FM6124 Event Data Recorder with F-RAM

OVERVIEW

The FM6124 is an Event Data Recorder with F-RAM memory that provides an integrated solution for digital events monitoring. Like PLC devices, the FM6124 provides simple device settings and data retrieval allowing easy system integration to short design-in cycle.

Access to the device is performed through an I^2C interface able to sustain communication speed up to 100kbps. The I^2C interface also provides the ability to place the FM6124 away from the host system and closer to the equipment and/or sensors it is intended to monitor. It also allows multiple devices to share the same I^2C bus.

The FM6124 features 12 digital inputs that can be individually configured to trigger event recording on either a rising or a falling edge. An on-chip Real Time Clock (RTC) with calendar provides a timestamp for each event recorded and can also be used as system clock and calendar. The event timestamp resolution is one second.

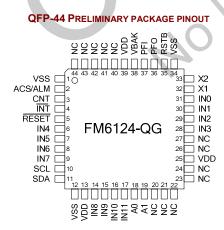
The on-chip 32KBytes F-RAM memory provides nonvolatile storage for event recording and a portion of it can also be used for nonvolatile User Data storage. Access to the User Data is performed like any other I²C memory device. Up to 32KB F-RAM can be reserved for Events recording. F-RAM can be treated as RAM and reads/writes at the speed of the I²C bus. It also offers effectively unlimited write endurance unlike other nonvolatile memory technologies.

Recorded events consist of 8 bytes. One byte defines the event code and the 7 remaining bytes contain timestamp data. The events are recorded in a circular buffer fashion and they are retrieved through l^2C accessible registers.

The FM6124 can capture and record up to 10K Events every second if no I^2C communication in taking place on the I^2C bus. In that case, the Events must last 15µs. During I^2C transactions, the device can still capture and record up to 5K Events per second having a minimum duration of 25µs.

Other features of the FM6124 include a 16-bit battery backed-up event counter, an early power fail monitoring input, and a user programmable 64-bit serial number.

The FM6124 is powered by a 3.0 to 3.6V supply, can function over the industrial temperature range, and is available in a QFP-44 package.



This is a product that has fixed target specifications but are subject to change pending characterization results.



FEATURES

Event Monitoring Features

- Continuously Monitor Input State Change
- 12 Digital Events Inputs Pins
- Configurable Events Trigger on Rising/Falling Edge
- Up to 10K Events per second Capture/Record rate
- Event duration can be as short as 15µS
- RTC Timestamp for each Recorded Event
- I²C Interface for Configuration and Data Read/Write
- Configurable F-RAM Segment Size for Event Recording

High Integration Device Replaces Multiple Parts

- Serial Nonvolatile Memory
- Real-time Clock (RTC) with Alarm
- Low V_{DD} Detection Drives Reset
- Watchdog Window Timer
- Early Power-Fail Warning/NMI
- 16-bit Nonvolatile Event Counter
- Serial Number with Write-lock for Security

Ferroelectric Nonvolatile RAM

- Configurable Size (Up to 24KB) F-RAM for User Data
- Dedicated I²C ID for User F-RAM
- Unlimited Read/Write Endurance
- 10 year Data Retention
 NoDelay[™] Writes

Real-time Clock/Calendar

- Backup Current under 1 µA
- Seconds through Centuries in BCD format
- Tracks Leap Years through 2099
- Uses Standard 32.768 kHz Crystal
- Software Calibration
- Supports Battery or Capacitor Backup

Processor Companion

- Active-low Reset Output for V_{DD} and Watchdog
- Programmable Low-V_{DD} Reset Thresholds
- Manual Reset Filtered and Debounced
- Programmable Watchdog Window Timer
- Nonvolatile Event Counter
- Comparator for Power-Fail Interrupt or Other Use
- 64-bit Programmable Serial Number with Lock

Easy to Use Configurations

- Operates from 3.0 to 3.6V
- QFP-44 10x10mm "Green"/RoHS Package
- Industrial Temperature Range -40°C to +85°C

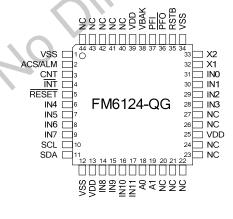
APPLICATIONS

- o Activity Monitoring
- o Industrial Automation Event Recording
- o Environmental Monitoring
- Vehicle & Pedestrian Traffic Counting
- Equipment Use monitoring
- Maintenance scheduling

PIN DESCRIPTION

Pin	Name	1/0	- Function
1	VSS	VSS	Ground
2	ACS	0	Alarm/Calibration/SquareWave: This is an open-drain output that requires an external pull-up resistor. In normal operation, this pin acts as the active-low alarm output. In Calibration mode, a 512 Hz square-wave is driven out. In SquareWave mode, the user
2	neb	Ŭ	may select a frequency of 1, 512, 4096, or 32768 Hz to be used as a continuous output. The SquareWave mode is entered by clearing the AL/SW and CAL bits in register 18h.
3	CNT	Ι	Event Counter Input: This input increments the counter when an edge is detected on this pin. The polarity is programmable and the counter value is nonvolatile or battery-backed, depending on the mode. This pin should be tied to ground if unused.
4	INT	О	Active Low output that can be configured to generate a low level when: -Event buffer is full -Activity on event input
5	RESET	Ι	Device Reset Input. This active-low input clears all volatile registers. Leave unconnected if not used, pin has internal pull-up.
6	IN4		Event Input Pin 4
7	IN5		Event Input Pin 5
8	IN6		Event Input Pin 6
9	IN7		Event Input Pin 7
10	SCL		Serial Clock: The serial clock input for the two-wire interface. Data is clocked out of the device on the SCL falling edge, and clocked in on the SCL rising edge. A pull-up resistor is required.
11	SDA		Serial Data/Address: This is a bi-directional pin used to shift serial data and addresses for the two-wire interface. It employs an open-drain output and is intended to be wire-OR'd with other devices on the two-wire bus. A pull-up resistor is required.
12	VSS	VSS	Ground
13	VDD	Supply	Supply voltage
14	IN8	Ι	Event Input Pin 8
15	IN9	Ι	Event Input Pin 9
16	IN10	Ι	Event Input Pin 10
17	IN11	Ι	Event Input Pin 11
18 19	A0 A1	Ι	Address 1-0: These pins are used to select one of up to 4 devices of the same type on the same two-wire bus. To select the device, the address value on the three pins must match the corresponding bits contained in the device address.
20-24	NC	NC	Leave these pins unconnected
25	VDD	Supply	Supply voltage
26, 27	NC	NC	Leave these pins unconnected
28	IN3	I	Event Input Pin 3
28	IN3 IN2	I	Event Input Pin 2
30	IN2 IN1	I	Event Input Pin 1
31	INI	I	Event Input Pin 0
32	X1	1	32.768 kHz crystal connection. When using an external oscillator, apply the clock to X1 and a DC mid-level to X2 (see Crystal Type section for suggestions).
33	X2		32.768 kHz crystal connection
34	VSS		Ground
35	RSTB	NC	Reset Out: This active-low output is open drain with weak pull-up. It is also an input when used as a manual reset. This pin should be left floating if unused.
36	PFO		Early Power-fail Output: This pin is the early power-fail output and is typically used to drive a microcontroller NMI pin. PFO drives low when the PFI voltage is <1.5V.
37	PFI		Early Power-fail Input: Typically connected to an unregulated power supply to detect an early power failure. This pin must be tied to ground if unused.
38	VBAK		Backup supply voltage: A 3V battery or a large value capacitor. If no backup supply is used, this pin should be tied to VSS and the VBC bit should be cleared.
39	VDD	Supply	Supply voltage
40-44	NC	NC	Leave these pins unconnected

QFP-44 PACKAGE PINOUT



FUNCTIONAL BLOCKS

The functional block diagram of the FM6124 is represented in the figure below. The FM6124 combines the following:

- 12 input pins individually configurable for Event recording
- Event Buffer memory for event storage implemented as F-RAM
- User accessible F-RAM memory
- User Accessible Real time clock (RTC) with alarm
- MCU companion features such as Event counter, Watchdog Timer, Power Fails Input/Output and programmable serial number
- I²C communication interface supporting two device IDs: one for User-F-RAM (0xA0) and one for Event Recorder/MCU Companion (0xD0)

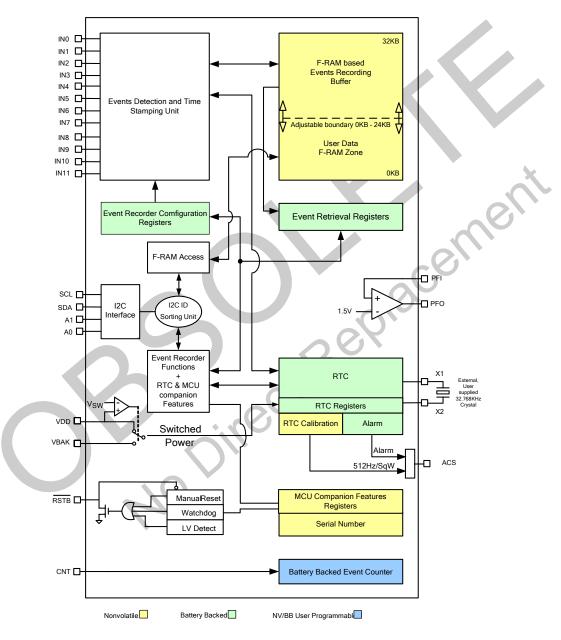


FIGURE 1. FM6124 BLOCK DIAGRAM

Two-wire (I²C) Interface

The FM6124 employs an industry standard two-wire (l^2C) bus that is familiar to many users. This product is unique since it incorporates two logical devices in one chip. Each logical device can be accessed individually and appear to the system software to be two separate products. One is the nonvolatile F-RAM memory device. It has a Slave Address (Slave ID = 1010b) that operates the same as a stand-alone memory device. The second device is Event Data Recorder and Companion which has a different Slave Address (Slave ID = 1101b).

By convention, any device that is sending data onto the bus is the transmitter while the target device for this data is the receiver. The device that is controlling the bus is the master. The master is responsible for generating the clock signal for all operations. Any device on the bus that is being controlled is a slave. The FM6124 is always a slave device.

The bus protocol is controlled by transition states in the SDA and SCL signals. There are four conditions: Start, Stop, Data bit, and Acknowledge. The figure below illustrates the signal conditions that specify the four states.

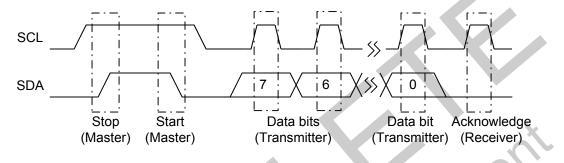


FIGURE 2. I²C BUS CONDITIONS AND TERMINOLOGY

Start Condition

A Start condition is indicated when the bus master drives SDA from high to low while the SCL signal is high. All read and write transactions begin with a Start condition. An operation in progress can be aborted by asserting a Start condition at any time. Aborting an operation using the Start condition will ready the FM6124 for a new operation.

If the power supply drops below the specified VTP during operation, any 2-wire transaction in progress will be aborted and the system must issue a Start condition prior to performing another operation.

Stop Condition

A Stop condition is indicated when the bus master drives SDA from low to high while the SCL signal is high. All operations must end with a Stop condition. If an operation is pending when a stop is asserted, the operation will be aborted. The master must have control of SDA (not a memory read) in order to assert a Stop condition.

Data/Address Transfer

All data transfers (including addresses) take place while the SCL signal is high. Except under the two conditions described above, the SDA signal should not change while SCL is high.

Acknowledge

The Acknowledge (ACK) takes place after the 8th data bit has been transferred in any transaction. During this state the transmitter must release the SDA bus to allow the receiver to drive it. The receiver drives the SDA signal low to acknowledge receipt of the byte. If the receiver does not drive SDA low, the condition is a No-Acknowledge (NACK) and the operation is aborted.

The receiver might NACK for two distinct reasons. First is that a byte transfer fails. In this case, the NACK ends the current operation so that the part can be addressed again. This allows the last byte to be recovered in the event of a communication error.

Second and most common, the receiver does not send an ACK to deliberately terminate an operation. For example, during a read operation, the FM6124 will continue to place data onto the bus as long as the receiver sends ACKs (and clocks). When a read operation is complete and no more data is needed, the receiver must NACK the last byte. If the receiver ACKs the last byte, this will cause the FM6124 to attempt to drive the bus on the next clock while the master is sending a new command such as a Stop.

Slave Address

The first byte that the FM6124 expects after a Start condition is the slave address. As shown in figures below, the slave address contains the Slave ID, Device Select address, and a bit that specifies if the transaction is a read or a write.

The FM6124 has two Slave Addresses (Slave IDs) associated with two logical devices. To access the memory device, bits 7-4 should be set to 1010b. The other logical device within the FM6124 is the Event Recorder configuration and data access, the real-time clock and MCU companion. To access this device, bits 7-4 of the slave address should be set to 1101b. A bus transaction with this slave address will not affect the memory in any way. The figures below illustrate the two Slave Addresses.

The Device Select bits allow multiple devices of the same type to reside on the 2-wire bus. The device select bits (bits 2-1) select one of four parts on a two-wire bus. They must match the corresponding value on the external address pins in order to select the device. Bit 0 is the read/write bit. A "1" indicates a read operation, and a "0" indicates a write operation.

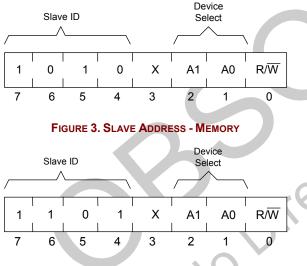


FIGURE 4. SLAVE ADDRESS - EDR/COMPANION

Addressing Overview – Memory

After the FM6124 acknowledges the Slave Address, the master can place the memory address on the bus for a write operation. The address requires two bytes.

The first is the MSB (upper byte). Following the MSB is the LSB (lower byte) which contains the remaining eight address bits. The address is latched internally. Each access causes the latched address to be incremented automatically. The current address is the value that is held in the latch, either a newly

written value or the address following the last access. The current address will be held as long as VDD > VTP or until a new value is written. Accesses to the clock do not affect the current memory address. Reads always use the current address. A random read address can be loaded by beginning a write operation as explained below.

After transmission of each data byte, just prior to the Acknowledge, the FM6124 increments the internal address. This allows the next sequential byte to be accessed with no additional addressing externally. After the last address is reached, the address latch will roll over to 0000h. There is no limit to the number of bytes that can be accessed with a single read or write operation.

Addressing Overview - EDR, RTC & Companion

The Event Recorder, RTC, and Processor Companion operate in a similar manner to the memory, except that it uses only one byte of address. Addresses 00h to 33h corresponds to special function registers. Attempting to load addresses above 33h is an illegal condition; the FM6124 will return a NACK and abort the 2-wire transaction.

Data Transfer

After the address information has been transmitted, data transfer between the bus master and the FM6124 begins. For a read, the FM6124 will place 8 data bits on the bus then wait for an ACK from the master. If the ACK occurs, the FM6124 will transfer the next byte. If the ACK is not sent, the FM6124 will end the read operation. For a write operation, the FM6124 will accept 8 data bits from the master then send an Acknowledge. All data transfer occurs MSB (most significant bit) first.

Memory Write Operation

All memory writes begin with a Slave Address, then a memory address. The bus master indicates a write operation by setting the slave address LSB to a 0. After addressing, the bus master sends each byte of data to the memory and the memory generates an Acknowledge condition. Any number of sequential bytes may be written. If the end of the address range is reached internally, the address counter will wrap to 0000h. Internally, the actual memory write occurs after the 8th data bit is transferred. It will be complete before the Acknowledge is sent. Therefore, if the user desires to abort a write without altering the memory contents, this should be done using a Start or Stop condition prior to the 8th data bit. The figures that follow illustrate a single- and multiple-writes to memory.

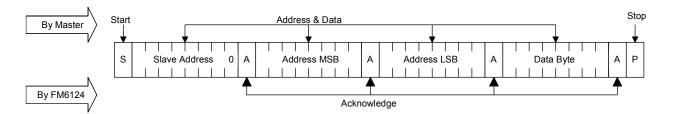
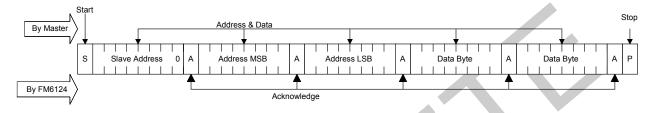


FIGURE 5. SINGLE BYTE MEMORY WRITE





Memory Read Operation

There are two types of memory read operations. They are current address read and selective address read. In a current address read, the FM6124 uses the internal address latch to supply the address. In a selective read, the user performs a procedure to first set the address to a specific value.

Current Address & Sequential Read

As mentioned above the FM6124 uses an internal latch to supply the address for a read operation. A current address read uses the existing value in the address latch as a starting place for the read operation. The system reads from the address immediately following that of the last operation.

To perform a current address read, the bus master supplies a slave address with the LSB set to 1. This indicates that a read operation is requested. After receiving the complete device address, the FM6124 will begin shifting data out from the current address on the next clock. The current address is the value held in the internal address latch.

Beginning with the current address, the bus master can read any number of bytes. Thus, a sequential read is simply a current address read with multiple byte transfers. After each byte the internal address counter will be incremented. Each time the bus master acknowledges a byte, this indicates that the FM6124 should read out the next sequential byte.

There are four ways to terminate a read operation. Failing to properly terminate the read will most likely create a bus contention as the FM6124 attempts to read out additional data onto the bus. The four valid methods follow.

- 1. The bus master issues a NACK in the 9th clock cycle and a Stop in the 10th clock cycle. This is illustrated in the diagrams below and is preferred.
- 2. The bus master issues a NACK in the 9^{th} clock cycle and a Start in the 10^{th} .
- 3. The bus master issues a Stop in the 9th clock cycle.
- 4. The bus master issues a Start in the 9th clock cycle.

If the internal address reaches the top of memory, it will wrap around to 0000h on the next read cycle. The figures below show the proper operation for current address reads.

Selective (Random) Read

There is a simple technique that allows a user to select a random address location as the starting point for a read operation. This involves using the first three bytes of a write operation to set the internal address followed by subsequent read operations. To perform a selective read, the bus master sends out the slave address with the LSB set to 0. This specifies a write operation. According to the write protocol, the bus master then sends the address bytes that are loaded into the internal address latch. After the FM6124 acknowledges the address, the bus master issues a Start condition. This simultaneously aborts the write operation and allows the read command to be issued with the slave address LSB set to a 1. The operation is now a read from the current address. Read operations are illustrated below.

RTC/Companion Write Operation

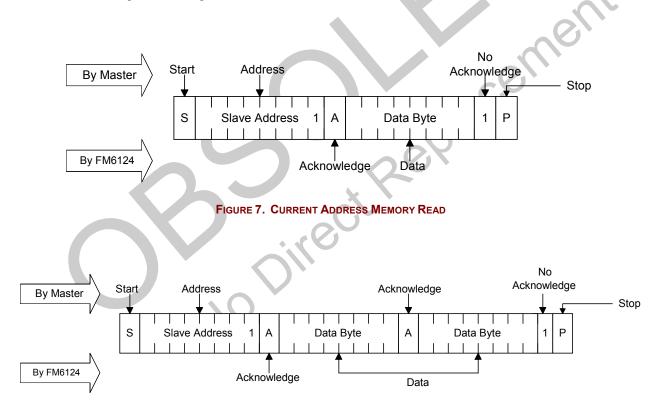
All RTC and Companion writes operate in a similar manner to memory writes. The distinction is that a different device ID is used and only one byte address is needed instead of two. Figure 10 illustrates a single byte write to this device.

RTC/Companion Read Operation

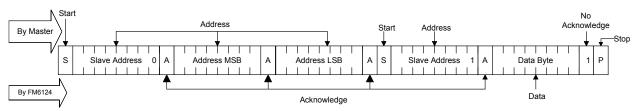
As with writes, a read operation begins with the Slave Address. To perform a register read, the bus

master supplies a Slave Address with the LSB set to 1. This indicates that a read operation is requested. After receiving the complete Slave Address, the FM6124 will begin shifting data out from the current register address on the next clock. Auto-increment operates for the special function registers as with the memory address. A current address read for the registers look exactly like the memory except that the device ID is different.

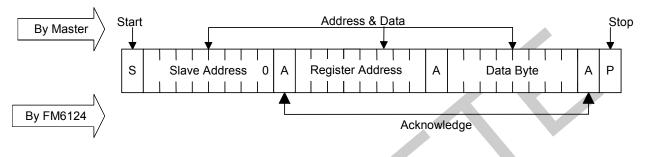
The FM6124 contains two separate address registers, one for the memory address and the other for the register address. This allows the contents of one address register to be modified without affecting the current address of the other register. For example, this would allow an interrupted read to the memory while still providing fast access to an RTC register. A subsequent memory read will then continue from the memory address where it previously left off, without requiring the load of a new memory address. However, a write sequence always requires an address to be supplied.













Delay When Switching from EDR/Companion ID to User F-RAM ID

When switching from FM6124 EDR/Companion ID to the User F-RAM ID, there will be a delay of ~100 μ s during which the FM6124 may not acknowledge to I²C ID sent to it. This delay is required for the internal logic of the FM6124 to perform the switchover from one ID to another.

