: NOACTIVE (NO ACTIVE blocks)

Read value: 1: No virtual base address was specified.

Bit 2

Name: DABORT (Data ABORT (create a data abort exception))

Read value: 1: Data access was attempted to or from an unmapped virtual flash memory region.

Bit 3

Name: FABORT (Fetch ABORT (create a fetch abort exception))

Read value: 1: An instruction fetch was attempted from an unmapped virtual flash memory region.

FLASHVIRTUALAddr[0..15]

Flash Virtual Address Register Group

Access type:

Read/Write

Reset value:

0x00000000

Function:

This group of registers holds the offset values used to calculate the virtual base address for each physical block. These values are calculated as follows:

Virtual base address = 0x02000000 + VIRTUALADDR<<13

Writes to these registers complete successfully only when mapping to the virtual flash memory region is not enabled.

Bit definitions:

Bits 0 .. 3

Name:	VIRTUALADDR (VIRTUAL offset ADDRess)
Value:	Physical block number (0-15) for calculating virtual base address.

IFDynamicControl

Dynamic Infrared Communications Control Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register is used to enable the transmission of the output waveform.

Bit definitions:

Bit 0

Name:	T_EN (Transmit Enabled)
Function:	Enables the transmission of the output waveform. The actual output waveform (TXD) is a function of BGEN, which controls 40 KHz pulse generation.
Meaning:	0: Disable output waveform (force transmit data to 0).
	1: Enable transmission of output waveform.

IFStaticControl

Static Infrared Communications Control Register

Access type:

Read/Write

Reset value:

0x00000002

Function:

This register contains bits that control the Infrared Communications Module.
<To conserve power, always set Bit 0 to 1 when the title terminates.>

Bit definitions:

Bit 0

Name: MODE (Select the IF MODE)

Function: Sets the transmit/receive mode of the Infrared Communications Module.

Meaning: 0: Set receive mode.

1: Set transmit mode.

Bit 1

Name: STDBY (STanDBY)

Function: Sets the operating mode of the Infrared Communications Module.

Meaning: 0: Sets operating mode to active.

1: Sets operating mode to stand-by.

Bit 2

Name: BGEN (Block GENerator enable)

Function: Controls the operation of the 40KHz pulse generator.

Meaning: 0: Enables 40KHz pulse generation.

1: Disables 40KHz pulse generation.

Bit 3

Name: BFLT (Block FiLTer enable)

Function: Controls the operation of the input waveform filter.

Meaning: 0: Enable waveform filter.

1: Disable waveform filter.

INTEnableReset

Disable Interrupt Settings Register

Access type:

Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control interrupt status.

Bit definitions:

Please refer to the list of interrupt sources in Table 1 for the name of each bit.

Write values 0: (invalid)

1: Disable interrupts.

Table 1: List of Interrupt Sources

Source number	Symbol	Vector	Type	Name of source
0	EXT0IRQ	IRQ	Level	Confirmation button
1	EXT1IRQ	IRQ	Level	Right direction button
2	EXT2IRQ	IRQ	Level	Left direction button
3	EXT3IRQ	IRQ	Level	Down direction button
4	EXT4IRQ	IRQ	Level	Up direction button
5	EXT5IRQ	IRQ	Level	Unused
6	SPIIRQ	FIQ	Positive edge	Synchronous Serial Communications Module
7	TC0IRQ	IRQ	Positive edge	Counter 0
8	TC1IRQ	IRQ	Positive edge	Counter 1
9	RTCIRQ	IRQ	Positive edge	Real-time clock
10	VOLIRQ	IRQ	Positive edge	Power-drop Detection Module
11	BATIRQ	IRQ	Both edges	PS interface power supply line
12	IFIRQ	IRQ	Both edges	Infrared Communication Module
13	TC2FIQ	FIQ	Positive edge	Counter 2

INTEnableSet

Interrupt Enable Set Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control interrupt enable states.

Bit definitions:

See the list of interrupt sources in Table 1 for names of bits.

Read values	0: Interrupts disabled.
	1: Interrupts enabled.
Write values	0: (Invalid)
	1: Enable interrupts.

INTRawStatus

Interrupt Raw Status Register

Access type:

Read

Reset value:

0x00000000

Function

This register contains a set of bits for monitoring interrupt signals.

Bit definitions:

See the list of interrupt sources in Table 1 for names of bits.

Read values	0: No interrupts generated.
	1: Interrupts generated.

INTSourceClear

Interrupt Source Clear Register

Access type:

Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control interrupt sources.

Bit definitions:

See the list of interrupt sources in Table 1 for names of bits.