If the host attempts to initiate a transaction to the FM6124 with a different slave address within this

100µs period, it is recommended that the host initiate a read command to the FM6124 with the ID projected to be used with the R/W bit set to 1 and then send a STOP if the FM6124 fails to respond.

At 100 kHz I²C communication speed a 100 μ S delay correspond to approximately one I²C read command. This means that if the FM6124 fails to respond to the first read operation, it will respond on the second one.

Register Map

The FM6124 Event Recorder, RTC, and processor companion functions are accessed via 51 special function registers, which are mapped to unique commands. The interface protocol is described in details in the following pages. The registers contain timekeeping data, alarm settings, control bits, and information flags. A description of each register follows the summary table.

TABLE 1. FM6124 REGISTER MAP OVERVIEW

Address					Data				Function	Туре
	7	6	5	4	3	2	1	0		
0x33		10 Ye	ears			Ye	ear		Event BCD Year	RO
0x32		10 Mo	nths		Month			Event BCD Month	RO	
0x31	10 Date				Da	ate		Event BCD Date	RO	
0x30				Da	y of week				Event BCD Day of Week	RO
0x2F		10 Ho	ours			Ho	ours		Event BCD Hours	RO
0x2E		10 Mir	nutes			Min	utes		Event BCD Minutes	RO
0x2D		10 Sec	onds			Sec	onds		Event BCD Seconds	RO
0x2C				Ev	vent Code				Event Code	RO
0x2B			Nu	mber of Un	read Events 1	Pecorded			Unread Events Counter MSB	RO
0x2A			INU			Recolucu			Unread Events Counter LSB	RO
0x29	P11	P10	P9	P8	P7	P6	P5	P4	Pin Pass-through B	RO
0x28		Reser	ved		P3	P2	P1	P0	Pin Pass-through A	RO
0x27			Re	served			NBEV	SNAP	Pin State-#Events snapshot	WO
0x26	P11	P10	P9	P8	P7	P6	P5	P4	Pin Event Enable B	R/W
0x25		Reser	ved		P3	P2	P1	P0	Pin Event Enable A	R/W
0x24	P11	P10	P9	P8	P7	P6	P5	P4	Pin Event Rise/Fall B	R/W
0x23		Reser	ved		P3	P2	P1	P0	Pin Event Rise/Fall A	R/W
0x22	P11	P10	P9	P8	P7	P6	P5	P4	Pin Event Interrupt B	R/W
0x21	CLEAR	BF	B75F	BHF	P3	P2	P1	P0	Pin Event Interrupt A	R/W
0x20	EBUFS	[ZE[1:0]	ERR	DIR		EDRC	MD[3:0]	10	Event Recorder Control	R/W
0x1D	~M	0	0	10 mo		Мс	onth		Alarm BCD Month	R/W
0x1C	~M	0	10	date		Da	ate		Alarm BCD Date	R/W
0x1B	~M	0	10	hours		Ho	ours		Alarm BCD Hours	R/W
0x1A	~M		10 minute	s		Min	utes		Alarm BCD Minutes	R/W
0x19	~M		10 second	s		Sec	onds		Alarm BCD Seconds	R/W
0x18	SNL	AL/SW	F1	F0	VBC	FC	VTP1	VTP0	Companion Control	R/W
0x17				Serial N	Number Byte	7			Serial number 7	R/W
0x16				Serial N	Number Byte	6			Serial number 6	R/W
0x15				Serial N	Number Byte	5			Serial number 5	R/W
0x14				Serial N	Number Byte	4			Serial number 4	R/W
0x13				Serial N	Number Byte	3			Serial number 3	R/W
0x12				Serial N	Number Byte	2			Serial number 2	R/W
0x11	Serial Number Byte 1								Serial number 1	R/W
0x10				Serial N	Number Byte	0			Serial number 0	R/W
0x0F				16-bit "CN	T Pin" Edge	Count			Edge Count MSB	RO
0x0E				io on en	I I III Luge	Count			Edge Count LSB	RO
0x0D	NVC	-	-	-	RC	WC	POLL	СР	Edge Count Control	R/W
0x0C	WDE	-	-	WDET4	WDET3	WDET2	WDET1	WDET0	Watchdog flags	R/W
0x0B	-	-	-	WDST4	WDST3	WDST2	WDST1	WDST0	Watchdog flags	R/W
0x0A	-	-	-	-	WR3	WR2	WR1	WR0	Watchdog Restart	R/W

0x09	EWDF	LWDF	POR	LB	-	-	-		Watchdog flags	R/W
0x08	10 year			Year				BCD Year	R/W	
0x07		10 m	nonth		Month				BCD Month	R/W
0x06	10 date			Date			BCD Date	R/W		
0x05	Day of week						BCD Day of Week	R/W		
0x04	10 hours			Hours			BCD Hours	R/W		
0x03		10 minutes				Min	utes		BCD Minutes	R/W
0x02		10 se	conds		Seconds				BCD Seconds	R/W
0x01			CALS	CAL4	CAL3	CAL2	CAL1	CAL0	CAL Control	R/W
0x00	~OSCEN	AF	CF	AEN	reserved	CAL	W	R	RTC Control	R/W

Battery-backed = \Box

Nonvolatile =

BB/NV User Programmable =

eplacemen

Note: When the device is first powered up and programmed, all timekeeping registers must be written because the battery-backed register values cannot be guaranteed. The table below shows the default values of the nonvolatile registers and some of the battery-backed bits. All other register values should be treated as unknown.

Default Register Values

Address	Hex Value
33h	0x00
32h	0x00
31h	0x00
30h	0x00
2Fh	0x00
2Eh	0x00
2Dh	0x00
2Ch	0x00
2Bh	0x00
2Ah	0x00
29h	0x00
28h	0x00
27h	0x00
26h	0x00
25h	0x00
24h	0x00
23h	0x00
22h	0x00
21h	0x00
20h	0x00
1Eh-1Fh	undefined

Address	Hex Value
1Dh	0x81
1Ch	0x81
1Bh	0x80
1Ah	0x80
19h	0x80
18h	0x40
17h	0x00
16h	0x00
15h	0x00
14h	0x00
13h	0x00
12h	0x00
11h	0x00
10h	0x00
0Fh	0x00
0Eh	0x00
0Dh	0x01
0Ch	0x00
0Bh	0x00
01h	0x00
00h	0x80

Event Recorder with Timestamp

The main feature of the FM6124 is its event recording capability. The FM6124 can monitor events occurring on each of its 12 digital input pins. When an event occurs the Event is recorded into the Event Buffer F-RAM memory along with current the timestamp. The recorded event data is retrieved through I^2C mapped registers.

The FM6124 is a highly integrated part able to operate in standalone mode and requiring a few external components to operate.

Dedicated F-RAM for Event Recording

Based on profiles set by the user, Events are recorded in nonvolatile F-RAM memory. Each event is timestamped automatically. A programmable amount of nonvolatile storage is available to record events (25%, 50%, 75% or 100%). Event recording is triggered by pin state changes. Events will be recorded in a circular buffer unless emptied by the host.

A host processor can download the Event log at any time via the I^2C interface. In addition the various resources such as the input pins and the RTC can be read directly through the serial interface.

Event Definition

An Event is defined as either a rising or a falling transition occurring on any given input pin. Each one of the input pin can be individually configured to react on a rising edge or a falling edge.

Two registers located at addresses 0x23 and 0x24 allow one to individually configure each one of the FM6124 Input pin to trigger an event recording on either a Low to High or a High to Low transition

- Setting the corresponding bit to 1 will trigger an Event recording on a Low to High transition
- Setting the corresponding bit to 0 will trigger an Event recording on a High to Low transition

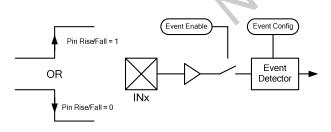


FIGURE 11. EVENT TYPE SUPPORTED

Simultaneous transitions on distinct input pins configured to react to these transitions will be considered as distinct events and they will be recorded as such.

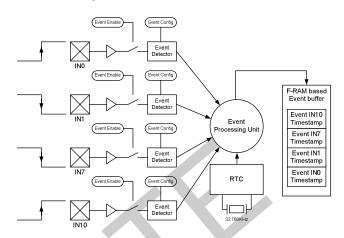


FIGURE 12. EVENT DETECTION AND STORAGE INTO F-RAM

The recording order of simultaneous events will be related to the input numbering. For example, if events occur simultaneously on IN0, IN1, IN7 and IN10, the first event recorded will be the one occurring on IN0 followed by the one on IN1 then IN7 and IN10.

When simultaneous events occur, it is possible that the timestamp recorded varies by 1 second.

Event Timestamp Content

Each time an event is recorded, the following parameters are saved in the nonvolatile Event buffer:

Event Timestamp content

_	Event Second
	Event Minute
	Event Hour
	Event Day of Week
	Event Date
	Event Month
	Event Year
	Event Code

FIGURE 13. EVENT TIMESTAMP CONTENT

Event Memory/User F-RAM Memory Size Configuration

The FM6124 contains a total of 32KBytes (256Kb) of F-RAM memory on chip. The F-RAM memory can be configured to serve for both Event Recording and for User Data saving. The portion of the memory reserved for User Data is defined through register and occupies the lower portion of the address range.

The upper portion of the F-RAM addresses is used to store Event type and timestamp information. The data associated with Event is accessible only through I²C Mapped registers.

The boundary between User FRAM and Event Recording FRAM is adjusted through configuration register in the Event recording portion of the device. The memory boundary can be changed at any time, however each time it is changed the entire F-RAM memory (Event Recording/ User Data) will be erased.

The portion of the F-RAM defined as User Memory Data memory use consistent two-byte addressing for the memory device rendering it code compatible to the standalone memory counterparts, such as the FM24xx but with the ability to be configured up to 24KB in size.

Up to 4000 events can be saved in the FM6124 Event buffer F-RAM memory. A percentage of the F-RAM can also be configured as User F-RAM that is accessed like standard I^2C based F-RAM and using a dedicated I^2C device ID for User F-RAM access.

The EBUFSIZE[1:0] portion of the Event Data Recorder control register (address 0x20) defines the portion of memory reserved for Event recording and User F-RAM size as shown in the table below:

EBUFSIZE[1:0]	Max number of Events	User F-RAM size
00	4000	0
01	3000	64 Kb
10	2000	128 Kb
11	1000	192 Kb

When the entire F-RAM memory is reserved for Event recording, there will be no User F-RAM available and the FM6124 will stop acknowledging on any I^2C transactions initiated with F-RAM / EEPROM device ID.

This allows one to share the I^2C bus between up to four FM6124 and up to eight I^2C based memory devices.

Event Buffer Architecture

The structure of the Event Buffer memory is analogue to a Circular buffer: Initially the Event data will be stored from a base address that we will call FP, for First pointer

Initially the FP pointer is likely to be at lowest possible F-RAM address. However, when the event buffer is full the address of the FP pointer will be incremented for each new event recorded.

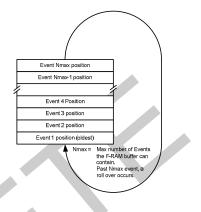


FIGURE 14. EVENT BUFFER OVERVIEW

Event Buffer Pointers

The management of Event recording and retrieval is handled using nonvolatile F-RAM based virtual pointers.

The addresses where these pointers point to are not directly accessible through the I^2C interface however the FM6124 provides commands to control the way those pointers behave.

Four Pointers are defined:

FP:	First pointer
RP:	Read Pointer
SP:	Stream Pointer
WP:	Write pointer

<u>First pointer</u>

The FP pointer is the reference pointer which is fixed for a given Event recorder configuration.

The first pointer actually indicates the position of oldest event recorded and it is used as a reference. The movements of the RP and WP pointers are referenced to the FP pointer.

Each time the FM6124 configuration is changed through EBUFSIZE[1:0] register, the position of FP will be reset to the lowest F-RAM address and it will not move until the Event buffer is filled.

When the number of event recorded exceeds the buffer capacity, the FP pointer address will be incremented each

time a new event is recorded in order to point to the oldest event.

Write pointer

The Write pointer is used by the Event recorder to indicate where the next Event will be recorded in the event buffer.

The WP pointer can only move in one direction: up from the FP address. For this reason the Event storing can be seen as a stack: more recent events being stored on top of older ones.

Eventually the WP pointer can roll over the maximum address. When this situation occurs the FP pointer will be incremented for each new event recorded. In applications where the EDR is placed close to the Host processor it is possible to configure the Event recorder to activate the INT output for the following situations:

- Buffer Full
- Buffer full at 75%
- Buffer full at 50%

This is done by setting the BF, B75F, B50F bit respectively in the PINEVENTA register

For situations where the FM6124 is remote from the host processor, it is always possible for the host processor to retrieve the number of event present in the event buffer by reading the Event buffer counter 16-bit register.

Read Pointer

The read pointer RP points to the next event to be read. This pointer is used to load the next (or previous) event data into the Event data read back registers (addresses 0x2C to 0x33) whenever the GET command (EDRCMD[3:0] = 0001 is sent to the FM6124.

Contrary to WP, the Read pointer can be configured to move from older to newer events but also from newer event toward older ones.

The direction of the RP pointer depends on:

- The amount of events stored into the Event memory buffer
- The value of the READDIR bit

When the READDIR bit is cleared, the RP will fetch event from FP toward WP. However RP cannot move farther than WP-1. In situation where FP has reached WP-1 any attempt to read extra events will make the Event data recorder to fill the Event data read back registers with 0xFF, set the ERR bit of the Event Data recorder configuration register and the RP address will not be incremented.

When the READDIR bit is set and the GET command is initiated, the RP will fetch the data associated with the next

events toward FP and place its content into the Event Data Read back registers. This provided that the RP pointer is "away" and up from FP address.

In the case where the GET command is sent to the Event Recorder while the RP = FP, if there is one event recorded at FP position its content will be placed in the Event Data register.

However if from that point a second attempt is made to read Event Data, the Event Data Recorder will fill the Event data readback registers with 0xFF, set the ERR bit of the Event Data recorder configuration register and the RP address will not be incremented (decremented).

Stream Pointer

The stream pointer is a dedicated read pointer that is used for event reading in stream operations. From a functional point of view the SP pointer is independent of the RP.

However, like the RP pointer, the direction into which the SP pointer will mode, depend on configuration of the DIR bit of the Event Buffer Control register: When the DIR bit is configured as 0, the SP pointer will mode from oldest event toward newer event. In the situation where the DIR bit is set to 1, the SP pointer moves from newer events towards older ones.

The value of SP is initialized at the moment the STREAM command (EDRCMD[3:0] = 0011 is initiated.

The SP pointer is used to automatically load the next event data into the Event data read back registers (addresses 0x2C to 0x33) after the last register (0x33) of the previous event content is read and the STREAM Event data recorder command is maintained.

As other pointers, the SP pointer address is not accessible to the user.

Retrieving the Number of Unread Events

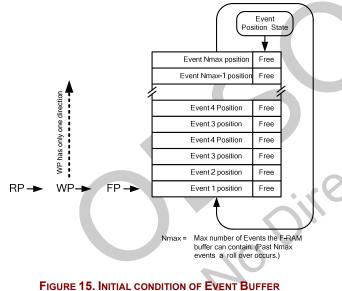
The FM6124 features a 16-bit register that indicate the number of unread event present in the Event Buffer memory. This 16-bit number actually corresponds to the number of events between WP and RP pointers. It is accessible through I^2C registers addresses 0x2A and 0x2B.

Before accessing the Unread Event counter, the host processor must latch a internal registers content into the 16-bit register.

This is performed by writing 0x02 into the Pin/Event Snapshot register.

Event buffer initial condition

Every time the FM6124 Event Buffer memory size configuration is changed, by Event Buffer Memory will be reinitialized the events records that may still be present into the F-RAM will be erased. This re-initialization process takes $\sim 100 \mu s$ and during that time no events will be recorded and I²C communication should be stopped. After initialization, the RP, WP, and FP pointers will all be pointing at the base memory address as shown in the diagram below.



The following example demonstrates the state of the pointers after 3 events have been captured and stored in F-RAM based Event buffer; the pointers will be positioned as shown in the figure below:

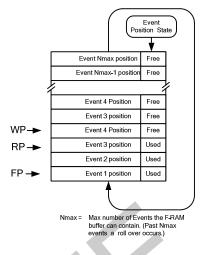


FIGURE 16. INTERNAL POINTERS AFTER 3 EVENTS

The FP, WP, and RP pointers will be set as follow:

- WP will point to the next free position
- RP point to the next Event to be read
- FP is fix

Each time a new Event is recorded the WP address is incremented. When WP increments beyond the Nmax position, it will roll-over to FP and so on. If FP reached WP, an error condition will occur.

As mentioned earlier, after a number of events have been recorded and a number of events have been read, the RP pointer will be away from FP and the WP pointer. In that situation the Event data recorder makes possible to retrieve either newer or older events by configuring the DIR bit accordingly. The following diagram illustrates the impact on the DIR register on the RP operation.

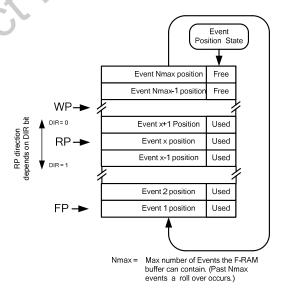


FIGURE 17. RP POINTER DIRECTION

EDR COMMAND SET

The FM6124 responds to commands written to address 0x20, through the I²C serial port. There are nine unique commands, eight of which are used to retrieve event data, the remaining one is used to set the partition between User Memory and Event Memory. The commands and their effect are listed below.

Table of Comm			
Command	EDRCMD[0:3]	DIR	EFFECT
SET DIR	0000	0	set direction dirrection to increment
OET DIR	0000	1	set pointer dirrection to decrement
OFT	0004	0	IF RP ≠ WP THEN read_buffer = event_buffer[RP] increment RP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF,FF] ERR = 0x01
GET	0001	1	IF RP ≠ FP THEN read_buffer = event_buffer[RP] decrement RP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF,FF] ERR = 0x01
GET KEEP	0010	х	read_buffer = event_buffer[RP]
STREAMING	0011	0	IF RP ≠ WP THEN read_buffer = event_buffer[SP] increment RP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF] ERR = 1 ENDIF SP = RP
GET	0011	1	IF RP ≠ FP THEN read_buffer = event_buffer[SP] decrement RP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF,FF] ERR = 1 ENDIF SP = RP
STREAMING GET KEEP	0100	0	IF SP ≠ WP THEN read_buffer = event_buffer[SP] increment SP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF,FF] ERR = 0x01 IF SP ≠ FP THEN
		1	read_buffer = event_buffer[SP] decrement SP ELSE read_buffer = [FF,FF,FF,FF,FF,FF,FF,FF] ERR = 0x01
SKIP	0101	0	IF (WP - RP) > 1 THEN increment RP IF RP > FP THEN decrement RP
FIRST	0110	х	RP = FP
LAST	0111	х	RP = WP - 1
SET EVENT BUFFER SIZE	1000	x	IF COMMAND[7:6] ≠ EBUFSIZE[1:0] THEN SP = RP = FP = WP = 0x00 event_buffer = [00,00,00,,00,00,00]
RESERVED	1001 to 1111	x	No Action

Each command is an eight bit assemblage of lesser registers EBUFSIZE[1:0], ERR, DIR & EDRCMD[3:0] The sequence of concatenation is shown in the table below.

COMMAND STRUCTURE: Address 0x20							
	Bit						
7	6	5	4	3	2	1	0
EBUFS	EBUFSIZE[1:0] ERR DIR EDRCMD[3:0]						

The EDRCMD[3:0] register specifies which of the nine unique commands should be executed.

The DIR register specifies whether the event pointer used by the command, should be incremented or decremented after the successful completion of the command.

The ERR register is used to signal the host, that the event pointer used during the last command can move no further in the direction specified by DIR.

The EBUFSIZE[1:0] register specifies the partitioning of User memory and Event Memory as indicated in the table below.

PARTITION SIZE:							
EBUFSIZE[1:0]	Event Memory	User Memory					
00	4000 events	0kB					
01	3000 events	8kB					
10	2000 events	16kB					
11	1000 events	24kB					
r							

ISSUING A COMMAND

When the host is ready to issue a command, the following procedure should be used.

Start I²C Send the EDR ID & [R/W] = 0 or write Send the command register start address "0x20" Send the EBUFSIZE[1:0]& ERR & DIR & EDRCMD[3:0] Stop I²C

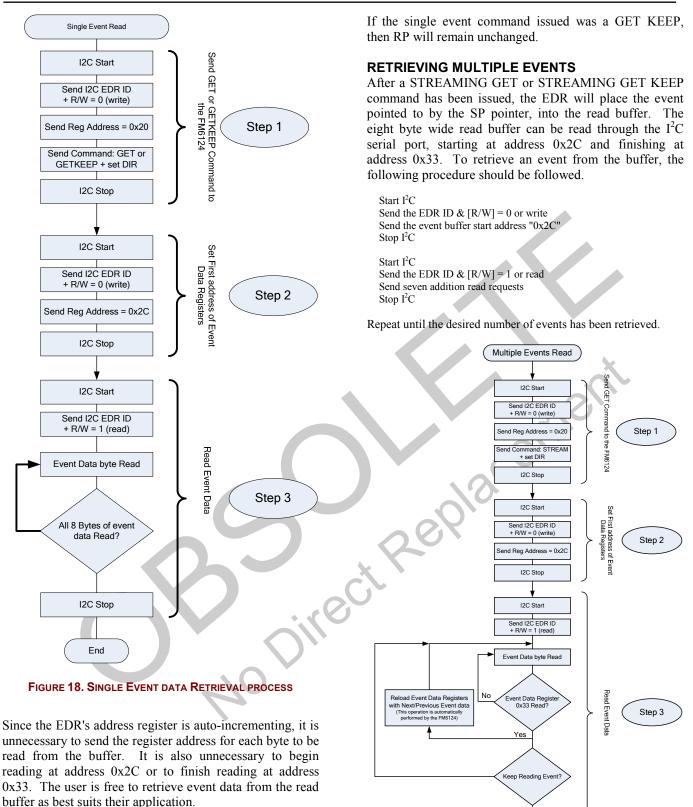
RETRIEVING SINGLE EVENTS

After a GET or KEEP command has been issued, the EDR will place the event pointed to by RP into the read buffer. The eight byte wide read buffer can be read through the I^2C serial port, starting at address 0x2C and finishing at address 0x33. To retrieve an event from the buffer, the following procedure should be followed.

Start I²C Send the EDR ID & R/W bit = 0 (write) Send the event buffer start address "0x2C" Stop I²C

Start I^2C Send the EDR ID & R/W = 1 (read) Send seven addition read requests Stop I^2C

RAMTRON



If the single event read command issued was a GET, then RP will be incremented or decremented as indicated by DIR.

FIGURE 19. MULTIPLE EVENTS RETRIEVAL

I2C Stop

End

Since the EDR's I^2C address register is auto-incrementing, it is unnecessary to send the subsequent address for each byte read from the buffer. It is also unnecessary to begin reading at address 0x2C. In order for the EDR to automatically place the next event pointed to by SP into the read buffer, you must read address 0x33. With this single exception, the user is free to retrieve event data from the read buffer as best suits their application.

If the multiple event read command issued was a STREAMING GET, then RP will be incremented or decremented as indicated by DIR.

If the multiple event read command issued was a STREAMING GET KEEP, then RP will remain unchanged.

Event Skipping Command

The SKIP Command (0101) can be used to increment RP (DIR = 0) or decrement RP (DIR = 1). Note that **RP** will always stay at least one Event before **WP** and that it will never be decremented "below" **FP**.

FIRST and LAST Commands

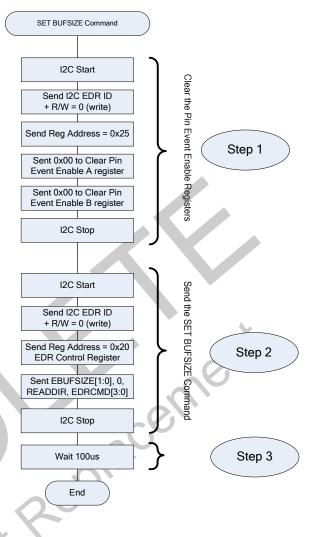
With the FIRST command, the user can move **RP** to the oldest Event immediately ($\mathbf{RP} = \mathbf{FP}$) With the LAST command, the user can move **RP** to the newest Event immediately ($\mathbf{RP} = \mathbf{WP} - 1$)

SET EVENT BUFFER SIZE Command

The SET EVENT BUFFER SIZE command will write the current value of EBUFSIZE[1:0] to the control register. The EBUFSIZE[1:0] register is only written to the control register during a SET EVENT BUFFER SIZE command and ignored at all other times.

When a SET EVENT BUFFER SIZE command is issued with a new EBUFSIZE[1:0] value, the event data memory is reinitialized and the event records stored there are lost. Reinitialization will only occur if the current value of EBUFSIZE[1:0] is different than the value stored in the control register.

Reinitialization takes approximately 100μ s to complete and during this period no event records can be generated and I²C communication should be suspended. Before issuing a SET EVENT BUFFER SIZE command, it is recommended to first disable event recording by clearing the pin event registers A & B, at addresses 0x25 & 0x26 respectively. The following procedure should be used.





In situations where the buffer size is unchanged but its content and pointers needs to be reinitialized, the host system should perform the following operations:

- 1. Clear the Pin Event Enable registers (Step 1 of previous diagram)
- 2. Send the SET EVENT BUFFER SIZE command with a different EBUFSIZE[1:0] than the current one (Step 2 of previous diagram)
- 3. Wait 100µS for the initialization process to complete (Step 3 of previous diagram)
- 4. Send the SET EVENT BUFFER SIZE command with the desired EBUFSIZE[1:0] value (Step 2 of previous diagram, repeat)
- 5. Wait 100µS for the initialization process to complete (Step 3 of previous diagram, repeat)

RSTB and RESET input pins

The FM6124 features an active-low RESET input pin. Applying a manual reset or a power-up Reset of the FM6124 will clear the volatile registers. However this will have no impact on the Event buffer content, their associated pointers, or the MCU companion registers.

The RESET pin features an internal pull-up resistor so if the reset pin in not used, it can be left unconnected.

Event Deletion

Events are not actually deleted until the circular buffer is overwritten with new data. Pointers are moved to effectively prevent access to automatically discarded data. A pointer move is the last thing done when an event is popped off of the buffer.

Safety provisions

To prevent data loss or corruption while Events are being read with the KEEP bit set to 0, Event Read Pointer is only moved at the following times:

- After all of the data of an event has been stored
- After the 8th data byte is transmitted when steaming Events
- As soon as 1 byte is written over the oldest Event, when recording an Event in a full buffer condition

The recording of the event data in the F-RAM Event buffer memory require $\sim 100\mu$ S per event. In the case of simultaneous events occurring on all 12 input of the FM6124, a total of 12 x 100μ S = 1.2ms will be required for the recording of all event Data. During that period of time, event can still be registered, but the Event content will be held into volatile registers.

If the supply voltage is lost before the completion of all events data transfer into F-RAM memory, the events that are still in volatile register will be lost. The device will resume normal operation when the supply comes back.

<u>Pin Snapshot</u>

It is possible to see the state of any pin at any time. Write 1 to the SNAP bit to capture the state of all pins. This bit will be automatically cleared (Write-Only).

Error Conditions

In situations where an error condition can occur, data sent back will be 0xFF. This includes reading illegal addresses and requesting more events and the buffer currently holds. The ERR bit will be set to 1 and Event registers will be set to 0xFF until Control Buffer is written.

Output Interrupt: INT pin

The INT pin is an active low output that will react on:

- Any event occurring on any input pin for which the corresponding bit of the "Gen. INT pulse on Pin Event" Registers have been set to 1.
- Full Buffer condition.

The setting of the "Gen. INT pulse on Pin Event" registers and resulting activity on INT pin is independent of corresponding event recording activation.

MCU Companion

The FM6124 includes a real-time clock (RTC) with alarm and a processor companion along with the EDR serial nonvolatile F-RAM. The companion is a highly integrated peripheral including a low- V_{DD} reset, a programmable watchdog timer, a 16-bit nonvolatile event counter, a comparator for early power-fail detection or other purposes, and a 64-bit serial number.

The real-time clock and supervisor functions are accessed under their own commands. The RTC/alarm and some control registers are maintained by the power source on the VBAK pin, allowing them to operate from battery or backup capacitor power when V_{DD} drops below a set threshold.

Processor Supervisor

Supervisors provide a host processor two basic functions: Detection of power supply fault conditions and a watchdog timer to escape a software lockup condition. The FM6124 device has a reset pin (RSTB) to drive a processor reset input during power faults, power-up, and software lockups. It is an open drain output with a weak internal pull-up to V_{DD}. This allows other reset sources to be wire-OR'd to the RSTB pin. When V_{DD} is above the programmed trip point, RSTB output is pulled weakly to V_{DD} . If V_{DD} drops below the reset trip point voltage level (V_{TP}) , the RSTB pin will be driven low. It will remain low until V_{DD} falls too low for circuit operation which is the V_{RST} level. When V_{DD} rises again above $V_{\text{TP}},$ RSTB continues to drive low for at least 50 ms (t_{RPU}) to ensure a robust system reset at a reliable V_{DD} level. After t_{RPU} has been met, the RSTB pin will return to the weak high state. While RSTB is asserted, serial bus activity is locked out even if a transaction

occurred as V_{DD} dropped below V_{TP} . A memory operation started while V_{DD} is above V_{TP} will be completed internally.

Table 2 below shows how bits VTP(1:0) control the trip point of the low- V_{DD} reset. They are located in register 18h, bits 0 and 1. The reset pin will drive low when V_{DD} is below the selected V_{TP} voltage. Figure 20 illustrates the reset operation in response to a low V_{DD} .

Table 2.		
VTP Setting	VTP1	VTP0
2.6V	0	0
2.75V	0	1
2.9V	1	0
3.0V	1	1

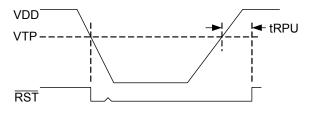
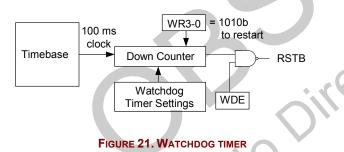


FIGURE 20. LOW VDD RESET

A watchdog timer can also be used to drive an active reset signal. The watchdog is a free-running programmable timer. The timeout period can be software programmed from 60 ms to 1.8 seconds in 60 ms increments via a 5-bit nonvolatile setting (register 0Ch).



The watchdog also incorporates a window timer feature that allows a delayed start. The starting time and ending time defines the window and each may be set independently. The starting time has 25 ms resolution and 0 ms to 775 ms range.

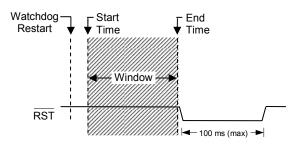


FIGURE 22. WINDOW TIMER

The watchdog EndTime value is located in register 0Ch, bits 4-0, the watchdog enable is bit 7. The watchdog is restarted by writing the pattern 1010b to the lower nibble of register 0Ah. Writing the correct pattern will also cause the timer to load new timeout values. Writing other patterns to this address will not affect its operation. Note the watchdog timer is free-running. Prior to enabling it, users should restart the timer as described above. This assures that the full timeout is provided immediately after enabling. The watchdog is disabled when V_{DD} drops below V_{TP}. Note setting the EndTime timeout setting to all zeroes (00000b) disables the timer to save power. The listing below summarizes the watchdog bits.

Watchdog StartTime	WDST4-0	0Bh, bits 4-0
Watchdog EndTime	WDET4-0	0Ch, bits 4-0
Watchdog Enable	WDE	0Ch, bit 7
Watchdog Restart	WR3-0	0Ah, bits 3-0
Watchdog Flags	EWDF,	09h, bit 7
	LWDF	09h, bit 6

The programmed StartTime value is a guaranteed maximum time while the EndTime value is a guaranteed minimum time, and both vary with temperature and V_{DD} voltage. The watchdog has two additional controls associated with its operation. The nonvolatile enable bit WDE allows the RSTB to go active if the watchdog reaches the timeout without being restarted. If a reset occurs, the timer will restart on the rising edge of the reset pulse. If WDE is not enabled, the watchdog timer still runs but has no effect on RSTB. The second control is a nibble that restarts the timer, thus preventing a reset. The timer should be restarted after changing the timeout value.

This procedure must be followed to properly load the watchdog registers:

		Address
1.	Write the StartTime value	0Bh

2. Write the EndTime value and WDE=1 0Ch

3. Issue a Restart command

The restart command in step 3 must be issued before t_{DOG2} , which was programmed in step 2. The window timer starts counting when the restart command is issued.

0Ah

Manual Reset

The RSTB is a bi-directional signal allowing the FM6124 to filter and de-bounce a manual reset switch. The RSTB input detects an external low condition and responds by driving the RSTB signal low for 100 ms (max.). This effectively filters and de-bounces a reset switch. After this timeout (t_{RPW}), the user may continue pulling down on the RSTB pin, but I²C commands will not be locked out.

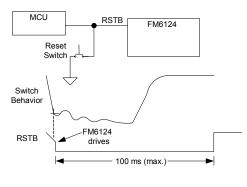


FIGURE 23. MANUAL RESET

Note the internal weak pull-up eliminates the need for additional external components.

Reset Flags

In case of a reset condition, a flag bit will be set to indicate the source of the reset. A low- V_{DD} reset is indicated by the POR bit, register 09h bit 5. There are two watchdog reset flags - one for an early fault (EWDF) and the other for a late fault (LWDF), located in register 09h bits 7 and 6. A manual reset will result in no flag being set, so the absence of a flag is a manual reset. Note that the bits are set in response to reset sources but they must be cleared by the user. It is possible to read the register and have both sources indicated if both have occurred since the user cleared them.

Power Fail Comparator

An analog comparator compares the PFI input pin to an onboard 1.5V reference. When the PFI input voltage drops below this threshold, the comparator will drive the PFO pin to a low state. The comparator has 100 mV of hysteresis (rising voltage only) to reduce noise sensitivity. The most common application of this comparator is to create an early warning power fail interrupt (NMI). This can be accomplished by connecting the PFI pin to an upstream power supply via a resistor divider. An application circuit is shown below. The comparator is a general purpose device and its application is not limited to the NMI function.

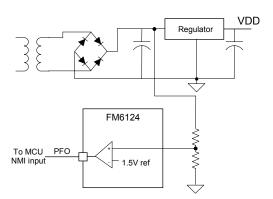


FIGURE 23. COMPARATOR AS POWER FAIL WARNING

If the power-fail comparator is not used, the PFI pin should be tied to either V_{DD} or V_{SS} . Note that the PFO output will drive to V_{DD} or V_{SS} as well.

Event Counter

The FM6124 offers the user a nonvolatile 16-bit event counter. The input pin CNT has a programmable edge detector. The CNT pin clocks the counter. The counter is located in registers 0E-0Fh. When the programmed edge polarity occurs, the counter will increment its count value. The register value is read by setting the RC bit (register 0Dh, bit 3) to 1. This takes a snapshot of the counter byte allowing a stable value even if a count occurs during the read. The register value can be written by first setting the WC bit (register 0Dh, bit 2) to 1. The user then may clear or preset the counter by writing to registers 0E-0Fh. Counts are blocked when the WC bit is set, so the user must clear the bit to allow counts.

The counter polarity control bit is CP, register 0Dh bit 0. When CP is 0, the counter increments on a falling edge of CNT, and when CP is set to 1, the counter increments on a rising edge of CNT. The polarity bit CP is nonvolatile.

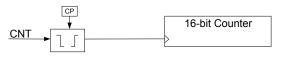


FIGURE 24. EVENT COUNTER

The counter <u>does not</u> wrap back to zero when it reaches the limit of 65,535 (FFFFh). Care must be taken prior to the rollover, and a subsequent counter reset operation must occur to continue counting.

There is also a control bit that allows the user to define the counter as nonvolatile or battery-backed. The counter is nonvolatile when the NVC bit (register 0Dh, bit 7) is logic 1 and battery-backed when the NVC bit is logic 0. Setting

the counter mode to battery-backed allows counter operation under V_{BAK} (as well as V_{DD}) power. The lowest operating voltage for battery-backed mode is 2.0V. When set to "nonvolatile" mode, the counter operates only when V_{DD} is applied and is above the V_{TP} voltage.

The event counter may be programmed to detect a tamper event, such as the system's case or access door being opened. A normally closed switch is tied to the CNT pin and the other contact to the case chassis, usually ground. The typical solution uses a pull-up resistor on the CNT pin and will continuously draw battery current. The FM6124 chip allows the user to invoke a polled mode, which occasionally samples the pin in order to minimize battery drain. It internally tries to pull the CNT pin up and if open circuit will be pulled up to a V_{IH} level, which will trip the edge detector and increment the event counter value. Setting the POLL bit (register 0Dh, bit 1) places the CNT pin into this mode. This mode allows the event counter to detect a rising edge tamper event but the user is restricted to operating in battery-backed mode (NVC=0) and using rising edge detection (CP=1). The CNT pin is polled once every 125ms. The additional average I_{BAK} current is less than 5nA. The polling timer circuit operates from the RTC, so the oscillator must be enabled for this to function properly.

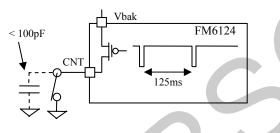


FIGURE 25. POLLED MODE ON CNT PIN DETECT TAMPER

In the polled mode, the internal pullup circuit can source a limited amount of current. The maximum capacitance (switch open circuit) allowed on the CNT pin is 100pF.

Serial Number

A memory location to write a 64-bit serial number is provided. It is a writeable nonvolatile memory block that can be locked by the user once the serial number is set. The 8 bytes of data and the lock bit are all accessed via unique commands for the RTC and Processor Companion registers. Therefore the serial number area is separate and distinct from the memory array. The serial number registers can be written an unlimited number of times, so these locations are general purpose memory. *However once the lock bit is set, the values cannot be altered and the lock cannot be removed.* Once locked the serial number registers can still be read by the system. The serial number is located in registers 10h to 17h. The lock bit is SNL, register 18h bit 7. Setting the SNL bit to a 1 disables writes to the serial number registers, and *the SNL bit cannot be cleared*.

Alarm

The alarm function compares user-programmed values to the corresponding time/date values and operates under V_{DD} or V_{BAK} power. When a match occurs, an alarm event occurs. The alarm drives an internal flag AF (register 00h, bit 6) and may drive the ACS pin, if desired, by setting the AL/SW bit (register 18h, bit 6) in the Companion Control register. The alarm condition is cleared by writing a '0' to the AF bit.

There are five alarm match fields. They are Month, Date, Hours, Minutes, and Seconds. Each of these fields also has a Match bit that is used to determine if the field is used in the alarm match logic. Setting the Match bit to '0' indicates that the corresponding field will be used in the match process.

Depending on the Match bits, the alarm can occur as specifically as one particular second on one day of the month, or as frequently as once per second continuously. The MSB of each Alarm register is a Match bit. Examples of the Match bit settings are shown in Table 4. Selecting none of the match bits (all '1's) indicates that no match is required. The alarm occurs every second. Setting the match select bit for seconds to '0' causes the logic to match the seconds alarm value to the current time of day. Since a match will occur for only one value per minute, the alarm occurs once per minute. Likewise setting the seconds and minutes match select bits causes an exact match of these values. Thus, an alarm will occur once per hour. Setting seconds, minutes, and hours causes a match once per day. Lastly, selecting all match-values causes an exact time and date match. Selecting other bit combinations will not produce meaningful results, however the alarm circuit will follow the functions described.

There are two ways a user can detect an alarm event, by reading the AF flag or monitoring the ACS pin. The interrupt pin on the host processor may be used to detect an alarm event. The AF flag in register 00h (bit 6) will indicate that a time/date match has occurred. The AF flag will be set to '1' when a match occurs. The AEN bit must be set to enable the AF flag on alarm matches. The flag and ACS pin will remain in this state until the AF bit is cleared by writing it to a '0'. Clearing the AEN bit will prevent further matches from setting AF but will not automatically clear the AF flag.

The RTC alarm is integrated into the special function registers and shares its output pin with the 512Hz

calibration and square wave outputs. When the RTC calibration mode is invoked by setting the CAL bit (register 00h, bit 2), the ACS output pin will be driven with a 512 Hz square wave and the alarm will continue to operate. Since most users only invoke the calibration mode during production this should have no impact on the otherwise normal operation of the alarm.

The ACS output may also be used to drive the system with a frequency other than 512 Hz. The AL/SW bit (register 18h, bit 6) must be '0'. A user-selectable frequency is provided by F0 and F1 (register 18h, bits 4 and 5). The other frequencies are 1, 4096, and 32768 Hz. If a

continuous frequency output is enabled with CAL mode, the alarm function will not be available.

Following is a summary table that shows the relationship between register control settings and the state of the ACS pin.

State	of Reg	Function of	
CAL	AEN	AL/SW	ACS pin
0	1	1	/Alarm
0	Х	0	Sq Wave out
1	Х	Х	512 Hz out
0	0	1	Hi-Z

Seconds	Minutes	Hours	Date	Months	Alarm condition
1	1	1	1	1	No match required = alarm 1/second
0	1	1	1	1	Alarm when seconds match = alarm 1/minute
0	0	1	1	1	Alarm when seconds, minutes match = alarm 1/hour
0	0	0	1	1	Alarm when seconds, minutes, hours match = alarm 1/date
0	0	0	0	1	Alarm when seconds, minutes, hours, date match = alarm 1/month

Table 4. Alarm Match Bit Examples

Real-time Clock Operation

The real-time clock (RTC) is a timekeeping device that can be capacitor- or battery-backed for permanently-powered operation. It offers a software calibration feature that allows high accuracy.

The RTC consists of an oscillator, clock divider, and a register system for user access. It divides down the 32.768 kHz time-base and provides a minimum resolution of seconds (1Hz). Static registers provide the user with read/write access to the time values. It includes registers for seconds, minutes, hours, dayof-the-week, date, months, and years. A block diagram shown in Figure 9 illustrates the RTC function.