Write values: 0: (Invalid)

1: Clear interrupt source.

INTStatus

Interrupt Status Register

Access type:

Read

Reset value:

0x00000000

Function:

This register contains a set of bits for monitoring interrupt signals.

Bit definitions:

See the list of interrupt sources in Table 1 for names of bits.

Read values:	0: Interrupts disabled or not generated.
	1: Interrupts generated.

LCDControl

LCD control register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control the LCD.

Bit definitions:

Bits 0 .. 2

Name: DISMOD (DISplay MODe)

Function: Turns on the LCD among the four available display regions.

Meaning: 0: Turn on entire screen.
1: Turn on only region 1.
2: Turn on only region 2.
3: Turn on only region 3.
4: Turn on only region 4.
5: Turn on regions 1 and 2.
6: Turn on regions 2 and 3.
7: Turn on regions 3 and 4.

Bit 3

Name: CPEN (LCD Charge Pump Enable(analog circuitry activation))

Meaning: 0: Disable LCD charge pump.
1: Enable LCD charge pump.

Bits 4, 5

Name: FR (LCD Frame Rate Selector)

Function: Set LCD frame rate

Meaning: 0: Reset LCD counter and disable LCD clock.
1: 64Hz frame rate.
2: 32Hz frame rate.
3: 16Hz frame rate.

Bit 6

Name: DISON (LCD Display ON)

Meaning: 0: Disable LCD display.
1: Enable LCD display.

Bit 7

Name:	ROT (LCD 180 degrees ROTation)
Meaning:	0: Normal display mode. 1: Inverted display mode.

LVDControl

Low Voltage Detection Control Register

Access type:

Read/Write

Reset value:

0x00000003

Function:

This register contains bits that control low voltage detection.

Bit definitions:

Bit 0

Name: BGSTBY (Band gap* STandBY mode)

Function: This bit controls the band gap for the voltage comparison operation and the LCD controller.

Meaning: 0: Band gap active.

1: Band gap in stand-by mode.

Bit 1

Name: COMPSTBY (COMParator STandBY mode)

Function: This bit controls the voltage comparison operation.

Meaning: 0: Voltage comparison active.

1: Voltage comparison in stand-by mode.

Bit 2

Name: AUTCNT (AUTomatic CoNTrol)

Function: This bit controls low voltage detection.

Meaning: 0: Periodic test mode inactive.

1: Periodic test mode active (1 second intervals).

*Note: The term “band gap” refers to the operational voltage range.

PIOClearOutputData

General-purpose IO Port Data Clear Register

Access type:

Write

Reset value:

0x00000000

Function:

This register contains bits for clearing the output ports.

Bit definitions:

Bit 1

Name: PIO1 (PIO 1 set data out)

Function: This bit controls the LED.

Write values: 0: (Invalid)

1: LED on.

Bit 5

Name: PIO5 (PIO 5 set data out)

Function: This bit controls speaker stand-by mode.

Write values: 0: (Invalid)

1: Set speaker to active.

Bit 6

Name: PIO6 (PIO 6 set data out)

Function: This bit switches the output level of the Infrared Communications Module.

Write values: 0: (Invalid)

1: Set the Infrared Communications Module to low output mode.

PIOControl

General-purpose IO Port Control Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains bits that set the input/output direction of the general-purpose I/O ports. Four bi-directional ports are available but ports 0, 2, 3 are unused.

<The unused ports will default to output and title developers should not change their settings.>

Bit definitions:

Bit 0, bit 1

Name:	P1DIR (Pio 1 DIRection)
Read values:	0: Port set to input.
	1: Port set to output.
Write values:	0: Set port to input.
	1: Set port to output.

PIOReadInputData

General-purpose IO Port Data Input Register

Access type:

Read

Reset value:

0x00000000

Function:

This register is used to read data from the input ports.

Bit definitions:

Bit 4

Name:	PIO4 (PlayStation interface)
Read values:	0: Not connected to the PlayStation.
	1: Connected to the PlayStation.
	* If the PlayStation is turned off, the value will be 0 even if connected to the PlayStation.

PIOSetOutputData

General-purpose IO Port Data Output Register

Access type:

Write

Reset value:

0x00000000

Function:

This register is used for writing to the output ports.

Bit definitions:

Bit 1

Name:	PIO1 (PIO 1 set data out)
Function:	This bit controls the LED.
Write values:	0: (Invalid)
	1: Turn off LED.

Bit 5

Name:	PIO5 (PIO 5 set data out)
Function:	This bit controls speaker stand-by mode.
Write values:	0: (Invalid)
	1: Set speaker to stand-by mode.