The user registers are synchronized with the timekeeper core using R and W bits in register 00h. The R bit is used to read the time. Changing the R bit from 0 to 1 transfers timekeeping information from the core into the user registers 02-08h that can be

read by the user. If a timekeeper update is pending when R is set, then the core will be updated prior to loading the user registers. The user registers are frozen and will not be updated again until the R bit is cleared to a '0'.

The W bit is used to write new time/date values. Setting the W bit to a '1' stops the RTC and allows the timekeeping core to be written with new data. Clearing it to '0' causes the RTC to start running based on the new values loaded in the timekeeper core. The RTC may be synchronized to another clock source. On the 8^{th} clock of the write to register 00h (W=0), the RTC starts counting with a timebase that has been reset to zero milliseconds.

Note: Users should be certain not to load invalid values, such as FFh, to the timekeeping registers. Updates to the timekeeping core occur continuously except when locked.

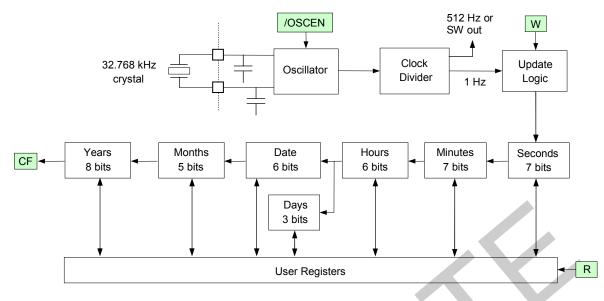


FIGURE 26. REAL TIME CLOCK CORE BLOCK DIAGRAM

Backup Power

The real-time clock and alarm are intended to be permanently powered. When the primary system power fails, the voltage on the VDD pin will drop. When the VDD voltage is less than V_{SW} , the RTC (and event counter) will switch to the backup power supply on VBAK. The clock operates at extremely low current in order to maximize battery or capacitor life. However, an advantage of combining a clock function with F-RAM memory is that data is not lost regardless of the backup power source.

Trickle Charger

To facilitate capacitor backup, the VBAK pin can optionally provide a trickle charge current. When the VBC bit (register 18h bit 3) is set to a '1', the V_{BAK} pin will source approximately 80 μ A until V_{BAK} reaches V_{DD} . This charges the capacitor to V_{DD} without an external diode and resistor charger. There is a Fast Charge mode which is enabled by the FC bit (register 18h, bit 2). In this mode the trickle charger current is set to approximately 1 mA, allowing a large backup capacitor to charge more quickly.

• In the case where no battery is used, the V_{BAK} pin should be tied to V_{SS} and VBC bit cleared.

^{*} Note: systems using lithium batteries should clear the VBC bit to 0 to prevent battery charging. The VBAK circuitry includes an internal 1 K Ω series resistor as a safety element.

Calibration

When the CAL bit in register 00h is set to a '1', the clock enters calibration mode. The FM6124 devices employ a digital method for calibrating the crystal oscillator frequency. The digital calibration scheme applies a digital correction to the RTC counters based on the calibration settings, CALS and CAL.4-0. In calibration mode (CAL=1), the ACS pin is driven with a 512 Hz (nominal) square wave and the alarm is temporarily unavailable. Any measured deviation from 512 Hz translates into a timekeeping error. The user measures the frequency and writes the appropriate correction value to the calibration register. The correction codes are listed in the table below. For convenience, the table also shows the frequency error in ppm. Positive ppm errors require a negative adjustment that removes pulses. Negative ppm errors require a positive correction that adds pulses. Positive ppm adjustments have the CALS (sign) bit set to 1, where as negative ppm adjustments have CALS = 0. After calibration, the clock will have a maximum error of ± 2.17 ppm or ± 0.09 minutes per month at the calibrated temperature.

The user will not be able to see the effect of the calibration setting on the 512 Hz output. The addition or subtraction of digital pulses occurs after the 512 Hz output.

The calibration setting is stored in F-RAM so it is not lost should the backup source fail. It is accessed with bits CAL.4-0 in register 01h. This value only can be written when the CAL bit is set to a 1. To exit the calibration mode, the user must clear the CAL bit to a ct Replaceme

logic 0. When the CAL bit is 0, the ACS pin will revert to the function according to Table 3.

Crystal Type

The crystal oscillator is designed to use a 12.5pF crystal without the need for external components, such as loading capacitors. The FM6124 device has built-in loading capacitors that match the crystal.

If a 32.768kHz crystal is not used, an external oscillator may be connected to the FM6124. Apply the oscillator to the X1 pin. Its high and low voltage levels can be driven rail-to-rail or amplitudes as low as approximately 500mV p-p. To ensure proper operation, a DC bias must be applied to the X2 pin. It should be centered between the high and low levels on the X1 pin. This can be accomplished with a voltage divider.

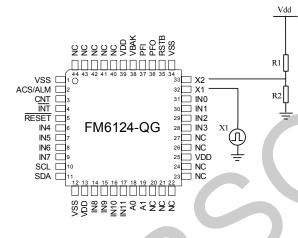


FIGURE 27. EXTERNAL OSCILLATOR

In the example, R1 and R2 are chosen such that the X2 voltage is centered around the oscillator drive levels. If you wish to avoid the DC current, you may choose to drive X1 with an external clock and X2 with an inverted clock using a CMOS inverter.

Layout Recommendations

The X1 and X2 crystal pins employ very high impedance circuits and the oscillator connected to these pins can be upset by noise or extra loading. To reduce RTC clock errors from signal switching noise, a guard ring should be placed around these pads and the guard ring grounded. High speed traces should be routed away from the X1/X2 pads. The X1 and X2 trace lengths should be less than 5 mm. The use of a ground plane on the backside or inner board layer is preferred. See layout example below.

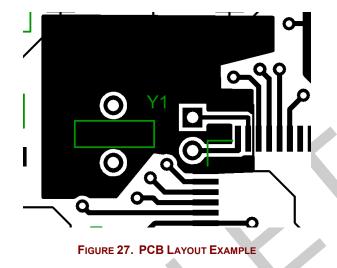


Table 5. Digital Calibration Adjustments

		ration Adjustmen		ion will achiev	e ± 2.17 PPM after calibration
	Measured Fre	quency Range	Error Range (I	PPM)	
	Min	Max	Min	Max	Program Calibration Register to:
0	512.0000	511.9989	0	2.17	000000
1	511.9989	511.9967	2.18	6.51	100001
2	511.9967	511.9944	6.52	10.85	100010
3	511.9944	511.9922	10.86	15.19	100011
4	511.9922	511.9900	15.20	19.53	100100
5	511.9900	511.9878	19.54	23.87	100101
6	511.9878	511.9856	23.88	28.21	100110
7	511.9856	511.9833	28.22	32.55	100111
8	511.9833	511.9811	32.56	36.89	101000
9	511.9811	511.9789	36.90	41.23	101001
10	511.9789	511.9767	41.24	45.57	101010
11	511.9767	511.9744	45.58	49.91	101011
12	511.9744	511.9722	49.92	54.25	101100
13	511.9722	511.9700	54.26	58.59	101101
14	511.9700	511.9678	58.60	62.93	101110
15	511.9678	511.9656	62.94	67.27	101111
16	511.9656	511.9633	67.28	71.61	110000
17	511.9633	511.9611	71.62	75.95	110001
18	511.9611	511.9589	75.96	80.29	110010
19	511.9589	511.9567	80.30	84.63	110011
20	511.9567	511.9544	84.64	88.97	110100
21	511.9544	511.9522	88.98	93.31	110101
22	511.9522	511.9500	93.32	97.65	110110
23	511.9500	511.9478	97.66	101.99	110111
24	511.9478	511.9456	102.00	106.33	111000
25	511.9456	511.9433	106.34	110.67	111001
26	511.9433	511.9411	110.68	115.01	111010
27	511.9411	511.9389	115.02	119.35	111011
28	511.9389	511.9367	119.36	123.69	111100
29	511.9367	511.9344	123.70	128.03	111101
30	511.9344	511.9322	128.04	132.37	111110
31	511.9322	511.9300	132.38	136.71	111111

	Negative	Calibration for fast	clocks: Calibra	tion will achiev	$ m ve \pm 2.17~PPM$ after calibration
	Measured Fre	quency Range	Error Ran	ge (PPM)	
	Min	Max	Min	Max	Program Calibration Register to:
0	512.0000	512.0011	0	2.17	000000
1	512.0011	512.0033	2.18	6.51	000001
2	512.0033	512.0056	6.52	10.85	000010
3	512.0056	512.0078	10.86	15.19	000011
4	512.0078	512.0100	15.20	19.53	000100
5	512.0100	512.0122	19.54	23.87	000101
6	512.0122	512.0144	23.88	28.21	000110
7	512.0144	512.0167	28.22	32.55	000111
8	512.0167	512.0189	32.56	36.89	001000
9	512.0189	512.0211	36.90	41.23	001001
10	512.0211	512.0233	41.24	45.57	001010
11	512.0233	512.0256	45.58	49.91	001011
12	512.0256	512.0278	49.92	54.25	001100
13	512.0278	512.0300	54.26	58.59	001101
14	512.0300	512.0322	58.60	62.93	001110
15	512.0322	512.0344	62.94	67.27	001111
16	512.0344	512.0367	67.28	71.61	010000
17	512.0367	512.0389	71.62	75.95	010001
18	512.0389	512.0411	75.96	80.29	010010
19	512.0411	512.0433	80.30	84.63	010011
20	512.0433	512.0456	84.64	88.97	010100
21	512.0456	512.0478	88.98	93.31	010101
22	512.0478	512.0500	93.32	97.65	010110
23	512.0500	512.0522	97.66	101.99	010111
24	512.0522	512.0544	102.00	106.33	011000
25	512.0544	512.0567	106.34	110.67	011001
26	512.0567	512.0589	110.68	115.01	011010
27	512.0589	512.0611	115.02	119.35	011011
28	512.0611	512.0633	119.36	123.69	011100
29	512.0633	512.0656	123.70	128.03	011101
30	512.0656	512.0678	128.04	132.37	011110
31	512.0678	512.0700	132.38	136.71	011111

Direct Replo

Detailed Register Description

The following table contains the description of the Event Data Recorder and MCU Companion I²C registers.

Address	Description	I										
0x33	Event Data											
	10s of Year Year											
	Contains the Event data Year in BCD format:											
	Upper Quartet contains 10s of Years											
	Lov	wer Quartet co	ontains unit of	Years								
0x32	Event Data BCD Month											
	D7	D6	D5	D4	D3	D2	D1	D0				
		10s of	Month			Ν	/Ionth					
	Contains the	Event data Mo	onth in BCD fo	ormat:								
	Up	per Quartet co	ntains 10s of 1	Month								
	Lov	wer Quartet co	ontains unit of	Month								
0x31	Event Data	BCD Date										
	D7	D6	D5	D4	D3	D2	D1	D 0				
		10s o	f Date				Date					
	Contains the		te in BCD for	mat:								
	Upper quartet contains 10s of Date											
	Lower Quartet contains unit of Date											
0x30	Event Data	of Week					-					
	D7	D6	D5	D4	D3	D2	D1	DO				
	Contains Eve	nt Data Day o	f week)				
0x2F	Event Data	BCD Hour	s			·						
	D 7	D6	D5	D4	D3	D2	D1	D0				
	10s of Hours											
	Contains the Event data hours in BCD format:											
	Upper quarters contains 10s of Hours											
	Lower Quartet contains unit of Hours											
0x2E	Event Data	BCD Minu	tes									
	D7	D6	D5	D4	D3	D2	D1	D0				
		10s of	Minutes			М	linutes					
	Contains the Event data Minutes in BCD format:											
	Upper quarters contains 10s of Minutes											
			ontains unit of	Minutes								
0x2D	Event Data	BCD Secon	ds 🖕	s O								
	D7	D6	D5	D4	D3	D2	D1	D0				
		10s of	Seconds			Se	econds					
	Contains the		conds in BCD	format:		5						
	Up	per quarters co	ontain 10s of s	econds								
	Lov	wer Quartet co	ontains unit of	seconds								

0x2C	Event Data	Code												
	D7	D6	D5	D4	D3	D2	D1	D0						
				Event	Code[7:0]									
Event Code	An Event coo	de is associate	d with Each re											
[7:0]	The table below shows the association between the Events and their corresponding Event Code.													
	Event		Event Cod	e	Event		Event Code	Event Code						
	EV_PINO_F		0x08		EV_PIN6_FA		0x14							
	EV_PINO_F		0x09		EV_PIN6_RI		0x15							
	EV_PIN1_F		0x0A		EV_PIN7_FA		0x16							
	EV_PINI_F EV PIN2 F		0x0B 0x0C		EV_PIN7_RI		0x17 0x18							
	EV_PIN2_F		0x0C 0x0D		EV_PIN8_FA		0x18 0x19							
	EV PIN3 F		0x0D 0x0E		EV PIN9 FA		0x13							
	EV PIN3 R		0x0E 0x0F		EV PIN9 RI		0x1B							
	EV PIN4 F		0x10		EV PIN10 F		0x1C							
	EV_PIN4_F	RISE	0x11		EV_PIN10_R	ISE	0x1D							
	EV_PIN5_F	PALL	0x12		EV_PIN11_F	'ALL	0x1E							
	EV_PIN5_F	RISE	0x13		EV_PIN11_R	ISE	0x1F							
0x2B	Unread Ev	ents Counte	r MSB											
	D7	D6	D5	D4	D3	D2	D1	D0						
0x2A	Unread Fw	Number of Unread Event Counter[15:8] Unread Events Counter LSB												
UX2A	DIT Cau EV	D6	D5	D4	D3	D2	D1	D0						
	D 7	D 0	-					DU						
					d Event Counter[7		D 00 D D 414							
# Unread					nt held in the FM	16124 Event	Buffer F-RAM	memory.						
Event Counter			the distance be		be performed be	foro roading	this register							
[15:0]	This is done	hy writing 0x(02 into the regi	ster located a	t address 27h	nore reading	uns register.							
0x29	Pin Pass-T					-	O							
	D7	D6	D5	D4	D3	D2	D1	DO						
	DII	DIO	DO	DO	107	D	DC	D4						
P11-P4	P11 Pood Onlyr	P10 This register o	P9	P8	P7 evel at the mome	P6	P5	P4						
0x28	Pin Pass-T		ontains the m	out pill logic id	ever at the monit	sit the shaps	not was periorn	ieu						
0120	D7	D6	D5	D4	D3	D2	D1	D0						
	<u> </u>			D4	03	D2		100						
			erved		P3	P2	P1	PO						
P3-P0				out pin logic le	evel at the mome	ent the snaps	hot was perform	ned						
0x27		#Events Sn												
	D7	D6	D5	D4	D3	D2	D1	DO						
	Reserved NBEV SNAP													
NBEV	Write Only: Writing a 1 into the NBEVP will perform a snapshot read of the internal unread Events Counter													
	register and write the corresponding input logic level into the Register 2Ah and 2Bh Write Only: Writing a 1 into the SNAP will perform a snapshot read of all 12 Input of the FM6124 and write the													
SNAP						f all 12 Input	t of the FM6124	and write the						
0.0(•	<u> </u>	evel into the R	legister 28h ai	nd 29h									
0x26	Pin Event I					54	DI	DA						
	D7	D6	D5	D4	D3	D2	D1	DO						
	P11	P10	Р9	P8	P7	P6	P5	P4						
P11-P4	Pin Event En													
	0: Event detection on the corresponding input pin is disabled													
			orresponding i	nput pin is Ac	tivated									
0x25	Pin Event I													
	D7	D6	D5	D4	D3	D2	D1	DO						
0420	-													
0.4-0	J	Res	erved		Р3									
P3 – P0	Pin Event En		erved		P3	P2	P1	PO						
	Pin Event En 0: Event dete	able ection on the c	erved orresponding i orresponding i		sabled	P2	P1	PO						

0x24	Pin Event Rise/Fall A											
	D7	D6	D5	D4	D3	D2	D1	D0				
	P11	P10	Р9	P8	Р7	P6	Р5	P4				
	Pin Event trig											
						nding Input pin						
0-22			a Low to High	transition on t	he correspor	nding Input pin						
0x23	Pin Event Rise/Fall A D7 D6 D5 D4 D3 D2 D1 D0											
	D 7			D4			D1	D0				
D2 D0	D . D . (1)		erved		P3	P2	P1	PO				
P3 – P0	Pin Event trig	gering condit	10n High to Low	transition on t	he correspor	nding Input pin						
						iding Input pin						
0x22	Pin Event I	<u> </u>										
	D7	D6	D5	D4	D3	D2	D1	D0				
	P11	P10	Р9	P8	P7	P6	P5	P4				
P11-P4	Pin Event Inte			10	1,	10	15	17				
	0: the interrup	ot pin will not	be activated w			ted on the correst		pin				
			vated (Low) w	hen a Event is	detected on	the corresponding	ng Input pin					
0x21	Pin Event in											
	D7	D6	D5	D4	D3	D2	D1	D0				
	CLEAR	BF	B75F	B50F	P3	P2	P1	PO				
CLEAR		his Bit will de	eactivate the IN	NT pin, when a	ctivated by a	a Pin Event inter	rupt	X				
BF	Buffer Full:	1.1.1.1.1		DIT (1			11 1141 1.					
B75F	Buffer 75% F		n will make th	e IN I to beco	ne Active th	en the Buffer Fu	III condition is	met				
D/JF			n will make th	e INT to beco	ne Active th	en the Buffer ge	ts 75% Full					
B50F	Buffer 50% F					en une Burier ge						
				e INT to becom	me Active th	en the Buffer ge	ts 50% Full					
P3 – P0	Pin Event Inte					C						
	0: the interrupt pin will not be activated when an Event will be detected on the corresponding input pin 1: The INT pin will be activated (Low) when a Event is detected on the corresponding Input pin											
0x20	Event Reco			nen a Event is	detected on	the corresponding	ig input pin					
0.120	D7	D6	D5	D4	D3	D2	D1	D0				
	I							D0				
EBUFSIZE	EBUFSI		ERR	READIR	DUESIZE 4	EDRC to a new value	MD[3:0]	want Duffar				
[1:0]												
[1.0]	Important: The user should disable Event Recording while resetting the Event Buffer. Setting the EBUESIZE bits to a different value will reset all the Event Buffer											
	LDOI SIZE	EBUFSIZE bits to a different value will reset all the Event Buffer.										
		EBU	FSIZE Max	x number of	Events	User F-RAM	size					
		00	400			0						
		01	300			8 KBytes						
		10	200			16 Kbytes						
		10				24 KBytes						
EDD					.1 1	5		D (C 1				
ERR					they have	reach the end	of the Event	Butter when				
DID			bit has no eff			(DD)		d (in at 1 4				
DIR						er (RP) will be	e decremente	1 (instead of				
	incremented) when an e	vent has been	read out of t	ne Event B	ullel.						

0x1D	Alarm – M	onth									
	D7	D6	D5	D4	D3	D2	D1	DO			
	M	0	0	10 Month	Month.3	Month.2	Month.1	Month.0			
		alarm value fo	r the month an	d the mask bit	to select or des	select the Mor	th value.				
/M							c. Setting this	bit to 1			
			gnore the Mor	nth value. Batte	ery-backed, rea	d/write.					
0x1C	Alarm – Da										
	D7	D6	D5	D4	D3	D2	D1	DO			
	М	0	10 date.1	10 date.0	Date.3	Date.2	Date.1	Date.0			
(a. c.				the mask bit to							
/M	Match: Setting this bit to 0 causes the Date value to be used in the alarm match logic. Setting this bit to 1 causes the match circuit to ignore the Date value. Battery-backed, read/write.										
0x1B	Alarm – He		ne Date value.	Вашегу-васке	d, read/write.						
UXID	D7	D6	D5	D4	D3	D2	D1	D0			
	_	-									
	M Contains the	0	10 hours.1	10 hours.0	Hours.3	Hours2	Hours.1	Hours.0			
/M				the mask bit t			s value. c. Setting this b	it to 1 causes			
/ 1 v1				e. Battery-back			c. Setting this t	in to 1 causes			
0x1A	Alarm – M		ne mours (uru	e. Butterj cuer							
•	D7	D6	D5	D4	D3	D2	D1	D0			
	M	10 min.2	10 min.1	10 min.0	Min.3	Min.2	Min.1	Min.0			
				and the mask b			-	Iviiii.0			
/M							gic. Setting thi	s bit to 1			
-				utes value. Bat							
0x19	Alarm – Se	conds						1			
	D7	D6	D5	D4	D3	D2	D 1	D0			
	M	10 sec.2	10 sec.1	10 sec.0	Seconds.3	Seconds.2	Seconds.1	Seconds.0			
·	Contains the			and the mask b							
/M							gic. Setting thi	s bit to 1			
			gnore the Seco	onds value. Ba	tery-backed, re	ead/write.					
0x18	Companion Control										
UAIO			D.#	E 1		D.A.	D.1	DO			
	D7	D6	D5	D4	D3	D2	D1	DO			
	D7 SNL	D6 AL/SW	F1	F0	VBC	FC	VTP1	VTP0			
SNL	D7 SNL Serial Number	D6 AL/SW er Lock: Settin	F1 g to a '1' make	F0	VBC	FC		VTP0			
SNL	D7 SNL Serial Numbo set to '1'. No	D6 AL/SW er Lock: Settin onvolatile, read	F1 g to a '1' make	F0 es registers 101	VBC to 17h and SN	FC NL read-only.	VTP1 SNL cannot be	VTP0 cleared once			
	D7 SNL Serial Numbo set to '1'. No Alarm/Squar	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select:	F1 g to a '1' make l/write.	F0 es registers 101 1', the alarm n	VBC to 17h and SM natch drives the	FC NL read-only. e ACS pin as v	VTP1 SNL cannot be vell as the AF t	VTP0 cleared once flag. When			
SNL	D7 SNL Serial Numberset to '1'. No Alarm/Squar- set to '0', the	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select:	F1 g to a '1' make l/write. : When set to ' re Wave Freq	F0 es registers 10 1', the alarm n will be driven	VBC to 17h and SM natch drives the	FC NL read-only. e ACS pin as v	VTP1 SNL cannot be	VTP0 cleared once flag. When			
SNL	D7 SNL Serial Numberset to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa atile, read/write Freq Select: 1	F1 g to a '1' make l/write. When set to ' re Wave Freq e.	F0 es registers 10l 1', the alarm n will be driven	VBC to 17h and SM patch drives the on the ACS pir	FC NL read-only. e ACS pin as v h, and an alarn	VTP1 SNL cannot be vell as the AF t	VTP0 cleared once flag. When ets the AF			
SNL AL/SW	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa atile, read/write Freq Select: T ivolatile.	F1 g to a '1' make l/write. When set to ' re Wave Freq e. These bits select	F0 es registers 101 1', the alarm n will be driven et the frequenc	VBC to 17h and SN natch drives the on the ACS pir	FC NL read-only. e ACS pin as v and an alarn bin when the C	VTP1 SNL cannot be well as the AF to match only se	VTP0 cleared once flag. When ets the AF			
SNL AL/SW	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa ttile, read/write Freq Select: T ivolatile. tting F(F1 g to a '1' make l/write. When set to ' re Wave Freq e. These bits select (1:0)	F0 es registers 101 1', the alarm n will be driven et the frequency Set	VBC to 17h and SM natch drives the on the ACS pir y on the ACS p ting F(FC NL read-only. e ACS pin as v a, and an alarn bin when the C (1:0)	VTP1 SNL cannot be well as the AF to match only se	VTP0 cleared once flag. When tts the AF			
SNL AL/SW	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor <u>Se</u>	D6 AL/SW er Lock: Settin nvolatile, reac e Wave Select selected Squa atile, read/write Freq Select: T ivolatile. tting F(1 Hz 0	F1 g to a '1' make l/write. When set to ' re Wave Freq e. These bits select (1:0) 00 (default)	F0 es registers 101 1', the alarm n will be driven of ct the frequency <u>Set</u> 409	VBC a to 17h and SM match drives the on the ACS pir y on the ACS p ting F(06 Hz 1	FC NL read-only. e ACS pin as v and an alarm bin when the C (1:0) 0	VTP1 SNL cannot be well as the AF to match only se	VTP0 cleared once flag. When tts the AF			
SNL AL/SW	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor <u>Se</u>	D6 AL/SW er Lock: Settin nvolatile, reac e Wave Select selected Squa atile, read/write Freq Select: T ivolatile. tting F(1 Hz 0	F1 g to a '1' make l/write. When set to ' re Wave Freq e. These bits select (1:0)	F0 es registers 101 1', the alarm n will be driven of ct the frequency <u>Set</u> 409	VBC a to 17h and SM match drives the on the ACS pir y on the ACS p ting F(06 Hz 1	FC NL read-only. e ACS pin as v a, and an alarn bin when the C (1:0)	VTP1 SNL cannot be well as the AF to match only se	VTP0 cleared once flag. When ets the AF			
SNL AL/SW F(1:0)	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor Se	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa atile, read/write Freq Select: T ivolatile. tting F(1 Hz 0 512 Hz 0	F1 g to a '1' make l/write. When set to ' re Wave Freq e. These bits select (1:0) 00 (default)	F0 es registers 101 1', the alarm n will be driven et the frequency Set 409 327	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1	FC NL read-only. e ACS pin as v and an alarn bin when the C 1:0) 0 1	VTP1 SNL cannot be well as the AF to match only se	VTP0 cleared once clag. When tts the AF W bits are			
SNL AL/SW F(1:0) VBC	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor Se VBAK Charg be supplied of	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa attile, read/write Freq Select: T ivolatile. tting F(1 Hz 0 512 Hz 0 ger Control: Se n V _{BAK} . Cleari	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 etting VBC to '0'	F0 es registers 101 1', the alarm n will be driven et the frequency Set 409 327 1' (and FC=0) 2 disables the c	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS r ting F(26 Hz 1 causes a 80 μ harge current.	FC NL read-only. e ACS pin as v and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backe	VTP1 SNL cannot be vell as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write.	VTP0 cleared once clag. When tts the AF W bits are rge current to			
SNL AL/SW F(1:0) VBC	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor <u>Se</u> VBAK Charg be supplied of Fast Charge:	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa tfile, read/write Freq Select: T ivolatile. tting F(1 Hz 0 512 Hz 0 ger Control: Se n V _{BAK} . Cleari Setting FC to	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 etting VBC to '0' '1' (and VBC)	F0 es registers 101 1', the alarm n will be driven et the frequency Set 409 327 1' (and FC=0) 2 disables the c =1) causes a ~	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 µz 1 harge current. 1 mA trickle ch 1	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed arge current t	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha	VTP0 cleared once clag. When tts the AF W bits are rge current to			
SNL AL/SW F(1:0) VBC FC	D7 SNL Serial Numb set to '1'. No Alarm/Squar set to '0', the flag. Nonvola Square Wave both '0'. Nor <u>Se</u> VBAK Charg be supplied of Fast Charge: Clearing VB	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa tfile, read/write Freq Select: T ivolatile. tting F(1 Hz 0 512 Hz 0 ger Control: Se n V _{BAK} . Cleari Setting FC to C to '0' disable	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 etting VBC to '0' '1' (and VBC es the charge c	F0 es registers 101 1', the alarm n will be driven et the frequency Set 409 327 1' (and FC=0) 2 disables the c =1) causes a ~ urrent. Battery	VBC a to 17h and SP natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 768Hz 1 causes a 80 μz harge current. mA trickle ch -backed, read/w	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t write.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied o	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			
SNL AL/SW F(1:0) VBC	D7 SNL Serial Numb set to '1'. No Alarm/Squar- set to '0', the flag. Nonvola Square Wave both '0'. Nor Se VBAK Charg be supplied of Fast Charge: Clearing VB	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa tfile, read/write Freq Select: T ivolatile. tting F(1 Hz C 512 Hz C 512 Hz C ger Control: Se n V _{BAK} . Cleari Setting FC to C to '0' disable These bits cont	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 etting VBC to '0' '1' (and VBC es the charge c rol the reset tr	F0 es registers 101 1', the alarm n will be driven the frequency Set 409 327 1' (and FC=0) 2 disables the c =1) causes a ~ urrent. Battery ip point for the	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 μL harge current. ImA trickle ch -backed, read/v low-V _{DD} reset	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t vrite.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied of nen V _{DD} is belo	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			
SNL AL/SW F(1:0) VBC FC	D7 SNL Serial Numbb set to '1'. No Alarm/Square set to '0', the flag. Nonvola Square Wave both '0'. Nor See VBAK Charge be supplied of Fast Charge: Clearing VBC VTP Select. ' selected V _{TP}	D6 AL/SW er Lock: Settin onvolatile, reac e Wave Select: selected Squa attile, read/write Freq Select: Freq Select: tivolatile. tting freq Select: twolatile. tting freq Select: twolatide. twolatide.	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 stting VBC to '0' '1' (and VBC es the charge c rol the reset tr set pin RSTB v	F0 es registers 101 1', the alarm n will be driven et the frequency Set 409 327 1' (and FC=0) 2 disables the c =1) causes a ~ urrent. Battery	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 μL harge current. ImA trickle ch -backed, read/v low-V _{DD} reset	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t vrite.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied of nen V _{DD} is belo	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			
SNL AL/SW F(1:0) VBC FC	D7 SNL Serial Numbb set to '1'. No Alarm/Square set to '0', the flag. Nonvola Square Wave both '0'. Nor Second VBAK Charge be supplied of Fast Charge: Clearing VB VTP Select. ' selected V _{TP} Set	D6 AL/SW er Lock: Settin nvolatile, reac e Wave Select: selected Squa attile, read/write Freq Select: roward Atting F(1) Volatile. tting F(1) 512 Hz C 512 Hz C ger Control: Set Setting FC to C to '0' disable Chese bits control voltage, the rest thing VTP VTP	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 Stting VBC to '0' '1' (and VBC es the charge c rol the reset tr set pin RSTB v (1:0)	F0 es registers 101 1', the alarm n will be driven of ct the frequency Set 409 327 1' (and FC=0) 2' disables the c =1) causes a ~ urrent. Battery ip point for the will drive low f	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 μL harge current. ImA trickle ch -backed, read/v low-V _{DD} reset	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t vrite.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied of nen V _{DD} is belo	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			
SNL AL/SW F(1:0) VBC FC	D7 SNL Serial Number set to '1'. Not Alarm/Squar- set to '0', the flag. Nonvola Square Waver both '0'. Nor <u>See</u> VBAK Charge VBAK Charge: Clearing VBO VTP Select. ' selected V _{TP} <u>See</u> 2.6	D6AL/SWer Lock: Settinonvolatile, reade Wave Select:selected Squaatile, read/writeFreq Select: 1twolatile.tting F(1 HzColspan="2">Colspan="2"Colspan="2">Colspan="2">Colspan="2"Colspan="2">Colspan="2" <tr <="" <td="" colspan="2" td="">Colspa</tr>	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 stting VBC to '0' '1' (and VBC es the charge c rol the reset tr set pin RSTB v	F0 es registers 101 1', the alarm n will be driven of ct the frequency Set 409 327 1' (and FC=0) 2' disables the c =1) causes a ~ urrent. Battery ip point for the will drive low f	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 μL harge current. ImA trickle ch -backed, read/v low-V _{DD} reset	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t vrite.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied of nen V _{DD} is belo	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			
SNL AL/SW F(1:0) VBC FC	D7 SNL Serial Number set to '1'. Not Alarm/Squar- set to '0', the flag. Nonvola Square Waver both '0'. Nor <u>See</u> VBAK Charge VBAK Charge: Clearing VBO VTP Select. ' selected V _{TP} <u>See</u> 2.6	D6AL/SWer Lock: Settinonvolatile, reade Wave Select:selected Squaatile, read/writeFreq Select: 1tivolatile.tting F(1 Hz C512 Hz CSetting FC to C to '0' disableThese bits control: Se m VBAK. Cleari Setting FC to C to '0' disableThese bits control: Se to C to '0' disableThese bits control: Se to C to '0' disableThese bits control: Se to C to '0' disableThese bits controlVTPOV 00Setting VTPOV 00'5V 01	F1 g to a '1' make l/write. When set to ' re Wave Freq ' e. These bits select (1:0) 00 (default) 01 Stting VBC to '0' '1' (and VBC es the charge c rol the reset tr set pin RSTB v (1:0)	F0 es registers 101 1', the alarm n will be driven of ct the frequency Set 409 327 1' (and FC=0) 2' disables the c =1) causes a ~ urrent. Battery ip point for the will drive low f	VBC a to 17h and SN natch drives the on the ACS pir y on the ACS p ting F(26 Hz 1 268Hz 1 causes a 80 μL harge current. ImA trickle ch -backed, read/v low-V _{DD} reset	FC NL read-only. e ACS pin as v a, and an alarn bin when the C 1:0) 0 1 A (1 mA if FC Battery-backed barge current t vrite.	VTP1 SNL cannot be well as the AF to match only se CAL and AL/SV =1) trickle cha d, read/write. o be supplied of nen V _{DD} is belo	VTP0 cleared once clag. When tts the AF W bits are rge current to n V _{BAK} .			

0x17	Serial Num	iber Byte 7										
UAI /	D7	D6	D5	D4	D3	D2	D1	D0				
	SN.63	SN.62	SN.61	SN.60	SN.59	SN.58	SN.57	SN.56				
0x16	Serial Num		511.01	311.00	511.39	511.30	511.57	511.50				
0410	D7	D6	D5	D4	D3	D2	D1	D0				
		-					SN.49					
0x15	SN.55 Serial Num	SN.54	SN.53	SN.52	SN.51	SN.50	511.49	SN.48				
UAIS	D7	D6	D5	D4	D3	D2	D1	D0				
	SN.47	SN.46	SN.45	SN.44			SN.41	SN.40				
0x14	Serial Num		5IN.43	5IN.44	SN.43	SN.42	511.41	511.40				
UAIT	D7	D6	D5	D4	D3	D2	D1	D0				
		-						-				
0x13	SN.39 Serial Num	SN.38	SN.37	SN.36	SN.35	SN.34	SN.33	SN.32				
UXIJ	D7	D6	D5	D4	D3	D2	D1	D0				
						Î.						
0x12	SN.31 Serial Num	SN.30	SN.29	SN.28	SN.27	SN.26	SN.25	SN.24				
0X12	D7	Der Byte 2 D6	D5	D4	D3	D2	D1	D0				
0x11	SN.23	SN.22	SN.21	SN.20	SN.19	SN.18	SN.17	SN.16				
UXII	Serial Num D7	Der Byte I	D5	D4	D3	D2	D1	DO				
		-										
010	SN.15 Serial Num	SN.14	SN.13	SN.12	SN.11	SN.10	SN.9	SN.8				
0x10	D7	Der Byte U	D5	D4	D3	D2	D1	D0				
		-										
	SN.7	SN.6 mber bytes are	SN.5	SN.4	SN.3	SN.2	SN.1	SN.0				
	All sellar hui	inder bytes are	read/write wite	en SNL-0, lea	d-only when s	SNL-1. NOIIVO	latile.					
0x0F	Event Cou	nter Byte 1				- Ci						
	D7	D6	D5	D4	D3	D2	D1	D0				
	EC.15	EC.14	EC.13	EC.12	EC.11	EC.10	EC.9	EC.8				
							atile when NV					
	Battery-back	ed when NVC			20							
0x0E		nter Byte 0		•								
	D7	D6	D5	D4	D3	D2	D1	D0				
	EC.7	EC.6	EC.5	EC.4	EC.3	EC.2	EC.1	EC.0				
		Event Counter Byte 0. Increments on programmed edge event on CNT input. Nonvolatile when NVC=1, Battery-backed when NVC=0, read/write.										
	Battery-back		=0, read/write.									
0x0D	Event Cou	nter Control										
	D7	D6	D5	D4	D3	D2	D1	D0				
	NVC	_		_	RC	WC	POLL	СР				
NVC		Volatile Count	er: Setting this	bit to 1 makes								
	V _{DD} is greate	Nonvolatile/Volatile Counter: Setting this bit to 1 makes the counter nonvolatile and counter operates only when V_{DD} is greater than V_{TP} . Setting this bit to 0 makes the counter volatile, which allows counter operation under										
2.0	V _{BAK} or V _{DD}	power. Nonvo	latile, read/wri	ite.								
RC		er. Setting this l ut missing cou					the system to r	ead the				
WC							e WC=1, the co	ounter is				
							o activate the c					
POLL	Polled Mode	: When POLL	=1, the CNT p	in is sampled	for 30µs every	125ms. If POI	LL is set, the N	VC bit is				
							st be enabled (/					
CD							tile, read/write					
СР	The CNT pir	i detects falling	g eages when C	r = 0, rising e	euges when CI	r = 1. Nonvola	tile, read/write	•				

0x0C	Watchdo	g Control							
	D7	D6	D5	D4]	D3	D2	D1	D0
	WDE	-	-	WDET4	WI	DET3	WDET2	WDET1	WDET0
WDE								al to go active. V	When WDE
			s no effect on th						
WDET(4:0)	Watchdog	EndTime: Set	s the ending tin	ne for the watch	idog w	indow tii	ner with 60 m	s (min.) resolut	ion. The
								() to be set. New	
			me may be set t					3:0). To save po	ower (disable
	timer en eu	it), the End in	the may be set t	0 ull 201003. 1 (JIIVOIUL	ne, reau	write.		
	7	Watchdog Er	ndTime	WDET4 W	VDET3	WDET	2 WDET1 V	<u>WDET0</u>	
	Ι	Disables Tim	er	0	0	0	0	0	
		(min.)	(max.)						
		60 ms	200 ms	0	0	0	0	1	
		120 ms	400 ms	0	0	0		0	
		180 ms	600 ms	0	0	0	1	1	
		:	:						
		1200 ms	4000 ms	1	0	1	0	0	
		1260 ms	4200 ms	1	ů	1	ů	ı 1	
		1320 ms	4400 ms	1	0	1	1	0	
		•	•						
		1740		1	-	1	0	1	
		1740 ms 1800 ms	5800 ms 6000 ms	1	1	1	0	0	\sim
		1800 ms	6200 ms	1	1	1	1	1	()
0x0B		g Control	0200 1115	1	1	1	ſ	1	
UXUD	D7	D6	D5	D4		D3	D2	D1	D0
		20	20						
WDST(4:0)	- Watchdog	- StartTima Sa	- ts the starting ti	WDST4		DST3	WDST2	WDST1 ms (max.) resolu	WDST0
WD31(4.0)								y) to be set. New	
								WR(3:0). Nonvo	
	read/write.								
		Vatchdog St			VDS13	WDST		VDST0	
	C C	ms (default		0	0	0	0	0	
		(min.) 7.5 ms	(max.) 25 ms	0	0	0	0	1	
		15.0 ms	50 ms		0	0		0	
		22.5 ms	75 ms	0	0	0	1	1	
		•		U	U	U	1	1	
		:							
		: 150 ms	: 500 ms	1	0	1		0	
	1	57.5 ms	525 ms	1	0	1	0	1	
				1 1 1			0		
		57.5 ms	525 ms	1 1 1	0	1	0	1	
		57.5 ms	525 ms	1 1 1	0	1	0	1	
		157.5 ms 165 ms 217.5 ms 225 ms	525 ms 550 ms	1 1 1 1	0	1	0 1 0	1 0	
		157.5 ms 165 ms 217.5 ms 225 ms 232.5 ms	525 ms 550 ms 725 ms	1 1 1 1 1 1	0 0 1	1 1 1	0 1 0	1 0 1	
0x0A		157.5 ms 165 ms 217.5 ms 225 ms 232.5 ms g Restart	525 ms 550 ms 725 ms 750 ms 775 ms	1	0 0 1 1 1	1 1 1 1 1	0 1 0 1 1	1 0 1 0 1	
0x0A		157.5 ms 165 ms 217.5 ms 225 ms 232.5 ms	525 ms 550 ms 725 ms 750 ms	1 1 1 1 1 1 D4	0 0 1 1 1	1 1 1	0 1 0	1 0 1	D0
0x0A	2 Watchdo	157.5 ms 165 ms 217.5 ms 225 ms 232.5 ms g Restart	525 ms 550 ms 725 ms 750 ms 775 ms	1	0 0 1 1 1	1 1 1 1 1	0 1 0 1 1	1 0 1 0 1	D0 WR0
0x0A WR(3:0)	Watchdog D7 - Watchdog	217.5 ms 165 ms 217.5 ms 225 ms 232.5 ms g Restart D6 - Restart. Writi	525 ms 550 ms 725 ms 750 ms 775 ms D5	1 	0 0 1 1 1 1 0 v 0) resta	1 1 1 1 D3 VR3 rts the w	0 1 0 1 1 1 D2 wR2 vatchdog time:	1 0 1 0 1 D1 wR1 r. The upper nil	WR0 oble contents
	Watchdog D7 - Watchdog	257.5 ms 165 ms 217.5 ms 225 ms 232.5 ms g Restart D6 - Restart. Writt ect this opera	525 ms 550 ms 725 ms 750 ms 775 ms D5	1 	0 0 1 1 1 1 0 v 0) resta	1 1 1 1 D3 VR3 rts the w	0 1 0 1 1 1 D2 wR2 vatchdog time:	1 0 1 0 1 D1 WR1	WR0 oble contents