Bit 6

Name:	PIO6 (PIO 6 set data out)
Function:	This bit selects the output level for the Infrared Communications Module.
Write values:	0: (Invalid)
	1: Sets Infrared Communications Module to high-output mode.

PMFrequency

System Clock Frequency Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains bits that control the system clock.

Bit definitions:

Bits 0 .. 3

Name:	FREQ (Clock FREQuency)
Function:	This bit selects the clock frequency.
Meaning:	0: 32.768KHz (default value)
	1: 62.5KHz(63.488KHz)
	2: 125KHz(126.976KHz)
	3: 250KHz(253.952KHz)
	4: 500KHz(507.904KHz)
	5: 1MHz(1.015808MHz)
	6: 2MHz(1.998848MHz)
	7: 4MHz(3.9977MHz)
	8 and higher: 8MHz(7.9954Hz)

Bit 4

Name:	LOCK (PLL LOCK bit status)
Function:	This bit is read only.
Meaning:	0: PLL not locked.
	1: PLL locked.

PMStandby

System Clock Stand-by Mode Register

Access type:

Write

Reset value:

0x00000000

Function:

This register sets stand-by mode.

Bit definitions:

Bit 0

Name:	STDBY (STanDBY mode)
Write values:	0: (Invalid)
	1: Set stand-by mode.

REGRemap

Memory Map Register

Access type:

Read/Write

Reset value:

0x0000000

Function:

This register contains a set of bits that control memory mapping.

Bit definitions:

Bit 0

Name: GENREM (GENeral REdefined Mapping)

Read values: 0: Memory mapping is in "immediately after reset" mode.

1: Memory mapping is in "after remap" mode.

Write values: 0: (Invalid)

1: Switch memory mapping to "after remap" mode.

No effect if memory mapping is already in "after remap" mode.

Bit 1

Name: FLASHVIR (FLASH VIRtual Mapping)

Read values: 0: Virtual flash memory region mapping is disabled.

1: Virtual flash memory region mapping is enabled.

Write values: 0: Disable virtual flash memory region mapping.

1: Enable virtual flash memory region mapping.

RTCCalendar

Real-time Clock Calendar Register

Access type:

Read

Reset value:

0x00980101

Function:

This register contains a set of bits for obtaining calendar information from the real-time clock.

Bit definitions:

Bits 0 .. 5

Name: DATE (RTC DATE counter)

Function: Obtain day as BCD value. The upper 2 bits represent the tens column (0 .. 3) and the lower 4 bits represent the ones column (0 .. 9).

Bits 8 .. 12

Name: MONTH (RTC MONTH counter)

Function Obtain month as BCD value. The uppermost bit represents the tens column (0,1), and the lower four bits represent the ones column (0 .. 9).

Bits 16 .. 23

Name: YEAR (RTC YEAR counter)

Function: Obtain year as BCD value. The upper 4 bits represent the tens column (0 .. 9), and the lower 4 bits represent the ones column (0 .. 9).

RTCClock

Real-time Clock Register

Access type:

Read

Reset value:

0x04000000

Function:

This register contains a set of bits for obtaining time information from the real-time clock.

Bit definitions:

Bits 0 .. 6

Name: SEC (RTC SECond counter)

Function: Obtains the seconds counter as a BCD value. The upper 3 bits represent the tens column (0 .. 5), and the lower 4 bits represent the ones column (0 .. 9).

Bits 8 .. 14

Name: MIN (RTC MINute counter)

Function Obtains the minutes counter as a BCD value. The upper 3 bits represent the tens column (0 .. 5), and the lower 4 bits represent the ones column (0 .. 9).

Bits 16 .. 21

Name: HOUR (RTC HOUR counter)

Function: Obtains the hours counter as a BCD value. The upper 2 bits represent the tens column (0 .. 2), and the lower 4 bits represent the ones column (0 .. 9).

Bits 24 .. 26

Name: DAY (RTC DAY counter)

Function: Obtains the day counter as a BCD value. 1 represents Sunday, 7 represents Saturday.

RTCControl

Real-time Clock Controller Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control the real-time clock.

Bit definitions:

Bit 0

Name: PRGSEL (ProGrammation mode SELector)

Function: Sets the frequency of the counter clock (F_RTC).

Meaning: 0: 1 Hz counter clock (F_RTC) (normal mode).

1: 4 KHz counter clock (F_RTC) (programming mode).

Bits 1 .. 3

Name: CNTSEL(RTC CouNTer SELection)

Function: Select counter for real-time clock.

Read values: 0: Seconds counter.

1: Minutes counter.

2: Hours counter.

3: Day-of-the-week counter.

4: Day counter.