0x09	Watchdo	g Flags								
	D7	D6	D5	D4	D3	D2	D1	DO		
	EWDF	LWDF	POR	LB	-	-	-	-		
EWDF	Early Wate	chdog Timer F	ault Flag: Whe	en a watchdog r	estart occurs to	o early (before	the programmed	l watchdog		
	StartTime)	Early Watchdog Timer Fault Flag: When a watchdog restart occurs too early (before the programmed watchdog StartTime), the RSTB pin is driven low and this flag is set. It must be cleared by the user. Note that both EWDF								
			both reset sourc	es have occurre	ed since the flag	gs were cleared	by the user. Ba	ttery-		
	backed, rea									
LWDF							er the programm			
							set. It must be c			
					f both reset sou	rces have occu	rred since the fla	igs were		
DOD			ery-backed, rea							
POR							be set to 1. A m			
							uld be set if both			
			nce the flags we	ere cleared by the	ne user. Battery	-backed, read/v	write. (internally	set, user		
I D	must clear			4 14 1			TC/alarra and an			
LB		up: If the v_{BA}	$_{\rm K}$ source drops	to a voltage lev	to he re initiali	o operate the R	TC/alarm when attery-backed re	ristor		
	V DD V BAK	, this on which uld be treated	as unknown Th	a user should (lo be le-initiali	zeu since the b	allery-backed le	gister		
			et, user must cle		leaf it to 0 wite	in mittanzing u	ie system. Datte	i y-backeu.		
0x08		ping – Years	•	ar onj.						
0400	D7	D6	, D5	D4	D3	D2	D1	D0		
		-								
	10 year.3	10 year.2	10 year.1	10 year.0	Year.3	Year.2	Year.1	Year.0		
							ears; upper nibb			
			s. Each nibble o	perates from 0	to 9. The range	for the register	is 0-99. Battery	-backed,		
	read/write.									
0x07	-	oing – Mont								
	D7	D6	D5	D4	D3	D2	D1	D0		
	0	0	0	10 Month	Month.3	Month 2	Month 1	Month.0		
	0 Contains th	÷	0 for the month.	10 Month Lower nibble c	Month.3 ontains the low	Month.2 er digit and ope	Month.1 erates from 0 to	Month.0 9; upper		
	Contains th	ne BCD digits	for the month.	Lower nibble c	ontains the low	er digit and ope	erates from 0 to	9; upper		
	Contains th	ne BCD digits e bit) contains	for the month.	Lower nibble c	ontains the low	er digit and ope		9; upper		
<u>0x06</u>	Contains the nibble (one backed, rea	ne BCD digits e bit) contains ad/write.	for the month.	Lower nibble c	ontains the low	er digit and ope	erates from 0 to	9; upper		
0x06	Contains the nibble (one backed, rea	ne BCD digits e bit) contains ad/write.	for the month. the upper digit	Lower nibble c	ontains the low	er digit and ope	erates from 0 to	9; upper		
0x06	Contains th nibble (one backed, rea Timekeep D7	ne BCD digits e bit) contains ad/write. ping – Date D6	for the month. the upper digit of the month D5	Lower nibble c and operates fr D4	ontains the low om 0 to 1. The D3	er digit and operange for the re	erates from 0 to gister is 1-12. B D1	9; upper attery- D0		
0x06	Contains th nibble (one backed, rea Timekeep D7 0	ne BCD digits e bit) contains ad/write. ping – Date D6	for the month. the upper digit of the month D5 10 date.1	Lower nibble c and operates fr D4 10 date.0	ontains the low om 0 to 1. The D3 Date.3	er digit and op range for the re D2 Date.2	erates from 0 to gister is 1-12. B D1 Date.1	9; upper attery- D0 Date.0		
0x06	Contains th nibble (one backed, rea Timekeep D7 0 Contains th	he BCD digits e bit) contains ad/write. ping – Date D6 0 he BCD digits	for the month. the upper digit of the month D5 10 date.1 for the date of	Lower nibble c and operates fr D4 10 date.0 the month. Low	ontains the low om 0 to 1. The D3 Date.3 ver nibble conta	er digit and op range for the re D2 Date.2 ins the lower d	erates from 0 to gister is 1-12. B D1 Date.1 igit and operates	9; upper attery- D0 Date.0 3 from 0 to		
0x06	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni	he BCD digits e bit) contains ad/write. ping – Date D6 0 he BCD digits bble contains	for the month. the upper digit of the month D5 10 date.1 for the date of	Lower nibble c and operates fr D4 10 date.0 the month. Low	ontains the low om 0 to 1. The D3 Date.3 ver nibble conta	er digit and op range for the re D2 Date.2 ins the lower d	erates from 0 to gister is 1-12. B D1 Date.1	9; upper attery- D0 Date.0 3 from 0 to		
	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea	ne BCD digits e bit) contains ad/write. Ding – Date D6 0 ne BCD digits bble contains ad/write.	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit	Lower nibble c and operates fr D4 10 date.0 the month. Low	ontains the low om 0 to 1. The D3 Date.3 ver nibble conta	er digit and op range for the re D2 Date.2 ins the lower d	erates from 0 to gister is 1-12. B D1 Date.1 igit and operates	9; upper attery- D0 Date.0 3 from 0 to		
	Contains th nibble (one backed, rea Timekeeg D7 0 Contains th 9; upper ni backed, rea Timekeeg	ne BCD digits e bit) contains ad/write. ping – Date D6 0 ne BCD digits bble contains ad/write. ping – Day o	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro	Date.3 Date 3 or 0 to 3. The	er digit and op range for the re D2 Date.2 ins the lower d range for the re	D1 Date.1 Digister is 1-12. B Date.1 Date.1 Date.1 Date.1 Date.1 Date.1	9; upper attery- D0 Date.0 5 from 0 to attery-		
	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea	ne BCD digits e bit) contains ad/write. Ding – Date D6 0 ne BCD digits bble contains ad/write.	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit	Lower nibble c and operates fr D4 10 date.0 the month. Low	ontains the low om 0 to 1. The D3 Date.3 ver nibble conta	er digit and op range for the re D2 Date.2 ins the lower d	erates from 0 to gister is 1-12. B D1 Date.1 igit and operates	9; upper attery- D0 Date.0 ; from 0 to		
	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0	ne BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o 0 0	for the month. the upper digit of the month D5 10 date 1 for the date of the upper digit of the week D5 0	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0	Dotains the low om 0 to 1. The D3 Date.3 //er nibble conta om 0 to 3. The p D3 0	er digit and operange for the re D2 Date.2 ins the lower d range for the re D2 Day.2	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 D21 D21	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0		
	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb	ne BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o 0 0 0 0 0	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corre	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the	Dotains the low om 0 to 1. The D3 Date.3 ver nibble conta om 0 to 3. The n D3 0 the week. Day of	er digit and operange for the re D2 Date.2 Date.2 Date for the re D2 Day.2 Day.2 Df the week is a	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter tha	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 at counts		
	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7	he BCD digits e bit) contains ad/write. ping – Date 0 he BCD digits bble contains ad/write. ping – Day o 0 0 ble contains a 7 then returns to	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corre to 1. The user n	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the	Dotains the low om 0 to 1. The D3 Date.3 ver nibble conta om 0 to 3. The n D3 0 the week. Day of	er digit and operange for the re D2 Date.2 Date.2 Date for the re D2 Day.2 Day.2 Df the week is a	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 at counts		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte	he BCD digits e bit) contains ad/write. Ding – Date 0 he BCD digits bble contains ad/write. Ding – Day o D6 0 ble contains a 7 then returns to ry-backed, rea	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corre to 1. The user n d/write.	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the	Dotains the low om 0 to 1. The D3 Date.3 ver nibble conta om 0 to 3. The n D3 0 the week. Day of	er digit and operange for the re D2 Date.2 Date.2 Date for the re D2 Day.2 Day.2 Df the week is a	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter tha	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 at counts		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibl from 1 to 7 date. Batte Timekeep	he BCD digits e bit) contains ad/write. Ding – Date 0 he BCD digits bble contains ad/write. Ding – Day o 0 ble contains a 7 then returns to ry-backed, reac Ding – Hour	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corrected to 1. The user m d/write. s	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea	Date.3 Date.3 ver nibble conta om 0 to 3. The normalized D3 ver nibble conta om 0 to 3. The normalized D3 0 the week. Day of ning to the day	Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.2 Date.3 Da	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter tha	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 at counts eed with the		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte	he BCD digits e bit) contains ad/write. Ding – Date 0 he BCD digits bble contains ad/write. Ding – Day o D6 0 ble contains a 7 then returns to ry-backed, rea	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corre to 1. The user n d/write.	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the	Dotains the low om 0 to 1. The D3 Date.3 ver nibble conta om 0 to 3. The n D3 0 the week. Day of	er digit and operange for the re D2 Date.2 Date.2 Date for the re D2 Day.2 Day.2 Df the week is a	D1 Date.1 igister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter tha	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 att counts		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7	ne BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o 0 ble contains a 7 then returns a ry-backed, rea ping – Hour D6	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corre to 1. The user m d/write. S D5	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea	Date.3 Date.3 Ver nibble conta om 0 to 3. The n D3 0 the week. Day on ning to the day D3	er digit and operange for the reprint of the week is a value, as the data the reprint of the rep	prates from 0 to gister is 1-12. B Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter that ay is not integrat D1 Day.1	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 at counts red with the D0		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0	he BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o 0 ble contains a 7 then returns a	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that correct to 1. The user m d/write. S D5 10 hours.1	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fr D4 0 elates to day of the nust assign mea D4 10 hours.0	Dotains the low om 0 to 1. The D3 Date.3 /er nibble conta om 0 to 3. The normalized D3 0 the week. Day of ning to the day D3 Hours.3	er digit and operange for the reprint of the week is a value, as the day of the week is a value.	D1 Date.1 igit and operates gister is 1-31. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 . ring counter that ay is not integrat D1 Hours.1	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 at counts ted with the D0 Hours.0		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0 Contains th	ne BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o 0 ble contains a 7 then returns a ry-backed, rea ping – Hour D6 0 ne BCD value	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corrector to 1. The user m d/write. S D5 10 hours.1 of hours in 24-1	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea D4 10 hours.0 hour format. Low	Date.3 Date.3 ver nibble conta Date.3 ver nibble conta D3 0 the week. Day of ning to the day D3 Hours.3 wer nibble conta	er digit and operange for the reference of the reference of the reference of the lower definition of the reference of the week is a value, as the definition of the definition of the lower definition of the lower definition of the reference of the week is a value, as the definition of the lower definition of th	D1 Date.1 igit and operates gister is 1-31. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter that ay is not integral D1 Hours.1 digit and operates	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 tt counts ted with the D0 Hours.0 es from 0		
0x05	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper	he BCD digits e bit) contains ad/write. ping – Date 0 he BCD digits bble contains ad/write. ping – Day o D6 0 ble contains a 7 then returns a 9 and 0 10 and 0 1	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that correct to 1. The user m d/write. S D5 10 hours.1 of hours in 24- its) contains the	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea D4 10 hours.0 hour format. Low	Date.3 Date.3 ver nibble conta Date.3 ver nibble conta D3 0 the week. Day of ning to the day D3 Hours.3 wer nibble conta	er digit and operange for the reference of the reference of the reference of the lower definition of the reference of the week is a value, as the definition of the definition of the lower definition of the lower definition of the reference of the week is a value, as the definition of the lower definition of th	D1 Date.1 igit and operates gister is 1-31. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 . ring counter that ay is not integrat D1 Hours.1	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 tt counts ted with the D0 Hours.0 es from 0		
0x05 0x04	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper Battery-bac	he BCD digits e bit) contains ad/write. ping – Date 0 ne BCD digits bble contains ad/write. ping – Day o D6 0 ble contains a 7 then returns to ry-backed, rea ping – Hour D6 0 he BCD value nibble (two b cked, read/write)	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corrector to 1. The user m d/write. S D5 10 hours.1 of hours in 24- its) contains the te.	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea D4 10 hours.0 hour format. Low	Date.3 Date.3 ver nibble conta Date.3 ver nibble conta D3 0 the week. Day of ning to the day D3 Hours.3 wer nibble conta	er digit and operange for the reference of the reference of the reference of the lower definition of the reference of the week is a value, as the definition of the definition of the lower definition of the lower definition of the reference of the week is a value, as the definition of the lower definition of th	D1 Date.1 igit and operates gister is 1-31. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter that ay is not integral D1 Hours.1 digit and operates	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 tt counts ted with the D0 Hours.0 es from 0		
0x05 0x04	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper Battery-base Timekeep	he BCD digits e bit) contains ad/write. ping – Date D6 0 he BCD digits bble contains ad/write. ping – Day o D6 0 ble contains a 7 then returns to ry-backed, rea ping – Hour D6 0 he BCD value r nibble (two b cked, read/write ping – Minu	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that corrector to 1. The user m d/write. S D5 10 hours.1 of hours in 24 its) contains the te. tes	Lower nibble c and operates fr D4 10 date.0 the month. Low and operates fro D4 0 elates to day of to nust assign mea D4 10 hours.0 hour format. Lo e upper digit an	Date.3 Date.3 ver nibble conta on 0 to 3. The normalized D3 0 0 0 0 0 0 0 0 0 0 0 0 0	er digit and operange for the re D2 Date.2 ins the lower d range for the re D2 Day.2 of the week is a value, as the day Hours2 tains the lower 0 to 2. The ran	prates from 0 to gister is 1-12. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter that ay is not integrated D1 Hours.1 digit and operated gister is not integrated	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 at counts red with the D0 Hours.0 es from 0 es from 0 er is 0-23.		
0x05 0x04	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibb from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper Battery-bac Timekeep D7	he BCD digits ad/write. ping – Date 0 0 he BCD digits bble contains ad/write. ping – Day o 0 ble contains a 7 then returns to ry-backed, read ping – Hour 0 he BCD value n BCD value n bble (two b cked, read/write) ping – Minu D6	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that correct to 1. The user m d/write. S D5 10 hours.1 of hours in 24 its) contains the te. tes D5	Lower nibble c and operates fro D4 10 date.0 the month. Low and operates fro D4 0 elates to day of t nust assign mea D4 10 hours.0 hour format. Lo e upper digit an D4	D3 Date.3 Date.3 ver nibble conta om 0 to 3. The n D3 0 the week. Day of ning to the day D3 Hours.3 over nibble cond d operates from D3	er digit and operange for the reference of the week is a value, as the data va	D1 Date.1 igit and operates gister is 1-31. B D1 D2 D1 D3 D1 D3 D1 D3 D1 D3 D1 D3 D1 D4 D1 D3 D1 D4 D1 D3 D1 digit and operates D1 Hours.1 digit and operates D1 Hours.1 digit and operates D1 Hours.1	9; upper attery- D0 Date.0 5 from 0 to attery- D0 Day.0 tt counts ted with the D0 Hours.0 es from 0 ter is 0-23.		
0x06 0x05 0x04 0x04	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibl from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper Battery-bac Timekeep D7 0 Contains th to 9; upper Battery-bac	he BCD digits e bit) contains ad/write. Ding – Date 0 he BCD digits bble contains ad/write. Ding – Day o D6 0 ble contains a 7 then returns to ry-backed, rea D6 0 he BCD value nibble (two b cked, read/write D6 0 10 min.2	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that correct to 1. The user m d/write. s D5 10 hours.1 of hours in 24-1 its) contains the te. tes D5 10 min.1	Lower nibble c and operates fro D4 10 date.0 the month. Low and operates fro D4 0 elates to day of the nust assign mea D4 10 hours.0 hour format. Lo e upper digit an D4 10 min.0	D3 Date.3 Ver nibble conta on 0 to 3. The normalized of the second ning to the day D3 Hours.3 over nibble cond d operates from D3 Min.3	er digit and operange for the re D2 Date.2 ins the lower d range for the re D2 Day.2 of the week is a value, as the day Hours2 tains the lower 0 to 2. The ran D2 Min.2	Prates from 0 to gister is 1-12. B D1 Date.1 igit and operates gister is 1-31. B D1 Day.1 ring counter that ay is not integrat D1 Hours.1 digit and operates gister is 1.31. B	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 at counts ted with the D0 Hours.0 es from 0 es from 0 es from 0 es from 0 Min.0		
0x05 0x04	Contains th nibble (one backed, rea Timekeep D7 0 Contains th 9; upper ni backed, rea Timekeep D7 0 Lower nibl from 1 to 7 date. Batte Timekeep D7 0 Contains th to 9; upper Battery-bac Timekeep D7 0 Contains th to 9; upper D7 0 Contains th to 9; upper	he BCD digits e bit) contains ad/write. Ding – Date 0 he BCD digits bble contains ad/write. Ding – Day o D6 0 ble contains a 7 then returns to ry-backed, rea D6 0 he BCD value nibble (two b cked, read/write D6 0 he BCD value nibble (two b cked, read/write D6 0 he BCD value nibble (two b cked, read/write D6 10 min.2 he BCD value	for the month. the upper digit of the month D5 10 date.1 for the date of the upper digit of the week D5 0 value that correct to 1. The user m d/write. S D5 10 hours.1 of hours in 24-1 its) contains the te. tes D5 10 min.1 of minutes. Low	Lower nibble c and operates fro D4 10 date.0 the month. Low and operates fro D4 0 elates to day of to nust assign mea D4 10 hours.0 hour format. Lo e upper digit an D4 10 min.0 wer nibble cont	D3 Date.3 Date.3 Ver nibble conta om 0 to 3. The n D3 0 the week. Day of ning to the day D3 Hours.3 ower nibble cond d operates from D3 Min.3 ains the lower of	er digit and operange for the re D2 Date.2 ins the lower d range for the re D2 Day.2 of the week is a value, as the day Hours2 tains the lower 0 to 2. The ran D2 Min.2 digit and operat	D1 Date.1 igit and operates gister is 1-31. B D1 D2 D1 D3 D1 D3 D1 D3 D1 D3 D1 D3 D1 D4 D1 D3 D1 D4 D1 D3 D1 digit and operates D1 Hours.1 digit and operates D1 Hours.1 digit and operates D1 Hours.1	9; upper attery- D0 Date.0 3 from 0 to attery- D0 Day.0 at counts at counts		

	Timekeeping – Seconds												
	D7	D6	D5	D4	D3	D2	D1	D0					
	0	10 sec.2	10 sec.1	10 sec.0	Seconds.3	Seconds.2	Seconds.1	Seconds.0					
	Contains the			ver nibble conta	ains the lower d	ligit and opera	tes from 0 to 9;						
							Battery-backed,						
0x01	CAL/Cont	rol											
	D7	D6	D5	D4	D3	D2	D1	DO					
	-	-	CALS	CAL.4	CAL.3	CAL.2	CAL.1	CAL.0					
CALS	Calibration Sign: Determines if the calibration adjustment is applied as an addition to or as a subtraction from time-base. This bit can be written only when CAL=1. Nonvolatile, read/write.												
CAL.4-0				the calibration	of the clock.	These bits can	be written only	when					
	CAL=1. No	nvolatile, read/	write.										
0x00	RTC/Alar	m Control											
0400	D7	D6	D5	D4	D3	D2	D1	D0					
			-			*							
OCCENT	OSCEN	AF	CF	AEN	Reserved	CAL	W	R					
/OSCEN							ator runs. Disab						
							ce or on a powe llator. Battery-b						
	read/write.	appi	ieu, tills bit is	internany set u	, which tur	iis off the oser	nator. Battery-0	ackeu,					
AF		This bit is set	to 1 when the	time and date r	natch the value	s stored in the	alarm registers	with the					
				to '0'. Battery-l									
CF							erflows from 99	to 00. This					
							should record th						
		rmation as nee	ded. The user	must clear the	CF bit to '0'. B	attery-backed.	(internally set,	user must					
4 5 3 7	clear bit)	1			4 53 4 1 4 4	10111							
AEN							ed), the ACS pi						
							leared, no new a ear AF. Battery-						
CAL													
						Calibration Mode: When CAL is set to 1, the clock enters calibration mode. When CAL is set to 0, the clock							
	operates normally, and the ACS pin is controlled by the RTC alarm. Battery-backed, read/write. Write Time. Setting the W bit to 1 freezes updates of the user timekeeping registers. The user can then write												
W				s updates of the									
W	Write Time. them with u	Setting the W pdated values.	bit to 1 freeze Setting the W	bit to 0 causes	e user timekeep	oing registers.		en write					
	Write Time. them with up timekeeping	Setting the W pdated values. counters. Batt	bit to 1 freeze Setting the W ery-backed, r	bit to 0 causes ead/write.	e user timekeep the contents of	oing registers. The time regis	The user can the ters to be transf	en write erred to the					
W	Write Time. them with u timekeeping Read Time.	Setting the W pdated values. counters. Batt Setting the R b	bit to 1 freeze Setting the W ery-backed, re it to '1' copie	bit to 0 causes ead/write. s a static image	e user timekeep the contents of of the timekee	bing registers. The time regis	The user can the ters to be transf places it into th	en write erred to the e user					
	Write Time. them with u timekeeping Read Time. registers. Th	Setting the W pdated values. counters. Batt Setting the R b we user can ther	bit to 1 freeze Setting the W ery-backed, re it to '1' copie read them w	bit to 0 causes ead/write. s a static image ithout concerns	e user timekeep the contents of of the timekee over changing	bing registers. The time regis pping core and values causing	The user can the ters to be transf places it into th g system errors.	en write erred to the e user The R bit					
	Write Time. them with u timekeeping Read Time. registers. Th	Setting the W pdated values. counters. Batt Setting the R t e user can ther 0 to 1 causes th	bit to 1 freeze Setting the W ery-backed, re it to '1' copie read them w	bit to 0 causes ead/write. s a static image ithout concerns	e user timekeep the contents of of the timekee over changing	bing registers. The time regis pping core and values causing	The user can the ters to be transf places it into th	en write erred to the e user The R bit					

Electrical Specifications

Absolute Maximum Ratings

Symbol	Description	Ratings
V _{DD}	Power Supply Voltage with respect to V _{SS}	-1.0V to +3.6V
V_{IN}	Voltage on any signal pin with respect to V_{SS}	-1.0V to +5.0V and
		$V_{IN} < V_{DD} + 1.0V$
V_{BAK}	Backup Supply Voltage	-1.0V to +4.5V
T _{STG}	Storage Temperature	-55°C to + 125°C
T _{LEAD}	Lead Temperature (Soldering, 10 seconds)	300° C
V_{ESD}	Electrostatic Discharge Voltage	
	- Human Body Model (JEDEC Std JESD22-A114-E)	TBD
	- Charged Device Model (JEDEC Std JESD22-C101-C)	TBD
	- Machine Model (JEDEC Std JESD22-A115-A)	TBD
	Package Moisture Sensitivity Level	MSL-3

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only, and the functional operation of the device at these or any other conditions above those listed in the operational section of this specification is not implied. Exposure to absolute maximum ratings conditions for extended periods may affect device reliability.

DC Operating C	Conditions (T _A	$= -40^{\circ} \text{ C to} + 85^{\circ}$	C, $V_{DD} = 3.0V$ to	0 3.6V unless	otherwise specified	I)
----------------	----------------------------	---	-----------------------	---------------	---------------------	----

Symbol	Parameter	Min	Тур	Max	Units	Notes
V _{DD}	Main Power Supply	3.0	-	3.6	V	1
I _{DD}	V _{DD} Supply Current (VBC=0)				mA	
I_{SB}	Standby Current Trickle Charger Off (VBC=0)			50	μΑ	3
V _{BAK}	RTC Backup Voltage	2.0	3.0	3.6	V	4
I _{BAK}	RTC Backup Current			1	μΑ	5
I _{BAKTC}	Trickle Charge Current with $V_{BAK}=0V$ Fast Charge Off (FC = 0) Fast Charge On (FC = 1)	50 200	2	200 2500	μΑ μΑ	6
V_{TP0}	V_{DD} Trip Point Voltage for MCU companion & RTC, VTP(1:0) = 00b	2.55	2.6	2.70	V	7
V_{TP1}	V_{DD} Trip Point Voltage for MCU companion & RTC, VTP(1:0) = 01b	2.70	2.75	2.85	V	7
V _{TP2}	V_{DD} Trip Point Voltage for MCU companion & RTC, VTP(1:0) = 10b	2.80	2.9	2.97	V	7
V _{TP3}	V_{DD} Trip Point Voltage for MCU companion & RTC, VTP(1:0) = 11b	2.93	3.0	3.13	V	7
V _{RST}	$V_{BAK} > V_{BAK} min$ $V_{BAK} < V_{BAK} min$	0 1.6			V V	
V_{SW}	Battery Switchover Voltage	2.0		2.7	V	
I _{LI}	Input Leakage Current			TBD	μΑ	
ILO	Output Leakage Current			TBD	μΑ	
V _{IL}	Input Low Voltage All inputs except as listed below CNT battery-backed ($V_{DD} < V_{SW}$) CNT ($V_{DD} > V_{SW}$)	-0.3 -0.3 -0.3		0.3 V _{DD} 0.5 0.8	V V V	
V _{IH}	Input High Voltage All inputs except as listed below CNT battery-backed ($V_{DD} < V_{SW}$) CNT $V_{DD} > V_{SW}$ PFI	$\begin{array}{c} 0.7 \ V_{DD} \\ V_{BAK} - 0.5 \\ 0.7 \ V_{DD} \\ - \end{array}$		$V_{DD} + 0.3$ $V_{BAK} + 0.3$ $V_{DD} + 0.3$ $V_{DD} + 0.3$	V V V V	

Continued >>

DC Oper	atting Conditions, continued $(1_A - 40)$ C to + 85	$C, V_{DD} = 5.0 V$	10 5.0 V un	Decoperating conditions, continued $(1_A - 40^\circ \text{C} \text{ to } + 85^\circ \text{C}, v_{DD} - 5.0^\circ \text{ to } 5.0^\circ \text{ unless offer wise spectrum})$				
Symbol	Parameter	Min	Тур	Max	Units	Notes		
V _{OL}	Output Low Voltage $@$ I _{OL} = 3 mA	-		0.4	V			
V _{OH}	Output High Voltage							
	(PFO) @ $I_{OH} = -2 mA$	$V_{DD}-0.8$		-	V			
R _{RSTB}	Pull-up resistance for RSTB inactive	50		400	KΩ			
V _{PFI}	Power Fail Input Reference Voltage	1.475	1.50	1.525	V			
V _{HYS}	Power Fail Input (PFI) Hysteresis (Rising)		-	100	mV			

DC Operating Conditions, continued ($T_A = -40^\circ \text{ C to} + 85^\circ \text{ C}$, $V_{DD} = 3.0 \text{ V to } 3.6 \text{ V}$ unless otherwise specified)

Notes

1. Full complete operation. Supervisory circuits, RTC, etc operate to lower voltages as specified.

2. All inputs at V_{SS} or V_{DD} , static. Trickle charger off (VBC=0).

3. The VBAK trickle charger automatically regulates the maximum voltage on this pin for capacitor backup applications.

- 4. $V_{BAK} = 3.0V, V_{DD} < V_{SW}$, oscillator running, CNT at VBAK.
- 5. V_{BAK} will source current when trickle charge is enabled (VBC bit=1), $V_{DD} > V_{BAK}$, and $V_{BAK} < V_{BAK}$ max.
- 6. This is the V_{DD} supply current contributed by enabling the trickle charger circuit, and does not account for I_{BAKTC} .

7. The minimum V_{DD} to guarantee the level of RSTB remains a valid V_{OL} level.

AC Parameters ($T_A = -40^\circ$ C to + 85° C, $V_{DD} = 3.0$ V to 3.6V, $C_L = 100$ pF unless otherwise specified)

Symbol	Parameter	Min	Max	Units	Notes
f _{SCL}	SCL Clock Frequency	0	100	kHz	
$t_{\rm LOW}$	Clock Low Period	4.7		μs	
t _{HIGH}	Clock High Period	4.0		μs	
t _{AA}	SCL Low to SDA Data Out Valid		3	μs	
t _{BUF}	Bus Free Before New Transmission	4.7		μs	
t _{HD:STA}	Start Condition Hold Time	4.0		μs	
t _{SU:STA}	Start Condition Setup for Repeated Start	4.7		μs	
t _{HD:DAT}	Data In Hold Time	0		ns	
t _{SU:DAT}	Data In Setup Time	TBD		ns	<i>y</i>
t _R	Input Rise Time		TBD	ns	1
t _F	Input Fall Time		TBD	ns	1
t _{SU:STO}	Stop Condition Setup Time	4.0	5	μs	
t _{DH}	Data Output Hold (from SCL @ VIL)	0		ns	

All SCL specifications as well as start and stop conditions apply to both read and write operations.

Supervisor Timing ($T_A = -40^\circ C$ to $+ 85^\circ C$, $V_{DD} = 3.0V$ to 3.6V)

Symbol	Parameter	Min	Max	Units	Notes
t _{RPW}	RSTB Pulse Width (active low time)	30	100	ms	
t _{RNR}	RSTB Response Time to $V_{DD} < V_{TP}$ (noise filter)	7	25	μs	1
t _{VR}	V _{DD} Rise Time	50	-	μs/V	1,2
t _{VF}	V _{DD} Fall Time	100	-	μs/V	1,2
t _{WDST}	Watchdog StartTime	0.3*t _{DOG1}	t _{DOG1}	ms	3
t _{WDET}	Watchdog EndTime	t _{DOG2}	3.3*t _{DOG2}	ms	3
f _{CNT}	Frequency of Event Counter	0	TBD	kHz	

Notes

1 This parameter is characterized but not tested.

2 Slope measured at any point on V_{DD} waveform.

3 t_{DOG1} is the programmed StartTime and t_{DOG2} is the programmed EndTime in registers 0Bh and 0Ch, $V_{DD} > V_{TP}$, and t_{RPU} satisfied. The StartTime has a resolution of 25ms. The EndTime has a resolution of 60ms.

4 The RSTB pin will drive low for this amount of time after the internal reset circuit is activated due to a watchdog, low voltage, or manual reset event.

Capacitance ($T_A = 25^\circ \text{ C}$, f=1.0 MHz, $V_{DD} = 3.0 \text{ V}$)

Symbol	Parameter	Тур	Max	Units	Notes
C _{IO}	Input/Output Capacitance	-	20	pF	1
C _{XTL}	X1, X2 Crystal pin Capacitance	25	-	pF	1, 2
C _{CNT}	Max. Allowable Capacitance on CNT (polled mode)	-	100	pF	

Notes

1 This parameter is characterized but not tested.

2 The crystal attached to the X1/X2 pins must be rated as 12.5pF.

Data Retention ($V_{DD} = 3.0V$ to 3.6V)

Parameter	Min	Units	Notes
Data Retention	10	Years	

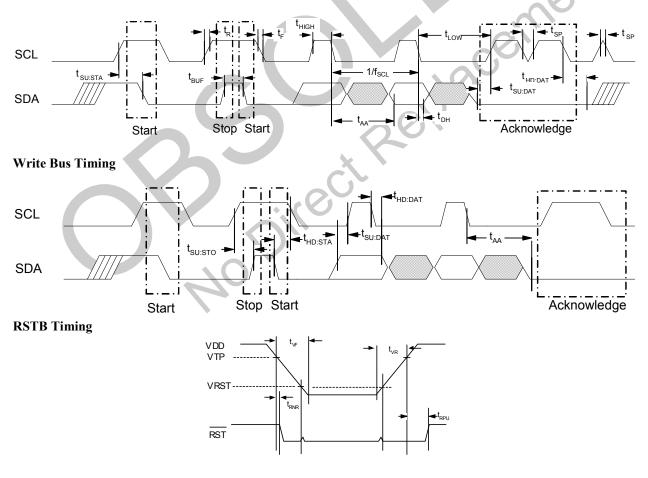
AC Test Conditions

Input Pulse Levels	10% and 90% of V_{DD}
Input Rise and Fall Times	5 ns
Input and Output Timing Levels	$0.5 V_{DD}$
Output Load Capacitance	100 pF

Diagram Notes

All start and stop timing parameters apply to both read and write cycles. Clock specifications are identical for read and write cycles. Write timing parameters apply to slave address, word address, and write data bits. Functional relationships are illustrated in the relevant data sheet sections. These diagrams illustrate the timing parameters only.

Read Bus Timing



FM6124 Interface Code Example

The following C code provides an example of basic interface to the FM6124 using Ramtron's High Performance VRS51L2070 (8051-based) MCU.

```
11
// $Date: 2008-03-10 13:04:43 -0400 (Mon, 10 Mar 2008) $
// $Rev: 250 $
// $Author: smalo $
// $fm6124 basic example vrs2070.c $
11
11
// Description: This file contains code examples for a
            Host MCU (Ramtron VRS51L2070) communicating with an
//
//
            Event Data Recorder (Ramtron FM6124 EDR).
            The host MCU is connected to the FM6124 with an I2C interface.
11
11
            The compiler used for this example is SDCC 2.7.0
11
           For the purpose of this example, the device selection bits
// Remarks:
            of the FM6124 (A0 and A1) are fixed to 1.
11
            See global variable "g_uiI2CDevSelect".
//
11
// Copyright (C) 2008 Ramtron International Corporation
11
#include <malloc.h>
#include "VRS51L2070 SDCC.h"
                           // VRS51L2070 registers definitions
#include "fm6124.h"
                           // FM6124 registers definitions
// Function Prototypes
// I2C Init functions
void
      InitI2C();
      IsI2CSlaveReady(ruint8 a_uiSlaveId);
rbool
// FM6124 Access functions
ruint8 FM6124ReadReg(ruint8 a_uiRegAddr);
rbool
      FM6124WriteReg(ruint8 a_uiRegAddr, ruint8 a_uiRegValue)
      FM6124WriteRTC(struct SBCDDate *a poDate);
rbool
ruint8 FM6124ReadFRAM(ruint16 a_uiFramAddr);
      FM6124WriteFRAM(ruint16 a_uiFramAddr, ruint8 a_uiValue);
rbool
      FM6124ReadEventAtRP(struct SEvent *a poEvent);
rbool
      FM6124StreamEventsAtRP(ruint8* a_puiEvents, ruint8 a_uiNbEvents);
rbool
// Utility functions
      Delay(ruint16 a uiDelayMs);
void
      CreateEventsP5(ruint16 a uiNbEvents);
rbool
// Global variables
const ruint8 g uiI2CDevSelect = 0x03 << 1;</pre>
```

// Function: main // Description: main function of the program // Parameters: None // Return value: int: Error Code // Remarks: For the purpose of this example, the return values of functions are not verified. In a real application, 11 11 return values should be verified in order to make sure that 11 any all operations have completed successfully. int main() { idata ruint8 l uiDataByte $= 0 \times 00;$ luiNbEvents idata ruint16 $= 0 \times 0000;$ xdata struct SBCDDate 1 oRTCValue; ___xdata struct SEvent l_oEvent; xdata ruint8* l puiEvents = NULL; // INIT Phase $^{\prime\prime}$ // Initialize I2C interface, making sure FM6124 is responding. InitI2C(); // Reset RTC to a defined value l oRTCValue.uiSeconds $= 0 \times 00;$ l oRTCValue.uiMinutes $= 0 \times 01;$ l oRTCValue.uiHours $= 0 \times 02;$ l oRTCValue.uiDay $= 0 \times 03;$ l oRTCValue.uiDate = 0x10;l oRTCValue.uiMonth $= 0 \times 09;$ l oRTCValue.uiYear $= 0 \times 07;$ FM6124WriteRTC(&l oRTCValue); // Configure Event Buffer to 1000 Events FM6124WriteReg(EDR REG BUFFER CTRL, EDR BC CMD EB SIZE | EDR BC VAR 3000 EVENTS); Delay(1); FM6124WriteReg(EDR REG BUFFER CTRL, EDR BC CMD EB SIZE | EDR BC VAR 1000 EVENTS); // Make sure FM6124 has enough time to setup the new Event Buffer (100us required) Delay(1); // F-RAM Write/Read // Write a Byte into the F-RAM at address 0x0000 FM6124WriteFRAM(0x0000, 0xBD); // Read a Byte of Data in the F-RAM at address 0x0000 l uiDataByte = FM6124ReadFRAM(0x0000); // Create Events
// Enable Event Recording of Digital Input 4-11 (Rising Edges) FM6124WriteReg(EDR_REG_PIN_RF_B, 0xFF); FM6124WriteReg(EDR REG PIN EE B, 0xFF); // Make sure FM6124 has enough time to enable Event Recording (100us required) Delay(1); // Disable Event Recording FM6124WriteReg(EDR REG PIN EE B, 0x00); // Read Events // Read Number of Events (Number of Events between RP and WP) FM6124WriteReg(EDR REG PIN SNAP, 0x02); l uiNbEvents = FM6124ReadReg(EDR REG EVENT COUNT MSB); l uiNbEvents <<= 8;</pre> l uiNbEvents |= FM6124ReadReg(EDR REG EVENT COUNT LSB); // Read newest event: // 1- Move RP to the last event FM6124WriteReg(EDR_REG_BUFFER_CTRL, EDR_BC_CMD_LAST); // 2- Read Event FM6124ReadEventAtRP(&l oEvent);

```
// Read 5 oldest Events by streaming:
   // 1- Move RP to first event:
   FM6124WriteReg(EDR REG BUFFER CTRL, EDR BC CMD FIRST);
   // 2- Allocate memory for 5 Events
   l puiEvents = malloc(5 * sizeof(struct SEvent));
   // 3- Stream 5 Events
   if (1 puiEvents != NULL)
   {
       FM6124StreamEventsAtRP(l_puiEvents, 5);
   }
   // Final note: to "see" the result of this example, it can be executed with
   // Ramtron's Versaware JTAG debuqger. At this point, you can verify that all operations
   // were completed successfully by looking at the memory content of the Host MCU
   // running this program.
   if (l uiDataByte != 0xBD)
   {
       return 1;
   if (l_uiNbEvents != 10)
   {
       return 2;
   }
   return 0;
}
InitI2C
// Function:
// Description:
                 Enable and initialize I2C in MASTER Mode
// Parameters:
                 None
// Return value:
                 None
// Remarks:
                 None
void
                                                              'Ce,
InitI2C()
{
   // Enable I2C module
   PERIPHEN1 |= 0x20;
   // Init the transmit portion of the I2CRXTX buffer
   // I2C Master
   I2CCONFIG
             = 0x01;
   I2CIDCFG
             = 0x41;
   // I2C Comm Speed = 96.15 Khz (Max speed of FM6124 is 100kHz)
             = 0 \times 0C;
   12CTIMING
   // Make sure FM6124 is responding
   while(!IsI2CSlaveReady(I2C_ID_EDR))
                                 O're'
   {
       Delay(1000);
   }
}
```

// Function: IsI2CSlaveReadv // Description: Check if an I2C Slave module is ready by issuing a write operation. We abort the write operation after the I2C Id 11 is transmitted. If the I2C Id is Acknowledged, it means // 11 that the I2C slave module is ready. ruint8 a uiSlaveId: Id of the Slave Module to check // Parameters: // Return value: rbool: RFALSE = Slave is not ready RTRUE = Slave is ready 11 // Remarks: None rbool IsI2CSlaveReady(ruint8 a uiSlaveId) // Send Write command to Slave Module I2CRXTX = a uiSlaveId | g uiI2CDevSelect | IC2 WRITE; // Wait for TX Buffer to be empty while(!(I2CSTATUS & 0x01)); // Wait for I2C to be Idle // This will generate a STOP and cancel the Write operation while(!(I2CSTATUS & 0x08)); if (I2CSTATUS & 0x40) { // No ACK received, Slave Module is not ready return RFALSE; } return RTRUE; 111111111111 FM6124ReadReq // Function: Read a Register of the FM6124 // Description: ruint8 a uiRegAddr // Parameters: // Return value: ruint8: Value of the register read To Read a register on the I2C, we first need to "fake" a write operation in order to send the register address to the // Remarks: 11 slave module. 11 ruint8 FM6124ReadReg(ruint8 a_uiRegAddr) ruint8 l uiValue = 0x00; // Make sure I2C is Idle while(!(I2CSTATUS & 0x08)); // Wait for TX Buffer to be empty while((I2CSTATUS & 0x01) == 0); // Dummy Read to clear the I2CRXAVF l_uiValue = I2CRXTX; $/\overline{/}$ "Fake" write operation to send the register address we want to read I2CRXTX = I2C_ID_EDR | g_uiI2CDevSelect | IC2_WRITE; // Wait for TX Buffer to be empty while((I2CSTATUS & 0x01) == 0); // Make sure we received an ACK. if (I2CSTATUS & 0xC0) { return 0xFF; // Send Read Address I2CRXTX = a uiReqAddr; // Wait for I2C to be Idle and generate a STOP while(!(I2CSTATUS & 0x08)); // Make sure we received an ACK. if (I2CSTATUS & 0xCO) return 0xFF; }

```
// READ OPERATION
   // Dummy Read to clear the I2CRXAVF
   l uiValue = I2CRXTX;
   // Make sure we will ACK the Data we will receive
   I2CCONFIG &= 0xFD;
   // Now, send the Read command to the FM6124
   I2CRXTX = I2C_ID_EDR | g_uiI2CDevSelect | IC2_READ;
   // Wait for Data to comeback
   while(!(I2CSTATUS & 0x02));
   l uiValue = I2CRXTX;
   // Stop the transaction
   I2CCONFIG |= 0x02;
   // Wait for I2C to be Idle and generate a STOP
   while(!(I2CSTATUS & 0x08));
   return l uiValue;
}
// Function:
                  FM6124WriteReg
// Description:
                  Write a Register of the FM6124
// Parameters:
                  ruint8 a_uiRegAddr
11
                  ruint8 a_uiRegValue
// Return value:
                  rbool: RTRUE = Operation Ok.
// Remarks:
                  None
rbool
FM6124WriteReg(ruint8 a uiRegAddr, ruint8 a uiRegValue)
{
   // Make sure I2C is Idle
   while(!(I2CSTATUS & 0x08));
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Send Write command
   I2CRXTX = I2C_ID_EDR | g_uiI2CDevSelect | IC2_WRITE;
   // Wait for TX Buffer to be empty
   while ((I2CSTATUS \& 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   }
   // Send Write Address
   I2CRXTX = a_uiRegAddr;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01)
                           == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   // Send Write Data
   I2CRXTX = a uiRegValue;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   }
   // Wait for I2C to be Idle, this will generate a STOP
   while(!(I2CSTATUS & 0x08));
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   }
   return RTRUE;
}
```