5: Month counter.

6: Year counter.

7: No counter selected.

RTCInc

Real-time Clock Increment Register

Access type:

Write

Reset value:

0x00000000

Function:

This register controls incrementing of the real-time clock. The register automatically resets after incrementing is completed.

Bit definitions:

Bit 0

Name:	INCCNT (RTC INCremen CouNTer)
Write value:	0: Indeterminate.
	1: Increment by 1 with clock (F_RTC).

TIMER[0..2]Control

Timer Control Register

Access type:

Read/Write

Reset value:

0x00000000

Function:

This register contains a set of bits that control the timer.

Bit definitions:

Bits 0 .. 1

Name: CLKDIV (ClocK DIVider)

Function: Selects frequency divider for counter clock.

Meaning 0: Counter clock frequency divider is 2.

1: Counter clock frequency divider is 32.

2: Counter clock frequency divider is 512.

3: (Undefined)

Bit 2

Name: ENTIMER (Enable TIMER)

Function: Enable timer operation.

Meaning: 0: Halt timer.

1: Enable timer.

TIMER[0..2]Load

Timer Load Register

Access type:

Read/Write

Reset value:

0x0000FFFF

Function:

This register is used to set the initial count of the counter.

Bit definitions:

Bits 0 .. 15

Name:	TL (Timer Load)
Meaning:	Initial count (16 bits).

TIMER[0..2]Value

Timer Value Register

Access type:

Read

Reset value:

0x0000FFFF

Function:

This register is used to read the counter value.

Bit definitions:

Bits 0 .. 15

Name of bit:	TV (Timer Value)
Read value:	Counter value. (16 bits)

Appendix B: List of Device Registers

Address	Register name	Access	MSB	Bit field					LSB
0x06000000	REGRemap	R/W						FLASHVIR	GENREM
0x06000004	FLASHREMAPStatus	R					FABORT	DABORT	NOACTIVE	MULTIPLE
0x06000008	FLASHACTIVEBlocks	R/W	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
			IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0x06000010	FLASHDataController	R/W		LOADSGN	LOADPAGE	WAIT	STDBY	BUSY	LOCK	ENPRG
0x06000100	FLASHVIRTUALAddr0	R/W					VIRTUALADDR			
0x06000104	FLASHVIRTUALAddr1	R/W					VIRTUALADDR			
0x06000108	FLASHVIRTUALAddr2	R/W					VIRTUALADDR			
0x0600010C	FLASHVIRTUALAddr3	R/W					VIRTUALADDR			
0x06000110	FLASHVIRTUALAddr4	R/W					VIRTUALADDR			
0x06000114	FLASHVIRTUALAddr5	R/W					VIRTUALADDR			
0x06000118	FLASHVIRTUALAddr6	R/W					VIRTUALADDR			
0x0600011C	FLASHVIRTUALAddr7	R/W					VIRTUALADDR			
0x06000120	FLASHVIRTUALAddr8	R/W					VIRTUALADDR			
0x06000124	FLASHVIRTUALAddr9	R/W					VIRTUALADDR			
0x06000128	FLASHVIRTUALAddr10	R/W					VIRTUALADDR			
0x0600012C	FLASHVIRTUALAddr11	R/W					VIRTUALADDR			
0x06000130	FLASHVIRTUALAddr12	R/W					VIRTUALADDR			
0x06000134	FLASHVIRTUALAddr13	R/W					VIRTUALADDR			
0x06000138	FLASHVIRTUALAddr14	R/W					VIRTUALADDR			
0x0600013C	FLASHVIRTUALAddr15	R/W					VIRTUALADDR			
0x0A000000	INTStatus	R			TC2FIQ	IFIRQ	BATIRQ	VOLIRQ	RTCIRQ	TC1IRQ
			TC0IRQ	SPIIRQ	EXT5IRQ	EXT4IRQ	EXT3IRQ	EXT2IRQ	EXT1IRQ	EXT0IRQ
0x0A000004	INTRawStatus	R			TC2FIQ	IFIRQ	BATIRQ	VOLIRQ	RTCIRQ	TC1IRQ
			TC0IRQ	SPIIRQ	EXT5IRQ	EXT4IRQ	EXT3IRQ	EXT2IRQ	EXT1IRQ	EXT0IRQ
0x0A000008	INTEnableSet	R/W			TC2FIQ	IFIRQ	BATIRQ	VOLIRQ	RTCIRQ	TC1IRQ
			TC0IRQ	SPIIRQ	EXT5IRQ	EXT4IRQ	EXT3IRQ	EXT2IRQ	EXT1IRQ	EXT0IRQ
0x0A00000C	INTEnableReset	W			TC2FIQ	IFIRQ	BATIRQ	VOLIRQ	RTCIRQ	TC1IRQ
			TC0IRQ	SPIIRQ	EXT5IRQ	EXT4IRQ	EXT3IRQ	EXT2IRQ	EXT1IRQ	EXT0IRQ
0x0A000010	INTSourceClear	W			TC2FIQ	IFIRQ	BATIRQ	VOLIRQ	RTCIRQ	TC1IRQ
			TC0IRQ	SPIIRQ						
0x0A800000	TIMER0Load	R/W	TL							
			TL							
0x0A800004	TIMER0Value	R	TV							
			TV							
0x0A800008	TIMER0Control	R/W						ENTIMER	CLKDIV	
0x0A800010	TIMER1Load	R/W	TL							
			TL							

0x0A800014	TIMER1Value	R	TV							
			TV							
0x0A800018	TIMER1Control	R/W						ENTIMER	CLKDIV	
0x0A800020	TIMER2Load	R/W	TL							
			TL							
0x0A800024	TIMER2Value	R	TV							
			TV							
0x0A800028	TIMER2Control	R/W						ENTIMER	CLKDIV	
0x0B000000	PMFrequency	R/W				LOCK	FREQ			
0x0B000004	PMStandby	W								STDBY
0x0B800000	RTCControl	R/W					CNTSEL			PRGSEL
0x0B800004	RTCInc	W								INCCNT
0x0B800008	RTCClock	R						DAY		
					HOUR					
				MIN						
				SEC						
0x0B80000C	RTCCalendar	R								
			YEAR							
						MONTH				
					DATE					
0x0C800000	IFStaticControl	R/W					BFLT	BGEN	STDBY	IFMODE
0x0C800004	IFDynamicControl	R/W								T_EN
0x0D000000	LCDControl	R/W	ROT	DISON	FR		CPEN	DISMOD		
0x0D000100	BackPlane0	R/W	FP31..0							
0x0D000104	BackPlane1	R/W	FP31..0							
0x0D000108	BackPlane2	R/W	FP31..0							
0x0D00010C	BackPlane3	R/W	FP31..0							
0x0D000110	BackPlane4	R/W	FP31..0							
0x0D000114	BackPlane5	R/W	FP31..0							
0x0D000118	BackPlane6	R/W	FP31..0							
0x0D00011C	BackPlane7	R/W	FP31..0							
0x0D000120	BackPlane8	R/W	FP31..0							
0x0D000124	BackPlane9	R/W	FP31..0							
0x0D000128	BackPlane10	R/W	FP31..0							
0x0D00012C	BackPlane11	R/W	FP31..0							
0x0D000130	BackPlane12	R/W	FP31..0							
0x0D000134	BackPlane13	R/W	FP31..0							
0x0D000138	BackPlane14	R/W	FP31..0							
0x0D00013C	BackPlane15	R/W	FP31..0							
0x0D000140	BackPlane16	R/W	FP31..0							

0x0D000144	BackPlane17	R/W	FP31..0							
0x0D000148	BackPlane18	R/W	FP31..0							
0x0D00014C	BackPlane19	R/W	FP31..0							
0x0D000150	BackPlane20	R/W	FP31..0							
0x0D000154	BackPlane21	R/W	FP31..0							
0x0D000158	BackPlane22	R/W	FP31..0							
0x0D00015C	BackPlane23	R/W	FP31..0							
0x0D000160	BackPlane24	R/W	FP31..0							
0x0D000164	BackPlane25	R/W	FP31..0							
0x0D000168	BackPlane26	R/W	FP31..0							
0x0D00016C	BackPlane27	R/W	FP31..0							
0x0D000170	BackPlane28	R/W	FP31..0							
0x0D000174	BackPlane29	R/W	FP31..0							
0x0D000178	BackPlane30	R/W	FP31..0							
0x0D00017C	BackPlane31	R/W	FP31..0							
0x0D800000	PIOControl	R/W					P3DIR	P2DIR	P1DIR	P0DIR
0x0D800004	PIOSetOutputData	W		PIO6	PIO5		PIO3	PIO2	PIO1	PIO0
0x0D800008	PIOClearOutputData	W		PIO6	PIO5		PIO3	PIO2	PIO1	PIO0
0x0D80000C	PIOReadInputData	R				PIO4	PIO3	PIO2	PIO1	PIO0
0x0D800010	DACControl	R/W								STDBY
0x0D800014	DACData	W	DACV							
			DACV							
0x0D800020	LVDControl	R/W						AUTCNT	COMPSTBY	BGSTBY