Ramtron

```
// Function:
                   FM6124WriteRTC
// Description:
                   Write all registers of the RTC of the FM6124.
// Parameters:
                   struct SBCDDate *a_poDate: Date to be written
// Return value:
                   rbool: RFALSE = Error during read operation.
// Remarks:
                   None
rbool
FM6124WriteRTC(struct SBCDDate *a poDate)
{
    ruint8 l_uiCounter
                              = 0 \times 00;
   ruint8 *1_puiDateElement
                              = NULL;
   ruint8 l uiRTCRegValue
                              = 0 \times 00;
    // First, we need to read the RTC register of the FM6124
   l uiRTCRegValue = FM6124ReadReg(EDR REG RTC);
   //\ensuremath{\,{\rm Next}} , we need to enable the writing of the RTC
    l uiRTCRegValue |= 0x02;
   FM6124WriteReg(EDR_REG_RTC, l_uiRTCRegValue);
    // Make sure I2C is Idle
   while(!(I2CSTATUS & 0x08));
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
    // Send Write command to Slave Module
   I2CRXTX = I2C_ID_EDR | g_uiI2CDevSelect | IC2_WRITE;
    // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
    // Make sure we received an ACK and that there was no error during transfer
    if (I2CSTATUS & 0xC0)
    {
       return RFALSE;
    // Send Write Address
   I2CRXTX = EDR REG RTC SECS;
    // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
    // Make sure we received an ACK and that there was no error during transfer
    if (I2CSTATUS & 0xC0)
    {
       return RFALSE;
    }
    // Write RTC Data
   l_puiDateElement = &(a_poDate->uiSeconds);
    for (1 uiCounter = 0; 1 uiCounter < 7; 1 uiCounter+
    {
       I2CRXTX = *1 puiDateElement;
       // Wait for TX Buffer to be empty
       while((I2CSTATUS & 0x01) == 0);
       // Make sure we received an ACK and that there was no error during transfer
          (I2CSTATUS & 0xC0)
       if
           return RFALSE;
       l puiDateElement++;
   }
    // Wait for I2C to be Idle, This will generate a STOP
   while(!(I2CSTATUS & 0x08));
    // Now, we need to restore the original RTC register value.
   // Clear "W" bit
    // Make sure Oscillator is running: Clear /OSCEN bit
   l uiRTCReqValue &= 0x7D;
   FM6124WriteReg(EDR REG RTC, 1 uiRTCRegValue);
   return RTRUE;
}
```

```
// Function:
                  FM6124ReadFRAM
// Description:
                  Read 1 byte of Data in the F-RAM of the FM6124
// Parameters:
                  ruint16 a uiFramAddr
// Return value:
                  ruint8: Byte read
                  To Read a data on the I2C, we first need to "fake"
// Remarks:
11
                  a write operation in order to send the memory address to the
11
                  slave module.
ruint8
FM6124ReadFRAM(ruint16 a_uiFramAddr)
{
   ruint8 l uiDataRead;
   // Make sure I2C is Idle
   while(!(I2CSTATUS & 0x08));
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Dummy Read to clear the I2CRXAVF
   l_uiDataRead = I2CRXTX;
// Send Write command to FM6124
   I2CRXTX = I2C ID FM | q uiI2CDevSelect | IC2 WRITE;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return 0xFF;
   // Send Read Address, 8 MSB
   I2CRXTX = (a uiFramAddr >> 8) & 0xFF;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return 0xFF;
   // Send Read Address, 8 LSB
   I2CRXTX = a uiFramAddr & 0xFF;
   // Wait for I2C to be Idle, This will generate a STOP
   while(!(I2CSTATUS & 0x08));
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return 0xFF;
   }
   // Now, send the Read command to the FM6124
   I2CRXTX = I2C ID FM | g_uiI2CDevSelect | IC2_READ;
   // Wait for Data to comeback
   while(!(I2CSTATUS & 0x02));
   l uiDataRead = I2CRXTX;
   // Stop the transaction
   I2CCONFIG | = 0x02;
   // Wait for I2C to be Idle, This will generate a STOP
   while(!(I2CSTATUS & 0x08));
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return 0xFF;
   }
   return l uiDataRead;
```

Ramtron

```
// Function:
                  FM6124WriteFRAM
// Description:
                  Write 1 byte of Data into the FRAM of the FM6124
// Parameters:
                  ruint16 a uiFramAddr
11
                  ruint8 a_uiValue
// Return value:
                  rbool: RTRUE = Operation successful.
// Remarks:
                  None
rbool
FM6124WriteFRAM(ruint16 a uiFramAddr, ruint8 a uiValue)
{
   // Make sure I2C is Idle
   while(!(I2CSTATUS & 0x08));
   // Wait for TX Buffer to be empty
   while ((I2CSTATUS \& 0x01) == 0);
   // Send Write command to FM6124
   I2CRXTX = I2C_ID_FM | g_uiI2CDevSelect | IC2_WRITE;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   // Send Write Address, 8 MSB
   I2CRXTX = (a uiFramAddr >> 8) & 0xFF;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   // Send Write Address, 8 LSB
   I2CRXTX = a_uiFramAddr & 0xFF;
   // Wait for TX Buffer to be empty
   while ((I2CSTATUS \& 0x01) == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   }
   // Send 1 Byte of Data
   I2CRXTX = a_uiValue;
   // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01)
                           == 0);
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   ł
       return RFALSE;
   }
   // Wait for I2C to be Idle, This will generate a STOP
   while(!(I2CSTATUS & 0x08));
   // Make sure we received an ACK and that there was no error during transfer
   if (I2CSTATUS & 0xC0)
   {
       return RFALSE;
   }
   return RTRUE;
}
```

// Function: FM6124ReadEventAtRP // Description: Read Event pointed by the current position of RP (Read Pointer) // Parameters: struct SEvent *a poEvent: Pointer to a SEvent, which 11 will receive the Event's values. rbool: RTRUE = Read Ok. // Return value: // Remarks: None rbool FM6124ReadEventAtRP(struct SEvent *a poEvent) // First, we need to indicate the FM6124 that we want to read an Event FM6124WriteReg(EDR REG BUFFER CTRL, EDR BC CMD GET); // Make sure I2C is Idle while(!(I2CSTATUS & 0x08)); // Wait for TX Buffer to be empty while((I2CSTATUS & 0x01) == 0); // Dummy Read, to clear the I2CRxAv flag. a poEvent->uiEventCode = I2CRXTX; $/\overline{/}$ "Fake" write operation to send the register address we want to read I2CRXTX = I2C ID EDR | g uiI2CDevSelect | IC2 WRITE; // Wait for TX Buffer to be empty while((I2CSTATUS & 0x01) == 0); // Make sure we received an ACK and that there was no error during transfer if (I2CSTATUS & 0xC0) { return RFALSE; // Send Read Address I2CRXTX = EDR REG EVT CODE; // Wait for I2C to be Idle and generate a STOP while(!(I2CSTATUS & 0x08)); // Make sure we received an ACK and that there was no error during transfer 3012C if (I2CSTATUS & 0xC0) { return RFALSE; } // READ EVENT // Dummy Read to clear the I2CRXAVF a_poEvent->uiEventCode = I2CRXTX; // Make sure we will ACK the Data we will receive I2CCONFIG &= 0xFD; // Now, send the Read command to the Slave ER I2CRXTX = I2C_ID_EDR | g_uiI2CDevSelect | IC2_READ; while(!(I2CSTATUS & 0x02)); a poEvent->uiEventCode = I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiSeconds I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiMinutes I2CRXTX; while(!(I2CSTATUS & 0x02)); a_poEvent->oBCDDate.uiHours I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiDay = I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiDate = I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiMonth = I2CRXTX; while(!(I2CSTATUS & 0x02)); a poEvent->oBCDDate.uiYear = I2CRXTX; // Stop the transaction I2CCONFIG $| = 0 \times 02;$ // Wait for I2C to be Idle and generate a STOP while(!(I2CSTATUS & 0x08)); return RTRUE;

}

```
// Function:
                   FM6124StreamEventsAtRP
// Description:
                   Read x events by Streaming, starting at RP
// Parameters:
                   rbool: RTRUE = Read Ok.
// Return value:
// Remarks:
                   User is responsible of:
11
                   1- Making sure a_puiEvents has enough space to contain
11
                      a uiNbEvents events.
                    2- There is at least a_uiNbEvents in the Event Buffer of the
11
11
                      FM6124.
rbool
FM6124StreamEventsAtRP(ruint8* a puiEvents, ruint8 a uiNbEvents)
{
    ruint8 l uiCurEvent
                           = 0 \times 00;
   ruint8 l uiCurData
                           = 0 \times 00;
          l bEventFF
                           = RFALSE;
   rbool
   ruint8* l_puiVufPrt
                           = a puiEvents;
    // Send STEAM command to the FM6124
   FM6124WriteReg(EDR REG BUFFER CTRL, EDR BC CMD STREAM);
    // Make sure I2C is Idle
   while(!(I2CSTATUS & 0x08));
    // Wait for TX Buffer to be empty
   while((I2CSTATUS & 0x01) == 0);
    // Dummy Read, to clear the I2CRxAv flag.
    *a puiEvents = I2CRXTX;
    // "Fake" write operation to send the register address we want to read
    I2CRXTX = I2C ID EDR | g uiI2CDevSelect | IC2 WRITE;
    // Wait for TX Buffer to be empty
    while((I2CSTATUS & 0x01) == 0);
    // Make sure we received an ACK and that there was no error during transfer
    if (I2CSTATUS & 0xC0)
    {
        return RFALSE;
        // Send Read Address
    I2CRXTX = EDR REG EVT CODE;
   while(!(I2CSTATUS & 0x08)); // Wait for I2C to be Idle: STOP
// Make sure we received an ACK and that there was no error during transfer
    if (I2CSTATUS & 0xC0)
    {
        return RFALSE;
    }
    // Start the Event Streaming by sending the read command.
    // Make sure we will ACK the Data we will receive
    I2CCONFIG &= 0xFD;
    // Dummy Read to clear the I2CRXAVF
    *a puiEvents = I2CRXTX;
    // Now, send the Read command
    I2CRXTX = I2C ID EDR | g uiI2CDevSelect | IC2 READ;
                 Rليم_
= 0;
-
    l uiCurEvent
                   = RFALSE;
    1 bEventFF
    while ((l_bEventFF == RFALSE) &&
           (l_uiCurEvent < a_uiNbEvents))</pre>
       l bEventFF = RTRUE;
        /\overline{/} Receive 8 bytes per Event
        for (l uiCurData = 0; l uiCurData < 8; l uiCurData++)</pre>
        {
            // Wait for Data to comeback
           while(!(I2CSTATUS & 0x02));
           *l puiVufPrt = I2CRXTX;
            if (*l_puiVufPrt != 0xFF)
               l bEventFF = RFALSE;
            l_puiVufPrt++;
        l uiCurEvent++;
```

```
Rev. 4.0 (EOL)
July 2010
```

```
}
   // Stop the streaming
   I2CCONFIG \mid = 0x02; // Wait for I2C to be Idle and generate a STOP
   while(!(I2CSTATUS & 0x08));
   if ((l_uiCurEvent != a_uiNbEvents) ||
      (l_bEventFF
                 == RTRUE))
   {
      return RFALSE;
   }
   return RTRUE;
}
// Function:
                Delay
// Description:
                Wait x milliseconds
// Parameters:
                ruint16 uiDelayMs: Delay, in ms
// Return value:
                None
                Calibrated for VRS51L2070 running at 40MHZ.
// Remarks:
11
                This function uses Timer0
void
                            oirect. Replacement
Delay(ruint16 a uiDelayMs)
{
   ruint16 l_uiDelayLoop = a_uiDelayMs;
   // Enable Timer 0
   PERIPHEN1 |= 0x01;
   while ( l uiDelayLoop > 0)
   {
      TOTICLKCFG &= 0xF0;
      // Timer0 reload value for 1ms @ 40Mhz
      TH0 = 0x63;
      TL0 = 0xC0;
      // Start Timer0
      TOCON = 0x04;
      // Wait for timer0 overflow
      while (!(TOCON & 0x80));
      // Stop Timer 0
      TOCON = 0x00;
      l_uiDelayLoop--;
   }
   // Disable Timer0
   PERIPHEN1 & OxFE;
}
```

Include File:

```
11
// $Date: 2008-03-10 13:04:27 -0400 (Mon, 10 Mar 2008) $
// $Rev: 249 $
// $Author: smalo $
// $HeadURL: file:...fm6124.h $
11
11
             This file contains C functions declarations and defines
// Description:
//
              for the Ramtron Event Data Recorder (EDR) FM6124.
//
// Remarks:
11
// Copyright (C) 2008 Ramtron International Corporation
11
#ifndef FM6124 H
#define FM6124 H
// Include and defines
// Ramtron basic type definitions
#include "ramtron_types.h"
// I2C Ids of the FM6124 \,
// The first byte of any I2C transaction contains the Slave Module Id
// Bits 7:1 are used for the Id
// Bit 0 indicates if the command is Read (1) or a Write (0)
#define I2C_ID_EDR
#define I2C_ID_FM
                   0xA0
                   0xD0
#define IC2 WRITE
                   0 \times 00
#define IC2_READ
                   0x01
// Type Definitions
THUTT
typedef struct SBCDDate
  ruint8 uiSeconds;
                           jirect
  ruint8
        uiMinutes;
  ruint8 uiHours;
  ruint8 uiDay;
  ruint8
        uiDate;
  ruint8 uiMonth;
  ruint8
        uiYear;
};
typedef struct SBCDAlarmDate
  ruint8
        uiSeconds:
  ruint8
        uiMinutes;
        uiHours;
  ruint8
  ruint8
        uiDate;
   ruint8
        uiMonth;
};
typedef struct SEvent
                uiEventCode;
  ruint8
  struct SBCDDate
                oBCDDate;
};
```

// FM6124 Enums enum EventType {

{			
	EV_PIN0_FALL	=	0x08,
	EV_PIN0_RISE	=	0x09,
	EV_PIN1_FALL	=	0x0A,
	EV_PIN1_RISE	=	0x0B,
	EV_PIN2_FALL	=	0x0C,
	EV_PIN2_RISE	=	0x0D,
	EV_PIN3_FALL	=	0x0E,
	EV_PIN3_RISE	=	0x0F,
			0x10,
	EV_PIN4_RISE	=	0x11,
	EV_PIN5_FALL	=	0x12,
	EV_PIN5_RISE	=	0x13,
	EV_PIN6_FALL	=	0x14,
	EV_PIN6_RISE	=	0x15,
	EV_PIN7_FALL	=	0x16,
	EV_PIN7_RISE	=	0x17,
	EV_PIN8_FALL	=	0x18,
	EV_PIN8_RISE	=	0x19,
	EV_PIN9_FALL	=	0x1A,
	EV_PIN9_RISE	=	0x1B,
	EV_PIN10_FALL		,
	EV_PIN10_RISE		
	EV_PIN11_FALL		,
	EV_PIN11_RISE	=	0x1F,
};			

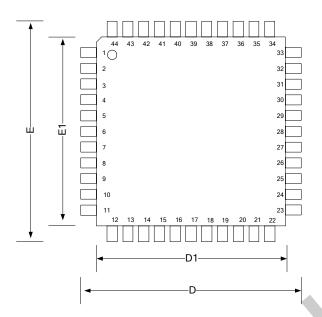
rectReplacement // FM6124 Event Data Recorder (EDR) Registers enum EDRRegisters

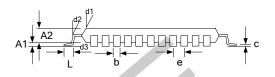
EDR_REG_RTC $= 0 \times 00,$ EDR_REG_CAL $= 0 \times 0 1$, EDR REG RTC SECS $= 0 \times 02$, EDR_REG_RTC_MINS = 0x03,EDR REG RTC HOURS = 0x04,EDR REG RTC DAY $= 0 \times 05$, EDR_REG_RTC_DATE $= 0 \times 06$, EDR_REG_RTC_MONTH EDR_REG_RTC_YEAR $= 0 \times 07$, 0x08, 0x09, EDR_REG_WD_FLAGS = EDR REG WD RESTART $= 0 \times 0 A$, EDR REG WD CTRLO $= 0 \times 0 B$, EDR REG WD CTRL1 $= 0 \times 0 C$, EDR REG CNT CTRL $= 0 \times 0 D,$ $= 0 \times 0 E,$ EDR REG CNT LSB 0x0E, EDR REG CNT MSB $= 0 \times 0 F$, EDR_REG_SN_BYTE0 $= 0 \times 10$, EDR_REG_SN_BYTE1 0x11, = 0x12, EDR_REG_SN_BYTE2 = EDR REG SN BYTE3 = 0x13,EDR REG SN BYTE4 = 0x14,EDR_REG_SN_BYTE5 = 0x15,EDR REG SN BYTE6 = 0x16,EDR_REG_SN_BYTE7 = 0x17,EDR REG CC = 0x18, EDR REG ALM SECS = 0x19,EDR_REG_ALM_MINS = 0x1A,EDR_REG_ALM_HOURS = 0x1B,EDR_REG_ALM_DATE $= 0 \times 1C$, EDR REG ALM MONTH = 0x1D, EDR_REG_BUFFER_CTRL = 0x20,

```
EDR REG PIN INT A
                                 = 0x21,
    EDR REG PIN INT B
                                 = 0x22,
    EDR REG PIN RF A
                                 = 0x23,
    EDR_REG_PIN_RF_B
EDR_REG_PIN_EE_A
                                = 0x24,
                                = 0x25,
    EDR REG PIN EE B
                                 = 0x26,
    EDR_REG_PIN_SNAP
                                 = 0x27,
    EDR_REG_PIN_STATE_A
EDR_REG_PIN_STATE_B
                                 = 0x28,
                                 = 0x29,
    EDR_REG_EVENT_COUNT_LSB = 0x2A,
    EDR_REG_EVENT_COUNT_MSB = 0x2B,
    EDR_REG_EVT_CODE
                                 = 0x2C,
    EDR REG EVT SECS
                                 = 0 \times 2D,
    EDR REG EVT MINS
                                = 0 \times 2 E
    EDR REG EVT HOURS
                                 = 0x2F,
    EDR_REG_EVT_DAY
                                 = 0 \times 30,
    EDR_REG_EVT_DATE
                                = 0x31,
    EDR REG EVT MONTH
                                 = 0x32,
    EDR_REG_EVT_YEAR
                                 = 0x33,
};
// Buffer Control Register Bits
enum
                                                      ct Replacement
{
    EDR BC CMD BITS
                                = 0 \times 0 F,
    EDR_BC_DIR_BIT
                                 = 0x10,
    EDR_BC_ERR_BIT
EDR_BC_VAR_BITS
                                 = 0x20,
                                 = 0 \times C0,
    EDR_BC_CMD_NOTHING
EDR_BC_CMD_GET
EDR_BC_CMD_GET_KEEP
                                 = 0 \times 0.
                                 = 0x1,
                                 = 0x2,
    EDR_BC_CMD_STREAM
                                 = 0x3,
    EDR_BC_CMD_STREAM_KEEP
EDR_BC_CMD_SKIP
                                 = 0x4,
                                 = 0x5,
    EDR BC CMD FIRST
                                 = 0x6,
    EDR_BC_CMD_LAST
                                 = 0x7
    EDR BC CMD EB SIZE
                                 =
                                   0x8,
    EDR_BC_VAR_4000_EVENTS
                                 = 0 \times 00,
    EDR_BC_VAR_3000_EVENTS
EDR_BC_VAR_2000_EVENTS
                                 = 0x40,
                                 =
                                   0x80,
    EDR_BC_VAR_1000_EVENTS
                                 = 0 \times C0,
};
// EDR_REG_PIN_INT_A Bits
enum
{
                                            0x10,
    EDR_PIN_INT_A_BHF_BIT
                                          =
                                          = 0x20,
    EDR PIN INT A B75F BIT
    EDR_PIN_INT_A_BF_BIT
                                            0x40,
                                          2
    EDR PIN INT A CLEAR BIT
                                          =
                                             0x80,
    EDR_PIN_INT_A_CLEAR_BIT_MASK
                                             0x7F,
};
#endif // FM6124_H
```

MECHANICAL DRAWING

QFP-44 Package





		—D1—		
	•	D		
ΓΔΒΙ Ε 1) DIME	NSIONS OF QFP-44 PACKAGI	=9		
Symbol	Description	Dimension (mm)	Tolerance (mm, º) / Notes	el'
D	Footprint	13.2	+/- 0.25	
D1	Body size	10	+/- 0.10	
Е	Footprint	13.2	+/- 0.25	
E1	Body size	10	+/- 0.10	
A1	Stand-off	0.25	Max	
A2	Body thickness	2.00		
L	Lead Length	0.88	+0.15 / -0.10	
b	Lead width	0.35	+/- 0.05	
С	L/C thickness	0.17	Max	
е	Lead pitch	0.8		
d1	Body edge angle	10°		
d2	Lead angle	6°	+/- 4°	
d3	Lead angle	0° to 7°		

ORDERING INFORMATION

Device Number	Total F-RAM Memory Size	Recorder Events F-RAM	User Data F-RAM Size	Package	Voltage	Temperature Range
FM6124-QG	32KB	1000 - 4000	0-24KB	QFP-44	3.0V to 3.6V	-40°C to +85°C

Revision History

Revision	Date	Summary	
1.0	04/11/2008	Initial release.	
4.0	7/20/2010	End of Life. No direct replacement.	

o Replacement

Rev. 4.0 (EOL) July 2010